

使用内存映象实现进程间全局变量计算机等级考试 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__E4_BD_BF_E7_94_A8_E5_86_85_E5_c97_645029.htm 使用内存映象实现进程间全局变量 编程时在同一程序中使用全局变量给我们带来了许多方便，大多数程序几乎都离不开全局变量。那么不同的程序之间呢？我们怎样在不同的程序（进程）之间共享数据呢？事实上，你可能已经注意到在不同的进程间使用全局变量在有些场合还是很有必要，这是一项相当有用的技术，基于其中特殊情况我们的确需要在我们开发的不同的程序间共享数据。比如有些软件你会发现安装后不止一个可执行文件，而且在运行时任务栏也不止出现一个进程，但在使用中你发现不同的进程竟向一个单独运行程序一样，效果很好。接下来我们仔细研究一下该技术：要在不同进程间共享全局变量，应该首先想到我们可以通过不同的进程在内存中的同一块存储区域读取数据，那里就象一个公共场合一样，我们把这样的区域叫做内存映象文件。通过windows api我们可以很容易的在内存中创建这么一个区域并轻易的读取它。这里需要用到两个关键的api函数：CreateFileMapping和MapViewOfFile,前者在内存中创建一个内存映象区，它需要一些参数包括这块区域的读写权限，唯一区域名称以及数据大小（这些参数将在下面代码的注释中给予说明，更多的参数说明请参看win sdk的帮助文件或查看msdn）；后者返回一个指向这块区域的指针以便在程序中调用。下面我们将通过一个简单的例子来说明，这个例子由两个工程组成，一个通过一个按钮负责向内存映象区写入一个长度为11个字符的字符

串（‘ how are you ‘，通过一个edit读入，如果edit中的字符数大于了11个程序将可能出错），另一个则通过一个按钮读出这个字符串，写入一个edit中。下面是部分的代码（为了节省篇幅，加上程序很简单，有些不会影想你阅读的代码没有贴出）：我们首先在第一个工程（窗口取名为SetValue）的单元的接口部分声明一个指向即将创建的内存区域的句柄和一个用来向这个区域写数据的指针变量：var

hFileMapping:HWND.//指向内存区域的句柄来源：考试大

SetString:pchar.//这里应该使用标准的windows字符串而非delphi特有的string类型

这样我们可以在按钮的单击事件中编写相应的代码：

```
procedure TForm1.Button1Click(Sender: TObject). begin
```

```
hFileMapping:=CreateFileMapping($FFFFFFFF,nil,Page_ReadWrite,0,11,\MapString\). // $FFFFFFFF表示使用虚拟文件，注意两个工程中应该使用同一个唯一的名称MapString if hFileMapping=0 then raise Exception.Create(\Error creating map file!\).
```

```
SetString:=MapViewOfFile(hFileMapping,File_Map_Write,0,0,11). strcpy(SetString,pchar(Edit1.Text)). //注意这里应该使用拷贝函数而不是直接赋值
```

```
SetString:=pchar(Edit1.Text) end.
```

当然，我们还可以在这个主工程中编写一个释放内存映象的过程：

```
procedure TForm1.Button2Click(Sender: TObject). begin
```

```
UnMapViewOfFile(SetString). if hFileMapping 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
```