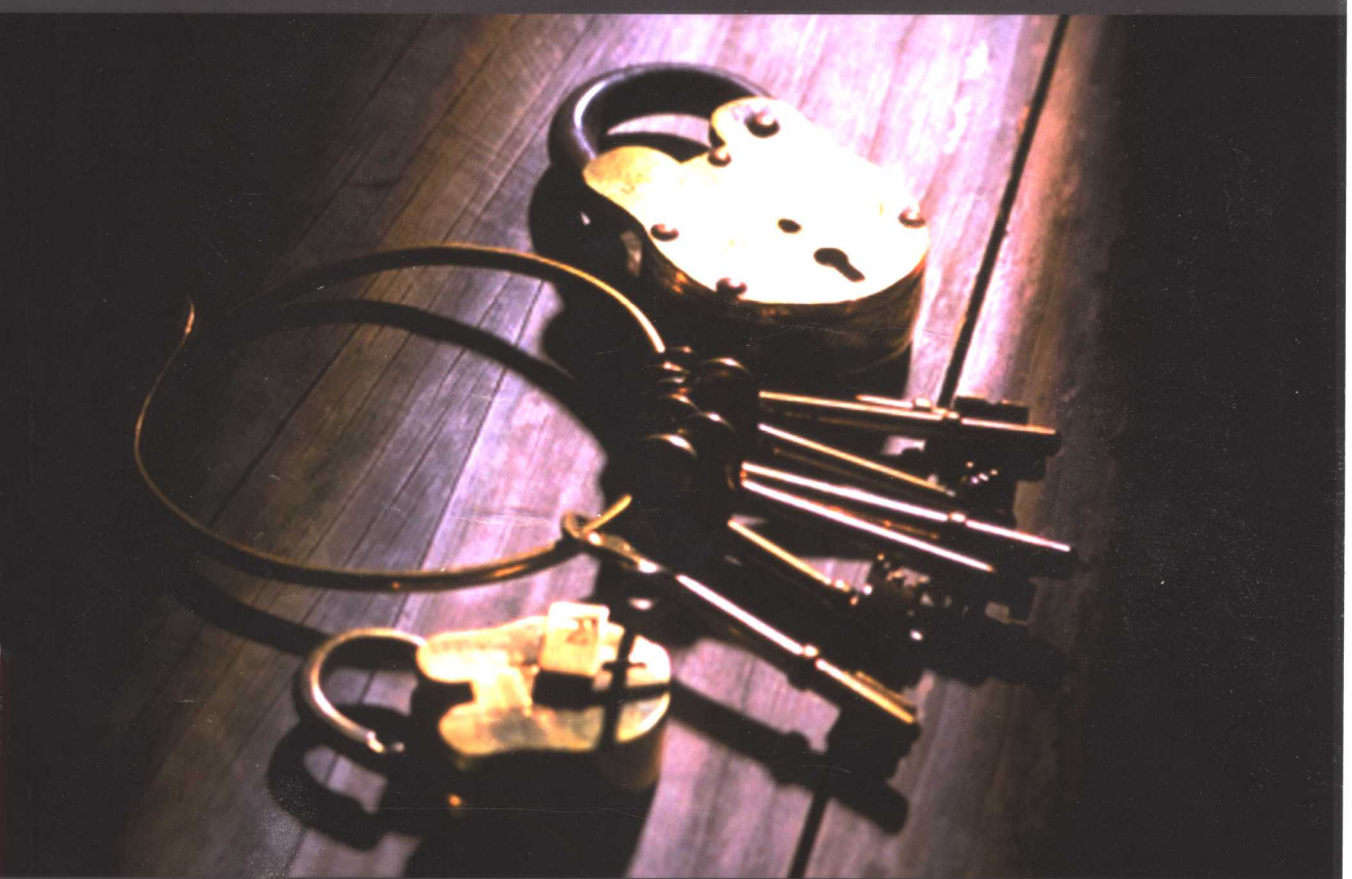
Windows 信息安全原理与实践

赵树升　赵韶平　编著



清华大学出版社

Windows 信息安全原理与实践

赵树升　赵韶平　编著

清华大学出版社

北　京

内 容 简 介

本书以 Windows 操作系统为基础，介绍了信息安全相关的基础知识（开发工具、调试工具、网络工具等）、软件保护（程序保护及软盘、硬盘、光盘数据保护）、病毒和反病毒技术（可执行文件病毒、脚本病毒、蠕虫等）、网络攻击与防护技术（如拒绝服务攻击、网络嗅探、扫描技术等）、常用的数据加密算法（DES、RSA、CryptoAPI 等）。本书理论与实践相结合，每节均配有翔实的例子，理论阐述简单易懂，实例描述选用流行的开发工具 VC++和 Masm32 汇编语言以及 VB。

本书可作为高等院校信息安全方面课程的教材，也可作为学过汇编语言和 C++语言的读者以及其他编程爱好者的信息安全方面的参考书。

图书在版编目（CIP）数据

Windows 信息安全原理与实践/赵树升，赵韶平编著.－北京：清华大学出版社，2004.9

ISBN 7-302-09559-0

Ⅰ.W…　Ⅱ.①赵…②赵…　Ⅲ.窗口软件，Windows－安全技术　Ⅳ.TP316

中国版本图书馆 CIP 数据核字（2004）第 095689 号

出 版 者：清华大学出版社　　**地　　址**：北京清华大学学研大厦
http://www.tup.com.cn　**邮　　编**：100084
社 总 机：010-62770175　**客户服务**：010-62776969

组稿编辑：刘利民
文稿编辑：刘　丽
封面设计：秦　铭
版式设计：郑轶文
印 装 者：三河市春园印刷有限公司
发 行 者：新华书店总店北京发行所
开　　本：185×260　**印张**：23.25　**字数**：532 千字
版　　次：2004 年 9 月第 1 版　2004 年 9 月第1 次印刷
书　　号：ISBN 7-302-09559-0/TP・6649
印　　数：1～5000
定　　价：38.00 元(附光盘 1 张)

序　言

随着信息技术的迅猛发展和我国社会信息化进程的不断加速，计算机网络和信息系统已经在政府、企业、社会团体以及家庭等多方面得到了普及，在人们生活中发挥着越来越重要的作用。由于信息系统的复杂性，一些系统中存在的漏洞被不断发现、一些系统提供的正常功能不断地被别有用心的人利用来破坏系统的正常使用甚至侵入系统窃取用户的重要数据。因此，信息系统存在的安全问题不仅关系着国家的安全与利益，也与人们的生活息息相关。在这种情况下，信息领域渴求一大批从事信息安全工作的专业人才。由于我国在这方面相对于发达国家起步较晚，信息安全方面人才的培养也起步较晚，造成信息安全方面的从业人员在数量上还远远满足不了社会实际需要；另外，信息安全问题也是每一个计算机用户难以回避的问题，因此，很多高校也开设了信息安全课程。怎样快速、高质量地培养信息安全方面人才，普及信息安全知识，是很多高校探索的课题。

由于信息安全理论的学习涉及很多方面专业知识，例如操作系统、多种开发工具、调试工具、网络协议等，令很多初学者有些望而生畏。他们一方面谴责入侵计算机系统的黑客和病毒制造者，另一方面又十分崇拜他们的编程技术，他们想增强自己的防范能力但又觉得安全方面的知识过于高深莫测。

本书的作者在给两届本科第七学期学生讲授过信息安全课程后，尝试把教案汇编成书。本书的目的是，通过选用流行的 Windows 操作系统，力图将深奥的理论通俗化，把通俗后的理论实例化。经过两届的尝试，既较好地激发了学生的学习兴趣，也激发了他们自己动手分析解决问题的兴趣。

本书的特点是：

1．基于具体的操作系统 Windows。虽然还有其他操作系统，在大学阶段接触和应用较多的还是 Windows，这样，拉近了理论和读者之间的距离。

2．理论与实例相结合。只有理论没有实例，教师教得枯燥，学生学得费力，兴趣也不大，不利于知识的掌握。本书的每节在阐述完理论后，都有对应实例讲解，还附有完整程序。

3．选用典型工具。书中使用的开发工具以汇编语言和 C++为主，还有部分的 VB 代码。大学四年级上学期，正是学生学习完操作系统、网络课程、C++和汇编语言时间不久，书中的知识对巩固他们所学的理论知识、增强他们的就业竞争能力有所帮助。

4．强调培养实践能力。作者在实例的讲解上下了很大功夫，条理清楚，分析细致，通俗易懂。

希望本书能够为引导初学者由易入难、尤其对培养面向应用型人才方面有一些益处。

何文趋

2004 年 6 月

目　录

第1章 基础知识

本章介绍了阅读和理解后续章节所需要的基础知识。本书用到的程序语言以汇编语言、C++语言和脚本语言 VB Script 为主；用到了大量的 Windows 系统提供的 API 函数和组件；使用的开发工具以 VC++、Masm32 为主；使用的调试分析工具以 W32Dasm 为主。因为不少地方需要网络编程，本章还介绍了网络命令和网络套接字编程。为了理解病毒和反病毒知识所需要的可执行文件，PE 结构知识也放在本章。

关于信息的定义有很多种，我国信息论专家钟义信把信息定义为：事物运动的状态和方式。一般来讲，信息具有普遍性和可识别性，具有存储性和可处理性，具有时效性和可共享性，具有增值性和可开发性，具有可控性和多效用性。

信息安全（information security）的定义是：一个国家的社会信息化状态不受外来的威胁和侵害，一个国家的信息技术体系不受外来的威胁与侵害。从技术上讲，保证信息的所有者能够安全可靠地拥有和使用信息资源。信息安全的基本属性有：

（1）保密性（confidentiality）：确保信息在存储、使用、传输过程中不会泄漏给非授权用户或实体。

（2）完整性（integrity）：确保信息在存储、使用、传输过程中不会被非授权用户篡改，同时还要防止授权用户对系统及信息进行不恰当的篡改，保持信息内、外部表示的一致性。

（3）可用性（availability）：确保授权用户或实体对信息及资源的正常使用不会被异常拒绝，允许其可靠而及时地访问信息及资源。

（4）可靠性（reliability）：以用户认可的质量连续服务于用户的特性。

信息安全的核心是密码技术。信息安全的其他重要措施有访问控制、防火墙技术与入侵检测等。

1.1 Windows 信息安全概述

Windows 是目前应用最广泛的操作系统，它和 DOS 的区别不只是操作的方便与功能的强大，更是注重了安全性。Windows 98/NT/2000 及以后操作系统使用 32 位内核，同时也是多任务和多线程的操作系统。系统由一组软件模块构成，它们被称为执行程序服务（executive program service），运行在内核模式（kernel mode）。在内核模式之外是用户模式（user mode）。用户模式由非特权的服务组成。内核服务包括对进程、线程、资源管理和文件与内存管理的系统调用。用户服务包括窗口、控件和消息的处理等。Windows 提供多种多样的应用程序编程接口（API），例如 TAPI（电话服务编程）、ODBC API（数据

库编程）来保证程序员方便安全地使用系统功能。使用虚拟内存机制来保证各进程的安全。这些知识在后面章节都会进行介绍。

1. Windows NT 安全特点

（1）Windows NT 以后使用 FAT 和 NTFS 文件系统，同时支持 CIFS（common internet file system）文件系统。NTFS 文件系统对文件进行访问控制，CIFS 为网络间文件和对象的共享提供了支持。

（2）使用域控制器（domain controller）管理网络中共享对象的安全访问。

（3）使用用户账号（user account）保证用户登录的合法性。

（4）使用用户权利（right）和权限（permission）来保证登录的用户合法使用系统资源，例如，在 NTFS 中对文件对象的权限有写入、读取、删除、运行等，并且进行的操作被写入日志文件。

（5）使用 3 种类型的用户组（全局组、本地组和特别组）来管理具有相同权利权限的用户。

（6）使用密码来保证合法用户登录的安全性，在 Windows NT 系统中，使用了 LanManager 认证和密码机制。

（7）Windows 使用注册表（registry）的数据库保存系统的配置信息，如用户账户、设备驱动程序信息等。

2. Windows 2000 安全特性

（1）活动目录

活动目录（active directory）是关于用户、硬件、应用和网络数据的存储中心，也存储用户的认证信息，及用户使用某一资源的授权信息等，这样可以简化管理，具有良好的可伸缩性。为了创建这种分层结构，活动目录使用域（domain）、对象和组织单元（organizational units）来管理和使用网络资源。

一个域是网络对象（包括组织单元、用户账号、组和计算机等）的集合，它们共享一个公共目录数据库，并组成活动目录中逻辑结构的核心单元。每个域中可能包含多个组织单元和用户（对象），这样更符合公司或企业的组织模式。

（2）Kerberos 认证

Kerberos 是基于共享密钥的认证协议，定义了客户端和称为密钥分配中心 KDC（key distribution center）的认证服务之间的安全交互过程。Windows 2000 中采用多种措施提供对 Kerberos 协议的支持以保证数据的完整性、可信度并允许对客户和服务器的认证：Kerberos 客户端使用基于 SSPI（microsoft security support provider interface）的 Windows 2000 安全提供者，初始 Kerberos 认证同 WinLogon 的单次登录进行了集成，而 Kerberos KDC 也同运行在域控制器中的安全服务进行了集成，并使用活动目录作为用户和组的账号数据库。

（3）PKI 体系结构

系统对 PKI（public key infrastructure）做了全面支持。PKI 在提供高强度安全性的同时，还与操作系统进行了紧密集成，并作为操作系统的一项基本服务而存在，避免了购买第三

方 PKI 所带来的额外开销。组成 Windows 2000 PKI 的基本逻辑组件中最核心的为微软证书服务系统（Microsoft Certificate Services），它允许用户配置一个或多个企业 CA（certification authority），这些 CA 支持证书的发放和废除，并与活动目录和策略配合，共同完成证书和废除信息的发布。

（4）智能卡技术

现在越来越多的企业正在寻找各种方法来提高其网络资源的安全性，智能卡（smart card）或称为灵巧卡，就是其中比较流行的一个。智能卡提供了让非授权人更难获取网络存取权限的一种简单方式，Windows 2000 对智能卡安全提供了内在支持。

（5）加密文件服务 EFS

EFS（encrypting file system）允许任何用户加密 NTFS 分区上的文件或者目录，并使之具有适当的用户权限。

3. Windows 2003 安全特性

在该系统下，曾经在 Windows 2000 Server 默认情况下能够运行的 20 多种服务被关闭或者使其以更低的权限运行。而且 IIS（internet information server）和 Telnet（远程访问服务）在默认的情况下都没有安装，并且这两个服务是在两个新的账户下运行，新账户的权限比正常系统账户的权限要低。如果恶意的黑客危及到这两个服务时，这种改变将直接改善服务器的安全性。除此之外，还有以下改进：

（1）系统的软件防火墙为网络服务器提供了基本的端口安全性。

（2）使用软件限制策略和强制执行机制，来限制系统上运行的未授权的可执行程序。

（3）当使用 IIS 6.0 的默认安装时，网页服务器的安全性将达到最大化。新的 IIS 6.0 安全特性包括可选择的加密服务，高级的摘要认证以及可配置的过程访问控制。

（4）新的摘要安全包支持在 RFC2617 中定义的摘要认证协议。该包对 IIS 和活动目录（AD）提供了更高级的保护。

（5）基于 IEEE 802.1X 规范改进了以太局域网和无线局域网的安全性，促进了用户和计算机的安全认证和授权。这些改进也支持公钥证书和智能卡的自动注册。

（6）对于所有的用户凭证，包括口令密码和 X.509 证书，凭证管理器提供了一个安全的仓库。这个特性使得单一的签名特性可以获得多个领域的信任。

（7）Internet 认证服务器和远程认证拨号用户服务器（IAS/RADIUS）控制远程的用户认证和授权访问。对于不同的连接类型，例如拨号上网，虚拟专用网（VPNs）以及防火墙连接，该服务很实用。

（8）联邦信息处理标准（FIPS）算法支持 SHA-1、DES、3DES 和一个随机数发生器。这种政府级的加密模式用于加密通过 VPN 使用第二层隧道协议（L2TP）和 IP 安全（IPSec）建立的连接，这种连接或是从客户端到服务器，或是从服务器到服务器，或是从网关到网关。

（9）安全套接字层（SSL）客户端认证的改进使得会话速度可以提高 35%，而且多个进程可以缓存和共享会话。这样可以减少用户对应用程序的认证，从而减少应用程序服务

器上的网络通信量和 CPU 工作周期。

（10）加密文件服务（EFS）的改进允许管理员和用户提供给多个用户访问多组加密文件的可能。它还提供了额外的文件存储保护和最大数量的用户容量。

1.2　Masm32 的使用

1.2.1　Masm32 的特点

Masm32 的原代码编辑、编译的程序是目录下的 Qeditor.exe，执行界面如图 1.1 所示。在 Masm32 目录下，有很多实例，涉及到很多方面的编程，参考这些例子对编程帮助很大。

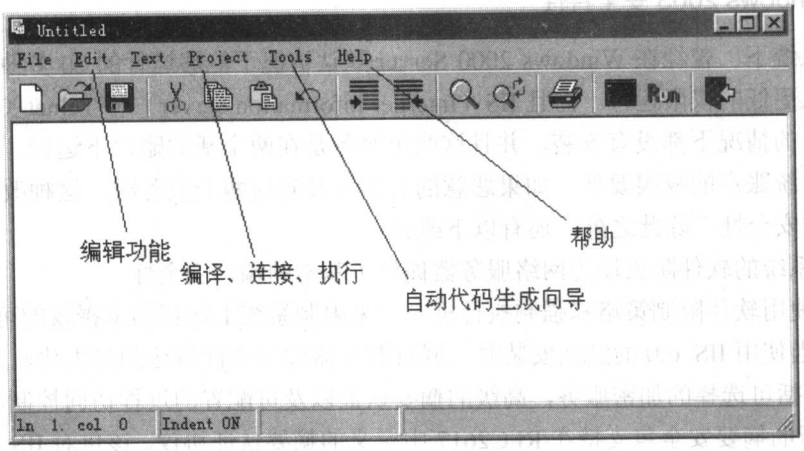


图 1.1　Masm32 开发界面

很多人使用过 DOS 下的汇编语言开发工具 Masm6.1x，但不能用它开发 Windows 下的 32 位程序。要用汇编语言编写使用 API 函数和带有图形界面窗口的程序，需要使用 Masm32。

很多网站都提供该软件的下载，下载后解压缩不需安装就可以直接使用。前者开发出的 DOS 程序和后者开发出的 Windows 程序的主要区别是，Windows 程序运行在保护模式下，Windows 把每一个 Win32 应用程序放到分开的虚拟地址空间中去运行，也就是说，每一个应用程序都拥有其相互独立的 4GB 地址空间，当然这倒不是说它们都拥有 4GB 的物理地址空间，而只是说能够在 4GB 的范围内寻址。操作系统将会在应用程序运行时完成 4GB 的虚拟地址和物理内存地址间的转换。在保护模式下，程序访问存储器所使用的逻辑地址称为虚拟地址（virual address，VA）。与实地址模式下的分段地址类似，虚拟地址也可写成“段：偏移量”的形式，这里的段是指段选择器。这就要求编写应用程序时必须遵守 Windows 的规范，否则极易引起内存的保护模式错误。而在过去的内存实模式下，所有的应用程序都运行于同一个地址空间，它们可以彼此读写别的程序的内容，这极易导致一个应用程序破坏另一个应用程序，甚至是操作系统的数据或代码。

1.2.2　Masm32 程序的结构

下面的程序段只是一个框架，和 DOS 下的汇编程序有区别，又有很多相似的地方：

```
.386
.MODEL Flat, STDCALL
.DATA
;初始化数据定义于此
...
.DATA?
;未初始化数据定义于此
...
.CONST
;常量定义于此
...
.CODE
;标号
;代码部分
...
end <label>
```

框架就这么简单，我们来解释如下：

.386：这是一个汇编语言伪指令，它告诉编译器程序是使用 80386 指令集编写的。还可以使用.486 和.586，但最安全的还是使用.386。对于每一种 CPU 有两套几乎功能相同伪指令：.386/.386P、486/.486P、586/.586P。带 P 的指令表明程序中可以用特权级指令。特权级指令是保留给操作系统的，如虚拟设备驱动程序。在多数时间，程序都无须运行在 RING0 层，故用不带后缀 P 的伪指令已足够了。

.MODEL FLAT，STDCALL：.MODEL 是用来指定内存模式的伪指令。在 Win32 下，只有一种内存模型，那就是 FLAT。STDCALL 告诉编译器参数的传递约定。参数的传递约定是指参数传递时的顺序（从左到右或从右到左）和由谁恢复堆栈指针（调用者或被调用者）。在 Win16 下有两种约定：C 和 Pascal。C 约定规定参数传递顺序是从右到左，即最右边的参数最先压栈，由调用者恢复堆栈指针。例如，为调用函数：

```
foo(int first_param,int second_param,int third_param );
```

按 C 约定的汇编代码应该是这样的：

```
push [third_param]
push [second_param]
push [first_param]
```

```
call foo
add esp, 3 * 4 ;调用者自己恢复堆栈指针
```

Pascal 约定和 C 约定正好相反，它规定参数是从左向右传递，由被调用者恢复堆栈。Win16 采用了 Pascal 约定，因为 Pascal 约定产生的代码量要小。当不知道参数的个数时，C 约定特别有用。如在函数 wsprintf () 中，wsprintf 预先并不知道要传递几个参数，所以它不知道如何恢复堆栈。STDCALL 是 C 约定和 Pascal 约定的混合体，它规定参数的传递是从右到左，恢复堆栈的工作交由被调用者。Win32 只用 STDCALL 约定，但除了一个特例，即 wsprintf。

.DATA、.DATA?、.CONST、.CODE：上面的 4 个伪指令是“分段”(section)伪指令。刚讲过 Win32 下没有“段”(segment)的概念，但是可以把程序分成不同的“分段”，即一个“分段”的开始是上一个“分段”的结束。Win32 中只有两种性质的“分段”：DATA 和 CODE，即数据段和代码段。

其中 DATA“分段”又分为 3 种：

.DATA 包括已初始化的数据。

.DATA?包括未初始化的数据。比如有时仅想预先分配一些内存但并不想指定初始值。使用未初始化的数据的优点是它不占据可执行文件的大小，例如，若要在 .DATA? 段中分配 10000B 的空间，可执行文件的大小无须增加 10000B，而仅仅是要告诉编译器在装载可执行文件时分配所需字节即可。

.CONST 包括常量定义。这些常量在程序运行过程中不能更改的。应用程序并不需要以上所有的“分段”，可以根据需要进行定义。

.CODE 这是代码“分段”。实际上，分段并不像在 DOS 下一样，为不同的段分别指出不同的段寄存器，因为 Windows 下只有一个 4GB 的段，Windows 程序中的分段表现在当程序装载时，赋予不同的分段以不同的属性，比如说当程序加载时，对于 Ring3 程序来说，.code 段是不可写的，而 .data 段是可写的，如果尝试像在 DOS 下一样写自己的代码部分，程序会出错。

<label> … end<label>：是用来惟一标识代码范围的标签，两个标签必须相同，应用程序的所有可执行代码必须在两个标签之间。

1.2.3 Masm32 使用举例

下面来看一个小的 Masm32 程序：

```
.386
.model flat, stdcall
option casemap :none   ; case sensitive
include \masm32\include\windows.inc
```

```
include \masm32\include\kernel32.inc
include \masm32\include\user32.inc
includelib \masm32\lib\kernel32.lib
includelib \masm32\lib\user32.lib
.data
  text1 db "汇编语言并不难",0
  text2 db "你好,欢迎你使用",0
.code
start:
  invoke MessageBox,0,offset text1,offset text2,MB_OK
  invoke ExitProcess,eax
end start
```

代码生成可执行文件后如图 1.2 所示。

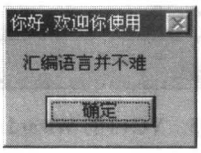


图 1.2　程序执行

下面解释这段代码：

option casemap:none 指标号大小写敏感。例如，前面的 start 和后面的 start 必须大小写一致。

include 指要包含的文件，和 C 语言的 include 很相似。一般程序都要包含文件 Windows.inc，它包含了很多常量的定义，如后面用到的 MB_OK。MessageBox 和 ExitProcess 是两个 API 函数，前者的声明在文件 user32.inc，后者的声明在文件 kernel32.inc。Masm32 中除了 API 函数的使用必须大小写与原函数保持一致，对指令无大小写要求。

includelib 指要求连接器连接的库文件名，与 include 对应。

invoke 是函数调用伪指令。例如 invoke ExitProcess,eax 相当于：

```
push eax
call ExitProcess
```

1.3　VC++的套接字类

VC++的 Wizard 向导功能可以帮助自动生成图形界面程序和控制台程序，如图 1.3 所示。单击“文件”→“新建”，出现 Wizard 界面，选择“工程”，再选中希望建立的程序方式，是图形方式还是控制台方式或别的，输入工程名，单击“确定”按钮，再根据提示

往下选择就可以得到工程的框架程序。

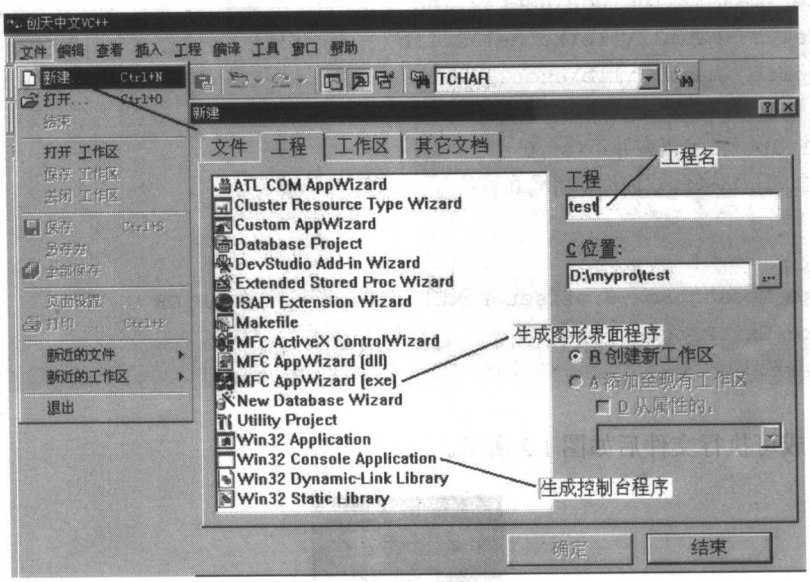


图 1.3　VC++的 Wizard 向导

1.3.1　VC++和网络套接字

VC++可以直接使用包含在头文件 winsock2.h 中的套接字函数。主要的函数如表 1-1 所示。

表 1-1　套接字 API 函数

序　号	名　字	作　用
1	socket	建立套接字
2	bind	将套接字和本地地址绑定
3	connect	和一个套接字建立连接
4	accept	接受一个套接字的连接请求
5	listen	服务端套接字侦听连接请求
6	send	向一个套接字发送数据
7	recv	从一个套接字接收数据
8	setsockopt	设置套接字选项

在第 4 章有这些函数的具体使用例子。但它们的使用并不方便，因此 VC++将它们封装成两个类：CAsyncSocket 和 CSocket。前者以非阻塞方式通信，后者以阻塞方式通信，前者是后者的基类。阻塞和非阻塞的区别是：例如非阻塞方式发送完数据就返回了，而阻塞方式发送完数据且对方正确接收到数据或者因错误发送超时才函数返回。