

 普通高等院校“十二五”规划教材

C语言 程序设计教程

		李延珩	主 编
	汪 青	刘玉秀	副主编
李 鹏	李 楠	矫桂秋	参 编

普通高等院校“十二五”规划教材

C 语言程序设计教程

主 编 李延珩
副主编 汪 青 刘玉秀
参 编 李 鹏 李 楠 矫桂秋

内 容 简 介

本书是根据教育部高等学校非计算机专业计算机基础教学指导委员会的教学基本要求并参考《全国计算机等级考试二级 C 语言程序设计考试大纲（2013 年版）》编写的教材。

本书为适应全国计算机等级考试二级 C 语言程序设计考试的要求，运行环境采用 Visual C++。本书主要内容包括：基本概念和基础知识、结构化程序设计、数组、函数、文件、面向对象程序设计基础知识等。本书旨在将面向过程程序设计和基本的面向对象程序设计有机地结合起来，使读者通过本书的学习，能够掌握开发小型应用系统的基础知识和能力。

本书内容丰富、概念清晰、注重实践、讲解图文并茂。适合作为高等院校各专业的程序设计课程教材，也可作为培训或自学教材。

图书在版编目（CIP）数据

C 语言程序设计教程/李延珩主编. —北京：中国铁道出版社，2014.2

普通高等院校“十二五”规划教材

ISBN 978-7-113-11103-8

I. ①C… II. ①李… 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（2014）第 021124 号

书 名：C 语言程序设计教程

作 者：李延珩 主编

策 划：周海燕

读者热线：400-668-0820

责任编辑：周海燕 鲍 闻

封面设计：刘 颖

封面制作：白 雪

责任校对：汤淑梅

责任印制：李 佳

出版发行：中国铁道出版社（100054，北京市西城区右安门西街 8 号）

网 址：<http://www.51eds.com>

印 刷：北京新魏印刷厂

版 次：2014 年 2 月第 1 版

2014 年 2 月第 1 次印刷

开 本：787mm×1092mm 1/16 印张：17.5 字数：443 千

印 数：1~3 000 册

书 号：ISBN 978-7-113-11103-8

定 价：34.00 元

版权所有 侵权必究

凡购买铁道版图书，如有印制质量问题，请与本社教材图书营销部联系调换。电话：（010）63550836

打击盗版举报电话：（010）51873659

本书是在《C/C++程序设计》(作者:李延珩,中国铁道出版社2007年出版)基础上修订而成,此次修订是根据教育部高等学校非计算机专业计算机基础教学指导委员会《关于进一步加强高校计算机基础教学的意见》的教学基本要求及《全国计算机等级考试二级C语言程序设计考试大纲(2013年版)》为高校非计算机专业学生编写的。

这次修订为适应全国计算机等级考试二级C语言程序设计考试的新要求,将C的运行环境由Turbo C改为Visual C++,详细介绍了Visual C++6.0集成开发环境的使用方法。书中所有实例全部在Visual C++环境下运行。为了更加突出实验的地位,还将实验指导从主教材中独立出来,并在各个实验中都增补了部分实验内容,旨在通过大量实验内容的训练,提高学生使用计算机解决问题的能力。主教材中的各章均对实例进行了修订、调整、完善和补充。

本次修订在对面向过程程序设计做较全面的介绍之后,保留了面向对象程序设计基本概念的介绍和MFC基本编程的介绍。通过该课程的学习,学生可初步掌握使用Visual C++开发图形用户界面下的简单应用程序;在以后的学习或工作中,通过补充适当的知识就能够使用Visual C++编程解决各自专业领域的计算机应用问题。

全书共分12章。前10章是传统的面向过程程序设计内容,包括基本概念和基础知识、结构化程序设计、数组、函数、文件等;第11章介绍面向对象程序设计基础知识,内容包括类和对象、继承和派生、多态性等;第12章介绍Visual C++编程基础知识,包括Windows界面编程和基于对话框的应用程序实现。

本书建议课堂教学36~54学时,上机实践24~30学时。为了解决内容多、课时少的矛盾,建议在课堂教学上采用案例或问题驱动式的教学方法,从而培养学生解决问题及程序设计能力,对于复杂的问题通过案例教学化繁为简。

本套书包括主教材与上机指导两本。全书由李延珩任主编,汪青、刘玉秀任副主编,李鹏、李楠、矫桂秋参与编写。具体编写分工如下:第1章、第2章、第4章、第6章由汪青编写;第3章、第9章由李鹏编写;第5章、第10章由李延珩、李楠编写;第7章、第8章由刘玉秀编写;第11章、第12章由李楠、矫桂秋编写。

由于时间紧迫,加之作者水平有限,书中难免有疏漏之处,恳请读者批评指正。

编 者

2014年1月

第 1 章 C 语言程序设计概述	1
1.1 计算机程序设计语言概述	1
1.2 C 语言程序结构特点及其运行	2
1.2.1 C 语言程序的结构特点	2
1.2.2 C 语言程序的运行	5
1.2.3 Visual C++6.0 集成开发环境的使用	5
1.3 程序设计算法及其描述方法	8
1.3.1 算法的概念及特征	9
1.3.2 算法的描述方法	10
1.3.3 程序设计风格	12
习题	13
第 2 章 数据类型、运算符和表达式	15
2.1 C 语言的数据类型	15
2.2 标识符、常量和变量	16
2.3 整型数据	18
2.4 实型数据	21
2.5 字符型数据	23
2.6 运算符及表达式	26
2.6.1 运算符概述	26
2.6.2 算术运算符和算术表达式	27
2.6.3 赋值运算符和赋值表达式	31
2.6.4 逗号运算符和逗号表达式	32
2.6.5 不同数据类型间的转换	33
2.6.6 位运算	34
习题	35
第 3 章 顺序结构程序设计	38
3.1 C 程序的基本结构及 C 语句的种类	38
3.1.1 结构化程序的三种基本结构	38
3.1.2 C 语句的种类	38
3.2 数据输入输出的实现	39
3.3 标准输出函数——printf 函数	40
3.3.1 printf 函数的一般调用形式	40
3.3.2 printf 函数中常用的格式控制	41
3.3.3 调用 printf 函数时的注意事项	42
3.4 标准输入函数——scanf 函数	43

3.4.1	scanf 函数的一般调用形式	43
3.4.2	scanf 函数中常用的格式控制	44
3.4.3	调用 scanf 函数时的注意事项	44
3.5	字符输入输出函数	45
3.5.1	字符输出函数 putchar	45
3.5.2	字符输入函数 getchar	46
3.6	顺序结构程序设计举例	47
习题		49
第 4 章	选择结构程序设计	51
4.1	关系运算符和关系表达式	51
4.1.1	关系运算符及其运算优先次序	51
4.1.2	关系表达式	52
4.2	逻辑运算符和逻辑表达式	52
4.2.1	逻辑运算符及其运算优先次序	52
4.2.2	逻辑表达式	53
4.3	if 语句	53
4.3.1	单分支 if 语句	53
4.3.2	双分支 if 语句	55
4.3.3	多分支 if 语句	56
4.3.4	if 语句的嵌套	58
4.4	条件运算符和条件表达式	60
4.5	switch 语句和 goto 语句	60
4.5.1	switch 语句	61
4.5.2	goto 语句	63
4.6	选择结构程序举例	63
习题		66
第 5 章	循环结构程序设计	70
5.1	while 循环语句	70
5.2	do...while 循环语句	71
5.3	for 循环语句	73
5.4	循环的嵌套	74
5.5	循环的退出	76
5.5.1	break 语句	76
5.5.2	continue 语句	77
5.6	用 goto 语句构成循环	78
5.7	循环结构程序设计举例	79
习题		83

第 6 章 数组	87
6.1 一维数组	87
6.1.1 一维数组的定义	87
6.1.2 一维数组元素的引用	88
6.1.3 一维数组的初始化	89
6.1.4 一维数组的应用	90
6.2 二维数组	95
6.2.1 二维数组的定义	96
6.2.2 二维数组的引用	96
6.2.3 二维数组的初始化	98
6.2.4 二维数组的应用	99
6.3 字符数组	100
6.3.1 字符数组的定义及结束标志	100
6.3.2 字符数组的初始化	101
6.3.3 字符数组的输入输出	102
6.3.4 字符串处理函数	103
习题	107
第 7 章 函数	112
7.1 函数概述	112
7.2 函数的定义	113
7.2.1 函数定义的格式	113
7.2.2 函数的说明（声明）	114
7.3 函数的调用	115
7.3.1 函数调用的格式	115
7.3.2 函数调用的方式	115
7.4 函数参数传递和函数的值	117
7.4.1 函数参数传递	117
7.4.2 函数的值	119
7.5 数组参数的传递	121
7.5.1 数组元素作为函数参数	121
7.5.2 数组名作为函数参数	122
7.6 函数的嵌套调用和递归调用	126
7.6.1 函数的嵌套调用	126
7.6.2 函数的递归调用	127
7.7 变量的作用域与存储类别	129
7.7.1 局部变量	130
7.7.2 全局变量	131
7.7.3 变量的存储类型	133
7.7.4 函数的作用域	136

7.8 编译预处理.....	137
7.8.1 文件包含.....	137
7.8.2 宏定义.....	139
7.8.3 条件编译.....	143
7.9 综合应用.....	145
习题.....	146
第 8 章 指针.....	150
8.1 指针的概念.....	150
8.2 指针变量和指针运算符.....	152
8.2.1 指针变量.....	152
8.2.2 指针运算符.....	153
8.2.3 指针变量的运算.....	154
8.2.4 指针变量作为函数参数.....	159
8.3 指针与一维数组.....	163
8.3.1 引用数组元素的数组名法.....	163
8.3.2 指向数组元素的指针.....	163
8.3.3 数组名作为函数参数.....	165
8.4 指针与字符串.....	167
8.4.1 指向字符串的指针变量.....	167
8.4.2 字符串指针作为函数参数.....	170
8.5 指针与多维数组.....	172
8.5.1 多维数组的地址.....	172
8.5.2 多维数组元素的引用方法.....	174
8.6 指针数组与多级指针.....	177
8.6.1 指针数组.....	177
8.6.2 多级指针.....	179
8.6.3 指针数组作为 main 函数的形参.....	181
8.7 指针与函数.....	182
8.7.1 指向函数的指针.....	182
8.7.2 返回指针值的函数.....	185
习题.....	185
第 9 章 结构体、共用体与枚举类型.....	188
9.1 结构体.....	188
9.1.1 结构体说明.....	188
9.1.2 结构体变量的定义.....	189
9.1.3 结构体变量的引用.....	189
9.1.4 结构体变量的初始化.....	190
9.2 结构体数组.....	191
9.2.1 结构体数组的定义.....	191

9.2.2	结构体数组的引用	191
9.2.3	结构体数组的初始化	191
9.3	结构体与指针	192
9.3.1	结构体变量指针的定义	192
9.3.2	结构体数组指针	193
9.3.3	用结构体变量和指向结构体的指针作为函数参数	194
9.4	共用体	195
9.4.1	共用体类型的定义	195
9.4.2	共用体变量的定义	196
9.4.3	共用体变量的引用	196
9.5	枚举类型	197
9.6	用 typedef 定义类型	198
9.7	用结构体构成链表	199
9.7.1	结点类型的定义	200
9.7.2	单向动态链表	200
习题		205
第 10 章	文件	209
10.1	文件概述	209
10.1.1	文件的类型	209
10.1.2	文件类型指针	210
10.2	文件的打开与关闭	211
10.2.1	文件的打开	211
10.2.2	文件的关闭	211
10.3	文件的读写操作	212
10.3.1	读写一个字符的函数	212
10.3.2	读写一个字符串的函数	213
10.3.3	文件的格式化读写函数	215
10.3.4	读写数据块的函数	216
10.4	文件的定位	220
10.4.1	文件的顺序存取和随机存取	220
10.4.2	rewind 函数	220
10.4.3	fseek 函数	220
10.4.4	ftell 函数	221
10.5	程序设计举例	221
习题		223
第 11 章	C++面向对象编程基础知识	227
11.1	面向对象的基本概念	227
11.2	类和对象的声明与访问	230
11.3	构造函数和析构函数	232

11.4 继承性与多态性.....	233
习题.....	236
第 12 章 VC++基础知识及应用	237
12.1 Visual C++开发环境简介	237
12.2 编写第一个 VC++项目	238
12.2.1 创建项目工作区	238
12.2.2 使用程序向导创建程序基本结构	238
12.2.3 设计窗体并添加 C++代码	239
12.3 VC++基本界面程序开发	242
12.4 综合应用.....	250
习题.....	257
附录 A 常用 ASCII 码表	259
附录 B C 语言中的关键字	260
附录 C 运算符及其结合性	261
附录 D C 库函数	263
附录 E 全国计算机等级考试二级 C 语言程序设计考试大纲（2013 年版）	268
参考文献	270

第 1 章 C 语言程序设计概述

计算机的工作过程就是执行各种程序的过程，而程序是用计算机程序设计语言编写的。C 语言作为一种长盛不衰的高级程序设计语言，被广泛应用于操作系统和应用软件的开发。本章主要介绍计算机程序设计语言的相关概念、C 语言程序的基本结构、C 程序的运行过程、程序设计算法及其描述方法、程序设计风格等内容。

1.1 计算机程序设计语言概述

计算机系统由硬件系统和软件系统两大部分组成。硬件系统指构成计算机的各种机械部件和电子元件组成的设备和装置，是组成计算机系统的物质基础。软件系统是控制、管理计算机各硬件设备协同工作的所有程序文件和数据文件的总称，是计算机系统的头脑和灵魂。没有软件的计算机是一台“裸机”，什么也不能干，计算机的工作过程就是执行各种程序的过程。程序是指为完成特定功能而编写的指令的集合，这组指令依据既定的逻辑控制计算机的运行。所有程序都是用计算机语言编写的。

随着计算机技术的迅速发展，计算机程序设计语言经历了机器语言、汇编语言到高级语言的发展历程。程序设计方法也伴随着计算机硬件技术的提高而不断发展，可分为三个阶段，即面向计算机的程序设计、面向过程的程序设计和面向对象的程序设计。

高级语言使用一种接近人的自然语言和数学公式的形式编写程序。使用高级语言编写的程序与计算机的硬件结构及指令系统无关，具有更强的表达能力，可方便地表示数据的运算和程序的控制结构，能更好地描述各种算法，而且易于学习掌握。近几十年来，为解决不同的实际问题出现了几百种高级语言，常见的有 Java、C、C++、C#、Pascal、Python、Prolog、BASIC、FoxPro、Perl、Ruby 等。

C 语言是 1972 年由美国的 Dennis Ritchie 设计发明的，它由早期的编程语言 BCPL (Basic Combined Programming Language) 发展演变而来。早期的 C 语言主要用于 UNIX 系统。由于 C 语言的强大功能和各方面的优点，逐渐为人们所认识，到了 20 世纪 80 年代，C 语言开始进入其他操作系统，并很快在各类大、中、小和微型计算机上得到了广泛的使用，成为当代最优秀的程序设计语言之一。

C 语言是一种面向过程的结构化程序设计语言，它层次清晰，便于按模块化方式组织程序，易于调试和维护。C 语言的表现能力和处理能力极强，它不仅具有丰富的运算符和数据类型，便于实现各类复杂的数据结构，还可以直接访问内存的物理地址，进行位 (bit) 一级的操作，因此 C 语言集高级语言和低级语言的功能于一体，既可用于系统软件的开发，也适合于应用软件的开发。此外，C 语言还具有语言简练、源程序短、效率高、可移植性强等特点。

随着微型计算机的日益普及，出现了许多 C 语言版本。由于没有统一的标准，使得这些 C

语言之间出现了一些不一致的地方。为了改变这种情况,美国国家标准学会(ANSI)为 C 语言制定了一套 ANSI 标准,成为现行的 C 语言标准。

面向过程的程序设计必须按照算法的实现过程逐条编写程序,通知计算机一步一步怎么做。20 世纪 80 年代后,面向对象的程序设计概念日益普及。所谓面向对象,是通过类和对象把程序涉及的数据结构和对它施行的操作有机地组织成模块,对数据及数据处理细节进行最大限度的封装,从而使开发出来的软件易于重用、修改、测试、维护和扩充,C++就是在 C 语言的基础上增加了面向对象的思想发展而来的。为了满足网页开发的需要,1994 年又出现了 Java 语言,Java 语言脱胎于 C++,不仅支持面向对象的程序设计思想,且具有软硬件平台无关的特点,适合于进行网络开发。2000 年,Microsoft 公司推出的 Microsoft Visual Studio.NET 是一个具有公共语言子集的开发平台,实现了多种语言及其类库的无缝集成,C#是专为这一平台推出的全新语言。C#也派生于 C 和 C++,具有语法简洁、面向对象、与 Web 紧密结合、卓越的安全性、灵活性和兼容性俱佳等特点,成为 .NET 平台一流的网络编程工具。

从面向过程的 C 到面向对象的 C++,再到 Windows 平台上可视化的 C++开发工具 Visual C++,直到网络平台上面向对象的 Java 和 C#,各种适应新的编程环境的程序设计语言层出不穷,但很多语言都是基于 C 语言发展而来的,而且开发系统程序(如操作系统和嵌入式系统)和低层应用程序(如接口程序、通信和自动控制等)仍然是非 C 莫属。C 语言在程序设计语言中可谓常青树,在近年来的世界程序设计语言排名中一直稳定或稍有上升,对于理工科学生来说,C 语言是一种最佳的编程入门学习语言。

1.2 C 语言程序结构特点及其运行

本节我们来初步了解 C 语言程序的结构特点,并通过 Visual C++6.0 集成开发环境学习 C 程序的运行方法。

1.2.1 C 语言程序的结构特点

下面通过几个简单的 C 语言程序,学习 C 语言程序的结构特点。

【例 1.1】在显示器上分行显示“Welcome you.”和“First C program.”。

程序如下:

```
#include<stdio.h>
void main()                                /* 函数名 */
{                                           /* 函数体放在一对花括号中 */
    printf("Welcome you.\nFirst C program.\n"); /* 输出内容 */
}
```

运行结果如图 1-1 所示。

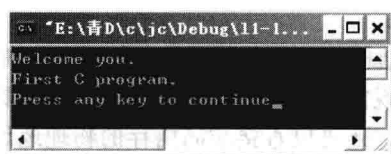


图 1-1 例 1.1 运行结果

本例是一个最简单的 C 语言程序。

第 1 行是编译预处理命令 include。当程序中要使用 C 系统提供的标准函数或其他文件时,

一定要用 include 命令包含函数对应的文件,以在编译前将对应文件嵌入到该处成为源程序的一部分。每一个程序都必须要有输出语句,因此必须包含标准输入输出头文件 stdio.h。

第2行是函数名 main()。一个 C 源程序由若干个函数组成,至少有一个用 main()命名的主函数,程序从主函数开始运行,也结束于主函数。函数都要有类型说明,放在函数名前, void 表示空类型,没有返回值。

第3行到第5行为函数体。函数体是程序的执行部分,由若干语句构成,写在一对花括号“{}”内,本例的函数体只有一条输出语句。C 语言的输入与输出操作都是由系统函数完成的, printf()为标准输出函数。printf()函数的输出内容取决于第一对双引号“”内的字符串,一般原样输出该字符串,仅在出现以“%”开头的输出格式时有变化。“\n”是换行符。

在一对“/*”和“*/”内的内容为注释。程序编译时,碰到“/*”将不对后面的内容进行编译直到出现“*/”为止。注释内容是对程序的解释以增加程序的可读性,不是执行代码。

【例 1.2】求-5 的绝对值。

程序如下:

```
#include<stdio.h>
#include<math.h>
void main()
{
    int x,y;
    x=-5;
    y=abs(x);
    printf("-5 的绝对值为: %d\n",y);
}
```

/* 定义程序中用到的变量 x 和 y */
/* 将初值-5 赋给变量 x */
/* 求-5 的绝对值, 结果赋给变量 y */
/* 输出变量 y 中的结果 */

运行结果如图 1-2 所示。

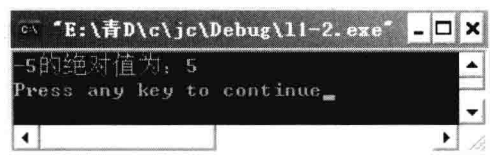


图 1-2 例 1.2 运行结果

本例使用了系统函数 abs()求绝对值,因此,要包含对应头文件 math.h。其他系统函数参见附录 D。%d 是输出格式,表示输出一个十进制整数,输出时其出现的位置将代入逗号后面对应变量的值。本例的%d 表示代入 y 的值 5,并以十进制整数形式显示。

【例 1.3】由键盘输入任意三个整数,输出其中的最大数。

我们用两种形式编写程序代码。

程序 1 如下:

```
#include <stdio.h>
void main( )
{
    int x,y,z,max;
    scanf("%d%d%d",&x,&y,&z);
    if(x>y)max=x;else max=y;
    if(max<z)max=z;
    printf("%d,%d,%d 中的最大值为: %d\n",x,y,z,max);
}
```

/* 运行时,由键盘输入 3 个整数 */
/* 比较 x 和 y, 将大者赋给 max */
/* 比较 max 和 z, 将大者赋给 max */
/* 输出结果 */

运行结果如图 1-3 所示。

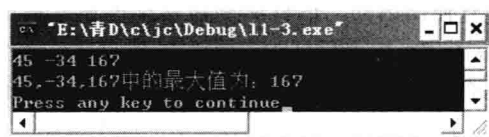


图 1-3 例 1.3 程序 1 运行结果

scanf()是系统提供的标准输入函数，该函数在程序运行时，为逗号后的各变量赋值，该函数要求变量名前须加取地址符“&”。用户每次运行程序，可根据实际情况随机灵活地输入不同的变量值。本例程序运行后，用“空格”隔开输入3个数后回车，系统自动顺次将数据赋值给变量x、y、z。

程序 2 如下：

```
#include <stdio.h>
void main()
{
    int max(int x,int y);      /* 对自定义函数 max 做声明*/
    int a,b,c,d;
    scanf("%d,%d,%d",&a,&b,&c);
    d=max(a,b);               /* 调用函数 max 找出 a, b 中大者赋给 d*/
    d=max(d,c);               /* 再次调用 max 找出 3 个数中大者赋给 d*/
    printf("max=%d\n",d);
}

int max(int x,int y)          /* 自定义函数 max, 从两个数中找出大者*/
{
    int m;
    if(x>y) m=x;              /* 比较 x 和 y, 将大者赋给 m */
    else m=y;
    return(m);               /* 返回找到的大数*/
}
```

运行结果如图 1-4 所示。

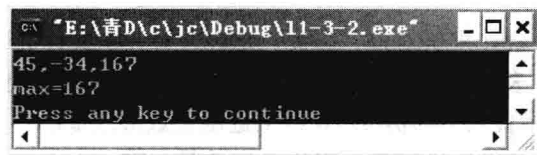


图 1-4 例 1.3 程序 2 运行结果

程序 2 中包括两个函数 main 和 max。max 函数的功能是找到任意两个数中的较大数，并将结果返回给调用它的函数。本例通过在主函数中两次调用 max()函数，完成从 3 个数找最大数功能。把一些功能相对独立的程序段编成一个函数，避免了重复代码，降低了代码量和复杂度，可以反复调用，C 语言就是通过函数实现了程序的模块化。

由上面例题可以看出：

(1) C 语言程序是由函数构成的。一个 C 源程序不论由多少个文件组成，包含多少个函数，都有且仅有一个 main 函数，即主函数。一个 C 程序总是从 main 函数开始执行，而不论 main 函数在程序中的什么位置，程序的执行也结束于主函数，其他函数通过函数调用被执行。

(2) 每个函数包括函数名和函数体两大部分。除主函数外,其他函数按C语言命名规则自主命名。函数体包含在一对大括号“{”和“}”中,由若干语句构成。

(3) C程序书写格式自由,可在一行内写多条语句,也可将一条语句写在多行上(上行末加续行符“\”)。每条语句必须以分号“;”结尾。

(4) 程序代码一般用小写字母书写,C语言严格区分大小写。

(5) 注释部分仅增加程序的可读性,是给用户看的,不会被编译,可以出现在程序的任意地方。写在双边注释符“/*...*/”中的注释内容可以占部分行、单行、多行;写在单边注释符“//”后的注释内容仅占一行,至行末为止。

(6) 源程序中的预处理命令通常放在源程序的最前面,预处理命令之后不能加分号。

(7) 函数体内语句一般按四大功能顺次排列,即首先对变量与函数做声明,然后为变量赋初值,再进行数据计算与处理,最后输出结果。

1.2.2 C语言程序的运行

使用高级语言编写的程序统称为源程序,源程序不能为计算机所识别。一个C语言程序的执行需要经过编辑、编译、连接和运行4个步骤,才能得到程序结果,如图1-5所示。

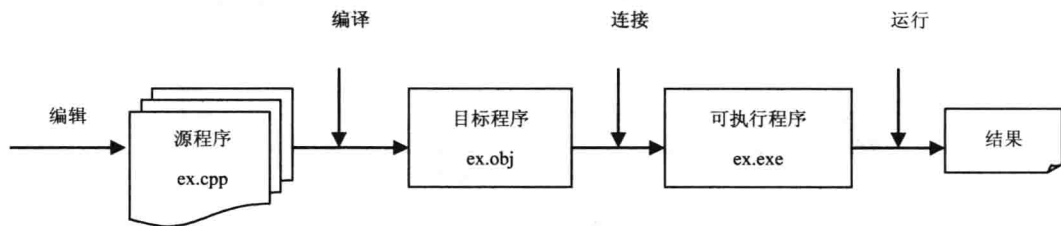


图 1-5 C语言程序运行步骤

(1) 编辑。通过纯文本编辑软件输入和编辑源代码,并保存成源程序文件。在VC++6.0环境下,源程序文件的扩展名为.c或.cpp,如ex.cpp。

(2) 编译。通过编译系统将源程序代码编译成计算机能直接识别的二进制形式,即目标程序。在编译过程中,系统自动检查源程序中的句法和语法错误,如发现错误,会报告错误类型及在程序中的位置,帮助用户修改错误。如未发现错误,则自动生成对应的目标文件。目标程序文件是与源程序同主名的.obj文件,如ex.obj。

(3) 连接。将形成的目标程序、库函数或其他目标程序连接装配成一个可执行程序。大多数的编译系统都允许使用一个命令同时完成编译和连接操作。可执行程序文件是与源程序同主名的.exe文件,如ex.exe。

(4) 运行。运行可执行程序,得到程序结果。

1.2.3 Visual C++6.0 集成开发环境的使用

集成开发环境(Integrated Development Environment, IDE)是集程序的编辑、编译、连接以及运行等功能于一体的程序开发软件,利用IDE可以大大提高程序开发的效率。目前,用于C程序开发的集成环境有Turbo C、Borland C、Microsoft C和Microsoft Visual C++等。Microsoft Visual C++6.0(简称VC++6.0)是Microsoft公司1998年推出的基于Windows平台的集成开发环境,不仅可以开发C++程序,也可以开发C程序。

本书内容均针对 VC++6.0 编写。

在 VC++6.0 中，一个 C 应用程序被称为一个项目或工程（project），它是由应用程序所需要的所有文件组成的一个有机整体，一般包括源文件、头文件和资源文件等。项目能自动将其包含的文件进行分类和管理，从而大大减轻了程序员的负担。项目被置于项目工作区（workspace）的管理之下。一个项目工作区可以包含多个项目，甚至是不同类型的项目，这些项目之间相互独立，但共用一个项目工作区的环境设置。

1. 在 VC++6.0 中运行一个 C 程序的步骤

（1）启动 VC++6.0。选择“开始”→“程序”→“Microsoft Visual C++6.0”命令。打开图 1-6 所示的窗口。

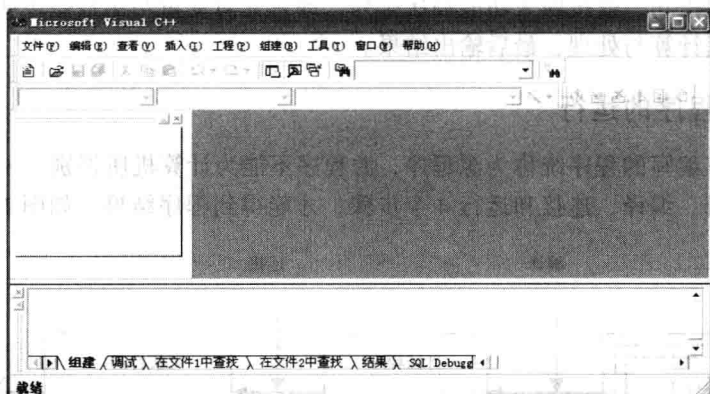


图 1-6 Visual C++6.0 主窗口

（2）创建项目工作区。在 VC++6.0 窗口中，选择“文件”→“新建”命令，弹出图 1-7 所示的“新建”对话框，选择“工程”选项卡中的 Win32 Console Application，指定项目名称和存放位置，然后单击“确定”按钮，出现图 1-8 所示对话框，选择“一个空工程”单选按钮后依次单击“完成”和“确定”按钮，创建项目和项目工作区，并在左侧项目工作区中显示与项目有关的信息。



图 1-7 新建项目对话框

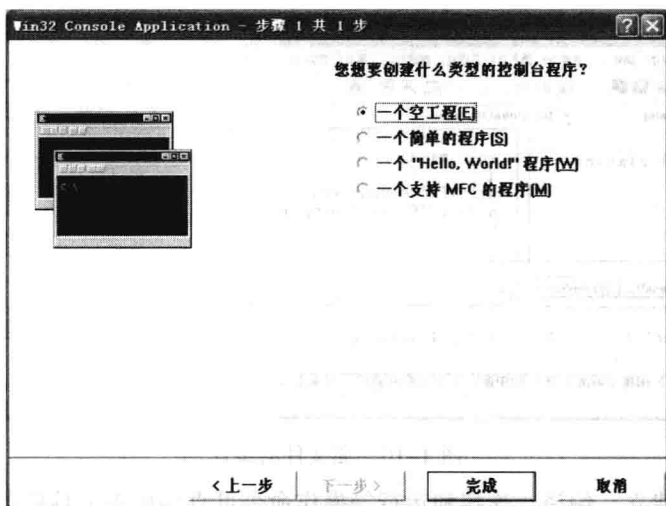


图 1-8 Win32 Console Application 对话框

(3) 创建源文件。打开“新建”对话框中的“文件”选项卡，选择 C++ Source File，指定文件名，如图 1-9 所示。单击“确定”按钮，创建源程序文件。

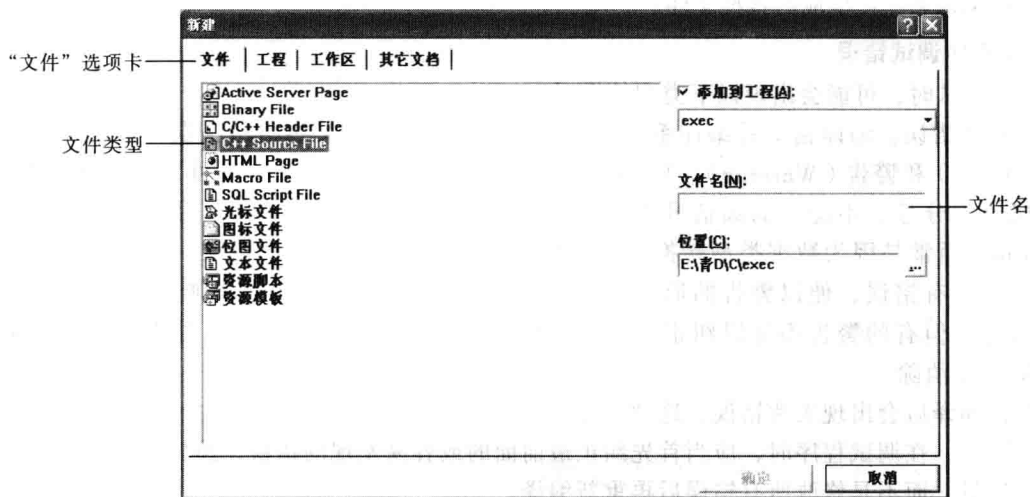


图 1-9 新建文件对话框

(4) 编辑源程序文件。在编辑窗口中输入编辑源代码，随时单击“保存”命令。

(5) 编译。选择“组件”→“编译”命令，对源程序进行编译，生成目标文件。编译结果会显示在下面的输出窗口中，如图 1-10 所示。如果编译有错误必须修改错误重新编译，直至编译通过。

(6) 连接。编译通过后，选择“组件”→“组件…”命令，生成可执行文件。

(7) 运行。选择“组件”→“执行…”命令，执行程序，得到结果。

另外，如果只是运行 C 程序，可以简化以上操作。步骤如下：打开 VC++6.0，单击“新建文本文件”，输入源程序，然后保存文件（文件名必须加.c 或.cpp 扩展名），再进行编译和运行即可。