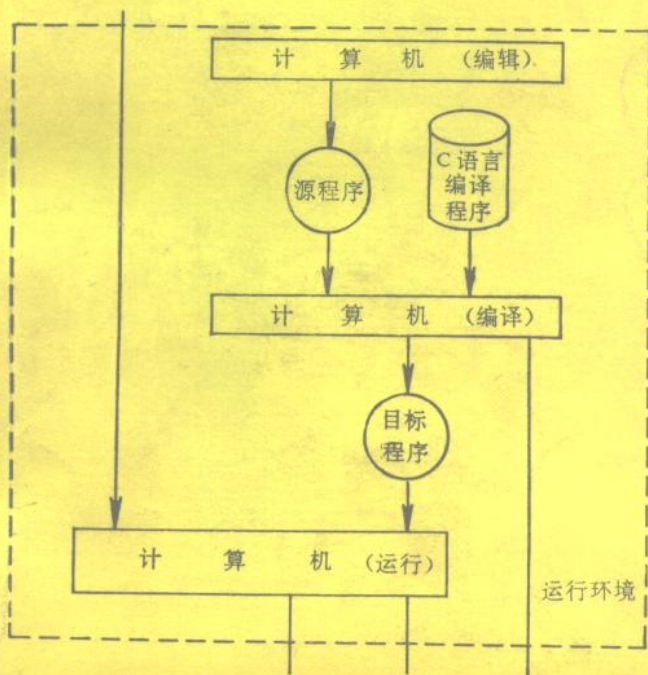


北京市高等教育自学考试用书

实用 C 语言程序 设计教程

孙玉方 张乃孝 编



北京市高等教育自学考试用书

实用C语言程序 设计教程

孙玉方 张乃孝 编

北京大学出版社

内 容 简 介

本书完整地介绍了C语言的概况、基本成分、程序设计方法以及与C语言有关的输入/输出功能及其运行环境。全书共分13章,在每章开头均有对该章主要内容的概括介绍,每章结尾有简明小结,其后附有习题。本书内容充实,例题丰富,突出实用,不只是讲述一般的概念,而主要是讲述C语言程序设计的方法和技巧,适合自学,便于读者较快地掌握C语言来解决实际问题。书末附有附录,给出了C语言程序设计中所要用到的一些子程序和库函数。

实用C语言程序设计教程

孙玉方 张乃孝编

责任编辑:李怀玺

*

北京大学出版社出版

(北京大学校内)

北京印刷三厂印刷

新华书店北京发行所发行 各地新华书店经售

•

850×1168毫米 32开本 15.5印张 400千字

1989年8月第一版 1991年8月第四次印刷

印数:7,001—15,000册

ISBN 7-301-00547-4/TP·005

定 价: 5.75 元

前 言

C语言诞生至今已近十五年，作为UNIX系统的孪生兄弟，它已受到国际计算机界越来越广泛的注意。1983年C语言的创始人D.M.里奇(Dennis M. Ritchie)与UNIX的主要开发者K.汤普逊(Ken Thompson)一起荣获了ACM图灵奖。评委对C语言作了高度评价，认为UNIX之所以有今天，主要是由于它选用了C语言作为描述工具，特别是由于C所具有的良好可移植性。它众多的特点已使它迅速成为人们进行程序设计特别是软件开发得心应手的工具，与FORTRAN, COBOL, BASIC和PASCAL一起作为最常用的语言之一。

鉴于UNIX系统在国内的日益普及及微、小型机在国内的广泛应用，对于C语言的学习和应用就成了极为迫切的事情。

计算机科学与技术自学考试委员会决定把C语言作为自学考试选考科目之一，并委托我们来编写这方面的教材。我们想，既然是自学考试教材，当然应该适合自学，而且自学者多数是要学以致用，所以我们尽可能全面而又系统地介绍C语言的有关成分，力求叙述条理化、用语通俗化，例子完整，说理清楚。在书的结构方面我们也作了一些不同于同类书的安排。比如，我们把整本书分成几部分，第一和第二章介绍C语言的概况，第三至第六章介绍C语言的基本成分，第七至第十章介绍C语言中较为复杂但仍较基本的内容，第十一和第十二章讲了余下的虽不是必要但仍很有用的内容，第十三章讲述了与C语言程序有关的输入/输出功能及其运行环境。对自学者来说，本书第一至第十章及第十三章的前一部分是最为重要的。

为了适应自学要求，在每章开头我们对该章的主要内容作一

概括介绍，每章结尾有一个简明的小结，其后附有习题，供读者练习和上机演示用。每章都给出了大量例子。这些例子我们都尽可能地上了机验证了，许多例子还有演示结果。

然而值得指出的是，本书虽是应自学考试委员会的要求编写的，但是并不是说本书只能为自学者所用。许多内容，比如指针、结构及表处理、预处理语句和输入/输出等方面所涉及的问题、所包含的内容是目前我们所看到的书中所不多见的。这样，即使对于水平较高的同志，我们相信他们也可以从中得到一些启示。本书之所以称为“实用C语言程序设计教程”，主要是要突出“实用”。它表明本书不只是讲述一般的概念，而主要是讲述C语言程序设计的方法和技巧，所以对于急于想掌握C语言来解决实际问题的读者来说，本书恰好是他可以信赖的朋友。

本书的主要内容如下：

第一章讲述C语言的发展历史和主要特点、基本语法规则及词汇。为了能上机实习，本章还对C语言程序的一般编写、编译和运行的方法、步骤作了介绍。

第二章是对C语言程序设计方法的一个速成介绍。结合例子引出了C语言的一些基本成分，这就为以后各章的深入叙述作了必要的准备。对于一些有其它语言程序设计经验的人来说，看完本章已经可以利用C进行一些基本的程序设计工作了。

第三章讲述C语言的基本成分之一——变量和常量，引入了C的类型概念，介绍了变量的基本类型：整型、字符型和浮点型。

第四章讲述C语言的各种运算符和与之有关的各种表达式。

第五章讲述C语言的主要语句：空语句，复合语句，if和switch条件分支，while、for、do-while循环以及goto、break和continue无条件转移语句。

第六章讲述C语言的主要组成机制——函数，包括函数定义、函数的调用及返回、函数的作用域规则和存储类。

有了上述一些知识就足以构造一些实用程序了。但是，为了应付较为复杂的局面，还必须掌握C中的构造类型和函数递归调用及其构成方法。接下去的几章主要介绍这些内容。

第七章讲述C中的数组和指针类型。数组和指针在C中是密切相关的两个概念，但对指针概念的理解却要花很大的功夫。本章举了大量例子来解释这一概念，并且在后续的章节中还不断地加以引伸和补充。

第八章讲述递归函数和递归调用。介绍了这种函数的构成及其调用过程，并对递归和迭代作了区分。

第九章和第十章讲述结构及其重要应用。结构是C中另一类很重要的构造类型。第九章介绍了其表示方法及意义、针对其成员的引用方法及有关操作。也阐述了它与函数、数组之间的关系。第十章涉及了结构的各种重要应用，如线性链接表、栈、队列和树等的操作，其中的许多例子和其处理方法都有明显的代表意义。

第十一章讲述C中另外一些构造类型：联合、枚举以及类型定义 typedef 的功能。

第十二章讲述C的预处理程序。这一部分虽然不是C语言最基本的成分，但却给C程序的编制带来极大方便。从某种意义上说，这也是C有别于其它语言的特殊之处。

第十三章讲述C语言程序中用到的输入/输出功能和系统调用命令。这一部分不是C语言的固有成分，但却是C语言程序中必不可少的。这些功能应由C语言的运行环境来提供。本书中我们以UNIX系统环境为例进行了解释。原则上所有这些使用方式和规则在任何支持C语言的系统上都是共同的。本章最后介绍了有关C语言程序编译更进一步的细节和预测试工具，这些内容对于掌握C语言程序设计方法是很有益处的。

根据最新的资料，附录中给出了C语言的输入/输出子程序、库函数和系统调用功能及基本使用方法。由于目前还没有一个公

认的标准，所以这些成分仅供参考，读者在使用时最好翻阅一下随机资料，以免出错。

最后是编写本书时所用到的主要参考资料。

本书第三至第六章由张乃孝同志编写，其余部分由孙玉方同志编写，全书由孙玉方同志做了整理。

本书的编写得到了杨芙清教授的指导，得到了北京大学出版社特别是邱淑清同志的帮助。俞士汶高级工程师在百忙中细心审阅了全书文稿，提出了许多宝贵而又具体的意见。在此对于上述同志表示衷心感谢。

由于编者水平所限，不当之处敬请有关专家和广大读者批评指正。

孙玉方 张乃孝

目 录

前 言	(1)
第一章 C语言的特点和基本语法规则	(1)
1.1 C语言的演变及特点	(1)
1.2 C语言程序的编写和运行	(5)
1.3 C语言程序的结构	(8)
1.4 C语言的词汇和语法规则	(10)
1.5 小结	(15)
习 题	(16)
第二章 C语言程序设计初步	(18)
2.1 变量、表达式和赋值	(18)
2.2 预处理语句 #define 和 #include	(21)
2.3 格式输入/输出函数 printf 和 scanf	(24)
2.4 控制流及语句	(28)
2.5 字符输入/输出函数 getchar 和 putchar	(35)
2.6 函数	(40)
2.7 数组、字符串和指针	(43)
2.8 小结	(53)
习 题	(54)
第三章 变量与常量	(57)
3.1 变量的说明	(57)
3.2 整数类型	(58)
3.3 浮点类型	(61)
3.4 字符类型	(64)
3.5 常量	(68)

3.6	类型的长度和 sizeof 运算符	(71)
3.7	小结	(72)
	习 题	(73)
第四章	运算符与表达式	(75)
4.1	表达式、语句和赋值	(75)
4.2	类型转换和强制运算符	(79)
4.3	算术运算符	(81)
4.4	增 1 和减 1 运算符	(86)
4.5	赋值表达式	(89)
4.6	关系运算符和逻辑运算符	(90)
4.7	移位运算符	(94)
4.8	表达式的求值	(97)
4.9	小结	(99)
	习 题	(102)
第五章	语句与控制流	(107)
5.1	复合语句	(107)
5.2	条件分支语句	(110)
5.3	循环语句	(117)
5.4	无条件分支语句	(124)
5.5	释例	(128)
5.6	小结	(131)
	习 题	(132)
第六章	函数	(136)
6.1	函数定义	(136)
6.2	返回及 return 语句	(140)
6.3	函数的应用	(145)
6.4	作用域规则	(150)
6.5	存储类	(153)

6.6 小结	(161)
习 题	(162)
第七章 数组和指针	(167)
7.1 一维数组	(167)
7.2 释例: 数组的应用	(171)
7.3 指针	(174)
7.4 指针运算	(184)
7.5 指针与数组	(187)
7.6 指针、数组与函数	(191)
7.7 字符数组与字符串处理	(203)
7.8 多维数组	(209)
7.9 指针数组与命令行参数	(211)
7.10 不规则数组	(216)
7.11 小结	(218)
习 题	(219)
第八章 递归函数与递归调用	(227)
8.1 递归函数与函数的递归调用	(227)
8.2 递归与迭代	(231)
8.3 函数用作自变量	(241)
8.4 小结	(247)
习 题	(248)
第九章 结构	(253)
9.1 结构的表示和意义	(254)
9.2 结构成员的引用及操作	(257)
9.3 结构与函数	(263)
9.4 结构与数组	(277)
9.5 结构置初值	(283)
9.6 小结	(293)
习 题	(294)

第十章 结构及表处理(296)

10.1 引用自身的结构(296)

10.2 线性链接表及其操作(299)

10.3 栈及其操作(310)

10.4 队列及其操作(318)

10.5 树(320)

10.6 小结(330)

习 题(331)

第十一章 联合、枚举和 typedef(335)

11.1 联合(335)

11.2 枚举类型(340)

11.3 类型定义(351)

11.4 位段(355)

11.5 小结(361)

习 题(362)

第十二章 预处理程序(367)

12.1 宏定义与宏替换(367)

12.2 文件包含(379)

12.3 条件编译(381)

12.4 行号控制(386)

12.5 释例(387)

12.6 小结(390)

习 题(391)

第十三章 输入/输出和 UNIX 系统环境(393)

13.1 标准库与标准输入/输出文件(394)

13.2 标准文件的字符输入/输出和格式输入/输出(395)

13.3 输入/输出改向和管道线(408)

13.4	一般文件输入/输出.....	(411)
13.5	字符串输入/输出及其它.....	(422)
13.6	前导文件ctype.h 及其操作.....	(426)
13.7	低级输入/输出和 UNIX 环境	(432)
13.8	释例: 用系统调用来编制程序	(441)
13.9	C 语言程序的编译和检验	(452)
13.10	小结.....	(456)
	习 题	(457)
附 录		(460)
	附录 A 操作系统服务子程序.....	(460)
	附录 B 输入/输出及通用库函数	(466)
	附录 C 数学库函数	(480)
参考文献		(483)

第一章 C 语言的特点和基本词法规则

本章对 C 语言的演变过程和特点作一概括介绍,以便使读者对 C 语言有一个总的印象。本章还对 C 语言的基本词汇和词法规则,C 语言程序的一般编写、编译和运行过程作了说明。我们相信,通过与机器对话和上机编程试验,可以大大加速你的学习进程,并为以后各章的学习打下基础。

1.1 C 语言的演变及特点

C 是一种程序设计语言,它最早是由贝尔实验室(美国电话电报公司 AT&T 属下的一个研究机构)的 D.M.里奇设计、并在 PDP-11 机上实现的。

C 的设计和实现与 UNIX 操作系统有着不可分割的密切关系,它最早就是为 UNIX 服务的,并一直作为 UNIX 的主力语言。

UNIX 系统是一个通用的、交互式的计算机软件系统。自 1970 年诞生以来,它经历了发展、完善和广泛应用的过程。目前它已在 16 位高档多用户微机上占了统治地位,而且肯定成为 32 位微机的主要操作系统,并可能进而占领小型机、中型机甚至一部分大型机的市场。UNIX 系统获得如此巨大的成功,是出乎其设计者的意料之外的。究其原因,最根本的是其可移植性,而这又是由于用 C 语言作为描述工具的缘故。

本小节简述 C 的演变及其特点,以便使读者从中领悟到 C 的一些本质和独到之处。

1.1.1 C 语言的演变

UNIX 系统的早期版本是用汇编语言写的。但是, 由于汇编语言的不可移植性, 以及描述问题效率低、编程困难、难于修改调试等原因, 促使 UNIX 的开发者 K. 汤普逊开发一种高级语言来描述 UNIX 系统。1970 年, K. 汤普逊实现了一种叫 B 的语言, 并用 B 写了 UNIX 操作系统和绝大多数上层实用程序。B 语言的主要思想源于 BCPL 语言, 并且和 BCPL 有许多类同之处, 比如都没有数据类型或者说只有一种数据类型 (机器字)。BCPL 是 M. 理查德 (M. Richards) 基于 CPL 语言在 1967 年提出的一种语言。而 CPL 又是基于 ALGOL 60 于 1963 年提出的。

由于 B 依赖于机器, 又无数据类型等原因使它未能流行开来。为了克服 B 的局限性, 里奇开始开发新的语言, 它就是 C, 1972 年 C 已正式投入了运行。图 1.1 反映了 C 的发展历史。

ALGOL 60 (1960 年提出)

↓

CPL (Combined Programming Language, 1963 年提出)

↓

BCPL (Basic Combined Programming Language, 1967 年提出)

↓

B (Ken Thompson, 1970 年实现)

↓

C (Dennis Ritchie, 1972 年实现)

图 1.1 C 语言演变历史

从其演变历史来看, C 属于 ALGOL 语言族系, 然而, C 有它的一些特色, 相对于算法语言、数据处理语言来说, 它更象一种系统程序设计语言。

的确, C 原本就是为了描述 UNIX 系统的。C 语言第一次大显身手是在 1973 年。这一年, UNIX 系统用 C 重写了一遍。系统的代码量比以前的版本大了三分之一, 加进了多道程序设计功

能，特别是整个系统（包括C语言编译程序本身）都建立在C语言基础上。C语言良好的可移植性为UNIX系统的移植打下了坚实的基础。所以可以说建立在C语言基础上的UNIX第五版（V₅）开创了UNIX系统发展的新局面。以后UNIX V₆，V₇，System III和最新的System V都是在V₅基础上发展、扩充的。

虽然最初的C语言是附属于UNIX系统并且在PDP-11上实现的，但是，目前C语言却独立于UNIX系统，独立于PDP-11机而蓬勃发展。它适应的机种从8位微型机（比如以Z80为CPU的Cromemco）到Cray巨型机。它依附的操作系统从8位微型机上的单用户CP/M到大型机的IBMV₅/370。它与FORTRAN，PASCAL等语言一样已经成了微、小、超小、大、超大和巨型机上共同使用的语言。

1.1.2 C语言特点

C有很多比较突出的优点，概括起来有：

1. 语言简洁。C语言是一种小型语言。在程序设计中“小”常常意味着“漂亮”。C的关键字比PASCAL少（在PASCAL中关键字称作保留字），但功能上却毫不逊色。C语言的简洁可从两方面讲：从使用者角度，它采用的表示法尽量简洁，如以{、}代替通常的begin、end作复合语句或函数体的“括号”，以++表示加1，--表示减1，运算符省写等等。从语言内部实现角度，它本身没有I/O设施，也没有并行操作、同步或协同程序等等复杂控制结构，这些成分由支持环境提供，通过显式函数调用来实现。

2. 表达能力强。它可以直接处理字符、数字、地址，可以完成通常要由机器指令来实现的普通的算术及逻辑运算。它反映了当代计算机的性能，所以有效到足以取代汇编语言来编写各种系统软件和应用软件。最明显的例证是UNIX系统。UNIX系统的核心、shell命令解释系统以及外层大量实用程序中，除了核心

内部的上千行代码（整个核心的5%以下）因为效率、机器表示等原因需要用汇编语言写外，其余的都是用C描述的。此外，它还成功地描述了INGRES, INFORMIX, UNIFY, dBASE III 数据库管理系统, Logo, Prolog 解释系统等。这些都生动地表明了C在开发软件时所具有的强大生命力。

3. 具有数据类型构造能力，并具有很强的控制流结构。它可以在基本类型（如：字符、整数、浮点数等）的基础上按层次方法逐层构造各种构造类型（如：数组，指针，结构和联合等），所以它的数据类型较为丰富。另一方面它的各种控制语句如 if, while, do while, switch 等，功能很强，足以描述结构良好的程序。

4. 模块化结构。C支持一种子程序结构，即外部函数。这种函数通过传值来调用。C语言不允许函数嵌套定义。它通过文件中的静态存储类允许使用某种形式的专用权。这种技巧连同典型的UNIX环境一起很容易支持用户定义的函数库及模块化程序设计。

5. 语言生成的代码质量高。相对汇编语言而言，许多高级语言的代码质量要低得多，所以，直到现在汇编语言仍是编写系统软件特别是操作系统、编译系统等软件的主要工具。但是C语言则不然，许多试验表明，针对同一个问题，用C语言描述，其代码效率只比汇编语言低10—20%。由于用高级语言描述比用汇编语言描述问题编程迅速，可读性好，特别是C语言还比较容易移植，而其效率又降低不大，所以C就成了人们描述系统软件和应用软件比较理想的工具。

6. 可移植性较好。这是指程序可以从一个环境不加或稍加改动就可搬到另一个完全不同的环境上运行。汇编语言因为依赖于机器硬件，所以根本不可移植，而一些高级语言，如FORTRAN等的编译程序也不可移植。目前许多不同机器上几乎都配有FORTRAN，但这基本上都是根据国际标准重新实现的。而C语

言目前在许多机器上出现，大部分却是由C语言编译移植得到的。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。C语言的编译程序便于移植，就使一个环境上用C编写的许多程序可以很方便地移到另一个环境上。这也就是当前众多的微型、小型机上流行UNIX和C的主要原因之一。

当然，正如其它语言一样，C也并不是没有缺点。比如：它不如现在的一些程序设计语言（如PASCAL, Ada）的类型性强，类型转换比较随便，不够安全；它允许编译程序在表达式中重新安排计算顺序和参数表；它没有动态数组界的检查。运算符优先级太多，容易混淆。但是这些缺点和它的许多长处比较起来是次要的问题，并不影响它成为一个富有生命力的杰出的描述工具。

一种计算机语言不可能包罗万象，也不可能所有领域中都适用。在商业程序设计、科学程序设计、系统程序设计和应用系统程序设计这四个主要领域中，C语言不太适用于商业数据处理（在这个领域里，COBOL语言可大显神威），在科学计算方面也不够理想（FORTRAN是这个王国中的骄子），但在后两个领域中，C是强有力的不可多得的工具。

1.2 C语言程序的编写和运行

程序设计就是把算法传送给计算机的一种活动，简单说来，算法是一个其步骤被完全确定了计算过程。

解决问题的过程从问题的说明开始，包括一个正确的执行程序。对于计算机来说，程序是解决问题的具体步骤，它需要用计算机可以理解的语言（比如C语言）来描述。下面是解决问题的具体过程：

- (1) 说明问题；
- (2) 找出解决问题的某种算法；
- (3) 用C语言（或别的程序设计语言）描述该算法；