

计算机二级DELPHI函数:利用Delphi建立精确计数器计算机二级考试 PDF转换可能丢失图片或格式，建议阅读原文

[https://www.100test.com/kao\\_ti2020/581/2021\\_2022\\_\\_E8\\_AE\\_A1\\_E7\\_AE\\_97\\_E6\\_9C\\_BA\\_E4\\_c97\\_581853.htm](https://www.100test.com/kao_ti2020/581/2021_2022__E8_AE_A1_E7_AE_97_E6_9C_BA_E4_c97_581853.htm)

在Windows中的很多场合下编程（例如工业控制、游戏）中需要比较精确的记时器，本文讨论的是在Delphi下实现记时器的若干方法以及它们的精度控制问题。在Delphi中最常用的是Timer控件，它的设置和使用都非常方便，理论上它的记时精度可以达到1ms(毫秒)。但是众所周知的，实际上Timer在记时间隔小于50ms之下是精度是十分差的。它只适用于对于精度要求不太高的场合。这里作者要介绍的是两种利用Windows API函数实现精确记时的方法。第一中方法是利用高性能频率记数（作者本人的称呼）法。利用这种方法要使用两个API函数QueryPerformanceFrequency和QueryPerformanceCounter。QueryPerformanceFrequency函数获得高性能频率计数器的震荡频率。调用该函数后，函数会将系统频率计数器的震荡频率（每毫秒）保存到一个LargeInteger中。不过利用该函数在几台机器上做过试验，结果都是1193180。读者朋友可以在自己的机器上试一下QueryPerformanceCounter函数获得系统频率计数器的震荡次数，结果也保存到一个LargeInteger中。很显然，如果在计时中首先使用QueryPerformanceFrequency获得高性能频率计数器每毫秒的震荡次数，然后在计时开始时使用QueryPerformanceCounter函数获得当前系统频率计数器的震荡次数。在计时结束时再调用QueryPerformanceCounter函数获得系统频率计数器的震荡次数。将两者相减，再将结果除以频率计数器每毫秒的震荡次数，就可以获得某一事件经

过的准确时间。(次数除以频率等于时间) 另外的一种精确记时器的功能是利用多媒体记时器函数(这也是作者的定义,因为这个系列的函数是在Winmm.dll中定义并且是为媒体播放服务的)。实现多媒体记时器首先要使用timeSetEvent函数建立计时事件。该函数在Delphi中的mmsystem.pas中有定义,定义如下: function timeSetEvent(uDelay, uResolution: UINT. lpFunction: TFNTTimeCallBack. dwUser: DWORD. uFlags: UINT): MMRESULT. stdcall 函数定义中参数uDelay定义延迟时间,以毫秒为单位,该参数相当于Timer控件的Interval属性。参数uResolution定义记时精度,如果要求尽可能高的精度,要将该参数设置为0;参数lpFunction定义了timeSetEvent函数的回调函数。该函数相当于一个定时中断处理函数,每当经过一个uDelay长度的时间间隔,该函数就会被调用,编程者可以在该函数中加入相应的处理语句。参数dwUser定义用户自定义的回调值,该值将传递给回调函数。参数uFlags定义定时类型,如果要不间断的记时,该值应设置为1。如果函数调用成功,在系统中建立了一个多媒体记时器对象,每当经过一个uDelay时间后lpFunction指定的函数都会被调用。同时函数返回一个对象标识,如果不再需要记时器则必须要使用timeKillEvent函数删除记时器对象。由于Windows是一个多任务的操作系统,因此基于API调用的记时器的精度都会受到其它很多因素的干扰。到底这两中记时器的精度如何,我们来使用以下的程序进行验证:设置三种记时器(Timer控件、高性能频率记数、多媒体记时器)。将它们的定时间隔设置为10毫秒,让它们不停工作直到达到一个比较长的时间(比如60秒),这样记时器的误差会被累计下来,然后同实际经

过的时间相比较，就可以得到它们的精度。下面是具体的检测程序。

```
unit Unit1. interface uses Windows, Messages, SysUtils,
Classes, Graphics, Controls, Forms, Dialogs, StdCtrls,
ExtCtrls,mmSystem. type TForm1 = class(TForm) Edit1: TEdit.
Edit2: TEdit. Edit3: TEdit. Button1: TButton. Button2: TButton.
Timer1: TTimer. procedure FormCreate(Sender: TObject).
procedure Button1Click(Sender: TObject). procedure
Timer1Timer(Sender: TObject). procedure Button2Click(Sender:
TObject). private { Private declarations } public { Public declarations
} end. var Form1: TForm1. actTime1,actTime2:Cardinal.
smmCount,sTimerCount,sPCount:Single. hTimeID:Integer.
iTen:Integer. proTimeCallBack:TFNTTimeCallBack. procedure
TimeProc(uTimerID, uMessage: UINT. dwUser, dw1, dw2:
DWORD) stdcall. procedure proEndCount. implementation {$R
*.DFM} //timeSetEvent的回调函数 procedure proEndCount.
begin actTime2:=GetTickCount-actTime1. Form1.Button2.Enabled
:=False. Form1.Button1.Enabled :=TRUE. Form1.Timer1.Enabled
:=False. smmCount:=60. sTimerCount:=60. spCount:=-1.
timeKillEvent(hTimeID). end. 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](http://www.100test.com)