

 普通高等教育计算机规划教材

C语言程序设计

周 林 刘嘉敏 等编著



提供电子教案

下载网址 <http://www.cmpedu.com>



机械工业出版社
CHINA MACHINE PRESS



普通高等教育计算机规划教材

C 语言程序设计

周 林 刘嘉敏 等编著



机械工业出版社

本书以标准 C 语言为框架,从培养应用型人才的角度出发,按照紧扣基础和面向应用的原则,系统地介绍了 C 语言编程的基本知识、结构化程序设计的基本思想和解决实际问题的技巧。全书共分 11 章,主要内容包括 C 语言程序设计的基本概念、语法规则,基本数据类型、运算符与表达式, C 程序设计初步,结构化程序设计,数组,函数,指针,结构体、共用体与枚举类型,位运算,文件和学生信息管理系统案例等。

本书实例丰富,体系结构清晰,强调基础,突出应用。内容安排循序渐进,理论结合实际,特别强调培养学生的应用创新能力和程序设计的思路、方法及良好的程序设计风格,以提高其解决实际问题的能力。

本书既适合作为高等学校“C 语言程序设计”课程的教材,也可作为全国计算机等级考试和广大计算机爱好者自学 C 语言的参考用书。

本书配有电子教案,需要的教师可登录 www.cmpedu.com 免费注册、审核通过后下载,或联系编辑索取(QQ: 2399929378, 电话: 010-88379753)。

图书在版编目(CIP)数据

C 语言程序设计 / 周林等编著. —北京: 机械工业出版社, 2013.8

普通高等教育计算机规划教材

ISBN 978-7-111-43079-7

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(2013)第 168473 号

机械工业出版社(北京市百万庄大街 22 号 邮政编码 100037)

责任编辑: 和庆娣

责任印制: 张楠

涿州市京南印刷厂印刷

2013 年 8 月第 1 版·第 1 次印刷

184mm×260mm·22.5 印张·558 千字

0001—3000 册

标准书号: ISBN 978-7-111-43079-7

定价: 49.00 元

凡购本书,如有缺页、倒页、脱页,由本社发行部调换

电话服务

网络服务

社服务中心: (010) 88361066

教材网: <http://www.cmpedu.com>

销售一部: (010) 68326294

机工官网: <http://www.cmpbook.com>

销售二部: (010) 88379649

机工官博: <http://weibo.com/cmp1952>

读者购书热线: (010) 88379203

封面无防伪标均为盗版

出版说明

信息技术是当今世界发展最快、渗透性最强、应用最广的关键技术，是推动经济增长和知识传播的重要引擎。在我国，随着国家信息化发展战略的贯彻实施，信息化建设已进入了全方位、多层次推进应用的新阶段。现在，掌握计算机技术已成为 21 世纪人才应具备的基础素质之一。

为了进一步推动计算机技术的发展，满足计算机学科教育的需求，机械工业出版社聘请了全国多所高等院校的一线教师，进行了充分的调研和讨论，针对计算机相关课程的特点，总结教学中的实践经验，组织出版了这套“普通高等教育计算机规划教材”。

本套教材具有以下特点：

- 1) 反映计算机技术领域的新发展和新应用。
- 2) 为了体现建设“立体化”精品教材的宗旨，本套教材为主干课程配备了电子教案、学习与上机指导、习题解答、多媒体光盘、课程设计和毕业设计指导等内容。
- 3) 针对多数学生的学习特点，采用通俗易懂的方法讲解知识，逻辑性强、层次分明、叙述准确而精炼、图文并茂，使学生可以快速掌握，学以致用。
- 4) 符合高等院校各专业人才的培养目标及课程体系的设置，注重培养学生的应用能力，强调知识、能力与素质的综合训练。
- 5) 注重教材的实用性、通用性，适合各类高等院校、高等职业学校及相关院校的教学，也可作为各类培训班和自学用书。

希望计算机教育界的专家和老师能提出宝贵的意见和建议。衷心感谢计算机教育工作者和广大读者的支持与帮助！

机械工业出版社

前 言

C 语言是一门优秀的计算机程序设计语言，具有功能丰富、表达能力强、使用灵活方便、编写程序效率高及可移植性等优点，兼有高级语言和低级语言的特点，是一门既适合编写系统软件，又适合设计应用软件的高级语言。

C 语言程序设计课程发展数十年，相关教材也层出不穷。由于这些教材各有定位，故存在一些难以顾及的地方。比如，有些教材中 C 语言知识讲授有余，而程序设计思想和应用实践阐述不足。因材施教是教学工作者一直传承的理念，要促进学生在计算机技术方面的提升，必须针对所培养人才的特点，更新教学内容和教学方法。为了更好地推动高等教育体制的改革、更好地服务于计算机应用型人才的教育工作，编者决定编写这部 C 语言程序设计教材。本书由长期承担计算机程序设计基础课程教学、具有丰富教学和实践经验的一线教师编写。本书以程序设计为主线，编程应用为驱动，通过丰富的实例详细介绍了 C 语言程序设计的思想和方法。本书的主要特点如下。

1. 面向初学者，语言通俗，叙述简单

本书在编写过程中，充分考虑了高等院校培养应用型本科专业学生的要求，在内容的编排上充分考虑了初学者的需求，针对每个知识点，采用通俗的语言进行讲解，使学生更加容易学习并掌握 C 语言，理解结构化程序设计的方法，学以致用。

2. 面向应用，便于提高

面向应用型高等院校，在保证学科体系完整的基础上，不过度强调理论的深度和难度，注重应用型人才专业技能和工程实用技术的培养。尽量淡化语法规则复杂的叙述，避免学生一开始就陷入烦琐的语法规则之中。指导学生学习 C 语言程序设计的目的不仅仅是为了记忆这些语法规则，更重要的是以 C 语言为平台，学习一种编程的方法，进而通过程序设计来解决日常生活当中面临的实际问题。

3. 实用性强

通过案例“学生信息管理系统”项目的学习与开发，将分散的知识点进行有机联系，真正达到学以致用的目的。读者可以通过借鉴课程中的项目经验，开发其他实用的小型应用软件。

全书由周林提出框架，负责统稿、定稿，刘嘉敏对全书进行了审校。本书第 2、3、4、5、6、7、8 章由周林编写，附录由刘嘉敏编写，第 1 章由李秀苑编写，第 9、10 章由何建军编写，第 11 章由颜烨编写。此外，在本书的编写过程中得到了众位专家和领导的关心与支持，在此一并表示诚挚的感谢。

我们力求在编写过程中做到严谨细致、精益求精，但由于编者水平有限，书中难免有疏漏和不当之处，恳请广大读者指正。

编 者

目 录

出版说明

前言

第1章 C语言概述 1

1.1 程序及程序设计语言 1

1.1.1 程序与程序设计 1

1.1.2 程序设计语言 2

1.1.3 C语言的发展历程 3

1.1.4 C语言的特点 4

1.2 一个基本的C应用程序 5

1.2.1 开发环境的安装 5

1.2.2 C应用程序 7

1.3 程序的编译和运行 9

1.3.1 编译 9

1.3.2 运行 10

1.3.3 Visual C++ 6.0 基本操作 10

1.4 常见错误 15

1.4.1 编译时的错误 15

1.4.2 运行时的错误 16

小结 16

习题 17

第2章 C语言基础 19

2.1 C语言的基本元素 19

2.1.1 标识符 19

2.1.2 关键字 20

2.1.3 分隔符 20

2.2 数据类型 21

2.2.1 基本数据类型 21

2.2.2 构造类型 22

2.2.3 指针类型 22

2.2.4 空类型 23

2.3 常量与变量 23

2.3.1 常量 23

2.3.2 变量 25

2.4 各类数值型数据间的混合运算 28

2.4.1 自动转换 28

2.4.2 强制转换 29

2.4.3 赋值表达式的类型转换 30

2.5 运算符与表达式 31

2.5.1 算术运算符和算术表达式 32

2.5.2 关系运算符和关系表达式 35

2.5.3 逻辑运算符和逻辑表达式 36

2.5.4 赋值运算符和赋值表达式 38

2.5.5 条件运算符和条件表达式 40

2.5.6 逗号运算符和逗号表达式 41

2.5.7 长度测试运算符 42

2.5.8 运算符的优先级与结合性 43

小结 44

习题 44

第3章 C程序设计初步 49

3.1 结构化程序设计思想 49

3.1.1 结构化程序设计的方法 49

3.1.2 结构化程序的标准 50

3.1.3 结构化程序设计的三种基本结构 50

3.2 C语句概述 51

3.2.1 C语句的作用与分类 51

3.2.2 最基本的语句——赋值语句 54

3.3 数据的输入与输出 55

3.3.1 有关输入/输出的概念 55

3.3.2 数据的输出函数 55

3.3.3 数据的输入函数 61

3.4 顺序结构程序设计 65

小结 67

习题 67

第4章 结构化程序设计 72

4.1 选择结构程序设计 72

4.1.1 if语句 72

4.1.2 嵌套的if语句 76

4.1.3 switch语句 77

4.1.4 选择结构程序举例	79	6.3 函数的调用	142
4.2 循环结构程序设计	83	6.3.1 函数调用的一般形式	142
4.2.1 while 语句	83	6.3.2 函数调用的方式	143
4.2.2 do-while 语句	84	6.3.3 函数的声明	145
4.2.3 for 语句	85	6.4 函数的参数传递方式	146
4.2.4 3 种循环结构的比较	86	6.4.1 值传递方式	147
4.2.5 循环的嵌套	87	6.4.2 地址传递方式	148
4.2.6 循环结构程序举例	89	6.5 函数的嵌套调用和递归调用	151
4.3 辅助控制语句	92	6.5.1 函数的嵌套调用	151
4.3.1 break 语句	92	6.5.2 函数的递归调用	152
4.3.2 continue 语句	92	6.6 变量的存储类型及作用域	153
小结	93	6.6.1 变量的存储类型	153
习题	94	6.6.2 局部变量	154
第 5 章 数组	100	6.6.3 全局变量	157
5.1 数组概述	100	6.6.4 内部函数和外部函数	159
5.2 一维数组	101	6.7 编译预处理	161
5.2.1 一维数组的定义	101	6.7.1 宏定义	161
5.2.2 一维数组的初始化	102	6.7.2 文件包含	163
5.2.3 一维数组的引用	103	6.7.3 条件编译	164
5.2.4 一维数组程序举例	104	6.8 应用举例	166
5.3 二维数组	109	小结	171
5.3.1 二维数组的定义	109	习题	172
5.3.2 二维数组的初始化	110	第 7 章 指针	179
5.3.3 二维数组的引用	113	7.1 指针概述	179
5.3.4 二维数组程序举例	114	7.1.1 地址、指针和指针变量	179
5.4 字符数组	119	7.1.2 指针变量的定义	181
5.4.1 字符数组的定义	119	7.1.3 指针变量的引用	183
5.4.2 字符数组的输入与输出	122	7.1.4 指针运算	187
5.4.3 常用字符串函数	124	7.2 指针与数组	189
5.4.4 字符数组程序举例	128	7.2.1 指针与一维数组	189
小结	131	7.2.2 指针与多维数组	200
习题	132	7.2.3 指针与字符串	207
第 6 章 函数	138	7.3 指针与函数	216
6.1 函数概述	138	7.3.1 指针作为函数参数	217
6.2 函数的定义	139	7.3.2 返回指针值的函数	219
6.2.1 无参函数的定义形式	139	7.3.3 指向函数的指针	221
6.2.2 有参函数的定义形式	140	7.4 复杂指针	226
6.2.3 空函数的定义形式	141	7.4.1 指针数组	226
6.2.4 函数的返回值	142	7.4.2 多级指针	228

7.4.3 指针数组作为 main 函数的形参···	231	9.1.2 左移运算符·····	294
7.5 内存的动态分配和指向它的		9.1.3 右移运算符·····	294
指针变量·····	232	9.1.4 按位与运算符·····	295
7.5.1 内存动态分配的概念·····	232	9.1.5 按位或运算符·····	296
7.5.2 怎样建立内存的动态分配·····	232	9.1.6 异或运算符·····	297
7.6 指针应用举例·····	234	9.1.7 位运算赋值运算符·····	298
小结·····	240	9.1.8 不同长度的数据进行位运算·····	298
习题·····	241	9.2 位运算的应用·····	300
第 8 章 结构体、共用体和枚举类型···	247	9.3 位运算举例·····	303
8.1 定义和使用结构体变量·····	247	小结·····	305
8.1.1 结构体类型的声明·····	247	习题·····	306
8.1.2 结构体变量的定义·····	248	第 10 章 文件·····	308
8.1.3 结构体变量的赋值·····	251	10.1 文件概述·····	308
8.1.4 结构体变量的引用·····	252	10.1.1 文件的概念·····	308
8.2 结构体与数组·····	255	10.1.2 文件的类型·····	308
8.2.1 结构体数组的定义·····	256	10.1.3 文件类型指针·····	309
8.2.2 结构体数组的初始化·····	257	10.2 文件的打开与关闭·····	309
8.2.3 结构体数组元素的引用·····	257	10.2.1 文件的打开·····	309
8.3 结构体与指针·····	259	10.2.2 文件的关闭·····	311
8.3.1 指向结构体变量的指针·····	259	10.3 文件的读/写·····	311
8.3.2 指向结构体数组的指针·····	261	10.3.1 文本文件的读/写·····	311
8.4 结构体与函数·····	263	10.3.2 二进制文件的读/写·····	313
8.4.1 结构体作为函数的参数·····	263	10.3.3 格式化读函数 fscanf 和写函数	
8.4.2 结构体作为函数的返回值·····	267	fprintf·····	314
8.5 共用体·····	270	10.4 文件的定位·····	314
8.5.1 共用体类型的声明·····	270	10.5 文件检测函数·····	315
8.5.2 共用体变量的定义·····	270	10.6 文件应用举例·····	316
8.5.3 共用体变量的引用·····	272	小结·····	321
8.6 枚举类型·····	275	习题·····	322
8.6.1 枚举类型的说明·····	275	第 11 章 C 语言系统开发案例——	
8.6.2 枚举类型变量的定义·····	276	学生信息管理系统·····	324
8.6.3 枚举类型变量的赋值和引用·····	277	11.1 总体方案的设计·····	324
8.7 用 typedef 声明新类型名·····	278	11.1.1 系统功能·····	324
8.8 应用举例·····	282	11.1.2 系统结构图·····	324
小结·····	285	11.2 数据结构的设计·····	325
习题·····	286	11.3 总控模块的设计·····	326
第 9 章 位运算·····	293	11.4 文件的打开和保存模块的	
9.1 位运算符·····	293	设计·····	328
9.1.1 取反运算符·····	293	11.4.1 文件打开模块·····	328

11.4.2 文件保存模块.....	330
11.5 添加和删除模块的设计	331
11.5.1 添加模块	331
11.5.2 删除模块	334
11.6 插入和更新模块的设计	335
11.6.1 插入模块	336
11.6.2 更新模块	337
11.7 查询和排序模块的设计	340
11.7.1 查询模块	341
11.7.2 排序模块	343
11.8 输出和退出模块的设计	344

11.8.1 输出模块	344
11.8.2 退出模块	345
小结.....	346
习题.....	346
附录.....	347
附录 A C 语言的关键字	347
附录 B 常用字符与 ASCII 码 对照表	347
附录 C 运算符的优先级与结合性 ...	348
附录 D C 语言常用库函数	349
参考文献.....	352

第1章 C语言概述

想要学好一门程序设计语言、做好程序设计，首先应该从了解程序、程序设计和程序设计语言开始。本章介绍程序、程序设计及程序设计语言的有关概念，了解结构化程序设计的基本思想；介绍 C 语言的发展概况，C 语言程序的基本结构、开发过程；介绍在 Microsoft Visual C++ 6.0 的集成开发环境下编译、连接和运行 C 程序的基本操作步骤。

通过本章的学习，读者可以了解关于程序、程序设计和程序设计语言等基本概念，并对 C 语言有一个初步了解，为进一步学习 C 语言程序设计打下基础。

1.1 程序及程序设计语言

现代计算机的基本工作原理是存储程序和程序控制。由此可以看出，计算机之所以能够自动工作，是因为有程序在运行和控制它。本节将讨论与程序有关的内容。

1.1.1 程序与程序设计

1. 程序

一般来说，程序就是人们通过计算机解决问题的方法和步骤，是一组计算机能识别和执行的指令序列。一条指令通常包括地址码和操作码两个部分，其中操作码规定了计算机要执行的基本操作。一组特定的指令用来完成一定的功能。

人们使用计算机解决问题，必须按照任务的要求写出一系列的指令，告诉计算机“干什么”和“怎么干”。当然，这些指令必须是计算机能够识别和执行的指令，即每一条指令是一台特定计算机指令系统中具有的指令。一台计算机的指令是有限的，但用它们可以编制出各种不同的程序，计算机的工作就是通过执行这些程序来完成人们交给它的任务。

2. 程序设计

程序设计就是编制并记录解决问题的方法和步骤的过程，换句话说，程序设计就是设计计算机工作所需要的程序，通过程序更好地控制计算机来完成约定的功能。程序设计是一个完整的过程，是软件活动的重要组成部分，而不是简单地编写程序，编写程序只是整个程序设计过程中的一个环节而已。从实践来看，编写程序这个任务在整个程序设计中占有的比例往往不到 1/3。在一个项目中，通常是将更多的精力和时间用在前期的问题分析、算法设计和后期的文档编写等方面。因此，想要提高自己的程序设计能力，除了“编写程序”以外，更应该注重培养分析问题、解决问题的能力，不要一味地去研究各种高深的语法细节。

当然，对于一些简单的问题，一般不会过多地考虑程序设计，甚至不需要计算机也能解决。但是，对于规模比较大的应用开发，往往需要有“工程”的思想指导程序设计。由于“软件危机”的出现，新的程序设计方法应运而生，从结构化程序设计到面向对象的程序设计，再到构件。

结构化程序设计就是按照一组能提高程序的可阅读性与易维护性的规则而进行程序设计

的方法，它可使程序具有合理的结构以便于保证和容易验证其正确性。结构化程序设计的原则是：采用自顶向下、逐步求精的方法进行程序设计；程序结构模块化，每个模块保持单入口和单出口的形式；使用 3 种基本控制结构描述程序流程。结构化程序设计方法是以过程为中心的开发方法。虽然结构化程序设计方法采用了许多符合人类思维习惯的原则和策略，但是，由于结构化程序设计方法把密切相关、相互依赖的数据和对数据的操作相互分离，这种实质上的依赖与形式上的分离使得大型程序不仅难以编写，而且难于调试和修改，更谈不上代码的重用。由于现代应用程序规模越来越大，对代码的可重用性与可扩展性的要求也相应提高，因此，面向对象的程序设计技术便广泛应用起来。

面向对象的程序设计技术是一种以对象为基础，以事件或消息来驱动对象执行处理的程序设计技术。

1.1.2 程序设计语言

程序设计语言是人与计算机之间进行信息交流的工具，是用来编写程序的工具。如同自然语言有汉语、英语等不同的种类，程序设计语言也分很多种，且不同语言的描述形式也各不相同。若要很好地使用计算机，就必须学会使用计算机能够理解的程序设计语言。

随着计算机软件和硬件技术的发展，计算机程序设计语言也在不断地发展，经历了从机器语言、汇编语言到高级语言的历程。

1. 机器语言

存储程序控制原理规定，在计算机内部用二进制表示指令和数据。而早期的计算机没有配置任何的软件，事实上，在当时还没有形成“软件”的整体概念。计算机只能识别“0”和“1”两种代码，所以，程序设计人员只能通过使用“0”和“1”这样的二进制代码来编写程序。机器语言就是由一种用“0”和“1”这样的二进制代码表示的能够被计算机直接识别和执行的计算机程序设计语言。

机器语言直接与具体计算机的指令相关，不同计算机的指令系统不一样，其机器语言的格式也不尽相同。因此，机器语言也称为面向机器的语言，使用机器语言编写的程序称为机器语言程序。这样的程序，由于其每条指令都对应着计算机一个特定的基本动作，所以程序占用内存少、执行效率高。但是，机器语言对于人来说，显得烦琐、复杂、难以记忆且晦涩难懂。用机器语言编制程序是相当费时和费力的工作，工作量大，还经常容易出错，且编制出来的程序修改起来十分困难，程序的通用型、移植性都很差。

2. 汇编语言

为了解决使用机器语言编写程序所带来的一系列问题、减轻程序设计人员的工作量和难度、提高程序设计的效率和质量，人们采用一定的助记符来表示机器语言。使用类似英语缩略词且带有助记性的符号来表示的语言称为汇编语言，又叫符号语言。汇编语言是机器语言符号化的结果。

用汇编语言编写的程序，称为汇编语言程序，计算机不能够直接接受，必须通过某种方法将这些符号翻译成计算机系统能够理解并执行的机器指令，然后交给计算机执行。这个翻译工作称为“汇编”，执行这个翻译工作的程序叫做汇编程序或汇编系统。

相对机器语言而言，汇编语言更易于读写，汇编语言程序易于检查和修改程序错误，而且指令、数据的存放位置可以由计算机系统自动分配。但是，汇编语言依然依赖于计算机的

系统结构，使用汇编语言编写计算机程序，程序员仍然需要十分熟悉计算机系统的结构，所以，从程序设计本身来看仍然是烦琐、低效率的。但是由于汇编语言与机器语言非常接近，具有执行效率高等特点，汇编语言适用于编写直接控制机器操作的底层程序。迄今为止，在某些特定的场合（如对时空效率要求很高的系统核心程序以及实时控制程序等），汇编语言仍然是十分有效的程序设计工具。

机器语言和汇编语言都依赖于具体的计算机结构，是面向机器的程序设计语言，编制程序的工作量大，移植性比较差，无通用性。一般把机器语言和汇编语言统称为低级语言。

3. 高级语言

为了克服低级语言的缺点，人们又开发出了高级语言。高级语言比较接近于数学语言或自然语言，易于理解，相对于低级语言来说，更符合人的认知模式。同时，在一定程度上可以不依赖于具体的计算机硬件，编写的程序能够在所有的机器上通用。

用高级语言编写的程序称为源程序。源程序的编制相对简单、直观、不容易出错，但计算机系统不能够直接接受和执行，只有通过翻译系统将用高级程序设计语言编写的源程序转换为计算机系统能够识别、理解的目标程序，才能被计算机系统执行。高级语言的翻译过程有两种不同的方式：第一种是解释执行方式，其特点是对源程序的每一可能的行为，都以机器语言编写一个子程序，用来模拟这一行为，完成这种翻译工作的程序叫做解释程序；第二种方式是将源程序全部翻译成等效的机器语言程序后，才可以执行的编译方式，编译方式中的翻译软件叫做编译系统。

FORTRAN 是第一个高级程序设计语言，又叫做公式翻译程序设计语言，是由美国计算机科学家约翰·巴克斯领导的小组于 1954 年在 IBM 公司的 704 系列计算机上开发的，主要应用于科学和工程的计算。目前在计算机中广泛使用、影响较大的高级语言主要有 FORTRAN、ALGOL、COBOL、BASIC、LISP、SNOBOL、PL/1、Pascal、C、PROLOG、Ada、C++、Delphi、Java 等。

总之，高级语言的发展经历了从传统的高级程序设计语言到结构化程序设计语言，从面向过程到面向对象的程序设计语言的过程，其发展趋势是朝着人们更容易学习、更容易理解和更容易使用的方向前进。

1.1.3 C 语言的发展历程

C 语言是高级程序设计语言，可以追溯到 ALGOL 60。ALGOL 60 是英国剑桥大学于 1960 年推出的一种面向问题的高级语言，但由于其自身的局限性，不宜用来编写系统程序。因此，1963 年剑桥大学在 ALGOL 60 的基础上开发出了 CPL 语言，但由于 CPL 语言规模比较大，难以实现。于是，Martin Richards 对 CPL 做了改进，于 1967 年推出了 BCPL 语言，BCPL 语言主要是对 CPL 语言进行了简化。

1970 年美国贝尔实验室以 BCPL 语言为基础，又做了进一步简化，同时，加强了硬件处理能力，推出了 B 语言。但后来发现，B 语言由于太过简单，导致功能有限。所以，UNIX 的研制者丹尼斯·里奇（Dennis Ritchie）于 1972~1973 年，在 B 语言的基础上，进一步完善和扩充，设计开发出了 C 语言。C 语言既保持了 BCPL 和 B 语言的优点，又克服了它们的缺点，最初用于实现 UNIX 操作系统。

自 C 语言被开发出来之后，在实际应用中也曾被多次改进。1973 年贝尔试验室的丹尼

斯·里奇和肯·汤普逊（Ken Thompson）用 C 语言将 UNIX 操作系统（即 UNIX 第 5 版）重写了一遍，使得 UNIX 操作系统成为第一个不是用汇编语言编写的主流操作系统。到 1975 年，UNIX 第 6 版公开发布。随着 UNIX 日益广泛的使用，C 语言的突出优点才引起人们的普遍注意并得到迅速推广。1978 年以后，C 语言已独立于 UNIX，先后移植到大、中、小、微型机上。现在 C 语言已风靡全世界，由于其自身的特点，适合作为系统描述语言，既可用于编写系统软件，也可用来开发应用软件，成为世界上应用最广泛的几种程序设计语言之一。

1978 年，Brian W·Kernighan 和丹尼斯·里奇以 UNIX 第 7 版中的 C 语言编译程序为基础，合著了《The C Programming Language》，这本书中介绍的 C 语言成为后来广泛使用的 C 语言版本的基础，它被称为标准 C。1983 年，美国国家标准化协会对 C 语言的各种版本做了扩充和完善，推出了新的标准，被称为 ANSI C，它比原来的标准 C 有了很大的改进和发展。1989 年，ANSI 对各种版本 C 进行发展和扩充并修订，发布了新的标准，称为 C89。1990 年，国际化标准组织 ISO 接受 C89 作为国际标准 ISO/IEC 9899: 1990，它和 ANSI 的 C89 基本上是相同的。目前，最新的 C 语言标准是 1999 年发布的 C99，C99 基本保留了 C89 的全部特性，并在 2000 年 3 月被 ANSI 采用。

1.1.4 C 语言的特点

C 语言自诞生之日起，就受到了全世界的关注。尤其是当开发者们为个人计算机提供 C 编译系统之后，C 语言的应用便得到了迅速推广。C 语言的诸多突出的特点使得它受到了大多数程序员和大众的喜爱。事实上，C 语言完全符合结构化、模块化程序设计规范，不仅适合编写系统软件，也可以开发应用程序。C 语言主要有如下一些显著的特点。

1. 语言简洁、紧凑，使用方便、灵活

C 语言总共有 37 个关键字、9 种控制语句，程序书写形式自由，主要用小写字母表示，压缩了一切不必要的成分。C 语言程序比其他许多高级语言程序简练，源程序较短，因此输入程序时工作量一般不大。

2. 运算符丰富，表达式多样化

C 语言的运算符包含的范围十分广泛，总共有 34 种运算符。C 语言把逗号、赋值和强制类型转换等都作为运算符处理，从而使 C 语言的运算符类型极其丰富，表达式类型多样化。灵活使用各种运算符可以实现在其他高级程序设计语言中难以实现的运算，增强了 C 语言的运算功能。

3. 数据类型丰富

C 语言的数据类型有字符型、整型、浮点型、数组类型、指针类型、结构体类型和共用体类型等，C99 又扩充了复数浮点类型、超长整型和布尔类型等。在 C 语言所支持的这些数据类型中，尤其是指针类型数据，使用十分灵活和多样化，能用来实现各种复杂的数据结构（如链表、树、栈）。

4. 具有结构化的控制语句

C 语言程序的主要结构成分是函数，函数允许一个程序中的各个任务分别定义和编码。用函数作为程序的模块单位便于实现程序的模块化。C 语言还提供了多种结构化的控制语句，如用于判断的 if...else 语句、switch 语句，用于实现循环的 while 语句、do...while 语句

和 for 语句等，便于采用自顶向下、逐步细化的结构化程序设计技术。

5. 语法限制不太严格，程序设计自由度大

一般的高级语言对语法的检查都比较严格，几乎能查出所有的语法错误，而 C 语言允许程序员有较大的自由度，因此放宽了语法检查。程序员应当仔细检查程序中有可能出现的错误，例如数组下标越界等，从而保证程序正确，而不要过多地依赖 C 语言编译程序查错。“限制”与“灵活”是一对矛盾。限制太严格，就失去灵活性；而强调灵活，就必然放松限制。对于刚刚接触 C 语言的初学者来说，编写一个正确的 C 语言程序可能会比编写一个其他高级语言程序要难一些。换句话说，对于使用 C 语言的人来说，可能要求相对高一些。

6. C 语言允许直接访问物理地址，可以直接对硬件进行操作

C 语言既具有高级语言的功能，又具有低级语言的许多功能，能对位、字节和地址进行操作，可以实现汇编语言的大部分功能，可直接对硬件进行操作。C 语言的这种双重性，使得它既是成功的系统描述语言，又是通用的程序设计语言。因此，既可以用 C 语言来编写大型系统软件，也可以用它编写各种应用软件，许多以前只能用汇编语言处理的问题，现在可以改用 C 语言处理了。

7. 生成目标代码质量高，程序执行效率高

C 语言的翻译程序是高效编译系统，生成的目标代码质量高。一般情况下，C 语言编译的目标代码只比汇编程序生成的目标代码效率低 10%~20%，而比其他高级语言高很多。

事实上，正是由于 C 语言的这些特点，使其得到了广泛的应用，成为软件行业的一种主流的程序设计语言，甚至在某些领域中的地位是无可替代的。

1.2 一个基本的 C 应用程序

C 语言自诞生以来，出现了许多编译系统，早期非常流行的有 Turbo C，它是美国 Borland 公司生产的一套 DOS 平台上的 C 语言编译系统。随着面向对象的程序设计语言 C++ 问世，又出现了 Borland C++、Visual C++ 等针对 C++ 语言的集成开发环境。虽然 C++ 集成开发环境是针对 C++ 的，而且增加了许多其他特性，但由于 C++ 是 C 语言的超集，所以单就 C 语言程序来说，凡是符合 ANSI C 的标准，在这些集成开发环境中都能够很好地编译和连接。

本书采用 Microsoft Visual C++ 6.0 作为 C 语言程序的开发工具，书中的所有实例程序均在 Visual C++ 6.0 环境下编写、调试和运行测试。下面介绍 Visual C++ 6.0 开发环境的安装和一个基本的 C 应用程序。

1.2.1 开发环境的安装

将 Visual C++ 6.0 系统光盘插入驱动器，可在“我的电脑”或“资源管理器”中双击光盘驱动器图标，或者下载 Visual C++ 6.0 的安装程序，找到文件 setup.exe，双击该文件，即可运行安装向导，如图 1-1 所示。

根据 Visual C++ 6.0 安装向导的中文提示，单击“下一步”按钮进行安装，系统提示是否接受协议，如图 1-2 所示，选择“接受协议”单选框。

单击图 1-2 中的“下一步”按钮，系统要求输入产品号、姓名和公司名称，输入正确出

现如图 1-3 所示的对话框，采用默认设置，单击“下一步”按钮，打开安装组件选择对话框，如图 1-4 所示。

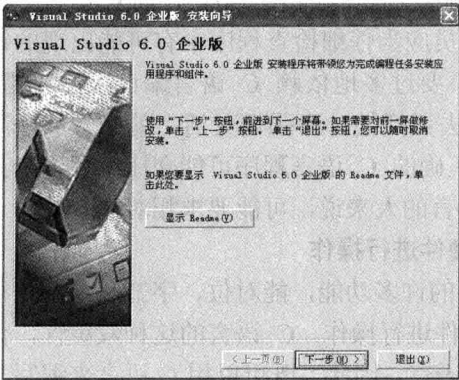


图 1-1 安装向导一

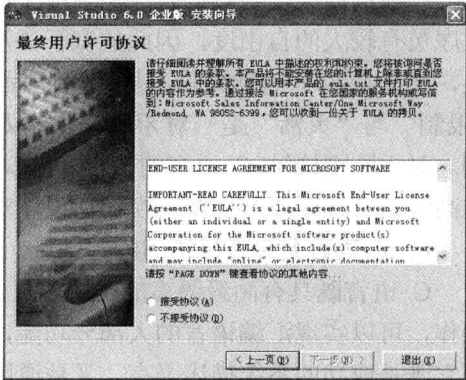


图 1-2 安装向导二

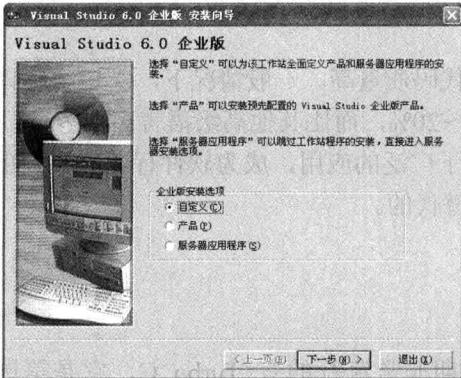


图 1-3 安装向导三

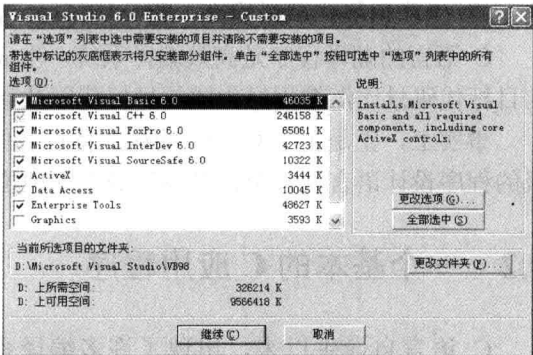


图 1-4 安装向导四

在选项列表中选中 Microsoft Visual C++ 6.0 复选框，安装程序开始复制文件，如图 1-5 所示。



图 1-5 安装向导五

安装完成后，可以选择安装 MSDN，MSDN 含有 Visual Studio 的帮助信息，但需要插入 MSDN 光盘，如图 1-6 所示。若没有光盘，此处单击“退出”按钮，结束 Visual C++ 的安装。

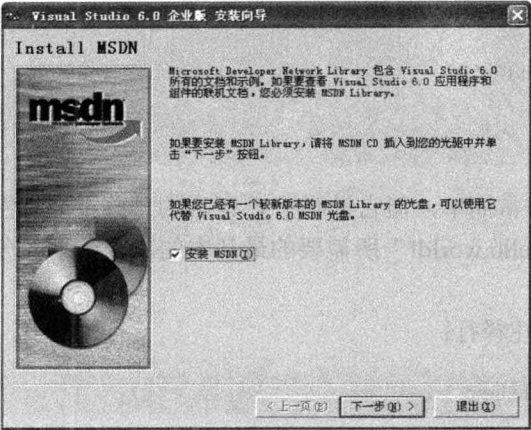


图 1-6 安装向导六

Visual C++ 6.0 安装完成后，可以在桌面上建立一个快捷方式，便于启动。单击任务栏上的“开始”按钮。在弹出的子菜单中选择“所有程序”，将鼠标指针移动到“Microsoft Visual Studio 6.0”菜单项，在出现的菜单项中右击“Microsoft Visual C++ 6.0”命令，弹出快捷菜单。选择“发送到”→“桌面快捷方式”命令，创建 Visual C++ 6.0 的桌面快捷方式，如图 1-7 所示。



图 1-7 创建 Visual C++ 6.0 桌面快捷方式

1.2.2 C 应用程序

和其他任何程序设计语言一样，C 语言也可以创建通常意义下的应用程序。下面编制一个仅在屏幕上显示字符串“Hello,World!”的最小的应用程序，以此说明 C 语言程序的基本结构。该程序存储在 hello.c 文件中。

【例 1-1】 一个基本的 C 应用程序，显示一行文本。

```
//简单的应用程序 hello.c
#include <stdio.h>
int main()
{
    printf("Hello,world!\n");
    return 0;
}
```

尽管上面这个程序非常简单，但它说明了 C 语言的一些重要特性。同时，这些程序行也包含了在屏幕上打印“Hello,world!”所需要的最基本的内容。下面分析这个程序中的每一行语句。

1) 程序的第一行是注释行：

```
//简单的应用程序 hello.c
```

注释是程序员为了提高程序的可读性和可理解性，在源程序的开始或中间对程序的功能、使用方法等所写的注解，可以用汉字或英文字符表示。注释仅用于阅读源程序，系统编译程序时，忽略其中的所有注释。

2) 程序的第二行：

```
#include <stdio.h>
```

其功能是进行有关预处理的操作，告诉编译器把标准输入/输出头文件“stdio.h”包括到这个程序中，该头文件中包含了在编译 printf 等标准输入/输出库函数时编译器需要使用的声明和信息。在程序编译之前，以#开始的行都会由预处理器来处理。

3) 程序的第三行：

```
int main()
```

这一行声明了一个函数，函数的名字为 main，返回值为整型。其中，int 是 C 语言中的关键字，这个关键字表示函数返回值的类型是整型。函数的内容从函数名后的左大括号“{”（第四行）开始，到与之匹配的右大括号“}”（第七行）为止。

一个完整的 C 语言程序，通常包含一个或多个函数，但必须包含一个 main() 函数，有且仅有一个。main() 函数是一个 C 语言程序执行的入口点，也就是说 C 语言程序都是从 main() 函数开始执行的。

4) 程序的第四行为一个左大括号“{”，在这里是函数开始的标志。

5) 程序的第五行：

```
printf("Hello,world!\n");
```

这一行是一条可执行语句，命令计算机去执行一个动作，也就是在屏幕上显示双引号内的“Hello,world!”字符。在 C 语言程序中，每条语句都必须使用分号“;”来结尾。

printf() 函数是实现输出的一个系统函数，向控制台屏幕按照指定的格式输出文字或符