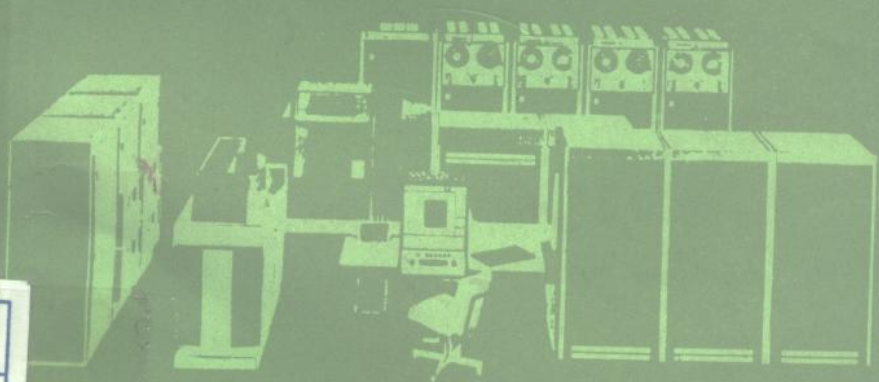


计算机基础知识丛书

C语言入门

孙玉方 仲萃豪 编著



科学出版社

计算机基础知识丛书

C 语言入门

孙玉方 仲萃豪 编著

科学出版社

1988

内 容 简 介

C语言是用于各种计算机程序设计的一个很好的工具。目前，它已成为微型、小型、超小型、中大型和巨型机的通用语言。

本书以丰富的实例，由浅入深地介绍了C语言的基本功能和进行程序设计的基本技巧。全书内容充实，较为实用。

本书可供大专院校计算机专业的学生、计算机用户、管理人员和其他工程技术人员阅读参考；也可作为C语言培训班教材。

计算机基础知识丛书

C语言入门

孙玉方 仲奉豪 编著

责任编辑 曾美玉

科学出版社出版

北京朝阳门内大街137号

中国科学院印刷厂印刷

新华书店北京发行所发行 各地新华书店经售

1988年11月第一版 开本：787×1092 1/32

1988年11月第一次印刷 印张：9 7/8

印数：0001—3,800 字数：221,000

ISBN 7-03-000543-0/TP·33

定价：5.60元

前 言

为迎接新的技术革命的到来，一个推广应用计算机的高潮正在全国兴起。广大科技人员、干部、工人和学生都迫切要求学习计算机基础知识。为了适应这一令人鼓舞的形势发展需要，中国计算机学会所属普及委员会决定编辑一套《计算机基础知识丛书》，将于近年内陆续出版。本书就是这套丛书之一。目前，已经出版的有：《计算机应用系统》、《微型计算机及其应用》、《数据库及其应用》和《UNIX 系统入门》。

计算机是信息处理的工具。现代计算机的迅猛发展加速了社会信息化的进程。信息的巨大价值日益被人们所认识，信息的快速生成、广泛传播和有效利用使人类更能高度发挥智力劳动的效益和施展社会组织的才能，从而极大地促使科学技术的进步和推动社会生产力的发展。普及计算机知识，特别是计算机应用方面的知识，正是整个信息生成、传输、利用过程中不可缺少的重要环节。我们就是基于这样的认识来组织编写《计算机基础知识丛书》的。它应该能为各行各业科技人员提供在自己的专业领域中应用计算机急需的基础知识，同时也为计算机专业科技人员更新和扩展自己的知识领域创造条件。

计算机科学技术是一门综合性很强的学科，发展又十分迅速。我们要求丛书能尽力做到先进性和实用性相结合，科学性和通俗性相结合，硬件和软件相结合。为了保持题材新颖、内容先进，我们采取分批确定选题、随定随写、尽快出版的方针。为了对广大科技人员应用计算机提供实际指导，我们

希望能结合一些应用计算机的实例来讲解原理，不但要介绍计算机有什么用，而且还要介绍它怎样用。我们要求丛书的大多数选题都具有中级科普读物的性质，便于自学，也能提供给相应程度的培训班用作参考教材。作为普及读物，我们自然要更加注意叙述生动形象、文笔通畅易懂，使广大读者通过阅读丛书获得收益。

由于我们经验不足，水平有限，疏漏不妥之处在所难免，尚希望读者批评指出，并提出宝贵意见。

《计算机基础知识丛书》编辑委员会

引 言

C语言对于各种计算机的程序设计是一个很好的工具，它的许多特点正被人们逐渐认识，因而也就越来越普遍地应用在许多程序设计领域中。目前，它已成为微型、小型、超级小型、中大型，甚至巨型机上通用的语言。

正因为它的应用越来越广泛，所以在它已流行了十几年后，近来国际有关组织正在对它进行国际标准化工作。

现在，C语言不但在商业领域中得到了计算机公司、用户的推崇，在学术界也倍受重视。1983年，UNIX系统和C语言的主要开发者 K. 汤普逊 (Ken Thompson) 和 D. 里奇 (Dennis Ritchie) 获得了 ACM 图灵奖。评审委员会对 UNIX 和 C 作了极高评价，并且特别指出，UNIX 系统之所以有今天，主要是因为有了 C 语言。

在我国，随着微型、小型机的大量普及，计算机的应用、系统软件和应用软件的开发工作日益重要，而 C 语言是开发软件极为有效的工具。从普通管理人员到专业技术人员，从一般用户到大学本科学生，都表现了对 C 语言的极大兴趣。许多人想了解它，掌握它。

本书是一本入门性质的书。既然是入门性质的书，就不可能对 C 语言作面面俱到的介绍，也不可能去陈述那些比较高深或比较冷僻的内容。因为这样既不利于读者很快建立起整体概念，也不利于读者集中精力于语言中最基本而且是最重要的部分。为此，本书主要侧重在初学者必须掌握，也是作者认为是比较重要的那些部分。至于那些省略的及还需要

有一定了解的内容在小节号后用星号*标出。对于那些需要进一步讨论的专题，用单独一章(第九章)作一简要介绍。

与一般语言不同，C语言与运行它的环境(主要是操作系统)有着非常密切的关系。由于C语言最初是在UNIX操作系统环境下开发出来的，而且主要用在UNIX操作系统环境下，因此，它与UNIX有着千丝万缕的联系。但是本书主要讲C语言的基本功能，并希望它对其它环境下C语言的使用有所帮助，所以书中有关其环境部分尽量略去。需要了解时，请参阅有关的使用手册。

本书编写的方法是从实例着手，由浅入深地介绍C语言的基本功能和进行程序设计的基本技巧。本书采用了如下结构安排：

第一章对C语言的特点和演变过程作了介绍。为了使初学者对C语言程序设计有一个大概了解，并为以后的实习作准备，还介绍了C语言程序的编辑、编译和运行的全过程。

第二章对C语言中使用的基本常量和变量作了介绍，特别是变量的基本类型、类型的自动转换及强制转换。

第三章介绍了C语言中的基本运算符和相应的表达式，最后列出了所有运算符、它们的优先级和结合性。

第四章介绍了构成C语言的基本组成单位——函数，讲述了函数的定义、调用、参数等概念，也讨论了变量作用域和存储类。

第五章给出了C语言的主要语句和控制流结构，介绍了各种语句的使用方法及有关注意事项。

第六章介绍了数组和指针，这两者之间有着异常紧密的关系。这里对数组与指针、地址及指针与函数、命令行参数等关系都作了一些探讨。最后以栈上的一系列操作为例总结了数组与指针的主要用法。

第七章介绍了C语言中另一类构造型的数据类型——`struct`。介绍了它的说明形式,它与数组、指针及函数的关系。

第八章介绍了C语言程序中常用的输入/输出和其它库函数。这部分是与C语言程序运行的环境密切相关的。

作为一种补充和扩展,第九章讨论了一些要进一步展开的专题,包括变量存储类、预处理功能、若干运算符和数据类型等。

第十章以一个例子为线索,介绍了用C语言进行程序设计的步骤和方法,供读者参考。

从内容上讲,全书分成三个层次:前五章是C语言中最基本的成分;而前八章已经包含了比较完整的C语言的内容;第九章和第十章的目的是为了补充和巩固已学的知识,这两章以及各章中用*号标出的小节,若从书中删去并不影响内容的完整性,读者可以根据需要决定取舍。

学习一种语言的基本方法就是多用这种语言去工作。对于C语言来说,亦是如此,但愿读者在实践中不断进步。书中的完整例子全部经过上机验证。

作者虽然对于UNIX系统及C语言有若干年研究、开发和教学方面的经验,但是如何编写一本科学普及读物却还是第一次。所以无论从材料的取舍、叙述的方法等方面都可能存在不足之处,恳切地希望读者提出宝贵意见,以利今后的提高。

作者

目 录

前言	
引言	vii
第一章 C语言概观	1
1.1 C语言的由来	1
1.2 C语言程序若干例	4
1.3 C语言程序的编译和运行	11
1.4 C语言的基本词法	20
第二章 常量和变量的基本类型	23
2.1 常量	23
2.2 数据类型分类	29
2.3 普通整型变量	30
2.4* 其它整型变量	33
2.5 浮点变量	35
2.6 变量初始化	36
2.7 基本数据类型的数值大小	37
2.8 混合类型的运算分量及类型自动转换	37
2.9 类型的强制转换	41
第三章 运算符和表达式	42
3.1 算术运算符	42
3.2 按位运算符	43
3.3 赋值运算符	45
3.4 关系运算符	46
3.5 逻辑运算符	48
3.6 左值和增1减1运算符	52
3.7 优先级和结合性	55

第四章	函数及变量存储类	57
4.1	函数及其定义	58
4.2	形参与函数调用	64
4.3	变量作用域及其存储类	69
4.4	C 语言程序一般结构	85
第五章	语句与控制流	89
5.1	复合语句和表达式语句	90
5.2	if 语句	91
5.3	while 与 do-while 语句	98
5.4	for 语句	102
5.5	跳转语句	113
5.6	switch 语句	118
第六章	数组与指针	124
6.1	指针及其简单运算	125
6.2	指针与函数的参数传递	129
6.3	数组与地址运算	139
6.4*	指针表达式及指针运算优先级	147
6.5	多维数组及数组的初始化	152
6.6	指针数组与命令行参数	159
6.7	栈的操作	168
第七章	结构	176
7.1	结构说明	176
7.2	结构数组	182
7.3	结构与指针及函数	187
7.4*	引用自身的结构	195
第八章	输入/输出、文件访问及系统调用	199
8.1	终端输入/输出	201
8.2	文件输入/输出	216
8.3*	系统调用	229
第九章	一些要进一步讨论的专题	238

9.1	类型定义 typedef	239
9.2	变量存储类的进一步讨论	242
9.3	递归函数与函数的递归调用	244
9.4	带参数的宏及条件编译	251
9.5	联合、枚举及其它	261
9.6	函数与指针	268
第十章	C 语言程序开发实例	276
10.1	问题的提出	276
10.2	与之有关的问题——动态存储分配	277
10.3	程序开发过程	278
参考文献	304

• ▼ •

第一章 C 语言概观

作为人与计算机交流工具的程序设计语言,经过人们不断地开发,至今已逾一千种,然而真正实用而且被人们广泛接受的也就是数十种。其中,ALGOL60、FORTRAN 主要用于科学和工程计算,所以常称为算法语言;COBOL 主要用于商业数据处理,所以也常称为数据处理语言;BASIC 主要用于人机对话,所以也称为交互式语言;PASCAL 侧重于描述编译程序等,所以也称为系统程序设计语言。

C,作为一种程序设计语言,它与这些语言有些什么关系,它是怎么来的?有什么基本组成部分?本章主要回答上述这些问题。

1.1 C 语言的由来

C语言是1972年由美国电话电报公司(AT&T)属下的贝尔实验室(Bell Labs.)开发的。语言初版的设计和实现都是由该实验室的D.里奇一个人独立完成的。当时开发C语言的主要原因为了更好地描述UNIX系统。不过目前C语言已经不局限于一种机器和一种操作系统,而在众多的微、小、超小、中、大型机和CP/M、MS-DOS、VMS等若干种操作系统上运行了。

C语言具有简洁的表达方式、强有力的控制流程、灵活的数据构造能力、丰富的运算符、可与汇编程序媲美的运行效率及良好的可移植性等众多优点。所以它的出现和崛起与

PASCAL 语言形成了一种对抗的态势。

PASCAL 语言作为结构化程序设计语言的代表,尤其适用于教学和描述算法,然而,它的实用性比 C 语言差。已经有许多 PASCAL 语言的爱好者在不知不觉中开始以 C 语言作为他们进行软件开发的工具了。其实追溯一下发展史可以发现,C 和 PASCAL 语言有同一个祖先,同属于 ALGOL 语言族系(图 1.1)。

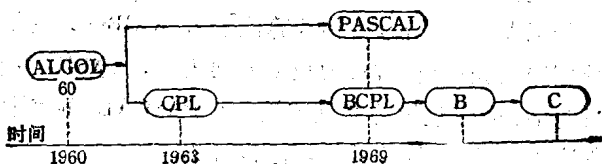


图 1.1 C 语言演变历史

UNIX 系统的早期版本曾经是用汇编语言写的。为了提高可读性和可移植性,K. 汤普逊 (UNIX 系统的主要开发者之一)于 1970 年在 BCPL 基础上实现了 B 语言,并用 B 语言写了 UNIX 系统的绝大多数实用程序。和 BCPL 一样,B 过于简单而且不适用于 PDP 11 机。为此,D. 里奇在 B 上加进了构造类型等必要的功能并作了归整,形成了新的语言——C。至于为什么取名为 C,众说不一,这里也就不加深究了。

一种计算机语言不可能包罗万象,也不可能对所有场合都适用。在商业应用、科学计算、系统程序开发和应用程序设计四个主要领域中,C 语言是后两个领域中不可多得的有力工具。因此,可以说 C 语言是一种更侧重于软件开发特别是系统程序设计的语言。为什么呢?这是因为,一方面它具有低级语言如汇编语言的性质,能直接反映机器硬件特性,编译后代码效率较高,而这是一般高级语言很难达到的。另一方

而它又具有高级语言所具有的所有重要特性，如丰富的控制流程和强有力的数据构造能力。这样，它编程效率很高，调试和维护容易。故有人称它是一种“中级”语言，这不无道理，从图 1.2 中我们可以看到 C 所处的层次。

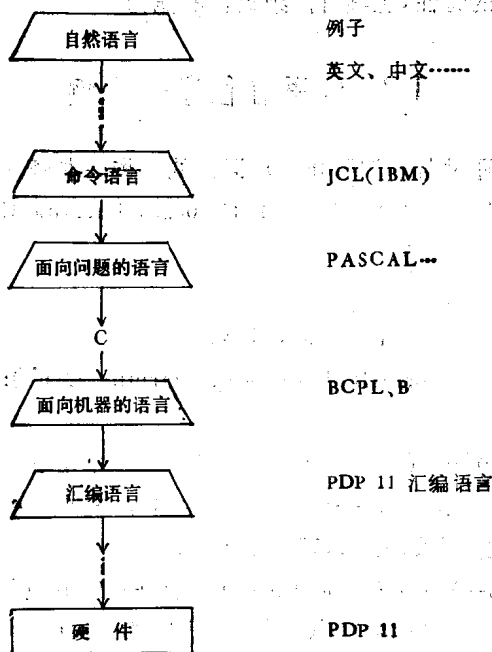


图 1.2 语言层次

由图 1.2 可见，愈是上层，语言与机器硬件距离愈远，和直接与硬件打交道的汇编语言也有很大的距离。搞程序设计的人都会有这样的切身体会：语言越“高级”，描述问题时越简单，对机器内部细节了解就越少；而使用的语言越“低级”，描述问题时就越复杂。

这种以问题为出发点的语言，因与机器硬件相距太远，所以不适合描述与机器关系极大的系统软件，特别是操作系统。

后来人们把高级语言的层次降低一些，开发了面向机器的语言，BCPL 和 B 就是这类语言。然而这种语言由于与机器关系过于密切，所以除了用于特定硬件的设计环境外，并没有别的用处。C 语言恰好介于这两级语言中间，所以它不仅具有低级语言的特性，也具有高级语言的能力。

1.2 C 语言程序若干例

本节通过若干例子介绍 C 语言的一些基本概念及成分。

例 1.1 打印一串字符：Let's begin learning C。其程序如下：

```
main( )
{
    /*Print a string*/
    printf("Let's begin learning C\n");
}
```

用高级语言(比如 C)写成的程序称为源程序。通常，一个 C 语言源程序放在一个或多个文件中，这种文件的名字应以 .c 结尾。比如例 1.1，可把整个程序放在文件 exam1.c 中。

C 程序以函数(详见第四章)作为基本构成单位，函数总是以名字后随圆括号标志，如例 1.1 中的 main ()。C 程序的每个文件中可包含若干个函数。在构成一个 C 程序的众多函数中，有且只能有一个函数的名字为 main。程序编写时不管其放在什么位置上，程序总是从它开始执行，并且正常情况下还以它为结束。其余的函数由它及受它调用的函数调用。

花括号{和}是一种语句括号，这不过是一般语言中 begin 和 end 的省写。花括号括起的成分在 C 语言中称为复合语句或分程序。函数最外层花括号括起的分程序也常称为函数

体。在例 1.1 中函数 `main ()` 只有一对花括号，它括起的就是函数体，而 `main()` 称为函数头。

以 `/*` 开头并以 `*/` 结尾的字符串是注释。注释只是为了说明问题，在编译时并不产生任何代码。注释可出现在任意合适的地方。注释对程序本身的功能没有丝毫影响。

```
printf ("Let's begin learning C\n");
```

是一个格式输出语句，其中，`printf ()` 是一个函数。要注意，C 中每一个语句后都有分号“`;`”，这是 C 语句的终止符，属于语句的一部分，而不是一般语言中的分隔符，所以在 C 程序中分号是不可省略的。

严格说来，C 中没有提供任何输入/输出手段，C 语言的输入/输出都要借助于宿主系统（即其环境，比如 UNIX、VMS、MS-DOS 等操作系统）来提供。这种输入/输出函数属于标准库函数，并不是 C 的法定成分，但总是由 C 程序来调用。为了后续章节的使用方便，下面，简单介绍几个常用的输入/输出函数（详见第八章）及其它有关问题。

1.2.1 常用的输入/输出函数

在 C 语言文本中并没有输入/输出语句，输入/输出处理是通过调用某些函数来完成的。由于实现的不同，输入/输出功能也可因系统而异。但是平常的一些输入/输出函数是所有系统通用的，比如下面介绍的 `printf ()`、`getchar ()` 和 `putchar ()`。

1. printf ()

`printf ()` 的主要功能是在终端屏幕上输出，在输出过程中进行格式化处理。它的一般形式是：

```
printf("format_info", arg1, arg2, ...);
```


这里 `arg1, arg2, ...` 可以不出现, 如例 1.1 所示。

`format_info` 是格式信息, 通常是一串以 `%` 标志的输出格式要求, 如表 1.1 所示。

表 1.1 常用的输出格式

表 示	功 能
<code>% d</code>	十进制整数
<code>% o</code>	八进制整数
<code>% x</code>	十六进制整数
<code>% f</code>	浮点数
<code>% c</code>	字 符
<code>% s</code>	字符串

在 `%` 与格式指示符 (`d, f` 等等) 之间可以有数字。比如, `%4 d` 表示输出的十进制数至少占 4 位, `% 6f` 表示输出的浮点数至少占 6 位, `% .2f` 表示小数点后数字占 2 位, 而 `%6.2f` 则表示输出的数字共占 6 位, 小数点后占 2 位。表 1.1 中所列的输出格式可以混合作为一个 `format_info` 看待, 从而按各种不同格式输出不同的量。下面是一些例子。

如果 `x` 的当前值是 13, 则

```
printf("%d is the current value of x\n", x);
```

将告诉 `printf()` 第一个实际参数 (简称实参) `arg1` (即 `x`) 的值以十进制整数 (因为是 `% d`) 输出。所以执行该语句的结果是:

```
13 is the current value of x
```

可以看到在 " " 之间的其它字符串按原样输出。这里的 `\n` 表示换行。

又如, `pi` 的当前值是 3.14159, 则

```
printf("% .2f is the current value of pi\n", pi);
```

将以浮点并且小数点后取两位 (因为是 `% .2f`) 的格式输出 `pi` 的