

关于线程同步的一些方法计算机等级考试 PDF转换可能丢失
图片或格式，建议阅读原文

https://www.100test.com/kao_ti2020/645/2021_2022__E5_85_B3_E4_BA_8E_E7_BA_BF_E7_c97_645035.htm 线程是进程内一个相对独立的、可调度的执行单元。一个应用可以有一个主线程，一个主线程可以有多个子线程，子线程还可以有自己的子线程，这样就构成了多线程应用了。由于多个线程往往会同时访问同一块内存区域，频繁的访问这块区域，将会增加产生线程冲突的概率。一旦产生了冲突，将会造成不可预料的结果（该公用区域的值是不可预料的）可见处理线程同步的必要性。注意：本文中出现的所有代码都是用DELPHI描述的，调试环境为Windows me，Delphi 6。其中所涉及的Windows API函数可以从MSDN获得详细的文档。首先引用一个实例来引出我们以下的讨论，该实例没有采取任何措施来避免线程冲突，它的主要过程为：由主线程启动两个线程对letters这个全局变量进行频繁的读写，然后分别把修改的结果显示到ListBox中。由于没有同步这两个线程，使得线程在修改letters时产生了不可预料的结果。大家可以看看它运行的结果。ListBox中的每一行的字母都应该一致，这就是线程冲突产生的结果。当两个线程同时访问该共享内存时，一个线程还未对该内存修改完，另一个线程又对该内存进行了修改，由于写值的过程没有被串行化，这样就产生了无效的结果。可见线程同步的重要性。以下是本例的代码 unit.pas文件

```
unit Unit1;
interface
uses Windows, Messages, SysUtils, Variants,
Classes, Graphics, Controls, Forms, Dialogs, StdCtrls;
//定义窗口类
type TForm1 = class(TForm)
  ListBox1: TListBox;
  ListBox2:
```

```
TListBox. Button1: TButton. procedure Button1Click(Sender:
TObject). private { Private declarations } public { Public declarations
} end. //定义线程类 type TListThread=class(TThread) private
Str:String. protected procedure AddToList.//将Str加入ListBox组件
Procedure Execute.override. public LBox:TListBox. end. //定义变
量 var Form1: TForm1.
Letters:String=\AAAAAAAAAAAAAAAAAAAAA\//全局变量
implementation {$R *.dfm} //线程类实现部分 procedure
TListThread.Execute. var I,J,K:Integer. begin for i:=0 to 50 do begin
for J:=1 to 20 do for K:=1 to 1000 do//循环1000次增加产生冲突
的几率 if letters[j] 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
```