

21世纪高校计算机规划教材

C语言程序设计

张菁 于晓强 主编

中国铁道出版社
CHINA RAILWAY PUBLISHING HOUSE

21 世纪高校计算机规划教材

C 语言程序设计

张 菁 于晓强 主 编
吕显强 赵秀岩 副主编
王 颖 张 鑫 参 编
王其华 刘 威



中国铁道出版社
CHINA RAILWAY PUBLISHING HOUSE

内 容 简 介

C 语言作为计算机程序设计入门语言，可以为学生思维能力的培养打下坚实的基础。

本书以培养学生程序设计能力为目标，以程序设计为主线，以 C 语言的相关知识为基础，以应用为驱动，通过案例和问题引入内容，重点讲解程序设计的思想和方法。全书共 10 章，包括 C 语言概述、数据类型与基本操作、逻辑运算和判断选取控制、循环控制、数组、函数、编译预处理、指针、结构体和共用体以及位运算和文件等方面的内容。

本书是学习 C 语言程序设计的初步教材，适合作为计算机专业教材，也可作为对 C 语言学习兴趣的读者的参考用书。

图书在版编目 (CIP) 数据

C 语言程序设计 / 张菁，于晓强主编. --北京：中国铁道出版社，2010.7(2019.1重印)

21 世纪高校计算机规划教材

ISBN 978-7-113-11480-0

I. ①C… II. ①张… ②于… III. ①
C 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 (2010) 第 144572 号

书 名：C 语言程序设计

作 者：张 菁 于晓强 主编

策划编辑：严晓舟 李志国

责任编辑：贾 星

编辑助理：张 丹

封面设计：付 巍

责任印制：郭向伟

读者热线电话：(010) 63550836

封面制作：白 雪

出版发行：中国铁道出版社（北京市西城区右安门西街 8 号 邮政编码：100054）

印 刷：北京铭成印刷有限公司

版 次：2010 年 8 月第 1 版 2019 年 1 月第 11 次印刷

开 本：787mm×1092mm 1/16 印张：18 字数：442 千

书 号：ISBN 978-7-113-11480-0

定 价：31.00 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，请与本社计算机图书批销部联系调换。

前 言

“计算机程序设计”是我国高等学校重要的计算机基础课程之一，以某种编程语言为平台，介绍计算机程序设计的思想和方法。其目标是要让学生掌握高级程序设计语言的基础知识，并在实践中逐步掌握计算机程序设计的思想和方法，培养学生分析问题和解决问题能力，以及提高学生计算机应用能力。因此，“计算机程序设计”是一门以培养学生程序设计基本方法和技能为目标，以提高学生实践能力为要旨的课程。

在我国高等院校计算机基础教育改革课题研究组的组织下，于2008年发表课题报告CFC2008，提出了复合型人才对于计算机应用能力和知识结构的要求，明确了课程体系与能力培养的关系。本书就是依据应用能力培养的要求，按照这一核心，结合学习特点，组织教师精心安排章节内容、编写思路、设计例题习题。编写过程中，力求先易后难、逐步深入，以保证能够让学生逐步理解、掌握相关章节要求，学习、理解并初步掌握基本的程序设计方法。

全书共分10章。第1章C语言概述，介绍了C语言基本运行环境和C语言的基础知识；第2章数据与基本操作，介绍了数据类型、运算符和表达式、基本输入/输出操作；第3章逻辑类型和判断选取控制，介绍了逻辑运算和判断选取控制；第4章循环控制，介绍了循环控制的基础和使用；第5章数组，介绍了数组的概念和使用方法；第6章函数，介绍了函数的使用、变量的存储类型；第7章编译预处理，介绍了宏定义、文件包含和条件编译；第8章指针，介绍了指针的概念和使用；第9章结构体和共用体，介绍了结构体的使用以及共用体、链表、枚举类型的概念；第10章位运算和文件，介绍了位运算和文件的基础。

本书由大连海洋大学和大连工业大学老师共同编写，由张菁和于晓强负责组织编写并统稿，由张菁、于晓强、吕显强、赵秀岩、王颖、张鑫、王其华、刘威共同参与编写。由于编者水平有限，疏漏和不妥之处在所难免，殷切希望广大读者给予批评指正。

编 者

2010年6月

目 录

第 1 章 C 语言概述	1
1.1 C 语言简介	1
1.1.1 什么是程序设计语言	1
1.1.2 C 语言的产生、发展与应用	1
1.1.3 C 语言的特点	2
1.2 程序设计初步	3
1.2.1 程序设计的一般步骤	3
1.2.2 编写第一个 C 语言程序	5
1.3 在 Visual C++ 环境中编程	6
1.3.1 Visual C++ 集成开发环境介绍	6
1.3.2 Visual C++ 环境中运行 C 语言源程序的步骤	8
1.4 在 Turbo C 环境中编程	11
1.4.1 Turbo C 集成开发环境介绍	11
1.4.2 Turbo C 环境中运行 C 语言源程序的步骤	14
1.5 学习 C 语言的几点建议	16
习题一	18
第 2 章 数据类型与基本操作	19
2.1 数据在计算机中的表示方法	19
2.1.1 数值的定点表示与浮点表示	19
2.1.2 字符的表示和存储方式	21
2.2 常量与变量	22
2.2.1 直接常量与符号常量	22
2.2.2 变量	23
2.3 基本数据类型	24
2.3.1 整型数据	25
2.3.2 实型数据	28
2.3.3 字符型数据	29
2.4 运算符与表达式	31
2.4.1 算术运算符	32
2.4.2 赋值运算符	34
2.4.3 关系运算符	35
2.4.4 逻辑运算符	35
2.4.5 条件运算符	36

2.4.6 强制类型转换运算符	36
2.5 数据的输入/输出	36
2.5.1 格式输出函数	37
2.5.2 格式输入函数	39
2.5.3 字符输入/输出函数	41
习题二	42
第 3 章 逻辑运算和判断选取控制	45
3.1 算法	45
3.1.1 算法的概述	45
3.1.2 描述算法的方法	46
3.2 结构化程序设计方法	50
3.3 C 语句的分类	52
3.4 逻辑判断运算	54
3.4.1 逻辑判断及 C 语言的逻辑值	54
3.4.2 关系运算	55
3.4.3 逻辑运算	56
3.5 if 语句	58
3.5.1 单分支 if 语句	58
3.5.2 双分支 if 语句	59
3.5.3 条件运算符	61
3.5.4 多分支 if 语句	63
3.5.5 if 语句的嵌套	66
3.6 switch 语句	68
习题三	70
第 4 章 循环控制	73
4.1 概述	73
4.2 goto 语句构成循环	73
4.3 while 语句	74
4.4 do...while 语句	76
4.5 for 语句	78
4.6 循环的嵌套	81
4.7 break 和 continue 语句	84
4.7.1 break 语句	84
4.7.2 continue 语句	85
4.8 程序举例	86
习题四	89
第 5 章 数组	93
5.1 一维数组	93

5.1.1	一维数组的定义与引用	94
5.1.2	一维数组的初始化	96
5.1.3	一维数组程序举例	98
5.2	二维数组	100
5.2.1	二维数组的定义和引用	100
5.2.2	二维数组的初始化	102
5.2.3	二维数组程序举例	103
5.3	字符数组与字符串	105
5.3.1	字符串的概念	105
5.3.2	字符数组的定义和初始化	105
5.3.3	字符数组的输入/输出	107
5.3.4	字符串处理函数	109
5.3.5	字符数组程序举例	111
	习题五	112
第 6 章	函数	113
6.1	函数的定义	113
6.1.1	无参函数的定义格式	113
6.1.2	有参函数的定义格式	114
6.1.3	空函数	114
6.2	函数的运用	115
6.2.1	函数的参数和返回值	115
6.2.2	函数的调用	118
6.2.3	数组参数	120
6.3	变量的作用域与生存期	124
6.3.1	局部变量	124
6.3.2	全局变量	124
6.3.3	变量的生存期	126
6.3.4	用 extern 声明外部变量	129
6.3.5	用 static 声明外部变量	130
6.3.6	函数的存储类别	131
6.4	函数的嵌套与递归调用	131
6.4.1	函数的嵌套调用	131
6.4.2	函数的递归调用	132
6.5	小结	134
	习题六	135
第 7 章	编译预处理	136
7.1	宏定义	136
7.1.1	不带参数的宏定义	136

7.1.2 带参数的宏定义	139
7.2 文件包含	141
7.3 条件编译	143
习题七	145
第 8 章 指针	146
8.1 地址和指针的概念	146
8.2 变量的指针和指向变量的指针变量	147
8.2.1 定义指针变量	147
8.2.2 指针变量的引用	148
8.2.3 指针变量作为函数参数	150
8.3 数组的指针和指向数组的指针变量	154
8.3.1 指向数组元素的指针	154
8.3.2 通过指针引用数组元素	155
8.3.3 数组名作函数参数	159
8.3.4 指向多维数组的指针和指针变量	166
8.4 字符串的指针和指向字符串的指针变量	169
8.4.1 字符串的表示形式	169
8.4.2 字符串指针作函数参数	171
8.4.3 对使用字符指针变量和字符数组的讨论	173
8.5 函数的指针和指向函数的指针变量	176
8.5.1 用函数指针变量调用函数	176
8.5.2 用指向函数的指针作函数参数	177
8.6 返回指针值的函数	180
8.7 指针数组和指向指针的指针	183
8.7.1 指针数组的概念	183
8.7.2 指向指针的指针	185
8.7.3 指针数组作 main()函数的形参	187
习题八	189
第 9 章 结构体与共用体	190
9.1 结构体类型的定义	190
9.2 结构变量的说明	191
9.3 结构变量成员的表示方法	193
9.4 结构变量的初始化	193
9.5 结构数组的定义	194
9.6 结构指针变量的说明和使用	195
9.6.1 指向结构变量的指针	195
9.6.2 指向结构数组的指针	197
9.6.3 结构指针变量作函数参数	198

9.7 共用体.....	200
9.7.1 联合的定义	200
9.7.2 联合变量的说明	200
9.8 链表	201
9.8.1 链表的概念	201
9.8.2 建立简单链表	202
9.8.3 输出链表	203
9.8.4 删除一个结点	203
9.8.5 插入结点	204
9.9 枚举类型	206
9.9.1 枚举类型的定义和枚举变量的说明	206
9.9.2 枚举类型变量的赋值和使用	206
9.10 类型定义符 typedef	207
习题九	208
第 10 章 位运算和文件	210
10.1 位运算	210
10.1.1 位运算概述	210
10.1.2 六种位运算	210
10.2 文件	215
10.2.1 为何引入文件	215
10.2.2 文件的定义	215
10.2.3 文件的分类	215
10.2.4 文件的操作过程	217
10.2.5 文件操作前的准备——文件指针	218
10.2.6 打开文件和关闭文件	219
10.2.7 读/写文件	223
10.2.8 读/写位置指针的定位操作	238
10.2.9 获取读/写位置指针当前所在位置的操作	241
10.2.10 文件状态检查函数	242
10.3 小结	246
习题十	247
附录 A 常用字符与 ASCII 码对照表	250
附录 B 在 Turbo C 集成环境下调试 C 程序的方法	251
附录 C 常用库函数介绍	256
附录 D Turbo C 编译错误信息汇总	269

第 1 章 | C 语言概述

1.1 C 语言简介

1.1.1 什么是程序设计语言

一台没有安装任何软件的计算机称为“裸机”，它是不能做任何工作的，只有安装了各种软件以后，计算机才能完成各种不同的工作，比如操作系统软件是管理、组织、协调计算机资源的最基本的软件，它是各种软件的核心与基础，所有其他程序的运行都要在操作系统的控制下进行；安装了办公软件，计算机才可以完成打字、排版、绘制表格的工作；安装了视频播放软件，用户就可以在计算机上观看视频。软件是整个计算机系统的重要组成部分，它驱使计算机中的硬件来完成各种不同的工作。

软件由程序和程序的相关文档组成（如源代码、说明书等），程序是软件中最重要的部分，而程序就是用各种具体的程序设计语言（如 C、BASIC、Pascal 等）来编写的。和人类之间的交流需要语言一样，人类和计算机之间也需要语言进行沟通，各种程序设计语言也就成为人与计算机之间交流的工具和媒介。人们使用程序设计语言编写一系列的指令（即程序），存储在计算机中，当执行程序时，计算机将自动按照一定的顺序逐条地调用指令来完成相应的工作。

1.1.2 C 语言的产生、发展与应用

C 语言是一种功能强大、已经广泛使用的计算机编程语言，它作为一种计算机高级语言，既具有一般高级语言的特性，也具有低级语言的一定特性。C 语言的这些特点使其成为程序员首选的编程语言之一。

早期的计算机都是用机器语言和汇编语言来编写程序代码的，它们编写的语句很难记忆，不容易理解，并且涉及很多计算机硬件知识，非专业人员很难掌握。在这样的背景下，很多研究人员开始致力于高级语言的开发。1963 年，英国剑桥大学推出了 CPL（Combind Programming Language）语言，1967 年英国剑桥大学的 Martin Richards 完成了对 CPL 的简化，形成 BCPL（Basic Combind Programming Language）语言。1970 年，美国电话电报公司（AT&T）的 Ken Thompson 在 BCPL 的基础上设计了一种无类型的、直接对机器字操作的 B 语言，并且用它编写出了 UNIX 操作系统。贝尔实验室的 Dennis Ritchies 于 1972 年设计出了 C 语言，它克服了 BCPL 语言和 B 语言过于简单、无数据类型等缺点。1973 年，Ken Thompson 和 Dennis

Ritchies 合作用 C 语言编写 UNIX 系统 90% 以上的部分, 并发表了 UNIX 5 版本。后来又对 C 语言做了多次改进, 1978 年以后, C 语言很快开始被广泛地应用于各类大、中、小和微型计算机上, 并且用来编写其他的操作系统, 成为当代最优秀的程序设计语言之一。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, 美国国家标准研究所 (ANSI) 为 C 语言制定了一套 ANSI 标准, 并成为现行的 C 语言标准。

高级语言经过几十年的发展, 已经达到了更高层次的水平, 尤其是以 Visual C++ 为代表的面向对象的程序设计方法, 得到了广泛的关注和应用, 但是新的程序设计语言的出现并不意味着 C 语言的消亡, 反而更加巩固了 C 语言的地位, 这是因为: 首先, C++、VC++ 都是以 C 语言为基础的, 只有打好基础, 上层建筑 (C++、VC++) 才能建造得更好、更稳; 其次, 在某些特定领域, 比如嵌入式软件开发领域, C 语言仍然占据主导地位。如图 1-1 所示为美国版 EETimes 子网站 Embedded—Systems Design, 13 年来针对嵌入式软件所应用的设计语言进行的调查统计。

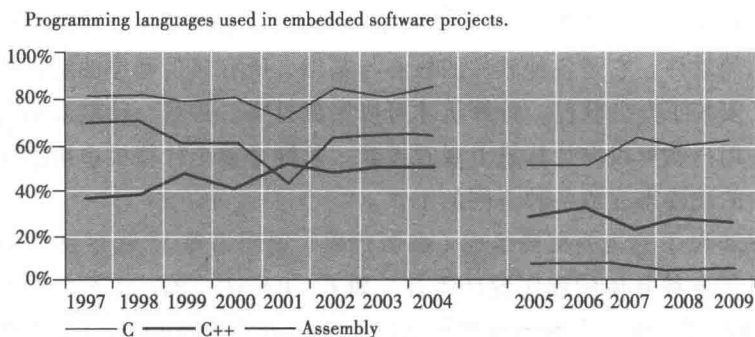


图 1-1 嵌入式软件所应用的程序设计语言调查统计图

从图 1-1 中可以看出, 在 1997—2009 年之间 C 语言都是嵌入式软件开发所使用最多的语言。近五年来, C 与 C++ 语言更瓜分了大部分原属汇编语言 (Assembly) 的版图, 其中较高层的 C++ 发展速度虽不如预期快, 但仍在嵌入式软件设计领域维持 27% 左右的占有率。整体看来, C++ 语言的使用率在 20 世纪 90 年代晚期加速上升, 到 2001 年达到高峰, 然后稍微下滑, 之后维持稳定。

1.1.3 C 语言的特点

C 语言发展如此迅速, 而且成为最受欢迎的语言之一, 主要因为它具有强大的功能, 许多著名的系统软件 (如 UNIX、PC-DOS、DBASE IV) 大部分都是由 C 语言编写的, 此外其他语言 (如 FORTRAN、Pascal、BASIC) 的许多编译器和解释器也都是用 C 语言编写的。目前 C 程序也已经用于解决物理学和工程学问题方面, 甚至用来为好莱坞电影制作特效。归纳起来 C 语言具有下列特点。

(1) 语言简洁、紧凑、灵活。

C 语言具有 32 个关键字, 9 种控制结构, 程序书写形式灵活自由, 主要用小写字母表示。

(2) 运算符丰富。

C语言共有34种运算符,其中包括一些C语言特有的运算符,比如增量运算符、逗号运算符、条件运算符等。使得C语言更容易实现算法。

(3) 功能丰富。

C语言具有各种各样的数据类型,能够方便地实现各种复杂的数据结构。并引入了指针概念,可使程序效率更高。另外,C语言也具有强大的图形功能,支持多种显示器和驱动器。此外其计算功能、逻辑判断功能也比较强大,可以实现做决策的目的。

(4) 结构化程序设计语言。

函数是C语言的基本单位,C语言的每项功能都是由函数实现的,这些函数可方便地调用,C语言还具有多种程序控制结构(如循环、条件),从而使程序完全结构化。

(5) 可移植性好。

C语言是一种可移植语言,在一个系统上编写的C程序可以不经修改就可以在其他系统上运行。和其他高级语言相比,C语言在可移植性方面处于领先地位,C编译器可在大约40种系统上使用,包括从使用8位微处理器的计算机到Cray超级计算机。

(6) 允许直接访问物理地址。

C语言中的位运算和指针运算,可以直接访问物理地址,可以实现汇编语言的大部分功能,直接对硬件进行操作。虽然C是一种高级语言,但同时它又兼具有低级语言的大部分功能,因此有时候也称C语言为“中级语言”。

(7) 高效性。

C语言在设计上充分利用了计算机计算能力强的优点,C程序代码紧凑而且运行速度快,它产生目标代码的执行效率仅次于用汇编语言编写的程序(大约低10%~20%),这在高级语言中已经是非常出众的了。

当然,任何事物都不可能是完美的,C语言也存在一些缺陷。例如,C语言的语法检查不太严格,而且程序书写自由度大,这些表达方面的自由无疑会增加出现编译错误的风险,尤其是C语言对指针(本书后面章节会学到)的使用,可能会使程序员犯一些非常难以察觉的编程错误,这就要求我们在编程的时候要多加注意这些问题,养成良好的编程习惯。正如一位计算机专家所说:“自由的代价是永远的警惕”。总的来说,C语言是一种优秀的、被广泛使用的程序设计语言。

1.2 程序设计初步

1.2.1 程序设计的一般步骤

对于初学程序设计的人们来说,其往往会认为所谓程序设计就是编写代码,然后运行成功就可以了,实际上这是一个误区,编写代码仅仅是程序设计中的一个步骤,而且还并不是最重要的环节。这是因为如果把前期的分析与设计做好,编写代码只是“水到渠成”的事情。基于软件工程的理论。如图1-2所示,我们把程序设计的过程分为以下几个步骤。

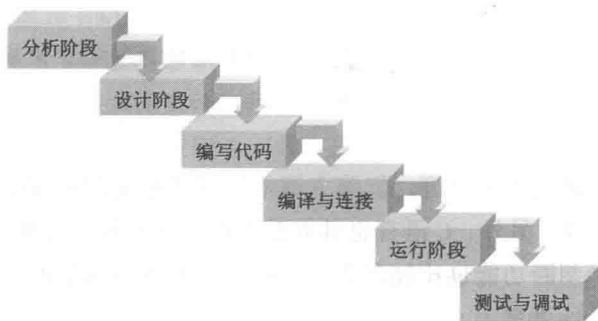


图 1-2 程序设计的 6 个步骤

(1) 分析阶段。

分析阶段是整个程序设计的基础，只有把待解决的问题进行准确的分析，才能保证最终程序的正确性和高效性。在这一步骤中，通过对问题性质的分析，要明确解决什么问题，编写的程序应该具有什么样的功能，也称为需求分析阶段。这一阶段并不关心如何完成程序。

(2) 设计阶段。

在对程序应该具有的功能有了一定的概念性认识之后，就要决定如何来完成这个程序了。在这一步骤中，基于在分析阶段得到的信息，要明确程序的界面应如何设计，程序要进行哪些计算和操作，数据结构将如何设计，特别是当处理复杂问题的时候，选择合适的数据结构会大大地提高程序的运行效率。这个阶段是对程序整体上的决策，并不涉及具体的代码实现细节。

(3) 代码编写阶段。

有了对程序的分析 and 设计之后，就将利用所学到的 C 语言知识编写程序代码。编写好的程序叫做源程序，并且把它保存到磁盘文件中。

(4) 编译与连接阶段。

编译是指将编辑好的源程序翻译成二进制的目标代码（计算机只能识别二进制的代码），在编译的时候，会对源程序进行语法检查，如果有错误，会在屏幕上提示出错信息（Error），这时候要对源程序进行修改再重新编译，直到通过编译为止。如果编译器提示出警告（Warning），并不会影响程序的运行。编译通过以后，得到的是以.obj为扩展名的（在MS-DOS环境下）二进制代码文件，这时，得到的二进制代码还不能直接运行，因为每一个模块都是单独编译的，必须把各个模块的二进制代码与系统标准模块（如C语言中的标准函数库）经过连接处理后，得到具有绝对地址的可执行文件，才能够在计算机中运行。在MS-DOS下，可执行文件的扩展名是.exe。

(5) 运行阶段。

如果程序编写正确，源程序经过编译和连接之后，便可以运行了。一些集成化的工具环境已经将编辑、编译、连接、调试工具集于一身，随后将会介绍一种集成化的C程序开发环境Turbo C。

(6) 测试和调试阶段。

某些情况下，程序虽然可以运行，但运行结果不正确，这是由于程序可能存在一些逻辑错误，计算机术语中称为bug。在实际编程时，无论程序规模大小，出现错误总是

难免的，这就需要对程序进行测试和调试工作。事实上，测试和调试程序往往比编程花费更多的时间。

所谓测试是指通过一些典型的例子（测试用例），尽可能多地发现程序中的错误。调试是指找出程序中错误的具体位置，并且改正错误。

1.2.2 编写第一个 C 语言程序

让我们通过一个简单的 C 语言程序，了解 C 语言的一些基本特征，进而可以自己编程实现一些简单的程序。

```
/*第一个 C 语言程序示例*/
#include<stdio.h>
main()                                /*主函数*/
{
    int n;                             /*声明一个整型变量 n*/
    n=1;                               /*给变量 n 赋值*/
    printf("it is the NO.%d C program\n",n); /*输出*/
}
```

运行结果在命令提示符窗口显示：it is the NO.1 C program

程序分析：

(1) 位于“/*”与“*/”之间的字符序列，称为注释，用于解释该程序的功能。程序开头部分的注释叫做宣言性注释，它是对程序的整体说明，主要描述内容包括：程序标题、程序功能说明、主要算法、接口说明、开发简历等；嵌在源程序体内部的注释叫做功能性注释，主要用来描述各条语句的功能。程序的注释不会给程序的编译、运行带来任何影响。

(2) #include<stdio.h>: #include 是一种 C 语言的预处理命令（Preprocessor Directive），预处理命令位于源程序或源文件的最前面。通常 C 编译器在编译前要对源程序做一些准备工作，称为预处理（Preprocessing）。#include 称为文件包含命令，其含义是把“<>”内的文件包含到本程序中，被包含的文件一般是由系统提供的，扩展名为“.h”，也称头文件或首部文件。本程序中用到的 stdio.h 是用于输入/输出的库函数文件，因为本程序中用到了 C 库函数中的格式输出函数 printf()。

(3) main()函数：C 语言以函数为模块的基本单位，一个完整的 C 程序可以由多个函数组成，但是必须包含一个而且只能包含一个 main()函数，main()函数又称为主函数，它是任何程序运行的入口，程序总是从 main()函数开始执行。

(4) 函数由函数首部和函数体两部分组成，函数首部即函数名，比如 main()，函数体由一对花括号{}括起来，注意花括号必须要成对出现。函数体内部又分为说明部分和执行部分，说明部分指对变量的类型声明，如本程序中的 int n;语句用来声明变量 n 是一个整型变量，执行部分用于完成程序功能，如本程序中的 print()语句用于实现输出功能。需要注意的是：C 语言规定，程序中所有用到的变量必须要先声明，然后才能使用。

(5) 函数体中的的语句必须以分号“;”结尾，但是程序开头预处理部分、函数首部以及花括号的后面不能有分号。

(6) 语句“n=1”中，“=”并不是我们平常认为的等号，它是一个赋值符号，其意义是把 n 这个变量赋值成 1。

(7) 从书写的角度来说，为了便于源程序的阅读、理解和修改，书写时要尽量做到：一条语句占一行，如果有必要还应在语句后面加上注释；“{}”通常表示程序的某一层结构，一般情况下，花括号独占一行，且与该结构语句的第一个字母对齐；低一层次的语句或说明应当比高一层次的语句缩进若干格后书写，同层次的语句采用对齐的方式书写，使程序看起来清晰、整齐。

1.3 在 Visual C++环境中编程

1.3.1 Visual C++集成开发环境介绍

Visual C++是一个功能强大的可视化软件开发工具。自 1993 年 Microsoft 公司推出 Visual C++ 1.0 后，随着其新版本的不断问世，Visual C++已成为程序员进行软件开发的首选工具。Visual C++ 6.0 不仅是一个 C++编译器，而且是一个基于 Windows 操作系统的可视化集成开发环境（integrated development environment, IDE）。Visual C++ 6.0 由许多组件组成，包括编辑器、调试器以及程序向导 AppWizard、类向导 Class Wizard 等开发工具。这些组件通过一个名为 Developer Studio 的组件集成为和谐的开发环境。

1. Developer Studio 开发环境界面

Developer Studio 开发环境有 4 个窗口：Workspace（工作空间）窗口、Output（输出）窗口、Edit（编辑）窗口和 Debug（调试）窗口。

(1) 工作空间窗口显示项目信息，窗口底部有三个选项卡，ClassView 选项卡中列出项目中的类和成员函数，ResourceView 选项卡列出项目的资源数据，FileView 选项卡列出项目的源文件、头文件。

(2) 输出窗口输出用户操作后的反馈信息，主要有以下几类信息：编译信息在编译时输出，主要显示编译时的警告和错误；调试信息在对程序进行调试时输出，主要显示程序的运行状况；结果信息在用户进行查找时输出，显示查找结果的位置。

(3) 编辑窗口用于编辑程序资源，包括菜单、对话框、图标、快捷键等。

(4) 调试窗口用于显示各种调试信息，包括变量信息、过程调用信息、内存信息和寄存器信息。

2. Visual C++菜单栏

Visual C++ 6.0 开发环境的菜单有：

File（文件）、Edit（编辑）、View（视图）、Insert（插入）、Project（工程）、Build（构建）、Tools（工具）、Windows（窗口）和 Help（帮助）。

下面介绍几个常用的菜单。

(1) File（文件）菜单中有 6 组选项，包括：

- New：新建一个文件、工程、工作区和其他文档。

- Open: 打开。
- Close: 关闭。
- Workspace: 工作区操作, 打开、保存和关闭工作区。
- Save: 文件保存。
- Print: 文件打印。
- Recent: 打开以前打开过的文件或者工作区。
- Exit: 退出。

(2) Edit (编辑) 编辑菜单中有 7 组选项, 包括:

- 撤销以前的编辑或重复前次编辑, 类似于 Office Word 中的撤销。
- 提供常见的编辑功能。
- 字符串查找和替换。
- Go to 和 Bookmark 编辑行定位和书签定位。
- Advanced 包括一些其他编辑手段。
- Breakpoints 设置断点, 调试时使用。
- 成员列表、函数参数信息、类型信息和自动完成。

(3) Build (构建) 菜单中包括 13 个选项:

- Compile: 编译当前文件。
- Build: 创建工程的可执行文件, 但是不运行。
- Rebuild All: 重新编译所有文件, 并连接生成可执行文件, 不运行。
- Batch Build: 成批编译、连接工程的不同设置。
- Clean: 删除编译、连接生成的中间文件和最终的可执行文件。
- Start Debug→Go: 开始调试, 在断点处停止。
- Start Debug→Step Into: 单步调试, 遇到函数进入函数体内部。
- Start Debug→Run to Cursor: 开始调试, 在光标处停止。
- Debugger Remote Connection: 用于远程连接调试。
- Execute: 运行编译、连接后产生的可执行文件。
- Set Active Configuration: 选择 Build 配置方式。
- Configuration: 增加或删除工程配置方式。
- Profile: 工程构建过程的描述文件。

3. Visual C++ 工具栏

Visual C++将常用的工具以图标形式排列在工具栏中, 使用时单击相应的图标按钮即可。每个按钮对应各自的功能, 常用的几个工具有: Standard 提供了最基本的功能, 如文件操作、编辑、查找等; Build 工程的编译、连接、调试等; Resource 添加各种类型资源; Edit 进行复制、剪切、粘贴等操作。

1.3.2 Visual C++环境中运行 C 语言源程序的步骤

用 C 语言编写的程序是可以在 C++中使用的, 下面将介绍如何用 Visual C++ 6.0 来调试 C

语言程序。

(1) 双击桌面上的 Visual C++快捷方式图标, 或者单击任务栏上“开始”按钮, 选择“所有程序”→Microsoft Visual Studio 6.0→Microsoft Visual C++ 6.0 命令, 单击打开 Microsoft Visual C++ 6.0 的操作界面, 如图 1-3 所示。

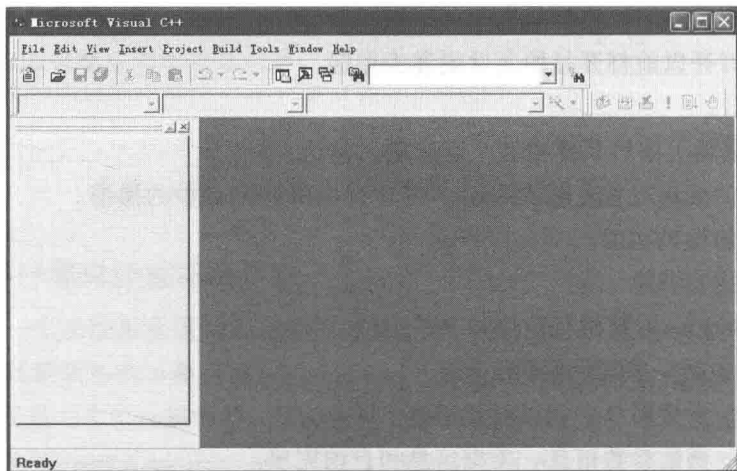


图 1-3 Microsoft Visual C++ 6.0 的操作界面

(2) 新建一个 C 语言源程序文件, 并将其命名为 first.c, 操作过程如下:

① 点击菜单栏上的 File 选项, 在下拉菜单中选择 New 命令, 这时会弹出一个 New 对话框, 选择对话框中的 Files 选项卡, 再选择 C++ Source File 选项。

② 在右边的 File 文本框中输入 C 源程序文件的名字, 本例中名为 first.c, 需要注意的是由于要调试 C 语言程序, 所以文件名一定要加上扩展名.c。如果不加, 系统会默认加上扩展名.cpp, 这是 C++源程序文件的扩展名。然后在下面的 Location 文本框中输入保存 C 源程序文件的磁盘路径, 最后单击 OK 按钮, 界面如图 1-4 所示。

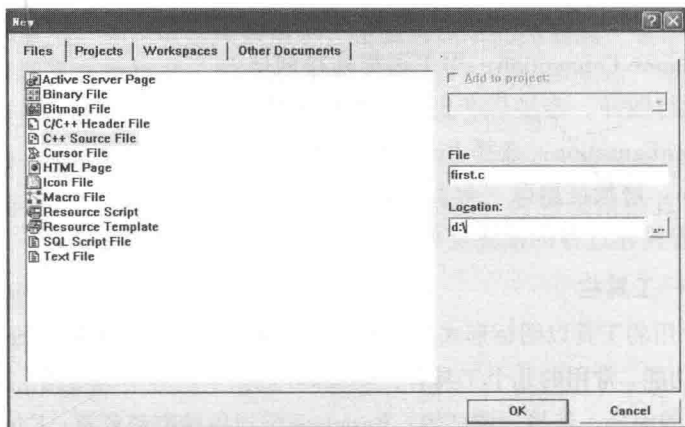


图 1-4 新建一个 C 文件

③ 在代码窗口内输入前面介绍的第一个 C 程序代码, 输入完毕后, 选择 File→Save 选项就可以保存该 C 源程序文件, 如果保存时想要更改文件名, 可以选择 File→Save As 命令, 为 C 源程序文件重新命名。