

Delphi编程实现Ping操作计算机等级考试 PDF转换可能丢失图片或格式，建议阅读原文

https://www.100test.com/kao_ti2020/641/2021_2022_Delphi_E7_BC_96_E7_c97_641455.htm 使用过网络的用户都熟悉“Ping”这个指令，它是一个DOS下的可执行文件，一般用它来检查网络连接的好坏程度。其基本原理是利用TCP/IP协议包中ICMP协议中的一个功能，即向所指定的计算机发送一个请求，收到请求的计算机返回一个应答，借此来判断该计算机是否在网上运行或者检查网络连接是否稳定可靠。在Ping程序执行过程中，双方计算机所消耗的资源都很少，因此，它是一个非常实用的工具。我们可以通过编程来实现“Ping”操作，对其加以改进，使之具有Windows的界面风格，显示比DOS更加直观。首先，对编程中需要的动态链接库作一简要说明：在Windows的System目录下，你可以找到Icmp.dll文件，该动态链接库提供了ICMP协议的所有功能，我们的编程就建立在对该动态链接库的调用上。Icmp.dll文件内的调用函数说明如下：1、IcmpCreateFile 打开一个句柄，通过该句柄你可以发送ICMP的请求回送报文。2、IcmpCloseHandle 关闭你通过IcmpCreateFile函数打开的句柄。3、IcmpSendEcho 通过你打开的句柄发送ICMP请求，在超时或应答报文接收后返回。其参数基本上和它的帧结构一致，可参看下面的程序部分，其具体含意你可以参看有关ICMP协议的书籍。初步了解了上述的三个函数后，我们就可以开始编程了。首先，我们的程序运行后应该有如图1所示的基本功能。为此，我们可先在Delphi的窗口中放入右上图中所示的控件，如按钮、编辑框和文本显示框等。然后，在程序的开始部分（FormCreate

) 对WinSocket进行初始化，其作用是申明使用的版本信息，同时调入Icmp.dll库。

```

type PIPOptionInformation =
^TIPOptionInformation. TIPOptionInformation = packed record
TTL: Byte. TOS: Byte. Flags: Byte. OptionsSize: Byte. OptionsData:
PChar. end. PIcmpEchoReply = ^TIcmpEchoReply.
TIcmpEchoReply = packed record Address: DWORD. Status:
DWORD. RTT: DWORD. DataSize: Word. Reserved: Word. Data:
Pointer. Options: TIPOptionInformation. end. TIcmpCreateFile =
function: THandle. stdcall. TIcmpCloseHandle =
function(IcmpHandle: THandle): Boolean. stdcall. TIcmpSendEcho
= function(IcmpHandle:THandle. DestinationAddress: DWORD.
RequestData: Pointer. RequestSize: Word. RequestOptions:
PIPOptionInformation. ReplyBuffer: Pointer. ReplySize: DWord.
Timeout: DWord ): DWord. stdcall. TMyPing = class(TForm)
Panel1: TPanel. Label1: TLabel. PingEdit: TEdit. ExeBtn: TButton.
Button2: TButton. Button3: TButton. StatusShow: TMemo.
procedure Button3Click(Sender: TObject). procedure
FormCreate(Sender: TObject). procedure ExeBtnClick(Sender:
TObject). private { Private declarations } hICMP: THANDLE.
IcmpCreateFile : TIcmpCreateFile. IcmpCloseHandle:
TIcmpCloseHandle. IcmpSendEcho: TIcmpSendEcho. public {
Public declarations } end. procedure TMyPing.FormCreate(Sender:
TObject). var WSAData: TWSAData. hICMPdll: HMODULE.
begin WSASStartup($101, WSAData). // Load the icmp.dll stuff
hICMPdll := LoadLibrary(\ ' icmp.dll\ '). @ICMPCreateFile :=
GetProcAddress(hICMPdll, \ ' IcmpCreateFile\ ').

```

```

@IcmpCloseHandle := GetProcAddress(hICMPdll, \
' IcmpCloseHandle\ '). @IcmpSendEcho :=
GetProcAddress(hICMPdll, \ ' IcmpSendEcho\ '). hICMP :=
IcmpCreateFile. StatusShow.Text := \ '. StatusShow.Lines.Add(\ '
目的IP地址 字节数 返回时间(毫秒)\ '). end. 接下来，就要进
行如下所示的Ping操作的实际编程过程了。 procedure
TMyPing.ExeBtnClick(Sender: TObject). var
IPOpt:TIPOptionInformation.// IP Options for packet to send
FIPAddress:DWORD. pReqData,pRevData:PChar.
pIPE:PIcmpEchoReply.// ICMP Echo reply buffer FSize: DWORD.
MyString:string. FTimeout:DWORD. BufferSize:DWORD. begin if
PingEdit.Text gt. \ ' then begin FIPAddress :=
inet_addr(PChar(PingEdit.Text)). FSize := 40. BufferSize :=
SizeOf(TICMPEchoReply) FSize. GetMem(pRevData,FSize).
GetMem(pIPE,BufferSize). FillChar(pIPE^, SizeOf(pIPE^), 0).
pIPE^.Data := pRevData. MyString := \ ' Hello,World\ ' .
pReqData := PChar(MyString). FillChar(IPOpt, Sizeof(IPOpt), 0).
IPOpt.TTL := 64. FTimeout := 4000. IcmpSendEcho(hICMP,
FIPAddress, pReqData, Length(MyString), @IPOpt, pIPE,
BufferSize, FTimeout). if pReqData^ =
pIPE^.Options.OptionsData^ then begin
StatusShow.Lines.Add(PChar(PingEdit.Text) \ ' \ '
IntToStr(pIPE^.DataSize) \ ' \ ' IntToStr(pIPE^.RTT)). end.
FreeMem(pRevData). FreeMem(pIPE). end end. 通过上面的编程
，我们就实现了Ping功能的界面操作。实际上，ICMP协议的功能
还有很多，都可以通过对Icmp.dll的函数调用来实现。

```

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