

实用

计算机入门丛书

吉林科学技术出版社

主 编：庞云阶 副主编：韩圭东

C 语言入门

赵春泉 编



实用计算机入门丛书

C 语言入门

赵春泉 编

吉林科学技术出版社

【吉】新登字 03 号

实用计算机入门丛书
C语言入门

赵春泉 编

责任编辑：王维义

封面设计：马腾骧

出版 吉林科学技术出版社 787×1092毫米32开本 10印张
211.000字
发行 新华书店总店北京发行所 1995年2月第1版
1995年2月第1次印刷 印数：1—6000册 定价：7.30元
印刷 吉林市怡春印刷厂 ISBN 7-5384-1378-2/TP·16

内容简介

C语言是一种通用的高级程序设计语言，但又具有低级程序设计语言的某些特征。因此，C语言是比较理想的系统软件开发语言。

本书共分十二章，比较全面、系统地介绍了C语言的基本概念和基本语法（数据类型、数据运算、各种语句和程序结构），以及使用C语言进行程序设计的基本方法。

全书由浅入深、通俗易懂。各章内容安排合理、深入浅出。例题比较丰富、简洁、剪表性剪。每章最后配有“注意点”一节，专门总结该章重点和编程中容易发生的问题。

凡是学过一门程序设计语言的读者都能读懂本书，并可掌握C语言及其程序设计的基本知识。本书可作为计算机普及教育、计算机培训班和大专院校非计算机专业的教材，也可供C语言的初学者自学使用。

编写说明

随着微型计算机的普及,人们对计算机知识的需要也愈加迫切,而初学者对计算机又感到高深莫测,无从下手,怎样才能更好地帮助初学者尽快地掌握微型计算机的操作、使用,这是我们编写这套微型计算机入门丛书的宗旨。

对于初学者,要尽快地掌握微型计算机的使用与操作,最重要的是要对微型计算机有一个基本的了解,掌握计算机的“个性”与“脾气”,这就需要一位好老师,《微型计算机入门丛书》能帮助您尽快与微机交上朋友。本套丛书以实用为其特点,从微机基础知识,到各种软件的使用,都讲得通俗易懂,一看就会,就能在微机上实现,特别适合初学者自学,是计算机初学者的良师益友。

微型计算机已开始进入办公室和家庭,它是现代化办公与现代化家庭的标志,掌握微型计算机知识,也是未来现代化社会的需要,所以说,微型计算机知识是有识之士必备的知识,是现代化社会的必修课。《微型计算机入门丛书》由著名计算机专家、吉林大学教授庞云阶主编,每个分册的编写者都是多年从事微机教学的有经验的老师,因此,这是一套非常实用的初学者的教材,相信它会对你掌握微机有很大帮助,本书由于编写时间仓促,编者水平有限,缺点和错误之处在所难免,恳请广大同行与读者批评指正。

编者

1994年8月

前 言

C语言是目前国内外比较流行、又很有发展前途的计算机高级程序设计语言。C语言的语法简洁、表达能力强,控制结构先进、合理,提供了丰富的数据类型和运算符,既可用于过程控制领域,又可在数值计算方面发挥作用,是比较通用的高级程序设计语言。另一方面,C语言的语法灵活,具有低级运算功能,其目标代码的质量可以与汇编语言相媲美,可以代替汇编语言,完成系统软件开发。因此,C语言是一种用途广泛、适用性较强的通用程序设计语言,是编写系统软件的理想语言。如著名的UNIX操作系统就是由C语言编写的。

C语言是广大计算机专业工作者喜爱和必学的一门工具语言。随着计算机技术和应用的不断普及,C语言开始为广大的计算机应用人员(主要指非计算机专业的计算机应用人员)所接受。学习C语言已成为广大计算机应用人员、非专业的计算机爱好者及一般计算机自学者的迫切需要。因此,可以把C语言作为计算机普及教育的一门课程。本书就是为此目的而编写的。

然而,C语言语法描述虽简洁,但又比较灵活、规则繁多、容易出错。目前,面向这类初学者的C语言教材已有不少,但大都是内容过细、篇幅较大,不利于短期教学和自学使用。为此,编写此教材的目标与C语言一样:简洁。本书具体有以

下几个特点：

1. 本书在语言表达上通俗易懂。不要求读者有较多的计算机专业知识。只要学过一门计算机语言或是参加过一定的计算机普及教育，即可读懂本书，并掌握C语言的基本知识。

2. 本书在内容安排上由浅入深。全书共分十二章及三个附录。每章一个概念、每章一个台阶。各章间难度跨越不大，有利于自学。学习上可从前向后、逐章学习，不要求跨章学习。

3. 全书可分三个学习阶段。第一阶段：第一章到第二章是C语言概念，包括C语言的发展和特点、数据、变量、语句及简单的C程序设计初步知识。第二阶段：第三章到第八章是基本知识，包括各种控制结构（转向、循环）、数组和函数概念，以及程序的结构和组织。这些内容是C语言的核心成分。读者应重点阅读这部分章节。第三阶段：第九章到第十二章是提高知识，包括字符数据、结构型数据（结构、联合和枚举）、文件概念及其他一些语法成分。它们是C语言的高级知识，用于编写复杂的实用程序。读者可在完成前两个学习阶段的基础上，进入本阶段学习。特别是想使用C语言实际开发应用软件的读者，更应注意本阶段各章节的学习。

4. 全书内容以微机上的C语言为背景，基本覆盖了ANSI C标准规定的C语言成分，并有重点地介绍了一批标准库函数，使读者具有一定的上机开发C应用程序的能力。

5. 为了减少篇幅，本书所配的例题都是经过精心选择的。做到简洁、突出重点、说明有力。全书并未选用计算机专业中的某些经典算法（如各种排序、迭代和回溯算法等），而是用最简单、最典型的算法和程序来说明遇到的语法现象，即点到为止。所有例题都在PC机上用Turbo C调试通过。若

在其他计算机系统上运行这些例题，可能需要做某些小小的修改。

6. 每章后的“注意点”一节是该章的要点和编程中经常出现的问题。C语言的初学者可把“注意点”的内容作为一章的小结，用于巩固所学的知识。C语言的编程者可把“注意点”用于程序调试中，帮助读者避免和修改程序的错误。

在本书编写过程中得到全炳哲老师的热情帮助，王松昕同志参加了部分程序调试工作，在此谨表示衷心感谢。

由于编者水平有限，书中肯定会有不少错误和不当之处，恳请专家和读者批评指正。

编者

1994·3

目 录

第一章 C语言概述	(1)
§ 1.1 C语言的发展及特点	(1)
§ 1.2 C程序	(2)
§ 1.3 C程序的构成及程序开发	(5)
1.3.1 C程序的构成	(5)
1.3.2 C函数的构成	(5)
1.3.3 库函数	(6)
1.3.4 C程序的开发流程	(7)
第二章 C程序设计初步	(9)
§ 2.1 数据类型、常量及变量	(9)
2.1.1 数据类型	(9)
2.1.2 常量	(9)
2.1.3 变量	(10)
§ 2.2 整型数据	(11)
2.2.1 整型常量	(11)
2.2.2 整型变量	(11)
2.2.3 整型表达式	(12)
§ 2.3 实型数据	(13)
2.3.1 实型常量	(13)
2.3.2 实型变量	(13)
2.3.3 实型表达式	(14)
2.3.4 整型和实型的混合表达式	(14)
§ 2.4 运算符和表达式	(14)

2.4.1	C 运算符简介	(15)
2.4.2	算术表达式	(16)
2.4.3	赋值运算符和赋值表达式	(18)
§ 2.5	数据输入和输出	(21)
2.5.1	数据输入输出函数	(21)
2.5.2	scanf 函数 (格式输入函数)	(21)
2.5.3	printf 函数 (格式输出函数)	(24)
2.5.4	其他输入输出函数	(27)
§ 2.6	注意点	(28)
2.6.1	几个要点	(28)
2.6.2	常见错误	(30)
第三章	逻辑运算和选择结构	(32)
§ 3.1	C 语言中的真假值	(32)
§ 3.2	关系表达式	(32)
3.2.1	关系运算符及其优先级	(32)
3.2.2	关系表达式	(33)
§ 3.3	逻辑表达式	(33)
3.3.1	逻辑运算符及其优先级	(34)
3.3.2	逻辑表达式	(34)
§ 3.4	if 语句	(36)
3.4.1	C 语言的三种 if 语句形式	(36)
3.4.2	if 语句的嵌套	(39)
§ 3.5	条件运算符	(41)
§ 3.6	switch 语句	(44)
§ 3.7	注意点	(46)
3.7.1	几个要点	(46)
3.7.2	常见错误	(48)
第四章	循环结构	(51)

§ 4.1	for 循环	(51)
§ 4.2	While 循环	(54)
§ 4.3	do-While 循环	(55)
§ 4.4	循环的嵌套	(59)
§ 4.5	注意点	(61)
4.5.1	几种循环的比较	(61)
4.5.2	使用循环语句要注意的问题	(62)
4.5.3	常见错误	(63)
第五章	其他控制结构	(65)
§ 5.1	goto 语句	(65)
§ 5.2	break 语句	(67)
§ 5.3	continue 语句	(67)
§ 5.4	注意点	(68)
第六章	数组和指针	(71)
§ 6.1	数组的说明及使用	(71)
6.1.1	一维数组的定义和引用	(71)
6.1.2	二维数组的定义和引用	(75)
§ 6.2	指针	(79)
6.2.1	指针的概念	(79)
6.2.2	指针变量的定义	(80)
6.2.3	指针变量的引用	(81)
6.2.4	指针表达式	(83)
§ 6.3	数组和指针的关系	(85)
6.3.1	指向数组元素的指针	(85)
6.3.2	通过指针引用数组元素	(86)
6.3.3	指向多维数组的指针	(89)
§ 6.4	注意点	(91)
6.4.1	几个要点	(91)

6.4.2	几个常见错误	(92)
第七章	函数	(95)
§ 7.1	函数的引入	(95)
§ 7.2	函数的定义	(98)
§ 7.3	函数的调用	(100)
7.3.1	函数调用的一般形式	(100)
7.3.2	函数调用的方式	(102)
7.3.3	函数的返回值	(104)
§ 7.4	函数的嵌套调用	(107)
§ 7.5	函数的递归调用	(107)
§ 7.6	函数的其他调用方式	(112)
7.6.1	main () 函数的参数	(112)
7.6.2	指针作函数的形参	(113)
7.6.3	数组作函数的实参	(115)
7.6.4	函数的指针和函数作函数的实参	(119)
7.6.5	返回指针的函数	(124)
§ 7.7	注意点	(126)
7.7.1	何时定义函数	(126)
7.7.2	使用函数中的常见错误	(128)
第八章	程序的组织	(132)
§ 8.1	C 程序的结构与组织	(132)
8.1.1	C 程序的结构	(132)
8.1.2	C 程序的组织	(133)
§ 8.2	程序变量的种类	(137)
8.2.1	局部变量和全局变量	(137)
8.2.2	动态存储变量和静态存储变量	(141)
§ 8.3	函数的组织	(150)
8.3.1	内部函数和外部函数	(150)

8.3.2	库函数	(151)
8.3.3	函数的原型	(154)
§ 8.4	注意点	(156)
8.4.1	几个要点	(156)
8.4.2	常见错误	(158)
第九章	字符型数据和字符串	(160)
§ 9.1	字符型数据	(160)
9.1.1	字符常量	(160)
9.1.2	字符变量	(161)
9.1.3	字符表达式	(164)
9.1.4	字符型数据的输入输出	(164)
§ 9.2	有关字符的函数	(165)
§ 9.3	字符串和字符数组	(168)
9.3.1	字符串	(168)
9.3.2	字符数组	(170)
9.3.3	字符串的输入输出	(174)
§ 9.4	字符串和指向字符串的指针	(177)
9.4.1	字符指针的定义	(177)
9.4.2	用字符指针处理字符串	(180)
9.4.3	字符指针作函数参数	(184)
§ 9.5	有关字符串的函数	(189)
9.5.1	字符串连接	(189)
9.5.2	字符串复制	(191)
9.5.3	字符串比较	(192)
9.5.4	字符串长度	(193)
9.5.5	字符串中的字符查找	(193)
9.5.6	字符串中查找字符串	(194)
9.5.7	字符串初始化	(195)
9.5.8	字符串转换	(196)

§ 9.6 注意点	(197)
9.6.1 几个要点	(197)
9.6.2 常见错误	(203)
第十章 结构类型、联合类型和枚举类型	(206)
§ 10.1 结构类型	(206)
§ 10.2 结构类型定义与结构变量	(208)
10.2.1 结构类型的一般定义形式	(208)
10.2.2 结构变量的定义	(209)
10.2.3 结构变量的初始化	(213)
§ 10.3 结构类型变量的引用	(215)
§ 10.4 结构类型数组	(217)
§ 10.5 指向结构的指针	(224)
10.5.1 指向结构变量的指针	(224)
10.5.2 指向结构数组的指针	(226)
10.5.3 用结构指针作函数实参	(229)
§ 10.6 联合类型	(231)
10.6.1 联合类型和联合类型变量的定义	(232)
10.6.2 联合变量的引用	(233)
§ 10.7 枚举类型	(239)
§ 10.8 用 typedef 定义类型	(241)
§ 10.9 注意点	(244)
第十一章 文件	(247)
§ 11.1 C 文件概述	(247)
11.1.1 文件的一般概念	(247)
11.1.2 C 文件	(249)
§ 11.2 文件的打开与关闭	(250)
11.2.1 文件打开函数 fopen ()	(250)
11.2.2 文件关闭函数 fclose ()	(253)

§ 11.3	文件的读写	(253)
11.3.1	fputc 函数和 fgetc 函数	(254)
11.3.2	fread 函数和 fwrite 函数	(256)
11.3.3	fprintf 函数和 fscanf 函数	(259)
11.3.4	其他读写函数	(259)
§ 11.4	文件的其他操作	(261)
11.4.1	文件的定位	(261)
11.4.2	文件的检测	(263)
§ 11.5	注意点	(264)
第十二章	C 语言的其他成分	(266)
§ 12.1	位运算	(266)
12.1.1	数的机内表示	(266)
12.1.2	位运算符	(267)
§ 12.2	逗号运算	(271)
§ 12.3	强制类型转换运算	(272)
§ 12.4	sizeof 运算符	(273)
§ 12.5	编译预处理	(274)
12.5.1	宏定义 (#define)	(275)
12.5.2	“文件包含”处理 (#include)	(279)
12.5.3	条件编译 (#if、#ifdef、#ifndef、#else、 #endif)	(280)
附录 A	C 语言的关键字	(283)
附录 B	常用 C 库函数	(284)
附录 C	常用字符的 ASCII 码表	(296)
参考文献	(301)

第一章 C 语言概述

§ 1.1 C 语言的发展及特点

C 语言是在国内外广泛流行的高级程序设计语言。虽然它是一种高级程序设计语言，但也提供丰富的低级操作，而它既适合于开发各种各样的应用软件，也适合于编写与计算机硬件密切相关的各种系统软件。因此，也可以把 C 语言看作是介于高级程序设计语言和低级程序设计语言之间的一种“中级”程序设计语言。

C 语言的设计目标是语言的级别达到高级程序设计语言。同时能够描述象操作系统那样的系统软件。C 语言的许多重要思想来自于 CPL (Combined Programming Language) 语言和 BCPL (Basic PCL) 语言。CPL 语言是在 ALGOL 60 语言基础上推出的一种高级程序设计语言，它比 ALGOL 60 更接近于硬件，但规模较大，难懂，也难实现。BCPL 语言是把 CPL 语言简化而来的语言。

1970 年 K. Thompson 在设计开发 UNIX 操作系统时，对 BCPL 语言做了进一步简化，设计了更简单的 B 语言。由于 BCPL 和 B 语言都过于简单，语言的描述能力有限，只适合于解决一些特殊类型的问题。于是，在 1972 年至 1973 年间，D. M. Ritchie 在 B 语言的基础上设计了 C 语言，C 语言保持了 B 语言的精粹特点，同时提供了丰富的描述手段（数

据类型和运算)。

1973年, K. Thompson 和 D. M. Ritchie 两人合作用 C 语言重写了 UNIX 操作系统的大部分代码(90%以上)。重写 UNIX 操作系统使 C 语言的能力得到充分的发挥。同时, UNIX 操作系统的发展和推广也促进了 C 语言的发展和普及。从某种意义上讲, C 语言和 UNIX 操作系统是紧密相联的。目前, 带有 UNIX 操作系统的计算机几乎都有 C 语言。但是 C 语言并非只限于 UNIX 操作系统, 目前几乎所有的计算机都配有 C 语言的编译程序及相应的开发工具。

C 语言具有如下一些特点:

- (1) C 语言的基本组成部分紧凑、简洁, 使用方便、灵活。
- (2) 具有丰富的数据类型和运算符。数据类型间转换自由, 便于写出复杂的表达式, 语言的表达能力强。
- (3) 提供丰富的控制语句, 保证程序的结构化和模块化。
- (4) 提供接近于汇编语言的描述手段, 如地址指针运算和二进制位运算等。
- (5) 生成的目标代码质量高。
- (6) 用 C 语言编写的程序具有良好的可移植性。

当然, 和其他程序设计语言一样, C 语言也具有其弱点, 如运算符的优先级较多, 不易记忆; 类型检验太弱, 转换太随便; 语言的语法较复杂, 没有提供较高级的模块化机制。此外, 目前不同机器上使用的 C 语言版本多种多样, 相互之间并不完全兼容。

§ 1.2 C 程序

任何一个程序都是由一个主程序、若干子程序和数据描