面向21世纪高校统编教材

C程序设计教程

C chengxu sheji jiaocheng

主 编 王柏盛

中国矿业大学出版社

前　言

C 语言是在国内外得到广泛使用的 DOS 操作系统下的现代化结构程序设计语言，是在 Windows 取代 DOS 操作系统后得以保留的 DOS 操作系统下的少数优秀程序设计语言之一。C 语言功能丰富，表达力强，使用方便灵活，目标程序效率高，可移植性好，既具有结构化高级程序设计语言的优点，又具有低级语言的绝大部分功能，它可以取代汇编语言用来编写系统软件、工具软件和控制软件。

现在，C 语言已不仅为计算机专业工作者所使用，而且也为广大计算机应用人员所喜爱和使用，许多人已经用它编写出很多优秀的应用软件。高等院校在非计算机专业普遍开设了 C 语言程序设计课程，大部分学生都选择 C 语言参加全国计算机二、三、四级等级考试。

目前 C 语言方面的书籍很多，根据我们讲授 C 语言所选用的几种教材来看，当前的一些 C 语言教材存在不少问题，主要是：

(1) 很多学生，包括学习过 C 语言的研究生都反映，学了 C 语言用不起来，还不如 BASIC 语言、FORTRAN 语言用起来得心应手，他们普遍感觉不到 C 语言的那些优点。我们认为，这实质上还是教材的问题，因为很多 C 语言教材只是偏重于讲解 C 语言的语法和程序设计方法，这和讲解其他高级语言几乎没有多少区别，而对于 C 语言丰富的库函数，如字符屏幕函数、图形函数、系统调用函数以及 C 语言与汇编语言的接口技术等，却很少涉及，而这些恰恰是体现 C 语言功能强大、使用方便灵活的众多优点之处。如果学生不掌握这些实用编程技术和了解使用这些强大功能，如何能体会到 C 语言的优点而喜爱 C 语言呢？

(2) 运算符的丰富和指针的灵活使用是 C 语言的一大特色，但是一些教材想讲清楚这些问题，不惜花费了大量篇幅，结果却适得其反，使学生越学越糊涂。实际上，指针的实质就是地址，是指向数据元素的地址，而一种数据元素在内存中占用一定字节数的存储单元，指针只能指向其存储单元的首地址，而不能指向其中的某个字节。因此，讲清楚了这些，各种数据结构的指针运算和使用也就容易掌握了。

(3) 还有一些问题，例如数组或变量的初始化问题，一些教材过分强调只有外部变量或静态变量才能在定义时赋初值。这样讲解不是把学生引入歧途，就是使学生糊涂。难道在一个函数中定义一个大数组时不能同时赋初值而非得在程序中用大量的赋值语句去赋值吗？难道非得定义静态数组或外部数组吗？在函数内部定义数组的同时赋初值是完全允许的，上机实验不难证明这一点。不一定非要强调这些概念问题，只要讲清数据的存储属性，系统给全局变量和静态变量分配固定的存储区，给局部变量分配动态的存储区，就完全可以了。

(4) 既然学习过一门高级语言是学习 C 语言的前提，那么 C 语言教材大可不必多层次地、反复地讲解那些很容易理解、掌握的基本知识。例如，运算符不一定非得分到各有关章节去讲解，数组也不需要放到循环后才讲，这种编排方法增加了篇幅，增加了学时，不适应当前减少学时、增开新学科、进行教学改革的需要。

(5) 不少教材不加检验地去照搬别的书中的内容，使得一些错误得不到纠正。例如，讲条件表达式在什么情况下可以取代条件语句时，许多教材都说：条件表达式不能取代一般的 if 语句，只有在 if 语句中内嵌的语句为赋值语句(且两个分支都给同一个变量赋值)时才能代替 if 语句。因此，只有类似

```
if (a>b) max=a; else max=b;
```

这种形式下才能用条件表达式语句去取代条件语句

```
max=a>b?a:b;
```

而

```
if (a>b) printf ("%d",a); else printf ("%d",b);
```

是不可以用

```
a>b? printf ("%d",a): printf ("%d",b);
```

取代的。

实际上这种说法是没有根据的，上机实验不难证明这一点。其实，条件表达式的定义本来是：

```
<表达式 1>?<表达式 2>:<表达式 3>
```

只要在

```
if (<表达式>) <语句 1> else <语句 2>
```

中的<语句 1>和<语句 2>都是表达式语句时就可以用条件表达式

```
<表达式 1> ?<表达式 2>:<表达式 3>
```

来代替 if 语句。既使表达式 1 为函数调用或无 else 子句也不是不可以的。例如程序

```
#include "stdio.h"
main()
 {
   int a=3,b=4;
   printf("%d\n",getch()-48?a:b);
   getch();
 }
```

仅当从键盘输入 0 时才输出 b 的值 4，否则输出 a 的值 3。

又如程序

```
main()
 {
  int a,b,c;
  printf("a,b=");
  scanf("%d,%d",&a,&b);
   /*用条件表达式取代无 else 子句的条件语句 if(a<b){c=a;a=b;b=c;}语句*/
  a<b?(c=a,a=b,b=c):0;  printf("%d,%d\n",a,b);
 }
```

无 else 子句，可以假设一个表达式作为条件表达式中的表达式 2 或表达式 3。

(6) 我们在教学和科研中发现 Turbo C 在表达式中存在严重的不一致性，这实在是一个遗憾。例如：

```
main()
```

```
{
 int i=1,j;
 j=i++*i++;printf("%d  ",j);
 i=1;printf("%d  ",i++*i++);
 i=1;printf("%d  ",j=i++*i++);
 i=1;printf("%d  ",i+++i+++i++);
 i=1; printf("%d\n", j=i+++i+++i++);
 i=1;j=i+++i+++i++;printf("%d  ",j);
}
```

程序运行结果是：1 2 2 6 6 3

可见 printf()函数中的表达式和赋值表达式存在严重的不一致性。

我们编写本书的目的，就是为了消除目前同类书中存在的上述诸多问题，并力求把本书编成一本概念清楚、阐述明确、结构合理、实用性强、重点突出的集教材、资料为一体的 C 程序设计书。

本书共分十章。第一章 C 语言概述，第二章常量、变量、运算符和表达式，第三章程序控制语句，第四章函数，第五章指针，第六章结构、联合、枚举和定义类型，第七章编译预处理命令，第八章文件，第九章字符屏幕和图形函数，第十章实用编程技术。

讲解本书约需 64～70 学时，其中上机实验 20 学时。不讲第九章和第十章需要 50 学时，其中包括 16 学时上机实验。有些章节（如第三章）有大量的例题，其中不少例题属于阅读、理解、加深题，学时不充分时可以跳过不讲。

由于计算机语言是一个有机的整体，各章节之间互相联系、互相渗透。例如，运算符为各章所用到；数组、函数、指针、结构于各章之间互相影响，指针可用于数组、函数、结构，数组也可用于指针、函数、结构；仅指针而言，就有指针数组、数组指针、指针的指针，指针函数、函数指针，指针结构、结构指针等。因此，没有必要完全由浅入深地编写 C 语言教材，因为这样做一方面会大大增加教材的篇幅和内容的重复，另一方面会增加教学时数。所以读者在阅读本书时，应该在学习后面章节的同时还需要经常复习前面的内容，以加深理解。

本书的第二、十章由王柏盛编写，第五章由孙建英编写，第四、七章由郭红编写，第一、八章由申艳光编写，第九章由王文鹏编写，第三、六章和全部附录由李万庆、余新宁、刘云编写。全书由王柏盛统稿、整理。

本书在编写时参考并引用了《Turbo C 使用大全》(徐金梧等编译)、《C 程序设计》(谭浩强编著)、《Tubro C 简明教程》(赵海庭编)、《Tubro C 2.0 实用高级编程技巧》(王军政编)等书中的一些内容和例题；同时，本书在编写、出版过程中得到了院系领导和有关同志的有力支持和帮助，在此一并深致谢意。

由于作者的水平有限，经验不足，时间仓促，书中难免存在许多不足之处，恳请广大读者和同行批评指正。

王柏盛

2002 年 6 月

目　录

第一章　C 语言概述

1.1　C 语言的起源

C 语言是 1972 年由美国人丹尼斯·里奇(Dennis Ritchie)设计发明的，并首次在 UNIX 操作系统的 DECPDP-11 计算机上使用。它是由早期的编程语言 BCPL(Basic Combined Programming Language)发展演变而来的。在 1970 年，美国贝尔实验室的肯·汤普森(Ken Tompson)以 BCPL 语言为基础设计出较先进且接近硬件的 B 语言，B 语言的出现促进了 70 年代 C 语言的问世。

随着微型计算机的日益普及，C 语言出现了多种版本。为了改变这些版本不一致的情况，1983 年，美国国家标准化协会为 C 语言制定了一套 ANSI 标准。Turbo C 完全是按照 ANSI 的 C 语言标准实施的，是一种快速、高效的编译程序。Turbo C 不仅提供了一个集成开发环境，而且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。

1.2　C 语言的特点

1. 结构化语言

结构化语言的一个显著特点是代码和数据的分隔化，即程序的各部分除了必要的信息交流外彼此互不影响、相互隔离。C 语言的主要结构成分是函数，函数是 C 语言的基本结构模块，所有的程序活动内容都包含其中。函数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。C 语言中，另一个实现程序结构化和分离化的方法是使用复合语句。复合语句是作为一个语句对待的，且是具有逻辑联系的程序语句的组合，它是一个逻辑单元。严格地说，C 语言并不是一种真正的结构化语言，因为它不允许在子程序或函数中再定义子程序或函数，但由于它的结构类似于 ALGOL、Pascal 和 Modula-2，通常还是把 C 语言称为结构化语言。因为，C 语言有现代化语言的各种数据结构，它的数据类型有整型、实型、字符型、数组、结构、联合及指针和无值型等，能用来实现各种复杂的数据结构的运算，如链表、树、图、堆栈等。另外，C 语言具有结构化的控制语句，如 if...else，while，do...while，for，switch 等语句。C 语言用函数作为程序模块以实现程序的模块化，是结构化的编程语言，符合现代化编程风格。

2. 简洁、紧凑、灵活

Turbo C 共有 43 个关键字、9 种控制语句，程序书写自由，主要以小写字母表示，压缩了一切不必要的成分。C 语言语法限制不太严格，程序设计自由度大，如对数组边界不作检查，整型、字符型、逻辑型数据都可以通用。

3. 运算符丰富

C 语言共有 44 种运算符，把括号、赋值、强制类型转换等都作为运算符处理，从而使 C 语言的运算类型极其丰富，表达式类型多样化，灵活使用各种运算符可以实现在其他高级语言中难以实现的运算。

4. 中级语言

C 语言把高级语言的基本结构与低级语言的实用性结合起来。它可以像汇编语言一样对位、字节和地址进行操作，这三者是计算机最基本的工作单元，可以实现汇编语言的绝大部分功能。

5. 可移植性好

用 C 语言编写的程序可移植性好，生成的目标代码质量高，程序执行速度快。

6. 功能强大

C 语言有丰富的库函数、强大的图形功能及预处理能力，与其他语言如汇编语言、Pascal 语言、数据库语言等容易接口，C 语言程序中还可以直接调用 DOS 命令。因此，C 语言适合于编写各种系统软件、工具软件等大型软件。目前，在工业计算机控制系统开发中，越来越多的编程人员使用 C 语言编写控制软件。

7. 编译语言

用 C 语言编写的源程序必须经过编译后才能运行。

1.3 C 语言的词法

1.3.1 字符集

C 语言的字符集包括：大小写英文字母、数字、下划线“_”及“ + - * / = , ; ? ! " % . & ^ # ' < > | ( ) { } [ ] ~ \ ”等。

1.3.2 关键字

关键字又称为保留字，C 语言编译系统对关键字赋有专门的含义。Turbo C 共有 43 个关键字：

描述存储属性的有：auto，extern，static，register；

描述数据类型的有：char，int，float，double，void，struct，union，enum，long，short，signed，unsigned；

描述语句的有：goto，if，else，switch，case，default，break，for，do，while，continue，return；

描述访问方式的有：const(常量修饰符)，volatile(易变量修饰符)；

描述编译状态的有：sizeof；

描述数据类型定义的有：typedef；

Turbo C 扩展的关键字有：asm，_cs，_ds，_es，_ss，cdecl，pascal，far，near，huge，interrupt。

1.3.3 标识符

C 语句中以标识符命名程序中的对象名，如函数、变量、符号常量、数组、结构、联合、指针、数据类型、存储属性、标号及宏等。

标识符是以字母、数字和下划线组成的，但第一个字符必须是字母或下划线。Turbo C 规定标识符的有效字符长度为 1～32 个字符。在C语言中，字母大小写是有区别的，COUNT、Count 和 count 为三个不同的标识符，习惯上符号常量、宏名等用大写字母，变量、函数名等用小写字母，系统变量由下划线开头。

标识符不能和 Turbo C 中的关键字相同，也不允许和已定义的函数名或 Turbo C 库函数中的函数名相同。main 虽然不是 ANSI 标准的关键字，但各种C语言版本都把它作为主函数名，我们也不应把它作为标识符。C 语言源程序名不属于 C 语句，属于操作系统，其命名要求符合操作系统文件名的规定，其扩展名通常为.C。

标识符的选择应该做到“见名知义”、“常用取简”、“专用取繁”。例如：count，name，year，month，student_number，display，screen_format 等，使人一目了然，以增强程序的可读性。例如：

正确的标识符名	不正确的标识符名	注　释
test_123	100	数字开头
_label_100	interger!	包含!字符
rectangle_area	chinese word	包含空格字符
circle_radius_r	for	是关键字
students	printf	与库函数中的输出函数同名

1.4 C 程序的组成和结构特点

C 语言的 43 个关键字加上语法规则，就构成了 C 编程语言。所有的 C 语言关键字都是小写字母。

C 语言程序包含一个或多个函数。惟一不可缺少的函数是 main()函数，它是程序开始运行时第一个被调用的函数。在一个好的 C 语言程序中，main()函数提纲性地列出程序所要做的事情，而这一提纲由一系列函数调用所组成。main 虽然不是 C 语言的组成部分，但还是应该把它视为关键字来对待，不要把它用作变量名，否则有可能使编译程序产生混乱。

1.4.1 程序举例

【例 1.1】

```
main()
{
 printf("This is a C Program.\n");
}
```

本程序的作用是输出以下一行信息：

```
This is a C Program.
```

其中，main 表示“主函数”，每一个 C 程序都必须有一个主函数。函数体由一对大括号“{}”括起来。本例中，主函数内只有一条输出语句，printf 是 C 语言中的输出函数，将双引号中的字符串原样输出。“\n”是换行符，即在输出信息“This is a C Program.”后回车换行。语句的最后有一个分号，是语句的组成部分，Turbo C 程序的任何语句都是以分号结束的。

【例 1.2】

```
main()                              /*求两数之和*/
 {
  int a,b,sum;                      /*定义变量*/
  a=123;
  b=456;
  sum=a+b;
  printf("sum is %d\n",sum);
 }
```

本程序的作用是求两个整数 a 和 b 之和 sum。“/*...*/”表示注释部分，注释只是为了阅读而加的，对编译和运行不起作用，注释可以加在程序中任何位置。第二行是变量定义部分，说明 a、b 和 sum 是整型变量(int)。第三行是两条赋值语句，使 a 和 b 的值分别是 123 和 456。第四行计算 a+b 的和并赋给 sum。第五行的“%d”表示按十进制整数格式输出 sum 的值。本例编译运行后输出的信息为：

```
sum is 579
```

【例 1.3】

```
main()                          /*主函数*/
 {
  int a,b,c;                    /*定义变量*/
  scanf("%d,%d",&a,&b);         /*输入 a 和 b 的值*/
  c=max(a,b);                   /*调用函数 max,将得到的值赋给 c*/
  printf("max=%d",c);           /*输出 c 的值*/
 }
int max(x,y)                    /*定义 max 函数,函数值为整型,x,y 为形式参数*/
int x,y;                        /*定义形参 x,y 为整型*/
  {
   int z;                       /*定义 max 函数中的 z 为整型变量*/
   if(x>y)z=x;else z=y;         /*求 x,y 的最大值并赋给 z*/
   return(z);                   /*从子函数返回调用处,z 为返回的函数值*/
  }
```

本程序包含两个函数：主调函数 main 和被调函数 max。max 的作用是将 x 和 y 中的较大值赋给 z。return 语句将 z 的值返回给主调函数 main，返回值是通过函数名 max 带回到 main()的调用处。main()函数中的 scanf 是“输入函数”的名字，其作用是通

过键盘输入 a 和 b 的值。&a 和&b 中的“&”的含义是“取变量的地址”，“%d,%d”是指按十进制整数形式输入。main()函数中的第四行为调用 max()函数，在调用时将实际参数 a 和 b 的值分别传送给 max()函数中的形式参数 x 和 y，执行 max 函数后得到一个返回值赋给主函数中的变量 c。本程序编译运行结果为：

```
8,5<Enter>      输入 8 和 5 给 a 和 b,<Enter>为回车键
max=8           输出 c 的值
```

本例中用到了函数调用、实参和形参等概念，这将在第四章中详细介绍。

1.4.2 结构特点

通过上述几个简单的例子可以看出以下几点：

1. C 程序是由函数构成的。函数是 C 程序的基本单位，一个 C 程序中必须有一个名为 main 的主函数和若干个子函数(可以没有)。被调函数可以是系统提供的库函数，如 printf 和 scanf 函数，也可以是用户自己编写的函数，如例 1.3 中的 max 函数。子函数相当于其他语言中的子程序。C 语言用函数实现其特定功能，可以说 C 语言是函数式的语言，程序的全部工作都是由函数来完成的。C 语言函数库十分丰富，Turbo C 提供了 300 多个库函数，使 C 语言容易实现程序模块化。

当一个程序较大，包括很多函数时，可以将其分成若干个文件，每个文件包括几个函数。

2. 一个 C 函数由两部分组成：(1) 函数说明部分，包括函数名、函数类型、函数属性、函数参数和参数类型，函数名后必须跟一对圆括号，但可以没有函数参数。(2) 函数体，即大括号内的部分，如果函数内有多对大括号，则最外层的一对大括号为函数体的范围。函数体一般包括变量定义和执行部分，也可以没有变量定义部分，甚至连执行部分也没有，这是一个空函数，它什么也不干。

3. 一个 C 程序总是从 main 函数开始执行，与 main 函数在程序中的位置无关，即它可以在其他函数的前面、后面或中间。

4. C 程序书写格式自由，一行可以写一条语句，也可以写多条语句，一条语句也可以分成几行写，没有行号，也不像 FORTRAN 和 COBOL 那样严格规定书写格式。

5. 每条语句以分号结束，分号是语句的组成部分，即使是最后一条语句，分号也不可缺少。

6. C 语言没有输入、输出语句，输入、输出是由库函数完成的，即 C 语言对输入、输出实行“函数化”。

7. 可以用“/*...*/”对程序中的任何部分作注释，以增强程序的可读性，“/*”和“*/”必须成对出现，且 / 与 * 之间不允许出现空格。

1.5 C源程序的编辑、编译、链接和运行

1.5.1 C源程序的编辑

C源程序可以用任何编辑程序来输入，如WPS、EDLIN，或在C的集成开发环境下编辑。C源程序通常以.C作为扩展名。

1.5.2 C源程序的编译和链接

由于C语言是编译语言，编辑的C源程序需要编译，编译后生成.obj文件。大部分较短的C程序完全可以包含在一个源文件里。然而，随着程序变长，编译时间也就长了，所以Turbo C允许把程序分割成许多块，分别装进不同文件里，每个文件可以单独编译。分块编译的优点在于再次编译时只需编译修改过的文件，这对于大型程序的调试所节省的时间是很可观的。

编译的.obj文件需要经过链接，才能生成一个可执行的代码文件.exe。这是因为编译的.obj文件的机器码指令的内存地址并没有绝对确定，而只是保存了一个偏移量；另外，还要实现对分块编译得到的.obj文件以及库函数的链接，因为库函数也是以浮动地址的形式保存的。

1.5.3 Turbo C的内存映射

编译后的Turbo C程序产生并使用四个不同的逻辑内存区域，它们分别有其特定的功能。第一个区域用来存储程序代码；第二个区域用来存储程序中的全局变量，称为静态存储区；其余的两个区域用作堆和堆栈，称为动态存储区。堆栈在程序运行中有很多用途，用于保存函数的返回地址、函数的变元及局部变量；堆栈还用来保存CPU的当前状态。堆是一个自由存储区域，程序通过Turbo C的动态分配函数来使用它，用于链表和树等的存储。

Turbo C程序的内存映射如图1.1所示。

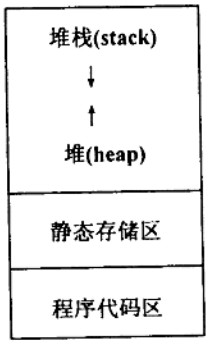

图1.1 Turbo C程序的内存映射

1.5.4 C源程序的调试过程

整个C源程序的编辑、编译、链接、运行、调试的全过程如图1.2所示。

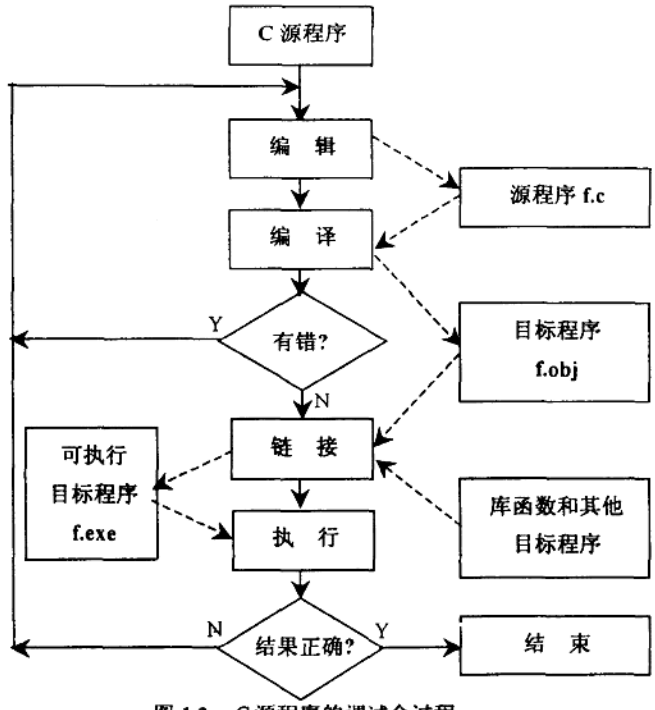

图 1.2 C源程序的调试全过程

1.6 标准输入、输出函数

1.6.1 格式化输入、输出函数

Turbo C 标准库提供了两个格式化输入、输出函数 scanf()和 printf()， 这两个函数可以在标准输入、输出设备上以各种不同的格式读写数据。printf()函数用来向标准输出设备(屏幕)写数据；scanf()函数用来从标准输入设备(键盘)上读数据。下面介绍这两个函数的用法。

1. printf()函数

(1) 函数功能

printf()函数是格式化输出函数，一般用于向标准输出设备按规定格式输出信息。在编写程序时经常会用到此函数。printf()函数的调用格式为：

printf("<格式字符串>",<参量表>);

格式字符串包括两类对象：一类是普通字符，这些字符在输出时将按原样输出；另一类是规定参数输出格式的格式字符，每一参数的输出格式都是以“%”开始，后面为一个格