

史美林 苏云清 编

C语言 程序设计基础

C语言
程序设计
基础

史美林
苏云清
编

清

8722I

162

社

清华大学出版社

C 语言程序设计基础

史美林 苏云清 编

清华大学出版社

C语言是一种通用的程序设计语言。本书根据作者在清华大学多次教学的讲义改编而成，内容包括C语言的数据类型、各种语句及其程序设计方法。对程序运行及UNIX操作系统也作了简要介绍。全书重点突出，例题丰富，便于自学。

本书可供大专院校师生、科技人员阅读，也可用作计算机人员的培训教材。

JS126/30
0

C语言程序设计基础

史美林 苏云清 编

☆

清华大学出版社出版

北京 清华园

北京市孙史山印刷厂印刷

新华书店北京发行所发行 各地新华书店经售

☆

开本：787×1092 1/16 印张：10 $\frac{5}{8}$ 字数：268千字

1986年12月第1版 1986年12月第1次印刷

印数：00001—10000

书号：15235·243 定价：2.25元

871058

前 言

C语言是一种通用的程序设计语言。它作为UNIX系统的主力语言，随着UNIX的发展而日趋普及。

C语言是一种面向结构的程序设计语言。它具有丰富的数据类型、灵活方便的多种运算符、新颖的控制流和数据结构，以及经济的表达式，能够有效地用来描述操作系统、编译程序以及编制各种各样的软件。由于它的通用性及可移植性，因此可以在不同机器上移植C语言程序。

随着国内计算机应用的广泛发展，许多计算机系统（特别是使用最广泛的微型机系统），都已经配备或希望配备UNIX操作系统。为了尽快地学习掌握UNIX操作系统，更好地开发系统的资源，充分发挥计算机的效益，人们迫切希望有一本比较适合我国国情、符合广大读者口味的介绍C语言的书。

为了满足读者的这种良好的愿望，根据UNIX操作系统和C语言的基本理论，结合多次操作系统及C语言教学的实践，编者在以前几次教学讲义的基础上修改整理成这样一本书，献给广大读者。

在编写本书的过程中，考虑到许多读者已经学习和实践过一些高级语言或汇编语言，没有必要去重复叙述程序设计的最基础的知识，因而把重点放到C语言的一些特点的描述方面。在“语法”和“算法”的关系上，侧重点在语法上。

为了有助于读者自学C语言，本书提供了许多完整的程序，并将运行情况 and 输入、输出情况作为注释附在程序的后面。程序的文件名按章进行统一编号，便于阅读和查找。书中还提供了一些思考题，以有助于读者弄清一些易于模糊的概念。

由于编者学习C语言的时间不长，水平有限，加上时间仓促，错误和不当之处肯定不少，敬请广大读者批评指正。

编 者

一九八四年十二月

于清华大学

目 录

第一章	基础知识	(1)
§1	计算机系统.....	(1)
§2	程序设计基础.....	(2)
§3	UNIX系统概述.....	(2)
§4	C语言简介.....	(4)
§5	C语言的程序结构.....	(5)
§6	C语言的基本符号.....	(6)
第二章	基本类型及运算	(8)
§1	基本类型.....	(8)
§2	基本运算.....	(9)
§3	变量说明.....	(13)
§4	常量定义.....	(14)
§5	数值的转换.....	(15)
第三章	简单的程序设计	(16)
§1	语句综述.....	(16)
§2	说明语句.....	(16)
§3	赋值语句.....	(17)
§4	输入语句.....	(18)
§5	输出语句.....	(19)
§6	应用举例.....	(22)
第四章	流程的控制	(27)
§1	条件语句.....	(27)
§2	复合语句.....	(28)
§3	while语句.....	(29)
§4	do 语句.....	(30)
§5	for语句.....	(31)
§6	开关语句.....	(32)
§7	转移语句.....	(34)
§8	应用举例.....	(35)
第五章	函数	(45)
§1	库函数.....	(45)
§2	函数定义和函数调用.....	(48)
§3	自动变量和外部变量.....	(51)
§4	静态变量和寄存器变量.....	(54)
§5	函数的递归.....	(58)

§6	分程序结构.....	(60)
§7	应用举例.....	(62)
第六章	构造型数据类型.....	(74)
§1	数组类型.....	(74)
§2	结构类型.....	(75)
§3	指针类型.....	(78)
§4	联合类型.....	(80)
§5	变量的初始化.....	(81)
§6	应用举例.....	(83)
第七章	文件.....	(99)
§1	文件的概念.....	(99)
§2	文件的处理.....	(102)
§3	UNIX 系统编程.....	(104)
§4	程序参数.....	(107)
§5	应用举例.....	(109)
第八章	程序的运行.....	(118)
§1	UNIX 命令简介.....	(118)
§2	文本的编辑.....	(120)
§3	编译和执行.....	(123)
§4	编译和执行错误.....	(125)
第九章	进一步讨论的问题.....	(127)
§1	C 预处理程序.....	(127)
§2	条件编译.....	(130)
§3	类型定义.....	(132)
§4	重新定向和管道.....	(135)
§5	shell 命令.....	(139)
§6	shell 语言.....	(142)
附录	(155)
附录 A	关键字.....	(155)
附录 B	ASCII 字符集.....	(155)
附录 C	运算符优先级和结合性.....	(156)
附录 D	C 语言主要功能表.....	(157)
附录 E	C 和 PASCAL 的比较表.....	(158)
附录 F	错误信息表.....	(160)
参考资料	(163)

第一章 基础知识

本章为全书提供一些基础知识。前两节简要地概括计算机系统和程序设计的有关概念。第三节初步介绍一下与C语言密切相关的UNIX操作系统。后三节是关于C语言的基础知识，分别叙述其发展情况、程序结构和基本符号。

§1. 计算机系统

计算工具是随着社会的需要和技术的进步由低级向高级逐步发展的。从我国唐宋时期的算盘到法国帕斯卡机械计算机(1642年),从计算尺(1654年)到手摇计算机(1887年),从电动计算机到电子计算机,其间经历1000多年,才达到今天的计算机水平。而明天呢,将是计算机世界。

自从世界上第一台电子计算机“ENIAC”在美国诞生之后,近四十年来,电子计算机得到了迅猛的发展和越来越广泛的应用。它已经经历了电子管、晶体管、集成电路和大规模集成电路四代,而第五代计算机的诞生已经为期不远了。由于它具有运算速度快,存储容量大,自动控制可靠等优点,因此,它在科学计算、信息管理及工业控制等方面发挥着巨大的威力,从空间技术的研究到家庭生活的安排都有计算机的用武之地。

今天,人们所讲的计算机是一个计算机系统,而不是一台“裸机”。一个完整的计算机系统由计算机硬件系统和计算机软件系统(俗称程序系统)组成,前者一般为组成计算机的物质设备,后者为管理和使用计算机的各种程序。有时人们将计算机的有关资料也归到软件中。

计算机硬件系统主要由处理机和外围设备组成。处理机包括存储器、运算器和控制器。其中存储器(简称“内存”或“主存”)主要有磁芯存储器和半导体存储器两种,它决定随机存取的容量以及运算速度和精度。外围设备包括外存设备、输入设备和输出设备。外存设备(又称“辅存”)通常有磁盘、磁带或磁鼓等,磁盘又有软盘和硬盘之分,用来记录大量暂时不用的信息,它们的容量可以是内存容量的成千上万倍。一个较好的计算机系统,都必须有外存设备,否则用途是非常有限的。

计算机软件系统的作用是管理计算机的所有软件及硬件资源,充分发挥它们的功能。它通常包括系统软件和应用软件两大类。系统软件主要有操作系统、各种语言的处理程序、机器维护程序,服务性程序和装配程序等,其中操作系统是最重要的系统软件。应用软件,又叫专用软件,是指为实现某种专门的目的而配制的软件,包括专用语言和应用程序等。

一个计算机的用户,首先关心的必定是使用什么机型,在什么操作系统下工作。同样一台计算机,可以使用不同的操作系统,比如,PDP-11/23计算机,可以使用RT-11操作系统、RSX-11M操作系统以及UNIX操作系统。本书中所涉及的程序主要是在PDP-11/23计算机上,在VENIX(UNIX的变种)操作系统控制下运行的,即使同一台计算机,由于使用的操作系统不同,功能的差异是很明显的。由此可知,计算机的软件系统是何等重要。

目前国内使用的最基本的计算机系统有:主机,终端设备(键盘和显示器),打印机,

软磁盘，操作系统以及几种语言处理程序。对于使用UNIX操作系统的用户来说，硬磁盘也是最基本的系统部件之一。

§ 2 程序设计基础

伴随着电子计算机的出现和发展，程序设计的产生和发展经历了由简单到复杂、从低级到高级的发展过程，具体表现为计算机语言的发展。

计算机诞生时，使用的是“机器语言”，它用二进制代码表示计算机的指令系统，编写程序。这种语言要求程序员非常熟悉计算机的组成与结构，一般用户是难以胜任的。而且编写程序非常繁琐，阅读程序又特别困难。后来人们用一些符号来代替每一个具体的指令，形成了“符号语言”；进而将存储地址等也用符号表示，这就是大家称呼的“汇编语言”。当然，计算机本身并不认识汇编语言，还要求有一个翻译过程，称之为“汇编程序”，简称“汇编”，将汇编语言翻译为机器语言，从而让计算机执行。汇编程序属于系统软件，由系统提供，用户使用。无论是机器语言，还是汇编语言，都与具体的计算机有关，即与其指令系统密切相关，因而称为面向机器的语言。

由于面向机器的语言的局限性，人们希望有一种与具体的机器无关的语言，最好接近于数学公式的描述。这就出现了各种算法语言，也称高级语言。高级语言编写程序比较方便，阅读起来也容易。当然，高级语言也必须翻译成机器语言，而且这种翻译更复杂。这种翻译通常称为“编译程序”，简称“编译”。比如FORTRAN语言、PASCAL语言和C语言就由“编译”来翻译，而BASIC语言通常靠“解释程序”边解释边执行。

电子计算机靠程序来自动执行，而程序靠程序员来编写和调试。程序员根据实际任务提出问题，然后确定计算方法，构造数学模型，画出流程图，选择合适的语言，编写程序并上机反复调试，直到满足实际需要为止。这就是程序设计的过程。

高质量的程序设计，要求程序员在编制程序的过程中正确处理时间与空间的关系，即在程序可靠、正确的基础上尽量节省时间和空间（运算速度快，运行时间省，程序精练，占据内存单元少）。当然，在时间与空间有矛盾时，应根据需要和可能解决主要矛盾，提高程序的质量。

§ 3 UNIX 系统概述

UNIX是一个通用的、多用户、交互型的操作系统。它是由Ken. Thompson 和 D. M. Ritchie 等人于1969年在Bell实验室，在MIT和GE联合开发的Multics的基础上发展起来的。

1970年形成了一个基本的操作系统，它是用汇编语言写的，在PDP-7计算机上运行。为了表示它与“Multics”既有联系又有区别，因此取名为“UNIX”其中“UNI”与“MULTI”相对，“X”与“CS”谐音。在作了适当的改进之后，形成了第二版的UNIX，它仍然用汇编语言编写，在PDP-7计算机上运行。

1971年，第三版UNIX在PDP-11计算机上实现。

1972年，改进成第四版，同时也在PDP-11/45计算机上实现。

1973年，D. M. Ritchie用C语言改写了用汇编语言写的UNIX的90%，成为第五版UNIX，并于1974年在UCACM上正式发表，同时开始在大学里传播。

1976年Bell实验室正式发表UNIX的第六版。引入了多道技术，并在PDP-11/34、/40、45和/60等计算机系统上运行。

1978年7月发表UNIX的第七版，并在1979年1月由Western Electronic正式承认。

1979年发表了VAX-11系统上的UNIX-32。

1980年加利福尼亚大学伯克莱分校发表了为VAX-11的带有虚拟存储的版本：4.0BSD和4.1BSD。而Microsoft则把UNIX引入到16位微型机系统上，发展成为XENIX系统。

1981年由AT&T开发了UNIX system III。

1983年由AT&T开发了UNIX system V，而伯克莱分校又发表了4.2BSD。

由此可知，从80年代初期起，UNIX已经从一个在实验室里的试验系统发展成为一个著名的计算机软件系统，而且从学术领域进入了商业市场。目前一些比较有影响的计算机工厂所生产的计算机系统几乎都配备了UNIX系统或它的变种（UNIX-like），在微型计算机上则更为突出。

UNIX系统在设计过程中有三个主要目标：

1. 简洁性。尽量保持操作系统简单。只把最基本的功能放在系统内部，而依靠通常的用户程序来提供完善性。

2. 通用性。运用单独一种方法为多种相关目的服务。比如同一个文件系统可以用来读（或写）文件设备和交互进程消息缓冲。

3. 方便性。它创造一种环境，把现有的小型程序比较自由地组合起来完成一个大任务。以小积大，无论从效率、正确性，还是时间和经济上都有好处。

UNIX系统在设计过程中将结构分成三层：

1. 最里层为UNIX核心，即UNIX操作系统。

2. 最外层为用户态程序，包括各种应用软件、实用程序以及其它系统软件。

3. 中间层是shell命令解释程序，是用户与系统核心的接口。

概括地说，就是核心层、接口层和用户层。一般我们把核心层称为操作系统，而把整个系统称为UNIX系统或UNIX机器。

UNIX操作系统主要有以下五个特点：

1. 文件系统的目录结构是树形的分级结构。它允许用户根据需要自由装卸。

2. 文件是没有内部结构的字符流序列，并把外部设备和目录与普通文件一样看待和管理。

3. UNIX提供了命令程序语言shell。它既可以象键盘命令那样直接使用，又可以象高级语言那样进行编程，功能很强，使用方便。

4. 在UNIX操作系统支持下，它的外层提供了大量的语言和子系统，以满足各种用户的需要。

5. UNIX操作系统支持下的软件系统都是用C语言编写的。甚至UNIX本身的90%也是用C语言编写的，因此易于修改、维护和移植。

UNIX操作系统产生于七十年代，发展最快却是最近几年。当前，国际上对UNIX有两种截然不同的看法。一种是“UNIX——从现在起就把它当作死尸”，另一种是“UNIX——从现在起就把它当作标准”。但是，大多数工业家确信，UNIX不仅是当前迅速涌现的超级微型机的理想操作系统，而且还将向小型机和超级小型机领域大量渗透，从而成为一种广泛采用的操作系统。

国内一些科技工作者和教育工作者已经对UNIX研究了多年。近几年，随着微型计算机的普及，UNIX操作系统也得以逐步推广。尽管UNIX操作系统本身不可能完美无缺，甚至开始使用时还会有些不习惯，但是，我们认为，对于研究和使用计算机的人来说，UNIX系统是一个比较好的系统；而对于学习和研究操作系统的人来说，UNIX则是现阶段的一个比较理想的操作系统。

§4 C语言简介

C语言的出现是与UNIX操作系统紧密联系在一起，C语言本身也有一个发展过程，目前仍然处于发展和完善过程之中。下表给出了C语言的发展过程。

语言名	设计者	年份
CPL	C. Strachey 等	1968
BCPL	M. Richards	1969
B	K. Thompson	1970
C	D. M. Ritchie	1972

C语言起源于1968年发表的CPL语言(Combined programming language语言)，它的许多重要思想来自于 Martin Richards在1969年研制的BCPL语言，以及以BCPL语言为基础的而由Ken Thompson在1970年研制成的B语言，K. Thompson用B语言写了第一个UNIX操作系统，用在PDP-7计算机上，(PDP-7计算机已被淘汰)。D. M. Ritchie 1972年在B的基础上研制出C语言，并用C语言写成了第一个在PDP-11计算机上实现的UNIX操作系统。1977年出现了独立于机器的C语言编译文本《可移植C语言编译程序》，从而大大简化了把C语言编译程序移植到新环境所需做的工作，这本身也就使UNIX操作系统迅速地在各种机器上实现。

C语言是以与PASCAL 及ALGOL等语言相类似的思想为基础而设计出来的面向结构的程序设计语言。其主要特征如下：

(1) 以英文小写字母为基础。在不使用小写字母的微型机中允许用大写字母。

(2) 程序是以与LISP及APL相同的函数集合构成的，函数的参数是按“值调用”方式进行。

(3) 指针可以作为数据来处理。

(4) 运算符非常多，有利于编写程序。

(5) 通过预处理可以进行宏调用。

(6) 程序表示简洁，键盘输入量小。

(7) 没有处理字符串的特别功能，但能区分字符及字符串。

(8) 没有作为语言组成部分的输入输出功能，但能通过函数程序来实现。

(9) 没有作为语言组成部分的文件类型，但能通过UNIX编程来解决。

根据C语言的这些特点，它与UNIX编程以及shell命令相结合，特别适宜于编写系统程序，这是它的最突出的优点。

当然，C语言也有缺点，至今还缺少一个国际上公认的完整的语法定义，语法关系的严格性不如FORTRAN 及PASCAL语言，不太适合于数值运算。

目前，带有UNIX操作系统的计算机几乎都有C语言，同样，C语言也可以用在某些非UNIX操作系统的机器上，当然其功能往往会受到一些影响。

§5 C 语言的程序结构

上一节中谈到C语言的特征时说：“程序是以与LISP及APL语言相同的函数集合构成的”。也就是说，任何C语言程序都是由一个或几个函数所组成。我们首先看一个简单的C语言程序。

程序如simple.c所示

```
/*
 * file :simple.c
 * date:11-nov-83
 */
main ( )
{int a, b, c, sum;
  a=1;  b=2;
  scanf ( "%d", &c );
  sum=a+b+c;
  printf ( "sum=%d\n", sum );
}
/* input :3
 * output:sum=6
 */
```

这个程序的功能是已知两个整数1和2，输入第三个整数，然后求和并输出结果。运行本程序时，输入（input）的整数是3，输出（output）的结果是sum=6。

在C语言中，通过/*...*/作为程序的注释部分，当一行不能满足注解要求时要另起一行，一般要求以*开头。注释可以出现在程序的任何部分，在简单的程序中可以没有注解。为了编写程序和阅读程序方便，本书中的C语言程序都有注释部分，一般在程序前后各有一部分，程序前面的注释表示程序的文件名称（filename）以及建立的日期（date），这样便于从外存储器中找到文件；程序后面的注解一般表示程序的运行情况，主要是程序的运行情况，输入情况，输出情况，这里的输入一般为终端键盘输入，输出通常指终端显示器输出。这样有利于读者学习和理解程序。

本程序的主体部分为

```
main ( )
{...
}
```

其中main是一个函数，而且是一个特殊的函数，所有C语言程序都必须以main函数开头。它实质上是C语言程序的首部。一般函数名后面为参数表，参数表在一对圆括号（）之中。main函数可以有参数（将在UNIX编程中叙述）。本程序中的main函数没有参数，即参数表为空，但一对圆括号（）必须有，不能省去。

{ } 是一对花括号，将构成函数的语句包括起来。C语言中的语句大致分为两类：一类为说明语句，用来描述数据，决定内存的分配；另一类为执行语句，用来描述对数据进行的操作，决定内存的内容。

本程序共有 6 个语句，

`int a, b, c, sum;` 是说明语句。它说明 `a, b, c` 和 `sum` 四个变量都是整数 (`integer`) 类型的变量。

`a=1; b=2;` 是两个赋值语句。等号 `=` 作为赋值运算符，它们分别将 1 和 2 赋给变量 `a` 和 `b`。
`scanf ("%d", &c);` 是一个输入语句。它按十进制 (由 `d` 指明) 给变量 `c` 输入一个整数，在程序执行时等待用户输入。本程序中输入整数 3。

`sum=a+b+c;` 是一个赋值语句。它将三个变量的值求和。

`printf ("sum=%d\n", sum);` 是一个输出语句。它首先输出字符串 `sum=`，然后按十进制 (由 `d` 指明) 输出变量 `sum` 的值。在本程序中输出 `sum=6`。

必须指出，输入语句和输出语句中的 `scanf` 和 `printf` 也是函数，它是由系统库提供的标准函数。当然，用户也可以根据自己定义函数，(将在函数一章中叙述)。

综上所述，C语言程序的基本结构为，

```
main ( )
{
语句
}
```

其中语句必须以分号结尾，因为分号是语句的终止符，它属于语句的一个组成部分。

思考题

1. 上述例题运行时，若输入 125，则输出结果如何？若输入 -3 呢？
2. 试将 C 语言的程序结构与你熟悉的其它语言 (如 BASIC, FORTRAN, PASCAL, COBOL, LISP, ALGOL, MACRO 等) 的程序结构进行比较。
3. 下列 C 语言程序有没有错误？若有，请指出。

```
/* file : error. c
 * date : 1-feb-84
 */
main ( ) ;
{ int a, b, c, sum;
  a := 1;    b := 2;
  scanf ("%d", &c),
  sum := a + b + c
  printf ("sum=", sum )
}.
/* The program is error program! */
```

§ 6 C 语言的基本符号

任何一个计算机系统所能使用的字符都是固定的、有限的。它要受硬件设备的限制。C

请言与其它语言一样。它的基本符号有英文字母、阿拉伯数字以及其它特殊符号。其中有：

英文字母26个（包括大写和小写）；

阿拉伯数字10个，即从0到9，

其它特殊符号主要指运算符，运算符通常由一至两个特殊符号组成，比如，

+, -, *, /, %, >, <, |, ^, ~, →等以及

==, !=, >=, <=, &&, ||, ++, --, >>, <<, (), []等。

标识符用来表示函数、类型或变量的名称，它是字母和数字的排列，但必须是字母开头。连字符_（低短横线）也称为字母，它在书写名称时很方便，大小写字母含意不同。标识符中只有前八个字符有意义，可以相互区别。但各种汇编程序和装入程序使用的外部标识符限制较多，用户必须注意有关资料的规定。

按照C语言的习惯，变量名用小写字母，符号常数全部用大写字符，函数名及外部变量名由六个字符组成。

根据程序设计的经验，标识符的选择采用“常用从简，专用从繁”的原则，一般不必太长，通常在六个字符之内能适应各种系统。

在UNIX操作系统中，C语言源程序的文件属性为.c。因此，完整的文件名称为***.c，这样有利于编译程序对C语言源程序进行编译。

C语言与其它语言一样，有一些具有特定含义的关键字，它们不允许作为用户标识符使用，一旦使用了就会出错。

关键字见附录A。

其中描述数据类型定义的一个：typedef；

描述存储方式的四个：auto, extern, static, register；

描述数据类型的九个：char, double, float, int, long, short, struct, union, unsigned；

描述语句的有13个：break, case, continue, default, do, else, for, goto, if, return, switch, while, sizeof。

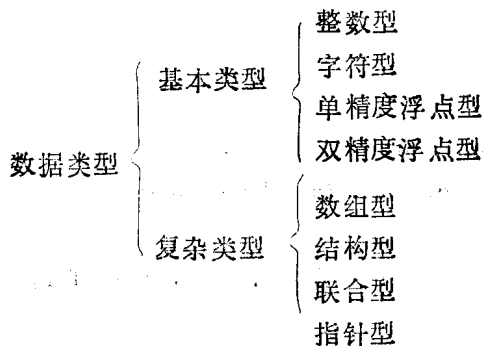
第二章 基本类型及运算

本章综合叙述C语言程序设计的一些共同问题。这主要是四种基本数据类型，若干种基本运算以及变量和常量的说明。具备这些知识，大大有利于后续各章节的学习。

§ 1 基本类型

程序设计中涉及的基本问题有两个，一个是数据的描述，一个是动作的描述。没有数据，程序就无法动作，而没有动作，程序就毫无作用。数据是动作的对象，而动作的结果会决定数据的内容。

数据在内存中存放，存放的情况由数据类型所决定。C语言有四种基本的数据类型和四种比较复杂的数据类型。复杂类型通常由基本类型构成，基本类型是复杂类型的元素类型（或基类型）。复杂类型又称为构造类型，将在第六章讨论。



其中单精度及双精度浮点型类似于其它高级语言中的实数类型，有时统称为浮点型。联合型与结构型共性较多，有时将联合型归纳到结构型之中。

一、整数型

整数型的标志是int，通常为一个字长（16位）。整数可以加限定词：

short表示短整数，一般与int相同；

long表示长整数，一般为双字长；

unsigned 表示无符号整数，即将16位的最高位作为数字位而不是作为符号位。

一般int的大小为-32768到+32767，short与之相同，unsigned为0到65535，long的范围是-2147483648到+2147483647。

二、字符型

字符型的标志是char，通常为一个字节（8位），大小为-128到+127位。实际使用了7位。字符在内存中通过其字符码而存在，一个字符对应一种字符码，字符码一般为0到127，其中32到126是可打印字符，共95个。其余的是不可打印字符，其中许多是控制字符，这些字符通常不能由终端键盘输入。

附录 B 提供了 ASCII 码字符集，并附加一些常用的非打印字符。

三、单精度浮点型

单精度浮点型的标志是 float，通常为两个字长（32 位）。

float 的大小为 -10^{38} 到 -10^{-38} ；0； 10^{-38} 到 10^{38} 。精度为小数点后取六位，实质上是小数点后第七位舍入到第六位。

四、双精度浮点型

双精度浮点型的标志为 double，通常为四个字长（64 位）。

在 PDP-11 计算机上，char 8 位；int 16 位；short 16 位；long 32 位；float 32 位；double 64 位。虽然 long 和 float 都占 32 位，但它们所表示的数的大小是完全不同的。

下表给出了几个不同机器下的基本类型数据及其精度。

	DEC PDP-11	Honeywell 6000	IBM370	Interdata 8/32
	ASCII	ASCII	EBCDIC	ASCII
char	8bits	9bits	8bits	8bits
int	16	36	32	32
short	16	36	16	16
long	32	36	32	32
float	32	36	32	32
double	64	72	64	64
range	$\pm 10^{\pm 38}$	$\pm 10^{\pm 38}$	$\pm 10^{\pm 76}$	$\pm 10^{\pm 76}$

VAX-11 除了 int 是 32bits 外，其余与 PDP-11 相同。

§ 2 基本运算

C 语言是提供运算符最多的高级语言之一。这些运算符的一部分与其它高级语言相同，另一部分则与汇编语言相似，甚至有人认为，可以用 C 语言代替汇编语言，这是因为 C 语言的指令级虽然高于硬件指令级，但运算对象却与硬件指令级相似。它的运算符反映了计算机底层硬件，它能对特定的物理地址进行访问，比如一个字的具体的位。

一、算术运算

算术运算符有两种：单目运算符和双目运算符。其操作数可以是四种基本类型，而字符型是通过字符码进行运算，其运算结果也是基本类型。

单目运算符仅有减法运算符（-），其结果是取操作数的负值。

双目运算符有五种，加（+），减（-），乘（*），除（/）和模（%）。加、减和乘与一般算术运算规则相同，但除法不同。若两个整数相除，则商舍去尾数，仅取整数部分。例如

5 / 10 等于 0，而 5.0 / 10.0 等于 0.5（前者是整除，后者为实除）。

模运算要求两个操作数都为整数，其结果为两个操作数整除后取余数。例如

5 % 10 等于 5，而 10 % 5 等于 0；

5 % 4 等于 1, 而 5 % 3 等于 2。

思考题

1. 5 % 2 等于 3 对不对? 为什么?
2. 若 $a < b$, 则 $a \% b$ 等于 a , $b \% a$ 等于 0 对不对? 为什么?

二、关系运算

关系运算符有六种: 等于 ($==$), 不等于 ($!=$), 大于 ($>$), 大于等于 ($>=$), 小于 ($<$), 小于等于 ($<=$)。

关系运算要求两个操作数是同一种数据类型, 其结果为是或否。但是C语言没有布尔型数据假 (false) 和真 (true), 它是通过整数 0 和非 0 来表示的。

三、逻辑运算

逻辑运算符有三种: 与 ($\&\&$), 或 ($\|\|$), 非 ($!$)。它要求操作数为整数零和非零, 其运算规则为:

1. 如两个操作数都为非零整数, 则相与的结果为 1, 否则为 0;
2. 如两个操作数都为零整数, 则相或的结果为 0, 否则为 1;
3. 如操作数为非零整数, 求非的结果为 1, 否则为 0。

四、按位运算

按位运算符有三种, 即:

按位与 ($\&$), 两操作数逐位求与;

按位或 ($\|\|$), 两操作数逐位求或;

按位加 (\wedge), 两操作数按位相加。

两个操作数都必须为整数。

五、求反运算

通过单目反码运算符 (\sim) 求操作数的反码。注意与单目减法运算符 ($-$) 和逻辑非运算符 ($!$) 相区别。

例如 $x = 7$; $y = \sim x$; 则 y 为 -8 。

$x = 7$; $y = -x$; 则 y 为 -7 。

$x = 7$; $y = !x$; 则 y 为 0。

六、自增自减运算

通过增 1 运算符 ($++$) 或减 1 运算符 ($--$) 将操作数自己增 1 或减 1。

$++n$ 等价于 $n = n + 1$ 。

$--n$ 等价于 $n = n - 1$ 。

运算符不仅可以作为一个变量的前缀 (如上例), 也可以作为一个变量的后缀, 但意义完全不同。自增或自减只对变量而言, 对表达式是不允许的, 程序设计时必须注意。

例如, 已知 $y = 5$, 则

$x = ++y$; 表示 y 自增后赋给 x , 即 $y = y + 1 = 5 + 1 = 6$, $x = y = 6$ 。

$x=y++$; 表示 y 赋给 x 后自增, 即 $x=y=5$, $y=y+1=5+1=6$ 。

七、移位运算

移位运算有两种, 即

右移运算符 ($>>$), 表示将操作数右移。

左移运算符 ($<<$), 表示将操作数左移。

例如 $x>>2$ 表示将 x 右移两位;

$y<<3$ 表示将 y 左移三位。

在左移或右移之后, 原操作数的值就发生了变化。

八、组合运算

C语言不仅有基本的赋值运算符 ($=$), 而且它能与 $+$, $-$, $*$, $/$, $\%$, $>>$, $<<$, $\&$, $\&$ 等运算符形成组合运算符, 进行自反操作。组合运算符是自右而左组合的, 两者之间不能有空隔。举例如下:

$x=-10$; 表示 $x=x-10$, 不是 x 等于 -10 ;

$x=\square-10$; 表示 x 等于 -10 ; \square 表格空白符。

$x=-\square 10$; 表示 $x=x-10$ 。

对于初学者来说, 为了表示 x 等于 -10 , 最好采用 $x=0-10$;

又如:

$y=y<<3|x$; 表示 y 左移三位后与 x 进行按位或运算。

$y>>3|x$; 表示 y 右移 (右移的位数为3和 x 按位或运算的结果)。

组合运算是C语言的一大特点, 使用起来很方便, 但是人们往往不习惯, 容易出错。

九、其它运算

C语言还有条件运算以及一些特殊的运算符。条件运算符为 $?:$, 其基本格式为

表达式 1	?	表达式 2	:	表达式 3
-------	---	-------	---	-------

即先计算表达式1 (条件表达式) 的值, 若非零取表达式2的值, 否则取表达式3的值。它是条件语句的一种简化形式。

逗号运算符, 常用于循环语句for语句。

部分选择符 \cdot 和 \rightarrow 分别用于结构和指针;

地址运算符 $\&$ 用于指针;

访问运算符 $*$ 用于指针。

圆括号 ($()$) 和方括号 [$]$] 的使用也很广泛, 特别是圆括号。

上述运算符的功能将在后续章节中逐步介绍。

表2.1给出C语言的全部运算符的优先级和结合性的规则。在同一行上的运算符具有相同的优先级, 而在同一列中的优先级是按自上而下顺序递减的。

附录C提供了运算符的分类、优先级和结合性。