边用边学
C语言实用技巧

韩兴吉 王殿元 主编　李书涛 审校

计　算　机　自　学　教　材

人民邮电出版社出版

□计算机自学教材

边用边学

C语言实用技巧

韩兴吉　王殿元　主编
陈朔鹰　王殿元　编著
陈　英　徐红娟

李书涛　审校

人民邮电出版社

边用边学C语言实用技巧

韩兴吉　王殿元　主编

李书涛　审校

※　※　※　※

杜占明　责任编辑

※　※　※　※

人民邮电出版社出版发行

（北京东城区朝内南小街南竹杆胡同111号）

北京顺义振华印刷厂印刷

新华书店总店北京发行所经销

※　※　※　※

开本：787×1092 1/16　印张：14.25　字数：330千字

1996年6月第1版　1996年6月第1次印刷

印数：1—6 000册

书号：ISBN7－115－06149－1/TP·306

定价：19.50元

内 容 简 介

C语言以其自身的优点，备受广大计算机编程者的喜爱，它可以用于各种类型的编程任务，它具有汇编语言的速度，却没有pascal语言那样严格的限制。

本书编者针对目前市场上C语言的书籍比较多，但大多数书籍偏重于理论，读起来比较枯燥的毛病，在长期的C语言教学过程中，独辟新径，把C语言的基础知识和编程技巧溶于C语言程序设计实例中。

书中简介了C语言基础知识，并以Turbo C2.0开发环境为例，介绍C语言集成环境和程序设计实例，书中全部程序实例中关键的语句、算法和使用的编程技巧，都在相应的程序语句后给出清楚的注释，以帮助读者在阅读程序时十分方便地理解语句的含义，掌握C语言的实质和程序附运行结果，便于读者自学。使读者在阅读C语言编程实例中，进入C语言世界。

本书由陈朔鹰、徐红娟、陈英和王殿元编著，李书涛同志审校。参加本书编写工作的还有：李立、张余、严开、潘达、王萧、徐克、王青等几位同学，由于作者水平有限，书中错误之处难免，敬请读者批评指正。

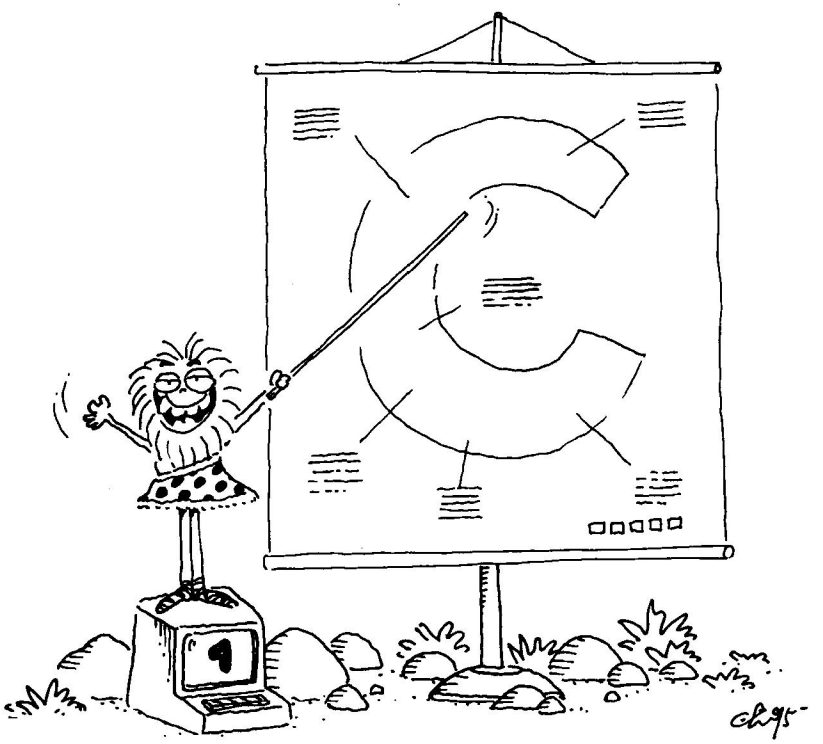
目　录

第一部分　C语言基础

第三部分　程序设计典型例题

第一部分

C 语言基础



第一步

C 语言概述

C 语言由 Dennis Ritchie 设计，并首次在一台使用 UNIX 操作系统的 DEC PDP－11 计算机上实现。C 语言是由一种早期的编程语言 BCPL 发展演变而来的，BCPL 语言目前在欧洲仍在使用。Martin Richards 改进了 BCPL 语言，从而促进了 Ken Thompson 所设计的 B 语言的发展，最终促成了 70 年代 C 语言的问世。

多年来，在第五版本的 UNIX 操作系统下写成的 C 语言一直是事实上的标准版本。BrianKernighan 和 Donnis Ritchie 在《C 程序语言》(The C Programming Language，Prentice－Hall，1978)一书中对此作了详尽的描述。随着微型计算机的日益普及，大量的 C 语言工具得以问世。在源程序水平上，这些工具程序可以说几乎是奇迹般地高度兼容。然而，由于没有统一的标准，这些工具之间也有不一致的地方。为了改变这种情况，ANSI 于 1983 年专门成立了一个委员会，为 C 语言制定了 ANSI 标准。Turbo C 正是完全按照 ANSI 的 C 语言标准设计的，它是一种快速、高效的编译程序。Turbo C 不仅提供一个集成开发环境，同时还按传统方式提供了一个命令行编译程序版本，以满足不同用户的需要。本书以 Turbo C 为样本，讲述 C 语言。

C 语言通常被称为结构语言，因为其结构类似于 ALGOL，Pascal 和 Modula－2。从技术上说，模块结构语言允许在子程序或函数中再定义子程序或函数。这样，按照作用域规则，“全程”和“局部”的概念就扩展了，这一规则决定了一个变量或子程序的“可见性”。由于 C 语言不允许在函数中建立函数，所以它不是真正的模块结构语言。

结构语言的一个显著特点是代码及数据的分隔化。分隔化是指一种语言的分段隔离能力，即程序的各个部分除了必要的信息交流外彼此互不影响，相互隔离。实现分隔化的常用方法是使用若干子程序，在子程序中分别定义各自的局部变量。由于使用局部变量，编程者能保证其子程序运行时不会对程序的其它部分产生副作用。由于 C 语言具有结构语言的这些特点，程序之间很容易实现程序段的共享。

结构语言使编程者有多种选择的可能性，它直接支持若干种循环结构，例如 while，do－while 和 for。在结构语言中，goto 语句常被禁止使用或者建议尽量避免使用，它不是象 BASIC 和 FORTRAN 语言中那样，作为一种常用的程序控制语句形式。结构语言允许编程者使用缩进书写形式编程。而不必象传统的 FORTRAN 语言那样有严格的书写区域限制。

结构语言趋于现代编程风格，而非结构语言较为陈旧。今天人们普遍认为结构语言层次清晰，比非结构语言更易于使用和维护。这是 C 语言备受欢迎的原因。

C 语言的主要结构成分是函数—C 语言独立的子程序。在 C 语言中，函数是基本的结构模块，所有的程序活动内容都包含在其中。函数可以在程序中被定义完成独立的任务，独立地编译成目标代码，这样可以实现程序的模块化。函数建立后，可以在各种情况下正确运行，而不必担心它对程序的其它部分产生不良影响。可以建立独立的函数这一点在大型软件工程中是非常关键的，因为只有这样才能避免不同编程者的程序由于偶然因素而互相干扰。

学习过 COBOL 或 BASIC 语言的程序设计人员就会知道，COBOL 是为使非专业编程者能阅读，甚至能读懂程序而设计的。而 BASIC 基本上是为非专业编程者设计的，它可以解决一些相对简单的问题。

相比之下，C 语言几乎可以说是独树一帜，它由一些具有实际系统开发经验的专业编程者设计，并在实际使用中得到了不断完善和考验，C 语言所能提供的恰好是编程者所要的：很少限制、很少缺陷、模块结构、彼此独立的函数和一些十分紧凑的关键字。使用 C 语言能达到接近汇编语言的效率，又兼有 ALGOL 或 Modula—2 语言的结构形式。毫无疑问，C 语言是第一流的专业编程者最欢迎的语言。

C 语言广泛用于各种类型的编程任务，最主要原因也许是编程者喜欢用它！它具有汇编语言的速度，FORTH 语言的可扩展性，但又很少有 Pascal 或 Modula—2 语言那样的严格限制。每一个 C 语言编程者都可以根据需要建立和维护属于自己的用户函数库，用户函数库可以用于许多不同的程序中。由于 C 语言允许，确切地说是提倡使用分块编译，因此使编程者易于管理大型软件工程，大大减少了重复工作。

本书为讲述方便，使用下列术语，读者应熟悉其含义。

源程序： 用户可读的程序文本文件，通常认为就是程序本身。源程序需要进行编译。

目标代码： 是由源程序翻译而成的机器码，计算机能读并能直接运行。目标代码还需要进行连接。

连接程序： 将各自分别编译后的程序连接为一个运行程序的程序。它将 C 语言的标准库函数与用户所编的程序联合在一起。连接后的程序是一个可运行程序。

库： 包含标准函数的文件，这些函数可在编写程序时直接调用。包括所有的输入/输出(I/O)函数，以及其它有用的函数。

编译状态： 在程序编译过程中所出现的事件，在编译状态通常发生的事件是语法错误。

运行状态： 在程序运行过程中所发生的事件。

C 语言中使用的关键字如下：

由 ANSI 标准推荐的 32 个关键字

auto	double	int	struct
break	else	long	switch
case	enum	register	typedef
char	extern	return	union
const	float	short	unsigned
continue	for	signed	void
default	goto	sizeof	volatile
do	if	static	while

Turbo C 扩展的关键字

asm	_cs	_ds	_es
_ss	cdecl	far	huge
interrupt	near	pascal	

C 语言的关键字，加上语法规则，就形成了 C 语言的程序，注意，C 语言的关键字都是小写的。按照 C 语言的书写习惯，变量、函数名等标识符一般也都采用小写形式。

第二步

变量、常量、运算符和表达式

变量、常量通过运算符组合在一起构成表达式。它们是组成C语言的基本元素。

1. 标识符名

C语言可以定义各种标识符作为变量、函数、标号及用户定义对象的名称。标识符的有效长度为1至32个字符。标识符的第一个字符必须是字母或下划线，后续字符可以是字母、数字或下划线。以下列举了几个正确的和不正确的标识符名称：

正确	不正确
count	1count
test23	hi! there
high __ balance	high..balance

在C语言中，大小写字母是有区别的，所以，count、Count 和 COUNT 是三个不同的标识符。标识符不能和C的关键字相同，也不应该和已定义的函数名或C的库函数名相同。

2. 数据类型

在C语言中有五种基本数据类型：字符型(char)、整型(int)、实型(float)、双精度实型(double)和无值型(void)。上述数据类型的长度如表2—1所示。

表2—1 C语言基本数据类型的长度和值域

类型	字节长度	值域
char	8	0至255
int	16	—32678至32767
float	32	3.4E—38至3.4E+38
double	64	1.7E—308至1.7E+308
void	0	无

字符型(char)变量用于存储 ASCII 码字符，也可存储 8 位二进制数。整型(int)变量用于存储整型量。实型(float)和双精度实型(double)变量用于存储实数(实数具有整数和小数二部分)。

无值型(void)变量有两个用途。第一个用途是明确地表示一个函数不返回任何值；第二个用途是产生同一类型的指针。

3. 类型修饰符

除了无值类型外，基本数据类型可以带有各种修饰前缀。修饰符用于明确基本数据类型的含义，以准确地适应不同情况下的要求。类型修饰符种类如下：

signed	有符号
unsigned	无符号
long	长
short	短

表 2—2　C 语言基本类型及其修饰符的所有组合

类型	字节长度	值　域
char	8	—128 至 127
unsigned chal	8	0 至 255
signed char	8	—128 至 127
int	16	—32768 至 32767
unsigned int	16	0 至 65535
signed int	16	—32768 至 32767
short int	16	—32768 至 32767
unsigned short int	16	0 至 65535
signed short int	16	—32768 至 32767
long int	32	—2147483648 至 2147483647
signed long int	32	—2147483648 至 2147483647
float	32	3.4E—38 至 3.4E+38
double	64	1.7E—308 至 1.7E+308
long double	64	1.7E—308 至 1.7E+308

修饰符 signed，unsigned，long 和 short 可以用于字符型和整型的基本类型。此外，long 还可用于双精度实型量。表 2—2 依据 ANSI 标准列举了全部允许的数据类型及它们的长度和取值范围。

整型前面的 signed 可以省略，因为整型定义本身规定就是有符号的。

有符号(signed)和无符号(unsigned)的整型量的区别在于它们的最高位的定义不同。如果定义的是有符号的整型(signed int)，C 编译程序所产生的代码就设定整型数的最高

位为符号位。

4. 访问修饰符

C语言有两个用于控制访问和修改变量的方式的修饰符，它们分别是常量(const)和易变量(volatile)。

const在程序运行过程中始终保持不变。例如：

```
const int a;
```

将产生整型变量a，其值不能被程序所修改，但可以在其它类型的表达式中使用。const型量可以在其初始化时直接被赋值，或通过某些硬件的方法赋值。

修饰符volatile用于提醒编译程序，该变量的值可以通过程序中未明确定义的方法来改变。例如，一个全程变量的地址可以传给操作系统的时钟例行程序，从而该变量可以用于存储系统的实时钟值。在这种情况下，变量的内容在程序中没有明确的赋值语句对它赋值时，也会发生改变。这一点是很重要的，因为在假定表达式内变量内容不变的前提下，Turbo C自动优化某些表达式。此外，在编译过程中，有的优化处理会改变表达式的求值顺序。修饰符volatile可防止上述情况发生。

5. 变量的说明

所有的变量在使用前都必须加以说明。说明的一般形式如下：

```
类型　变量表;
```

这里，类型(type)必须是C语言的有效数据类型。变量表(variable—list)可以是一个或多个标识符名，中间用逗号分隔。例如：

```
int i,j,l;
short int si;
unsigncd int ui;
double balance,profit,los;
```

请注意：在C语言中，变量名与它们的类型毫无关系。

6. 局部变量

在函数内部说明的变量称为局部变量。局部变量也可以只在复合语句中定义和使用，换句话说，在复合语句以外并不"知道"有该变量存在。请记住，复合语句以左花括号开始，至右花括号结束。

掌握局部变量的最重要的一个问题是，仅仅当说明它们的复合语句被执行时，局部变量才存在。也就是说，当程序运行到这个复合语句时，它们才产生，退出时它们则消失。

由于局部变量是在一个函数或复合语句中建立并消失的，所以当函数或复合语句运行后它们所定义的局部变量内容就丢失了。这一点在函数调用时尤其值得注意。当函数被调用时，其中的局部变量就建立，而当函数返回时，这些变量就不再存在了。这就意味着，在函数调用中局部变量无法返回。

除非特别加以说明，局部变量是存储在堆栈内的。而堆栈是在内存中的一个动态变化