

计算机公共课系列教材

C 语言程序设计

主 编 杨健甯 副主编 汪同庆



WUHAN UNIVERSITY PRESS

武汉大学出版社

计算机公共课系列教材

C 语言程序设计

主 编 杨健磊

副主编 汪同庆

参 编 周雅洁 黄 磊 杨麋丞

江苏工业学院图书馆
藏书章



WUHAN UNIVERSITY PRESS

武汉大学出版社

图书在版编目(CIP)数据

C 语言程序设计/杨健霖主编;汪同庆副主编. —武汉: 武汉大学出版社, 2006. 1

(计算机公共课系列教材)

ISBN 7-307-04939-2

I. C… II. ①杨… ②汪… III. C 语言—程序设计—高等学校—教材 IV. TP312

中国版本图书馆 CIP 数据核字(2006)第 005545 号

责任编辑: 黄金文 史新奎 责任校对: 程小宜 版式设计: 支笛

出版发行: 武汉大学出版社 (430072 武昌 珞珈山)

(电子邮件: wdp4@whu.edu.cn 网址: www.wdp.com.cn)

印刷: 湖北恒泰印务有限公司

开本: 787×980 1/16 印张: 17 字数: 350 千字

版次: 2006 年 1 月第 1 版 2006 年 1 月第 1 次印刷

ISBN 7-307-04939-2/TP·192 定价: 23.00 元

版权所有, 不得翻印; 凡购我社的图书, 如有缺页、倒页、脱页等质量问题, 请与当地图书销售部门联系调换。

计算机公共课系列教材

编 委 会

主 任:杨健露

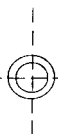
副 主 任:熊建强 李俊娥 殷 朴

编 委:(以姓氏笔画为序)

刘 英 何 宁 汪同庆 杨运伟

吴黎兵 罗云芳 黄文斌 康 卓

执行编委:黄金文 杨 华



内 容 简 介

C 语言是目前被人们广泛使用的程序设计语言之一,是许多计算机专业人员和计算机爱好者编制软件的强有力的开发工具,也是国内外大学普遍开设的计算机基础课程之一。本书详细介绍了运用 C 语言编制计算机程序的原理与方法,并穿插有大量实例,具有较强的启迪性,可使读者迅速掌握 C 语言程序设计技术。

此书可作为普通高等院校本、专科学生的教学用书,也可供计算机水平考试培训及一般工程技术人员自学使用。



前言

C 语言使用方便、功能强大、移植性好,它能产生高效率的目标代码,兼具高级语言和低级语言的优点,是一种优秀的结构化程序设计语言。C 语言作为一种既适合于开发系统软件又适合于开发应用软件的语言,已经成为计算机程序设计语言的主流语种之一,得到广泛的认可。

二十多年来,除了计算机专业人员外,其他行业的广大计算机应用人员也喜欢使用 C 语言。全国计算机等级考试、全国计算机应用技术证书考试、全国计算机软件著作权及水平考试等都将 C 语言一定程度地纳入了考试范围。随着 C 语言在国内普及、推广、应用的需要,全国许多高校已不仅对计算机专业的学生,而且对广大非计算机专业的学生也相继开设了 C 语言程序设计课程。此外,成人教育、函授教育等同样广泛开设了 C 语言程序设计课程。

C 语言与其他高级语言相比更复杂一些。这是由于它规则较多,涵盖的知识面更广,尤其涉及一些机器及环境方面的实现细节,使用灵活,难点较多,容易出错,初学者不易掌握。

本教材的对象主要为广大的本科生、专科生。其特点有:

1. 本着不苛求读者具备太多计算机专门知识也能学好 C 语言的愿望,尽量做到叙述通俗易懂,一方面要有利于组织教学,另一方面又要有利于自学。
2. 学习的目的在于应用。通过学习,读者应该能做到自己动手编写程序来解决问题。本教材强调了算法在编程中的重要性,同时也希望通过学习,读者能养成良好的编程习惯和风格。
3. 知识的积累有一定的过程,循序渐进是必要的,帮助读者建立正确、清晰的概念是本教材的主要任务。
4. 章节的安排尽量做到结构合理,难点和重点突出。既要说明问题,又不能过于烦琐,让人产生畏难情绪。为此,本教材提供了适量的例子程序。

本教材以标准 C 语言为基础,共包括十二章以及必要的附录。第一章介绍了 C 语言的发展、特点和程序结构,阐述程序设计的基本方法、算法的概念及其表示,最后介绍了 C 语言应用程序的开发过程;第二章介绍了 C 语言的字符集和保留字、C 语言的基本数据类型、常量和变量、基本运算符与表达式,阐述变量的基本概念和类型说明、基本运算符的优先级和结合性以及数据类型转换等内容;第三章介绍了 C 语言语句的 5 大分类,详细讲解赋值语句及最基本的输入输出函数 `printf()`, `scanf()`,



putchar(), getchar() 等的基本格式及使用时注意的要点;第四章介绍了用 C 语言实现选择结构所使用的两种语句:if 条件语句和 switch 多分支选择语句;第五章介绍了用 C 语言实现循环结构所使用的 3 种语句:while 语句、do-while 语句、for 语句,最后介绍了多重循环的使用、跳出循环和 goto 语句;第六章介绍了 C 语言中构造类型数据中的一种:数组,下标运算符的使用、一维、二维以及字符数组的定义、初始化和引用、字符串数据的处理和字符串处理函数;第七章介绍了 C 语言中函数的概念和定义、参数传递、函数调用,包括函数的嵌套调用、递归调用等关于函数的基本知识以及变量的作用范围、变量的存储类别以及内、外部函数等基本概念;第八章介绍了 C 语言中最复杂但最重要的部分:指针,涉及指针的概念、指针变量的定义及引用方式、指针变量的运算、利用指针变量构成复杂的数据类型以及指针变量的典型应用等;第九章介绍了 C 语言中另两种构造数据类型:结构体和共用体。结构体类型的定义、结构体类型变量的定义、初始化和引用、结构体成员运算符的使用、结构体数组、链表、共用体类型的定义、共用体类型变量的定义和引用以及自定义数据类型的应用;第十章介绍了 C 语言中编译预处理的概念和常用的编译预处理命令、宏定义、文件包含以及条件编译等;第十一章介绍了 C 语言中有关位运算的功能,位运算是 C 语言区别于其他高级语言的一大特色,主要介绍位运算的概念和几种常用的位运算,同时还介绍了有关“位段”的概念;第十二章介绍了 C 语言中有关文件操作的基本知识,即文件与“流”的基本概念、与文件有关的数据缓冲区、文件类型指针、文件的打开与关闭、文件的存取(包括字符读写函数、格式化读写函数、成块数据的读写函数)以及文件定位等。

本教材第一、第二章由汪同庆编写,第三至第五章由黄磊编写,第六、第七章由杨健露编写,第八、第九章由杨麋丞编写,第十至第十二章由周雅洁编写。在编写过程中,得到武汉大学教务部、武汉大学计算中心和武汉大学出版社领导的大力支持,许多老师给予了帮助,提出了宝贵意见,在此表示衷心的感谢。

作为课堂教学的补充,我们同时组织了《C 语言程序设计实验教程》、《C 语言程序设计习题集》作为配套教材。

由于计算机技术发展迅速以及我们水平有限,加之时间紧迫,书中定会存在不少错误和遗漏,恳请得到广大读者的批评指正。

编 者

2005 年 12 月

目 录

第一章 C 语言的发展、特点与程序结构	1
1.1 C 语言的发展与特点	1
1.1.1 C 语言的发展	1
1.1.2 C 语言的特点	2
1.2 程序设计的基本方法	3
1.2.1 计算机程序	3
1.2.2 面向过程的程序设计	4
1.2.3 面向对象的程序设计	4
1.3 算法及其表示	5
1.3.1 算法的概念	5
1.3.2 算法的表示	6
1.4 C 语言程序的基本结构	9
1.4.1 简单的 C 语言程序介绍	9
1.4.2 C 语言程序基本结构	11
1.5 应用程序的开发过程	12
第二章 数据类型、运算符和表达式	15
2.1 C 语言的字符集和保留字	15
2.1.1 C 语言的字符集	15
2.1.2 C 语言的保留字	16
2.1.3 C 语言的词法约定	16
2.2 数据类型	18
2.2.1 C 语言的数据类型	18
2.2.2 C 语言的基本数据类型	19
2.2.3 不同数据类型间的转换与运算	20
2.3 常量、变量和标准函数	22
2.3.1 常量	22
2.3.2 变量	27



2.3.3 标准函数	32
2.4 基本运算符及其表达式	33
2.4.1 算术运算符与算术表达式	34
2.4.2 关系运算符与关系表达式	38
2.4.3 逻辑运算符与逻辑表达式	40
2.4.4 条件运算符与条件表达式	42
2.4.5 赋值运算符与赋值表达式	43
2.4.6 逗号运算符与逗号表达式	46
第三章 基本语句与顺序结构程序设计	47
3.1 C 语言程序的基本语句	47
3.1.1 表达式语句	47
3.1.2 函数调用语句	48
3.1.3 控制语句	48
3.1.4 复合语句	49
3.1.5 空语句	49
3.2 赋值语句	50
3.3 数据的输入输出	51
3.3.1 格式输入输出函数 (printf 函数和 scanf 函数)	51
3.3.2 字符输入输出函数 (putchar 函数和 getchar 函数)	61
第四章 选择结构程序设计	63
4.1 用 if 语句实现选择结构	63
4.1.1 if 语句的三种形式	63
4.1.2 if 语句的嵌套	70
4.2 用 switch 语句实现多分支选择结构	75
4.3 用条件表达式实现选择结构	79
第五章 循环结构程序设计	81
5.1 用 while 语句实现循环结构	81
5.2 用 do-while 语句实现循环结构	85
5.3 用 for 语句实现循环结构	88
5.4 continue 语句和 break 语句	92
5.4.1 continue 语句	92
5.4.2 break 语句	93
5.5 循环的嵌套	95



5.5.1 循环的嵌套	95
5.5.2 程序举例	98
5.5.3 循环在数值计算中的应用	103
5.6 goto 语句和语句标号的使用	106
第六章 数 组	108
6.1 一维数组	108
6.1.1 一维数组的定义、初始化和引用	108
6.1.2 一维数组的输入输出	111
6.1.3 一维数组应用实例	112
6.2 二维数组	115
6.2.1 二维数组的定义、初始化和引用	115
6.2.2 二维数组的输入输出	117
6.2.3 二维数组应用实例	118
6.3 字符数组	120
6.3.1 字符数组的定义、初始化和引用	120
6.3.2 字符数组的输入输出	122
6.3.3 字符串处理函数	123
6.3.4 字符数组应用实例	126
第七章 函 数	129
7.1 函数的概述	129
7.1.1 C 程序的模块化结构	130
7.1.2 函数的分类与定义	131
7.1.3 函数调用的一般形式	132
7.1.4 函数的参数	134
7.1.5 函数的返回值	136
7.1.6 对被调用函数的说明	138
7.1.7 数组作为函数参数	138
7.2 函数的嵌套调用和递归调用	140
7.2.1 函数的嵌套调用	140
7.2.2 函数的递归调用	142
7.3 变量的作用域和生存期	145
7.3.1 变量的作用域	145
7.3.2 变量的存储类别	147
7.4 内部函数和外部函数	151



7.4.1 内部函数	151
7.4.2 外部函数	152
7.5 综合应用举例(一)	152
第八章 指 针	155
8.1 指针和指针变量的概念	155
8.1.1 地址和指针	155
8.1.2 指针变量	156
8.2 变量的指针和指向变量的指针变量	156
8.2.1 指针变量的定义	157
8.2.2 指针变量的引用	157
8.2.3 指针变量的初始化	158
8.2.4 指针变量的运算	159
8.2.5 指针变量作为函数参数	161
8.3 数组的指针和指向数组的指针变量	162
8.3.1 一维数组的指针和指向一维数组元素的指针变量	162
8.3.2 二维数组的指针和指向二维数组的指针变量	165
8.3.3 数组名作为函数参数	169
8.4 字符串的指针和指向字符串的指针变量	171
8.4.1 字符串指针变量的定义与引用	171
8.4.2 字符串指针变量与字符数组	172
8.4.3 字符串指针变量作为函数参数	173
8.5 指针数组和指向指针的指针	174
8.5.1 指针数组	174
8.5.2 指向指针的指针	176
8.6 指向函数的指针变量和返回指针的函数	178
8.6.1 函数的指针	178
8.6.2 指向函数的指针变量的定义与引用	178
8.6.3 指向函数的指针变量作为函数参数	180
8.6.4 返回指针的函数	183
第九章 结构体和共用体	185
9.1 结构体	185
9.1.1 结构体类型的定义	185
9.1.2 结构体类型变量的定义	186
9.1.3 结构体类型变量的初始化和引用	189



9.1.4 结构体数组	191
9.1.5 指向结构体类型变量的指针变量	194
9.2 链表	196
9.2.1 链表的概念	196
9.2.2 用指针和结构体实现链表	196
9.2.3 对单向链表的操作	198
9.3 共用体	202
9.3.1 共用体类型的定义	202
9.3.2 共用体类型变量的定义	203
9.3.3 共用体类型变量的引用	205
9.4 自定义数据类型	208
9.5 综合应用举例(二)	210
第十章 编译预处理	216
10.1 编译预处理的概念	216
10.2 宏定义	216
10.2.1 不带参数的宏定义	216
10.2.2 带参数的宏定义	219
10.3 文件包含	220
10.4 条件编译	221
第十一章 位 运 算	223
11.1 位运算的概念	223
11.2 位运算符的含义及其使用	224
11.2.1 按位“与”运算(&)	224
11.2.2 按位“或”运算()	224
11.2.3 按位“非”运算(~)	225
11.2.4 按位“异或”运算(^)	225
11.2.5 “左移”运算(<<)	225
11.2.6 “右移”运算(>>)	226
11.2.7 长度不同的两个数进行位运算的运算规则	227
11.2.8 位复合赋值运算符	227
11.3 位段	227
第十二章 文 件	231
12.1 文件与文件类型指针	231



12.1.1 文件	231
12.1.2 文件数据的存储形式	232
12.1.3 文件的处理方法	232
12.2 文件的打开与关闭	233
12.2.1 文件的打开	233
12.2.2 文件的关闭	235
12.3 文件的存取	236
12.3.1 字符读写(函数 fgetc()和函数 fputc())	236
12.3.2 字符串读写(函数 fgets()和函数 fputs())	238
12.3.3 格式读写(函数 fscanf()和函数 fprintf())	240
12.3.4 数据块读写(函数 fread()和函数 fwrite())	240
12.4 文件的定位	241
12.4.1 函数 rewind()	242
12.4.2 函数 fseek()	242
12.4.3 ftell 函数	243
12.5 综合应用举例(三)	243
 附录 I ASC II 码表	 249
附录 II C 语言中的保留字	251
附录 III 运算符、优先级和结合性	253
附录 IV C 语言常用库函数	255
参考文献	260

第一章 C 语言的发展、特点与程序结构

本章主要介绍了 C 语言的发展、特点和程序结构,重点阐述了程序设计的基本方法,包括面向过程的程序设计和面向对象的程序设计;算法的概念及其表示,包括用自然语言、流程图、伪代码、N-S 流程图的表示方法。最后介绍了 C 语言应用程序的开发过程。

1.1 C 语言的发展与特点

1.1.1 C 语言的发展

C 语言是一种广泛流行且深受用户青睐的高级程序设计语言。它适用于开发系统软件(如操作系统、编译程序、汇编程序、数据库管理系统等),也适用于开发应用软件(如数值计算、文字处理、控制系统、游戏程序等),目前已经成为计算机程序设计语言的主流语种。

C 语言是在 B 语言基础上发展起来的,与 PASCAL 语言一并同属于 ALGOL(Algorithmic Language)语言族系。

1960 年 ALGOL60 问世,这是一种适用于科学与工程计算的高级语言。这种语言不能操作硬件,不适合编写计算机系统程序;而汇编语言能充分体现计算机硬件的指令级特性,形成的代码也有较高的质量,但它的可读性、可移植性以及描述问题的性能都远不及高级语言。能否开创一种既有汇编语言特性,又有高级语言功能的计算机语言呢?C 语言就是在此背景下诞生的。

1963 年英国剑桥大学推出了 CPL(Combined Programming Language)语言,这种语言虽然可以操作硬件,但系统规模较大,难以实现。1967 年英国剑桥大学的 M. Richards 对 CPL 语言进行了优化,推出了 BCPL(Basic Combined Programming Language)语言。BCPL 语言只是 CPL 语言的改良版,使用起来仍有很大的局限性。1970 年美国 Bell 实验室的 K. Thompson 在 BCPL 语言基础上,对 BCPL 语言进行了进一步的简化,设计出了很接近硬件的 B 语言,并用 B 语言编写了第一个 UNIX 操作系统。“B 语言”的意思是将 CPL 语言煮干,提炼出它的精华。1973 年,B 语言也给人“煮”了一下,美国 Bell 实验室的 D. M. Ritchie 在 B 语言的基础上最终设计出了一种新的语言,他使用了 BCPL 的第二个字母作为这种语言的名字,这就是 C 语言。



C 语言最初用于 PDP-11 计算机上的 UNIX 操作系统。1973 年 D. M. Ritchie 和 K. Thompson 合作将 UNIX 操作系统用 C 语言改写了一遍(即 UNIX 第 5 版),把 UNIX 推进到一个新阶段。以后的 UNIX 第 6 版、第 7 版以及 System III 和 System V 都是在 UNIX 第 5 版的基础上发展起来的。随着 UNIX 操作系统的日益广泛使用,C 语言也得到了迅速的发展。1977 年,出现了不依赖于具体机器的 C 语言编译文本,继而出现了各种不同版本的 C 语言。

1983 年,美国国家标准化协会(ANSI—American National Standards Institute)对已出现的各种 C 语言版本进行了扩充,制定了一套 C 语言的标准,称为 ANSI C。1987 年 ANSI 又推出了新的 C 语言标准,即 87 ANSI C。目前流行的各种 C 语言编译系统都是以 87 ANSI C 为基础设计的,如 Microsoft C, Turbo C, Borland C, Quick C 和 AT&T C 等。

本书采用 Borland C++3.1 的 DOS 版(简称 BC31)作为 C 语言程序设计的编译环境。BC31 是 Borland 公司继 Turbo C 2.0 后的更新产品,它吸取了以往产品中几乎所有的优点,而且能支持鼠标和多窗口操作。

1.1.2 C 语言的特点

C 语言之所以广泛流行且深受用户青睐,是因为它以其简洁、灵活、表达能力强、产生的目标代码质量高和可移植性好等为基本特色,归纳起来有以下主要特点:

1. C 语言是结构化程序设计语言

C 语言提供了结构化程序所必需的基本控制语句,如条件判断语句和循环语句等,实现了对逻辑流的有效控制。C 语言的源程序由函数组成,每个函数各自独立,把函数作为模块化设计的基本单位。C 语言的源文件可以分割成多个源程序,进行单独编译后可连接生成可执行文件,为开发大型软件提供了极大的方便。C 语言提供了多种存储属性,通过对数据的存储域控制提高了程序的可靠性。

2. 具有丰富的数据类型

C 语言除提供整型、实型、字符型等基本数据类型外,还提供了用基本数据类型构造出的各种复杂的数据结构,如数组、结构、联合等。C 语言还提供了与地址密切相关的指针类型。此外,用户还可以根据需要自定义数据类型。

3. 具有丰富的运算符

C 语言提供了多达 44 种运算符,运算能力十分丰富,它把括号、逗号、赋值等都作为运算符来处理。丰富的数据类型与丰富的运算符相结合,使 C 语言的表达力更具灵活性,同时也提高了执行效率。



4. C 语言结构紧凑,使用方便、灵活

C 语言只有 32 个保留字(关键字),9 种控制语句,大量的标准库函数可供直接调用;C 语言程序书写形式自由,有些表达式可以用简洁式书写,源程序简练,大大提高了程序设计的效率和质量。

5. C 语言具有低级语言的功能

C 语言既具有高级语言面向用户、可读性强、容易编程和维护等特点,又具有汇编语言面向硬件和系统的特点,提供了对位、字节和地址等直接访问硬件的操作。所以也可以说,C 语言是高级语言中的低级语言。

6. C 程序移植性好

C 语言具有执行效率高、程序可移植等特点。C 语言程序本身独立于机器硬件,可以在多种环境下运行,从而形成了多种版本的 C 语言。

由于 C 语言具有以上诸多特点,因此从 C 语言产生到现在一直长盛不衰,迅速得到广泛的应用和普及。特别是在微型计算机系统的软件开发和各种软件工具的研制中,使用 C 语言的趋势与日俱增,呈现可能取代汇编语言的发展趋势。

1.2 程序设计的基本方法

1.2.1 计算机程序

为了利用计算机来处理问题,必须编写使计算机能按人的意愿工作的程序。所谓程序,就是计算机解决问题所需的一系列代码化指令、符号化指令或符号化语句。著名的计算机科学家沃思(Wirth)提出了一个著名的公式来表达程序的实质:

$$\text{程序} = \text{数据结构} + \text{算法}$$

即“程序就是在数据的某些特定的表示方式和结构的基础上,对抽象算法的具体描述”。当然,在实际编写计算机程序时,要遵循程序设计方法,在运行程序时要有软件环境的支持。因此,可以将上述公式扩充为:

$$\text{程序} = \text{数据结构} + \text{算法} + \text{程序设计方法} + \text{语言工具}$$

即一个应用程序体现了四个方面的成分:采用的描述和存储数据的数据结构,采用的解决问题的算法,采用的程序设计的方法和采用的语言工具和编程环境。

在学习利用计算机语言编写程序时要掌握三个基本概念。一是语言的语法规则,包括常量、变量、运算符、表达式、函数和语句的使用规则;二是语义规则,包括单词和符号的含义及其使用规则;三是语用规则,即善于利用语法规则和语义规则正确组织程序的技能,使程序结构精练、执行效率高。



此外,要弄清“语言”和“程序”的关系。语言是构成程序的指令集合及其规则,程序是用语言为实现某一算法组成的指令序列。学计算机语言是为了掌握编程工具,它本身不是目的。当然,脱离具体的语言去学习编程是十分困难的,因为两者有密切的联系。

1.2.2 面向过程的程序设计

所谓面向过程的程序设计,是指利用面向过程的语言工具(如 Basic、Pascal、Fortran 和 C 语言等)进行程序开发中的各项活动。

1969 年,荷兰学者 E. W. Dijkstra 对面向过程的程序设计语言提出了结构化的程序设计思想,规定一个结构化程序由顺序结构、选择(分支)结构和循环控制结构三种基本结构组成。同时规定了三种基本结构之间可以并列和互相包含,不允许交叉和从一个结构直接转到另一个结构的内部。这就是面向过程的程序,也称结构化程序。

使用面向过程的程序设计语言编写结构化程序,其基本方法是:把一个需要求解的复杂问题分为若干个子问题来处理,每个子问题控制在一个可调试或可操作的范围(或模块)内,设计时遵循自顶向下、逐步细化、模块化设计和结构化编码的原则。

“自顶向下”就是将整个待解决的问题按步骤、有次序地进行分层,明确先做什么,再做什么,各层包含什么内容。

“逐步细化”就是对分层后的每一层功能进行详细设计,并仔细检验其算法的正确性。只有当本层功能及其算法正确无误之后,才能向下一层细化。如果每一层的设计都没有问题,则整个程序功能及其算法就是正确的。

“模块化设计”就是将处理问题的整个程序分为多个模块,其中有一个主模块和若干个子模块,由主模块控制各个处理子问题的子模块,最终实现整个程序的功能。模块化设计的思想是一种“分而治之”的思想,即把一个复杂的问题分为若干个子问题来处理,这样就简单多了,也便于程序的检验和调试。所谓模块是指一个能完成某项特定功能,既可以组合又可以分解的程序单元。

“结构化编码”是指在进行结构化程序设计之后,用结构化语言编写程序的过程。利用结构化语言编写程序是非常方便的。

结构化程序的主要优点是编程简单、结构性强、可读性好,执行时(除遇到特殊流程控制语句外)总是按事先设计的控制流程自顶而下顺序执行,时序特征明显。遵循这种结构的程序只有一个入口和一个出口。但结构化程序也存在缺点,如数据与程序模块的分离和程序的可重用性差等。

1.2.3 面向对象的程序设计

相对结构化程序设计而言,面向对象的程序设计是一个全新的概念。在面向对