

# 实用 C 语言基础教程

吕凤翥 王克宁 编著

北京大学出版社

新登字(京)159号

实用 C 语言基础教程

吕凤翥 王克宁 著

责任编辑：张豫夫

\*

北京大学出版社出版发行

(北京大学校内)

北京大学印刷厂印刷

新华书店经售

\*

850×1168毫米 32开本 17印张 420千字

1992年12月第一版 1992年12月第一次印刷

印数：00001-6,000册

ISBN 7-301-01918-1/TP·155

定价：12.00元

# 前 言

为了降低成本,减少程序开发和移植的费用以及便于计算机之间的连接,开放系统的应用愈来愈普遍。UNIX 作为开放系统的基础,已成为标准的工业通用操作系统。作为 UNIX 开发工具和应用环境的 C 语言是目前最重要的通用程序设计语言,得到了广泛的应用。尤其需要指出的是,Microsoft C 和 Turbo C 提供了良好的编程环境和工具,使得目前 C 语言已成为最大众化的 IBM PC/AT 以及 386/486 个人机上 MS-DOS 和 MS-Windows 系统中最广泛地应用的程序设计语言——在我国尤其如此。C 语言之所以如此广泛地应用,主要原因有两点:一是 C 与 UNIX 之间存在着密不可分的关系;二是 C 语言本身是软件工程中倡导的结构化程序设计语言,且语句简练、书写灵活、运行效率高、处理能力强并具有很好的移植性。

C 语言与 UNIX 系统类似,当初都是为专家设计的,对初学者来说,常常感到无从下手或神秘莫测。为了更快地让读者学会使用 C 语言,作者根据使用 C 语言近十年的实践经验和教学、培训、辅导的体会编写了本书。本书的特点是易学易懂、由浅入深、实践性强,具有丰富的例子并提供了若干实用的附录供使用时参考。本书对读者常犯和易犯的错误作了重点讲述,适当地介绍了 C 语言的实现机制,以帮助读者理解 C 语言的原理,适用于自学。目前出版的许多有关 C 语言中文书籍中,普遍存在着编排错误多、侧重在某一专题或方向、要求读者有较高的计算机知识等不足。在本书的编写过程中,作者参阅了大量的 C 语言文献,力求弥补上述不足。书中的所有示例均在 SCO XENIX 系统 V/386 版本 2.3, Turbo C 版本 2.0 和 Microsoft C 版本 6.0 中作了验证。书中采用了 SCO

XENIX 系统 V/386 版本 2.3 中 386 模式的运行结果。在附录中介绍了目前最广泛使用的 Turbo C, Microsoft C 和 UNIX/XENIX C 编程环境。

学习 C 语言的最好方法是上机实践。掌握了 C 语言的语言规范并不等于掌握了 C 语言, C 语言十分灵活, 要在实践中不断提高应用水平。为了提高读者的学习效率, 作者已将本书中的所有例题制作成软盘由北京大学出版社向读者提供。由于作者水平有限, 书中错误在所难免, 欢迎广大读者批评指正。

作者

1992 年 10 月

# 目 录

第一章 C 语言概述 .....	1
§ 1.1 C 语言的由来与发展 .....	1
§ 1.2 C 语言的特点及应用 .....	2
§ 1.3 C 语言的词法及其规则 .....	6
§ 1.4 C 语言程序设计的实现 .....	11
§ 1.5 C 语言的程序结构举例 .....	14
§ 1.6 C 语言的标准和发展 .....	18
思考题 .....	19
上机练习题 .....	20
第二章 常量和变量 .....	21
§ 2.1 常    量 .....	21
§ 2.2 变    量 .....	27
§ 2.3 数    组 .....	30
§ 2.4 综合举例 .....	36
§ 2.5 ANSI 扩充功能 .....	39
思考题 .....	41
上机练习题 .....	41
第三章 运算符与表达式 .....	43
§ 3.1 运算符的种类 .....	43
§ 3.2 运算符的优先级和结合性 .....	50
§ 3.3 表    达    式 .....	52
§ 3.4 数据类型转换 .....	77

§ 3.5	误差和溢出	79
§ 3.6	综合举例	84
	思考题	87
	上机练习题	87
第四章	C 语言的语句	93
§ 4.1	表达式语句和空语句	93
§ 4.2	复合语句	94
§ 4.3	分支语句	95
§ 4.4	循环语句	102
§ 4.5	转向语句	117
§ 4.6	综合例题	127
	思考题	133
	上机练习题	133
第五章	函数和存储类型	137
§ 5.1	C 语言程序函数的特点	137
§ 5.2	函数的定义和说明	140
§ 5.3	函数的调用	143
§ 5.4	作用域规则	151
§ 5.5	存储类型	153
§ 5.6	递归函数	167
§ 5.7	综合练习	171
	思考题	179
	上机练习题	179
第六章	指针	182
§ 6.1	指针的定义	182
§ 6.2	指针的说明	183
§ 6.3	指针的赋值	184
§ 6.4	指针的运算	187
§ 6.5	指针与数组	192

§ 6.6	指针和函数	217
§ 6.7	综合举例	230
	思考题	237
	上机练习题	237
第七章	结 构	242
§ 7.1	结构的定义和说明	242
§ 7.2	结构成员表示和赋值	247
§ 7.3	结构与数组	253
§ 7.4	结构与指针	255
§ 7.5	结构和函数	258
§ 7.6	结构的嵌套	263
§ 7.7	综合举例	266
	思考题	280
	上机练习题	282
第八章	联合、枚举和其他	285
§ 8.1	联 合	285
§ 8.2	枚 举	289
§ 8.3	类型定义	293
§ 8.4	位 段	295
	思考题	300
	上机练习题	301
第九章	预处理程序	304
§ 9.1	宏 定 义	304
§ 9.2	文件包含	315
§ 9.3	条件编译	316
§ 9.4	行 控 制	320
§ 9.5	ANSI 扩充	324
	思考题	326
	上机练习题	327

第十章	输入/输出函数和文件操作	330
§ 10.1	文件的概念	330
§ 10.2	标准文件的输入输出函数	332
§ 10.3	一般文件的输入输出函数	345
§ 10.4	其他常用函数	374
§ 10.5	低级的输入输出函数	378
§ 10.6	综合举例	386
思考题		391
上机练习题		391
第十一章	常见编程错误分析	395
§ 11.1	有关运算符方面的常见错误分析	395
§ 11.2	有关类型方面的常见错误分析	399
§ 11.3	有关语句方面常见错误分析	403
§ 11.4	有关函数方面的常见错误分析	405
§ 11.5	有关指针和数组方面的常见错误分析	407
§ 11.6	有关预处理方面的常见错误分析	412
§ 11.7	有关二义性方面常见错误分析	416
§ 11.8	其他方面常见错误分析	421
附录 A	ASCII 编码表	424
附录 B	C 语言运算符的优先级和结合性	426
附录 C	C 语言语法汇总	428
§ C.1	单词	428
§ C.2	表达式	430
§ C.3	说明	431
§ C.4	语句	433
§ C.5	定义	433
§ C.6	预处理程序指令	434
§ C.7	K. R 标准 C 与 ANSI C 的区别	434
§ C.8	XENIX/UNIX 系统中的 C 实现	435

§ C.9 MS-DOS 中 TC 和 MC 的实现 .....	436
附录 D C 语言的可移植性 .....	438
§ D.1 概 述 .....	438
§ D.2 程序的可移植性 .....	438
§ D.3 机器的硬件 .....	438
§ D.4 编译程序的差异 .....	441
§ D.5 程序环境的差异 .....	442
§ D.6 数据的可移植性 .....	442
附录 E UNIX/XENIX 系统中 C 语言编程环境 .....	443
§ E.1 基本操作命令 .....	443
§ E.2 vi 编 辑 器 .....	445
§ E.3 C 语言程序的编译和执行 .....	451
§ E.4 C 语言开发实用程序和工具 .....	455
附录 F MS-DOS 中 Turbo C 版本 2.0 编程环境 .....	469
§ F.1 Turbo C 集成开发环境 .....	469
§ F.2 Turbo C 的命令行方式 .....	493
§ F.3 调试程序的若干说明 .....	495
附录 G MS-DOS 中 Microsoft C 版本 6.0 编程环境 .....	498
§ G.1 程序员工作台命令 pwb 的使用方法 .....	498
§ G.2 编译器命令 CL 的使用方法 .....	515
§ G.3 连接器命令 LINK 的使用方法 .....	521
§ G.4 程序库命令 LIB 的使用方法 .....	526
附录 H 参 考 文 献 .....	531

# 第一章 C 语言概述

本章首先介绍 C 语言的由来和发展,介绍 C 语言的特点和应用。接着,为使读者具体实现 C 语言的程序,介绍 C 语言程序设计的实现。本章还介绍 C 语言的词法及其规则。最后,举例说明 C 语言的程序结构,以便使读者对 C 语言建立一个整体的概念。

## § 1.1 C 语言的由来与发展

C 语言诞生于 1972 年,它是由美国电话电报公司(AT&T)贝尔实验室的 D. M. Ritchie 设计的,并首次在一台使用 UNIX 操作系统的 DEC PDP- 11 计算机上实现的。

C 语言是在一种称为 B 语言的基础上,克服了 B 语言依赖于机器又无数据类型等局限性开发的语言。B 语言是由 UNIX 操作系统的设计者 K. Thompson 实现的。B 语言的主要思想来自于 BCPL (Basic Combined Programming Language) 语言,它们有较多类同之处。BCPL 语言是 M. Richards 基于 CPL(Combined Programming Language) 语言于 1967 年提出的一种语言,而 CPL 语言又是于 1963 年根据 ALGOL60 提出来的。因此,追溯历史,可以看出 C 语言是由 ALGOL 语言系统演变而来的。即:

ALGOL60(1960 年提出) CPL(1963 年提出)

BCPL(1967 年提出) B(1970 年实现)

C(1972 年实现)

C 语言的设计和实现与 UNIX 操作系统有着极其密切的联系。

C 语言的出现是为 UNIX 系统服务的。C 语言出世后就用它改写了 UNIX 系统(1973 年), 于是, 建立在 C 语言基础之上的 UNIX 第五版本(V5)开创了 UNIX 系统发展的新局面。以后的 UNIX 各种版本都是在 C 语言上开发并发展的。同时, UNIX 系统的巨大成功和广泛应用也极大地促进了 C 语言的发展和使用的。到了 80 年代, C 语言在非 UNIX 系统的其它操作系统上也开始使用, 很快地, 它与 FORTRAN, PASCAL 等其他高级语言一样已成为大、中、小和微型机上共同使用的语言。

1983 年, C 语言和 UNIX 系统的开发者一起获得了当年度的 ACM 图灵奖, 进而确定了 C 语言的学术地位, 得到了高度的评价。

## § 1.2 C 语言的特点及应用

C 语言与 BASIC, FORTRAN, COBOL 和 PASCAL 语言相比是年轻的语言, 因此, 它吸取了其他语言的长处, 也避免了其他语言的不足。C 语言具有很多的特点, 不愧为一种后来居上的语言。其特点主要有如下几个方面:

### 1. C 语言是一种结构化语言

结构语言是现代的编程风格, 它层次清晰, 易于调试和维护。在常用的高级语言中, C 语言和 PASCAL, Ada 等语言都属于结构化语言, 而 FORTRAN, COBOL 和 BASIC 语言都属于非结构化语言。但 C 语言和 PASCAL 语言不完全相同, C 语言是不完全的结构化语言, 主要表现在:

(1) 在同一个函数中, 允许使用 goto 转移语句。在实际编程中, 通常仅在同层次或由内层到外层应用 goto 语句。

(2) 所有函数都是平等的和同级的, 可以相互调用, 不存在嵌套关系。

(3) 在同一个控制流或函数中, 允许有多个出口。

这些不仅不影响 C 语言的应用, 相反使 C 语言的使用更灵活, 运行速度更快。

C 语言的主要结构成分是函数, 函数是基本的结构模块。C 语言的程序可以象搭积木那样组装函数, 形成多层次的函数调用关系。函数定义为完成一个独立任务的程序段, 函数可以独立地编译成目标代码, 因此, 通过函数实现了 C 语言程序的模块化。函数之间规定了多种传递信息的方式, 保证了函数之间的通信。为了便于模块化, C 语言的程序采用了函数串的结构方式, 即一个程序由若干个文件组成, 一个文件由若干个函数组成, 其中有一个是主函数 `main()`, 而其余的函数将被主函数和其它函数调用, 于是 C 语言程序形成了以主函数为头的函数串结构形式。因此, C 语言是以函数为基础的语言。

为了便于结构化, C 语言具有一套完整的控制流语句群, 包含了循环、分支和转向等各种语句, 对转向语句做了限制, 例如, 只允许 `goto` 语句用于函数体内, 这就保证了函数是基本的结构模块。

为了便于结构化, C 语言采用了构造型数据类型。在 C 语言的数据类型中, 除了有基本数据类型(`char`, `int`, `float` 等)外, 还具有构造类型(数组、结构、联合和枚举)。

## 2. C 语言的语句简炼, 书写灵活

C 语言的这一特点具体表现在如下几个方面:

(1) 用较少的关键字表达了较丰富的功能。C 的关键字比 PASCAL 少, 有些关键字用符号来代替。例如: 用花括号 { 和 } 代替复合语句中通常用的 `begin` 和 `end`, 用 `++` 和 `--` 运算符表示加 1 和减 1 操作, 使用三目运算符 `?:` 可以表示一个简单的 `if-else` 语句, 这一切都使得 C 语言书写简洁。另外, 在表达式中可以使用赋值表达式, 例如:

```
...  
while((ch = getchar()) != EOF)
```

...

(2) 类型说明符采用缩写方式, 使得书写简单。例如: 整型说明符用 `int`, 字符型说明符用 `char` 等等。

(3) 具有预处理功能, 采用宏定义和文件包含等预处理语句, 可以使得书写简洁, 同时也便于修改和移植。

(4) 从语言内部实现角度来看, C 语言本身不提供 I/O 手段, 也不提供并行操作等复杂控制结构, 而这些功能由支持环境提供, 使得编译程序简洁。

### 3. C 语言的处理能力强

C 语言不仅可以对数值和字符进行算术和逻辑运算, 而且还具有一些特殊的处理功能:

(1) 由于 C 语言引进了指针, 增强了构造型变量, 特别是数组和结构的存取功能。同时也增加了地址运算的能力, 特别是对于字符串处理十分方便。

(2) C 语言具有二进制位运算功能和寄存器存取功能, 大大地加强了对硬件性能的描述, 在许多方面具有汇编语言的特性, 因此, 代码效率比较高。统计表明, 用 C 语言编写的程序编译以后所得到的代码比用汇编语言直接编写的代码效率仅低 15—20%, 这是其他高级语言难以比拟的。因此, 不仅用 C 语言编写 UNIX 操作系统, 而且还编写其他系统软件和应用软件, 例如, 它成功地描述了 INFORMIX, UNIFY 数据库管理系统, Logo, Prolog 解释系统以及 X - Windows, AutoCAD 绘图系统等等。

### 4. C 语言具有很好的移植性

用户很希望所编写的程序能移到不同类型机器上运行, 而 C 语言的程序就具有很好的移植性。其主要原因有:

(1) C 编译程序用少于 1 万条 C 代码写成, 因此便于修改。

(2) 与机器相关的因素与主代码分开, 通过预处理程序和其他实用程序来实现。

(3) 语义性不强。一些解释留给具体实现去规定。

(4) 在所有的 C 语言实现中, 公共函数库, 尤其 I/O 函数库几乎完全相同。

因此, C 语言编译可以比较方便地移植到许多机器上。统计资料表明, 不同机器上的 C 编译程序 80% 的代码是公共的。因此, C 语言目前在众多的微型机和小型机上广泛的使用。

总之, C 语言是一种与 UNIX 密切相关的强功能、高效率、面向程序员的结构化系统程序设计语言。它主要用于实现 UNIX 操作系统核心及各种实用程序和程序开发工具, 包括 C 编译器本身在内。在网络通信、数据库管理系统、图形处理、文字处理、电子表格、软件开发工具和办公室自动化等很多方面, C 语言已成为主要程序设计语言。

C 语言也不可避免的具有一些不足之处, 主要表现为:

(1) C 语言是一种非强类型语言, 数据类型及其值在运算时不一定严格对应。例如, char 型可作为 int 型值参与运算; unsigned int 可进行位操作运算; 不存在区别整数的逻辑值; 任何表达式均可作为判断真假的条件表达式等等。另外, 数据类型转换比较随便, 这里有不安全因素。

(2) C 语言的编译程序在很多情况下, 允许表达式和参数表内重新排列求值顺序。这一点在有副作用运算符出现时, 要注意其二义性, 即在不同编译系统下可能得到不同的结果。

(3) C 语言不具备自动数据边界检查功能。因此, 在对数组等具有边界的数据类型赋值时, 要注意不要超越边界, 否则将会把数据搞乱。

(4) C 语言有较多的运算符和较复杂的优先级, 使初学者感到难记和麻烦。例如, 同一个符号 \* , 表示不同的含义, 在做运算符使用时, \* 为单目运算符表示取内容, \* 为双目运算符表示乘法, 在做说明符使用时, \* 表示指针, 另外, 斜杠与星号(/ \* ) 又组成注释符。

C 语言的以上四点不足之处, 在学习 C 语言中要充分注意, 否则将会出现问题。

由于 UNIX 事实上已成为目前计算机工业界主流倡导的开放系统的基础, C 语言又与 UNIX 之间存在着密不可分的关系, 因此, C 语言在计算机语言中占有十分重要的地位。它吸引了一大批极其忠实的支持者, 同时又招致了十分挑剔的反对者。使用 C 语言编程的程序员都无不为之自豪, 称赞 C 语言的多功能性和灵活性, 而反对者却指责它简洁的语法规则和可能的二义性。计算机专家一般都站在支持者的行列, 而反对者多为新手。还有一种误解, 说 C 语言是系统程序设计语言, 不适宜商业应用。C 语言目前已在数据库系统、应用和报表生成器、第四代语言、窗口和菜单系统等方面起着决定性的作用。

由于 UNIX 和 C 最初是为贝尔实验室的专家设计的, 大量围绕 UNIX 和 C 语言的“秘诀”和“学问”也随之发展起来。历史上的多数 UNIX 和 C“名家”都是通过已成为名家的人的言传身教学到技能的。时至今日, 能有一位经验非常丰富的人来回答问题并解释 UNIX 和 C 上较深奥的某些方面, 仍是大有收益的。UNIX 和 C 与当前所提倡的软件“对用户透明”的思想背道而驰, 越是了解系统的内部功能和实现机制, 越能使系统更好地为自己服务并提高工作效率和程序的运行效率。最后, 要强调的一点是, 学习 UNIX 和 C 的最好途径是实践, 即上机操作。上面所述, 作者深有体会。作者每次实践后, 对 UNIX 和 C 都有更深、更进一步的认识。如果读者在学习本书的过程中能不断地在机器上实践, 相信你很快会成为一名 C 语言专家。

### § 1.3 C 语言的词法及其规则

从 C 语言的程序来看, 它是从字符开始由小到大组成的。具体

表示如下:

字符 单词 语句 函数 文件 程序

可见,字符是其最基本的元素,单词是由字符组成的基本语法单位。下面将讨论 C 语言的字符集和单词。

### § 1.3.1 C 语言的字符集

C 语言定义了两个字符集: C 字符集和可表示字符集。C 字符集由字母、数字、空白、标点和特殊字符组成,可表示字符集由 C 字符集加用户能以图形方式用一个字符表示出来的符号,包括汉字和图形字符。可表示字符集的范围取决于所用的终端或字符设备类型。除在字符常量、字符串常量和注释中可使用可表示字符外, C 程序只能使用 C 字符集中的字符。由此可知,汉字在 C 语言中只能作为字符常量、字符串常量或注释使用。

#### 1. 字母和数字

小写字母: a b c d e f g h i j k l m n o p q r s t u v w x y z

大写字母: A B C D E F G H I J K L M N O P Q R S T U V W  
X Y Z

数字: 0 1 2 3 4 5 6 7 8 9

#### 2. 空白字符

空白字符由空格符(space)、回车符(carriage return)、换行符(line feed)、换页符(form feed)、水平制表符(horizontal table)、垂直制表符(vertical table)组成。空白字符除在字符常量和字符串常量中使用外,在程序的其他地方编译程序忽略不计,程序员可使用额外空白字符增加程序的可读性。

#### 3. 标点和特殊字符

C 字符集中的标点和特殊字符用于许多目的,从组织程序的正文到指定或定义程序执行或完成的任务。

表 1.1 C 字符集中的标点和特殊字符

字符	名称	字符	名称	字符	名称
,	逗号	{	左花括号	#	数字号
.	点	}	右花括号	%	百分号
;	分号		大于号	&	和号
:	冒号		小于号	^	脱字符
'	单引号	!	惊叹号	*	乘号
"	双引号		竖线	-	减号
(	左括号	/	斜线	=	等号
)	右括号	\	反斜线	+	加号
[	左方括号	~	非		
]	右方括号	_	下划线		

### § 1.3.2 C 语言的词汇及语法规则

由若干个有序的字符组成单词。单词的集合称为词汇。C 语言的单词有：标识符、关键字、运算符、分隔符、常量、字符串和注释符。下面分别描述。

#### 1. 标识符

标识符是用户为程序中所使用的变量、函数和标号等所起的名字，由字母、下划线和数字组成的字符串组成，且第一个字符必须是字母或下划线。一般情况下，由下划线打头的标识符是由系统本身使用的。C 语言对标识符的字符个数一般不做限制，但是字符过多会受到机器的限制，有些微机对标识符只区分前 8 位，即两个标识符的前 8 位相同，则认为是一个标识符。另外，多数 C 编译程序区分大写和小写字母。