



# 实用C语言编程技术

● 韩仲清

唐先余

编译

● 严素华

唐常杰

● 四川大学出版社

TTP12  
H26/1

# 实用C语言编程技术

韩仲清 唐先余 编译  
严素华 唐常杰



四川大学出版社

1992年·成都

0022787

(川)新登字014号

责任编辑：杨守智  
封面设计：冯先洁  
版式设计：杨守智

## 实用 C 语言编程技术

韩仲清 唐先余 编译  
严素华 唐常杰

---

四川大学出版社出版发行 (成都市望江路29号)  
四川省新华书店经销 成都教育印刷厂印刷  
787×1092毫米 32开本 13印张 250千字  
1991年7月第1版 1992年7月第2次印刷  
印数：5001—15000册

---

ISBN 7-5614-0390-9/TP·9 定价：4.20元

## 重 印 说 明

本书自出版以来，深受读者欢迎，很快销售一空。在这重印之际，我们根据教学的情况，对原文给出了适当的评注，这可以帮助读者更快地理解和应用C语言。

我们在本科生、进修生教学中用本书作教材时，曾采用如下安排：每周上课 2 学时，有辅导的实习 2 学时，自己上机 2 学时。第1—3周略讲或完全自学前三章；第四周讲实例 10.1—10.2节；5—8周讲完四—七章；然后，以实例为主线，组织专题，讲授加课堂讨论。实例中需用什么，再复习、补充什么。考核时按照实习40%，小论文10%，期末考试（开卷或闭卷）50%加权。收到较好效果。用书单位宜根据学生基础及上机条件，制定自己的实习教学计划。

本书有配套的软盘，录有主要程序，特别是第十章的程序。欢迎读者邮购，每套80元。地址：成都四川大学计算机系（610064）。

唐常杰  
韩仲清

1992.4.1

# 前 言

C程序设计语言自1978年发表以来，经历了十多年实际应用的检验，越来越走红，已经成为目前最为流行的计算机语言之一。为什么会如此？这是因为C语言有其它众多语言不可比拟的优越性：

## 1. 可移植性好。

由于C语言并不过份的依赖于计算机硬件环境，一个C语言程序，不需改动或改动很少就可以在不同的机器上运行。因而，目前广泛使用的大、中、小、微型机上都可以使用C语言。从这点上来说，C语言的生命力相当旺盛。

## 2. 有多种操作系统的支持

不但闻名于世的UNIX操作系统支持C语言，而且广为流行的PC-DOS和MS-DOS也支持C语言。这就使得C语言不仅有一个优越的硬件环境，而且还有一个良好的软件环境。

## 3. 特别适合系统软件开发

C语言除了具有通用程序设计语言的功能外，还有其专用功能，这就是特别适用于编写系统软件。因而有人把C语言说成是“高级语言中的低级语言”，或“低级语言中的高级语言”。很多著名的软件，如：UNIX系统，DBASE III

数据库管理系统等均是用C语言写成的。

鉴于以上优点，C语言展示了其广泛的使用价值和远大的发展前景，已经引起了计算机界的极大重视，其显著标志是：

(1) 国内外的计算机部门和公司，特别是软件开发公司，招聘和接收计算机技术人员时，十分注意使用C语言编程的能力。目前已有一种倾向，就是把是否懂C语言作为是否懂计算机软件的标志。可以说，不懂C语言的程序员将是不受欢迎的程序员。

(2) 国家劳动人事部在全国计算机应用软件人员任职资格（水平）考试大纲中规定，从1992年起，把C语言列为正式的选考语言（实际上，90年考题已有C语言内容）。

(3) 各大专院校纷纷将C语言由原来的选修课定为必修课，而且增加了学时和上机实习时间。各培训机构也纷纷将C语言列为培训课程，计算机专业的本科生和硕士研究生，越来越重视C语言基础训练。

从以上可以看出人们对C语言是何等的重视。在这种形势下，我们翻译了由美国贝尔实验室著名的计算机专家，C语言的发明者Brian W. Kernighan和Dennis M. Ritchie于1988年最新出版的《The C Programming Language》一书。为使国内读者能够在目前流行的PC机或长城0520CH及其兼容机上运行C语言，我们增补了第九章C语言的微机版本，介绍如何在PC机上运行C语言程序，以及第十章Turbo-C程序实例及编程经验。

学习任何一种计算机语言都需要上机作大量实习。我们建议读者读完第一章后，就粗读第九章和第十章，并选定一

种编译器（推荐用Turbo-C），一边学习，一边把第十章中的实例录入、编译、改进，通过实践，学好C语言。

在编译本书的过程中，我们力求准确。当然，书中仍可能还有错漏，请读者指正。

第一至第六章由韩仲清、魏幼勋、刘元社、廖兴祥、严素华翻译，第七至九章和附录由唐先余翻译，全书由韩仲清统稿，唐常杰撰写了第十章并审校全书。

编译者

1990.10.15日于四川大学

# 目 录

## 第一章 C语言概述

§ 1.1 起步 .....	( 1 )
§ 1.2 变量和算术表达式 .....	( 5 )
§ 1.3 FOR语句 .....	( 12 )
§ 1.4 符号常数 .....	( 10 )
§ 1.5 字符的输入和输出 .....	( 15 )
1.5.1 文件的复制 .....	( 16 )
1.5.2 字符计数 .....	( 18 )
1.5.3 行计数 .....	( 20 )
1.5.4 单词计数 .....	( 22 )
§ 1.6 数组 .....	( 25 )
§ 1.7 函数 .....	( 28 )
§ 1.8 参数——传值调用 .....	( 33 )
§ 1.9 字符数组 .....	( 34 )
§ 1.10 外部变量与作用域 .....	( 39 )

## 第二章 类型、运算符和表达式

§ 2.1 变量名 .....	( 45 )
§ 2.2 数据类型及大小 .....	( 46 )
§ 2.3 常数 .....	( 47 )
§ 2.4 说明 .....	( 51 )



§ 2.5	算术运算符.....	(53)
§ 2.6	关系运算符和逻辑运算符.....	(54)
§ 2.7	类型转换.....	(66)
§ 2.8	增 1 和减 1 运算符.....	(61)
§ 2.9	逐位运算符.....	(64)
§ 2.10	赋值运算符和表达式.....	(66)
§ 2.11	条件表达式.....	(98)
§ 2.12	优先级与运算顺序.....	(70)

### 第三章 控制流程

§ 3.1	语句和块.....	(73)
§ 3.2	IF-ELSE.....	(74)
§ 3.3	ELSE-IF.....	(75)
§ 3.4	开关语句SWITCH.....	(78)
§ 3.5	WHILE和FOR循环.....	(81)
§ 3.6	DO-WHILE循环.....	(86)
§ 3.7	中止与继续.....	(88)
§ 3.8	转向和标号.....	(89)

### 第四章 函数和程序结构

§ 4.1	函数的基础.....	(93)
§ 4.2	返回非整数值的函数.....	(98)
§ 4.3	外部变量.....	(101)
§ 4.4	作用域规则.....	(111)
§ 4.5	头文件.....	(113)
§ 4.6	静态变量.....	(115)
§ 4.7	寄存器变量.....	(117)
§ 4.8	块结构.....	(118)

§ 4.9	初始化	( 119 )
§ 4.10	递归	( 121 )
§ 4.11	C 预处理程序	( 124 )
4.11.1	文件包含	( 124 )
4.11.2	宏替换	( 125 )
4.11.3	条件包含	( 127 )

## 第五章 指针和数组

§ 5.1	指针和地址	( 130 )
§ 5.2	指针和函数参数	( 132 )
§ 5.3	指针和数组	( 136 )
§ 5.4	地址运算	( 140 )
§ 5.5	字符指针和函数	( 145 )
§ 5.6	指针数组和指向指针的指针	( 150 )
§ 5.7	多维数组	( 155 )
§ 5.8	指针数组的初始化	( 158 )
§ 5.9	指针与多维数组	( 160 )
§ 5.10	命令行参数	( 161 )
§ 5.11	指向函数的指针	( 168 )
§ 5.12	复杂说明	( 170 )

## 第六章 结构

§ 6.1	结构的基础	( 181 )
§ 6.2	结构与函数	( 184 )
§ 6.3	结构数组	( 189 )
§ 6.4	指向结构的指针	( 195 )
§ 6.5	自引结构	( 198 )
§ 6.6	查表	( 205 )

§ 6.7 类型定义 .....	( 209 )
§ 6.8 联合 .....	( 222 )
§ 6.9 字段 .....	( 214 )

## 第七章 输入和输出

§ 7.1 标准输入和输出 .....	( 217 )
§ 7.2 格式输出—PRINTF .....	( 222 )
§ 7.3 可变长度的参数表 .....	( 223 )
§ 7.4 格式化输入—SCANF .....	( 226 )
§ 7.5 文件存取 .....	( 230 )
§ 7.6 出错处理—STDERR和EXIT .....	( 235 )
§ 7.7 整行输入和输出 .....	( 237 )
§ 7.8 其它函数 .....	( 239 )
7.8.1 字符串操作 .....	( 240 )
7.8.2 字符类别的测试和转换 .....	( 240 )
7.8.3 退回一个字符—UNGETC .....	( 241 )
7.8.4 命令的执行 .....	( 241 )
7.8.5 存储管理 .....	( 242 )
7.8.6 数学函数 .....	( 243 )
7.8.7 随机函数 .....	( 243 )

## 第八章 与UNIX系统的接口

§ 8.1 文件描述符 .....	( 246 )
§ 8.2 低级I/O——读和写 .....	( 247 )
§ 8.3 打开、创建、关闭、拆开 .....	( 240 )
§ 8.4 随机存取——LSEEK .....	( 252 )
§ 8.5 例子—FOPEN和GETC的实现 .....	( 255 )
§ 8.6 列目录表的实例 .....	( 261 )

§ 8.7 存储分配程序实例 .....	( 271 )
----------------------	---------

## 第九章 C语言的微机版本

§ 9.1 MS-C的操作和使用 .....	( 280 )
9.1.1 概述 .....	( 281 )
9.1.2 编译软件的组织 .....	( 283 )
9.1.3 MS-C编译命令的执行 .....	( 286 )
9.1.4 运行连接程序 .....	( 288 )
9.1.5 库的管理 .....	( 291 )
§ 9.2 TURBO C的操作和使用 .....	( 295 )
9.2.1 概述 .....	( 296 )
9.2.2 TURBO C的安装 .....	( 226 )
9.2.3 TURBO C的集成开发环境 .....	( 298 )
9.2.4 集成环境中编译与执行的处理过程 .....	( 301 )
9.2.5 DOS环境中编译与执行的处理过程 .....	( 303 )

## 第10章 Turbo-C程序实例及编程经验

§ 10.1 大程序框架 .....	( 309 )
§ 10.2 复制文件 .....	( 314 )
§ 10.3 列目录程序 .....	( 316 )
§ 10.4 高效的全盘复制程序 .....	( 323 )
§ 10.5 打印机字体控制 .....	( 332 )
§ 10.6 工具函数模块 .....	( 338 )
§ 10.7 建立用户库 .....	( 345 )
§ 10.8 按记录号随机存取文件 .....	( 348 )
§ 10.9 Turbo-C 不经汉化处理汉字 .....	( 355 )
§ 10.10 建立 Shell .....	( 357 )
§ 10.11 批处理菜单程序 .....	( 397 )

§ 10.12 编程经验 .....	( 361 )
--------------------	---------

## 附录A 标准库

A1 输入和输出; <Stdio.h> .....	( 369 )
A1.1 文件操作 .....	( 369 )
A1.2 格式输出 .....	( 372 )
A1.3 格式输入 .....	( 375 )
A1.4 字符输入和输出函数 .....	( 377 )
A1.5 直接输入和输出函数 .....	( 379 )
A1.6 文件定位函数 .....	( 379 )
A1.7 出错函数 .....	( 380 )
A 2 字符类别测试; <Ctype.h> .....	( 381 )
A 3 串函数; <String.h> .....	( 382 )
A 4 数学函数; <math.h> .....	( 385 )
A 5 实用函数; <Stdlib.h> .....	( 387 )
A 6 诊断; <assert.h> .....	( 391 )
A 7 可变参数表; <Stdarg.h> .....	( 391 )
A 8 非局部转移; <setjmp.h> .....	( 392 )
A 9 信号; <Signal.h> .....	( 393 )
A10. 日期和时间函数; <time.h> .....	( 394 )
A11. 实施过程定义的各种界限; <limits.h>和 <float.h>.....	( 397 )

# 第一章 C 语言入门

为了使读者更快地学会 C 语言，本章首先给出一些 C 语言的程序，让读者看看 C 语言究竟是什么样子，了解 C 语言的一个概貌。对于 C 语言的一些细节、规则以及有关的概念这里暂不述及。

## § 1.1 起 步

通过编写程序来学习程序设计语言这是一种好方法。写出的第一个程序要尽可能的简单，例如打印或显示一串字符。

打印下列单词：

hello, world

要输出这两个英文单词，必须事先编写一段程序，然后对它进行编译、连接、运行并输出所需要的结果。

用 C 语言编写打印“hello, world”的程序如下：

```
#include<stdio.h> /* std—standard标准 */  
main ( ) /* oi—input output输入输出 */  
{
```

```
printf ("hello, world\n");  
}
```

如何运行这个程序取决于所使用的系统。如果是在 UNIX 操作系统下，那么生成的源程序文件名的后缀为“.c”。如果用 hello.c 作为输出“hello, world”单词的源程序文件的名字，那么在 UNIX 系统下，可以用“cc”命令对源程序进行编译：

```
cc hello.c
```

如果程序没有任何错误，那么编译就会正常进行，而且生成一个可执行文件 a.out。如果要运行可执行文件 a.out，则敲入命令：

```
a.out
```

即可输出运行结果：

```
hello, world
```

在其它操作系统下，这种编译、装配、运行等规则会有所不同（如何在 PC-DOS 下运行，请参见第九章）。现在对程序自身作一些说明。不论程序的大小如何，都是由函数和变量组成的。一个函数中含有进行计算操作的语句，变量用于存储计算的值。C 语言的函数类似于 FORTRAN 子例程子程序和函数或 PASCAL 的过程和函数。上面的例子是一个命名为 main 的函数。函数可以取任何名字，而“main”是一个特殊的名字，在程序执行时必须以 main 开始，这意味着每个程序都必须在某处有一个 main。main 也可以调用其它函数以帮助完成自身的工作。那些被调用的函数有的是自己事先写好的，有的来自事先写进程序库的函数。程序的第一行：

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

告诉编译程序要把一个称为 `stdio.h` 的标准输入输出函数头文件包含进来，这里的 `h` 是 `head` 的缩写。该文件由若干个函数头和一些宏代换组成，因此称为头文件。该行大多写在源程序文件的开头。标准库在第七章和附录B中描述。

数据与函数之间进行通讯的一种方法是将称为自变量的一个参数表提供给函数，函数名后用圆括号括起来的部分就叫参数表。

```
#include <stdio.h> 含标准库的信息
```

```
main ( )           定义无参函数名
```

```
{                  main语句的开始
```

```
printf("hello, world\n");
```

```
main调用库函数依次打印字符;\n为换行符。
```

```
}                main语句的结束。
```

在这个例子中，`main`定义的是一个没有参数的函数，称为无参函数，用空表 `( )` 来表示。

函数中的语句用花括号括起来，`main`函数仅包含一个语句：

```
printf("hello, world\n");
```

其中`printf`是由系统提供的函数。这个例子就是用 `"hello, world\n"` 参数来调用`printf`函数。`printf`是用于打印输出的库函数，其中的字符必须用`"`和`"`括起来。

如象`"hello, world\n"`，这种用双引号括起来的字符序列称为字符串或字符常数。这里我们仅用了字符串作为`printf`函数的参数。

字符串中的字符`"\n"`是C语言的换行标记。当打印



中遇到“\n”时，系统将打印头自动移到下一个新行的左边开始打印。

如果没有“\n”换行符，在输出打印时就不会换行。所以在printf参数中必须写上“\n”作为换行符。若函数写成：

```
printf("hello, world  
");
```

那么在C语言编译时将产生出错信息。

printf决不会自动换行。因此，可以用多个较短的字符来多次调用printf函数以得到一个较长的输出行。前面的例子也可以写成：

```
#include <stdio.h>  
main( )  
{  
    printf("hello, ");  
    printf("world");  
    printf("\n");  
}
```

这与上一种方法的输出结果完全一样。

注意：“\n”。是一个整体，仅表示单个字符。象“\n”这样在反斜线后再跟一个字符称为转义序列，它用于表示“回车”、“退路”等程序中看不见的特殊字符。还有一些转义序列如：\t表示制表符，\b表示空格符，\"表示双引号和\\表示反斜线。在2.3节中有一个完整的表，专门用来说明这些字符。

**练习1—1** 在Turbo—C中运行“hello, world”程序，