

A large, stylized letter 'C' that serves as the background for the cover. The 'C' is composed of two concentric, slightly irregular shapes. The outer shape is a dark blue color, and the inner shape is a vibrant red color. The space between the two shapes is black, creating a high-contrast, graphic effect.

C语言教程

——程序设计指导

美 Al Kelley, Ira Pohl 著
蔡宗泽 刘超 译

TP312C/FZ11

C 语 言 教 程

程序设计指导

〔美〕 Al Kelley Ira Pohl 著
费宗莲 刘 超 译

人 民 邮 电 出 版 社

登记证号(京)143号

A Book on C
An Introduction to Programming in C
Al Kelley Ira Pohl
The Benjamin/Cummings Publishing
Company, Inc
1984

内 容 提 要

C是一种通用的程序设计语言。它简洁有效,经济实用,可移植性强。C语言适用于从微型机直至大型机的各类计算机,在系统程序设计和应用程序设计中均已得到广泛的应用。

本书全面系统地介绍了C语言,并包括了最新发展的特点,通俗易懂。全书共分12章。第1章概述了C的基本特点,第2章至第10章详细地介绍C的功能,第11章讲述了C/UNIX接口。每章都附有小结和习题。本书还通过剖析大量的程序实例,使读者加深理解概念便于掌握这一语言。

本书可用作大专院校有关专业的教科书,也可供科学技术人员和从事计算机程序设计的人员作为自学教材或参考书。本书还可以作为初学者的入门读物。

C语言教程

程序设计指导

(美)Al Kelley Ira Pohl 著

费宗莲 刘超 译

责任编辑 马月梅

人民邮电出版社出版发行

北京东长安街27号

北京顺义振华印刷厂印刷

新华书店总店科技发行所经销

开本:850×1168 1/32 1989年7月 第一版

印张:13 16/32 页数:216 1993年1月 北京第2次印刷

字数:356千字 印数:9 001—14 000册

ISBN7-115-04043-5/TP·043

定价:10.30元

译者序

本书是一本实用的C语言教程,适合于大学教学或个人自学。它既可用作初学者的教科书,又可作为有经验的程序员的参考书。

全书共分12章,具有以下特点:

1. 内容完整,循序渐进。每一章均包括介绍概念、分析例题、归纳重点等环节,通俗易懂。
2. 编排合理,结构灵活。译者在用本书作为大学“C语言程序设计”课程的讲学中,即按现成的章节为大纲,方便有效。
3. 例题丰富,涉及面广又有实用价值,尤其是采用了交互式方法,便于读者边学边实践。
4. 内容新颖,有一定深度。本书包括了象枚举类型这样的新特点,也包括了引用自身的结构、表处理和C/UNIX接口等专题。

C语言自从1972年由丹尼斯·里奇设计并在UNIX操作系统上实现以来,得到越来越广泛的应用。它适用于从微型机直至大型机的各类计算机,它适用于系统程序设计和应用程序设计。C是十分有效和富于表达力的语言。

我国的“七五”规划中,UNIX系统的国产化被立为重点科技攻关项目,C的推广也势在必行。然而,目前国内C语言方面的专著甚少,这本译著的出版正是在这样的时机奉献给我国广大的科技工作者。

在此,谨向在翻译过程中给予我们热情支持和帮助的金茂忠副教授和沈英同志表示感谢。

前 言

一本好的 C 语言书应该反映出这一通用程序语言的优美简洁和强大威力。本书通过在许多应用领域里的交互式运行程序来介绍 C 语言程序设计。本书循序渐进地全面地介绍了 C 语言,其中包括枚举类型和结构赋值这样一些新特点。为了说明每一个重要的语言特点,本书列举了大量的程序实例,并且列出了许多表格来归纳关键的内容,同时还为读者以后作参考提供了方便。每一章结束时都有小结和习题。小结概括了一章中介绍的重要概念。习题则丰富和扩充了课文内容。

本书是一本 C 语言的通用教程,适合于学生和专业科技人员使用。它可以用作学生学习程序设计的初步入门指南或者中等水平的教材,也可以方便地结合一些其它课程,例如操作系统、软件设计、比较程序语言、图形学、数学结构、数据库系统、数值分析、编程方法学和科学应用等作为参考书使用。C 能够应用于以上任何一个专业。对于实现这些应用所需的所有 C 的特点,本书都作了详细解释。本书之所以适用于数据结构课程是因为书中讨论了先进的数据结构,例如枚举类型、联合、引用自身的结构和不等长数组等。对于涉及到 UNIX 操作系统的课程而言,本书揭示了 UNIX 的文件结构和系统例程,从而使 C 语言程序员能够添加现存的 UNIX 库,并且更好地理解在 UNIX 环境下的 C 程序。在应用程序和科学程序方面,本书讨论了如何写实例函数库。书中列出许多工作代码,所反映的内容包括了统计、解方程、排序、文本处理、文件处理和玩游戏。

完整的 C 语言 本书向计算机专业人员全面地介绍了 C 语言,其中包括枚举类型、表处理和 C/UNIX 接口。第 1 章是语言概述。在读完第 1 章以后,专业人员就应该能写 C 程序了。书中每一章节都

是自行独立的,所以有基础的读者可以根据自己的需要跳过某些章节。第11章特别向那些在UNIX环境里用C的专业系统程序员进行了专题介绍。

交互式环境 本书自始至终采用了现代化的交互式环境,并且提倡在实践中学习。所采用的标准输入/输出是屏幕和终端,与此有关的问题在本书中都作了一一解释。因此,本书对于小型家用和商用计算机用户以及大型交互式系统用户都是适用的。唯一的假设条件是读者能够使用交互式系统。本书的参考系统是UC Berkeley 4.2 UNIX和它的编译程序及系统。但是,书中的内容能够应用于任一现有的C系统。课文还描述了如何测试你所用机器上的本地系统以及与所用系统有关的特性。

工作代码 本书的教学方法是通过例题、解释和语法来介绍语言。全书通篇都采用工作代码,即利用小而有用的例题来说明一些重要的技术概念。小,则容易理解。说它有用,是因为程序设计是基于积木式分层原则的,最终都是实用的。所描述的程序可以用于实际的系统中。很明显,作者的观点是:学习语言者应当勇于实践、乐在其中。

灵活的编排 本书的结构灵活,使用方便。第1章采用了一种独特的方法,即通过程序分解说明来阐述新的语言特点。第1章共分两部分。第一部分介绍交互式终端输入/输出所需的重要的程序设计技术,这些都是必须理解的基本内容。第二部分进一步研究整个语言,这对于那些有经验的、熟悉C语言特点的程序员来说是容易理解接受的。但是,当选用这本书作为一门程序设计的初级教程时,第二部分最好留待以后再学。

第2章描述语言的词法和有选择性地介绍一些语法规则来说明C语言的结构。教师在教学中可以决定是否教第2章中所描述的BNF表示法。如果跳过这一表示法并不影响连贯性。本书采用了BNF格式语法说明,使学生有机会学习这种标准的程序语言说明格式。另外,语言的成份完全通过例题及其一般性解释来描述。

包括枚举类型在内的新特点 第3章至第9章逐个地介绍C语言的特点。其中进行了许多深入的专题讨论,在初读时可以跳过而不会影响对整体的理解。例如枚举类型和结构赋值,虽然现行的许多系统已经包括这些特点,但并不是所有的系统都有。另外,还讨论了一些与机器有关的特点,例如字的大小的设置和浮点表示法等。对于一个初学者来说,不一定需要考虑这些细节。至于怎样在一个不同于我们所假定的参考系统的本地系统上检查这些设置,在习题中作了有关讨论。

递归和表处理 递归对于初学者来说常常是一个神秘的课题。第8章清晰地讨论了这一专题,并且阐述了递归相对迭代的优越性。第10章介绍了基本的表处理技术。这些技术知识在深入的程序设计和数据结构课程中都是必不可少的基础。

UNIX 接口 第11章描述了与UNIX的连接。这一章介绍了如何进行文件处理,并且提供了进行系统程序设计所必要的语言结构。同时,用了相当的篇幅讨论了各种各样的标准库输入/输出函数。读者可以根据具体需要有选择地研究这些深入的课题。这些内容组成了程序设计实践中极好的高级课程基础,也成为用C作为执行语言的高级计算机科学课程的辅助教材。

表格、小结和习题 本书通过许多表格和清单简明扼要地归纳了关键内容。这些表格帮助读者检查自己对语言的理解。例如,C具有丰富的运算符,而且允许几乎任意一种有用的运算符之间的组合。理解每一个单个的运算符和复合运算符的求值顺序和结合性是很重要的。这些内容在通篇课文中都以表格的形式加以说明。作为一本参考书籍,表格和代码都能方便地查阅。习题检验了语言的基本特点,并且讨论了进一步的与所用系统有关的特点。许多习题都是针对实际问题的。有些练习的用意是严格地检验读者对C的语法或者语义的理解。还有些题目进行了指导性的讨论,这可能超出了课文的范围,但却可能引起某些读者的特别兴趣。总之,这些习题包括了各种不同难度的题目,所以教师在布置作业时可以根据不同的对象出题。

目 录

第0章 从0开始.....	1
0.1 什么是C语言	2
0.2 习题	4
第1章 C的概述	5
1.1 编程和准备	5
1.2 程序输出	7
1.3 变量、表达式和赋值	10
1.4 #define 和 #include 的用法	13
1.5 printf()和 scanf()的用法	17
1.6 控制流	21
1.7 getchar()和 putchar()的用法	29
1.8 函数	35
1.9 数组、串和指针	40
1.10 文件	53
1.11 实例:随机数和最佳算法	58
1.12 小结	63
1.13 习题	64
第2章 语法和词法	68
2.1 字符和记号	68
2.2 语法规则	70
2.3 标识符	73
2.4 关键字	74
2.5 运算符和分隔符	74
2.6 串	75
2.7 注释	76

2.8	常量	7 7
2.9	小结	7 8
2.10	习题	7 8
第3章	说明、表达式、赋值、数据类型	8 0
3.1	说明	8 0
3.2	基本数据类型	8 1
3.3	表达式、语句和赋值	8 3
3.4	数据类型 int	8 5
3.5	用 int 数据的算术运算符	8 7
3.6	整型变量的加 1 和减 1 运算符	9 0
3.7	整型 short、long 和 unsigned	9 2
3.8	浮点型	9 4
3.9	数学函数	9 8
3.10	数据类型 char	9 9
3.11	赋值运算符	102
3.12	转换和强制转换	103
3.13	常量	106
3.14	十六进制和八进制常量	107
3.15	求值顺序	111
3.16	小结	111
3.17	习题	112
第4章	控制流	118
4.1	关系运算符、相等运算符和逻辑运算符	118
4.2	关系运算符和表达式	119
4.3	相等运算符和表达式	122
4.4	逻辑运算符和表达式	124
4.5	复合语句	129
4.6	空语句	130
4.7	if 和 if-else 语句	130

4.8	while 语句	136
4.9	for 语句	138
4.10	实例:布尔变量	140
4.11	逗号运算符	141
4.12	do 语句	143
4.13	实例:FIBONACCI 数	145
4.14	实例:素数	147
4.15	小结	149
4.16	习题	150
第 5 章	函数	161
5.1	函数定义	161
5.2	return 语句	164
5.3	函数调用和值传递	169
5.4	类型说明符 void	171
5.5	实例:初等统计	172
5.6	作用域规则	175
5.7	平行和嵌套程序块	177
5.8	存储类 auto	178
5.9	存储类 extern	179
5.10	存储类 register	182
5.11	存储类 static	183
5.12	静态外部变量	184
5.13	小结	186
5.14	习题	187
第 6 章	转移语句、按位表达式和枚举类型	194
6.1	goto 语句	194
6.2	break 和 continue 语句	196
6.3	switch 语句	198
6.4	条件运算符	199

6.5	按位运算符及其表达式	201
6.6	掩码	204
6.7	合并与分解	205
6.8	枚举类型	207
6.9	实例：“纸、石头、剪子”的游戏	210
6.10	小结	216
6.11	习题	217
第7章	指针、数组和串	224
7.1	指针	224
7.2	函数的指针参数	227
7.3	一维数组	229
7.4	指针与数组的关系	232
7.5	用作函数参量的数组	234
7.6	实例：冒泡排序	235
7.7	实例：合并与合并排序	237
7.8	串	241
7.9	实例：单词计数	246
7.10	多维数组	248
7.11	指针数组	250
7.12	main()的参量	255
7.13	不等长数组	256
7.14	小结	258
7.15	习题	259
第8章	递归、函数参量及预处理器	266
8.1	递归	266
8.2	实例：快速排序	270
8.3	实例：求第k个顺序元素	275
8.4	函数参量	276
8.5	实例：用二分法求函数的根	277

8.6	预处理器	281
8.7	实例:qsort()的应用	288
8.8	小结	290
8.9	习题	291
第9章	结构、联合和类型定义	299
9.1	typedef 的应用	299
9.2	结构	301
9.3	结构成员的存取	305
9.4	结构与函数	306
9.5	运算符的优先级和结合律:总结	310
9.6	实例:复数算术	311
9.7	实例:学生记录	314
9.8	联合	316
9.9	字段	319
9.10	结构的初始化	321
9.11	实例:玩扑克	322
9.12	小结	325
9.13	习题	326
第10章	结构和表处理	332
10.1	引用自身的结构	332
10.2	线性链表	334
10.3	表运算	336
10.4	一些表处理函数	340
10.5	栈	345
10.6	实例:波兰表示法和栈计算	348
10.7	二叉树	350
10.8	普通链表	355
10.9	小结	360
10.10	习题	361

第 11 章 输入/输出和 UNIX 环境	367
11.1 输出函数 printf()	367
11.2 输入函数 scanf()	372
11.3 相关联的函数 fprintf()、sprintf()、fscanf() 和 sscanf()	375
11.4 文件	376
11.5 ctype.h 中的宏	382
11.6 实例:压缩 C 程序	383
11.7 文件描述字 输入/输出	388
11.8 UNIX 中的重定向和管道传送	393
11.9 实例:打印目录	394
11.10 cc 编译器	399
11.11 C 验证器 lint	403
11.12 C 的整齐打印命令 cb	404
11.13 命令 make	405
11.14 C 的前景:结论性评注	406
11.15 小结	408
11.16 习题	410
附录 ASCII 字符码	413
索引 部分程序和函数	414

第 0 章 从 0 开始

在 C 语言里,0 是自然的起点。C 从 0 开始计数。C 用 0 代表假,而用非 0 代表真。C 数组的下标以 0 为下界。C 字符串用 0 作为串结束的标记。C 指针用 0 表示空值。本书将把你带入这样一个神秘的世界,使你享受到用 C 语言编程的喜悦。

C 语言是由贝尔(Bell)实验室的丹尼斯·里奇(Dennis Ritchie)设计的,于 1972 年在 PDP-11 上实现。当时 C 是为 UNIX¹ 操作系统设计的系统语言。UNIX 的研制者肯·汤姆逊(Ken Thompson)应用汇编语言(assembly)和命名为 B 的语言于 1970 年发表了 UNIX 的初版。为了克服 B 的局限性,C 语言应运而生。

B 语言起源于 BCPL,而 BCPL 是马丁·理查德(Martin Ritchards)于 1967 年发表的一种无类型系统程序语言。它的基本数据类型是机器字,指针和地址算术在该语言里用得很多。这跟结构程序的想法形成鲜明对照,结构程序是以使用象类似 ALGOL 语言那样的强类型语言为特点的。C 的发明起源于 B 和 BCPL,但又结合了类型概念。

C 是在这些语言的基础上发展起来的一种成熟的通用语言。它的定义在 1978 年由布赖恩·W.科宁汉(Brian W. Kernighan)和丹尼斯·里奇所著的《程序设计语言 C》(The C Programming Language)一书的附录“C 参考手册”(C Reference Manual)中阐述得很清楚。是近推荐的语言标准由丹尼斯·里奇于 1983 年 6 月在贝尔实验室文件《程序设计语言 C——参考手册》(The C Programming Language——Reference manual)中发表。本书所描述的 C 语言符合此文件推荐的标准。一些扩充的语言特点,例如枚举类型等,也已包括在内。

0.1 什么是 C 语言

C 是一种小型语言。在程序设计中,小即是美。C 所用的关键字,即被保留的字,比 Pascal 少,然而 C 却被公认为是比较强有力的语言。C 的强有力来源于精心选用的控制结构和数据类型。只要使用得当,它们几乎是不受限制的。正因为其规模缩减到最小,C 语言是易于学习的。

C 是 UNIX 自带的语言,而 UNIX 是最重要的交互式操作系统。语言的普及性并不取决于其本身的特性,系统环境才是语言成功的秘诀。例如,C 并不需要内部输入/输出结构或复杂的中断处理器,而是依赖于实现这些功能的库例程。

C 是可移植的。由于具有小的特点和最初定义在小型机 PDP-11 上,C 具有可移植性。语义常常是留给具体实现去决定,不适当地追求一致性会造成代码翻译的复杂化和高难度。代码可以容易地裁剪到新的宿主机上去。C 编译器能用少于 10000 条的 C 代码写成,并且只要用几个人月就可以移植到一个新的系统上。另外,系统实用程序和预处理器使程序员将可能出现的与机器有关的因素与主代码隔离开。这就容易在不同的 C 系统之间重新作出定义。

C 是简洁的。C 具有一套强有力的运算符。其中有许多反映出它的设计者的个人喜好和继承了原始环境中已有的内容。例如,加 1 运算符++在 PDP-11 机器语言中有着直接对应的符号。在表达式里,非直接引用和地址算术可以复合在一起,从而用一条语句或者一个表达式便可实现其它语言可能要用多条语句才能实现的功能。这样做,在有些人看来是十分简洁的,但对某些人来说却由于代码简短而感到难以理解。软件生产率研究表明,程序员平均一天仅能产生少量的工作代码。简洁的语言明显地提高了程序员的潜在生产率。

C 是模块式的。C 支持一种例程风格,即外部函数,它通过传值来调用参数。C 不容许函数嵌套。C 通过使用在文件内的静态存储类

提供有限的专用形式。这些特点加上典型的 UNIX 环境,就足以支持用户定义的函数库和模块式程序。

当然,C 也不是天衣无缝的。它不如其它较新的程序设计语言那样具有强类型。它允许编译器在表达式和参数表内重新排列求值顺序。它不具有自动数据边界检查。某些运算符,例如 * 和 = 等,具有多重用途。编程中常见的错误之一是在应该用运算符 == 的地方却用了运算符 =。

不管怎样,C 是一种优美的语言。它使程序员对机器的存取不受拘束。它所具有的不完善之处比起完全受限制要容易对付。C 以它的字谜般的字符和无拘无束的特性而引起人们广泛的注意。C 程序员应当努力做到函数的模块化和影响的最小化。应当鼓励 C 程序员多实践和交互式工作。确实,多实践和交互式工作正是本书的特色。我们的观点是:C 是微机 and 大型机种上均可使用的交互式编程语言。语言、系统和机器三者组成一个动态的统一体。只有通过实践,我们才能彻底地了解掌握它。作为一本书的开端,我们按照传统方式写出第一个程序,这一令人费解的写法也许会激发读者的好奇心。

/* * * 传统的第一个程序—令人费解的写法 * * */

```
main()
{
    union{
        char what[16];
        int cipher[4];
    }mystery, * p;
    p=&mystery;
    p->cipher[0]=0x6c6c8568;
    p->cipher[1]=0x77202c6f;
    p->cipher[2]=0x546c726f;
    p->cipher[3]=0x0000000a;
    printf("%s",p->what);
}
```

0.2 习题

1. 运行以上传统的第一个程序。注意该程序是与所用的系统有关的。它是为 VAX 机器设计的。通常,它将正确地运行在任一应用 ASCII 字符集的 32 位机器上。这个程序要等到学了第 9 章才能全部理解。当然,我们完全可以用不神秘的、适合于任一机器的方式写出一个等价的程序。