

高等院校计算机系列教材



GAODENGYUANXIAO
JISUANJI
XILIEJIAOCAI



C语言程序设计

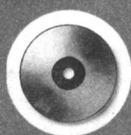
主 编: 曾碧卿 陈香兰

中南大学出版社

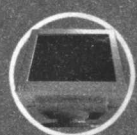
CYUANCHENGXU
SHEJI



高等院校计算机系列教材



GAODENGYUANXIAO
JISUANJI
XILIEJIAOCAI



C语言程序设计

主 编: 曾碧卿 陈香兰 中南大学出版社

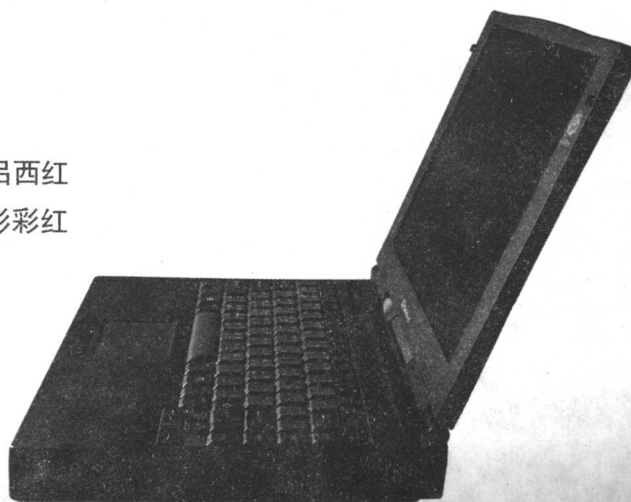
主 编: 曾碧卿 陈香兰

副主编: 孙德才 罗庆云 王新祥 吴湘华

编 委 (按姓氏笔画排序)

邓会敏 王晓霞 史长琼 伍友龙 吕西红

李远辉 陈 敏 罗丹霞 曹水莲 彭彩红



图书在版编目(CIP)数据

C 语言程序设计/曾碧卿,陈香兰主编. —长沙:中南大学出版社, 2006. 1

ISBN 7-81105-263-6

I. C... II. ①曾...②陈 III. C 语言—程序设计 IV. TP312

中国版本图书馆 CIP 数据核字(2005)第 160479 号

C 语言程序设计

主 编 曾碧卿 陈香兰

☐ 责任编辑 陈应征 谭晓萍

☐ 责任印制 文桂武

☐ 出版发行 中南大学出版社

社址:长沙市麓山南路

邮编:410083

发行科电话:0731-8876770

传真:0731-8710482

☐ 印 装 长沙市华中印刷厂

☐ 开 本 787 × 1092 1/16 ☐ 印张 26.25 ☐ 字数 662 千字

☐ 版 次 2006 年 1 月第 1 版 ☐ 2006 年 1 月第 1 次印刷

☐ 书 号 ISBN 7-81105-263-6/G · 098

☐ 定 价 36.00 元

图书出现印装问题,请与出版社调换

前言

“C 语言程序设计”是一门重要的专业基础课,它是所有理工科专业本科教学计划中的主要必修课程之一,在人才培养中占有重要的地位和作用。对计算机及其相关专业而言,C 语言是许多后续专业课程的基础,因此,在本专科生教学体系中,该门课程的教学仅是起点,而非终点。

通过本教材的学习,旨在使学生掌握 C 语言的基本语法、基本语句、基本控制结构以及自顶向下结构化程序设计的基本方法,使学生认识到算法、良好的程序设计风格以及实践在本课程学习中的重要性,培养学生熟练使用 C 语言编程分析和解决实际问题的能力。该课程为学生进一步学习其他专业课程和今后从事软件开发工作打下坚实的基础。本书在编写过程中,以应用为背景,以知识为主线,以提高能力和兴趣为目的,变应试为应用,从而培养大学生在专业方面的综合素质。

本着面向未来的精神,本教材的教学宗旨是要努力把 C 语言从应试课程转变为一种实践工具,主要目的不再局限于使学生单纯地了解 and 掌握 C 语言的基本语法规则,而是要致力于培养学生运用 C 语言解决实际问题的编程能力,以 C 语言为工具,介绍程序设计的基本思想和方法,培养学生无论以后在学习、工作中使用什么语言编程,都能灵活应用这些思想和方法的能力。

本书共分为三篇,第一篇介绍 C 程序设计的理论及应用,第二篇给出了 C 程序设计的 18 个实验,第三篇则给出了 5 个课程设计题目。第一篇共分 10 章,主要介绍了 C 语言的基本概念、理论及应用。全书按以下方式进行组织:第 1 章 C 语言基础,主要是介绍 C 语言的发展、特点及应用,简单的 C 程序、C 语言的数据类型、常量、变量、运算符及表达式等;第 2 章讲述简单的 C 程序设计,包括赋值语句、算法、结构化程序设计方法、数据的基本输入输出及顺序结构程序设计;第 3 章主要讲述选择结构程序设计,包括关系运算符及表达式、逻辑运算符及表达式、if 语句、条件运算符及表达式、switch 语句等;第 4 章讲述循环结构程序设计,包括 while、for、break 及 continue 语句等;第 5 章讲述数组,包括一维数组、二维数组、多维数组及字符数组等;第 6 章阐述了 C 语言中的函数,包括了函数的定义、调用、变量的作用域、函数与数组、变量的存储类型、内外函数以及多文件程序等;第 7 章阐述了指针,主要内容有:指针与地址、指针变量的定义与使用、指针数组、指针与函数等;第 8 章讲述结构体、共用体与枚举类型,包括有结构体的定义、结构体变量的使用、结构体数组、结构体指针、结构体函数、共用体、枚举类型以及用户定义类型等;第 9 章介绍了 C 语言高级程序设计,包括编译预处理、位运算以及链表等应用;第 10 章介绍了 C 语言文件方面的内容,包括缓冲文件系统和非缓冲文件系统文件的基本操作等内容。本书基本内容的课堂讲授和实验教学的建议学时数为 80~90 学时。教材中带“*”为较高要求部分,带“**”为较难掌握部分,请教师根据实际情况取舍。

本书由曾碧卿博士、陈香兰教授主编,孙德才、罗庆云、王新祥和吴湘华任副主编,陈敏、王晓霞、罗丹霞、曹水莲、彭彩红、史长琼、邓会敏、伍友龙、吕西红、李远辉参加编写,

全书最后由曾碧卿博士修改统稿。

中南大学出版社陈应征和谭晓萍两位编辑对于本书的编写和出版提供了很多的指导和帮助,在此表示真诚的感谢!

由于作者水平有限,缺点和错误在所难免,恳请读者批评指正。作者的电子邮箱: zengbiqing0528@163.com。

本书配有 PPT 电子课件,需要使用的教师请与主编联系。

谢谢阅读本书的读者!

曾碧卿

2005 年 12 月于长沙

目 录

第一篇 C 程序设计理论与应用部分

第 1 章 C 语言基础	(3)
1.1 C 语言简介	(3)
1.1.1 几种常用语言的简单介绍	(4)
1.1.2 C 语言的发展简介	(4)
1.1.3 C 语言的特点和应用	(5)
1.2 简单的 C 语言程序解析	(6)
1.2.1 C 语言程序的组成规则	(7)
1.2.2 C 语言程序的基本符号	(7)
1.3 C 语言数据类型	(9)
1.3.1 整型数据	(10)
1.3.2 实型数据	(10)
1.3.3 字符型数据	(11)
1.3.4 枚举型数据	(11)
1.4 常量与变量	(11)
1.4.1 常量	(11)
1.4.2 变量	(14)
1.5 运算符和表达式	(20)
1.5.1 算术运算符和表达式	(20)
1.5.2 赋值运算符和赋值表达式	(21)
1.5.3 逗号运算符和逗号表达式	(24)
1.5.4 类型转换	(24)
本章小结	(27)
习 题	(27)
第 2 章 简单 C 程序设计	(31)
2.1 C 语句概述	(31)
2.2 赋值语句	(33)
2.3 算法	(33)
2.3.1 算法的概念、特征及组成要素	(34)

2.3.2 算法的描述	(35)
2.4 结构化程序设计方法	(38)
2.5 数据的输入输出	(40)
2.5.1 格式化的输入与输出	(40)
2.5.2 字符数据的输入输出	(49)
2.6 顺序结构程序设计	(51)
本章小结	(53)
习题	(54)
第3章 选择结构程序设计	(55)
3.1 关系运算符和关系表达式	(55)
3.1.1 关系运算符	(55)
3.1.2 关系表达式	(55)
3.2 逻辑运算符与逻辑表达式	(56)
3.2.1 逻辑运算符	(56)
3.2.2 逻辑表达式	(57)
3.3 if 语句	(58)
3.3.1 if 语句的格式	(59)
3.3.2 if 语句的嵌套与嵌套匹配原则	(62)
3.4 条件运算符和条件表达式	(64)
3.5 switch 语句	(66)
3.6 程序举例	(69)
本章小结	(74)
习题	(75)
第4章 循环结构程序设计	(76)
4.1 循环概述	(76)
4.2 用 goto 语句和 if 语句构成循环	(77)
4.3 while 语句	(79)
4.4 do-while 语句	(82)
4.5 for 语句	(84)
4.5.1 for 语句的一般形式	(84)
4.5.2 for 循环的变量	(85)
4.5.3 无限循环	(86)
4.5.4 无循环体 for 循环	(86)
4.6 循环的嵌套	(89)
4.7 几种循环的比较	(92)
4.8 break 和 continue 语句	(92)
4.8.1 break 语句	(92)

4.8.2 continue 语句	(93)
4.9 程序举例	(94)
本章小结	(102)
习题	(102)
第5章 数 组	(108)
5.1 一维数组	(108)
5.1.1 一维数组的声明	(108)
5.1.2 一维数组元素的引用	(110)
5.1.3 一维数组的赋值	(111)
5.1.4 一维数组越界检查	(112)
5.1.5 一维数组的应用举例	(114)
5.2 二维数组与多维数组	(119)
5.2.1 二维数组的声明与初始化	(119)
5.2.2 二维数组元素的表示方法	(121)
5.2.3 多维数组	(123)
5.2.4 二维数组的应用举例	(124)
5.3 字符与数组	(128)
5.3.1 字符型数据	(128)
5.3.2 一维字符数组的声明及初始化	(129)
5.3.3 一维字符数组的输入与输出	(130)
5.3.4 二维字符数组	(132)
5.3.5 字符串处理函数	(133)
5.3.6 字符数组程序举例	(136)
本章小结	(138)
习题	(138)
第6章 函 数	(140)
6.1 函数概述	(140)
6.1.1 模块化程序设计	(140)
6.1.2 函数的分类	(142)
6.1.3 C 语言的库函数	(143)
6.2 函数的定义	(143)
6.3 函数的调用	(145)
6.3.1 函数的简单调用	(146)
6.3.2 函数的嵌套调用	(151)
*6.3.3 函数的递归调用	(155)
6.4 变量的作用域	(158)
6.4.1 局部变量	(159)

6.4.2 全局变量	(160)
6.4.3 局部变量与全局变量	(162)
6.5 函数与数组	(163)
6.5.1 数组元素作为函数的参数	(163)
6.5.2 数组名作为函数的参数	(164)
6.6 变量的存储类型	(169)
6.6.1 动态存储方式与静态存储方式	(169)
6.6.2 动态存储变量	(170)
6.6.3 静态存储变量	(171)
6.6.6 存储类型小结	(174)
**6.7 内外部函数与多文件程序的运行	(174)
6.7.1 内部函数和外部函数	(175)
6.7.2 多文件程序的运行	(176)
本章小结	(177)
习题	(178)

第7章 指 针

(180)

7.1 地址与指针	(180)
7.2 指针变量的定义和使用	(181)
7.2.1 指针变量的类型	(181)
7.2.2 指针变量的定义及其初始化	(182)
7.2.3 指针的使用	(184)
7.3 指针的算术运算和指针的比较	(189)
7.4 指针与数组	(192)
7.4.1 通过指针引用数组	(192)
7.4.2 数组名与指针	(197)
7.4.3 指向数组的指针变量	(200)
7.4.4 指针数组	(203)
7.5 指针与字符串	(216)
7.6 指向指针的指针	(221)
7.7 指针与函数	(225)
7.7.1 指针变量作为函数的参数	(225)
7.7.2 返回指针的函数	(241)
*7.7.3 指向函数的指针	(244)
7.8 指针应用中的一些问题	(248)
本章小结	(252)
习题	(253)

第8章 结构体、共用体与枚举类型	(255)
8.1 概述	(255)
8.2 结构体定义	(256)
8.2.1 结构体类型的定义	(256)
8.2.2 结构体变量定义	(257)
8.3 结构体变量的使用	(260)
8.3.1 结构体变量初始化	(260)
8.3.2 结构体变量引用	(261)
8.4 结构体数组	(263)
8.4.1 结构体数组定义与初始化	(263)
8.4.2 结构体数组引用	(265)
8.4.3 结构体数组使用实例	(266)
8.5 结构体与指针	(268)
8.5.1 指向结构体的指针	(268)
*8.5.2 指向结构体数组的指针	(270)
*8.6 结构体与函数	(272)
8.6.1 结构体变量作为函数的参数	(272)
8.6.2 结构体变量指针作为函数的参数	(274)
8.6.3 结构体变量指针应用实例	(276)
*8.7 共用体	(278)
8.7.1 共用体的概念和定义	(278)
8.7.2 共用体的引用	(279)
8.7.2 共用体的使用实例	(280)
*8.8 枚举类型	(282)
8.8.1 枚举类型的概念、定义及其功能	(282)
8.8.2 枚举类型的使用实例	(283)
8.9 用户定义类型	(284)
8.9.1 用户定义类型的概念与定义	(284)
8.9.2 用户定义类型的应用	(285)
本章小结	(286)
习题	(286)
第9章 C语言高级程序设计	(288)
9.1 编译预处理命令	(288)
9.1.1 宏	(288)
9.1.2 文件包含	(290)
9.1.3 条件编译	(292)
9.2 位运算	(294)

9.2.1	位运算和位运算符	(294)
9.2.2	位运算符的使用	(295)
9.2.3	位段	(300)
9.3	结构体高级应用——链表	(303)
9.3.1	链表和动态存储分配概述	(303)
9.3.2	单链表	(305)
9.3.3	遍历链表	(311)
9.3.4	双向链表	(312)
9.3.5	循环链表	(313)
9.3.6	链表应用实例	(313)
本章小结		(320)
习题		(322)
第10章	文 件	(326)
10.1	文件的概述	(326)
10.1.1	文件的概念	(326)
10.1.2	文件的分类	(326)
10.1.3	操作系统对文件的处理方式	(327)
10.2	缓冲文件系统文件的基本操作	(328)
10.2.1	文件类型指针	(328)
10.2.2	文件打开与关闭函数	(329)
10.2.3	文件读写函数	(331)
10.2.4	文件定位与文件检测	(339)
10.3	非缓冲文件系统文件的基本操作	(342)
10.3.1	文件的创建函数、打开函数和关闭函数	(342)
10.3.2	文件读/写函数	(343)
10.3.3	文件定位函数	(343)
本章小结		(344)
习题		(345)

第二篇 C 语言程序设计上机实验部分

预备实验:	Turbo C 集成开发环境	(349)
实验一	数据类型和表达式	(356)
实验二	顺序结构程序设计	(358)
实验三	选择结构程序设计	(360)
实验四	循环结构程序设计(一)	(362)
实验五	循环结构程序设计(二)	(364)
实验六	循环结构程序设计(三)	(366)

实验七 一维数组	(368)
实验八 二维数组	(370)
实验九 字符数组	(372)
实验十 函数定义及简单调用	(374)
实验十一 函数嵌套调用和递归调用	(376)
实验十二 函数与数组以及多文件程序编译	(378)
实验十三 简单指针	(381)
实验十四 指针、函数和数组	(384)
实验十五 结构体与共用体	(387)
实验十六 C语言高级程序设计: 链表	(389)
实验十七 C语言高级程序设计: 位运算	(390)
实验十八 文 件	(392)

第三篇 C 程序设计课程设计部分

课程设计一 编写万年历系统	(397)
课程设计二 五子棋游戏	(398)
课程设计三 纸牌算法	(401)
课程设计四 销售管理系统	(403)
课程设计五 学生成绩文件管理	(405)

第一篇

C 程序设计理论与应用部分

第1章 C语言基础

C语言是一种结构化的计算机程序设计语言。它既具有高级语言的特点,又具有低级语言的功能。它不仅适合于作为系统描述语言写系统软件,也可以写应用程序。C语言是当今国际上最流行的、最有发展前途的计算机高级语言。本章主要介绍什么叫C语言;C语言的发展历史;C语言的特点和应用;C语言基本符号与组成规则;C语言中使用的常量、变量、表达式及其数据类型的定义等知识。为学好后几章的程序设计打下良好的基础知识。

1.1 C语言简介

C语言作为一种适合于各领域、各层次的语言,其重要性已经达到了不能用语言描述的地步了。它之所以能被广泛地应用,是因为它集计算机硬件与软件开发于一身,这是以前的高级语言所不能达到的。它既有高级语言的可读性,又有低级汇编语言的软件开发功能,所以受到人们的喜爱,它适合编写系统软件,它也是一种编译语言。C语言是一种得到广泛重视并普遍应用的计算机程序设计语言,也是国际上公认的最重要的几种通用程序设计语言之一。它主要用来编写:

1. 系统软件(操作系统、编译系统等。与C语言同时出名的多用户操作系统UNIX现在都是用C语言程序编制的)。
2. 应用软件(C语言是工业控制单片机的开发语言之一,并能进行图形处理)。
3. 数据处理软件(如企业管理)。
4. 数值计算等应用于各个领域的软件。

C语言基本能做到“只要人能想到就能实现”,所以C语言的最大限制不再是使用语言环境而是人的思维。C语言有很强的可读性、可移植性,可将其他语言程序很容易改成C语言的程序。

尽管当初C语言是为编写UNIX操作系统而设计的,但却不依赖于UNIX操作系统,可在多种操作系统的环境下运行,从普通的C语言到面向对象的C++(它的变种为JAVA)以及可视C语言(VISUAL C)都是针对软件开发要求而产生和发展的。虽然这个发展仍在继续,但C语言的基本功能不变,所以学习了C语言之后再学C++语言、JAVA、VC语言就很容易了。

C语言不像其他语言那样容易,在深层次应用上,要涉及到许多相关的知识,例如:数据结构、软件工程、数据库系统、人工智能、操作系统等。当学到函数、数组、指针、结构体时,就会有所体会。

自从1946年世界上第一台计算机诞生以来,计算机使得科学技术发生了翻天覆地的变化,计算机已作为一种文化渗透到世界每一个角落。其中作为使用计算机的工具语言对世界影响较大的有三种,它们分别是:汇编语言(机器语言)、数据库及C语言。

1.1.1 几种常用语言的简单介绍

早期的操作系统等系统软件，主要是用汇编语言编写的，它依赖于计算机硬件，程序的可读性和可移植性都很差。若用高级语言来编写，又难以实现汇编语言能直接对硬件进行操作的某些功能。为此，人们开始寻找一种既具有高级语言特性，又具有低级语言特性的语言。20 世纪 60 年代，计算机及其应用的飞速发展，使得程序量越来越大。而当时，汇编语言太难于编程，FORTRAN 语言又太不精确，使得程序可靠性的验证无法实现，出现了软件危机。当时给人们的印象是：软件越来越大，错误越来越多，交付日期越来越长，软件越来越不可靠。1964 年提出了“计算机软件”的概念。1968 年提出了“软件工程”的概念。中心思想是如何研制大型软件，提高软件生产率，使之具有较高的可靠性和可维护性。这时，人们自然想到一个优秀的、但又不太合格的工具语言 ALOGL60，对它进行改造的工作也就顺理成章开始了。高级语言是面向数学公式的，低级语言是面向机器硬件的。下面介绍几种常用语言的功能，如：机器语言、数据库管理系统、FORTRAN、COBOL、BASIC、ALGOL60 及 C 等。

机器语言：所谓机器语言，就是一种计算机的指令集合。

Zilog 公司的 Z80 微处理器(1975 年)，Motorola 公司的 M6502 及以其为 CPU 的 APPLE II 计算机；Intel 公司的 8088 及以其为 CPU 的 IBM-PC 计算机，8088 及 80386 成为现在流行的微型计算机及汇编语言的主要教学和应用语言。

数据库管理系统：最成功的要属 1986 年美国 Fox Software 公司推出的 FoxBASE+。其特点是：速度快，用户接口好，特别适合于各种信息管理，如财务管理、物资管理、档案管理、图书资料检索与管理。现在常用的有 FoxBASE、FoxPro、VF 等兼容系列。

FORTRAN：数值计算语言(Formula Translation 公式翻译)。从 1954 年提出，曾在 20 世纪 60~90 年代初在全世界得到最广泛的应用。特点是：执行效率较高，标准化程度高，程序结构灵活，但结构不精确。进入 90 年代由 C 及 BASIC 语言取代。

COBOL：通用商务语言(Common Business Oriented Language)。1960 年由美国推出。主要用于商务处理、企业管理等方面。由于编程规则众多繁琐、程序代码较长、译成机器语言后效率低等缺点。当进入 90 年代后，COBOL 所工作的领域被 C 语言及数据库系统所取代。

BASIC：初学者通用符号指令代码(Beginner's All-purpose Symbolic Instruction Code)。1969 年提出，1971 年形成标准版本。是由美国 Dartmouth 学院 John Kemeny 和 Thomas Kurts 两位教授创立的。语言简练，容易掌握。有较强的会话功能和程序的编辑修改、调试功能。既能进行科学计算，又能进行非数值计算。开发语言规模小，功能多但每一项都不是特别出色。

ALGOL60：算法语言(Algorithmic Language)。该语言在 1958 年提出，1960 年确定标准。它是经典(精确)语言中成功的一个。但由于过于精密，执行效率低于 40%，而 C 语言仅低于 10%。在 20 世纪 60~90 年代，一直是计算机系统的教学语言。直到 C 语言出现并广泛应用后，C 语言完全取代了它。

1.1.2 C 语言的发展简介

C 语言是由贝尔实验室的 Dennis Ritchie 和 Brian Kernighan 在 1972 年根据 Thompson 的 B 语言设计的，而 B 语言又是由一种早期的编程语言 BCPA(Basic Combined Programming Lan-

guage)发展演变而来的。BCPL的根源可以追溯到1960年的ALGOL 60 (Algol Programming Language), ALGOL 60是一种面向问题的高级语言,距离硬件较远。1963年英国剑桥大学推出CPL(Combined Programming Language)语言。CPL修改了ALGOL 60,使其能够直接做较低层的操作。1967年英国剑桥大学的Martin Richards对CPL做了改进,推出了BCPL语言。

具体各种语言产生年代以及改进功能等见下表1-1。

表1-1 各种语言产生年代以及改进功能

年代	语言名称	改进功能	设计人或机构
1960	ALGOL	面向问题(远离硬件)	
1963	CPL	部分驱动硬件	英国剑桥大学
1967	BCPL	直接驱动硬件,但太简单	英国剑桥大学
1970	B	接近硬件,写出了UNIX	美国贝尔实验室
1972	C	直接驱动硬件,功能强大	美国贝尔实验室
	C++	面向对象 VC++ 可视化编程	

1.1.3 C语言的特点和应用

任何一种语言都有它的特点和独特之处,这是它能够生存和不断发展的前提。

1. C语言的主要特点

(1) 语言简洁、紧凑、使用方便、灵活(32个关键字、9种控制语句)。C语言程序书写格式自由。没有行号,一行可以写几个语句,一个语句可以写在多行上。

(2) 运算符十分丰富(34个)。C语言把括号、赋值号、强制类型转换等符号都作为运算符处理。通过灵活使用多种运算符来达到其他高级语言中难以实现的运算功能。

(3) 数据结构丰富多彩。C语言除了具有其他高级语言所包含的各种数据结构以外,还包含指针类型、结构体类型、共用体类型数据,以此来实现比较复杂的数据结构(如链表、栈、树、图等)的运算。

(4) 有结构化控制语句(if-else、while、do-while、for)。C语言是理想的结构化语言,符合现代化编程风格。

(5) 语法限制不太严格,程序设计自由度大。一般的高级语言语法检查比较严格,能检查出几乎所有的语法错误。而C语言允许程序编写者有较大的自由度,因此放宽了语法检查。

(6) 允许直接访问物理地址、位操作(位操作、可直接写汇编语句,加asm做前缀)。可以直接对硬件进行操作。

(7) 生成目标代码质量高,程序执行效率高(因为.exe、.com文件执行效率高)。

(8) 可移植性好(对于机器型号、操作系统要求不高,可转换成其他语言)。

2. C语言的应用

(1) 科学计算(被突出的系统软件编写能力所掩盖)。