

新世纪计算机基础课实验教程丛书

# C 语言程序设计 实验教程

编者 姜学锋 尹令平 张彦春

西北工业大学出版社

**【内容简介】** 本书是谭浩强主编的《C 程序设计》(第 3 版)教材的配套实验教程。全书分为 3 章,具体介绍了 Visual C++、Borland C++、Turbo C 开发工具的使用方法以及 Visual C++ 的调试方法。书中实验内容按课程教材和教学大纲要求编写,分验证型实验和设计型实验,突出综合性实验,并结合算法设计、数据结构知识要求设计了一些有难度的算法实验题目。本书还包括按 3 个项目专题编排的专题实验内容,其目的是使读者能获取设计 C 程序项目的初步知识和工程经验,为读者进行 C 语言课程设计提供了有价值的参考。

本书可作为大学本科 C 语言程序设计课程的实验教学教材,也可作为自学者的学习参考。

### 图书在版编目(CIP)数据

C 语言程序设计实验教程/姜学锋等编. —西安:西北工业大学出版社,2007.3  
(新世纪计算机基础课实验教程丛书)

ISBN 978-7-5612-2125-9

I. C… II. 姜… III. 程序设计—实验—高等学校—教材 IV. TP36

中国版本图书馆 CIP 数据核字(2006)第 108375 号

出版发行:西北工业大学出版社

通信地址:西安市友谊西路 127 号,邮编:710072

电 话:(029)88493844 88491757

网 址:www.nwpup.com

印 刷 者:陕西向阳印务有限公司

开 本:787 mm×1 092 mm 1/16

印 张:12

字 数:285 千字

版 次:2007 年 3 月第 1 版 2007 年 3 月第 1 次印刷

定 价:17.00 元

## 编 委 会

---

主 任：樊晓桢

副 主 任：张彦春

编 委：冯 萍 危建国 苗克坚

张艳宁 康慕宁 蔡皖东

姜学锋

# 总 序

近年来，我国在计算机应用、计算机软件和网络通信类相关专业的人才培养方面，取得了长足的进展。但学生在走进企业、科研单位之后，往往深刻地感觉到缺乏实际开发设计的经验，不善于综合运用所学理论，对知识的把握缺乏融会贯通的能力。综合考察目前高等院校教学大纲、课程设置以及内容安排等方面的情况，多数学校还是比较重视训练学生的实践能力的。但是从安排实践的内容来看，基本上围绕相关课程狭小的教学内容而展开，在难度上体现不够，缺乏综合性实验训练，而且实验内容高度抽象并脱离现实，学生很难获得针对具体问题的独立分析能力训练以及综合运用所学知识的整体训练机会。

由此可以看出，大多数学生实践能力训练与国内精品课程的要求相比较，还是存在一些差距。为此，我们针对当前高等院校计算机软、硬件和网络通信类相关课程教学中存在的问题，紧扣培养创新型学生的中心要求，参考了国内外知名大学相关课程成功的教学经验，设计编写了这套“新世纪计算机基础课实验教程”丛书，其目的就是通过实践训练，把知识获取和实践能力两个方面有机地结合起来。

这套“新世纪计算机基础课实验教程”丛书覆盖了计算机基础课的实验内容，包括“大学计算机基础”“程序设计”“微型计算机原理及应用”等课程，学生们可以在教师的指导下，逐步设计实现这些实验内容，并进行综合实验。通过实验，一方面可以结合课程的教学内容循序渐进地进行实验方面的实践训练；另一方面在参与一系列综合实验创新实验和自主实验的实践过程中，还能提高学生综合运用所学知识解决实际问题的能力，增强学生对相关课程具体内容的理解和掌握能力，培养学生对整体课程知识综合应用和融会贯通能力。

参加这套丛书编写的教师都有丰富的教学、科研等多方面的经验。实验教程中的实验内容，都来自教师们具体的教学科研实践，许多实验装置和软件都是由教师自己根据具体的教学要求设计完成的，再结合众多公司、厂商的大力支持，使得所选实验内容与教学内容配合紧密，实验难度与规模适宜。

最后，感谢西北工业大学出版社的大力支持，使出版这套丛书的计划得以实现。

丛书编委会

2006年8月

# 前 言

本书是谭浩强主编的《C 程序设计》(第 3 版)教材的配套实验教程。在正式出版前,已经在西北工业大学计算机学院以及面向非计算机专业的实验教学中使用多年,此次出版对原来的内容进行了一些修订和补充。

全书分为 3 章。第 1 章具体介绍了 Visual C++、Borland C++、Turbo C 开发工具的使用方法;第 2 章详细介绍了程序设计中的调试技术和 Visual C++ 的调试方法;第 3 章为与教材和教学大纲对应的基础实验内容。本书的特点是实验内容按课程教材和教学大纲要求编写,分验证型实验和设计型实验,突出综合性实验,并结合算法设计、数据结构知识要求设计了有难度的算法实验题目。这样由浅入深,循序渐进的内容安排,便于学生理解和掌握。验证型实验的目的是“学”,读者可以根据其中的具体实验步骤一步一步进行,从中“模仿”操作,并逐步掌握各种操作、技巧和知识概念。设计型实验的目的是“用”,读者根据实验题目要求,编写程序。

本书专题实验内容是按项目专题编排的实验,其目的是使读者能获取设计 C 程序项目的初步知识和工程经验,为读者进行 C 语言课程设计提供了有价值的参考。专题实验分三个专题,读者可以结合自身的实际情况来选做。

(1) Borland 文本与图形专题是针对 Borland C++ 3.1 或 Turbo C 2.0 环境下的项目开发,侧重训练在这两种环境中编制图形、界面程序。

(2) Windows 专题侧重训练利用 C 语言编制 Windows 窗口应用程序,使读者初步掌握 Windows SDK 开发的基本要领以及 Windows 窗口、菜单、对话框、控件、事件等常用元素的编程方法。

(3) Allegro 开发库专题利用 Visual C++ 6.0 开发控制台程序,侧重训练控制台环境下的图形、多媒体、事件处理程序。

限于篇幅,本书将专题实验内容制作成在线浏览的电子书,读者可到网上下载。

本书实验内容主要使用 Visual C++ 6.0 环境,部分内容在 Borland C++ 3.1、Turbo C 2.0 环境下也能正确运行,读者可以只选定一种编译系统来学习。

本书提供实验中的源程序和相关文件以及操作录像资料,读者可以自行从 FTP(<ftp://202.117.94.248/book/ceguide>)中下载。

本书第 2 章、第 3 章由姜学锋编写,第 1 章的 1.1 至 1.3 节及附录部分由尹令平编写,第 1 章的 1.4 节由张彦春编写。

由于编者水平有限,书中难免存在错误,敬请读者批评指正。

编 者

2006 年 12 月

# 目 录

第 1 章 开发环境及上机操作.....	1
1.1 C 语言开发环境简介 .....	1
1.2 Visual C++ 6.0 开发环境及上机操作 .....	4
1.3 Borland C++ 3.1 开发环境及上机操作 .....	58
1.4 Turbo C 2.0 开发环境及上机操作 .....	87
第 2 章 程序调试技术 .....	98
2.1 概述.....	98
2.2 程序调试的方法 .....	100
2.3 常见编译系统调试功能 .....	104
2.4 Visual C++ 6.0 调试方法 .....	107
第 3 章 基础实验内容.....	124
3.1 实验一 C 语言程序初步 .....	126
3.2 实验二 输入与输出 .....	133
3.3 实验三 选择结构 .....	137
3.4 实验四 循环结构 .....	140
3.5 实验五 数组 .....	144
3.6 实验六 函数与预处理命令 .....	148
3.7 实验七 指针 .....	153
3.8 实验八 结构体与位运算 .....	156
3.9 实验九 链表与数据结构 .....	158
3.10 实验十 文件.....	159
附录 常见编译错误信息.....	161
附录 1 Visual C++ 6.0 编译错误信息 .....	161
附录 2 Turbo C 与 Borland C++ 编译错误信息 .....	170
参考文献.....	183

# 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。在计算机发展的早期,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了各种高级语言,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)。

编译器可以生成用在与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、连接成为执行代码,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

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。当分析问题变得容易时,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),但更确切地应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。类似的,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)是这其中的佼佼者。

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使用最广泛的高级程序设计语言。那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 年代至 90 年代,PC 机上的 C 语言编译工具主要为:

- (1) Borland 的 Turbo C/ C++ 和 Borland C++ ;
- (2) Microsoft 的 Visual C++ ;
- (3) Watcom C/ C++ ;
- (4) 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,然而凭着它在操作系统上的优势和不懈的创新努力,终于在 1996 年前后将其余三个竞争者逐出 C 语言编译工具市场,使 Visual C++ 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。它简单、易用、快速,使许多初学者毫不犹豫地选择它,然而它较大程度地存在不严谨的缺陷。

(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,使得它的开发功能实在太少了。在 256MB 内存的奔 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 Microsoft Visual C++ 的开发部门中,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,终成利器。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++。它确实是最优化的 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上来研究它了。

(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,对它的支持也是在不断更新的,然而 Linux 和 GCC 过于专业化的特点使得初学者望而生畏。

### 1.1.3 集成开发环境

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:

(1) Turbo C 2.01,<http://bdn.borland.com/article/0,1410,20841,00.html>

(2) Borland C++ 3.1,<http://bdn.borland.com/>

(3) Visual C++,<http://www.microsoft.com/china/msdn/vstudio/default.aspx>

(4) RHIDE,<http://www.rhide.com/>

(5) Eclipse,<http://www.eclipse.org/>

(6) Dev- C++,<http://www.bloodshed.net/devcpp.html>

使用 MingW 编译器,<http://www.mingw.org/>

(7) Kdevelop,<http://www.kdevelop.org/>

其中,(4)~(7)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工具的开发软件。常用的 RAD 工具有:

(1) Borland C++ Builder;

(2) Borland Delphi/kylix;

- (3) Visual Basic;
- (4) PowerBuilder。

RAD 工具目前是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 VC6)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,就能举一反三掌握其他版本的使用。

### 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,网上也有将其菜单简单汉化的版本,但是许多翻译与英文原意不符,因此本书推荐读者使用 Visual C++ 6.0 英文版。其实 Visual C++ 6.0 的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 是必须的途径。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,读者可以从下面的网站 <http://msdn.microsoft.com/library/chs/> 中在线获取这些信息,如图 1.1 所示。



图 1.1 MSDN 中文网站

### 1.2.2 安装 Visual C++ 6.0

在使用 Visual C++ 6.0 开发应用程序之前,必须先将 Visual C++ 6.0 安装到计算机上。可以使用安装程序进行新的安装或更新已有的安装。如果是升级安装,安装程序在安装过程中仅仅复制新的或更新过的文件。此外,安装程序还允许根据配置情况对安装进行裁剪。

Visual C++ 6.0 的安装是十分简单和友好的,且具有高度自动化的特点,下面简单介绍一下安装的过程。

(1)在 CD-ROM 驱动器中插入 Visual C++ 6.0 系统光盘,运行“Setup.exe”命令。如图 1.2 所示,弹出“Visual Studio 6.0 Enterprise Edition”对话框,单击“Next”按钮。

(2)在其后弹出的“End User License Agreement”对话框和“Product Number and User ID”对话框中输入协议许可、ID 号、用户名称和公司名称等信息,再单击“Next”按钮,出现安装 Java 虚拟机的画面(见图 1.3),执行“Next”按钮后,安装过程会使操作系统重新启动再继续下一步的安装。

(3)弹出“Visual Studio 6.0 Enterprise Edition”对话框,如图 1.4 所示;选择“Custom”选项,然后单击“Next”按钮。

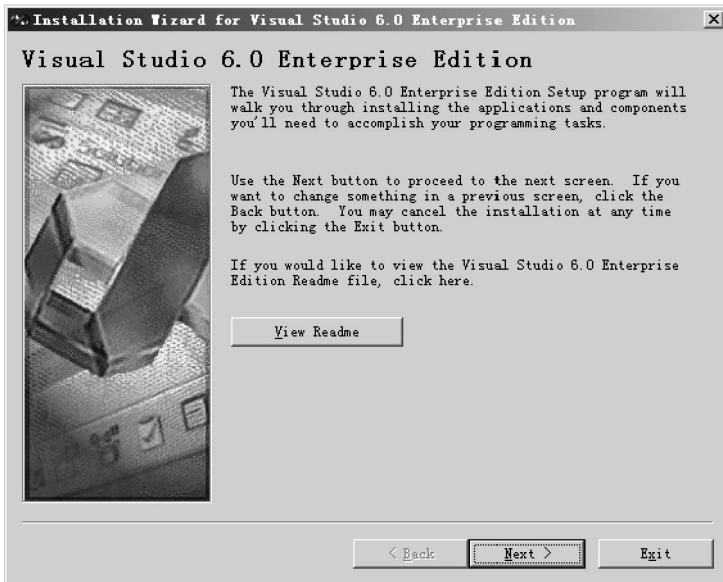


图 1.2 “Visual Studio 6.0 Enterprise Edition”对话框



图 1.3 安装 Java 虚拟机

(4)弹出“Choose Common Install Folder”对话框,如图 1.5 所示;选择安装“目的文件夹”之后,然后单击“Next”按钮。

(5)接下来出现“组件搜寻”对话框(见图 1.6),该过程将耗费较长的时间,须耐心等待一阵子,搜寻完成后会自动进入到下一步骤。

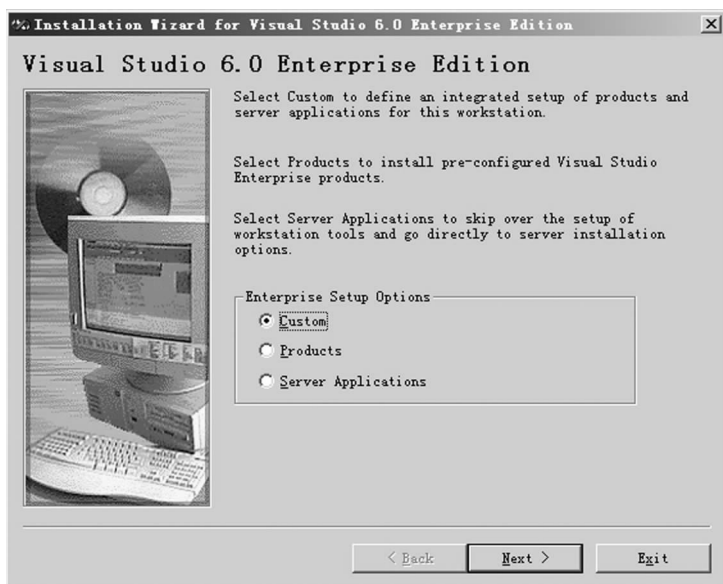


图 1.4 “Visual Studio 6.0 Enterprise Edition”对话框

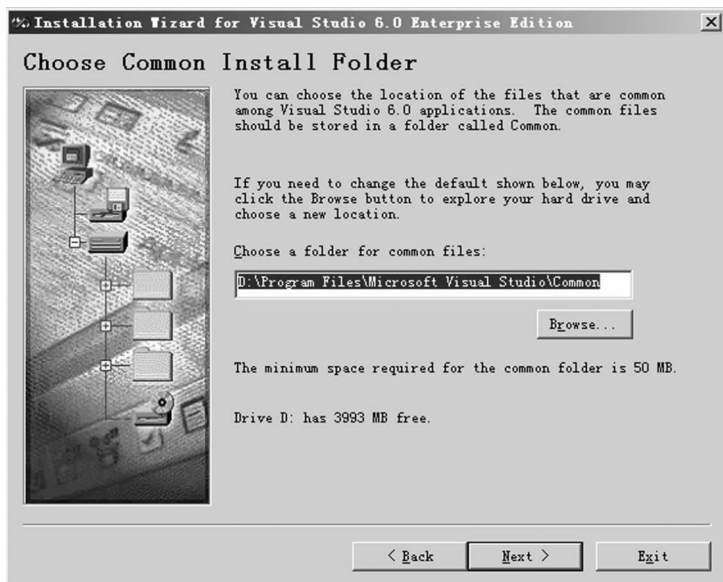


图 1.5 “Choose Common Install Folder”对话框

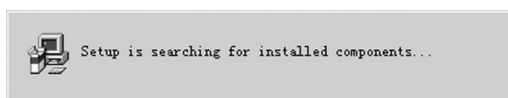


图 1.6 “组件搜寻”对话框

(6)弹出“Visual Studio 6.0 Enterprise Custom”对话框,如图 1.7 所示;选择安装“Microsoft Visual C++ 6.0”选项后,单击“Continue”按钮。

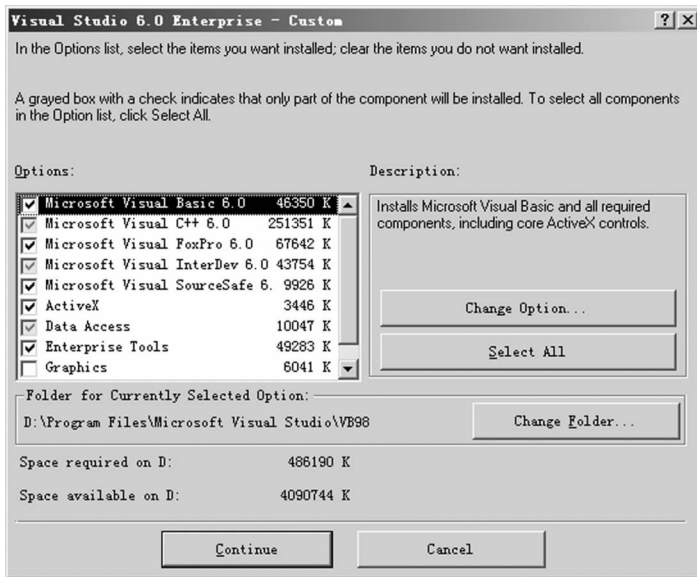


图 1.7 “Visual Studio 6.0 Enterprise Custom”对话框

(7)弹出“Setup Environment Variables”对话框,如图 1.8 所示。注意这里必须选择“Register Environment Variables”选项,使得安装程序能够自动进行 Visual C++ 6.0 环境变量和注册表的处理工作,否则以后需要人工进行这些复杂的处理工作,单击“OK”按钮。



图 1.8 “Setup Environment Variables”对话框

(8)由此便开始安装,屏幕上出现安装进度指示对话框,如图 1.9 所示。

(9)等待一段时间,安装完 Visual C++ 6.0 后,安装程序将会重新启动操作系统,如图 1.10 所示。

(10)系统重启后便进入“Install MSDN”对话框,如图 1.11 所示。如果用户想进一步安装 Visual C++ 6.0 所带的开发文档和例子,则单击“Next”按钮进一步安装。否则,单击“Exit”按钮退出安装。

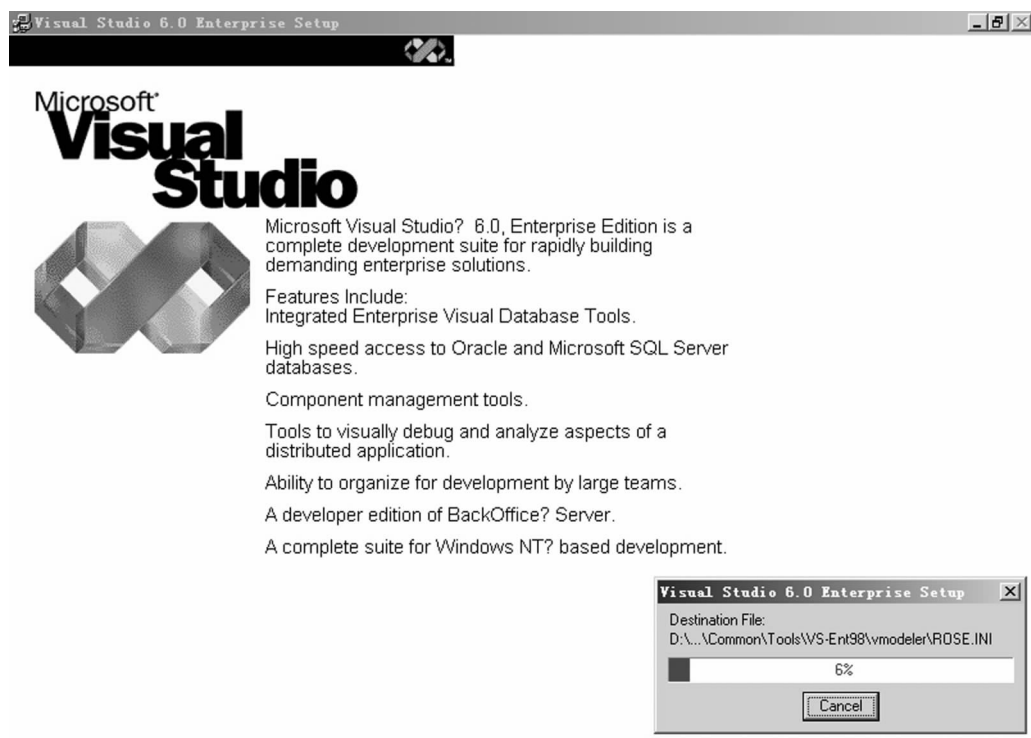


图 1.9 安装进度指示对话框

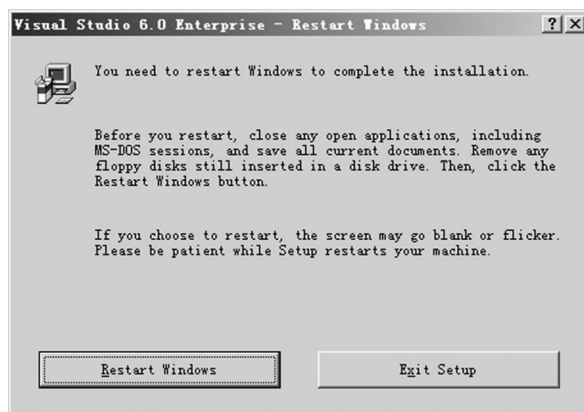


图 1.10

(11) 由于 MSDN 安装程序是简体中文的, 因此按照其提示进一步来到“安装类型”对话框, 如图 1.12 所示。

图 1.12 指明安装 MSDN 有三种选择:

- 1) 典型安装, 只安装必须的文件, 大部分的内容仍然在光盘上。
- 2) 自定义安装, 由用户选择, 需要有一定的专业知识。
- 3) 完全安装, 将全部内容安装到硬盘上, 与 Visual C++ 6.0 合在一起大约需要 1GB 空间。



图 1.11 “Install MSDN”对话框



图 1.12 “安装类型”对话框

(12) 为保证硬盘包含足够的开发文档及例子,在硬盘可用空间内尽可能选择“完全安装”,然后开始安装 MSDN,如图 1.13 所示。

(13) 完成 MSDN 的安装后,安装会自动将 MSDN 与 Visual C++ 6.0 聚合起来,从而可以在 Visual C++ 6.0 环境中在线使用 MSDN 信息。至此,Visual C++ 6.0 安装完成。



图 1.13 “安装进度指示”对话框

### 1.2.3 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(见图 1.14)后,Visual C++ 6.0 启动且进入开发环境中。

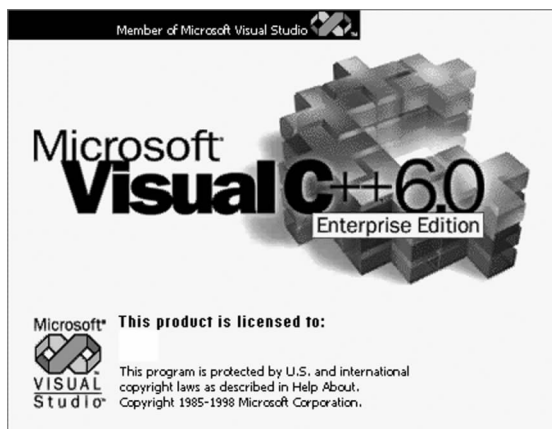


图 1.14 版权信息画面

退出 Visual C++ 6.0 的方法是在 VC6 主窗口中选择“File”(文件)菜单下的“Exit”命令,或者将 VC6 主窗口关闭。

此为试读,需要完整PDF请访问: [www.ertongbook.com](http://www.ertongbook.com)