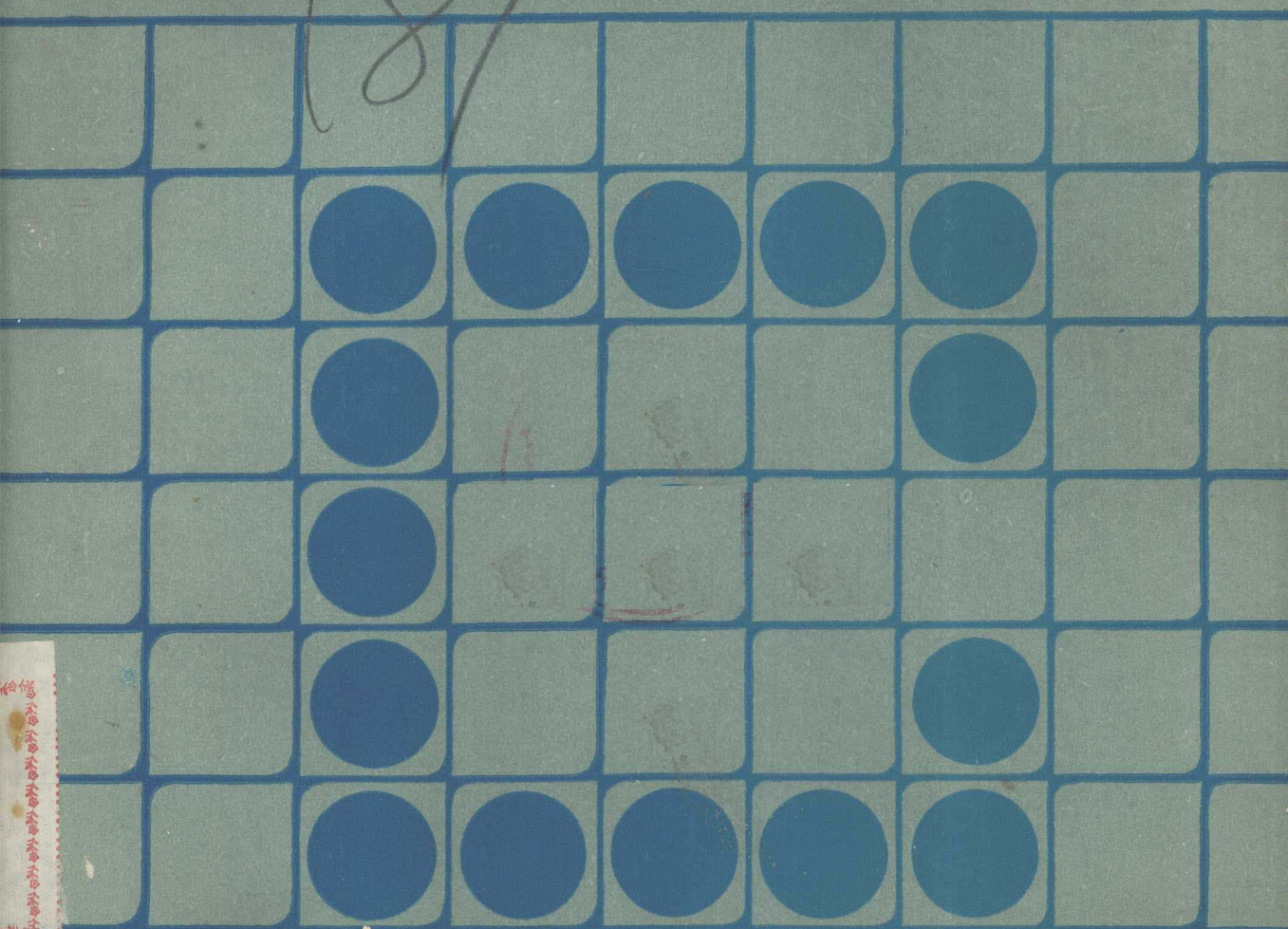
中国计算机技术培训网统编教材

C语言程序设计

殷树勋 主编



北京科学技术出版社

C 语言程序设计

殷树勋　李力充　陈　军　陆　军　编
殷　树　勋　主　编

北京科学技术出版社

内 容 提 要

C 语言是一种通用的程序设计语言。它具有丰富的数据类型、灵活多样的运算符、良好的控制流和数据结构，以及简洁的表达形式，能有效地用来描述操作系统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。

本书以国产长城系列微型机上所配置的 Lattice C 编译程序为主要蓝本，介绍C语言的数据类型、各种语句，并通过大量程序实例介绍C 语言程序设计的方法。全书重点突出，叙述简明，便于自学。

本书可作为计算机技术人员的培训教材，也可作为大专院校教材及其他科技人员自学参考书。

C 语言程序设计
[redacted]勋 主编

北京科学技术出版社出版
（北京西直门外南路19号）

新华书店首都发行所发行　　各地新华书店经售
中国科学技术情报研究所印刷厂印刷

*

787×1092毫米　16开本 12 印张 289 千字
1988年 5 月第一版 1988年 5 月第一次印刷
印数1— 7·200 册

ISBN7-5304-0204-8/T·23　　定价：2.80 元

出 版 说 明

在世界新技术革命中，计算机已成为一个崭新的力量，成为最活跃，最先进的核心技术之一，在信息社会中发挥着她的强大威力。在这种形势下，要使我国计算机应用事业尽快地赶上世界先进水平，人才的培养是十分重要的。电子部计算机技术培训中心和中国计算机技术服务公司技术培训网担负着在全国范围内对计算机应用人才进行培养的重任。

为了能迅速、有效地提高计算机技术培训的质量，使技术培训向正规化、系列化、分层次方面发展，为在我国建立一支宏大的应用计算机的队伍，电子部计算机技术培训中心、中国计算机技术服务公司培训网和中国计算机学会技术培训学组组织了培训网系统内的培训中心、培训部门及部分高等院校、科研所、计算机生产厂等单位的计算机专家组成了全国计算机技术培训网教材编审委员会。教材编审委员会从国内外计算机技术发展和我国实际情况出发，会同北京地区六家出版社，经过有计划地选题、拟定大纲、大纲审定、指定主审主编、具体编写、审稿、编辑、出版、印刷、发行等整个环节，在全网范围内已经编写教材八十余种。自一九八六年开始编委会在事务处理、工业控制、微机局部网络、微机硬件分析和维修以及中华学习机等方面组织了一批丛书和系列教材。这些教材从一九八七年开始陆续与广大读者见面。

这些教材的主要对象是非汉字计算机专业的广大科技人员和管理人员（在培训过程中将分成初、中、高各级技术人员的层次培训），也可以作为高等院校的教学参考书及大专院校学生和从事计算机应用人员的自学教材。

这些教材本着两个指导思想进行编写，即实用性强：让读者学完后能立即用上；跟踪新技术，新成果、新趋势快：让读者及时掌握最先进的技术服务于社会。在培训工作方面遵循三条宗旨，即面向全国、面向应用、面向用户，为读者用好计算机服务。

我们热忱地欢迎有更多各方面的计算机专家参加培训教材的编写工作，热忱欢迎广大读者进行批评和帮助，也热诚欢迎更多的出版社支持我们的工作。

中国计算机技术培训网教材编审委员会

1987年5月

编 委 会 名 单

前　言

C语言是一种通用程序设计语言，它具有表达简洁、控制流与数据结构先进和操作功能丰富等特点。它原来是在UNIX操作系统上发展起来的，并且UNIX本身就是由C语言写成。然而C语言并不局限于某一操作系统或机器，因此很容易写出在支持C语言的任何机器上运行而不需要作修改的程序，就是说它具有优良的“可移植性”。C语言不但常用来编写系统程序，而且对于写数据处理程序也同样很有用。

由于C语言具有上述特点，因此C语言在微型机上也流行起来，成为微型机上有影响的一种程序设计语言。许多的软件开发人员常用它来编写数据通信软件和CAD（计算机辅助设计）软件。

IBM PC-XT及国产长城0520系列微型机和长城286微型机等都支持多种版本的C语言编译程序。广大计算机用户越来越广泛地希望尽快地掌握C语言，以适应软件开发工作的需要。本书就是为适应这种要求而编写的。

本书先从C语言程序结构开始，由浅入深地介绍数据类型、变量、程序流程控制、数组、函数、结构、指针等内容，以帮助读者尽快地掌握C语言。每个章节都配有程序实例，所有实例都已经过直接测试。作者希望通过这些程序实例能展示出C语言程序设计的风格和原则。

本书是计算机技术培训教材，适用于各种技术培训班，同时也可用作高等学校的C语言教材。

本书的第一、二、三、七、九章由殷树勋同志编写，第四、五、八章由陈军同志编写，第六章由陆军同志编写，第十、十一、十二、十三、十四、十五章由李力充同志编写。此外，李力充和陈军同志负责整理、编写了附录。殷树勋同志担任全书的主编。

由于编者水平有限，且时间仓促，缺点与错误在所难免，敬请读者批评指正。

编　者

1987年11月

目　录

第一章 概 述

1.1 C语言简介

C语言最初是在1972年，由丹尼斯·瑞切（Dennis Ritchie）为DEC公司PDP-11计算机配置UNIX操作系统而设计出来的。这个操作系统以及所有基本的UNIX应用程序都用C语言编写而成。

C语言是一种通用程序设计语言。尽管C语言是在UNIX系统上发展起来的，但它并不只限于某一种操作系统或机器。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硬件环境下的工作过程。由于用C语言编写的程序具有优良的“可移植性”，因此不需要作大的修改就能在任何支持C语言的机器上运行。

C语言具有以下主要特征：

（1）以英文小写字母为基础。但在不使用小写字母的微型机中也允许用大写字母。

（2）程序由函数的集合构成，函数参数则按“值调用”的方式进行。

（3）指针可作为数据来处理。

（4）运算符多，有利于编写程序。

（5）通过预处理可以进行宏调用。

（6）程序表达简洁。

（7）无处理字符串的特别功能，但能区分字符及字符串。

（8）输入输出功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，但能通过操作系统来解决。

由于上述特点，使得C语言在微型机上也流行起来，成为微型机上有影响的一种程序设计语言。

目前，国产长城系列微型机配有多种不同版本的C语言编译程序，本书所介绍的是其中的Latticec可移植编译程序。这是一种功能较齐全的语言版本，并且已对它进行了汉化。与标准C文本相比，Lattice C编译文本存在着某些差异。这些差异，对于那些过去已熟悉了标准文本的人来说，可能会感到有些不便，但是对于未曾使用过别的版本的人来说，却带来了许多方便，因为Lattice C所作的大部分改动都是基于使语言更易移植和更易理解的考虑之上作出的。关于二者间存在的差别，我们将结合具体内容的介绍予以说明。

1.2 C语言的程序结构

任何C语言程序都是由一个或几个函数组成的。下面请看一个简单的C语言程序。

程序1-1

```
/* 该程序实现两整数相加并显示结果 */
```

```
main ()
{
  int  value1, value2, sum;
  /*赋值和计算结果*/
  value1 = 50;
  value2 = 25;
  sum = vaule1 + value2;
  /*显示结果*/
  printf ("%d和%d的和是%d\n", value1, value2, sum);
}
```

程序1 1输出

50和25的和是75

这个程序的功能是求两个整数50和25的和并输出结果。

在C语言中，用/*……*/作为程序的注释部分。在本例中，分别在程序的三处都作了注释。当注释在一行中写不下时，可另起一行，但一般要求该行以*开头。在标准文本中，不允许注释嵌套，但在Lattice C编译程序中则允许注释嵌套，即只要在注释结束前使每个/*都有一个*/与之对应就行。

程序的主体部分为

```
main ()
  {……
  }
```

其中main是一个函数，且是一个特殊的函数，所有C语言程序都必须有main函数。一般来说，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，但在一般情况下却没有参数，即参数表为空，这时一对圆括号()必须有，不能省略。

花括号{}将组成该函数的那些语句括起来。

int value1, value2, sum; 是说明语句，它说明value1、value2和sum是整数型变量。接下来是两句赋值语句，将变量value1和value2分别赋值为50和25。sum = value1 + value2;也是赋值语句，它将value1和value2求和后赋值给sum。最后是输出语句，它将计算结果输出到显示器上。每句语句都应以分号结束。

需要指出，上述输出语句中的printf也是函数，它是由系统提供的标准库函数，因此本例中的输出是通过调用标准库函数来实现的。标准库函数printf的详细内容，我们将在后面逐步介绍。

1.3 程序的编译、连接和执行

采用C语言等高级语言编写的程序称为源程序。计算机不能直接运行源程序，因为它不能理解源程序中各种符号的含义。要使源程序变成能运行的程序，必须将该源程序的各条语句“翻译”成能被计算机接受的机器指令，即生成目标程序。承担这种“翻译”工作的程序就是所谓“编译程序”。

Lattice C编译程序分两个阶段对一个C语言源程序进行编译。图1-1表示一个C语言源

程序从编译、连接到执行的过程。

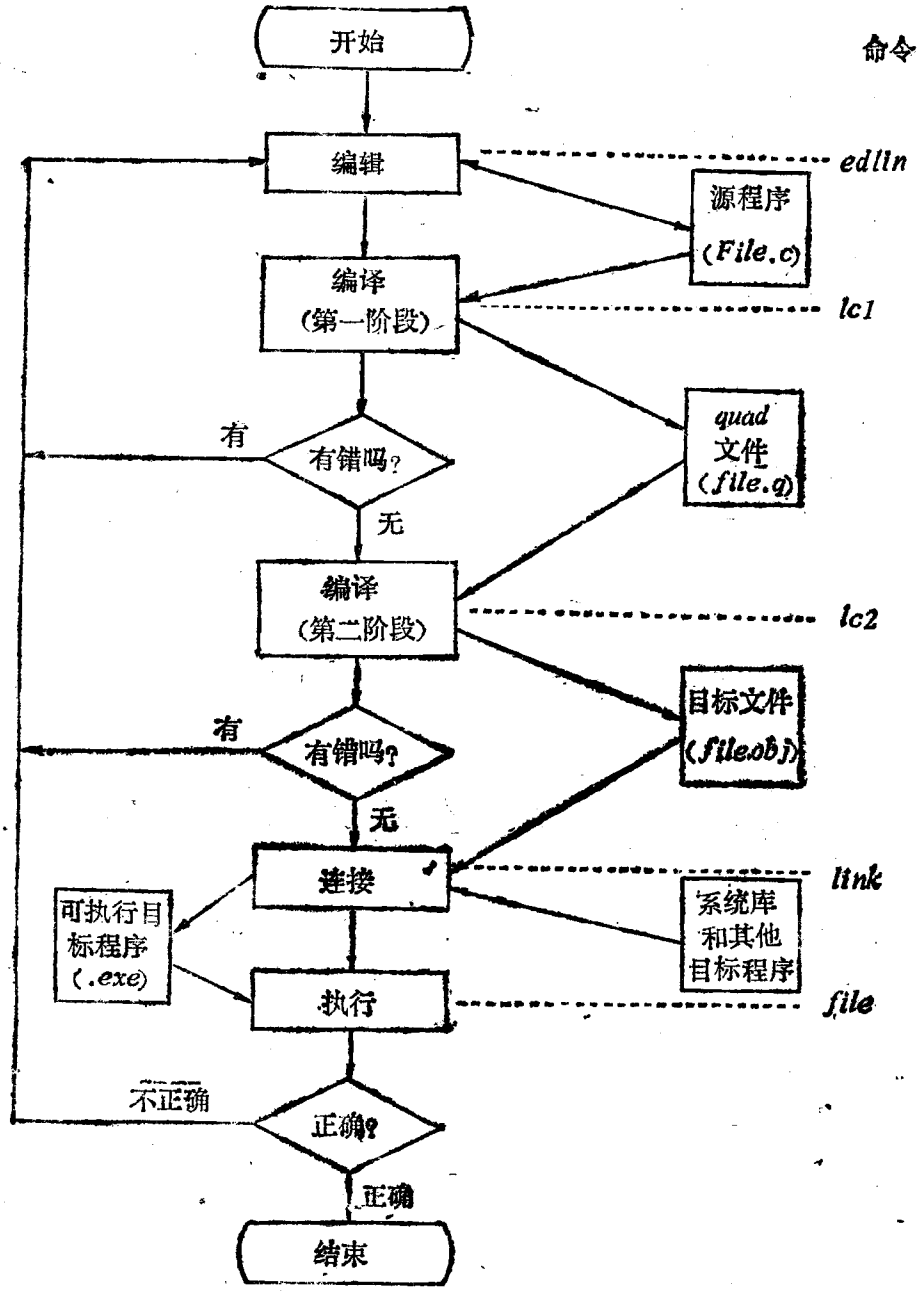


图 1-1 C程序的编译、连接和执行

通常先利用操作系统提供的文本编辑程序（例如PC-DOS提供的edlin）编辑生成C语言源程序（注意：源程序文件名后一定要带扩展名C），然后对该源程序进行编译。下面假设某源程序文件名为file.c，来说明C语言源程序的编译、连接方法。

步骤一：执行编译的第一阶段。

键入命令：

lc1 b:file<ENTER>

其中盘符b:表示文件在B盘上，如不带盘符则认为是现行盘；文件名后的扩展名c可带可不带。符号<ENTER>表示按回车键。

编译的第一阶段是对源程序进行文法和句法分析，同时执行所有的预处理函数（关于C预处理程序在后面要作介绍），生成包含程序中各种标识符信息的符号表并产生称为“guad”的中间文件（以q为扩展名，即file·q），它提供了程序所说明的基本功能。编译过程中如发现程序有错误，则编译程序将给出错误提示信息。根据错误提示信息修改程序后，再一次进行第一阶段的编译，直至编译通过。

步骤二：执行编译的第二阶段。

键入命令：

lc2 b:file<ENTER>

该阶段对在第一阶段中生成的中间文件，即file·q进行扫描，生成一个Intel 8086格式的目标文件file·obj。第二阶段将给出如下形式的信息：

Module Size P = pppp D = dddd

其中“pppp”指明模块可执行部分的字节数(十六进制数)；“dddd”则说明数据部分的字节数（十六进制数）。

和第一阶段一样，作为第二阶段的输入文件的file·q，扩展名q可以不指定。

步骤三：连接。

键入命令：

link cs+b:file, b:file, , lcs<ENTER>

程序连接是将源程序编译后生成的目标文件连接起来形成可执行文件。连接是不可缺少的，因为：第一，编译第二阶段生成的目标文件处在不可运行状态；第二，一般C程序的执行是从定义为“main”的函数开始的，但在调用“main”函数之前必须完成大量与系统相关的处理，而完成这些处理的模块是通过连接组装到程序中来的；第三，多数程序要使用一些未在当前模块中定义的函数，必须将它们连接在一起才能运行。

连接时要求有三种类型的输入：

（1）起动文件cs．obj 该目标文件定义了Lattice C编译程序下所有被编译的C语言程序的PC-DOS入口。命令中不键入扩展名obj。

（2）用户生成的目标文件 在本例中就是file．obj。命令中不键入扩展名obj。

（3）库文件lcs．lib 它必须指定为在连接期间用来进行检索的程序库。命令中不键入扩展名lib。

命令中的第二个file是运行文件名，连接的结果将生成运行文件file．exe。当然，用户也可以起另外的名字作为运行文件名。后面的两个连续逗号表示产生一个连接报表文件，文件扩展名为·map。在本例中则产生名为file·map的连接报表文件。如果在两个逗号间键入NULL，则每次连接结束就生成名为NULL·MAP的连接文件，文件中的信息保持最新的信息，即本次连接的报表信息复盖掉上一次连接的报表信息。连接报表文件中存放着程序段和数据段的地址、长度等信息。

在连接过程中，如果连接程序找不到命令中所列出的某个目标文件，则将停止处理且不生成·exe文件。当连接程序在指定的目标文件中未能找到全部外部项时，将出现错误信息

“Unresolved Externals”，后面紧跟着未定义的外部项名。

连接成功后，生成可执行文件，在本例中即文件file·exe。执行方法是键入文件名（不带扩展名），即：

b：file〈ENTER〉

第二章　数据类型及运算

在这一章里，我们将讨论数据的基本类型及其运算，以及变量和常量的说明。

2.1 数 据 类 型

数据是程序处理的对象，因此程序设计的基本问题之一就是如何对数据进行描述。C 语言有四种基本的数据类型，即：整数型、浮点型、双精度浮点型和字符型。

2.1.1 整 数 型

整数型的标志是int，由一个或多个数字序列组成，数字序列前如有负号则表示是一个负整数。对于16位微型机来说，可表达的整数范围是－32768～＋32767。

整数型数据除了十进制数外，还包括八进制数和十六进制数。如果某一整型数的第一位为数字 0，则表示该数是八进制数。例如，0771表示一个八进制数，其十进制值为505（7×64＋7×8＋1）。如果某一整数以ox开头，则表示该数为十六进制数，后面紧跟的数位由 0～9 和字母a～f（或A～F）表示，字母a～f所表示的值为10～15。例如，ox5EB表示一个十六进制数，其十进制值为1515（$16^2\times5+16\times14+11$）。

2.1.2 浮 点 型

浮点型的标志是float，它定义了一个32位带符号的浮点数，其中 8 位为二进制指数，24 位为以规格化形式存放的小数。精度大致相当于 6 或 7 个十进制位，可以表示$\pm10^{37}$～$\pm10^{38}$范围内的浮点数。

2.1.3 双 精 度 型

双精度型的标志是double，类似于浮点型数，只是精度比浮点型(float)高，它定义了一个64位的带符号浮点数，其中11位为二进制指数，53位为以规格化形式存放的小数。精度大致相当于15或16个十进制位，可以表示$\pm10^{307}$～$\pm10^{308}$范围内的浮点数。

2.1.4 字 符 型

字符型的标志为char，它定义了一个 8 位的无符号整数。根据标准ASCII码格式，生成用ASCII码表示的文本字符时，第 7 位复位。

2.1.5 限定词 long、short和unsigned

正如浮点数可以通过double类型扩大精度，整型数可以通过加限定词扩大精度。限定词有long、short和unsigned。下面分别介绍这三个限定词的含义。

（1）限定词long

限定词long或long int定义了一个32位的带符号整数，可表示的整数范围是－2147483648～＋2147483647。

（2）限定词short

限定词short或short int与int同义。

（3）限定词unsigned

限定词unsigned或unsigned int定义了一个16位的无符号整数，可表示的整数范围为0～＋65535。

2.2 变量说明

在C语言程序中，所有的变量都必须加以说明，没有隐含规则。变量说明的格式如下：

类型　　变量表；

举例：

```
int a, b, c;  (表示a、b、c为整数型变量)
char x, y;  (表示x和y为字符型变量)
float f1, f2;  (表示f1和f2为浮点型变量)
short s1, s2;  (表示s1、s2为短整数型变量，其意义与int s1, s2; 相同)
unsigned xyz;  (表示xyz为无符号整数型变量)
```

变量名由1到8个字符组成，一般以小写字母开头，后面跟字母或数字。变量表中，变量之间用逗号隔开，变量说明语句以分号结束。

2.3 常　　量

C语言中规定了两种常量：数值常数和字符常数。

数值常数包括整常数和浮点常数。整常数由一系列数字组成，可以是十进制数、八进制数和十六进制数。浮点常数就是实数。每个浮点常数都取双精度。

字符常数是括在单引号内的一个字符，例如'a'。其值为该字符的ASCII码值。如字符'a'的ASCII码值为97或0141（即八进制141）。

此外，某些非图形字符、单引号本身以及反斜线等按下表中的换码序列来表示：

换行	LF	\n
水平制表	HT	\t
退一格	BS	\b
回车	CR	\r
走纸	FF	\f
反斜线	\	\\
单引号	'	\'
位型	ddd	\ddd

其中，换码字符\ddd由反斜线后跟1、2或3个八进制数组成，这些八进制数用来表示所希望的字符值。例如\0表示字符NULL，它的ASCII码值为0。

字符串是由双引号括起来的字符序列，如“ABC”，编译程序在每个字符串末端都加上

空字符（NULL）'\0'，表示字符串的结束。

2.4 算术运算和算术表达式

2.4.1 算 术 运 算 符

算术运算符有两种：单目运算符和双目运算符。

单目运算符只有减法运算符（－）一种，它是取操作数的负值。例如－a，即取变量a的负值。

双目运算符共有五种：加（＋），减（－），乘（＊），除（/）和模运算符（%）。加、减和乘与一般算术运算相同，但除法稍有不同。若两整数相除，则商去掉尾数，仅取整数部分。如欲避免这一情况，应改为实数相除。例如

5/10等于0，而5.0/10.0等于0.5。

模运算要求两个操作数都是整数，其结果为两操作数整除后取其余数。例如

5%10等于5，而10%5等于0；

13%4等于1，13%5等于3，等等。

2.4.2 自增自减运算

自增运算符为＋＋，其功能是使操作数本身加1。自减运算符为－－，其功能是使操作数本身减1。例如

＋＋n等价于n=n+1。

－－n等价于n=n-1。

自增自减运算只适用于变量，不能用于表达式。自增自减运算符可作为变量的前缀（先自增自减），也可作为变量的后缀（后自增自减），二者的意义是不同的。

例如，已知x=1，则

y=＋＋x；表示变量x先自增1，然后再赋给y。结果为y=x=2。

y=x＋＋；表示变量x先赋值给y，然后再自增1。结果为x=2，y=1。

2.4.3 算术表达式和赋值运算符

如上所述，加法运算符（＋）用来实现两个数（或变量）的相加，减法运算符（－）用来实现两个数（或变量）的相减，乘法运算符（＊）用来实现两个数（或变量）的相乘，除法运算符（/）用来实现两个数（或变量）的相除等等。这些变量和算术运算符的组合就称为算术表达式。那么，在存在两个以上运算符的算术表达式中，如何决定运算次序呢？下面是一个程序例子。

程序2-1

```
/*说明各种算术运算符*/
main ()
{
  int a=100;
  int b=2;
```

```
    int c=25;
    int d=4;
    int result;
    result=a-b; /*减运算*/
    printf ("a-b=%d\n", result);
    result=b*c; /*乘运算*/
    printf ("b*c=%d\n", result);
    result=a/c; /*除运算*/
    printf ("a/c=%d\n", result);
    result=a+b*c;
    printf ("a+b*c=%d\n", result);
    printf ("a*b+c*d=%d\n", a*b+c*d);
}
```

程序2-1输出

```
a-b=98
b*c=50
a/c=4
a+b*c=150
a*b+c*d=300
```

从程序运行结果可以看到，在存在+、-、*、/四种算术运算符的表达式中，*、/的优先级别比+、-高，这和先乘除后加减的四则运算规则是一致的。

表2-1列出了C语言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则按自上而下的顺序递减。表中有很多运算符尚未提及，它们将在后面逐步介绍。

表 2-1 运算符优先级及结合性

运算符	结合性
() [] → .	从左到右
! ~ ++ -- * *sizeof	从右到左
* / %	从左到右
+ -	从左到右
≪ ≫	从左到右
<<= > >=	从左到右
== ! =	从左到右
&	从左到右
∧	从左到右
|	从左到右
&&	从左到右
||	从左到右
? :	从左到右
= += -=等	从右到左
,	从左到右

C语言中的赋值运算符（=）能与+，-，*，/，%等运算符形成组合运算符。组合运算符是自右而左组合的，中间不能有空隔。

例如，式

```
i=i+2;
```

中，左边被右边重复了，因此可用组合运算符压缩为

```
i=+2;
```

或 `i+=2;`

式中，除了赋值运算符左边的变量外，其他的变量和常数都只计算一次。此外，组合运算符的优先级除了逗号外，比其他运算符都低（参阅表2-1）。因此，赋值语句

```
x*=y+1;
```

实际上是

```
x=x*(y+1);
```

而不是

```
x=x*y+1;
```

2.5 类型转换

当不同类型的操作数出现在表达式中时，将根据一定的规则隐含地转换成相同类型。下面先看一个程序例子。

程序2-2

```
/*基本类型转换*/
mafn ( )
{
  float f1=123.125,f2;
  double d1,d2=12.1232152;
  int  i1, i2=150;
  long int l1, l2=4500000;
  char  c='a';
  i1=f1;       /*浮点型转换为整数型*/
  printf ("%f 赋值给整型变量: %d\n", f1, i1) ;
  f1=i2;       /*整数型转换为浮点型*/
  printf ("%d 赋值给浮点型变量: %f\n", i2, f1) ;
  f1=i2/100; /*整型数相除*/
  printf ("%d 除以 100 : %f\n", i2, f1) ;
  f2=i2/100.0; /*整型数除以浮点数*/
  printf ("%d 除以 100.0: %f\n", i2, f2);
  d1=f1+d2; /*双精度数和浮点数相加*/
  printf ("%f 加%-15.8f 等于%-15.8f\n", f2, d2, d1) ;
  l1=l2/i2; /* 长整数和短整数相除*/
  printf ("%-ld除以%d等于%-ld\n", l2, i2, l1) ;
```