实用软件详解丛书

实用C语言详解

●徐宝文　编著

●電子工業出版社

内容简介

本书与其它同类书籍的最大不同，也是该书的主要特点之一是：本书是以词条的形式对C语言进行了全面而深入的解析。其优点是对不管熟悉或不熟悉C语言的人在使用C语言时都不可避免地要查阅有关的内容，而该书以中文拼音和英文字母为序进行编排，使读者可以快捷而准确地找到自己所要的东西。

本书收集了覆盖K & R·C，ANSI C标准以及有关各种版本诸如Microsoft C、Quick C. UNik C、Turbo C Borlond C++等的全部命令、功能、编程和应用，以及他们之间的比较等全部内容。只要有关C的词条，在本书中都能查到。

每一词条除给出其性质、功能、描述、一般用法等等外，还给出了一些实例，这些实例除说明本词条的使用外，并结合实际讲解了编程的方法和技巧。本书是一本实用性很强的书。

本书很适用于所有C使用者。

实用软件详解丛书

实用C语言详解

徐宝文　编著

责任编辑　王昌铭

*

电子工业出版社出版

北京市海淀区万寿路173信箱(100036)

电子工业出版社发行　各地新华书店经销

电子部情报所印刷厂印刷

*

开本：787×1092毫米 1/16　印张：34.5　字数：1310千字

1996年1月第1版　1996年1月第1次印刷

印数：60[illegible]册　定价：38.00元

ISBN7-5053-3245-7/TP.1201

序　言

人类社会已进入信息时代，而信息时代的基础则是计算机和通信以及两者的紧密结合。这种结合正在改变着人们的生活、学习和工作方式，推动着社会的进步。计算机作为信息社会的支柱产业之一，逐渐形成了一种**新兴的计算机文化**。作为一种文化，人人必然要和计算机交往。这种交往是用与计算机硬件不可分割的软件来实现的。软件的第二次开发涉及到所有要使用计算机的人们。因此，为使自己能适应、驾驭这瞬息万变的世界，必需掌握必要的软件知识并自如地应用它，以使自己在当今激烈的社会竞争中多一块成功的砝码，也为我们的事业蕴藏一份能量。

我们深为计算机科学技术发展的日新月异以及计算机对科学技术与人类社会的发展所起的巨大推动作用感到欢欣鼓舞，同时也感到自己有责任、有义务为计算机及软件技术在我国的普及、提高与应用多作一份贡献，为社会主义的信息事业多尽绵薄之力。为此，在电子工业出版社的支持与鼓励下，经征求各方专家意见，决定编写一套《实用软件详解丛书》。

这套丛书面向软件的开发与应用，兼顾初、中、高等各个层次的读者与用户，力争普及与提高相结合。本丛书将跟踪计算机软件技术的最新发展，组织专人编写，分批出版。其中每一本书的主编均请工作在计算机软件教学、科研与应用第一线的专家担任，力争以较高的质量满足广大读者的需要。

本丛书的特点是“实用”与“详解”。所谓“实用”，指每一本书均面向广大计算机用户，少谈理论，多论使用，使读者能把所学到的知识直接应用到教学、科研与生产中；所谓“详解”，指对所涉及到的每一概念、每一问题都尽可能作全面深入的阐述，力戒蜻蜓点水，使读者读后真正受益。

竭诚欢迎广大读者对本丛书提出批评与改进意见。

《实用软件详解丛书》编委会

一九九四年四月

《实用软件详解丛书》编委会

前　言

C是一种通用程序设计语言，它具有丰富的表达式、现代控制结构与数据结构，它既不大也不非常高级，这在过程性程序设计语言历史上是反传统的。如果不以过高的标准去要求，C语言主要有如下一些优点和特点：

表达能力强　可直接处理地址、字符与数字，可完成很低级的机器级算术、逻辑运算，可用于编写操作系统及有关的系统软件与支撑软件，在这方面它的功能相当于汇编语言。

简洁、短小，编译程序可以在内存很小的机器上运行　它尽可能使各种单词所含字符数少（如，用“{”与“}”取代Pascal等一般语言中的“bggin”与“end”，在表达式中不用标式符式的运算符符号）。它没有提供同步、互斥、并行处理、异常处理等复杂设施，许多功能都通过系统调用等来支撑。

数据类型丰富，数据结构描述能力强　C既提供了字符、整数、实数（浮点数）等基本纯量类型，也具有诸如数组、结构、联合等构造类型及指针类型；能用来实现各种复杂的数据结构（如各类链表、树、栈等）的操作与表示。其指针类型比一般语言中同一类型更为低级，在某种意义上就是机器地址的对应物，这使得它使用起来更为灵活、多样，当然，这也带来了可靠性与可读性等方面的一些问题。变量、函数等可具有多种存储类，有助于实施数据隐茂与模块化程序设计。

控制结构丰富、结构化程度高　C具有丰富的顺序、选择（if与switch）和循环（while与do while）控制结构，而break与continue语句等则有助于减少程序中的goto语句的使用、降低程序复杂性。如果运用得当，可以设计出结构良好的程序。

运算符丰富、简洁、多样　C语言共有34种运算符，这恐怕是现有有名的语言中运算符最多的。C不仅包含常见的算术、逻辑、关系等运算符，而且把括号、逗号，甚至于赋值号都当作标点符号处理。这使得C所能处理的运算种类极其丰富，表达式种类多样化，灵活而有效地使用各种逻辑符可以实现在其它高级语言中难以实现的运算功能。

目标代码质量高　这里指时间、空间质量两个方面。一般高级语言中由于一些结构不能直接反映汇编或机器语言结构，加之目标代码中要为完备性与安全性等检查插入检查代码，因此，尽管编译程序作了优化处理，但运行效率很低。而C语言的各种结构（如指针、数组）却力图反映机器指令与编码结构，编译程序不需作较多的工作即可生成时空效率较高的代码。据有关统计，C的目标代码效率一般只比汇编语言低百分之十至十二，这是一般高级语言无法比拟的。

C将大部分决策留给程序员去做，对诸如类型转换、类型兼容性检查等工作限制很少。但要注意，虽然这不失为C的一大优点，但如果没有一个好的风格去指导而不加节制的使用这种灵活性，有可能编写出质量（不是效率）很低的程序，特别是对没有成熟的程序设计经验的人更是如此。

可移植性好　这是许多C的倡导者与支持者都强调的C的一大优点。他们认为，汇编语言由于过于依赖于机器而很难移植；而其它一些高级语言，如Pascal与FORTRAN，现有各种机器上的编译程序基本上都是按照标准或自己的要求重新实现的，很少有哪一个是移植其它机器上的编译程序的。而目前许多机器上所配置的C编译程序一般都是通过移植得到的，而

统计资料又表明，各种机器上的C编译程序中至少有百分之八十的代码是公共的。

C语言最早是由贝尔实验室的D. M. Ritchie等人在总结了ALGOL60、CPL、BCPL、B等语言经验的基础上设计出来的，是为了描述和实现UNIX操作系统而研制的。C语言问世后又作了多次改进与完善，但在最初几年一直未能推向社会，主要还是在贝尔实验室内部使用。在某种意义上说，直到UNIX被社会真正接受后，C语言才真正引起了人们的注意。到七十年代末八十年代初，C语言已经可以在各种大、中、小、微型机上使用，并能独立于UNIX而存在。C语言在其发展过程中出现了很多不同的版本，这些版本虽然都保持了C的基本特色，但各版本之间的差异严重妨碍了C语言的应用，因而人们迫切需要对C进行标准化。为些，美国ANSI于1985年提出了正式标准草案，1989年ISO/IEC提出国际标准草案，随后ANSI、ISO/IEC的正式标准公布(即所谓标准C或ANSI C)，从此，C就有了一个通用的国际标准。

到了八十年代初，B. Stroustrup系统地作了在C语言中加入类设施的研究，并在1983年、1984年间研制出了C++语言，C++是在C的基础上扩充了类、继承、重载、虚函数、模板等面向对象设施而形成的一个新的面向对象语言，它与C的兼容性极好，所有正确的C程序几乎都是正确的C++程序。正如C与UNIX相得益彰一样，C++在某种程序上几乎成了面向对象程序设计语言的代名词。

目前在微机上流行的C与C++版本有Microsoft C/C++、Turbo C/C++、Borland C/C++、Visual C/C++等，它们都与ANSI C(ISO C)兼容，并在后者的基础上作了适当的扩充。因此只有先学好ANSI C(ISO C)才能真正学好用好各种C与C++编译系统。为了帮助读者学好用好ANSI C与C++，我们编写了这本书。

本书以词条形式详细介绍、分析、讨论了C的各个词法、语法、语用现象以及各个标准库函数。对每一词条，除了作详细介绍外，还对其使用风格、使用技巧以及在使用中应注意的问题作了十分详细的讨论与分析，在必要时还给出了实例，而这是许多C语言教科书及参考书所望尘莫及的。因此，本书既适合于C语言的初学者作为学习C语言的基本参考书，也可以作为有一定C语言基础的读者进一步研究C语言的参考书。

参加本书资料收集、编写、抄写、校对以及对本书的编写有过关心与帮助的有：王振宇、孙志挥、钟洛、赵红光、赵力、施江、孙小龙、黄曙萍、徐东生等同志，我的几个在学研究生徐勇、黄芳、王宇华、刘永红、周晓宇等对我的工作也支持不少，在此一并致谢。

徐宝文

1995年秋于东南大学

使用说明

一、本书以词典形式详解标准C语言及其标准库函数，涉及语言背景、基本术语、说明、语句、表达式、宏、指令、头文件、库函数、环境等各个方面，共收入词条六百六十多条。

二、编排体例上，先排以汉字开始的词条（按拼音顺序排列）、再排以拉丁字母（英文字母）开始的词条（按字母顺序排列），最后排以其它字符开始的词条（按ASCII字符集规定的顺序排列）。

三、对每一词条的解释由词条名、种类、语法（对可用语法描述的某些语言成分）、用法（对库函数）、描述、风格、实例、备注、隶属词条、相关词条、下属词条等部分组成（其中语法、用法、风格、实例、备注、隶属词条、相关词条、下属词条等部分可缺省），并以此序排列。

四、种类指该词条所属类别，包括说明、语句、表达式、库函数等种类，一个词条可以属于多个种类。

五、语法指当词条为语句、说明、表达式等时所适用的语法规则。

六、用法指当词条为库函数时对其用法的描述，包括格式、参数、返值等几个方面。

七、描述指对词条的完整说明，为一个词条的主体部分。

八、风格指在词条使用中的风格要求，如程序与语句的格式等。

九、实例部分用例子给出相应语句、表达式、说明、库函数等的用法。

十、备注是对词条的进一步解释说明，包括使用中的注意事项等。

十一、隶属词条给出当前词条在概念或结构上所属于的词条，如“return语句”隶属于“跳转语句”词条。

十二、相关词条给出与当前词条相关的词条，如“return语句”的相关词条有“goto语句”等。

十三、下属词条给出在概念或结构上下属于当前词条的词条，如“跳转语句”的下属词条有“return”语句等。

目　录

以汉字起首的词条

A～B

C

D

E～G

H

J

K～L

M～Q

R～S

T～X

Y

Z

以拉丁字母起首的词条

A～B

C

F

G～J

L

M～O

P～R

S

T～U

V～W

以特殊字符起首的词条

以汉字起首的词条

按位或表达式

种类

表达式

语法

〈按位或表达式〉::=〈按位异或表达式〉
|〈按位或表达式〉¦〈按位异或表达式〉

描述

按位或表达式是一种以按位或运算符¦为运算符、以两个整类型的子表达式作为其运算分量的二元整类型表达式，它用于对两个运算分量的位模式逐位进行“或”运算，即，仅当两个运算分量的位模式中对应位均为0时，结果的相应位才为0；否则，结果的相应位为1。

按位或表达式的求值结合规则为自左而右，其优先级比其它按位表达式都低。C中另外还有一种与按位或表达式类似的逻辑或表达式。

隶属词条

表达式，逻辑与表达式

相关词条

按位与表达式，移位表达式，逻辑或表达式，字位

下属词条

¦，按位异或表达式

按位异或表达式

种类

表达式

语法

〈按位异或表达式〉::=〈按位与表达式〉
|〈按位异或表达式〉^〈按位与表达式〉

描述

按位异或表达式是一种以按位异或运算符^为运算符、以两个整类型的子表达式作为其运算分量的二元整类型表达式，它用于对两个运算分量的位模式逐位进行“异或”运算，即，仅当两个运算分量位模式中对应位不相同时，结果的相应位才为1；否则，结果的相应位为0。

按位异或表达式的求值结合规则为自左至右，其优先级比按位与表达式低，比按位或表达式高。注意，C中没有提供与按位异或表达式对应的逻辑异或表达式。

隶属词条

表达式，按位或表达式

相关词条

移位表达式，～，字位

下属词条

^，按位与表达式

按位与表达式

种类

表达式

语法

〈按位与表达式〉::=〈相等类表达式〉
|〈按位与表达式〉&〈相等类表达式〉

描述

按位与表达式是一种以按位与运算符 & 为运算符、以两个整类型的子表达式作为其运算分量的二元整类型表达式，它用于对两个运算分量的位模式逐位进行“与”运算，即，仅当两个运算分量位模式中对应位均为 1 时，结果的相应位才为 1；否则，结果的相应位为 0。

按位与表达式求值的结合规则是自左至右。

隶属词条

表达式，按位异或表达式

相关词条

移位表达式，按位或表达式，逻辑与表达式，字位

下属词条

&，相等类表达式

按位运算符

种类

表达式

描述

按位运算符除～之外都是二元运算符(～是一元运算符)，用于对它所作用的两个运算分量进行按位运算。这些运算符中除了移位运算符<<与>>位于同一优先级上外，其余均处于不同的优先极。C 语言共提供了六个按位运算符，现将它们及其功能列出如下：

～　按位求补运算符，用于对运算分量逐位求补。

<<　左移位运算符，用于使左运算分量值左移右运算分量值所指定的位数。

>>　右移位运算符，用于使左运算分量值右移右运算分量值所指定的位数。

&　按位与运算符，用于对两个运算分量值的机器表示按位进行与运算。

^　按位异或运算符，用于对两个运算分量值的机器表示按位进行异或运算。

¦　按位或运算符，用于对两个运算分量值的机器表示按位进行或运算。

这六个按位运算符都要求各自的一个或两个运算分量都必须是整类型的(即只能是字符、枚举或整数类型)。必要时，在进行实际按位运算之前要对运算分量作整提升。对于<<与>>这两个移位运算符，运算结果的类型为左运算分量整提升后的类型；对于 &、^ 与 ¦ 这三个按位逻辑运算符，由于两个运算分量要整提升为同一类型的，故运算结果的类型即为运算分量整提升后的类型；对于按位求补运算符～，运算结果类型亦为运算分量整提升后的类型。

实例

下面的程序用于把一整数翻译成由 0 和 1 组成的二进制数，以显示其位模式。该程序中使用了多个按位运算符：

```
#include<limits.h>
#include<stdlib.h>
#include<stdio.h>

/*PRINTBIT 用于把一无符号整数翻译成二进制串*/
void
PRINTBIT(unsigned int number)
{ unsigned int i;
  unsigned int bits=sizeof(int)*CHAR_BIT;      /*bits 中存放执行环境中 int 类型对象所占字
                                                 位数*/
  unsigned int ckecker=1;      /*ckecker 用于指示由右至左的位数，初置设为1*/

  ckecker<<=bits-1;      /*将 ckecker 中非零位移到其机器表示的最左边，这样做是为了移植
                           性*/

  /*下一语句从左至右打印各位，每四位之间留一空隔*/
  for(i=1;i<=bits;i++){
    putchar((number & checker)?'1':'0';   /*打印 ckecker 非零位所对应的 number 中对应位*/
    if (i%4==0)
      putchar(' ');      /*每四位之间打印一空隔*/
```

```
    ckecker>>=1;      1 * 准备打印 number 中下一位 * /
  }
  putchar('\n');
}

main(int argc,char * argv[])
{
  int n1,n2;

  if (argc !=3){
    fprintf(stderr,"Usage: example int1 int2\n");
    exit(EXIT_FAILURE);
  }

  n1=atoi(argv[1]);
  n2=atoi(argv[2]);

  printf("bit patterns for %d and %d:\n",n1,n2);
  PRINTBIT(n1);
  PRINTBIT(n2);

  printf("~%d:\n",n1);
  PRINTBIT(~n1);

  printf("~%d:\n",n2);
  PRINTBIT(~n2);

  printf("%d & %d:\n",n1,n2);
  PRINTBIT(n1 & n2);

  printf("%d ^ %d:\n",n1,n2);
  PRINTBIT(n1 ^ n2);

  printf("%d|%d:\n",n1,n2);
  PRINTBIT(n1|n2);

  return(EXIT_SUCCESS);
  }
```

隶属词条

表达式,运算符

相关词条

一元表达式,移位表达式,按位与表达式,按位异或表达式,按位或表达式,字位

下属词条

~, <<, >>, &, ^, |

按引用传递参数

种类

函数

描述

按引用传递又叫按地址传递、按单元传递、按引用调用、按地址调用、传地址等,是C语言的一种参数与变元传递方式。具有这种传递方式的变元在传递给相应的参数时,不是把变元的值传给(赋给)相应参数,而是把变元的地址传给(赋给)参数。在函数体中,所有对参数的操作实际上就是对相应变元的操作。

与Pascal等语言不同,在C语言中,决定参数传递方式的因素不是参数说明形式,而是参数的类型。以下几种类型的参数所对应的变元只能通过按引用传递方式传递给被调用函数:

- 数组类型
- 函数类型