**Windows黑客编程基础**

上面的题目带有“黑客”两个字，请大家别误会了，其实没有多少是讲黑客的，这完全是一篇菜鸟级的编程杂谈，如果您已是高手，就不必在此浪费时间了 。

前几天在网上看了“病毒”兄写的《WIN下编程须知》一文，觉得在编程方面要写出一篇适合初学者们看的入门级文章的确很重要，可惜病毒兄只在该文里介绍了线程、消息、句柄等几个基本概念。很多初学者看了对编程还是感到很迷惑，一个从来没有写过程序的人如何入门？如何在短时间内写出自己的程序来？笔者带着这些问题写了这篇文章。这也是笔者在学习编程的初期所遇到的困惑，在此根据笔者的个人理解将其整理成文，希望能引起广大菜鸟们的共鸣，对初学者们有所帮助。

从理论上说，任何一门语言都可以在任何一个系统上编程，只要找到该系统提供的“接口”和对系统内部机制有深入的了解就可以了，至少我是这么认为的。正如c语言可以在windows下编程，也同样可以在Linux上大放异彩一样。

编程是一项很繁杂的工作，除了应用编程工具之外，了解系统本身内部工作机理非常重要，这是你写出稳定兼容的程序所必不可少的前提条件。你要在哪一种系统上编程就要对该系统的机制进行研究，至少你应该知道一个程序在那个系统上是如何运行的。

　　一、了解Windows 内部机制

　　Windows 是一个“基于事件的，消息驱动的”操作系统。

在Windows下执行一个程序，只要用户进行了影响窗口的动作（如改变窗口大小或移动、单击鼠标等）该动作就会触发一个相应的“事件”。系统每次检测到一个事件时，就会给程序发送一个“消息”，从而使程序可以处理该事件。每个Windows 应用程序都是基于事件和消息的，而且包含一个主事件循环，它不停地、反复地检测是否有用户事件发生。每次检测到一个用户事件，程序就对该事件做出响应，处理完再等待下一个事件的发生。Windows 下的应用程序不断地重复这一过程，直至用户终止程序，用代码来描述实际上也就是一个消息处理过程的while循环语句。

　　下面便简单介绍一下与 Windows 系统密切相关的几个基本概念：

　　⒈窗口：这是我要说的第一个概念。似乎是地球人都知道的事儿了，窗口是Windows本身以及Windows 环境下的应用程序的基本界面单位，但是很多人都误以为只有具有标题栏、状态栏、最大化、最小化按钮这样标准的方框才叫窗口。其实窗口的概念很广，例如按钮和对话框等也是窗口哦，只不过是一种特殊的窗口罢了。从用户的角度看，窗口就是显示在屏幕上的一个矩形区域，其外观独立于应用程序，事实上它就是生成该窗口的应用程序与用户间的直观接口；从应用程序的角度看，窗口是受其控制的一部分矩形屏幕区。应用程序生成并控制与窗口有关的一切内容，包括窗口的大小、风格、位置以及窗口内显示的内容等。用户打开一个应用程序后，程序将创建一个窗口，并在那里默默地等待用户的要求。每当用户选择窗口中的选项，程序即对此做出响应。

⒉程序：通常说的程序都是指一个能让计算机识别的文件，接触得最多的便是.exe型的可执行文件，这个不难理解。

⒊进程：说到进程，学过《操作系统》的人都很清楚，所谓进程就是应用程序的执行实例（或称一个执行程序）需要注意的是：进程是程序动态的描述，而上面说到的程序是静态的描述，两者有本质的区别。举个例子，从网上Down了一个瑞星杀毒软件到C盘但没有运行，那个.exe 可执行文件叫做程序，它是一个二进制码的文件。一旦双击了exe文件图标运行程序，那个“正在运行着的瑞星杀毒”便称为进程，它在双击的那一刻被系统创建，当你关机或者在任务栏的图标上单击鼠标右键选“退出”时，进程便消亡，彻底结束了生命。进程经历了由“创建”到“消亡”的生命期，而程序自始至终存在于你的硬盘上，不管你的机器是否启动。

⒋线程：线程是进程中的一个执行单元，同一个进程中的各个线程对应于一组CPU指令、一组CPU寄存器以及一堆栈。进程本来就具有动态的含义，然而实质上是通过线程来执行体现的，从这个意义上说，Windows 中进程的动态性意义已经不是很明显了，只算是给程序所占的资源划定一个范围而已（个人观点，纯属个人理解，不必引起争议！），真正具有动态性意义的是线程。以前在大二学习操作系统课的时候就有个同学跟笔者提起这点，笔者还跟他驳得面红耳赤呢！现在想想，觉得很有道理，不得不佩服那位同学对Windows内部机制了解得如此清楚。之所以在此花那么多的篇幅说线程，是因为下面将要介绍到多线程编程技巧，如果不理解这点，那就很难应用到实践上，希望大家明白。

⒌消息：我们几乎做每一个动作都会产生一个消息，在用鼠标指点江山的今天，鼠标被移动会产生WM\_MOUSEMOVE消息，鼠标左键被按下会产生WM\_LBUTTONDOWN的消息，鼠标右键按下便产生WM\_RBUTTONDOWN消息等等。所有的这些都可以通过GetMessage，SendMessage等函数得到,以后的操作中我们会经常接触到这些函数。

⒍事件：何谓事件？从它的字面意思我们就可以明白它的含义，如在程序运行的过程中改变窗口的大小或者移动窗口等，都会触发相应的“事件”。

⒎句柄：单单一个“柄”字便可以解释它的意思了，我们天气热摇扇子的时候只要抓住扇柄便可以控制整个扇子的运动了，在程序中也差不多是这个意思。通常一个句柄就可以传递我们所要做的事情。有经验的读者肯定清楚，编写程序总是要和各种句柄打交道的，句柄是系统用来标识不同对象类型的工具，如窗口、菜单等，这些东西在系统中被视为不同类型的对象，用不同的句柄将他们区分开来。看看C++ 教材中是如何给句柄下定义的：“在Win32里，句柄是指向一个无值型对象（void \*）的指针，是一个4字节长的数据”。虽然我对它的本质是什么还是很迷惑，但我知道句柄并不是一个真正意义上的指针。从结构上看，句柄的确是一个指针，尽管它没有指向用于存储某个对象的内存位置（很多书都这么说，这正是我的迷惑所在），而实际上句柄指向的是一个包含了对该对象进行的引用的位置。在编程时，只要抓住了对象的句柄就可以对该对象进行操作了（我在《一个简单木马程序的编写与伪装策略》中说到的对QQ密码的截获就是要找到QQ登陆窗口的句柄后才开始截密行动的）。下面再举个例子来说明句柄的运用：编一个程序，使QQ登陆窗口的号码框和密码框均变黑，相关代

码及解释：

void \_\_fastcall Tform1::formCreate(TObject \*Sender)

{

HWND hCurWindow，HC,HE;//定义三个窗口句柄变量，hCurWindow用于存放QQ用户登陆窗口的句柄，HC、HE分别存放

//号码框和密码框的句柄。

if((hCurWindow= FindWindow(NULL,"QQ用户登录"))!=0||(hCurWindow=FindWindow(NULL,"OICQ用户登录"))!=0)

{//很明显，调用FindWindow（）函数去获得QQ登陆窗口的句柄

String str;

str.sprintf("0x%x",hCurWindow);

}

TCHAR wClassName[255];//类名变量

HC=GetWindow(hCurWindow, GW\_CHILD);//得到号码框的句柄

HE=GetWindow(HC, GW\_HWNDNEXT);//接着得到密码框的句柄

GetClassName(HE, wClassName, sizeof(wClassName));//得到类名

GetClassName(HC, wClassName, sizeof(wClassName));//得到类名

EnableWindow(HE,false);//使窗口失效

EnableWindow(HC,false);//使窗口失效

}

　　以上代码在C++ Builder下编译通过，只要运行次程序，QQ登陆窗口的号码框和密码框马上变黑色，无非是EnableWindow（）函数所起的作用。你还可以添加一个Timer控件，将上面的代码copy到void \_\_fastcall Tform1::Timer1Timer

(TObject \*Sender)函数中，并在后边加上这一句代码：

SendMessage(hCurWindow,WM\_CLOSE,0,0); 使QQ一启动就关闭，让别人永远也用不了QQ，挺有趣儿的哦:).

⒏API与SDK：API是英文 Application Programming Interface 的简称，意为“应用程序接口”，泛指系统为应用程序提供的一系列接口函数。其实质是程序内的一套函数调用，在编程的时候可以直接调用，而不必知道其内部实现的过程，只知道它的原型和返回值就可以了，此外，手头经常放着一本“Windows API大全”之类的书也是必不可少的，不然你根本不知道哪些API是干什么用的，瞎编也编不出什么东西来。在后面我们会介绍调用API编程的例子，调用API编程工作虽然烦琐，但由于API函数都被封装在dll库里，程序只有在运行的时候才调用的，因此程序的体积小而且运行效率高。SDK是英文 Software Development Kit 的缩写,指“软件开发工具包”，在防火墙的设计中就经常涉及到SDK。有关基本的概念就谈这些，那些C/C++的基本语法、什么是面向对象等知识请大家查阅相关的书，此类书籍各大书店已汗牛充栋，不再多叙。下面直接谈谈语种和编程工具的选择问题，这也是初学者们最迷惑的问题。

二、编程语言以及工具的选择：

　　从上面的介绍我们对Windows 有了进一步的了解，现在就该开始行动了，选择要学的语言和工具是第一步，而且是非常重要的一步工作，笔者建议一切以简单、易接受为原则，不然你会自信心大减的，何必偏要跟自己过不去自讨苦吃呢？在开始的时候很多人都感到迷惑，目前的编程语言那么多，有c、c++、c#、java、汇编、html等等，究竟学哪些好呢？最开始我该学什么呢？甚至有人将vc、c++ builder也列为两种不同的语言！这些都是对编程语言缺乏了解造成的。笔者开始的时候也犯过同样的错误，曾经给自己写过一份计划书：先学c语言，接着学c++、c#、java、汇编、vb、vc、c++ builder……，哪一种语言用多少时间去专攻等等，现在回想起来觉得多么的可笑！只要学得精，一门就够了。从实用的角度来讲，C++ 是最好的选择（个人意见，其实每一种语言都很好），而VC和C++ Builder是其相应开发工具的两大主流，笔者极力推荐初学者使用C++ Builder，因为很容易上手，如果一下子就用VC的话，也许会打击你的自信心：）。

　　三、谈谈促进编程能力提高的两个途径

　　如果你是一个黑客技术的狂热者的话，到雅虎去搜索黑客教程的时候就会发现，很多的中文教程在谈到如何进行黑客编程时，十有八九都会介绍以下两大最佳途径：一、读程序；二、写程序，并且都提出了教程作者的看法，下面我想谈谈这方面的个人观点。

　　⒈读程序：我将读程序放在前面是有原因的。在你没有阅读过一份完整的源代码之前，你别指望能写出有多好的程序来！这是对每一位初学者的忠告也是警告，而且必须具备一定的语言基础知识，这里的基础知识主要是指语法知识，最起码要能读懂别人的程序的每一行意思。有没有程序的设计思想，在这个时期并不重要，只要具备一定的语法基础就可以了，思想可以通过阅读完别人的源程序后分析得来。记得在大一学习C语言的时候，我们都很重视语法的学习，整天都看教材、做练习，而且赶在老师的讲课前预习，课后又复习巩固，将一些语法点记得滚瓜烂熟，可后来一到做课程设计的时候，坐在电脑面前简直是老鼠拖鸡蛋—

无从下手了，而且不断的问自己：“我平时的努力哪去了？语法都会了呀，怎么还是做不出程序来？”相信很多人都像笔者以前那样，错误地以为学会了语法就等于掌握了编程。编程的能力包括经验、技巧、耐心等几个因素，而并非想象中的那样简单，更不要以为编程就是简简单单的写程序！其实学一门语言并不需要刻意去记那些条条框框的语法，在看代码的时候，遇到了不明白的地方再去查相关的资料，一点一点补充基础知识再配合源程序的思路，这时的理解才是最深刻的，我可以肯定地说，这个时候对语法的接受程度绝对比你刚开始时的死记要强！读程序也不能单纯地读，要真正做到“俯而读，昂而思”。好的代码是百读不厌的，比如Shotgun的那道构造洪水Ping攻击的代码，我至少读了20遍。笔者喜欢将从网上搜集来的代码打印到纸上（尽管学校的打印费贵得要命，打一份代码就得花去十几块甚至几十块大洋~~~），然后边看边做好眉批，遇到一个新函数记下它的功能，一些忘记了的知识在旁边标出来，还可以写上对程序的看法等等。特别是遇到了一些新的API函数，最好标出来，对你以后编程的时候也许会用得着，最后别忘了分析一下程序的思路，这样对你以后编写类似的程序很有帮助的。

⒉写程序：问题可谈到点子上了，学那么多语言，读那么多程序最终还不是为了写程序，做出适合需要的软件来？“君子性非异也，善加于物也”，笔者认为一切从借鉴开始，先是修改别人的程序，等到有了一定的程度再写出属于自己的程序。刚开始写程序，不要奢望一下子写出很出色的程序来，“万丈高楼平底起”，编程贵在动手，只要你动手去写了，就算只有一句“printf(“Hello!”);”也是一次进步！此外，还要依照自身的能力循序渐进地写，开始的时候写一点功能简单的、篇幅短小的代码，力求简洁、完整，“麻雀虽小，但五脏俱全”，然后在此基础上进行扩充，一点一点添加功能，下面笔者摘录一位国内一流编程高手、“豪杰超级解霸”的作者梁肇新的编程心得，请大家看

看一个成功的程序员是如何写程序的，希望对广大菜鸟有所启发：写程序的方法：在Win98的环境中，先写主干，用最少的代码实现最基本的功能。然后一点点添加功能，每加一点都要调试。尽量少用动态分配、全局变量。充分利用操作系统直接提供的API。在Win98下调试通过之后，再在Win95下调试通过，然后是Win97，WindowsME，WinNT4.0。这样才能写出稳定、快速的程序。

　　给程序员的建议：1、不要急于求成，这样往往欲速不达。2、不要什么东西都想学，什么都没掌握。3、每天都要自我总结，分析自己的错误率和废码率，不断加强自我管理。4、代码格式很重要。代码要规范、严谨，效率要高。5、不要盲从简单的开发工具（这点笔者不是很同意，最起码要有一定的功底的人才敢这么说）。6、有了成果要公开，不要舍不得，不然很快会过时的（以上两段摘自《程序员》增值合订本2001.上册P18，请读者前往参考）。

参考书籍：

《Windows C 程序设计》，清华大学出版社

《超级解霸梁肇新》，《程序员》合订本

黑客编程的几个基本技巧

　　以下将要谈到的几个基本技巧很重要，虽然对于编程高手来说这是在玩小孩子把戏，但对于一位初学者，掌握以下几个技巧将为你的编程扫清道路，而且很容易编写出有趣的程序，培养你对编程的兴趣。

　　技巧⒈学会修改注册表。

相信大家都知道当浏览了一些网页恶意代码，IE标题、默认主页等被改得面目全非，这就是通过改动注册表来更改系统设置的例子。Windows中的注册表是个好东东，它是windows系统的灵魂，是许多软件记录数据的地方（当然也包括windows本身）。windows通过它记录大量的数据，然后在下一次启动时再读取相应的数据来设置系统。通过控制注册表就可以控制整个系统，所以很多的黑客程序都在注册表上动手脚（尤其是木马程序和作恶剧程序），学会修改注册表可以实现一些有趣而强大的功能。我们完全可以通过编程来操作注册表，达到与手动更改注册表编辑器产生一样的效果。“超级兔子”中的大部分功能就是通过修改注册表来完成的。操作注册表有专门的API函数，大家可以参考有关资料，下面笔者以C++ Builder为例说明如何在程序中操作注册表：

程序二：编程修改IE标题内容

新建一个工程，在Unit1.h文件中包含Registry单元：

#include

然后就可以在.cpp文件操作注册表了，接着来！在窗体的OnCreate()里加入以下代码（你可以在try{}里面加入任何操作注册表的代码）：

TRegistry\* Registry;

Registry = new TRegistry();创建一个TRegistry类型的对象Registry，用于修改注册表。

try{

Registry->RootKey = HKEY\_CURRENT\_USER;//设置主键，这是必不可少的，设置好主键后，就可以操作这个主

//键下所有的键值了。

if( Registry->OpenKey("Software\Microsoft\Internet Explorer\Main",FALSE))//调用OpenKey（）

//打开.

//括号里所指的键

{

Registry->WriteString("Window Title",”台湾是中国的一部分，世界上只有一个中国！”);

//调用WriteString（）往注册表里写入IE标题

Registry->CloseKey();//关闭该键

}

else

{//如果打开失败的话

Registry->CreateKey("Software\Microsoft\Internet Explorer\Main");//就调用CreateKey（）

//新建上述键

Registry->WriteString("Window Title","台湾是中国的一部分，世界上只有一个中国！");//再写入//IE标

//题内容

Registry->CloseKey();//最后关闭该键，这个也不能忽视，它跟上面的OpenKey成对使用的}//End of try

\_\_finally

{//要是出错，跳到这里处理

Registry->CloseKey();//关闭所要打开的键

delete Registry;//销毁Registry对象，释放资源。

}

　　编译运行上面的代码就可以将IE的标题改为“台湾是中国的一部分，世界上只有一个中国！”了。笔者写了个小程序，可以测出当前的IE标题和默认主页是什么，并可随意修改他们，还可以禁止别人修改你的默认主页和注册表编辑器.。

　　技巧⒉调用API编程

　　其实这是最简单的，API是系统在DLL里为我们提供的程序接口，可以直接调用的。只要我们有一本《Windows API大全》之类的书就足够了，下面举个简单的例子：

程序三：调用API函数隐藏Windows的任务栏：

HWND WndHandle;//定义句柄类型变量

WndHandle=FindWindow("Shell\_TrayWnd",NULL);//调用API函数FindWindow（）获得任务栏的句柄

ShowWindow(WndHandle,SW\_HIDE);//再调用API函数ShowWindow（）隐藏任务栏

大家看到，在上面调用API函数FindWindow（）和ShowWindow（）的过程中，只要我们知道函数的

名字和括号里的参数是什么就行了，至于实现的过程不必理会，也轮不到我们这些菜鸟去理会：）学会

调用API，你可以写出功能强大的程序来，这一技巧对于初学者来说是必须掌握的（代码请参考黑防光

盘）。

技巧⒊多线程编程技术

通过上一篇的介绍，大家都很清楚线程的概念了，它是进程内部的一个执行单元（如一个函数等），上期说了那么多理论，现在该派上用场了。编写多线程应用程序是指使程序在运行时创建多个线程并发地运行于同一个进程中。今年6月份横空出世的“中国黑客”病毒不是采用了全球独创的“三线程技术”吗？虽然笔者没机会分析它的样本代码，但此种病毒的工作效率如此之高是与它的多线程技术分不开的。

使用多线程技术编程有如下优点：

　　①提高CPU的利用率。由于多线程并发运行，可以使用户在做一件事情的时候还可以做另外一件事。特别是在多个CPU的情况下，更可以充分地利用硬件资源的优势：将一个大任务分成几个小任务，由不同的CPU来合作完成。

②采用多线程技术，可以设置每个线程的优先级，调整工作的进度。

清楚了使用多线程技术的优势之后，下面便来谈谈如何在C++ Builder环境下开发多线程的应用程序，在C++Builder 环境中，通过 TThread 类就可以很方便地编写多线程应用程序（但不能直接使用，因此要派生新类），

具体流程如下：

　　从TThread 类派生出一个新的线程类->创建线程对象->设置线程对象的属性项->挂起或唤醒线程（根据具体情况操作）->结束线程。要说明一点的是：在应用程序中要合理地设置线程的优先级。不要因为某些线程的优先级很高而使其他一些线程因为等不到CPU的处理时间而被“饿死”，也不要因为线程的级别都差不多而导致的频繁切换花费大量的CPU时间。（本段引自《C++ Builder 5 编程实例与技巧》P284）。

技巧⒋让程序实现后台监控

　　这是一个很基本的技巧。如果你是一个木马程序的爱好者，当你阅读众多的木马源程序的时候，就会发现100%的木马程序都很注意自身的后台监控本领，也就是隐身技术，面对不同的系统要施展不同的对策才能实现。很多杀毒程序就采用了这种后台监控技术，使程序随着系统的启动而运行，然后在后台悄悄地监视系统的一举一动，一发现有不对路的程序就把它“揪”出来示众。实现程序的后台监控技术有如下几个关键：

①正常运行时，不显示程序的窗体；

②系统每次启动都自动运行程序一次；

③程序图标不显示在任务栏上；

④不显示在按Ctrl+Alt+Del 调出的任务列表中；

⑤通过热键可以调出隐藏的窗体

　　实现方法：对于①，要不显示窗体，我们可以编辑WinMain函数，设置ShowMainform值为False就可以隐藏程序的窗体了。参考代码：Application->ShowMainform = false ；对于②，可以利用技巧1所介绍的方法修改注册表，键值如下：HKEY\_LOCAL\_MACHINESOFTWAREMicrosoftWindowsCurrentVersionRun ，使用的是WriteString（）方法。这是冰河等多种旧木马惯用的启动手段之一（当然还有文件关联、注入dll等方法）；对于③，要使程序图标不显示在任务栏上，也很简单，调用API函数SetWindowLong 可以让程序运行后不出现在任务栏里，不过要放在窗体的OnCreate（）里面。代码如下：

SetWindowLong(Application->Handle,GWL\_EXstyle,WS\_EX\_TOOLWINDOW);

对于④，调用RegisterServiceProcess API 函数将程序注册成为一个服务模式程序，让它运行在较高的优先级下,就不会出现在程序列表中（对Win9X有效，WinNT/2000/XP下无效）。具体的代码请参考笔者的《一个简单木马程序的编写与伪装策略》一文，不在此重叙对于⑤，要先定义捕获Windows消息WM\_HOTKEY的钩子函数，然后向Windows 加入一个全局原子，并保留其句柄，最后向Windows登记热键，这个可以调用API函数RegisterHotKey来实现。

技巧⒌使用定时触发器

在C++ Builder 环境下，定时触发器即Timer控件，有时候我们希望程序隔一段时间重复执行相同的动作，比如对QQ密码截获的时候，就要隔一段间隔寻找一次QQ登录窗口。在C++ Builder中，只要将执行这些动作的代码放到一个Timer中去就OK了。听说“中国黑客”病毒运行几分钟后就自动创建一个新的线程，用于寻找OICQ的“发送消息”窗口，在10分钟内一直在找，一旦找到就将“去\*\*\*\*\*功”等带有政治色彩的言论发送给受害者QQ上的好友，10分钟后自动结束该线程。我想在查找“发送消息”窗口的10分钟内就运用了定时器，该病毒是用汇编开发的。可是在C++ Builder中是如何运用的呢？其实控件的出现使得编程变得很简单，添加一个Timer控件，设置几下控件的属性，双击Timer控件，将代码放到里面去就行了。程序执行的时候，相隔指定的时间就重复执行里面的代码了。实际上笔者在上一期的“程序一”中寻找QQ登录窗口时，就运用了定时器.有关编程技巧的介绍到此为止，请读者参考另类书籍，掌握更多的黑客编程技巧，编写出受欢迎的黑客程序

来。

　 五、Socket 编程与网络通信基础

由于本文的主题是“黑客编程基础”，而黑客是互连网上“来无影，去无踪”的黑衣人，如冰河、网络神偷等黑客程序都是基于互连网的，谈黑客编程离开网络编程就会大失其味。所以，下面接着谈谈网络编程，大凡基于网络应用的程序都离不开Socket。Socket 为套接字之意，是作为计算机与计算机之间通信的接口。有关Socket的概念在第6期《黑客防线》的《Socket 编程的基础和基本过程》一文中有详细的描述，请大家参考，不在此多叙。需要指出的是：Winsock 是访问众多的基层网络协议的一种接口，在每个Win32平台上，它都以不同的形式存在着，Winsock 是网络编程的接口，不是协议，这是容易弄错的地方。现在来谈谈Winsock 编程的过程，大凡在Win32平台上的Winsock编程都要经过下列的基本步骤：定义变量->获得Winsock版本->加载Winsock库->初始化->创建套接字->设置套接字选项->关闭套接字->卸载Winsock库，释放所有资源。下面以一道极其简单的程序来说明如何进行Winsock编程。程序四：编一个程序来获取本地机器的IP地址。使用Winsock提供的API函数是最基本的网络技术，为了给初学者看个清楚，笔者打算在Visual C++ 和C++ Builder下各写一个，便于大家区分这两种不同的编程工具的特性（对于本程序来说，他们都差不多，而对于某些通信程序，他们实现起来就相差很远了，但本质是差不多的）。先来看Visual C++ 下的源程序，实现步骤：打开Visual C++ ，从“File”菜单中的“New”新建一个工程，选中“Win 32 Console Application” ,意思是说生成的是Win32的控制台程序。另外，初学者要注意一点：只要程序中用到了 Winsock API 函数，都要在工程设置的Link 中增加 Ws2\_32.lib 文件，不然程序将不能通过编译，方法是：点击“Project”菜单，选择“Settings... ALT+F7” ，在弹出的“Project Settings”对话框右侧选“Link”标签，再在“ProjectOptions”下方的编辑框中增加Ws2\_32.lib文件，点“OK”就可以了。

　　加载好文件之后，就可以在CheckIP.cpp文件里加入以下代码了：

//-------Begin from ------------

//包含需要使用的头文件

#include "stdafx.h"

#include "windows.h"

#include

#include "stdio.h"

#include "stdlib.h"

#include "string.h"

void CheckIP(void) //定义CheckIP（）函数，用于获取本机IP地址

{

WORD wVersionRequested;// WORD类型变量，用于存放Winsock版本的正确值SADATA wsaData;

char name[255];//定义用于存放获得的主机名的变量

CString ip;//定义IP地址变量

PHOSTENT hostinfo;

wVersionRequested = MAKEWORD( 2, 0 );

//调用MAKEWORD（）获得Winsock版本的正确值，用于下面的加载Winsock库

if ( WSAStartup( wVersionRequested, &wsaData ) == 0 ) {

//现在是加载Winsock库，如果WSAStartup（）函数返回值为0，说明加载成功，程序可以继续

//往下执行

if( gethostname ( name, sizeof(name)) == 0) {

//如果成功地将本地主机名存放入由name参数指定的缓冲区中

if((hostinfo = gethostbyname(name)) != NULL) {

//这是获取主机名，如果获得主机名成功的话，将返回一个指针，指向hostinfo，hostinfo

//为PHOSTENT型的变量，下面即将用到这个结构体

LPCSTR ip = inet\_ntoa (\*(struct in\_addr \*)\*hostinfo->h\_addr\_list);

//调用inet\_ntoa（）函数，将hostinfo结构变量中的h\_addr\_list转化为标准的点分表示的IP

//地址（如192.168.0.1）

printf("%sn",ip);//输出IP地址 } }

WSACleanup( );//卸载Winsock库，并释放所有资源 } }

int main(int argc, char\* argv[])//主函数，程序的入口

{

CheckIP();//调用CheckIP()函数获得、输出IP地址

return 0;//由于main（）定义为int型，所以应带回一个int型的数值

}

　　下面接着来看看在C++ Builder 下如何实现，其实两者的思想是一样的，只是在C++ Builder下实现

的界面友好点而已，实现方法：打开C++ Builder 5，默认情况下已经新建一个工程，保存这个工程文件

就可以了，构造如下面图4所示的界面，在相应之处添入下面的代码即可程序代码：

//包含头文件

#include

#include

#pragma hdrstop

#include "Unit1.h"

#pragma package(smart\_init)

#pragma resource "\*.dfm"

Tform1 \*form1;

\_\_fastcall Tform1::Tform1(TComponent\* Owner)

: Tform(Owner)

{ }

void Tform1::GetHostIpAddress()

{// GetHostIpAddress（）获得本机IP地址

struct hostent \*thisHost;

struct in\_addr in;

char MyName[80];

char \*ptr;

WORD wVersionRequested;

WSADATA wsaData;

int err;

wVersionRequested = MAKEWORD( 2, 0 );

err = WSAStartup( wVersionRequested, &wsaData );

if( err != 0 )

return;

if(LOBYTE( wsaData.wVersion ) != 2 ||

HIBYTE( wsaData.wVersion ) != 0 )

{ WSACleanup( );

return; }

if(gethostname(MyName,80)==SOCKET\_ERROR)

return;

if(!(thisHost=gethostbyname(MyName)))

return;

memset((void \*)&in,sizeof(in),0);

in.s\_addr=\*((unsigned long \*)thisHost->h\_addr\_list[0]);

if(!(ptr=inet\_ntoa(in)))

return;

WSACleanup( );

Edit1->Text=AnsiString(ptr);}

void \_\_fastcall Tform1::formCreate(TObject \*Sender)

{ GetHostIpAddress();}

void \_\_fastcall Tform1::Button1Click(TObject \*Sender)

{Close();//添加一个“确定”按钮，点击即关闭程序。}

程序在 C++ Builder 5 下编译通过，通过比较你会发现他们是大同小异的，对于同一程序，

两者工具各有秋千，至于选择哪种由你决定，最好是两者相得益彰。“临渊羡鱼，不如退而结网”，虽说“通往电脑的路不止一条”，然而对于编程，道路却只有一条，就是：动手去做，亲身实践。

兄弟，愿你成为一个出色的舵手，用代码去驽驾电脑的世界，用编程去填充七彩的人生！