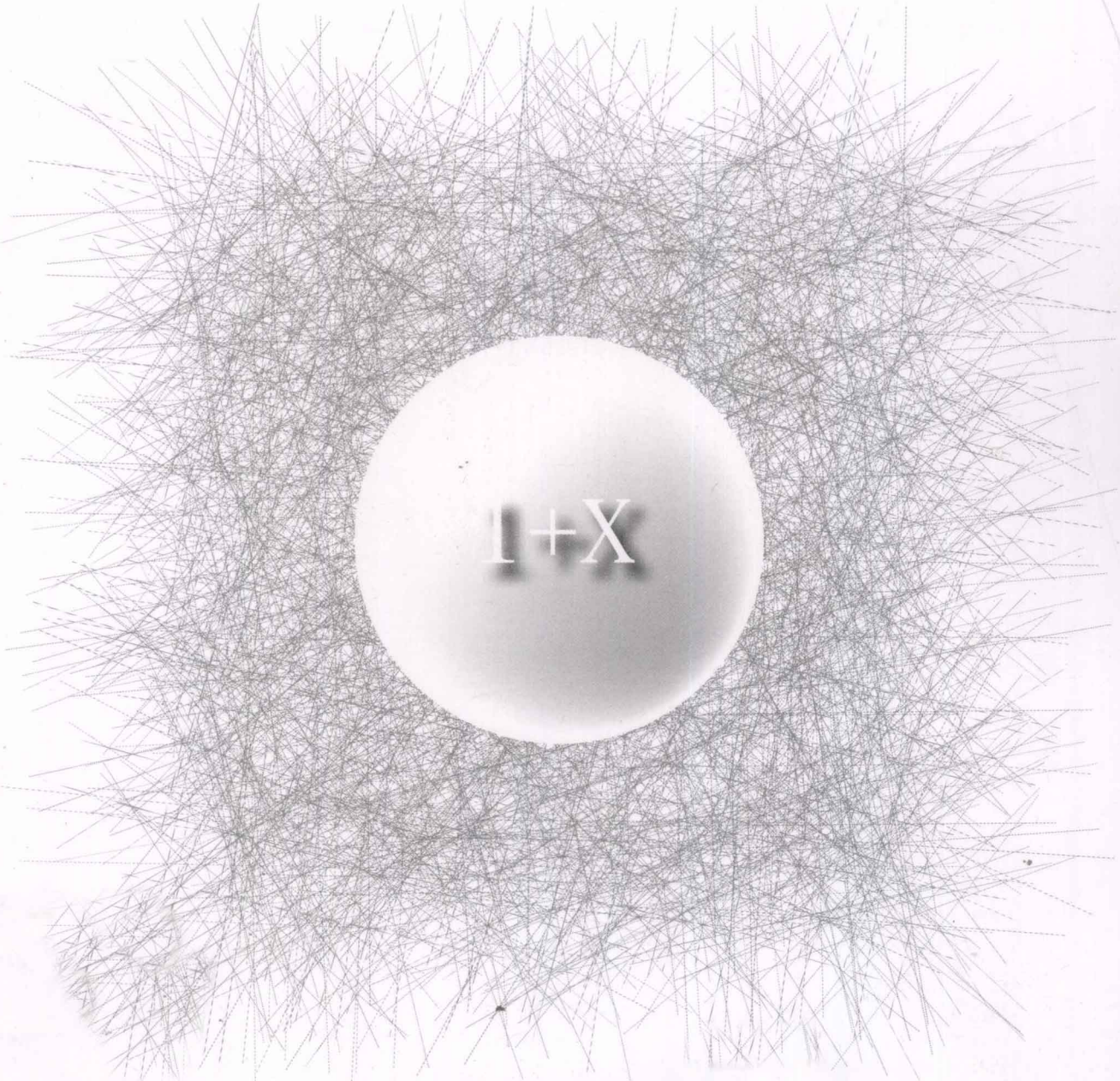
大学计算机基础教育规划教材

C程序设计实验教程

姜学锋 魏 英 编著

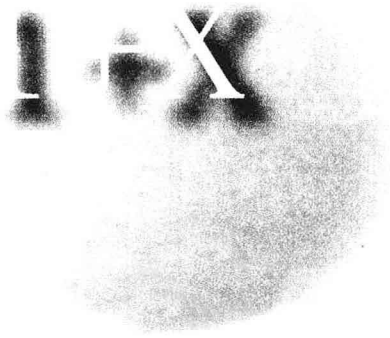


清华大学出版社

大学计算机基础教育规划教材

C程序设计实验教程

姜学锋 魏 英 编著



清华大学出版社
北 京

内 容 简 介

本书是姜学锋主编的《C 程序设计》教材的配套实验教程。全书分为 4 部分，其中详细介绍了开发工具的使用方法和程序调试技术。实验内容按课程教材和教学大纲要求设计，分验证型实验和设计型实验，突出综合性实验，并结合算法、数据结构知识设计了一些有难度的实验题目。本书还包括课程设计专题实验内容，其目的是使读者能够训练应用程序开发，获取设计 C 程序项目的初步知识和工程经验，掌握高级编程技术，为后续专业学习和职业发展打下坚实的实践基础。

本书的作者长期从事计算机基础教学和软件开发科研工作，具有丰富的教学经验和软件开发经验。全书贯彻“精讲多练、提升技能、开拓设计”的教学理念和实践，精心策划、准确定位、结构清晰、语言通俗易懂，内容由浅入深、实验循序渐进。验证型实验体现“学”，设计型实验体现“用”，课程设计体现“提升和开拓”，核心目标是技能和计算思维能力训练。

本书适合作为高等学校各专业程序设计课程的实验教材，可以独立设课，也可作为自学者的学习参考用书。

图书在版编目（CIP）数据

C 程序设计实验教程/姜学锋，魏英编著. —北京：清华大学出版社，2011.3
（大学计算机基础教育规划教材）
ISBN 978-7-302-24943-6

Ⅰ. ①C… Ⅱ. ①姜… ②魏… Ⅲ. ①C 语言－程序设计－高等学校－教材 Ⅳ. ①TP312

中国版本图书馆 CIP 数据核字(2011)第 023895 号

责任编辑：张　民　薛　阳
责任校对：梁　毅
责任印制：杨　艳

出版发行：清华大学出版社　　**地　　址**：北京清华大学学研大厦 A 座
http://www.tup.com.cn　　**邮　　编**：100084
社　　总　　机：010-62770175　　**邮　　购**：010-62786544
投稿与读者服务：010-62795954，jsjjc@tup.tsinghua.edu.cn
质　量　反　馈：010-62772015，zhiliang@tup.tsinghua.edu.cn
印 装 者：北京鑫海金澳胶印有限公司
经　　销：全国新华书店
开　　本：185×260　　**印　　张**：13.25　　**字　　数**：312 千字
版　　次：2011 年 3 月第 1 版　　**印　　次**：2011 年 3 月第 1 次印刷
印　　数：1～4000
定　　价：21.00 元

产品编号：041692-01

C程序设计实验教程 前言

本书是姜学锋主编的《C程序设计》一书的配套实验教程。C程序设计是一门实践性非常强的课程，本书根据编者多年从事C程序设计教学的经验，按照循序渐进的原则，安排了“开发环境及上机操作”、“程序调试技术”、“基础实验内容”、“课程设计”4个部分，力求使读者能够逐步掌握程序设计的方法以及C语言的精髓。

各部分的主要内容如下：

第1部分：“开发环境及上机操作”。这一部分详细介绍了Visual C++ 6.0和Code ::Blocks两种国内外高校普遍使用的C语言开发环境，包括安装过程、环境配置、联机帮助、下载地址等。同时，为力求让读者能够充分掌握程序开发的具体步骤，提高独立编写程序的能力，这一部分还通过实例为读者系统介绍了使用这两种工具编写程序的过程。

第2部分：“程序调试技术”。程序调试，是将编制的程序投入实际运行前，用手工或编译程序等方法进行测试，修正语法错误和逻辑错误的过程。如果对程序调试的知识认识不够，开发一个大型程序往往会花费大量的时间和精力，使人精疲力竭。这一部分为读者详细介绍了程序调试和错误处理的方法，以提高编程效率，达到事半功倍的效果。

第3部分：“基础实验内容”。在这一部分中，针对课程的重点和难点设计实验内容，分为验证型实验和设计型实验。两种实验内容将“教”与“学”的理念贯穿于整本书中，不仅提供给读者一些练习题，还使读者在练习的过程中培养出好的编程习惯。

第4部分：“课程设计”。大部分学生仅仅通过课堂学习后，编程能力还难以达到开发实际应用程序的要求，特别是对程序设计的热点问题如网络编程、数据库编程、多媒体编程等知识的理解和应用都还不够。“课程设计”部分将开发环境和C语言的知识点应用相结合，选取经典实例，让读者能够将程序设计和自身的专业结合起来，为将来的课程设计做好准备。

本书第1～2章由姜学锋编写，第3～4章由魏英编写。全书由姜学锋主编。西北工业大学计算机基础教学部的各位同事对全书内容提出了许多宝贵的意见，特别是尹令平和汪芳老师对本书的讲义版编写提供了很多帮助，为本书的编写打下了良好的基础，使本书更加完善；同时，本书的编写得到了各级领导的关心和支持，清华大学出版社对本书的出版十分重视并做了精心安排，使本书得以在短时间内出版。本书的一些例题还参考了

大量网络上的相关资料。在此，对所有鼓励、支持和帮助本书编写工作的领导、专家、同事、网友和广大读者表示真挚的谢意！

由于时间紧迫以及作者水平有限，书中难免有错误、疏漏之处，恳请读者批评指正。

编者

2011年1月于西北工业大学

序

大学计算机基础教育规划教材

进入21世纪，社会信息化不断向纵深发展，各行各业的信息化进程不断加速。我国的高等教育也进入了一个新的历史发展时期，尤其是高校的计算机基础教育，正在步入更加科学，更加合理，更加符合21世纪高校人才培养目标的新阶段。

为了进一步推动高校计算机基础教育的发展，教育部高等学校计算机科学与技术教学指导委员会近期发布了《关于进一步加强高等学校计算机基础教学的意见暨计算机基础课程教学基本要求》(以下简称《教学基本要求》)。《教学基本要求》针对计算机基础教学的现状与发展，提出了计算机基础教学改革的指导思想；按照分类、分层次组织教学的思路，《教学基本要求》提出了计算机基础课程教学内容的知识结构与课程设置。《教学基本要求》认为，计算机基础教学的典型核心课程包括大学计算机基础、计算机程序设计基础、计算机硬件技术基础(微机原理与接口、单片机原理与应用)、数据库技术及应用、多媒体技术及应用、计算机网络技术及应用。《教学基本要求》中介绍了上述六门核心课程的主要内容，这为今后的课程建设及教材编写提供了重要的依据。在下一步计算机课程规划工作中，建议各校采用“1＋X”的方案，即“大学计算机基础”＋ 若干必修或选修课程。

教材是实现教学要求的重要保证。为了更好地促进高校计算机基础教育的改革，我们组织了国内部分高校教师进行了深入的讨论和研究，根据《教学基本要求》中的相关课程教学基本要求组织编写了这套“大学计算机基础教育规划教材”。

本套教材的特点如下：

(1) 体系完整，内容先进，符合大学非计算机专业学生的特点，注重应用，强调实践。

(2) 教材的作者来自全国各个高校，都是教育部高等学校计算机基础课程教学指导委员会推荐的专家、教授和教学骨干。

(3) 注重立体化教材的建设，除主教材外，还配有多媒体电子教案、习题与实验指导，以及教学网站和教学资源库等。

(4) 注重案例教材和实验教材的建设，适应教师指导下的学生自主学习的教学模式。

(5) 及时更新版本，力图反映计算机技术的新发展。

本套教材将随着高校计算机基础教育的发展不断调整，希望各位专家、教师和读者不吝提出宝贵的意见和建议，我们将根据大家的意见不断改进本套教材的组织、编写工作，为我国的计算机基础教育的教材建设和人才培养做出更大的贡献。

“大学计算机基础教育规划教材”丛书主编

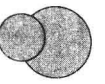
教育部高等学校计算机基础课程教学指导委员会副主任委员

冯博琴

C 程序设计实验教程

目 录

第1章



开发环境及上机操作

1.1 C语言开发环境简介

1.1.1 编译器和连接器

计算机是按照计算机指令自动工作的，计算机的工作过程就是指令的执行过程。让计算机执行什么样的工作，得到什么样的结果的过程本质上就是编写什么样指令的过程。在计算机发展早期，编写计算机指令是一件非常复杂的事情，后来人们逐步设计出各种高级语言，大大简化了指令（程序，指令的集合）设计的难度，并且提高了程序生产效率。

编译器是将一种计算机语言翻译为另一种计算机语言的程序。编译器将源语言（Source Language）编写的程序（简称源程序）作为输入，翻译产生用目标语言（Target Language）编写的等价程序。源程序一般为高级语言（High-level Language），例如C、C++等。而目标语言则是汇编语言或目标机器的目标代码（Object code，有时也称做机器代码 Machine Code）。

编译器可以生成用于编译器本身所在的计算机和操作系统（或平台）相同的环境下运行的目标代码，这种编译器叫“本地编译器”；编译器也可以生成用来在其他平台上运行的目标代码，这种编译器叫做交叉编译器，交叉编译器在生成新的硬件平台时非常有用。编译器有两种方式可以执行高级语言程序：一是通过解释程序；二是通过编译、连接生成执行代码。第一种方式，解释程序能够直接执行高级语言源程序。这种方式非常方便，但是效率不高，而且没有安装解释程序的计算机不能执行，例如Java语言等就是采用解释方式。第二种方式，使用编译器，将高级语言源程序编译、连接成执行代码，也就是二进制的机器指令，从而允许用户直接执行程序，C语言、C++语言等就是这样的方式。

尽管经过编译过程后，高级语言源程序转换成二进制的执行代码了；但在大多数的操作系统上，执行这些执行代码是按“进程”方式管理的，因此，这些二进制的执行代码还需要增加与进程和操作系统相关的执行代码，这个过程就称为“连接”。完成这种连接工作的程序称为“连接器”。

下面是高级语言源程序编译、连接为执行代码的过程示意图。

源程序→[编译器]→目标代码→[连接器]→执行代码

而C语言、C++语言源程序编译过程中还包括预处理（Pre-Processing）、二次编译

(Compiling)两个过程,其目的是为了编译优化。

1.1.2 C语言编译器

20世纪50年代,IBM的John Backus带领一个研究小组对FORTRAN语言及其编译器进行开发。但由于当时人们对编译理论了解不多,开发工作变得既复杂又艰苦。与此同时,Noam Chomsky开始了他对自然语言结构的研究。他的发现最终使得编译器的结构异常简单,甚至还带有了一些自动化。Chomsky架构中的上下文无关文法被证明是程序设计语言中最有用的,它与有限状态自动机(Finite Automaton)和正则表达式(Regular Expression)的研究引出了表示程序设计语言的单词的符号方式。当分析问题变得好懂起来时,人们就在开发程序上花费了很大的工夫来研究这一部分的编译器自动构造。这些程序最初被称为编译器的编译器(Compiler-compiler),但更应确切地称为分析程序生成器(Parser Generator),这是因为它们仅仅能够自动处理编译的一部分。这些程序中最著名的是Yacc(Yet Another Compiler-compiler),它是由Steve Johnson在1975年为UNIX系统编写的。类似地,有限状态自动机的研究也发展了一种称为扫描程序生成器(Scanner Generator)的工具,Lex(与Yacc同时,由Mike Lesk为UNIX系统开发)是其中的佼佼者。

在1973年,美国贝尔实验室的D. M. Ritchie在B语言的基础上最终设计出了一种新的语言,这就是C语言。1978年Brian W. Kernighian和Dennis M. Ritchie出版了名著《The C Programming Language》,从而使C语言成为目前世界上流行最广泛的高级程序设计语言。此时人们对操作系统和编译原理的研究均有了较大的进步,这就为后来的C语言及其编译工具的开发奠定了理论基础。

在20世纪80—90年代PC上的C语言编译工具主要为:

- Borland的Turbo C/C++和Borland C++;
- Microsoft的Visual C++;
- Watcom C/C++;
- Symantec C/C++。

其中,1983年Borland公司推出了Turbo Pascal,开创了编译工具的新时代;1987年,又发布了Turbo C 1.0,首次提供C语言集成开发环境工具;1990在Turbo C基础上推出了C++开发工具Turbo C/C++;1992发布Borland C/C++ 3.1,将C语言编译工具引向巅峰。而Watcom C/C++是以在DOS下能够产生最优化程序代码而闻名的,再加上当时最有名的DOS Extender厂商PharLap公司也是使用Watcom C/C++,因此Watcom C/C++在专业的C/C++程序员以及系统程序员心中是第一位的C/C++开发工具。虽然Microsoft的Visual C++开始时表现平平,然而凭着自己在操作系统上的优势和不懈的创新努力,终于在1996年左右将其余三个竞争者逐出C语言编译工具市场,成为在Windows平台上一枝独秀的编译器工具。

对于读者来说,选择什么样的编译工具与学习C语言本身没有太大关系。语法严谨的编译器尽管使读者感到困难,但对学习程序设计十分必要。而且由于转换编译环境是非常漫长且成本巨大的行为,因此大多数程序员往往偏好少数几种编译工具。下面给出

几个编译工具的比较。

(1) Turbo C 2.0，这是 C 语言早期强有力的工具之一，堪称 C 语言开发中的经典；它的简单、易用、快速使许多初学者毫不犹豫地选择它，然而它存在很大的不严谨性。

(2) Borland C/C++ 3.1，实事求是地说，这是一个非常完美的编译器工具；然而自 1994 年后 Borland 公司就再也没有对它进行技术更新的事实，使得它的开发功能实在太少了。在 256MB 内存 Pentium 4 计算机上只能使用最多 64KB 的数组，在真彩色的显示卡上只能显示 16 色，没有鼠标，没有汉字，没有网络，没有声卡，没有光驱，更别提 U 盘了。

(3) Microsoft Visual C++，在 Micrsoft Visual C++ 部门中，有许多员工来自类似 Borland 公司的软件开发公司，这使得 Microsoft Visual C++ 的技术集百家精华于一身，终成大器。即使已经没有任何对手的 Visual C++，依然在按照自己的计划发展着，从当年击溃其他竞争者的 Visual C++ 4.0 到今日的 Visual C++ 2005，Visual C++ 的 Roadmap 使得使用 Visual C++ 的程序员时刻拥有最新的技术，与 Microsoft 一同成长。

(4) Watcom C/C++，确实是最优化的 C 语言编译器。几经转折，Watcom C/C++ 已经成为“开放源码”的一员(http://www.openwatcom.org)，现在计算机专业人士可以从编译原理的角度来研究它了。

(5) GCC(DJGPP，http://www.delorie.com/djgpp/)，这是随 Linux 成长起来的 C 语言编译器，能在今日成为 Visual C++ 潜在竞争者的只有它了。由于 GCC 是整个 Linux 平台的支撑工具，对它的支持也在不断更新。然而 Linux 和 GCC 过于专业化的特点，使得初学者对此望而生畏。

1.1.3　集成开发环境(IDE)

集成开发环境(IDE，Integrated Development Environment)是指将程序开发中的编辑器、编译器、调试器合为一体，使得程序的编码开发过程能够在一个软件环境中完成。一个 IDE 至少要能够实现编辑、编译、连接、运行、调试功能。常用的 IDE 如下：

- Turbo C 2.01，http://bdn.borland.com/article/014102084100.html。
- Borland C++ 3.1，http://bdn.borland.com/。
- Visual C++，http://www.microsoft.com/china/msdn/vstudio/default.aspx。
- RHIDE，http://www.rhide.com/。
- Eclipse，http://www.eclipse.org/。
- Dev-C++，http://www.bloodshed.net/devcpp.html。
- MingW 编译器，http://www.mingw.org/。
- Kdevelop，http://www.kdevelop.org/。

其中，RHIDE、Eclipse、Dev-C++、MingW 编译器均是“开放源码”的。

许多人对一个编译器的印象就是 IDE，事实上 IDE 仅是一个开发环境。目前的编译器均是采用命令行方式工作的，只不过 IDE 对其做了更好的外壳包装。像 Visual C++ 与其 IDE 是紧密在一起的，而 RHIDE 与 GCC 则是松散的。读者可以先熟悉其中一种，其他 IDE 也就自然融会贯通了。

学习C语言不应只局限于使用一种编译环境，希望读者能掌握一种以上的编译和运行C程序的环境与工具，例如：Visual C++ 和 GCC。

1.1.4 快速应用开发(RAD)工具

原型化快速应用开发(RAD，Rapid Application Development)工具是指结合了直观的设计工具、优化的编译器、交互式调试器和完善的工具组件，从而为开发者提供了快速开发网络、桌面和数据库等应用程序所需要的工具的开发软件。常用的RAD工具有：

- Borland C++ Builder；
- Borland Delphi/kylix；
- Visual Basic；
- PowerBuilder。

RAD工具是目前程序员最好的开发工具，这类工具普遍采用了面向对象的设计方法。

1.2 Visual C++ 6.0开发环境及上机操作

Visual C++是目前在Windows操作系统上用得最多的C语言编译系统，现在常用的是Visual C++ 6.0(简称VC 6)版本，它可用于Win32平台应用程序(Application)、服务(Service)和控件(Control)的开发。

Visual C++从1998年发行6.0版本以来，又发行了Visual C++ 2002、Visual C++ 2003、Visual C++ 2005版本，这些不同的Visual C++版本的上机操作方法是大同小异的，掌握了其中的一种，就会举一反三学习使用其他版本。

1.2.1 Visual C++ 6.0简介

Visual C++ 6.0开发环境Developer Studio是在Windows环境下运行的一套集成工具，由文本编辑、资源编辑器、项目建立工具、优化编译器、增量连接器、源代码浏览器、集成调试器等组成。使用Developer Studio，不仅可以创建由Visual C++ 6.0使用的源文件和其他文档，而且可以创建、查看和编写任何与Active部件有关的文档(ActiveX文档)。

在Developer Studio中，可以在项目工作区中组织文件(File)、项目(Project)和子项目，可以使用工作区窗口来查看和访问项目中的各种元素。项目工作区可以含有多个项目，每个项目要么是顶层项目，要么是其他项目的子项目。

在Visual C++ 6.0中，可以使用向导(Wizard)、MFC类库和活动模板库(ATL)来开发Windows应用程序。向导用于帮助用户生成各种不同类型应用程序的基本框架。例如，可以使用Win32 Application和Win32 Console Application生成Windows应用程序和控制台程序，可以使用MFC AppWizard来生成完整的从开始文件出发的基于MFC类库的源文件和资源文件，可以使用MFC ActiveX Control Wizard生成创建Active控制所需要的全部开始文件(如源文件、头文件、资源文件、模块定义文件、项目文件、对象描述语言文件等)，使用Custom AppWizard来创建自定义的项目类型，并将其添加到创建项目

时的可用项目类型列表中。

在创建应用程序的基本框架后，可以使用 ClassWizard 来创建新类，定义消息处理函数，覆盖虚拟函数，从对话框、表单视图或者记录视图的控件中获取数据并验证数据的合法性，在自动化对象中添加属性、事件和方法。此外，还可以使用 Wizard 来定义消息处理函数并浏览实现文件(.cpp)。

Visual C++ 6.0 没有中文版，网上有也是将其菜单简单汉化的，而且许多翻译与英文原意不符，因此本书推荐使用 Visual C++ 6.0 英文版。其实 Visual C++ 6.0 英文界面并不是学习的障碍，其附带的英文 MSDN 信息倒是一大学习难题。

MSDN 是指 Microsoft Software Development Network(微软软件开发网络)，它是 Microsoft 为程序员进行 Windows 系统上的软件开发提供的开发工具，包含大量开发示例、帮助信息、技术信息、知识等；MSDN 是一部“开发者的百科全书”，信息庞大，查询方便，很多帮助项都有源程序示范。对于 Windows 开发来说，了解并利用 MSDN 是必需的途径。以前的 MSDN 基本上都是英文的，对于使用中文的读者来说，它是很大的障碍。但从 Visual C++ 2005 开始，Microsoft 提供了简体中文版本，并且 MSDN 中文化工作已经在进行中，读者可以从图 1.1 所示的网站(http://msdn.microsoft.com/library/chs/)中在线获取这些信息。

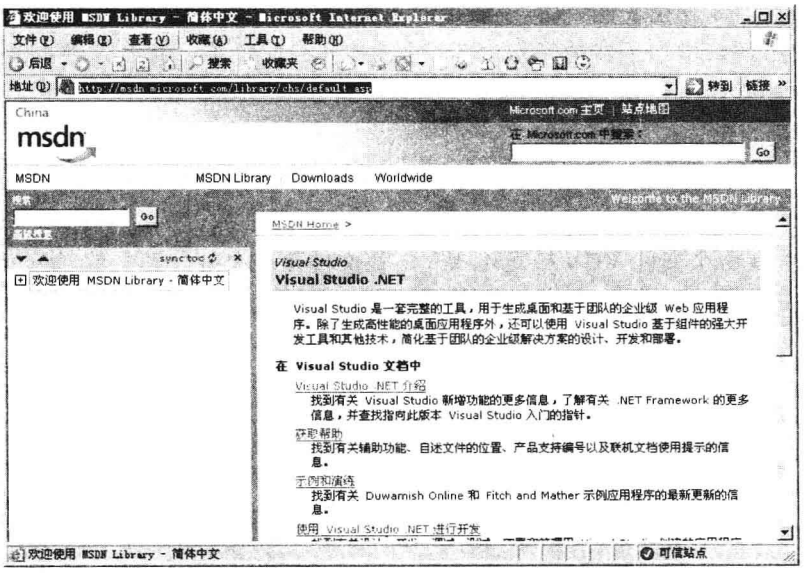


图　1.1

1.2.2　启动和退出 Visual C++ 6.0

Visual C++ 6.0 安装结束后，在 Windows“开始”菜单的“程序”子菜单中就会出现 Microsoft Visual Studio 子菜单。可以单击 Windows“开始”按钮，从“开始”菜单启动 Visual C++ 6.0，经过版权信息画面(如图 1.2 所示)后，Visual C++ 6.0 启动且进入开发环境中。

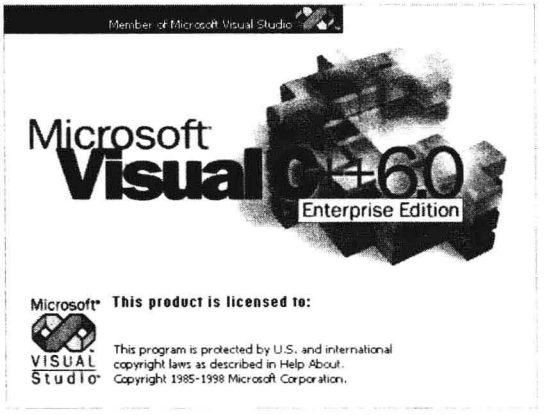


图 1.2

退出 Visual C++ 6.0 的方法是在 VC 6 主窗口中选择 File(文件)菜单下的 Exit 命令,或者将 VC 6 主窗口关闭。

1.2.3 配置 Visual C++ 6.0

Visual C++ 6.0 完成安装后,一般可以不必进行人工配置。然而在改变 Visual C++ 6.0 的环境后,或者增加了新的开发库后,就需要进行配置了。有时这些配置不正确时,还会使得 Visual C++ 6.0 编译时出现一大串莫名其妙的错误信息。

环境变量的设置如下:

Visual C++ 6.0 的环境变量主要包括:①Visual C++ 6.0 系统目录;②INCLUDE 头文件目录;③LIB 连接库文件目录。在 VC 6 系统目录(××××××\MSVS60\VC98\Bin)中有一个文件 VCVARS32.BAT,该文件是一个批处理文件,执行该文件后就会自动添加正确的 Visual C++ 6.0 设置路径和环境变量。

增加新开发库的配置如下:

Visual C++ 6.0 允许开发者增加新的开发库,通常这些开发库包含若干.H 文件和.LIB 文件(或.DLL 文件),本书这里仅讨论使用静态的开发库,因此不包括.DLL 文件。

如图 1.3 所示,在 Visual C++ 6.0 主窗口中选择菜单 Tools|Options…命令,打开

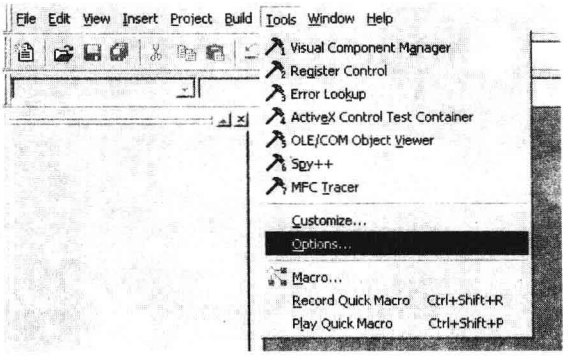


图 1.3

Options 对话框，选择 Directories 标签，如图 1.4 所示。

在 Options 对话框中就可以配置新开发库。其中 Show directories for 用来指明配置参数的类型，可能的类型如图 1.5 所示，下拉框中各个项目的含义如下：

- Executable files：表示可执行文件路径设置。
- Include files：表示编译头文件路径。
- Library files：表示编译库文件路径。
- Source files：表示源代码搜索路径。

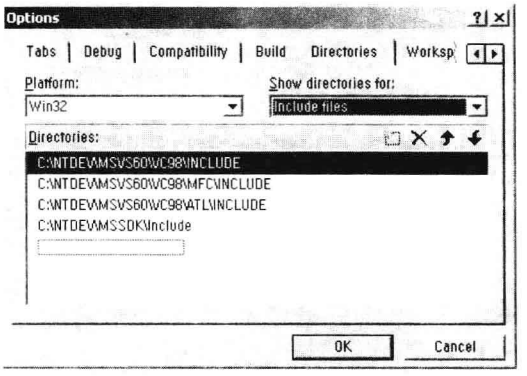


图　1.4

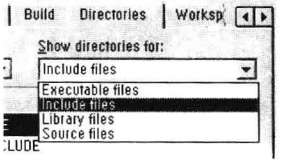


图　1.5

对于新开发库，通常需要设置 Include files 和 Library files。设置方法如图 1.6 所示。

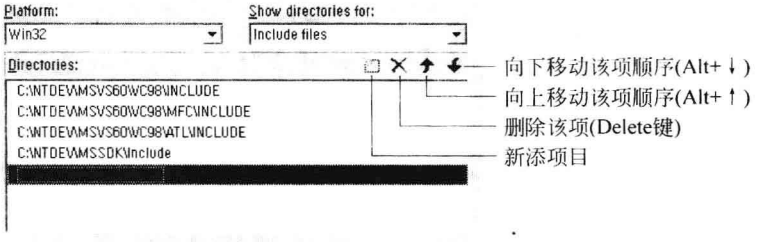


图　1.6

单击“新添项目”按钮，如图 1.7 所示。

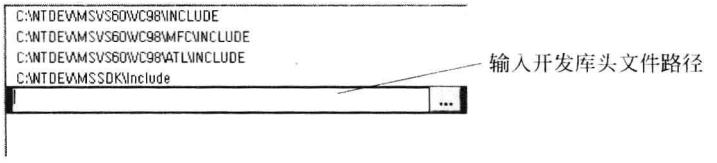


图　1.7

输入开发库头文件路径或者使用 ... 来浏览查找路径，如图 1.7 所示。

选择 Library files，按照上述步骤新添库文件路径。