

计算机类主干课程系列教材

程序设计： C语言

◎ 韩海 梁庆中 主编

 科学出版社

计算机类主干课程系列教材

程序设计:C语言

韩海 梁庆中 主编

科学出版社

北京

版权所有,侵权必究

举报电话:010-64030229;010-64034315;13501151303

内 容 简 介

本书从数据在内存中的存储形式入手,由浅入深地介绍C语言的语法知识,以需求作为驱动,先引导读者发现此时需要有某种功能,再介绍如何用C语言实现这种功能,并通过有趣味性和实用性的编程实例讲解程序设计的方法和技巧。在讲解循环、函数、指针等C语言的难点时,对计算机硬件、数据结构、软件工程等后续知识略有介绍,引导读者进一步探究软件开发所需要的知识体系,同时也说明计算机专业知识之间是相互关联的。每一章的后面设置了“实践建议”,引导读者通过实际操作巩固书中所述的相关规则、规定。本书以VC++6.0为编程环境,程序示例结构规范,完整程序均在该环境下测试通过。

本书面向仅具备少量计算机基础知识的初学者,内容全面、结构合理、逻辑性强,各章节各个知识块之间联系紧密、环环相扣,可以作为高等学校计算机专业的教材,用于非计算机专业学生时,可对部分章节作适当删减。

图书在版编目(CIP)数据

程序设计:C语言/韩海,梁庆中主编. —北京:科学出版社,2015.1
计算机类主干课程系列教材
ISBN 978-7-03-042589-8

I. ①程… II. ①韩… ②梁… 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272476号

责任编辑:张颖兵 闫 陶/责任校对:肖 婷
责任印制:高 嵘/封面设计:陈明亮

科学出版社 出版

北京东黄城根北街16号

邮政编码:100717

<http://www.sciencep.com>

武汉市首壹印务有限公司印刷

科学出版社发行 各地新华书店经销

*

开本:787×1092 1/16

2015年1月第一版 印张:19 3/4

2015年1月第一次印刷 字数:420 000

定价:43.00元

(如有印装质量问题,我社负责调换)

前 言

C 语言作为学习计算机程序设计的第一门基础课程,不只是学习 C 语言的基本语法规则,应从“命令计算机按步骤执行”的角度掌握这种语言,更重要的是领会“过程式程序设计”的思想。为此,本书在讲解 C 语言的各个要点时,总是从解决实际问题的需求出发,首先是“我想要命令计算机做什么”,然后才是如何命令计算机这样做。

虽然 C 语言是比较成熟的大众化编程语言,现行教材也很丰富,但本书在讲解知识点及安排实践环节方面仍有独到之处。

(1) VC++6.0 作为编程环境。让读者从学习 C 语言的第一天起就熟悉实用的软件开发环境,不仅让读者有实战的感觉,还可以体会标准 C 是 C++ 的一部分。由于该环境不以 DOS 窗口为主要工作方式,本书弱化了 printf 和 scanf 等输入/输出方法中过于繁杂的格式规定,仅保留基本的输入/输出手段。

(2) 关于原理的讲解详尽、透彻。本书不仅告诉读者“是什么”,而且努力告诉读者“为什么”,从而使读者在理解的基础上知道“怎么做”。例如,函数是 C 语言的基本单位,执行一个 C 语言程序总是在不断地调用函数和从函数返回。本书以栈中存储入口参数及返回地址的方式详细讲解如何从调用语句进入被调函数内部,又如何从函数内部返回调用语句继续执行,从而解决了递归原理的问题。本书还明确了 main 是自定义函数,运行程序从操作系统调用 main 函数开始,以解答关于“程序运行时如何处理程序中罗列的一个个函数”的疑问。

(3) 知识点之间逻辑性强、次序合理。本书在安排各知识点的次序时以知识点之间的逻辑关系为依据,“后面内容需要以前面章节为基础”的现象比比皆是。例如,第 4 章先讲解完整的 if...else 结构,然后说明没有 else 的 if 语句,以“后者是前者的特例”归结两者的关系。又如,函数与指针的有关知识点存在相互交叉的现象,本书选择指针在前、函数在后的方式安排内容,让读者在掌握指针的概念及基本用法之后学习“以指针作为函数的入口参数”、“返回指针的函数”、“指向函数的指针”等内容。

(4) 知识点与示例程序结合紧密。本书在讲解知识点之后都精心安排一定数量的例子程序,这些例子不仅与知识点联系紧密,还能体现相关用法的特点。例如,第 5 章讲解 while 和 do...while 的两个例子都能说明关于这两个语句的要点:例 5.7“辗转相除法求最大公约数”是预先不知道循环次数而知道循环结束条件的情况,这是 while 语句的特点;例 5.9“用迭代法求平方根”是先做一次操作再判断循环条件,这是 do...while 语句的特点。这样的例子显然比数列的例子更有说服力,可以回答读者在学习该知识点时的疑惑:这个例子用 for 语句不是更合适吗?

(5) 例子程序有一定的趣味性,对于有一定难度的例子会有较详细的分析。本书所选例子程序具有较好的代表性和趣味性,如水仙花数、验证哥德巴赫猜想、杨辉三角形、奇幻方等,让读者看到用 C 语言编程不仅可以解决简单的数值计算问题,还有更广阔的应用空间。

循环是 C 语言的第一个难点,从第 5 章开始,本书在讲解有一定难度的例子程序时都增加了“分析”环节,如例 5.16,引导读者如何由粗到精地一步步把自然思维转变为最后的程序。

(6) 用“小知识”和“思考题”提醒读者一些注意事项。本书在正文中安排了一些清新醒目

的“小知识”和“思考题”，一般用于提醒读者在相关知识点上需要对书中所述用法给予关注，也用于讲述编程技巧。

(7) 每章安排“实践建议”。本书每一章后都安排了“实践建议”，主要是针对本章有关 C 语言规则、规定的验证性实验。例如，1.5 节讲述了在计算机中以补码存储整数和以符号、阶码及尾数的方式存储实数，章后“实践建议”中给出一个显示二进制存储形式的程序，观察该程序的运行结果可以验证数据是否按 1.5 节所述的方式存储。

为了配合相关章节的学习并巩固每一章的知识要点，每一章后还配有一定量的习题，所选习题与知识点联系紧密。虽然书后附录已经给出习题解答，但对于编程类问题，一题多解是非常常见的，希望读者不拘泥于答案给出的解法，开拓思路，把程序设计得更简洁、更有技巧。

本书编者是江汉大学数学与计算机科学学院计算机专业的专职教师，长期从事计算机专业课程教学工作，具有多年的教学经验和心得。本书在编写过程中，力求把 C 语言语法知识和程序设计思想有机结合，以前者为手段，以后者为目的。本书的编写得到了江汉大学数学和计算机科学学院教师的大力支持，在此表示衷心的感谢。

由于编者水平有限，书中难免有不足之处，敬请广大读者批评指正。

编者
2014 年 3 月

目 录

第 1 章 C 语言概述	1
1.1 计算机语言	1
1.2 C 语言的发展与特点	3
1.3 VC 环境下的上机操作步骤	3
1.4 数制及相互转换	8
1.5 数据存储	12
1.6 流程图	16
实践建议	18
习题 1	19
第 2 章 数据与运算	21
2.1 数据类型	21
2.2 常量	22
2.3 变量	23
2.4 常用运算符与表达式	26
2.5 常用的数学函数	29
2.6 C 语言对运算符的扩展	30
2.7 字符型数据	32
实践建议	34
习题 2	35
第 3 章 输入/输出与顺序结构	36
3.1 标准输入/输出函数	36
3.2 顺序结构的基本流程	40
3.3 字符型数据的输入/输出	44
实践建议	46
习题 3	46
第 4 章 分支结构	49
4.1 逻辑表达式	49
4.2 if 语句	51
4.3 多路分支	58
实践建议	61
习题 4	61
第 5 章 循环结构	63
5.1 循环的基本概念	63
5.2 for 语句控制的循环	64
5.3 while 语句控制的循环	71

5.4	do...while 语句控制的循环	75
5.5	break 和 continue	77
5.6	循环的典型应用	80
5.7	循环嵌套	86
	实践建议	91
	习题 5	92
第 6 章	数组	95
6.1	数组与存储分配	95
6.2	访问数组元素	98
6.3	二维数组	104
6.4	字符数组与字符串	110
	实践建议	118
	习题 6	120
第 7 章	指针	125
7.1	指针的基本概念	125
7.2	指针的基本用法	129
7.3	指向数组元素的指针	132
7.4	指针数组	139
7.5	二级指针	141
	实践建议	146
	习题 7	148
第 8 章	函数	152
8.1	函数的基本概念	152
8.2	自定义函数及其调用	155
8.3	函数调用过程	161
8.4	变量的类别	166
8.5	函数的递归调用	173
8.6	与函数有关的指针	180
8.7	模块化编程	187
	实践建议	190
	习题 8	192
第 9 章	结构体	195
9.1	在内存中存放二维表数据	195
9.2	结构体变量	196
9.3	结构体数组	203
9.4	与结构体有关的指针	205
9.5	链表	210
9.6	复杂的链式结构	222
9.7	共用体	223
	实践建议	225

习题 9	228
第 10 章 文件	230
10.1 文件的相关概念	230
10.2 文件操作函数	235
10.3 有关文件操作的程序设计	241
实践建议	246
习题 10	247
第 11 章 预处理与位运算	249
11.1 预处理命令	249
11.2 枚举类型	254
11.3 位运算	257
实践建议	261
习题 11	262
附录 A 标准 ASCII 表	263
附录 B 运算符优先级与结合性	264
附录 C 常用库函数	265
附录 D 习题解答	268
参考文献	306

第 1 章 C 语言概述

要使计算机按照人的意图工作,就必须使用计算机能接受并执行的指令。用二进制的 0 和 1 编写的指令就是面向机器的,被称为机器指令。为解决特定问题而设计的指令序列就是程序。由于不同型号计算机的指令系统各不相同,所以可以在一种计算机上执行的程序有可能在另一种计算机上不能执行。高级语言的出现增强了程序的可移植性,高级语言程序必须翻译成机器指令后才能被计算机执行。

1.1 计算机语言

计算机语言也称为程序设计语言,是用来书写程序的语言,是人命令计算机完成工作的工具。每一种语言都规定了一组有确定含义的符号、符号组合和一组规则。根据规则构成的符号串的总称就是语言。计算机语言经历了由机器语言、汇编语言到高级语言、第四代语言的发展历程。

机器语言是二进制形式的机器指令的集合。机器语言的指令和涉及的数据都是二进制形式。直接采用机器语言来书写程序是一件非常烦琐而枯燥的工作,而且容易出错。用各种字母组合或单词缩写代替相应的机器指令,用十进制代替二进制写数据,这样更便于程序设计,于是产生了汇编语言,它是符号化的机器语言。例如,在以 Intel 8086/8088 为(中央处理器, Central Processing Unit, CPU)的计算机中,把整数 1 存入寄存器 AX 的操作,机器指令是 101110000000000100000000,而用汇编指令写出来则是 MOV AX,1。

显然,汇编语言比机器语言更直观和容易记忆,它用 MOV 表示“传送”,用 ADD 表示“做加法”等。当然,计算机不能直接按照这类符号执行相应的操作,而需要把它们翻译成相应的机器指令,翻译工作被交给一个称为汇编程序的翻译程序来完成。被翻译的汇编语言程序称为源程序,由汇编程序翻译出的结果是目标程序,而翻译的过程则称为汇编。图 1.1 所示为汇编的过程。



图 1.1 把汇编语言源程序翻译成目标程序

用汇编语言编写程序仍然很烦琐、费时、容易出错且难以维护,没有经过专业学习的人几乎无法使用这种语言。直到高级语言产生以后,程序设计才逐渐大众化。高级语言的表示方法更接近数学方法和自然语言。例如,命令计算机完成两个数据相加并把结果存放到变量 c 中,用高级语言可写成 $c=a+b$ 。用高级语言书写的程序不用修改或只作少量修改就可用于各种不同的计算机。当然,计算机也不能直接“认识”高级语言的语句,也必须经过翻译。这里承担翻译工作的程序称为编译程序,被翻译的高级语言程序称为源程序,翻译的结果是目标程序。一条高级语言的语句通常会翻译成一组机器指令。图 1.2 所示为编译的过程。



图 1.2 把高级语言源程序编译成目标程序

早期的程序规模较小,“翻译”得到的目标程序可以在计算机上直接运行。但随着程序规模的不断扩大,尤其是模块化编程方式出现之后,一个程序可以交由多人分别进行设计,各人编写的源程序可以分别编译。这种情况下生成的目标程序不能直接运行,需要把几个目标程序“组装”起来,形成最终的可执行程序。

另外,一些经常使用、能够完成一定功能的规范化程序段被集中起来存放在“程序库”中,当设计人员需要用到某种功能时,可直接使用相应的程序段,而不必重复设计这些代码。组装目标程序时可以把程序库中的程序段也一并组装到最后的可执行程序中。“连接程序”用来完成“组装”的工作。图 1.3 所示为从源程序到可执行程序的处理过程。

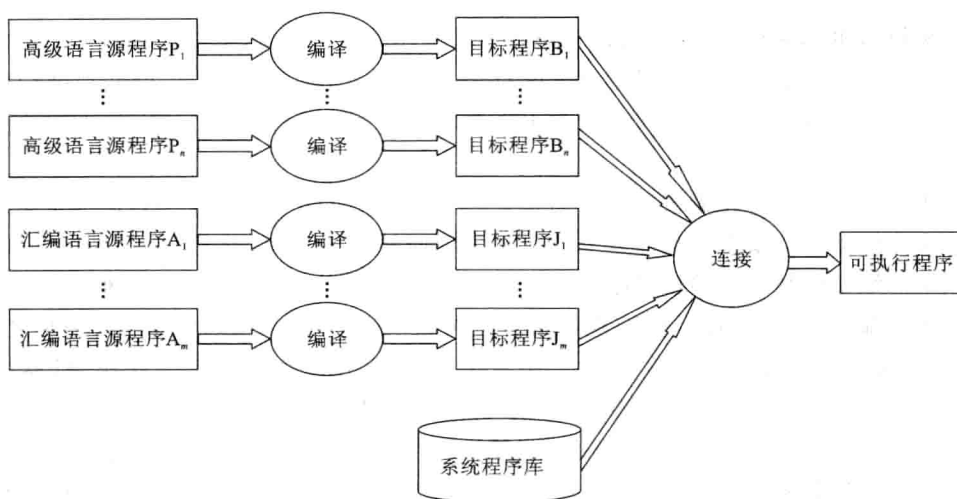


图 1.3 从源程序到可执行程序的处理过程

与汇编语言相比,高级语言程序经过编译得到的目标代码较长,执行程序的效率较低。尽管如此,由于高级语言与自然语言和数学语言较为接近,容易被人理解,因而得到了广泛应用。目前已经出现了多种高级语言,它们的语法规则和应用范围不尽相同。其中 C 语言以简洁、高效和功能强大等优点脱颖而出,受到广大程序员的欢迎,成为最普及的几种程序设计语言之一。

第四代语言的出现让程序设计更加便捷,它以数据库管理系统所提供的功能为核心,通常具备开发大型应用软件的开发生态环境,以简洁的命令实现,如报表生成、多窗口表格设计、菜单生成、图形图像处理等功能,为程序员提供了更简洁的编程方式。与高级语言相比,第四代语言的特点之一是“非过程化”,用户只需告知系统做什么,而无须说明怎么做,因此可大大提高软件生产率。

目前实际使用的软件开发环境往往是高级语言与第四代语言的结合体,在设计用户界面、操作数据库时可以使用其中的第四代语言部分,在面对一些计算、处理类问题时则可以使用高级语言部分。VC++6.0 就是这样一个软件开发平台,本书将以该平台作为开发环境介绍 C

语言的语法规则、编程技巧,以及在该平台上进行编程练习的操作步骤。

1.2 C语言的发展与特点

C语言由BCPL(Basic Combined Programming Language)发展而来。1963年剑桥大学在算法语言ALGOL60的基础上推出CPL(Combined Programming Language)。1967年剑桥大学的Martin Richards对CPL进行简化,推出BCPL语言。1970年美国贝尔实验室的Ken Thompson对BCPL进一步简化,取BCPL的第一个字母命名新推出的B语言。由于B语言过于简单,功能有限,1972年贝尔实验室的Dennis Ritchie对B语言进行丰富和完善,并以BCPL的第二个字母命名这种语言,即C语言。

1978年,Brian Kernighan和Dennis Ritchie发表了The C Programming Language,这成为标准C语言的基础。1983年,美国国家标准协会(American National Standards Institute, ANSI)开始着手制定C语言标准,称为ANSI C。经过6年努力,这个标准于1989年底被采纳,并于1990年启用。该标准同时被国际标准化组织(International Standards Organization, ISO)采纳,因此也被称为ANSI/ISO C。此后,在C的基础上又发展出C++和C#。

C语言主要有以下几方面的特点。

(1) C语言精简、灵活,为程序员提供了较大的自由度。C语言只有32个关键字,如表1.1所示,而BASIC语言的关键字则超过100个。C语言程序格式较为灵活,既可以在一行内写多条语句,也可以将一条较长的语句拆分后写在连续的若干行里。

表 1.1 C语言的关键字

auto	const	double	float	int	short	struct	unsigned
break	continue	else	for	long	signed	switch	void
case	default	enum	goto	register	sizeof	typedef	volatile
char	do	extern	if	return	static	union	while

(2) C语言提供了较为丰富的运算符,支持结构体、枚举等自定义数据类型,同时它也支持只有低级语言才能实现的对二进制位、字节和地址进行的操作。C语言既具有低级语言操控硬件的一部分功能,又具备高级语言简洁直观的表达形式。例如,增量操作在汇编语言中用INC指令实现,在C语言中相应地有++运算符。C语言不仅支持整型、实型和字符型等标准数据类型,还支持数组、结构体以及指针,可用来构造链表、栈和树等复杂的数据结构。

(3) C语言是结构化的程序设计语言,实现了程序与数据的分离,用C语言可以编写出通用性较强且符合软件工程规范的程序模块。

(4) C语言程序的目标代码质量较高。相比而言,C语言源程序经编译后产生目标代码的效率高于其他高级语言的编译结果,仅次于汇编语言源程序所产生的目标代码。

1.3 VC环境下的上机操作步骤

一般来说,一个C程序的开发过程主要包括以下四个基本步骤,各步骤之间的衔接关系如图1.4所示。

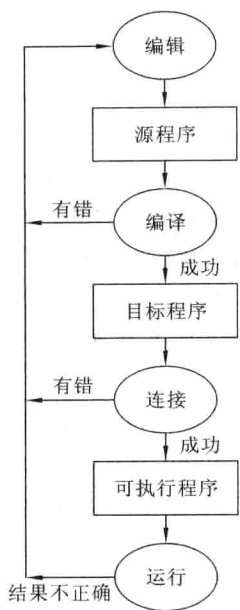


图 1.4 用 C 语言编程的操作步骤

第 1 步:编辑。通过任何一种文本编辑器(如 Windows 中的记事本)输入已经编写好的程序,存盘以后就得到一个源程序,通常源程序的扩展名是. C 或者. CPP。实用的 C 语言开发环境都自带编辑器,可以在其中录入、修改源程序。

第 2 步:编译。使用编译程序把源程序翻译成目标程序。如果源程序没有语法错误,则编译程序将翻译结果存盘,即目标程序。目标程序的文件主名与源文件相同,但扩展名为. OBJ,例如, MYPROG. CPP 经编译后产生的目标程序文件为 MYPROG. OBJ。如果源程序中有语法错误,则不会产生目标文件,编译程序会给出相应的错误提示。程序员必须分析、查找相应的错误,回到第 1 步,通过编辑器对源程序进行修改,存盘后再重新编译,如此反复直至编译成功。

第 3 步:连接。使用连接程序把目标程序和系统程序库进行连接装配。如果连接成功,连接程序会将连接结果存放到磁盘上,形成可执行文件。可执行文件的扩展名为. EXE。若连接失败,则不会产生可执行文件,连接程序会给出相应的错误的提示。程序员需要分析、查找产生错误的原因,更正错误以后,重新存盘、编译、连接。

第 4 步:运行。命令计算机执行第 3 步得到的可执行程序,查看输出结果。如果对各种可能出现的情况程序都能正确地工作,则程序可以正式交付使用。若在运行阶段发现程序运行结果不正确,则必须分析、查找产生问题的原因,回到第 1 步,修改源程序,然后重新编译、连接和运行,直至程序能正确工作为止。

目前常见的 C 语言版本主要有 Borland 公司的 Turbo C、Microsoft 公司的 VC++6.0、VC.NET 等。其中 VC++6.0 得到了广泛应用,用该版本开发的应用软件不胜枚举。VC++6.0 是一个集成开发环境,它为程序员提供了从编辑、编译、连接到运行四个阶段的简捷操作,为程序开发工作带来很大的方便。本书中所有的完整程序示例均在 VC++6.0 环境中测试通过。

VC++6.0 为程序员提供了编制各种类型软件的方式,本书仅介绍其中编写控制台程序的操作步骤。

第 1 步:新建控制台工程及源程序文件。

启动 VC++6.0(以下简称 VC)后,如图 1.5 所示,单击“文件”菜单栏,选择其中的“新建”菜单项,打开图 1.6 所示的“新建”对话框。

在图 1.6 所示的对话框中选择“工程”标签,单击“控制台工程(Win32 Console Application)”选项,然后在右侧设置新建该工程所在的文件夹,最后输入工程的名称。注意,如图 1.6 所示,设置工程所在文件夹时,只设置为 D:\VC_PROJ\,输入工程名称 MYPROJ 时,该名称会自动添加到所设置的文件夹后面。另外,新建工程所在的文件夹(例如,图 1.6 中设置的 D:\VC_PROJ\MYPROJ)不能是已经存在的文件夹。

完成上述设置后,单击“确定”按钮,可以见到如图 1.6 所示的对话框关闭,出现新的对话框,如图 1.7 所示。单击选择其中的“空工程”(An empty project)单选按钮,再单击“完成”按

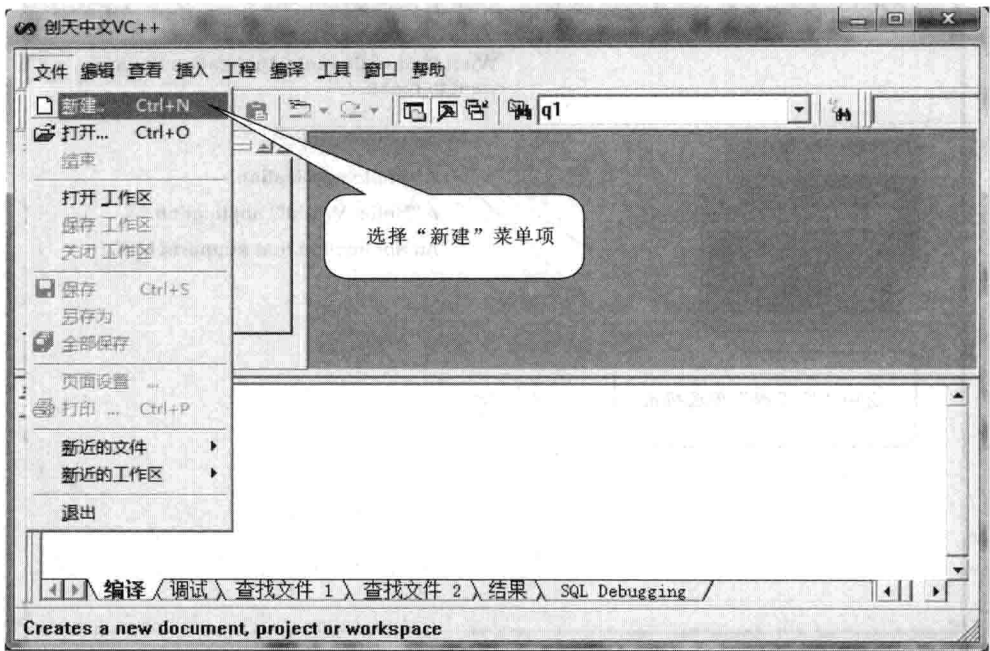


图 1.5 在 VC 集成环境中选择“新建”菜单项

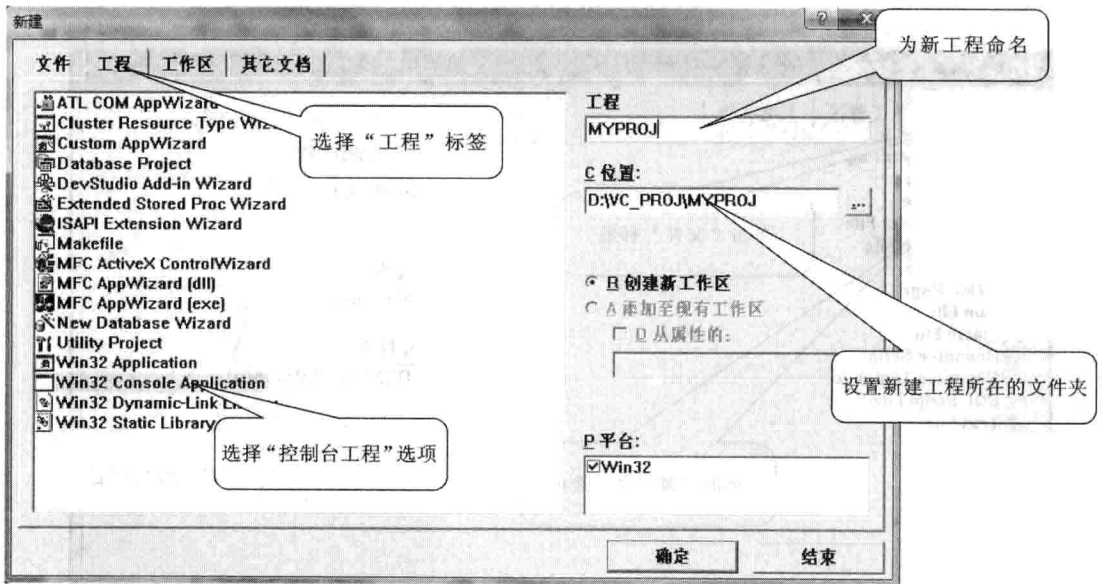


图 1.6 在 VC 集成环境中新建控制台工程

钮,并在接着出现的对新建工程有关信息进行确认的对话框中直接单击“确定”按钮。

再次单击“文件”菜单栏的“新建”菜单项,在图 1.8 所示的“新建”对话框中选择“文件”标签,单击下面的“C++源程序”(C++ Source File)选项,并输入源程序名称,系统将自动以 .CPP 作为源程序文件的扩展名。

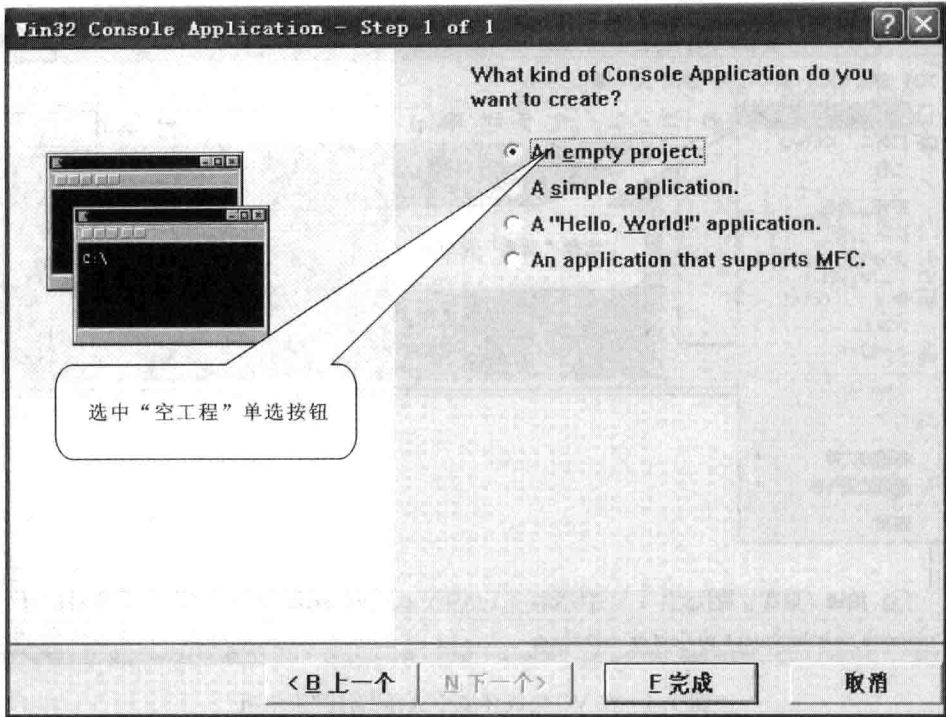


图 1.7 新建空的控制台工程

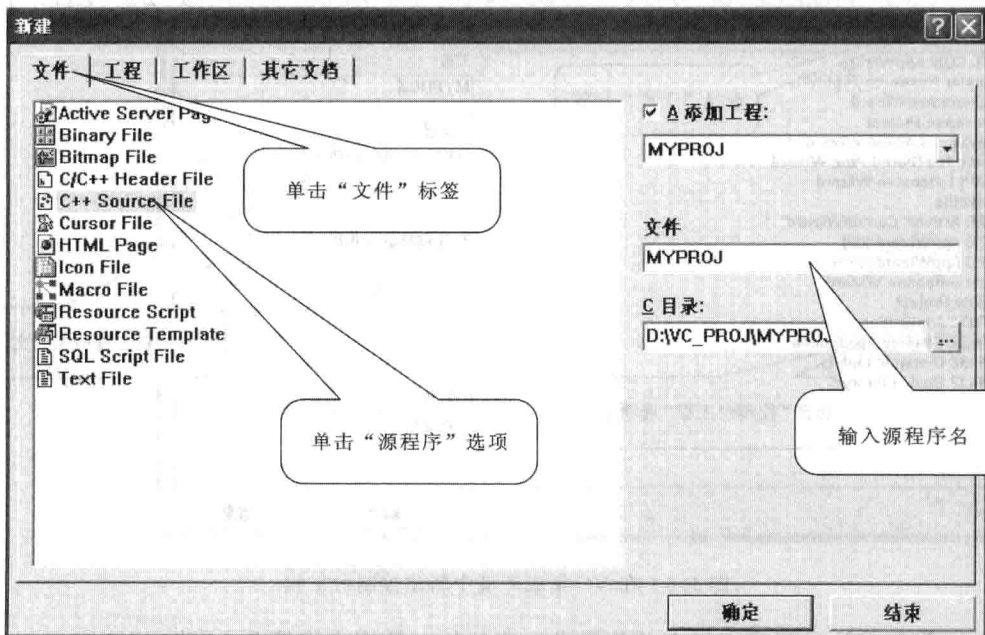


图 1.8 新建空的源程序文件

完成设置后单击“确定”按钮，可以见到图 1.9 所示的界面。此时磁盘上已经新建了相应的文件夹及若干文件，打开相应的文件夹，已建立的文件如图 1.10 所示，其中的 MYPROJ.

CPP 是源程序文件。

第 2 步:编辑源程序文件。

完成第 1 步的新建工作之后,在图 1.9 所示的编辑窗中录入程序代码,例如,录入下面一段最简单的程序代码:

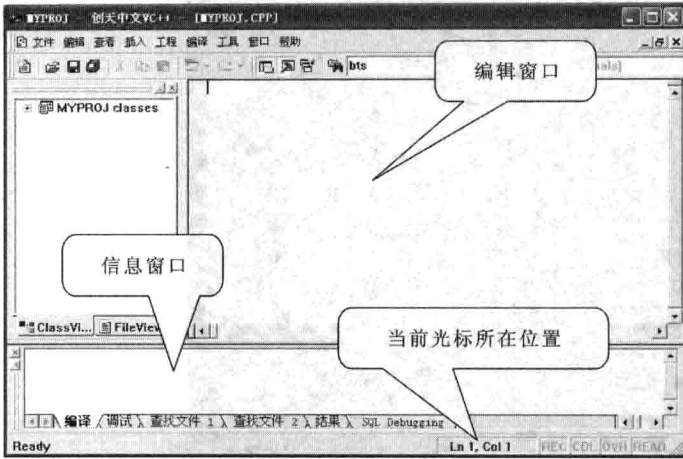


图 1.9 新建空工程及源程序文件后的 VC 集成环境界面

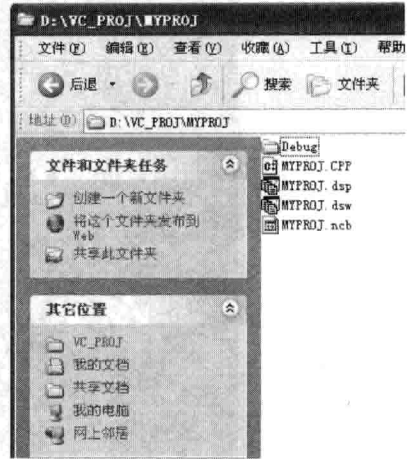


图 1.10 新建工程文件夹中的文件

```
#include "stdio.h"
void main()
{
    printf("This is my first C program.\n");
}
```

录入完成后,单击快捷按钮中的磁盘符号将源程序存盘。

第 3 步:编译、连接并运行。

虽然编译、连接、运行是几个独立的步骤,但 VC 集成环境为这几个步骤提供了快捷按钮,如图 1.11 所示。实际操作中通常是单击红色的“!”按钮计算机依次完成编译、连接、运行三项操作。如果编译或者连接过程中出现错误,在图 1.9 所示的“信息窗口”中会出现相应的错误提示信息,编程者可以根据提示信息对源程序进行修改,修改完成后重新存盘、编译、连接、运行。当编译成功时,磁盘上会存储一个扩展名为 .OBJ 的目标文件;连接成功后又会上存储一个扩展名为 .EXE 的可执行文件。在 VC 环境中,目标文件的文件主名与源程序主名相同,而可执行文件的文件主名与工程名相同。按照第 2 步的操作示例,编译产生的目标文件是 MYPROJ.OBJ,连接产生的可执行文件是 MYPROJ.EXE,这两个文件被存放在图 1.10 所示的 Debug 文件夹中。

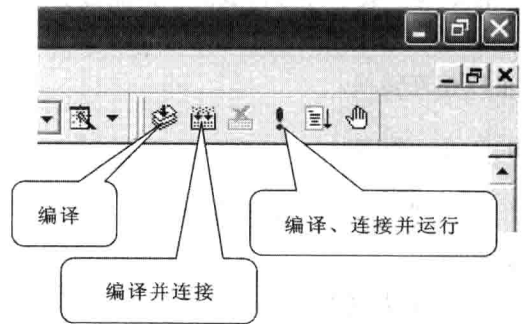


图 1.11 编译、连接、运行快捷按钮

在单击红色的“!”按钮之后,如果编译、连接都没有错误,紧接着程序将被运行,会出现一

个黑底白字的窗口,显示程序运行的情况。对于上述第 2 步录入的程序,该窗口中将显示如图 1.12 所示的界面。查看程序运行结果后,按照界面上的提示 Press any key to continue 按任意键关闭该窗口。

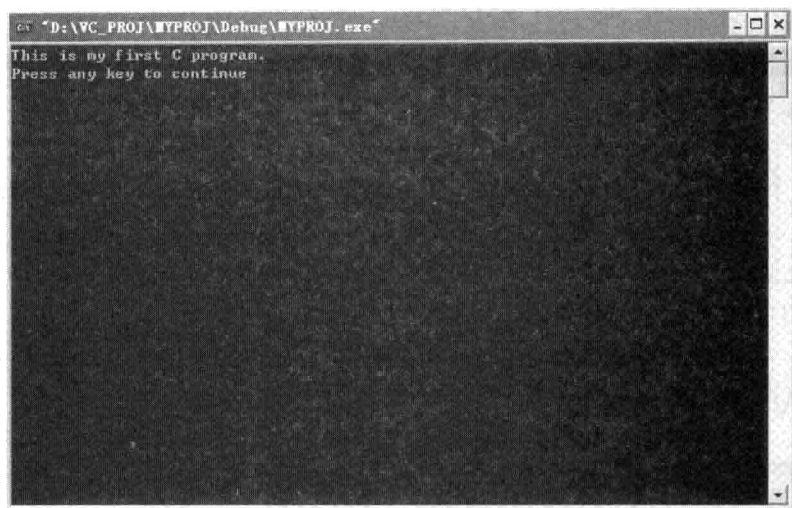


图 1.12 程序运行时的显示界面

小知识: VC 可以新建的对象很多,如果按本节的步骤操作时有误,一个简单的方法是关闭 VC 后重新操作,并参见本章的“实践建议”之问题 3。

1.4 数制及相互转换

人们在日常生活中每天都要与数打交道,通常都使用十进制数,而计算机内部使用的是由 0 和 1 两个数字符号构成的二进制数。二进制数在书写时过于冗长、容易出错,所以在计算机领域还经常使用八进制数和十六进制数。同一个数值可以用不同的数制表示,不同数制之间可以相互转换。

1.4.1 数制

描述一个数制需要说明三方面的问题:有哪些计数符号、每个数位代表的含义和基本的计数规则。

1. 计数符号

表示数据必然要使用各种符号,一种数制所有计数符号构成的集合称为该数制的数符集。 k 进制的数符集包含 k 个符号。例如:

二进制数符集中只有两个符号:0 和 1;

八进制数符集中有 8 个符号:0、1、2、3、4、5、6、7;

十进制数符集中有 10 个符号:0、1、2、3、4、5、6、7、8、9;

十六进制数符集中有 16 个符号:0、1、 \dots 、9、A、B、C、D、E、F。

各个计数符号是有顺序的,二进制、八进制和十进制的每个计数符号就是它的序号值,十

六进制的后 6 个计数符号的序号值依次是 10、11、12、13、14 和 15。

2. 基数和权

如果用 k 进制表示一个数 $a_n a_{n-1} \cdots a_1 a_0 . a_{-1} a_{-2} \cdots a_{-m}$, 则第 i 位 ($i = n, n-1, \cdots, 0, -1, -2, \cdots, -m$) 上的数符 a_i 所代表的含义是 $a_i \times k^i$ 。 k 称为这个数制的基数, k^i 称为 k 进制数第 i 位的权。个位、十位、百位……是十进制各个数位上的权特有的名称, 相应的权是 10^0 、 10^1 、 10^2 ……

3. 计数规则

简单地说, k 进制数的计数规则就是“逢 k 进 1, 借 1 当 k ”。

1.4.2 十进制数转换成其他进制数

短除法是十进制正整数转换成二进制数的常用方法。把十进制数连续除以 2, 直到商是 0 为止, 记录每次的商和余数, 没有余数则记余数为 0。以记录余数的相反次序依次写出各个余数, 结果就是相应的二进制数。

【例 1.1】 把十进制数 145 转换成二进制数。

2	1451	↑
2	720	
2	360	
2	180	
2	91	
2	40	
2	20	
2	11	
	0		

把余数按由下向上的次序写出, 可知与十进制数 145 等值的二进制数是 10010001。

短除法不仅用于十进制数到二进制数的转换, 也用于十进制数转换成其他进制数。十进制数转换成八进制数或者十六进制数的方法与转换成二进制数的方法相似, 只是除数分别是 8 和 16。在转换成十六进制数时还需要把大于 9 的余数写成相应的十六进制数符。

【例 1.2】 把十进制数 15785 转换成十六进制数。

16	157859	↑
16	986A	
16	61D	
16	33	
	0		

余数中, 10 对应十六进制数符 A, 13 对应 D。所以, 与十进制数 15785 等值的十六进制数是 3DA9。

把十进制纯小数转换成二进制数时, 人工计算可以采用连续乘以 2 的方法。每次把当前纯小数乘以 2, 分别记载整数部分和纯小数部分。如果当前纯小数部分是 0 则计算完成, 按得到整数部分的原有次序写出各个二进制位即可。但是, 如果纯小数部分始终不是 0 则会一直持续计算下去, 此时一般是指定小数点后取 n 位, 即作了 n 次乘法计算后终止。这种情况下, 转换得到的是小数点后取 n 位的近似值。