

21世纪高等学校计算机教育实用规划教材

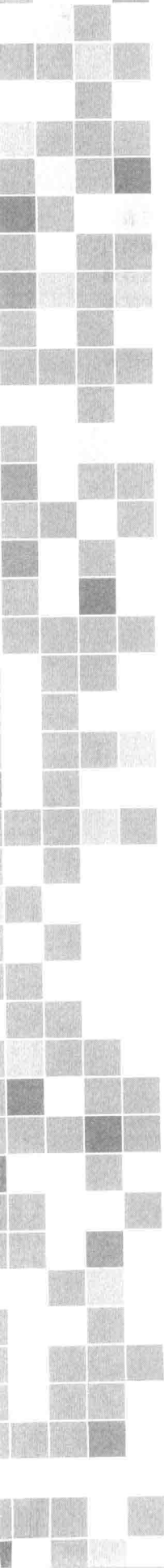
C语言程序设计

周 红 主编

李丽华 刘 英 胡滢滨 副主编



清华大学出版社



21世纪高等学校计算机教育实用规划教材

C语言程序设计

周 红 主编

李丽华 刘 英 胡滢滨 副主编

清华大学出版社
北京

内 容 简 介

本书共有 12 章,从开发环境、第一个 C 语言程序开始,介绍了 C 语言常用的数据类型、运算符与表达式、三种基本控制结构、数组、函数、指针、宏定义、结构体、共用体、枚举等自定义类型、动态分配及撤销内存、链表、位运算、文件等主要内容。在介绍内容的过程中加入了大量的生活实例和图示说明,使得内容通俗易懂,重点突出,难点讲透。重点讲解解题思路,贯穿“以程序编写带动语法教学”的模式,引导读者掌握 C 语言的核心编程方法,提高应用能力。

本书非常适合初次学习程序者作为入门教材,配套资料还提供了电子课件、习题答案及例题程序源代码。

本书封面贴有清华大学出版社防伪标签,无标签者不得销售。

版权所有,侵权必究。侵权举报电话:010-62782989 13701121933

图书在版编目(CIP)数据

C 语言程序设计/周红主编. —北京:清华大学出版社,2014

21 世纪高等学校计算机教育实用规划教材

ISBN 978-7-302-37457-2

I. ①C… II. ①周… 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 170682 号

责任编辑:付弘宇 李 晔

封面设计:常雪影

责任校对:梁 毅

责任印制:何 芊

出版发行:清华大学出版社

网 址: <http://www.tup.com.cn>, <http://www.wqbook.com>

地 址:北京清华大学学研大厦 A 座 邮 编:100084

社 总 机:010-62770175 邮 购:010-62786544

投稿与读者服务:010-62776969, c-service@tup.tsinghua.edu.cn

质 量 反 馈:010-62772015, zhiliang@tup.tsinghua.edu.cn

课 件 下 载: <http://www.tup.com.cn>, 010-62795954

印 刷 者:北京富博印刷有限公司

装 订 者:北京市密云县京文制本装订厂

经 销:全国新华书店

开 本:185mm×260mm 印 张:22.25 字 数:556 千字

版 次:2014 年 9 月第 1 版 印 次:2014 年 9 月第 1 次印刷

印 数:1~2500

定 价:39.50 元

产品编号:061214-01

出版说明

随着我国高等教育规模的扩大以及产业结构调整的进一步完善,社会对高层次应用型人才的需求将更加迫切。各地高校紧密结合地方经济建设发展需要,科学运用市场调节机制,合理调整和配置教育资源,在改革和改造传统学科专业的基础上,加强工程型和应用型学科专业建设,积极设置主要面向地方支柱产业、高新技术产业、服务业的工程型和应用型学科专业,积极为地方经济建设输送各类应用型人才。各高校加大了使用信息科学等现代科学技术提升、改造传统学科专业的力度,从而实现传统学科专业向工程型和应用型学科专业的发展与转变。在发挥传统学科专业师资力量强、办学经验丰富、教学资源充裕等优势的同时,不断更新教学内容、改革课程体系,使工程型和应用型学科专业教育与经济建设相适应。计算机课程教学在从传统学科向工程型和应用型学科转变中起着至关重要的作用,工程型和应用型学科专业中的计算机课程设置、内容体系和教学手段及方法等也具有不同于传统学科的鲜明特点。

为了配合高校工程型和应用型学科专业的建设和发展,急需出版一批内容新、体系新、方法新、手段新的高水平计算机课程教材。目前,工程型和应用型学科专业计算机课程教材的建设工作仍滞后于教学改革的实践,如现有的计算机教材中有不少内容陈旧(依然用传统专业计算机教材代替工程型和应用型学科专业教材),重理论、轻实践,不能满足新的教学计划、课程设置的需要;一些课程的教材可供选择的品种太少;一些基础课的教材虽然品种较多,但低水平重复严重;有些教材内容庞杂,书越编越厚;专业课教材、教学辅助教材及教学参考书短缺,等等,都不利于学生能力的提高和素质的培养。为此,在教育部相关教学指导委员会专家的指导和建议下,清华大学出版社组织出版本系列教材,以满足工程型和应用型学科专业计算机课程教学的需要。本系列教材在规划过程中体现了如下一些基本原则和特点。

(1) 面向工程型与应用型学科专业,强调计算机在各专业中的应用。教材内容坚持基本理论适度,反映基本理论和原理的综合应用,强调实践和应用环节。

(2) 反映教学需要,促进教学发展。教材规划以新的工程型和应用型专业目录为依据。教材要适应多样化的教学需要,正确把握教学内容和课程体系的改革方向,在选择教材内容和编写体系时注意体现素质教育、创新能力与实践能力的培养,为学生知识、能力、素质协调发展创造条件。

(3) 实施精品战略,突出重点,保证质量。规划教材建设仍然把重点放在公共基础课和专业基础课的教材建设上;特别注意选择并安排一部分原来基础比较好的优秀教材或讲义修订再版,逐步形成精品教材;提倡并鼓励编写体现工程型和应用型专业教学内容和课程体系改革成果的教材。

(4) 主张一纲多本,合理配套。基础课和专业基础课教材要配套,同一门课程可以有多本具有不同内容特点的教材。处理好教材统一性与多样化,基本教材与辅助教材,教学参考书,文字教材与软件教材的关系,实现教材系列资源配套。

(5) 依靠专家,择优选用。在制订教材规划时要依靠各课程专家在调查研究本课程教材建设现状的基础上提出规划选题。在落实主编人选时,要引入竞争机制,通过申报、评审确定主编。书稿完成后要认真实行审稿程序,确保出书质量。

繁荣教材出版事业,提高教材质量的关键是教师。建立一支高水平的以老带新的教材编写队伍才能保证教材的编写质量和建设力度,希望有志于教材建设的教师能够加入到我们的编写队伍中来。

21 世纪高等学校计算机教育实用规划教材编委会

联系人:魏江江 weijj@tup.tsinghua.edu.cn

前 言

C 语言历史悠久,博大精深。从 C 语言产生到现在,它已经成为最重要和最流行的编程语言之一。它是入门级语言,在各种流行编程语言中,都能看到 C 语言的影子。它既具有高级语言的强大功能,又有很多用来直接操作计算机硬件的功能。学习和掌握 C 语言这门实用性强的技术,既增进了对于计算机底层工作机制的了解,增加就业机会,又相当于获取了开启程序设计大门的钥匙,为进一步学习 C++、Java、C# 等其他高级语言打下了坚实的基础。因此,C 语言程序设计课程的地位举足轻重。

经调查研究发现,大部分初学者在学习这门课程时会遇到一些问题。例如,有些教材没有讲清楚基础概念,习惯使用专业术语来解释专业术语,导致学生无法理解,因此编程也就无从下手;有些教材没有循序渐进地培养学生建立编程思路,因此无法取得学以致用效果;有些教材在讲授程序的过程中,没有注重利用生活实例、其他学科的知识辅助教学,没有形成系统的教学过程,使得学生感到内容枯燥烦琐,学过后印象不深刻。本书正是针对这些问题而编写的一本适合初学者的 C 语言基础教材。略有计算机基础的人,都可利用本书学会 C 语言编程。与其他教材相比,本书有如下几个方面的特点。

1. 按照学习逻辑安排章节顺序

有些 C 语言教材会存在前面章节的内容需要用到后面章节知识的情况。如果在前面章节中没有必要的解释,学生就会产生疑问,类似这样的疑问多了,学习起来就会感到毫无头绪。例如,许多教材的第一个程序都会用到“`#include<stdio.h>`”这个命令,要理解这个命令的含义,涉及宏定义中文件包含的知识。很多教材对此没有做详细的说明,导致学生写了很多程序过后,仍然不理解为什么几乎每个程序前面都要写这个命令,这样的学习效果可想而知。

本书根据人类学习新知识时的思维过程设置内容的顺序。在讲解学生第一次接触的新知识时,都做出必要的交代。若在讲解前面章节时需要用到后面章节的知识,就将这部分知识提到前面来细讲,将最初的程序作为“树的根基”,根据完善“树”的需要来不断扩充知识点作为它的枝干和树叶。

2. 对比理解内容

本书注重将其他学科的知识为我所用、触类旁通。用学生熟悉的英语来对比 C 语言,拉近初学者对 C 语言的认识距离。例如,教材中的算法、数据类型、运算符等术语对于学生来说都非常陌生,可以利用英语对比理解它们。例如,假如说 C 语言是英语,C 语言程序就

是英语文章；算法就是写英语文章的思路；数据类型就是单词的类型，像动词、名词等；运算符和表达式就是短语；顺序、选择、循环结构就是语法结构等。通过一系列的对比，就会让学生对这门课的内容有基本的了解。

3. 提炼重点和难点

本书在力求知识点涵盖面广的前提下，对重点和难点知识（例如，函数、数组、指针等）以“少而精”的原则进行细化、提炼，用大量的笔墨对这些内容进行详细的解剖式分析，将内容讲解透彻。

4. 概念形象化

有些 C 语言概念比较抽象，如果用专业术语去讲解专业术语，不容易理解。本教材大量运用生活实例细致地解释抽象的概念，将 C 语言形象化、直观化。例如，很多学生觉得指针的概念较难理解，书写程序时能避则避。针对这个问题，本书将变量比喻为空调，将指针比喻为空调的遥控器。可以将遥控器指向空调，按下遥控器的按钮来间接控制空调。同样的道理，在程序中也可以通过指向变量的指针来间接控制变量。这样帮助学生建立对指针建立起直观的认识后，学生才能对指针运用自如。

本书在介绍内容的过程中加入了大量的实例和图示说明，使得内容通俗易懂，言简意赅，重点突出，难点讲透。本书共包含 12 章的内容。俗话说：“千里之行，始于足下”。要想学好这门语言，首先要扎实地掌握这门语言的基础知识。因此，第 1 章详细介绍了开发环境的使用和第一个 C 语言程序的每个细节。第 2 章介绍了 C 语言的语法重点：数据类型、运算符和表达式。第 3 章详细说明了三种基本控制结构及辅助控制结构。第 4 章介绍了数组的知识。第 5 章介绍了各种函数和带参数的宏。第 6 章重点介绍了指针、指针与数组、指针与函数。第 7 章说明了结构体、共用体、枚举类型。第 8 章介绍了动态分配内存、动态撤销内存。第 9 章介绍了链表的知识。第 10 章介绍了位运算的相关知识。第 11 章介绍了文件的操作。第 12 章列举了一些程序中的常见错误。每个章节中的重点和难点章节都在章节前面加了◇标注，每个章节后面都配有丰富的课后练习。

本书重点讲解解题思路，贯穿“以程序编写带动语法教学”的模式，引导读者掌握 C 语言的核心编程方法，提高应用能力。本书实例都通过 Visual C++ 6.0 调试，读者可以在 Windows 环境下使用本书学习 C 语言。本书配有完善的辅助资料，包含教材中所有例题的程序、课后习题的参考答案、制作精美的教学课件，读者（尤其是教师）可以在清华大学出版社的网站(<http://www.tup.tsinghua.edu.cn>)进行下载使用。本书非常适合用作高等院校及高职高专的教材，也适合初次学习程序者作为入门的教材使用。

本书由周红老师整体架构。第 1、5、6、11、12 章由周红老师编写，第 2、4、8 章由李丽华老师编写、第 3、10 由刘英老师编写，第 7、9 章由胡滢滨老师编写。陶德元教书对本书的编写和出版进行了指导，提出了许多建设性的建议。清华大学出版社的工作人员为本书的编写和出版也付出了辛勤劳动。在本书完成之际，一并向他们表示诚挚的感谢。

在本书的编写过程中，作者查阅、借鉴了许多参考资料，在此谨向诸多学者、同仁表示由衷的敬意和感谢。尽管在写作过程中作者投入了大量的时间和精力，但由于时间及水平有

限,错误和不足之处仍在所难免,敬请读者批评指正,我们会积极对本书进行修订和补充(E-mail: zh_scujcc@sina.com)。

本书的配套课件和习题答案可以从清华大学出版社网站 www.tup.com.cn 下载,在本书和课件的使用中遇到任何问题,请联系 fuhy@tup.tsinghua.edu.cn。

编 者

2014 年 6 月



第 1 章 C 语言概述	1
1.1 什么是 C 语言	1
1.2 C 语言的发展史	2
1.3 Visual C++ 6.0 集成环境	3
1.4 第一个 C 语言程序	15
1.5 书写程序的规则	17
1.6 习题	19
第 2 章 数据类型、运算符与表达式	20
2.1 C 语言的数据类型	20
2.2 常量和变量	21
2.2.1 标识符	21
2.2.2 常量	22
2.2.3 变量	23
2.2.4 整型数据	27
2.2.5 实型数据	30
2.2.6 字符型数据	32
2.2.7 字符串常量	35
◇ 2.2.8 不同数据类型间的转换	36
2.3 运算符和表达式	36
2.3.1 算术运算	38
2.3.2 赋值运算	39
2.3.3 类型转换运算	43
◇ 2.3.4 自增、自减运算	44
2.3.5 关系运算	45
2.3.6 逻辑运算	46
2.3.7 条件运算	48
2.3.8 逗号运算	49
2.4 习题	49

第3章 三种基本控制结构	51
3.1 算法	51
3.1.1 算法的概念	51
3.1.2 算法的表示方法	52
3.2 C 语句	54
3.3 数据输入与输出	55
3.3.1 格式输入输出函数	55
3.3.2 字符输入输出函数	65
3.4 结构化的程序设计方法	66
3.5 顺序结构程序设计	66
3.6 选择结构程序设计	68
3.6.1 单分支选择结构	68
3.6.2 多分支选择结构	72
◇ 3.6.3 嵌套选择结构	78
3.7 循环结构程序设计	84
3.7.1 while 结构	84
3.7.2 do while 结构	86
◇ 3.7.3 for 结构	90
◇ 3.7.4 循环嵌套结构	93
3.7.5 有关循环语句的建议	97
3.8 break 和 continue 辅助控制语句	98
◇ 3.8.1 break 语句	98
3.8.2 continue 语句	101
3.9 习题	102
第4章 数组	104
4.1 一维数组	105
4.1.1 一维数组的定义	105
4.1.2 一维数组元素的引用	106
4.1.3 一维数组的初始化	109
4.2 二维数组	120
4.2.1 二维数组的定义	120
4.2.2 二维数组元素的引用	121
4.2.3 二维数组的初始化	122
4.3 字符数组	126
4.3.1 字符串	126
4.3.2 字符数组的输入输出	127
◇ 4.3.3 字符串处理函数	132

4.4 习题	142
第5章 函数	143
5.1 函数的定义	144
5.2 函数的调用	146
5.2.1 函数调用的格式	146
5.2.2 使用函数调用的形式	147
5.3 函数的声明	150
5.3.1 函数声明的意义	150
5.3.2 函数声明的格式	150
◇ 5.4 函数参数的传值	151
5.4.1 单向值传递	151
5.4.2 双向地址传递	153
◇ 5.5 嵌套调用和递归调用	157
5.5.1 嵌套调用	157
5.5.2 递归调用	158
5.6 变量的使用范围和存储类别	161
5.6.1 局部变量与全局变量	161
5.6.2 变量的存储类别	164
5.7 多文件程序	169
5.7.1 C源文件的划分	169
5.7.2 C源文件的格式	169
5.8 用带参数的宏代替公式型函数	174
5.9 习题	177
第6章 指针	179
6.1 指针的概念	179
6.1.1 计算机中的内存	179
6.1.2 指针变量就是存储地址的变量	180
6.2 指向变量的指针	180
6.2.1 指针变量的定义	180
6.2.2 指针变量的初始化和赋值	180
6.2.3 使用指针变量	181
6.3 指向数组的指针	183
6.3.1 指针和数组名	183
6.3.2 指针的赋值运算	184
6.3.3 指针的算术和关系运算	184
◇ 6.3.4 利用指针引用数组元素	188
◇ 6.3.5 指针作函数的参数	190

6.4 指向字符串的指针	203
6.4.1 字符指针和字符数组	203
◇6.4.2 利用字符指针处理字符串	207
◇6.5 指针数组和指向指针的指针	212
6.5.1 指针数组	212
6.5.2 指向指针的指针	214
6.5.3 数组指针	216
6.6 函数指针与指针函数	217
6.6.1 函数指针	217
6.6.2 指针函数	219
6.7 习题	221
第7章 构造数据类型	223
7.1 概述	223
7.2 结构体	223
7.2.1 结构体类型的定义	224
7.2.2 结构体类型变量的定义及初始化	225
7.2.3 引用结构体类型变量的成员	231
◇7.2.4 结构体数组	235
7.2.5 指向结构体类型数据的指针	237
7.2.6 结构体在函数间的传递	241
7.3 共用体	244
7.3.1 共用体类型及变量的定义	244
7.3.2 共用体变量的引用	247
7.4 枚举类型	249
7.4.1 枚举类型及变量的定义	250
7.4.2 枚举类型的引用	251
7.5 用 typedef 定义类型	254
7.6 习题	255
第8章 动态分配及撤销内存	256
8.1 动态分配与撤销内存的概念	256
◇8.2 malloc 与 free 函数	257
8.3 calloc 与 free 函数	259
8.4 free 函数与指针	261
8.5 习题	263

第 9 章 链表	264
9.1 链表的结构	264
9.2 链表的创建及遍历输出	265
◇9.3 链表的插入	269
◇9.4 链表的删除	276
9.5 链表的销毁	284
9.6 循环链表	287
9.7 双链表	287
9.8 栈和队列	287
9.9 习题	287
第 10 章 位运算	289
10.1 位运算的概念	289
10.2 位运算符与位运算	290
10.2.1 “按位与”运算符(&)	290
10.2.2 “按位或”运算符()	291
10.2.3 “按位异或”运算符(^)	292
10.2.4 “取反”运算符(~)	294
◇10.2.5 “左移”运算符(<<)	294
◇10.2.6 “右移”运算符(>>)	295
10.3 位运算举例	297
10.4 位段	301
10.4.1 位段的概念	301
10.4.2 位段的声明和位段变量的定义	301
10.4.3 位段的使用	303
10.5 习题	304
第 11 章 文件	305
11.1 文件的概念	305
11.2 文件类型指针	309
11.3 文件的打开与关闭	309
11.3.1 文件的打开	309
11.3.2 文件的关闭	310
◇11.4 文件的读写操作	311
11.4.1 文件读函数	311
11.4.2 文件写函数	314

11.5	文件检测函数	318
11.6	文件的定位	319
11.7	习题	321
第 12 章	C 程序的常见错误	322
附录 A	ASCII 码表	331
附录 B	C 语言常用库函数	335
参考文献		341

本章重点：

- ◆ C 语言的基本程序结构。
- ◆ C 语言程序的开发过程。

本章难点：

- ◆ C 语言的编程环境使用。

本章要求：

- ◆ 了解 C 语言的基本作用、发展史及其语言特点。
- ◆ 理解 C 语言第一个程序中每个细节的意义，熟记 C 语言程序的基本格式。
- ◆ 掌握 C 语言程序的开发过程：编辑、编译、连接、运行。

1.1 什么是 C 语言

C 语言是一种能够让人类跟计算机交流的语言。如果用户命令计算机做一件事情，直接使用人类的自然语言跟它交流，计算机是听不懂的。这是因为计算机只能直接识别和处理二进制码，例如“01110110”，而人类又很难了解这些二进制码的含义。因此，我们需要一种介于人类的自然语言和二进制码之间的语言。这种语言既能让人类看懂，又能被翻译成二进制码，供计算机识别。这种连接人类和计算机的中间语言，就是程序设计语言。其作用示意图如图 1-1 所示。C 语言是一种在国际上通用的程序设计语言，也是早期最经典、最强大的面向过程的程序设计语言之一。它既可用于写系统软件（例如，Linux 操作系统等），也可用来写应用软件（例如，媒体播放器等）。目前国内外许多高校都将其作为程序设计的入门课程，它是开启程序设计大门的钥匙，是通往 C++、Java、C# 等其他语言世界的桥梁。

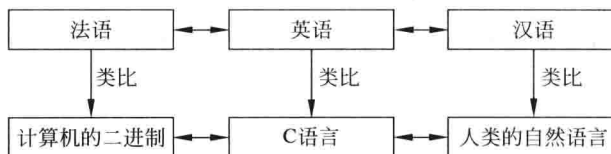


图 1-1 C 语言的作用

1.2 C 语言的发展史

早期的操作系统,例如 UNIX 操作系统主要是用汇编语言编写的。由于汇编语言依赖于计算机硬件,程序的可读性和可移植性都比较差。为了提高可读性和可移植性,考虑改用高级语言。但汇编语言可以直接对硬件进行操作,例如,对内存地址的操作、位(bit)操作等,而一般高级语言难以实现汇编语言的这些功能。因此人们期望能够找到一种既具有一般高级语言特性,又具有低级语言特性,集它们的优点于一身的中间语言。C 语言就在这种情况下应运而生了,之后成为了国际上广泛流行的计算机高级语言。

早期 C 语言是在 B 语言的基础上发展起来的,它的根源可以追溯到 ALGOL 60。1960 年出现的 ALGOL 60 是一种面向问题的高级语言,它离硬件比较远,不宜用来编写系统程序,1963 年英国的剑桥大学推出了 CPL(Combined Programming Language)语言。CPL 语言在 ALGOL 60 的基础上接近硬件一些,但规模比较大,难以实现。1967 年英国剑桥大学的 Martin Richards 对 CPL 语言做了简化,推出了 BCPL(Basic Combined Programming Language)语言。1970 年,美国贝尔实验室的 Ken Thompson 以 BCPL 语言为基础,又作了进一步简化,它使得 BCPL 能挤压在 8KB 内存中运行,这个很简单的而且很接近硬件的语言就是 B 语言(取 BCPL 的第一个字母),它在 DEC PDP-7 上实现。1971 年在 PDP-11/20 上实现了 B 语言,Thompson 用它写了第一个 UNIX 操作系统。但 B 语言过于简单,功能有限,并且和 BCPL 都是“无类型”的语言。

1972 年至 1973 年间,贝尔实验室的 D. M. Ritchie 在 B 语言的基础上设计出了 C 语言(取 BCPL 的第二个字母)。C 语言既保持了 BCPL 和 B 语言的优点(精练,接近硬件),又克服了它们的缺点(过于简单,数据无类型等)。最初的 C 语言只是为描述和实现 UNIX 操作系统提供一种工具语言而设计的。1973 年,K. Thompson 和 D. M. Ritchie 两人合作把 UNIX 的 90%以上用 C 改写,即 UNIX 第 5 版。这样,UNIX 使分散的计算系统之间的大规模联网以及互联网成为可能。

后来,C 语言多次做了改进,但主要还是在贝尔实验室内部使用。直到 1975 年 UNIX 第 6 版公布后,C 语言的突出优点才引起人们普遍注意。1977 年出现了不依赖于具体机器的 C 语言编译文本《可移植 C 语言编译程序》,使 C 移植到其他机器时所需做的工作大大简化了,这也推动了 UNIX 操作系统迅速地在各种机器上实现。例如,VAX、AT&T 等计算机系统都相继开发了 UNIX。随着 UNIX 的日益广泛使用,C 语言也迅速得到推广。1978 年以后,C 语言已先后移植到大、中、小、微型机上,如 IBM System/370、Honeywell 6000 和 Interdata 8/32,已独立于 UNIX 和 PDP。

1983 年,美国国家标准协会(ANSI)委任 X3J11 委员会对 C 语言进行标准化。该委员会的工作于 1989 年 12 月 14 日正式被批准为 ANS X3.159—1989,并于 1990 年春天颁布。ANSI C 增加了一些来自 C++的内容,并支持多国字符集。同时还规定了 C 运行库函数的标准。一年后,该标准被 ISO(国际标准化组织)接受,这就是 C89 标准。后来经过多次技术勘误,于 1999 年产生了新的 C99 标准。C99 标准保持了几乎所有 C89 的特征,并添加了一些库函数和其他特殊应用。

现在 C 语言已风靡全世界,成为世界上应用最广泛的几种计算机语言之一。它适合作为系统描述语言,既可用于写系统软件,也可用于写应用软件。

1.3 Visual C++ 6.0 集成环境

下面介绍利用 Visual C++ 6.0 集成环境开发 C 程序的过程。

1. 启动 Visual C++ 6.0 环境

选择“开始”→“所有程序”→Microsoft Visual Studio 6.0→Microsoft Visual C++ 6.0 命令,启动 Visual C++。Microsoft Visual C++ 6.0 的主窗口如图 1-2 所示。

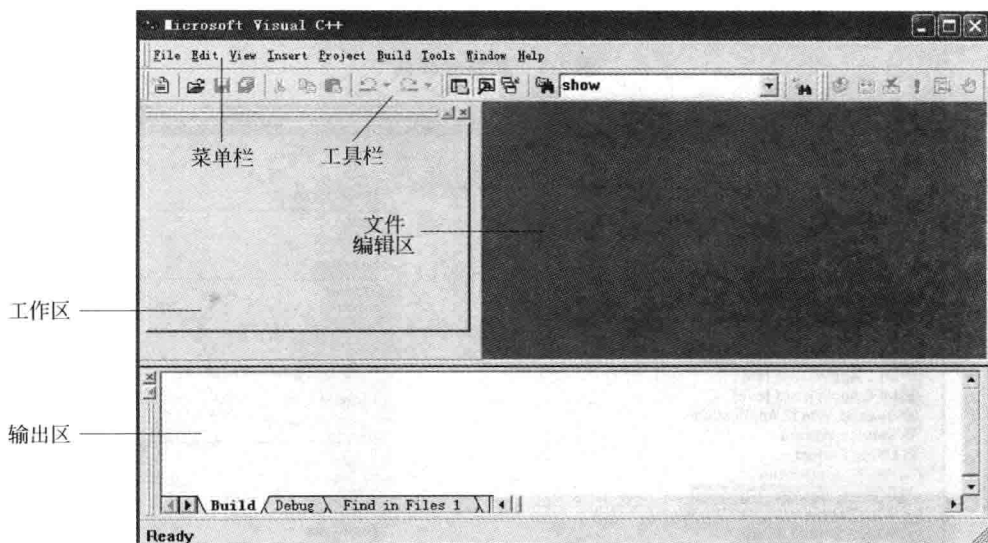


图 1-2 Microsoft Visual C++ 主窗口

在 Visual C++ 6.0 的主窗口中,最上面的区域是菜单栏和工具栏。菜单栏中有文件、编辑等常用选项,工具栏有保存、打印等图标工具,这些都可供用户使用。左边灰色空白部分是项目工作区。当创建或打开工作区后,可以在此处显示工作区中的文件结构或类结构。右边是文件编辑区,打开文件后,在文件编辑区可以编写文件内容。下面是程序输出区,编译、运行程序后,程序的错误提示、运行结果等内容都在此处显示。

2. 编辑源程序文件

1) 建立新工程(Project)

(1) 选择 File→New 命令,弹出 New 对话框(如图 1-3 所示)。

(2) 单击 Projects 选项卡,选择 Win32 Console Application 选项,在 Project name 文本框中输入工程名称,在 Location 文本框中输入或选择新工程所在的硬盘位置,例如,D:\test。单击 OK 按钮,如图 1-4 所示。弹出 Win32 Console Application-Step 1 of 1 对话框,如图 1-5 所示。

(3) 单击 Finish 按钮。系统显示 New Project Information 对话框,如图 1-6 所示。单击 OK 按钮转到主窗口,如图 1-7 所示。