



语言程序设计基础 与实例操作教程

骆焦煌 主编



中央广播电视大学出版社



C语言程序设计基础 与实例操作教程

第2版 王 强

清华大学出版社

C 语言程序设计基础与 实例操作教程

骆焦煌 主编

中央广播电视大学出版社

北 京

内容简介

全书共分为 11 章，分别介绍了 C 语言概述，数据类型、运算符与表达式，顺序结构，选择结构程序设计，循环结构程序设计，数组，函数，编译预处理，指针，结构体与共用体及文件等内容。

图书在版编目 (CIP) 数据

C 语言程序设计基础与实例操作教程/ 骆焦煌主编.
—北京: 中央广播电视大学出版社, 2014.1
ISBN 978-7-304-05643-8

I. ①C… II. ①骆… III. ①C 语言—程序设计
IV. ①TP312

中国版本图书馆 CIP 数据核字 (2014) 第 139941 号

版权所有，翻印必究。

C 语言程序设计基础与实例操作教程

骆焦煌 主编

出版·发行: 中央广播电视大学出版社
电话: 营销中心 010-58840200 总编室 010-68182524
网址: <http://www.crtvup.com.cn>
地址: 北京市海淀区西四环中路 45 号
邮编: 100039
经销: 新华书店北京发行所

策划编辑: 苏 醒	责任编辑: 谷春林
印刷: 北京雷杰印刷有限公司	印数: 0001~3000
版本: 2014 年 1 月第 1 版	2014 年 1 月第 2 次印刷
开本: 787×1092 1/16	印张: 17.25 字数: 406 千字

书号: ISBN 978-7-304-05643-8
定价: 43.00 元

(如有缺页或倒装, 本社负责退换)

前言

PREFACE

C语言是目前国际上最流行和使用最广泛的计算机高级编程语言之一，因其简洁、表达能力强、功能丰富、可移植性好和目标程序质量高，受到编程人员的普遍青睐。

目前，C语言程序设计书籍版本繁多，但普遍存在的问题是针对性不强、技能训练的实践性不够，过于重视理论知识。本书的编写采用“以理论够用为基础，以实例操作为主线，重在培养操作技能”的模式，每章附有大量的练习题和实验。

本书融合了作者多年的教学实践和项目开发经验，具有如下特点：

(1) 基础知识够用为度。内容全面，结构合理，文字简练。

(2) 采用理论和实例相结合的方式，便于读者掌握知识点、提高实际操作能力。

(3) 注重现实社会发展和就业的需求，以培养岗位群的综合能力为目标，注重训练，强化应用，有针对性地培养读者的职业技能。

(4) 本书配有电子教学课件、实验项目及综合练习题。

本书通过大量的例题讲解了C语言程序设计的基本思想、方法和解决实际问题的技能。全书共分为11章，分别介绍了C语言概述，数据类型、运算符与表达式，顺序结构，选择结构程序设计，循环结构程序设计，数组，函数，编译预处理，指针，结构体与共用体及文件等内容。

本书由骆焦煌担任主编，由马静媛、王紫薇和邓博担任副主编，参加编写的还有梁艳和江荔。具体分工如下：骆焦煌编写第1章至第5章和综合练习题，



并负责全书的统稿工作；马静媛编写第 6 至 7 章；王紫薇和邓博共同编写第 8 至 10 章；梁艳和江荔共同编写第 11 章和附录部分。

由于编写时间仓促，加之编者水平有限，书中难免有不足之处，敬请广大读者批评指正。

编 者

目 录

CONTENTS

第 1 章 C 语言概述

1.1 C 语言出现的历史背景	1
1.2 C 语言的特点	2
1.3 简单的 C 程序介绍	3
1.4 C 程序上机调试过程	5
1.5 Turbo C 的集成开发环境	6
1.6 习题	9
1.7 C 语言的运行环境和运行过程实验	11

第 2 章 数据类型、运算符与表达式

2.1 C 语言的数据类型	13
2.2 常量与变量	14
2.3 整型数据	15
2.4 浮点型数据	17
2.5 字符型数据	18
2.6 变量赋初值	21
2.7 各类数值型数据间的混合运算	21
2.8 算术运算符和算术表达式	21
2.9 赋值运算符和赋值表达式	23
2.10 逗号运算符和逗号表达式	26
2.11 位运算符	26
2.12 习题	29
2.13 数据类型、运算符和表达式实验	32



第 3 章 顺序结构

3.1 C 语句概述.....	34
3.2 赋值语句	35
3.3 数据输入输出的概念及在 C 语言中的实现.....	35
3.4 字符数据的输入输出	36
3.5 格式输入与输出	37
3.6 顺序结构程序设计举例	41
3.7 习题	41
3.8 顺序结构实验	43

第 4 章 选择结构程序设计

4.1 关系运算符和关系表达式.....	45
4.2 逻辑运算符和逻辑表达式.....	46
4.3 if 语句	46
4.4 switch 语句	52
4.5 程序举例	53
4.6 习题	54
4.7 选择结构实验	58

第 5 章 循环结构程序设计

5.1 概述	62
5.2 goto 语句以及用 goto 语句构成循环	62
5.3 用 while 语句实现循环.....	63
5.4 用 do-while 语句实现循环	64
5.5 用 for 语句实现循环.....	66
5.6 循环的嵌套	69
5.7 几种循环的比较	70
5.8 break 语句和 continue 语句.....	70
5.9 程序举例	71
5.10 习题	74
5.11 循环结构实验.....	79

第6章 数 组

6.1 一维数组的定义和引用	84
6.2 二维数组的定义和引用	89
6.3 字符数组	93
6.4 习题	100
6.5 数组实验	105

第7章 函 数

7.1 概述	108
7.2 函数定义的一般形式	110
7.3 函数参数和函数的值	111
7.4 函数的调用	114
7.5 函数的嵌套调用	117
7.6 函数的递归调用	118
7.7 数组作为函数参数	120
7.8 局部变量和全局变量	125
7.9 变量的存储类别	127
7.10 内部函数和外部函数	132
7.11 习题	133
7.12 函数的定义及调用实验	137

第8章 编译预处理

8.1 宏定义	140
8.2 “文件包含”处理	144
8.3 条件编译	146
8.4 习题	147
8.5 编译预处理实验	149

第9章 指 针

9.1 指针的概念	151
9.2 指针和变量	152

9.3 指针和数组	156
9.4 字符串和指针变量	163
9.5 函数和指针	167
9.6 指针数组和指向指针变量的指针	169
9.7 习题	172
9.8 指针实验	174

第 10 章 结构体与共用体

10.1 概述	176
10.2 结构类型与结构变量的定义	177
10.3 结构体变量的初始化和引用	180
10.4 结构体数组	183
10.5 结构体指针	185
10.6 结构体指针与链表	187
10.7 共用体	195
10.8 枚举类型	198
10.9 typedef 类型定义	200
10.10 习题	201
10.11 结构体和共用体实验	204

第 11 章 文 件

11.1 文件概述	206
11.2 标准文件操作	208
11.3 非标准文件操作	215
11.4 习题	218
11.5 文件实验	219

综合练习题	221
-------------	-----

参考文献	253
------------	-----

附录 1 常用字符与 ASCII 代码对照表	254
------------------------------	-----

附录 2 C 语言运算符的优先级与结合性	256
----------------------------	-----

附录 3 C 语言常用的库函数	257
-----------------------	-----

附录 4 C 语言中最容易出现错误的参考	264
----------------------------	-----

第1章 C语言概述

C语言自问世以来,对程序设计语言的发展产生了巨大的影响,它已成为国内外应用最广泛的高级程序设计语言之一。它不仅可用来编写系统软件,也可以用来编写应用软件。

本章将对C语言的发展历史和特点、程序结构和书写格式、程序的编译和运行等方面进行详细介绍,使读者对C语言有一个初步的认识。



知识点

- C语言的背景。
- C语言的特点。
- C语言函数的基本概念。
- C语言程序的上机调度过程。

1.1 C语言出现的历史背景

1.1.1 C语言的发展史

C语言可以说是伴随着对操作系统的改写而产生的。早期的操作系统等系统软件主要是用汇编语言编写的,但汇编语言过分依赖计算机硬件,而且程序的可读性和可移植性比较差,因此人们就想找到既具有汇编语言的某些特性(如直接对硬件地址进行操作),又有一般高级语言特性(如可读性和可移植性好)的语言,C语言就是在这种情况下产生的。

C语言是从B语言发展来的,它的发展历程如下:

1. ALGOL 60 语言

1960年出现的ALGOL 60 (ALGOrithmic Language 60)是一种面向问题的高级语言,它离硬件比较远,不宜用来编写系统软件。

2. CPL 语言

1963年英国的剑桥大学推出了CPL (Combined Programming Language) 语言。CPL语言在ALGOL 60的基础上更接近硬件一些,但规模比较大,在实际应用中难以实现。

3. BCPL 语言

1967年英国剑桥大学的Martin Richards对CPL语言进行简化得到BCPL(Basic Combined Programming Language)语言。



4. B 语言

1970 年美国贝尔实验室的 Ken Thompson 以 BCPL 语言为基础, 进一步简化, 设计出很简单的且接近硬件的 B 语言 (取 BCPL 的第一个字母)。但 B 语言过于简单, 功能有限。

5. C 语言

1972 年至 1973 年间, 贝尔实验室的 D. M. Ritchie 在 B 语言的基础上设计出 C 语言 (取 BCPL 的第二个字母)。C 语言保持了 BCPL 和 B 语言的优点 (如接近硬件、精练), 又克服了它们的缺点 (过于简单、数据无类型等)。1973 年, K. Thompson 和 D. M. Ritchie 两人合作把 Unix 操作系统 90% 以上的组成部分用 C 语言改写, 成为第 5 版 Unix。

1.1.2 C 语言的使用及其标准化

1. C 语言的使用

(1) C 语言在底层编程中的应用。C 语言比较贴近硬件, 因此在单片机编程, 通信接口编程, 自动化控制方面的编程, 手机、POS 机、银行自动存取款机等方面的编程使用 C 语言比较多。

(2) C 语言在图形编程中的应用。C 语言图形编程上应用较多, 如图形化的游戏开发、图形界面的设计等。

(3) C 语言在普通信息系统方面的应用。在数据库技术普及之前, 使用 C 语言编写的信息系统比比皆是。因为 C 语言是一种很好的结构化程序设计语言, 并且 C 语言中有丰富的数据类型 (如数组、结构体类型等), 为数据的处理提供了很好的工具。

2. C 语言的标准化

1978 年, 以 Unix 第 7 版中的 C 编译程序为基础, Brian. W. kernighan 和 Dennis. M. Ritchie 合著了影响深远的名著 *The C Programming Language*。这本书介绍的 C 语言成为以后各种版本 C 语言的基础, 称为标准 C 语言。

1983 年, 美国国家标准化协会 (American National Standards Institute, ANSI) 根据 C 的发展变化和扩充, 制定了新的标准, 称为 ANSI C。1988 年 Brian W. Kernighan 和 Dennis M. Ritchie 按 ANSI C 标准重新修改 *The C Programming Language* 一书。

1987 年, ANSI 又公布了新标准——87 ANSI C。这个标准在 1990 年时被国际标准化组织 (International Standard Organization, ISO) 所接受, 成为以后流行的 C 语言的基础。

1.2 C 语言的特点

(1) C 语言能实现汇编语言的大部分功能。C 语言可以直接对硬件进行操作, 能进行位 (bit) 操作。因此, C 语言既有高级语言的功能, 又具有低级语言的许多功能。它既可以用来写系统软件, 又可以用来写应用软件。

(2) 用 C 语言写的程序可移植性好。用标准 C 语言编写的程序几乎不用做修改就能用

于各种型号的计算机和各种操作系统。

(3) C 程序生成的目标代码质量高, 程序执行效率高。与其他高级语言相比 (如 PASCAL、BASIC 等), C 程序的代码效率和质量更高, 但它仍比汇编程序生成的目标代码效率低 10%~20%。

(4) 具有结构化的控制语句, 是一种结构化程序设计语言。C 语言中有 3 种基本结构的语句, 如 if...else 语句、while 语句、switch 语句、for 语句, 并且用函数作为程序的模块单位, 实现模块化程序设计。

(5) 语言简洁、紧凑, 使用方便、灵活。C 语言中一共只有 32 个关键字, 9 种控制语句, 程序书写形式自由 (如一行可以写多个语句, 一个语句也可以分成几行写), 压缩了一切不必要的成分, 减少了代码中关键字所占用的存储空间。

(6) 运算符丰富。C 语言中的运算符共有 34 种, 包括算术运算符、逻辑运算符、关系运算符、位运算符、自增自减运算符等。并且 C 语言将括号、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。

(7) 数据结构丰富。C 语言中数据类型有整型、实型、字符型、数组类型、指针类型、结构体类型、枚举类型等, 还可以自定义数据类型。能够用来实现各种如链表、树、栈等复杂数据结构的运算。

1.3 简单的 C 程序介绍

例 1.1 输出一行信息

```
#include <stdio.h>    /*文件包含*/
void main()           /*主函数*/
{
    /*函数体开始*/
    printf("This is a C program.\n"); /*输出语句*/
}                      /*函数体结束*/
```

本程序的作用是输出一行信息: This is a C program.

说明:

(1) main 是主函数名, void 是函数类型。

(2) include 是文件包含命令, 扩展名为 h 的文件称为头文件, 表示在程序中要到这个文件中的函数。使用标准库函数时应在程序开头一行写: #include <stdio.h>。

(3) 每个 C 程序必须有一个主函数 main。

(4) {} 是函数开始和结束的标志, 不可省略。

(5) printf 是函数调用语句, printf 函数的功能是把要输出的内容送到显示器去显示。

(6) 每个 C 语句以分号结束。

例 1.2 求两数之和

```
void main()           /*求两数之和*/
```

```

{
    int a,b,sum;    /*声明，定义整形变量 a,b,sum*/
    /*以下 3 行为 C 语句*/
    a=123;          /*将 123 赋给变量 a*/
    b=456;          /*将 456 赋给变量 b*/
    sum=a+b;        /*计算变量 a 和 b 的和，并将值赋给变量 sum*/
    printf ("sum is%d\n",sum); /*输出结果 sum*/

}

```

程序输出结果如下：

sum is 579

说明：/*.....*/表示注释。注释只是给人看的，对编译和运行不起作用。所以可以用汉字或英文字符表示，可以出现在一行中的最右侧，也可以单独成为一行。

例 1.3 基本程序结构例子：主函数与自定义函数及调用

```

#include <stdio.h>

void main()          /*主函数*/
{
    int max(int x,int y); /*对被调用函数 max 的声明*/
    int a,b,c;          /*定义变量 a、b、c */
    scanf("%d,%d",&a,&b); /*输入变量 a 和 b 的值*/
    c=max(a,b);         /*调用 max 函数，将得到的值赋给 c */
    printf("max=%d\n",c); /*输出 c 的值*/
}

int max(int x,int y);
{
    int z;
    if(x>y)
        z=x;
    else
        z=y;
    return(z);
}

```

程序运行情况如下：

输入 8,5 ✓ (输入 8 和 5 赋给 a 和 b)。

输出 max=8 (输出 c 的值)

说明：本程序包括 main 和被调用函数 max 两个函数。max 函数的作用是将 x 和 y 中较大者的值赋给变量 z。return 语句将 z 的值返回给主调函数 main。

通过前面 3 个例子，我们可以看到 C 语言具有的特点：

(1) C 程序是以函数为基本单位，整个程序由函数组成。其中主函数是一个特殊的函数，一个完整的 C 程序至少要有有一个且仅有一个主函数，它是程序启动时的唯一入口。除主函数外，C 程序还可包含若干其他 C 标准库函数和用户自定义的函数。这种函数结构的特点使 C 语言便于实现模块化的程序结构。

(2) 函数是由函数说明和函数体两部分组成。函数说明部分即函数的第一行，包括对函数名、函数类型、形式参数等的定义和说明；函数体即函数首部下面的大括号{……}内的部分，包括对变量的定义和执行程序两部分，由一系列语句和注释组成。整个函数体由一对花括号括起来。

(3) C 语言中的每个语句必须以“;”（分号）结束。分号是 C 语言语句的标志。C 语言书写比较自由，一行可以写多个语句（每个语句以;结束），一个 C 语句也可以写在多行上。

(4) C 程序中的/*……*/是用作注释的。注释是用来给读程序的人看的，在程序编译中并不编译。/*和*/必须成对使用，在它们之间的内容是注释内容。

(5) C 语言中没有输入/输出语句。C 程序中的输入/输出是由标准函数来完成的，例如，scanf()函数是用来输入，而 printf()函数是用来输出的。

1.4 C 程序上机调试过程

用高级语言编写的程序称为“源程序”。从根本上说，计算机只能识别和执行由 1 和 0 组成的二进制的指令，而不能识别和执行用高级语言写的指令。为了使计算机能执行高级语言源程序，必须用一种称为“编译程序”的软件，把源程序翻译成二进制形式的“目标程序”，然后将该目标程序与系统的函数库和其他目标程序连接起来，形成可执行的目标程序，具体过程如图 1-1 和 1-2 所示。

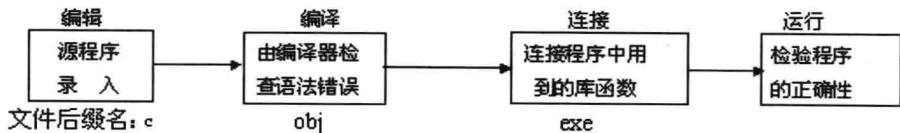


图 1-1

在机器上运行 C 程序的步骤：

1. 编辑

编辑是指使用文本编辑工具软件输入和修改 C 语言源程序，最后以文本文件的形式存放在磁盘上，文件名由用户自己选定，扩展名一般为 c，如 sample1.c、sort.c 等。

2. 编译

编译是将 C 源程序翻译成二进制目标程序。在 Turbo C 2.0 环境下目标文件的扩展名为 obj。其他编译环境下由系统自动确定。

3. 连接

连接是用连接程序将编译过的目标程序和程序中用到的库函数连接装配在一起，形成可执行的程序代码（二进制代码）。可执行文件的扩展名一般为 `exe`。

4. 运行

运行是将可执行文件调入内存运行，以获得程序的运行结果。通常，在 DOS 操作系统提示符下直接键入可执行文件名，或在 Windows 操作系统下用鼠标双击可执行文件图标即可。

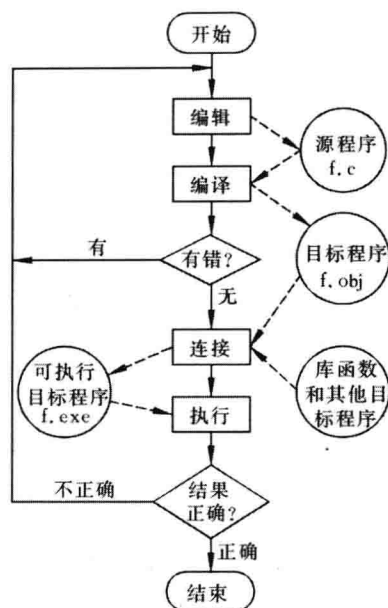


图 1 - 2

1.5 Turbo C 的集成开发环境

目前广泛使用的 C 语言源程序编译工具基本部分是相同的，但也有一些不同。在计算机上使用的有 Microsoft C、Turbo C、Quick C、Borland C 等，它们的不同版本之间有些差异，但都支持 ANSI C。下面介绍 Turbo C/C++ 3.0 编译工具的使用。

1. 启动 Turbo C 集成环境

启动 Turbo C 之前，首先要将系统安装在指定的目录下，然后执行相关的命令将 TCPP.exe 加载到内存中就可以使用了。

Turbo C 启动后，屏幕首先出现 Turbo C 主屏幕和版本信息。按任意键版本信息消失，这时系统进入 Turbo C 编辑环境。其界面如图 1 - 3 所示。



图 1 - 3

2. 编辑 C 的源程序

输入源程序之后（结果如图 1 - 4），将文件保存（如需要的话）。保存方法是在【File】菜单下选择【Save】选项，并在随后的对话框中输入文件名 first.c，选择【OK】。

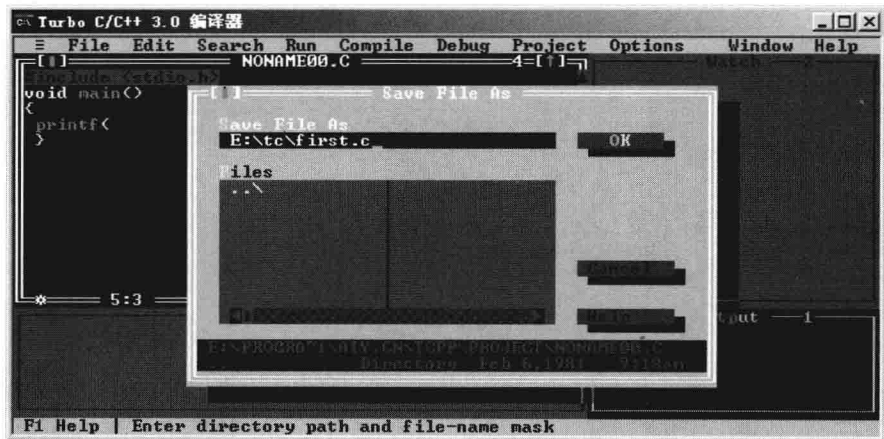


图 1 - 4

3. 编译源文件

在【Complie】菜单中，选择【Complie】或按下快捷键 Alt+F9，程序将被编译，如图 1 - 5 所示。如果程序出现语法错误，系统将具体显示错误的原因，此时，按任意键均可退出显示错误的对话框，按 Enter 键回到编辑状态程序出错的地方，方便对程序的修改。