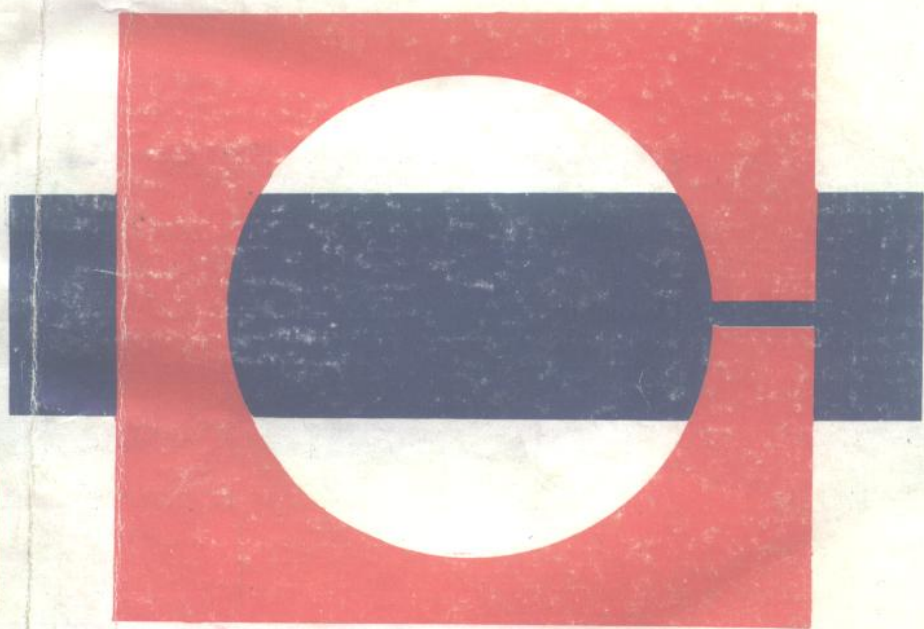


实用 C语言及其程序设计

鲁 岩 编 著



大连理工大学出版社

(辽)新登字 16 号

JS176/03

内 容 提 要

本书以 Turbo C 和 MicrSoft C 编译系统为背景,全面系统地、深入浅出地描述了 C 语言的基本知识及其程序设计方法。尤其是书中通过大量实例对 C 语言中的学习难点(如函数、指针、结构、与汇编语言的接口、图形设计等)以及编程中容易犯的错误均作了较全面的介绍,并配有详细的附录供读者在学习 C 语言和程序设计时参考。

本书可作为大专院校计算机专业和非计算机专业的教科书,也可作为计算机人员的培训教材以及工程技术人员的自学教材和参考书。

实用 C 语言及其程序设计

Shiyong C Yuyan Jiqi Chengxu Sheji

鲁 岩 编 著

大连理工大学出版社出版发行 (邮政编码:116024)

大连金州印刷厂印刷

开本:850×1168 1/32 印张:14.75 字数:366 千字
1993 年 1 月第 1 版 1993 年 1 月第 1 次印刷
印数:1—6000 册

责任编辑:方延明

责任校对:寸 土

封面设计:姜严军

ISBN 7-5611-0679-3/TP·39

定价:7.00 元

前 言

随着 UNIX 操作系统在国际上的广泛流行, C 语言在计算机系统开发和应用领域已占据了主导地位, 以致于有些计算机专家这样来评价 C 语言: “不会使用 C 语言, 就等于不是真正懂得计算机”。

C 语言的声望如此之高, 能够在微、小、中和大型计算机上迅速得到普及, 并已成为个人计算机用户所使用的最重要的程序设计语言, UNIX 操作系统的成功固然是一个重要的方面, 但其最根本的原因还在于 C 语言本身所独具的特性。C 语言被认为是一种能高上去又能低得下来的“中级语言”, 一方面, 用 C 语言可以编制其它各种“高级语言”所能编制的各种应用程序, 因此, 它具有“高级语言”的特点; 另一方面, 它又具有“低级语言”的功能, 用它编制“汇编语言”所能编制的绝大部分程序。除此之外, C 语言还具有结构清晰、数据类型丰富、语句精练、灵活、效率高、表达能力强、书写简练, 以及可移植性好等许多优点。在我国, 大部分高等院校的有关专业都开设了 C 语言课程。

但是, 人们在学习使用 C 语言的过程中, 往往会感到有些困难, 归纳起来有四点: ① C 语言不同于其它高级程序设计语言, 它的结构化强, 并具有与汇编语言打交道的能力, 以及以函数调用为主的程序设计风格, 所以初学者不大习惯这种语言的编程方法。② C 语言中出现了一些在其它语言中所没有的数据类型, 如指针型、结构型、联合型等; 引进了预处理程序的概念; 特殊的 I/O 操作等问题。这些都是 C 语言的重要贡献, 有些比较难于理解和掌握。③

由于 C 语言的类型检验功能较弱,转换比较随便,指针使用灵活等,所以有时不太安全,编程容易出错,往往在调试程序过程中感到有些费力,并且出错比较难查。④虽然目前有关 C 语言方面的书籍已有不少,但是能从学习方法上给读者提供一套从初级到高级的 C 语言编程技术方面的书还太少。目前一般 C 语言书的内容组织结构大致有如下几种:一是入门方面的书。这种书虽然通俗易懂,介绍了 C 语言方面的基本知识,但深入学习,则显内容不足。二是比较规范的 C 语言程序设计方面的书。其特点是内容较全,面面俱到,但对于一些问题特别是难点问题的解释比较浮浅。在大多数情况下,这类书并不提供一个实际的上机语言环境,很难使读者通过边上机边学习来掌握 C 语言的内容。三是属于专题性质方面的 C 语言书。如,C 语言图形设计、C 语言窗口技术、C 语言算法、C 语言题解等。这些书大部分是由国外书籍翻译而来,并直接面向某个应用方面,层次较高,一般都是大块大块的程序罗列,没有一定 C 语言编程技术训练和 C 语言基础知识的人是不太容易看懂和掌握的。

综上所述,C 语言虽然是一门灵活、实用的程序设计语言,但学习起来并非容易之事。为了解决这个问题,作者总结了近年来在 C 语言程序设计和教学方面的经验编写成本书。出版本书旨在为广大读者提供一本能够深入浅出地系统学习 C 语言的基本知识及其开发环境,熟练掌握 C 语言程序设计方法与技术的参考书。使初学者既能很快入门,又能深入了解 C 语言;使有一定 C 语言知识的读者能从中更进一步地掌握 C 语言的性能,加深对 C 语言中一些难点的理解。书中提供了一些实用的例子来解释 C 语言中的各种语法现象、使用规则、难点及其编程技巧,尽可能使它们容易理解和掌握。

本书希望能够形成如下一些特点:

- 对 C 语言程序设计内容和程序设计技术有全面完整的介

绍和总结。

- 在介绍 C 语言内容的同时介绍一种 C 语言编译系统的使用(本书选择 Turbo C),使读者能边学习边上机,有利于掌握书中的内容。

- 所选取的实例能全面地反映 C 语言中各种语法现象,从初级到高级,有利于读者在学习中掌握 C 语言的要点和难点。

- 对 C 语言程序设计中经常容易出现的错误,在每一章节都有介绍,以引起读者的足够重视。尤其对指针、函数、结构以及文件、图形等操作都做了较详尽的说明和应用举例。

- 在讲解 C 语言的基本内容基础上,注重引导读者能运用这些知识编写出较大规模的 C 语言程序系统,同时介绍了在 C 语言程序设计中常用的一些技术与处理方法。

- 有较详细的附录供读者查阅参考。

书中全部实例均在 Turbo C 2.0 版本下运行通过。除第十四章中的部分内容外,其余各章中的内容与当前在微机上流行的 C 语言版本(如 Micro Soft C)是通用的。

本书中有关作者的研究和开发成果是在国家自然科学基金青年基金项目资助下所取得的。

大连理工大学系统工程研究所曲晓飞副教授积极倡导本书的编写,并对编写大纲和编写内容提出了宝贵的建议;荣兰英老师对全部书稿进行了誊清和校对;袁帅与黄宜卓两位老师也给本书提出了不少的修改意见。在此谨向他们表示深深的感谢。

由于作者水平有限,书中定有错误和不当之处,敬请广大读者给予批评指正。

作者

1992 年 4 月

目 录

前 言

第一章 C 语言概述	1
第一节 C 语言的发展历史	1
第二节 C 语言的特点	2
第三节 C 语言程序结构	4
第四节 C 语言的基本元素	9
第五节 数据类型	15
第六节 运算符及其优先级	17
第二章 结构化程序设计基础	34
第一节 结构化程序设计思想	34
第二节 if 条件分支语句	35
第三节 控制结构中的跳转语句	39
第四节 switch 开关分支语句	40
第五节 while 循环语句	42
第六节 do...while 循环语句	43
第七节 for 循环语句	45
第八节 goto 语句与标号	46
第九节 for 循环与 while 循环的转换	48
第三章 用于标准设备的 I/O 函数	50
第一节 概述	50
第二节 格式化输出函数——printf()	51
第三节 格式化输入函数——scanf()	54
第四节 无格式字符 I/O 函数	57

第五节	无格式字符串 I/O 函数	58
第四章	数组	60
第一节	数组的概念	60
第二节	数组的赋值与引用	62
第三节	字符数组	65
第四节	字符串数组	68
第五章	指针	73
第一节	指针的概念	73
第二节	指针的说明、初始化与空指针	74
第三节	指针运算	77
第四节	指针的动态存储分配与收回	81
第五节	指针与字符串	83
第六节	指针与数组	97
第七节	指针数组	99
第八节	多级指针	100
第九节	使用指针时易犯的错误	102
第六章	函数	106
第一节	程序和函数	106
第二节	函数的类型、参数与返回值	109
第三节	函数与数组	115
第四节	函数与字符串处理	119
第五节	主函数的参数传递	122
第六节	指针型函数	128
第七节	指向函数的指针	134
第八节	可变数目参数函数的设计	141
第九节	递归函数	143
第七章	存储类与分割编译	147
第一节	局部变量与全局变量	147

第二节	auto 存储类型	150
第三节	register 存储类型	150
第四节	static 存储类型	151
第五节	extern 存储类型	153
第六节	模块化设计与分割编译	155
第八章	C 语言预处理器	161
第一节	文件包含预处理	161
第二节	定义预处理	163
第三节	条件编译预处理	168
第九章	结构与联合	173
第一节	结构	173
第二节	结构的定义与说明	174
第三节	结构体的引用与初始化	178
第四节	结构数组	182
第五节	结构指针	185
第六节	结构与函数	188
第七节	结构嵌套	193
第八节	位字段	200
第九节	定义新类型	205
第十节	联合	208
第十一节	枚举类型	210
第十章	I/O 函数	213
第一节	I/O 函数简介	213
第二节	文件指针与流	216
第三节	面向标准设备的 I/O 函数	218
第四节	文本文件与二进制文件	224
第五节	文件 I/O	225
第六节	文件 I/O 设计实例	232

第七节	系统 I/O 函数	248
第十一章	C 语言与汇编语言接口	256
第一节	C 语言与汇编语言的接口机制	256
第二节	参数传递与引用	262
第三节	实例分析	270
第四节	混合模式编程	272
第五节	嵌入式汇编	274
第十二章	系统功能调用	281
第一节	BIOS 功能调用	282
第二节	DOS 功能调用	295
第十三章	程序的组织与管理	299
第一节	建立工程文件	299
第二节	库管理实用程序——TLIB	304
第三节	程序管理实例	308
第十四章	C 语言图形	315
第一节	图形与硬件	316
第二节	调色板	317
第三节	文本与图形	320
第四节	Turbo C 图形设计	331
第十五章	C 语言程序设计实例	357
第一节	一个窗口系统的设计	357
第二节	数学函数设计	383
第三节	字符串处理程序包	393
第四节	常见的程序设计错误	405
附录 A	ASCII 字符集	416
附录 B	键盘键位码表	421
附录 C	Turbo C 包含文件列表	424
附录 D	Turbo C 保留字	425

附录 E	C 语言转义序列表	426
附录 F	C 语言格式字符表	427
附录 G	C 语言运算符	428
附录 H	颜色与符号对照表	431
附录 I	使用 Turbo C 集成开发环境	432
附录 J	Turbo C 2.0 函数库分类列表	438
	一、输入输出函数	438
	二、数学函数	443
	三、字符串操作函数	446
	四、内存管理函数	450
	五、类型函数	451
	六、时间与日期函数	452
	七、目录控制函数	453
	八、转换函数	454
	九、进程控制函数	455

第一章 C 语言概述

第一节 C 语言的发展历史

C 语言最早是用来描述 UNIX 操作系统及其上层软件的。C 语言的发展与 UNIX 操作系统一样,经历了产生、完善、广泛应用的过程。

C 语言的前身是 BCPL(Basic Combined Programm Language)语言。BCPL 是由 Martin Richards 在 60 年代末设计的,具有结构化好,有指针处理方式和直接与内存地址打交道等特点。BCPL 目前在欧洲仍在使⽤,而 Ken. Thompson 在 BCPL 的基础上做了进一步的发展,于 1971 年在 PDP-11/20 计算机上实现了“B 语言”,并使用 B 语言编写了 UNIX 操作系统和绝大多数应用程序。但由于“B 语言”的一些缺陷,特别是它的无类型性,使得在实际应用中遇到了相当多的困难。后来,美国电话电报公司贝尔实验室的 Dennis Ritchie 在参加 PDP-11 的 UNIX 操作系统的开发工作中,对“B 语言”做了进一步的扩充与完善,并吸收了 Algol 68 和 Pascal 语言的一些优点,从 1971 年开始,用了一年的时间写了第一个 C 语言编译程序,并于 1972 年投入了使用。

1973 年 Ken. Thompson 和 Dennis Ritchie 把 UNIX 操作系统用 C 语言重新写了一遍,使 UNIX 系统的代码量比以前的版本增加了三分之一,并加进了多道程序设计功能。由于 C 语言具有良好

的可移植性,因而为 UNIX 系统后来的蓬勃发展起到了很好的作用。

虽然最初 C 语言是隶属于 UNIX 操作系统,并且是在 PDP-11 计算机上实现的,但是现在的 C 语言却独立于 UNIX 操作系统。它的适应机种从 8 位机到巨型机,并与 FORTRAN、PASCAL 等语言一样,已经成了微型机到巨型机上所共同使用的语言。

今天 UNIX 操作系统已在全世界范围内取得了巨大的成功。从某种角度上可以说,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。

第二节 C 语言的特点

C 语言的特点很多,但大致可以归结为如下几点。

1. 语言表达能力强

C 语言同时兼有“高级语言”和“低级语言”的特点,是一种功能很强的通用程序设计语言。它有直接处理字符地址的功能,并可以完成通常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,有效到足以取代用汇编语言来编写各种系统软件和应用软件。在 UNIX 操作系统中,除了核心程序(Kernel)中的 10%以下的程序由于效率和机器硬件表达的需要是用汇编语言编写的以外,其余程序全部都是用 C 语言编写的。

2. 结构化好

C 语言中提供了一整套称为控制结构的语句(如 if, if...else, while, for, do...while, 以及 switch 等),还具有以花括号“{}”为分隔的代码块化(或叫复合语句)的程序结构,和以函数调用为主的程序设计风格。保证了 C 语言所编写的程序具有良好的结构,易于阅读理解、开发与维护。此外, C 语言中有些存储类型(如,静态 static 类型、外部 extern 类型)在功能上有助于数据隐藏的模块化结构

程序设计。正是由于 C 语言的结构化好的这一特点,使得它能够运用于复杂的程序设计任务。在 C 语言的程序设计中,一个复杂的程序可分割为许多子程序(或叫函数),每个子程序可单独编写和测试,并且许多子程序不一定需要由同一个程序员来编写。

3. 具有较强的数据类型构造能力

C 语言除了有基本的数据类型(如字符型、整型、和浮点型等)以外,还具有构造类型(如数组、结构、指针、联合、枚举等)。在这些数据类型的基础上,还可以通过使用 `typedef` 语句定义新的数据类型。C 语言中丰富的数据类型及其构造能力是其它语言很难与之相比的。

4. 语言精练

在 C 语言中只有 32 个标准的关键字(又叫保留字)和 45 个标准的操作符(也叫运算符)组成了它的程序语句。除了在表示方法上尽可能简洁外,C 语言中的许多成分都是通过显式函数调用来完成的,因而所写出的 C 语言程序非常简洁而功能很强。

5. 可移植性好

在 C 语言的语句中,没有依赖于硬件的输入/输出的语句。需要和诸如磁盘或视频显示这样的外部设备进行交互的任务,是由系统提供的标准库例行程序来完成的。而这些库例行程序本身并不是 C 语言的组成部分。因此,C 语言本身不依赖于硬件系统,从而使得 C 程序很容易从一种计算机环境上不加修改或少加修改就可以移植到与该计算机硬件环境不同的其它计算机系统上。这也是为什么 C 语言能广泛应用于计算机领域的主要原因之一。

6. 生成的代码质量高

针对同一个问题,用 C 语言来描述,通常其代码效率只比用汇编语言来描述低 10%~20%。而由于用 C 语言来描述问题要比用汇编语言描述问题的编程迅速,可读性好,而其效率又降低不大,特别是 C 语言还比较容易移植,因此,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,C 语言的确可以和汇编语言相媲美。

第三节 C 语言程序结构

本节将通过下面这段实例说明一下 C 语言程序的结构、特点及其设计风格。

例 1.1 下列程序完成从算式

$$1+2+3+\cdots+N$$

中找出一个其和大于 5000 的最小整数 N。

```
1  /*      example 1.1      ( get-n. c )
2      This program is to look for a minimum
3      value of N, which makes the sum is great
4      then 5000.      */
5
6  #include <stdio.h>
7  #define SUM_VALUE 5000
8
9  /* This function calculates the sum of
10  1 + 2 + 3 + ... + Limit.      */
11  sum()
12  {
13      int total = 0, Limit = 0;
14      while ( total <= SUM_VALUE )
15          {
16              Limit++;
17              total = total + Limit;
18          }
19      return(Limit);
20  }
```

```

21
22 /* This is the main program. */
23 main()
24 {
25     printf("The least value of N is %d.\n",sum());
26 }

```

将上述程序按附录I所述步聚输入计算机中，并起名为 get-n.c，存入磁盘，编译运行结束后，按下 ALT-F5，屏幕上显示出下面信息：

The least value of N is 100 .

上述程序虽并不复杂，但已足以说明 C 语言程序的结构及其特点。现将其总结归纳如下：

●函数与主函数

C 语言中函数的概念类似于其它语言中的子程序的概念，有所不同的是 C 语言中的主程序也被称为函数，并且被定义成名字为 main() 的主函数。因此，一个 C 语言程序是由一个或多个函数组成，其中必须有一个主函数 main()。一个可执行的 C 语言程序总是从 main() 函数开始执行。

C 语言的函数有两类：一类是用户定义的函数；一类是系统提供的标准函数，在例 1.1 中所用到的系统标准函数是 printf()，它是系统提供的面向标准输出设备的格式输出函数。而 sum() 和 main() 是两个用户定义的函数，其中 main() 是该程序的主函数。

●注释

在 C 语言程序中，以“/*”开头到“*/”结尾中的所有内容表示的意义是一个注释。注释用于帮助阅读和理解程序，在编译时，注释行被忽略掉，并不产生代码行。注释可以写在程序的任何位置上，“/*”与“*/”可以不在同一行上。如果注释写在程序的开头，一般用来说明整个程序的功能或注意事项等。也可以嵌在程序语句行中或某行语句的尾部，一般用来说明一段程序的功能或某一

行程序的作用。当然,注释越多,程序的可读性和可维护性也就越好。

●花括号“{”和“}”的作用

花括号在 C 语言程序中是用来构成函数体和复合语句(也叫程序块,即逻辑上相关的一组语句的集合)的分隔符。一个 C 语言程序中至少要有一对花括号,以表示程序体的开始和结束。

●程序语句

在 C 语言程序中,一条完整的语句必须以分号“;”结束。C 语言程序语句可分为下面四类:

1. 说明语句

用来说明变量的类型和初值。如,例 1.1 中的语句:

```
int total = 0, Limit = 0;
```

它说明 total 和 Limit 都是整型变量,并且初值都为 0。

2. 表达式语句

用来描述逻辑运算、算术运算或产生某种特定动作的语句。

如,程序中的语句:

```
total = total + Limit;
```

```
printf("The least value of N is %d . \n", sum());
```

```
Limit++;
```

等,都是表达式语句。

3. 程序控制语句

用来描述语句的执行条件与执行顺序的语句。如,程序中的语句:

```
while ( total <= SUM_VALUE)
```

```
{循环体语句}
```

就是控制语句。

4. 复合语句

由花括号“{”和“}”括起来的逻辑上相关的一组语句。如,程序

中的语句：

```
{  
    Limit++;  
    total = total + Limit;  
}
```

即为一个复合语句。

●大小写字母敏感性

C语言惯用小写字母。如，C语言中的保留字及系统提供的标准库中所有函数的名称均使用小写字母。在C语言中，变量A和a是完全两个不同的变量，因为它们在内存中所分配的不是同一个地址。因此，字母大小写的敏感性应在C语言程序设计中引起足够的重视。

另外，在C语言中，以下划线“-”字符开头的标识符一般由内部使用，所以，在C语言程序设计时，最好不要用下划线字符作标识符的开头字符。

●程序的书写格式

在C语言程序书写中，并不规定一行只能写一个语句。如，例1.1中的sum()函数完全可以写成下面形式：

```
/*      This function calculates the sum of  
1 + 2 + 3 + ... + Limit. */  
sum() { int total = 0,  
    Limit = 0;    while ( total <= SUM_VALUE ) { Limit++;  
    total = total + Limit; } return(Limit); }
```

这种写法虽然允许，并且也是正确的，但却不如前面所写的那样逻辑上清晰易懂。因此，良好的C语言程序书写风格，会给程序带来可读性和可操作性。建议使用锯齿形的程序书写风格，因为这种书写风格具有层次性和逻辑性。

●预处理特性

在C语言中除了上面所述的四类语句外，还有一类语句，这类语句的作用不是实现程序的功能，而是给C编译系统发布信息