孟庆昌 孙玉方 编

C语言程序设计

C语言程序设计

上海交通大学出版

C 语 言 程 序 设 计

孟庆昌　孙玉方　编

上海交通大学出版社

内 容 简 介

C 程序设计语言是一种普遍应用于大、中、小、微型计算机的计算机语言，具有简洁、实用、代码质量高，特别是可移植性好等特点。

本书是作者在广泛收集参考有关资料的基础上，结合自己讲授和使用C 语言的体会，根据教学实践和学员们的反馈意见，对初稿全面修改而成。全书共分七章，内容包括：C语言概述，数据、表达式和赋值语句，语句与控制流，函数与程序结构，数组和指针，结构和联合，输入/输出及C程序与UNIX 系统的接口等。

为便于教学和读者阅读，书中各章都配有习题和上机实习题，并给出了大量的实例，其中许多是完整的，可以直接上机运行。

本书可作为高等院校计算机专业的教材，也可作为计算机科技工作者应用C语言进行软件开发的实用参考书。中国软件行业协会高档微机协会为本书摄制了教学录像片。

C语言程序设计
交通大学出版社出版
（淮海中路1984弄19号）
新华书店上海发行所发行
北京市振华印刷厂 印刷厂印装

开本 787×1092 1/32 印张 14.875 字数 33.5千字
1987年8月第一版1987年9月第一次印刷
印数：1－5000册 定价： 3.50元
书号：ISBN7－313－00042－1/TP31
科目158－288

前　言

随着 UNIX 分时系统在国际上的广泛流行，近年来 C 程序设计语言在软件工程领域里颇为引人注目。

C 语言最早是用来描述 UNIX 操作系统及其上层软件的，但目前它的应用已不限于 UNIX 系统，而在多个操作系统和多种计算机硬件上实现了。实现它的机种从 8 位微型机到最高档的超高速机 Cray，遍及微、小、中、大型计算机。尽管 C 语言在精确性和严密性上尚不如某些语言（如 Pascal、Algol 68），但由于它具有简洁、实用、代码质量高，特别是可移植性好等特点，所以受到计算机用户特别是软件工程技术人员的赞赏，并得到越来越广泛的应用。

这些年我们在学习、研究 UNIX 系统的同时，在学校和有关讲习班里多次讲述了 UNIX 核心结构、UNIX 操作系统原理及 UNIX 使用等，深感熟悉 C 语言对研究和开发 UNIX的重要性。为了更好地提高教学质量，帮助广大用户更好地使用和开发 UNIX 系统，我们参照有关资料，结合自己讲授和使用 C 语言的体会，把手稿加以整理，形成了初稿，由北京信息工程学院和有关讲习班采用。根据教学实践和学员们提出的意见，我们对初稿进行了全面修改，特别是在后几章，补充了许多更为实用的例子，对指针、结构等 C 语言中的难点作了更为深入、系统的剖析和归纳，以期使本书更为完善。

本书共分七章，后面有若干个附录。

第一章简述C语言的发展历史、特点，并且结合简单例子对C语言的编写、编译和运行全过程作了概括介绍，目的是让读者熟悉C语言运行环境，以便在后面的学习中通过上机实习来加深对C语言的了解。

第二章介绍C语言的基本概念：标识符、变量、常量、数据及变量的简单类型。表达式是语言的基本组成部分，在本章中对各种主要表达式都作了介绍。在表达式基础上介绍了一种最简单最基本的语句，即赋值语句。最后介绍C语言众多的运算符，给出其优先顺序。

第三章介绍C语言主要的语句和控制流：条件语句、开关语句、各种循环语句以及各种转移语句。

第四章介绍构成C程序的主要成分——函数。围绕函数介绍了函数结构、函数的返回类型、参数和递归调用。联系到函数，我们介绍了C语言特有的变量存储类。最后介绍了C语言的程序结构和预处理程序。

除了第二章介绍的几种基本数据类型外，第五章和第六章还专门介绍了几种复杂的数据类型。

第五章介绍指针和数组，主要包括它们的定义、使用、相互关系及它们与别的语言成分（函数、其他类型的数据）之间的联系。

第六章重点介绍结构和联合，包括结构的组成、其成分的访问，结构数组与指针，引用自身的结构及其所构成的多种数据结构，联合的定义、联合与结构之间的不同点，以及枚举类型。

最后一章介绍的内容虽然不是C语言的成分，但却是使用C语言完成工作所必需的，主要包括输入输出以及C语言使用与UNIX系统（主要是文件系统）的关系。

后面的附录列出了C语言作者Ritchie所给出《C语言参考手册》，一部分编写C语言程序所用到的标准函数和系统调用，C语言编译程序在若干微机上的实现。最后列出了本书编写时所参考的主要资料。

本书编写过程中，我们注意说理与实例并重，以实例作为说理的基础，以使读者对一些概念加深直观理解，而说理又把实例中的现像加以归纳和提高。本文是从实用角度来讲述C语言的，由于各种条件限制，有些说理部分可能不很严格。我们认为学习语言的最好方法是实践，所以本书中的许多实例都是完整的，可以直接上机运行。我们还给出了许多习题，希望广大读者尽量多做习题、多上机实习，通过练习来巩固自己所学的知识，为以后用C语言开展工作打下基础。

本书可作为高等院校计算机程序设计语言的教材，也可作为广大计算机科技工作者应用C语言进行软件开发时实用的参考书。

本书是在《计算机研究与发展》编辑部1985年出版的《C语言及其程序设计》（孙玉方、孟庆昌著）的基础上修编而成，其中前三章和附录C由孙玉方修订，其余各章及附录由孟庆昌修订，并由孟庆昌统一整理。

本书的修编和出版过程中，得到过徐国平、董洪皋等同志的帮助和支持，在此一并表示衷心的感谢。

作者　1986年10月

于北京

目　录

第一章 C语言概述

1.1 C语言历史和特点

1.1.1 C语言的演变历史

C语言是UNIX系统的主力语言，它与UNIX系统有着互相依存、休戚与共的紧密关系。由于汇编语言的不可移植，以及描述问题的效率不如高级语言，特别是可读性差，所以汤普逊（K.Thompson）决定开发一种高级语言来描述UNIX系统。1971年，汤普逊在PDP-11/20上实现了B语言并用B语言编写了UNIX操作系统和绝大多数实用程序。B语言主要思想源于BCPL语言。BCPL语言是理查德(M.Richards)基于CPL语言在1969年发表的一种语言，它是一种单一数据型语言，至今仍具有一定的生命力。不过，由于B语言解释执行速度慢以及不适应PDP-11的编址方式等原因，所以没有盛行起来，而导致里奇（D.M.Ritchie）开发一种新的语言，这就是后来的C语言。里奇用了一年时间写了第一个C语言编译程序。1972年C语言便投入了使用。

1973年汤普逊和里奇把UNIX系统用C语言重新书写了一遍，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）建立在C语言基础上，而C语言又具有良好的可移植

性，所以第五版的UNIX系统（UNIX V_5）就奠定了UNIX系统的基础。以后UNIX V_6，UNIX V_7，System Ⅲ 和最新的System V，都是在UNIX V_5基础上发展、扩充的。

今天UNIX系统已在全世界范围内取得了巨大的成功，它几乎成了16位微型机的标准操作系统，可以断言，它也肯定能成为32位超级微型机上的主流操作系统。目前在许多小型机、超级小型机，甚至像IBM370、4300、UNIVAC、Honeywell等大型机上也有它的踪迹。从某种角度可以说，没有C 语言就没有 UNIX 今天的巨大成功；当然，没有UNIX，C语言也不会有今天。

虽然最初的C语言是附属于UNIX系统且在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。它附着的操作系统可从8位微型机上的单用户CP/M 直到大型机的 IBM VS/370。它与 Fortran、Pascal 等语言一样已经成了微、超小、小、大、超大和巨型机上共同使用的语言。

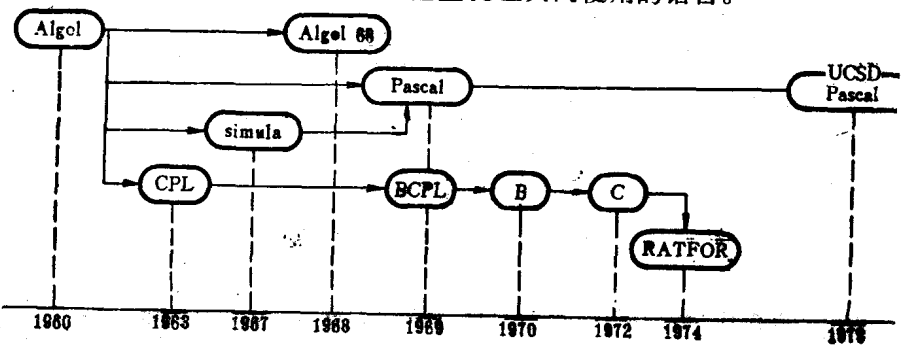

图 1-1 Algol 60语言、Pascal语言与C语言的关系

人们喜欢把Pascal 语言与C 语言进行比较，其实它们都是Algol 语言系统的发展，并各具特 点。图 1-1 列 出了Algol语言属系的一些主要语言的演变情况。

1.1.2 C语言的特点

C 语言的特点是多方面的，大致可归结为以下几点：

（1） 语言表达能力强，它可以直接处理字符、数字、地址，可以完成通常要由硬件来实现的普通的算术及逻辑运算。它反映了当前计算机的性能，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系统主要分三层：核心（操作系统）、与外层的接口口shell 命令解释程序以及外层大量的子系统，包括各种软件工具、实用程序和应用软件。这些程序中除了核心内部的1000 行 左右（整个核心 的 10% 以下）因为效率、机器硬件表达需要用汇编语言写外，其余的都是用C语言描述 的。当然C 语言同样具有表达数值处理、字处理功能，所以它又是一种通用语言。

（2） 具有数据类型构造能力及很强的控制流结构。在基本类型(如：字符、整数、浮点数等)的基础上，它可以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。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。此外，它的有些存储类（ 如静 态 static，外部 extern ）在功能上有助于数据隐藏的模 块 化 结构程序设计。

（8） 语言本身简洁，编译程序小。C 语言除了在表示法上尽可能简洁（ 比如，以{、}代替通常的begin、end 做

复合语句括号，运算符尽量缩写等）以外，语言的许多成分，都通过显式函数调用来完成，比如C语言中没有I/O设施，也没有并行操作、同步或协同程序等复杂控制。另一方面，它在运行时所需要的支持少，占用的存储空间也小。

（4） 语言生成的代码质量高。高级语言能否用来描述系统软件，特别像操作系统、编译程序等，除了语言表达能力以外还有一个重要因素是该语言的代码质量。如果代码质量低，系统开销会增大，实际上不可行，所以直到现在汇编语言仍是编写系统软件的主要工具。许多高级语言相对汇编语言而言其代码质量要低得多，但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%。而由于用高级语言描述比汇编语言描述问题编程迅速，可读性好，特别是C语言移植后效率降低不多，所以C语言就成了人们描述系统软件和应用软件比较理想的工具。在代码效率方面的确可以和汇编语言媲美。

（5） 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语言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语言，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语言编译程序80%的代码是公共的。由于C语言的编译程序便于移植，使得在一个环境上用C语言编写的许多程序可以很方便地移到另一个环境上。

C语言的优点很多，但也有一些不足之处，例如运算符

优先级太多，不便于记忆，有些还与常规约定有所不同（如字位与、或级别较低）；类型检验太弱，转换比较随便，所以不太安全（当然，目前有一个预处理程序lint，它主要用来检验类型协调匹配，帮助解决移植中的一些问题）。尽管如此，C语言仍不失为一个实用的通用程序设计语言，特别是一种强有力的系统程序设计语言。我们说它实用，是指它表达能力很强。C语言在理论研究方面可能不如Pascal、Algol 68等语言，但从出产品的实用角度来看，C语言要比它们强有力，这与Algol60和Fortran语言之间的关系有相似之处。由于上述的几个突出优点，人们对C语言倾注越来越多的关心，以至在世界上使用、研究C语言的人数正以爆炸般的方式迅猛扩大。

1.2 C语言的一般介绍

本节以若干小型例子给出C程序的一些概貌，使读者对C程序有一个初步的印象，便于后面几章深入地开展对于C语言及用C语言进行程序设计的讨论。另一方面，对于C语言程序编制、编译、运行的过程也作一概括介绍，以有助于读者上机进行实际练习。

1.2.1 词汇表

任何一种高级语言，都需要有自己的基本词汇表，C语言也不例外，其基本词汇表由下列几部分构成：

数字：0，1，2，…，9；

英文字母：A、B、C、…、Z，a、b、c、…、z；

由于UNIX系统喜欢小写字母，在内部处理时往往以小

写字母为主，所以仅有大写字母的终端在使用UNIX时会遇到困难。

下线字符：_，这个下线符起一个英文字母的作用，所以在构成标识符等语法成分时，以下线“_”打头，在C语言中是合法的，而在通常的语言中是不允许的。

特殊符号，主要包括下列一些运算符和关键字：

①运算符：+、-、*、/、%（取余）、=（赋值）、<、>、<=、>=、!=（不等于）、==（全等比较）、<<（左移）、>>（右移）、&（逻辑位与）、|（逻辑位或）、&&（合取）、||（析取）、^（逻辑位异或）、~（按位求反）、（、）、[、]、→（指向）、.（成员）、!（反）、?：（三元运算符）等（详见表2-4）。

② 关键字：int、char、float、double、struct、union、long、short、unsigned、auto、extern、register、static、typedef、goto、return、sizeof、break、for、continue、if、else、do、while、switch、case、default、enum。

在某些具体实现中还保留了如下两个字：fortran和asm。

下面几个字虽然不属于关键字，但建议读者把它们看成关键字，而不要在程序中随意使用，以免造成混淆，这些字是：define、undef、include、ifdef、ifndef、endif及line。这些字主要用在C语言的预处理程序中。

在C语言中，关键字都有固定含义，不能作为一般标识符使用。

如同自然语言及编织、音乐等活动中的专用语言一样，C语言也有它特定的语法规定。利用基本词汇表中的一些符

号和关键字，按照给定的语法规则，就可以进一步构造其他符号、语句和程序。在程序中不允许出现上述词汇表中没有的符号，比如，不允许将二数相乘的乘号 * 写成×，也不允许出现违反语法规则所构成的符号。另外，在C 语言中有些符号在不同的上下文中具有不同的含义（至少是表面上），比如 *，在变量的类型说明中，可能解释成“某某变量是一个指针”，即 * 是指针类型的标志；但在赋值语句赋值号的右边，它又表示“把某某变量的内容赋给左边的量”，即 * 又是“变量内容”的标志。这些地方需要特别小心，必要时我们会时常提醒读者注意。

1.2.2 C 语言程序结构

（1） C 程序和主函数 一个计算机的程序主要由两部分内容组成：一部分是关于程序所要实现的算法描述，这种描述一般由一系列语句组成，而这些语句又可能按描述对象的要求以及语言本身的规则构成复合语句、分程序或函数、过程等；另外一部分是关于这些算法所要操作的对象（通常用数据来表示）的描述。在程序中出现的数据，可以是常量，也可以是变量。对数据的描述就是对程序中所用到的常量和变量进行相应说明，特别是对变量要进行类型说明。下面举几个简单例子，说明如何用C 语言来编写程序以及C 程序的基本结构，使读者对它有一个直观的了解。应注意，与 Basic等语言不同，C语言中是不允许有行号的。

〔例1.1〕

```
/* small.c, The smallest C Program */
main(  )/* There is no executable code */
{
}
```

针对这一例子说明以下几点：

① C程序一般由一个或多个函数组成，这些函数可驻留在一个或几个源文件中，这些文件都以.c结尾。比如本例中，C语言程序只有一个函数，并且驻留在一个文件small.c中。组成一个程序的若干函数中必须有一个且只能有一个名为main的函数，可运行的C程序总是从main开始执行。一个函数在其名字之后一定要有一对圆括号“（”和“）”，这是函数的标志。圆括号中放参数，参数可有可无，根据具体情况而定，如本例中就没有参数。

② 程序中以“/*”开头到“*/”结尾所表示的意义是一个注释。注释帮助读者阅读和理解程序，但在编译时，注释行是要忽略掉的，即，它不产生代码行。注释在各种语言中都是希望有的，甚至是越多越好。注释行可以在程序的开头，这一般说明整个程序的功能和注意事项；也可以插在程序语句行中或在某行语句的尾部，主要用来说明一段程序的功能或某一行程序的作用。在本例中，程序开头及第一行后都有注释。要注意，C语言中注释不能嵌套。

在例1.1中有一对花括号，在这里花括号可以看成程序体括号，类似于Pascal等语言中的“begin”和“end”，这里不过是一种简略表示，它也可以用来括起任何一组语句，从而构成一个有意义的复合语句或分程序。花括号中可以有任何意义的语句，包括空语句在内。要注意在一个函数中至少要有一对花括号，也就是程序体括号，而不管程序体是否为空（就如本例那样）。本例是一个有意义、正确的C程序，实际上它什么也没有干。

上述程序也可以写成一行。

〔例1.2〕