

高职高专公共基础课规划教材

GAOZHIGAOZHUANGONGGONGJICHUKEGUIHUAJIAOCAI



C语言 程序设计基础

■ 桂绍勇 郭小进 编



中国电力出版社

<http://jc.cepp.com.cn>



C语言 程序设计基础

■ 桂绍勇 郭小进 编
■ 戴宝华 主审



中国电力出版社

<http://jc.cepp.com.cn>



林妹收默野基共公基高限高

GAOZHIGONGJIGUNJICHUKESHIHUAJIAOCHENG

内 容 提 要

本书为高职高专公共基础课规划教材。

C语言是目前被广泛应用的一种编程语言,不但在计算机编程中得到广泛应用,而且在其他领域也成为主要的编程工具,例如在单片机、DSP、嵌入系统等系统开发中更是作为基础性的开发语言。

本书根据高职高专教育的教学特点和教学要求,结合编者多年的计算机程序开发、单片机开发和嵌入系统开发的经验,并吸取了市面上同类图书的精华后,精心规划、编写而成。

本书通俗易懂,内容由易到难,概念清晰,重点突出,习题丰富。

本书可作为高职高专院校C语言程序设计课程的教材或参考用书,也可以作为相关人员的自学教材。

图书在版编目(CIP)数据

C语言程序设计基础/桂绍勇,郭小进编. —北京:中国电力出版社,2008

高职高专公共基础课规划教材

ISBN 978-7-5083-6491-9

I. C… II. ①桂…②郭… III. C语言—程序设计—高等学校:技术学校—教材 IV. TP312

中国版本图书馆CIP数据核字(2008)第011033号

中国电力出版社出版、发行

(北京三里河路6号 100044 http://jc.cepp.com.cn)

北京丰源印刷厂印刷

各地新华书店经售

*

2008年2月第一版 2008年2月北京第一次印刷

787毫米×1092毫米 16开本 13.5印张 318千字

定价 19.80元

敬告读者

本书封面贴有防伪标签,加热后中心图案消失
本书如有印装质量问题,我社发行部负责退换

版权专有 翻印必究

前言

C语言是目前运用最广泛的计算机编程语言之一。

C语言属于结构化程序设计语言，它具有简洁、灵活、高效、可移植性强等特点，能直接对位、字节和地址进行操作，具有丰富的数据类型。C语言已被广泛的应用于系统软件和应用软件的开发中。

C语言历史悠久，自从其诞生到现在得到了人们的广泛认同，不仅计算机专业人员喜欢使用，非计算机专业人员也越来越重视把C语言作为自己应用领域中的主要程序设计语言之一。例如原来的单片机程序都是用汇编语言编写的，汇编语言生涩难懂，而且效率低，自从引入C语言编程环境后，大部分开发人员都转向用C语言设计单片机程序。目前几乎所有型号的单片机都支持C语言程序设计方式。改用C语言作为开发工具后，使得单片机程序的开发变得方便而且高效。

掌握C语言已成为迈入计算机程序领域的一块敲门砖，可以毫不夸张的说，是否掌握C语言可作为程序员水平的一个尺度。而C语言由于其特殊的灵活性使得学习起来有一定的困难，学生总抱怨不好学，老师也抱怨不好教。因此，编写一本概念清晰、层次分明、由浅入深、容易掌握的教材显得日趋迫切，本书的编写正是朝着这个方向努力的。

多年来，我们一直在武汉电力职业技术学院从事C语言程序设计课程的教学工作，对学生学习C语言时的一些难点问题较为清楚；同时，我们也有多年利用C语言编程的实践经验，清楚C语言的特点、C语言编程技巧以及用C语言编程时应注意的问题。所以在编写本书时，尽量体现出教学和编程实践中的这些经验，使初学者少走弯路。本书特色为：

- (1) 结构合理，内容循序渐进，由浅如深，符合学生学习特征和认知方式；
- (2) 叙述通俗流畅，条理清楚，可读性强；
- (3) 示例较多，习题设计合理，可促进学生多上机来掌握知识；
- (4) 本书可分为C语言基础和C语言应用两大部分，以利于按需施教，如课时紧张，则第9、10章应用部分可以不讲；
- (5) 书中用较大篇幅介绍了位运算、在文本模式和图形模式下的编程开发等内容，紧密联系生活和生产实践，以利于对C语言的应用；
- (6) 图文并茂，简洁易懂、易于学习。

本书共10章。其中第1、3、7、8、9、10章由桂绍勇编写，第2、4、5、6章由郭小进编写。在本书的编写过程中，得到了西安电子科技大学戴宝华副教授的大力支持和帮助，在此致以衷心的感谢！

限于作者水平，错漏之处在所难免，恳请读者不吝赐教，提出批评指正意见。

编者

目 录

前言

第1章 概述	1
1.1 简单C程序介绍	1
1.1.1 几个简单C程序例子	1
1.1.2 程序分析	3
1.1.3 C语言程序结构	4
1.2 C语言历史、特点与发展	5
1.2.1 C语言的历史	5
1.2.2 C语言的特点	5
1.2.3 C语言的发展	5
1.3 C语言开发环境介绍	6
1.3.1 C语言开发调试步骤	6
1.3.2 C程序上机环境介绍	7
习题1	9
第2章 数据类型和运算符	11
2.1 进制基本知识	11
2.1.1 进制	11
2.1.2 二进制数的基本运算	13
2.1.3 进制之间转换	13
2.1.4 字节	15
2.2 数据表示形式和编码	15
2.2.1 有符号数和无符号数	15
2.2.2 原码	16
2.2.3 补码	16
2.2.4 小数的表示	16
2.2.5 编码	17
2.3 数据类型、常量和变量	17
2.3.1 数据类型相关概念	17
2.3.2 基本数据类型	18
2.3.3 常量	19
2.3.4 变量	21
2.4 运算符和表达式	22
2.4.1 算术运算符	23

2.4.2	赋值运算符	23
2.4.3	关系运算符和逻辑运算符	24
2.4.4	运算符优先级和结合性	24
2.4.5	复合赋值运算符	25
2.4.6	自加和自减运算符	25
2.4.7	sizeof 运算符和逗号运算符	26
2.4.8	表达式	27
2.4.9	表达式值的类型和类型转换	27
习题 2		29
第 3 章	三种基本程序结构	32
3.1	算法	32
3.1.1	算法设计	32
3.1.2	算法的描述	33
3.2	顺序结构	33
3.2.1	C 语言语句概述	34
3.2.2	格式输入与输出	35
3.2.3	顺序结构程序举例	37
3.3	选择结构	38
3.3.1	if 语句	38
3.3.2	条件运算符	39
3.3.3	switch 语句	40
3.3.4	选择结构程序举例	41
3.4	循环结构	43
3.4.1	while 语句	43
3.4.2	do~while 语句	44
3.4.3	for 语句	45
3.4.4	break 和 continue 语句	47
3.4.5	循环结构程序举例	48
3.5	程序结构的嵌套	50
3.5.1	if 语句的嵌套	50
3.5.2	程序结构嵌套举例	51
习题 3		53
第 4 章	数组和字符串	61
4.1	一维数组	61
4.1.1	一维数组的定义	61
4.1.2	一维数组的引用	61
4.1.3	一维数组的初始化	63
4.1.4	一维数组的应用举例	64

4.2	二维数组	65
4.2.1	二维数组的定义	66
4.2.2	二维数组的引用	66
4.2.3	二维数组的初始化	67
4.2.4	二维数组的应用举例	69
4.3	字符串和字符数组	70
4.3.1	字符数组的定义和引用	70
4.3.2	字符数组的初始化	71
4.3.3	字符数组的应用	71
	习题4	79
第5章	函数	81
5.1	函数的定义和声明	81
5.1.1	函数的定义	81
5.1.2	函数声明	83
5.2	函数的调用	84
5.2.1	函数的调用	84
5.2.2	函数执行过程	85
5.2.3	实参和形参之间值的传递	86
5.3	函数的递归调用	87
5.4	数组作为函数的参数	88
5.5	局部变量、全局变量、变量存储类型	90
5.5.1	变量的作用范围	90
5.5.2	全局变量和局部变量	91
5.5.3	变量的存储类型	92
	习题5	95
第6章	指针	97
6.1	地址	97
6.1.1	地址和变量	97
6.1.2	直接访问和间接访问	97
6.2	指针	98
6.2.1	指针的定义	98
6.2.2	指针赋值	98
6.2.3	通过指针访问变量	99
6.2.4	指针类型(指针指向变量的类型)	100
6.3	指针和数组	100
6.3.1	指向数组的指针	100
6.3.2	指针的运算	101
6.4	指针和函数	103
6.4.1	指针作为函数的参数	103

6.4.2	指针作为函数的返回值	105
习题 6		105
第 7 章	结构体、联合体和枚举	109
7.1	结构体类型	109
7.1.1	结构体类型的定义	109
7.1.2	结构体类型变量的定义	110
7.1.3	结构体类型变量的初始化和引用	111
7.1.4	结构体与数组	113
7.1.5	结构体数组应用举例	115
7.1.6	结构体作为函数参数	117
7.2	联合(共用)体类型	119
7.2.1	联合体类型的定义	119
7.2.2	联合体类型变量的定义、初始化和引用	120
7.3	枚举类型	121
7.3.1	枚举类型的定义	121
7.3.2	枚举类型变量的初始化和引用	122
7.4	自定义类型	123
习题 7		124
第 8 章	位运算	132
8.1	位运算符和位运算	132
8.1.1	按位与运算符——&	132
8.1.2	按位或运算符——	132
8.1.3	按位异或运算符——^	133
8.1.4	按位取反运算符——~	133
8.1.5	左移运算符——<<	133
8.1.6	右移运算符——>>	133
8.1.7	位复合赋值运算符	134
8.1.8	不同类型量参与位运算	134
8.2	位运算表达式和位运算的特殊用途	135
8.2.1	位运算表达式	135
8.2.2	置位	135
8.2.3	清零	136
8.2.4	取反	137
8.2.5	位检测	137
8.3	位段	138
8.3.1	位段的概念与定义	138
8.3.2	位段的使用说明	140
习题 8		141

第 9 章 文件	145
9.1 文件概述	145
9.1.1 文件的基本概念	145
9.1.2 文件的分类	145
9.1.3 文件操作的一般步骤	146
9.2 文件结构体和文件类型指针	146
9.2.1 文件结构体	146
9.2.2 文件类型指针	147
9.3 文件的打开和关闭	147
9.3.1 文件的打开	147
9.3.2 文件的关闭	149
9.4 文件的读写	150
9.4.1 字符读写函数 fputc 和 fgetc	150
9.4.2 字符串读写函数 fputs 和 fgets	152
9.4.3 格式化读写函数 fprintf 和 fscanf	153
9.4.4 数据块读写函数 fwrite 和 fread	154
9.5 文件定位和文件检测	156
9.5.1 ftell 函数	156
9.5.2 rewind 函数	156
9.5.3 fseek 函数	156
9.5.4 文件检测	158
习题 9	158
第 10 章 综合举例	163
10.1 文本模式程序设计	163
10.1.1 文本模式简介	163
10.1.2 文本模式函数简介	163
10.1.3 文本模式下的弹出菜单例子	166
10.2 图形模式程序设计	175
10.2.1 图形模式简介	175
10.2.2 图形模式函数简介	175
10.2.3 图形模式下时钟的例子	183
10.3 多文件程序和工程	185
10.3.1 工程的建立与编译连接	186
10.3.2 举例	187
10.3.3 外部变量和外部函数	187
10.3.4 文件包含实现多文件程序	188
10.3.5 工程中附加从属关系的指定	188
习题 10	188

第 1 章

概 述

C 语言,是目前国际上流行最广,使用人数最多的一种计算机程序设计语言。C 语言既可以用来编写应用程序,也可以用来编写系统软件,如著名操作系统 Windows、Linux、Unix 等,其内核大部分都是用 C 语言开发的。

随着操作系统的发展,编程环境和手段的改变,出现了很多其他编程语言,如现今非常流行的 Java、C++、C# 等。但不管编程语言如何的千变万化,C 语言仍然是这些新兴语言的基础,C 语言的编程风格、语法本质仍然可以在这些语言中找到。学习好 C 语言是学习好其他编程语言的基础,是向计算机编程领域迈出了第一步。

本章从几个简单的 C 语言程序的例子,引领读者初见 C 语言的端倪,了解 C 语言的结构、特点、发展历程;通过对 C 语言上机环境的介绍,使读者掌握 C 语言开发工具的使用,C 语言上机试验的步骤,程序的输入、保存、打开、编译、调试及运行等方法。

1.1 简单 C 程序介绍

1.1.1 几个简单 C 程序例子

有这样一个问题:有 2 个佐料瓶,分别盛放着酱油和醋。现要将两个瓶内的佐料互换,请列出操作步骤(程序)。

这个问题很简单,可以:

- ①取一空瓶;
- ②先将酱油倒入空瓶;
- ③再将醋倒入酱油瓶;
- ④最后将空瓶内的酱油倒入醋瓶。

其实,这就是程序。但这个工作直接让计算机做是做不了的,计算机听不懂人类语言,必须想办法将问题抽象,并依照先前人机约定好的语法规则将要执行的工作程序化,写出计算机能听懂的语言,才能让计算机为人们做事。

这个问题可以抽象为将两个数的交换,用数 A 和数 B 表示两个瓶子,值为 3 则表示装的是酱油,值为 4 则表示装的是醋。C 语言可以表示如 [例 1-1]。

【例 1-1】 编写 C 程序,将两个数 A 和 B 的值互换。

```
main()  
{  
    int A=3,B=4,C;  
    C=A;  
    A=B;
```

```

B=C;
printf("A=%d,B=%d\n",A,B);
}

```

【例 1-1】的程序就是一个完整的 C 语言程序，运行后可以得到以下结果：

A=4, B=3

初学者对程序中有些地方应该能看得懂，如第 4、5、6 行；但有些地方并不一定看得太明白，如第 3 行，main 是什么？printf 是什么？

其实这就是前面提到的“先前人机约定好的语法规则”，这些问题在后续课程的学习中会逐一得到解决。其实，学习本课程目的就是学习这些语法规则，写出能让计算机理解的 C 语言程序来。

看【例 1-2】和【例 1-3】的程序。

```

【例 1-2】
main()
{
    printf("*****\n");
    printf(" *      C 语言      *\n");
    printf("*****\n");
}

```

当程序运行后，屏幕上会显示：

```

*****
*      C 语言      *
*****

```

【例 1-3】有两个数 a、b，编写 C 程序输出其中最大者的值。

```

main()
{
    int a, b, c;
    scanf("%d%d", &a, &b);
    c=max(a, b);
    printf("Max is %d\n", c);
}

```

```

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

```

当程序运行后，屏幕上会出现：

23 76 ↵

Max is 76

第一行是键盘输入部分，“↵”表示空格，“↵”表示回车键。第二行是程序运行后的输出结果。

1.1.2 程序分析

先来分析 [例 1-1]。为了便于叙述，在 [例 1-1] 程序的基础上加入了一些注释。

```
main()
{
    int A=3, B=4, C;
    C=A;
    A=B;
    B=C;
    printf("A=%d, B=%d\n", A, B);
}
```

/* 第 1 行：主函数函数头 */
/* 第 2 行：主函数体开始 */
/* 第 3 行：定义 3 个变量，并且 A 赋初值 3, B 赋初值 4 */
/* 第 4 行：将 A 的值赋给 C */
/* 第 5 行：用 B 的值取代 A 的值，A 原来的值在 C 中 */
/* 第 6 行：用 C 的值取代 B 的值 */
/* 第 7 行：输出交换后 A 和 B 的值 */
/* 第 8 行：主函数体结束 */

(1) 介于“/*”和“*/”之间的部分，包括“/*”和“*/”符号本身称为注释，它们不是 C 语言的语句，不参与程序执行，只起到描述程序作用和备忘的功能。虽然注释不是程序必需部分，但合理的利用好注释能够培养良好的编程习惯。C 语言中，注释不允许嵌套使用，即不能“/*”和“*/”内再出现“/**/”部分。

(2) 程序的第 1 行是主函数 main 的“函数头”。可以看到，在先后出现的三个例题中都有 main 函数。C 语言规定，函数名为 main 的函数是 C 程序的主函数，在一个能够独立运行的程序中有且仅有一个 main 函数。

(3) 程序第 2 行的“{”和第 8 行的“}”将其中 5 条语句括在一起，构成 main 函数的函数体。

(4) 程序第 3 行定义 3 个整数型变量，并且为 A 和 B 赋了初值。int 表示整数类型，有关变量定义、变量类型在第 2 章中具体讨论。

(5) 程序的 4、5、6 行是相似的语句，这种语句是赋值语句，运算方向从右至左。如“C=A;”这行，运算方向 $C \leftarrow A$ ，将 A 的值赋给 C。

(6) 程序第 7 行是输出语句，调用了库函数 printf 输出变量 A 和 B 的值。“%d”表示十进制整数类型。有关 printf 函数的使用在第 3 章中具体讨论。

[例 1-2] 程序比较简单，只有 3 条语句。程序执行时会按原样输出系统函数 printf 中双引号中的内容，“\n”表示换行。

[例 1-3] 程序包括两个函数：主函数 main 和被调用函数 max。程序的第 4 行调用了库函数 scanf，从键盘读取数据输入到变量 a 和 b 中。第 5 行是赋值语句，运算方向从右至左，所以先执行调用 max 函数部分，程序将跳转到第 8 行执行。max 函数利用 if 语句（有关 if 语句的使用在第三章中具体讨论）比较 x 和 y 中的较大者，并将其赋给 z，最后通过 return

语句返回给调用者，即回到程序第 5 行，将 max 的返回值赋给变量 c，最后输出 c 的值。

1.1.3 C 语言程序结构

通过以上几个例题的分析，找出一般规律，可以看到：

一、C 语言程序是由函数构成的

一个能够独立运行的程序中至少应有一个 main 函数，且只能有一个。程序中可以包含多个函数，如 [例 1-3] 包含两个函数。函数是 C 语言程序的基本单位。函数之间可以相互调用，当发生函数调用时，程序转到被调用函数执行，执行完被调用函数的所有语句后又返回到调用处或它的下一语句处。被调用函数可以是系统已提供的库函数，如前例中调用的 scanf 和 printf 函数，也可以是自定义函数，如 [例 1-3] 中的 max 函数。

C 语言通过函数实现程序的模块化，这就是所谓第三代“面向结构化的程序设计语言”。

二、程序总是从 main 函数开始执行

main 函数是 C 语言程序中一个特殊的函数。它是整个 C 程序的“入口”和应该的“出口”。不论 main 函数在程序书写时处于什么位置，它都首先被执行。在 [例 1-3] 中，如果将 max 函数和 main 函数出现的位置对调，不会影响程序的执行顺序。

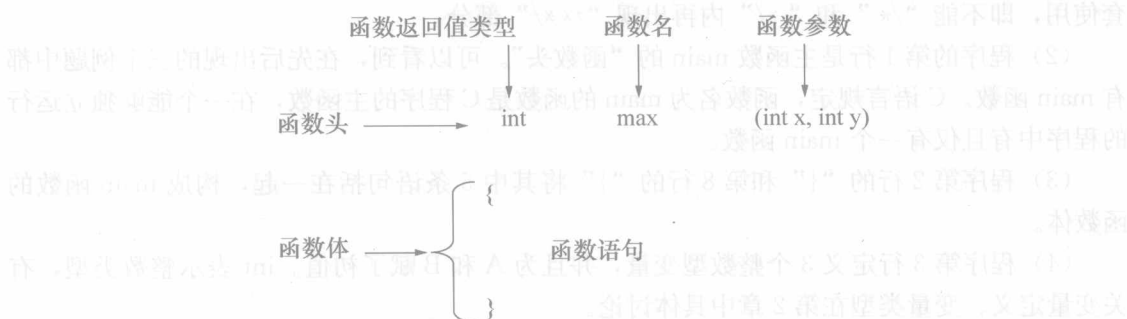
在某个局部位置，C 程序语句的执行顺序是从前向后，从上至下的。

三、C 语言函数由函数头和函数体构成

函数头部分包含对函数返回值类型的指定，函数名称，函数调用时应提供的参数个数以及类型的描述等信息。

函数体由一对正反大括号以及之间的 C 语句构成。

如 [例 1-3] 中 max 函数的结构为：



函数头中函数返回值类型、函数名、函数参数的前后顺序不能颠倒。函数参数部分由一对小括号首尾括起来紧跟在函数名的后面。

函数体一般包含变量定义和语句执行两部分。

(1) 变量定义。如 [例 1-3] 中 max 函数的“int z;”。其实 main 函数也不例外，请观察几个例题程序中 main 函数体开始部分。

(2) 语句执行部分。由若干条 C 语句构成。注意每条语句末尾都有“;”，这一点务必注意。

当然，有些情况下也可以没有变量定义部分，如 [例 1-2]。甚至两者都没有，如：

```

void function()
{
    ...
}
  
```

也是合法的。

四、C 语言对大小写字符敏感

与其他语言不一样, C 语言对大小写字符敏感。如定义变量时, 变量名 ABC 和 abc 代表的是两个完全不同的变量名。

C 语言本身没有输入输出语句。输入和输出操作是由库函数 scanf 和 printf 等函数来完成的。

1.2 C 语言历史、特点与发展

1.2.1 C 语言的历史

C 语言是介于汇编语言和高级语言之间的语言, 是集汇编语言和高级语言的优点于一身的程序设计语言。

1972 年, C 语言在美国贝尔实验室里问世, 其发展历程如下:

- ALGOL 60。由 ACM (Association for Computing Machinery) 国际计算机组织委员会于 19 世纪 60 年代早期设计。

- CPL (Combined Programming Language 混合编程语言)。由剑桥大学和伦敦大学于 1963 年开发而成。

- BCPL (Basic Combined Programming Language 基础混合编程语言)。由剑桥大学的 Martin Richards 于 1967 年发明。

- B。由贝尔实验室的 Ken Thompson 于 1970 年发明

- C。由贝尔实验室的 Dennis Ritchie 于 1972 年发明。

1.2.2 C 语言的特点

(1) 优点

简洁、紧凑, 使用方便、灵活, 易于学习和应用。仅有 32 个关键字, 9 种控制语句, 程序的书写形式也很自由。

C 语言是面向结构化程序设计的语言, 具有结构化的控制语句, 以函数作为程序模块以实现程序的模块化。

C 语言允许直接对位、字节和地址进行操作, 能实现汇编语言的大部分功能。因此 C 语言即具有高级语言特点又具有低级语言的功能, 所以人们把 C 语言称为是中级语言。

数据类型丰富。C 语言除具有基本数据类型整形 (int)、实型 (float 和 double)、字符型 (char) 外, 还有各种构造类型。利用这些数据类型可以实现复杂的数据结构, 如堆栈、队列、链表等。

C 语言生成的目标代码质量高, 程序执行效率高。

(2) 弱点

语法限制不严格, 这使得编程者无法过多地依赖 C 编译程序去查错。

缺少实时检查功能, 如数组越界等。

1.2.3 C 语言的发展

C 语言从其诞生到现在取得了长足的发展。先后有不同版本的 C 语言开发工具出现, 如

Quick C, Microsoft C, Turbo C, MS-C, Visual C 等。C 语言常被用来编写应用程序或操作系统, 如比较著名的 Unix、Linux、Windows 操作系统, 其内核就是用 C 语言编写的。

归纳起来它的发展可以表现为以下两方面

(1) 以 C 语言为依托, 各种不同版本的程序设计语言的诞生和发展。如 20 世纪 90 年代风靡全球的 C++ 语言就是在 C 语言基础上发展起来的。C++ 语言正如其名字一样, 它是 C 语言的延伸和拓展, 学会了 C 语言, 对学习 C++ 语言有很大帮助。另外, 像目前比较流行的 Java 语言、C# 语言等都是借鉴 C 语言诞生和发展的。

(2) 作为语言本身, C 语言仍然在被大量使用。如单片机和嵌入式系统行业, 被认为是 C 语言的黄金行业。C 语言因为可以和汇编语言结合来控制硬件, 在这个行业里广受欢迎。

今天, 越来越多的人在学习 C 语言, 使用 C 语言, 用 C 语言开发各个领域的软件。C 语言是目前世界上流行时间最长、范围最广的程序设计语言之一。

1.3 C 语言开发环境介绍

1.3.1 C 语言开发调试步骤

C 语言程序上机调试步骤如图 1-1 所示。

(1) 编辑。此阶段是程序录入阶段, 在程序编辑工具中将要录入的 C 程序输入。如果将输入内容保存为磁盘文件, 其扩展名为 “.c”, 称其为源程序文件。

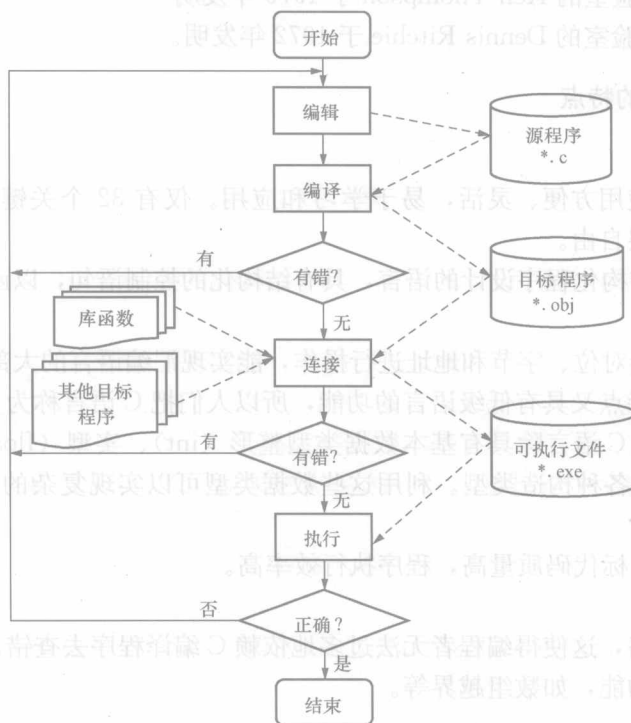


图 1-1 C 语言程序上机调试步骤

(2) 编译。此阶段将启动系统编译器编译源程序文件中的各行语句，如果有语法错误将会显示错误信息，提示用户改正；如果没有错误则会生成扩展名为“.obj”的磁盘文件，称其为目标文件。

(3) 连接。此阶段将启动系统连接器将已生成的目标文件与程序中调用的库函数对应的目标文件以及其他目标文件进行连接，即联编。如果有连接错误，将显示错误类型，提示用户改正；如果没有，则会生成磁盘文件“.exe”，称其为可执行文件。这个文件就是编写C语言程序想最终得到的结果，是一切工作的收获。此文件可以脱离开发环境独立运行。

(4) 执行。得到可执行文件后，还要观察程序实际执行的过程和结果，看是不是按照事先设计的步骤正确运行，如果有误，则出现了逻辑错误，应重新修改程序，直到正确为止。

1.3.2 C程序上机环境介绍

本书采用 Turbo C 2.0 软件作为开发工具。以下所述环境以及书中所有程序都是针对此环境的。

一、Turbo C 2.0 集成开发环境介绍

启动 Turbo C 2.0 后，会出现如图 1-2 所示界面。

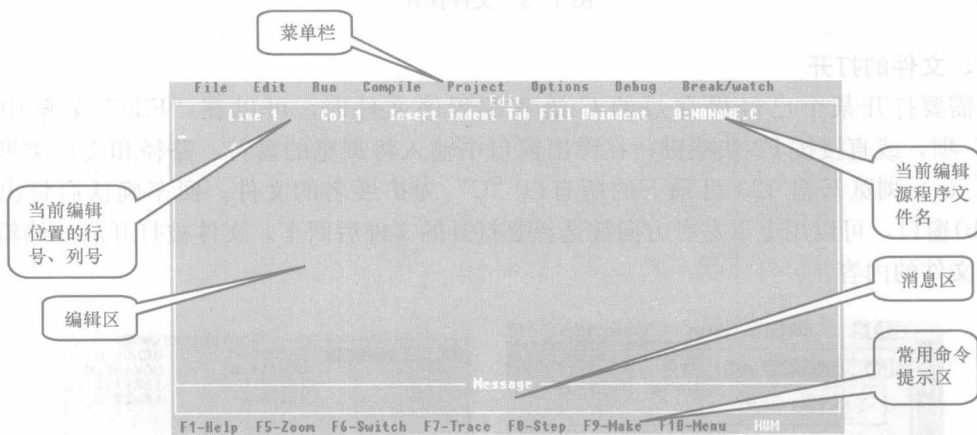


图 1-2 Turbo C 2.0 启动界面

(1) 菜单栏包含：File、Edit、Run、Compile、Project、Options、Debug、Break/watch 一共 8 栏菜单。分别用于：文件操作、编辑、运行、编译、项目管理、选项、调试、中断/观察等功能。每栏菜单首字母红色高亮表示可以通过 Alt 和本字母组成快捷键激活本菜单，如“Alt+F”可以激活 File 菜单。

(2) 当前编辑位置的行号和列号会在 Line 和 Col 后显示，用于标明当前程序输入位置。

(3) 编辑区是用来输入程序的区域，C 程序会以黄色显示。

(4) 当前正在编辑的 C 源程序名会在编辑区的右上角显示。如果显示的是“NONAME.C”，表示目前还没有为本源程序命名存盘。

(5) 消息区用于显示编译和连接时出现的错误信息。

(6) 常用命令提示区用于显示一些常用加速键的提示信息。