

高职高专计算机教育规划教材

C语言程序设计 实例教程

潘志安 朱运乔 余小燕 主编

高职高专计算机教育规划教材

C 语言程序设计实例教程

潘志安 朱运乔 余小燕 主 编
冯 毅 沈小波 李 岚 副主编
袁 瑛 魏 华 沈 平 参 编

内 容 简 介

C 语言从诞生以来就显示了很强的生命力,不仅可以利用它编写应用软件,而且特别适合编写系统软件(如系统内核、驱动程序、嵌入式开发)。即使在.NET、Java 等高级语言流行的今天,C 语言作为训练学习者程序设计的基本思想也是大有裨益的。

本书内容共分 12 章:C 语言概论,数据类型、变量和常量,运算符和表达式,标准输入/输出,分支结构,循环结构,数组,函数,编译预处理,指针,结构体、共用体及枚举类型,文件 I/O。

本书是作者根据多年的教学经验编写而成,每一章均包括必须掌握的基本知识与技能、扩展知识与技能,特别适合根据学习者的实际状况进行分层、分级教学。

本书适合作为高职高专院校计算机类、电子信息类专业的教材或教学参考书,也可作为软件设计与开发人员的培训教材或自学参考书。

图书在版编目(CIP)数据

C 语言程序设计实例教程 / 潘志安, 朱运乔, 余小燕

主编. — 北京: 中国铁道出版社, 2012.7

高职高专计算机教育规划教材

ISBN 978-7-113-14622-1

I. ①C… II. ①潘… ②朱… ③余… III. ①

C 语言—程序设计—高等职业教育—教材 IV. ①TP312

中国版本图书馆 CIP 数据核字(2012)第 087922 号

书 名: C 语言程序设计实例教程

作 者: 潘志安 朱运乔 余小燕 主编

策 划: 翟玉峰

读者热线: 400-668-0820

责任编辑: 翟玉峰

编辑助理: 何 佳

封面设计: 付 巍

封面制作: 刘 颖

责任印制: 李 佳

出版发行: 中国铁道出版社(100054, 北京市西城区右安门西街 8 号)

网 址: <http://www.51eds.com>

印 刷: 中国铁道出版社印刷厂

版 次: 2012 年 7 月第 1 版 2012 年 7 月第 1 次印刷

开 本: 787mm×1092mm 1/16 印张: 16 字数: 385 千

印 数: 1~3 000 册

书 号: ISBN 978-7-113-14622-1

定 价: 30.00 元

版权所有 侵权必究

凡购买铁道版图书,如有印制质量问题,请与本社教材图书营销部联系调换。电话:(010) 63550836

打击盗版举报电话:(010) 63549504

在当今社会发展进程中,人才的生态平衡非常重要。特别是随着我国高等教育大众化时代的来临,高职高专学生的整体状况与以往有了很大的不同,对现代高职高专学生的教育,从教育理念、教学内容、教学方法等方面都需要进行变革,教材也需要变革。编者根据多年的教学经验,本着体系恰当、内容实用、难度合适、循序渐进、通俗易懂、便于理解的原则,编写了本书。

本书具有下列特点:

(1) 体系恰当,适合分层分级教学。本书在内容架构上,包含“基本知识与技能”、“扩展知识与技能”,前者是基础,要求所有学生掌握;后者是提高,在基础知识和技能掌握较好的情况下深入学习。

(2) 教学内容精练。本书根据高职高专学生的特点和课时安排,在内容齐全的情况下,对例题进行了合理精简,做到重点突出,易于理解。

(3) 实用性强,较好地把握了应试与应用的关系。本课程的学习目的,主要是为后续的 C#、Java 语言的学习奠定基本编程思想,同时也将通过计算机二级(C 语言)等级考试作为目标之一,配备了适量的模拟习题和实例。学完本教材,将为后续 C#、Java 语言以及数据结构等课程的学习打下良好基础,而且具备通过全国计算机等级考试二级(C 语言)的能力。

(4) 强调学习方法和策略。C 语言的学习,要能循序渐进地达到三个层次:一是要能顺利排除程序中的语法错误;二是要能看懂别人编写的程序,理解其编程思想和算法,以及如何实现算法,并能逐步学会排除程序中的逻辑错误;三是要能自己编写程序来解决实际问题,即具备分析问题和解决问题的能力。因此,本书在“典型案例”中,会给出算法(即编程思想),帮助学习者学会如何分析问题和解决问题,起到举一反三的作用。

全书共分 12 章:第 1 章是 C 语言概论,第 2 章介绍数据类型、变量和常量,第 3 章介绍运算符和表达式,第 4 章介绍标准输入/输出,第 5、6 章分别介绍分支结构和循环结构,第 7 章介绍数组,第 8 章介绍函数,第 9 章介绍编译预处理,第 10 章介绍指针,第 11 章主要介绍结构体、共用体及枚举类型,第 12 章介绍文件 I/O。

本书内容翔实,层次分明,结构紧凑,叙述深入浅出、通俗易懂,适合作为高职高专院校计算机类、电子信息类专业的教材或教学参考书,也可作为软件设计与开发人员的培训教材或自学参考书。每章后面均附有适量习题,既有利于学生巩固和提高所学知识,又方便 C 语言二级考试参考。

本书由潘志安、朱运乔、余小燕(江苏畜牧兽医职业技术学院)任主编,冯毅(天津商务职业学院)、沈小波、李岚任副主编,袁瑛、魏华、沈平参与了编写工作。在此向所有在本书编写和出版过程中提供了无私帮助的人士表示衷心感谢!

另外,由于编写时间仓促、编者水平有限,书中难免存在疏漏、不妥之处,恳请专家和广大读者批评指正,并多提宝贵意见。同时期待专家和读者能及时与编者联系,以便不断对本书进行修正和完善(E-mail: xgzypza@163.com)。

编者

2012 年 5 月

第 1 章 C 语言概论	1
1.1 基本知识与技能	1
1.1.1 概述	1
1.1.2 C 语言源程序的基本结构	2
1.1.3 C 语言程序的执行过程	5
1.1.4 Turbo C 2.0 语言集成开发环境	6
1.2 知识与技能扩展	9
1.2.1 在 Visual C++ 6.0 开发环境下运行 C 程序	9
1.2.2 算法及其表示	11
小结	12
习题	13
第 2 章 数据类型、变量和常量	14
2.1 引例分析	14
2.2 基本知识与技能	15
2.2.1 数据类型	15
2.2.2 常量及变量	15
2.2.3 整型数据	18
2.2.4 实型数据	21
2.2.5 字符型数据	23
2.3 知识与技能扩展	25
2.3.1 常量的数据类型及后缀表示	25
2.3.2 赋值转换	26
2.4 典型案例	28
小结	30
习题	30
第 3 章 运算符和表达式	31
3.1 引例分析	31
3.2 基本知识与技能	32
3.2.1 算术运算符及算术表达式	32
3.2.2 赋值运算符及赋值运算表达式	35
3.2.3 自增和自减运算及其表达式	36
3.2.4 关系运算和逻辑运算	38
3.3 知识与技能扩展	40
3.3.1 位运算	40
3.3.2 其他运算符及其表达式	43
3.4 典型案例	44
小结	45
习题	45

第4章 标准输入/输出	47
4.1 引例分析	47
4.2 基本知识与技能	48
4.2.1 关于数据输入/输出	48
4.2.2 格式化输出——printf()函数	48
4.2.3 格式化输入——scanf()函数	50
4.3 知识与技能扩展	52
4.3.1 字符数据输出——putchar()函数	52
4.3.2 字符数据输入——getchar()函数	52
4.4 典型案例	53
小结	54
习题	54
第5章 分支结构	58
5.1 引例分析	58
5.2 基本知识与技能	59
5.2.1 程序的基本结构	59
5.2.2 C语言的语句	59
5.2.3 分支结构和if语句	60
5.2.4 switch多路分支语句	64
5.3 知识与技能扩展	67
5.3.1 语句标号	67
5.3.2 goto语句	67
5.4 典型案例	67
小结	70
习题	71
第6章 循环结构	74
6.1 引例分析	74
6.2 基本知识与技能	75
6.2.1 while语句	75
6.2.2 for语句	77
6.2.3 do...while语句	79
6.3 知识与技能扩展	80
6.3.1 循环结构的嵌套	80
6.3.2 循环结构中的break和continue	81
6.4 典型案例	83
小结	90
习题	90
第7章 数组	94
7.1 引例分析	94
7.2 基本知识与技能	95
7.2.1 一维数组	95
7.2.2 二维数组	97
7.2.3 字符数组及字符串	100

7.3 知识与技能扩展	103
7.3.1 常用字符串处理函数	103
7.3.2 多维数组	106
7.4 典型案例	106
小结	113
习题	113
第 8 章 函数	117
8.1 引例分析	117
8.2 基本知识与技能	118
8.2.1 C 语言函数的分类	119
8.2.2 函数的定义与函数说明	119
8.2.3 函数的调用	124
8.2.4 函数的嵌套调用与递归调用	126
8.2.5 数组作为函数参数	130
8.3 知识与技能扩展	133
8.3.1 变量的作用范围	133
8.3.2 变量的动态存储与静态存储	136
8.3.3 内部函数和外部函数	139
8.3.4 多个源程序文件的编译和连接	141
8.4 典型案例	141
小结	144
习题	145
第 9 章 编译预处理	148
9.1 引例分析	148
9.2 基本知识与技能	149
9.2.1 宏定义	149
9.2.2 文件包含	152
9.3 知识与技能扩展	153
9.3.1 条件编译	153
9.3.2 typedef 关键字与宏定义的区别	155
9.4 典型案例	157
小结	159
习题	159
第 10 章 指针	163
10.1 引例分析	163
10.2 基本知识与技能	164
10.2.1 变量的指针和指针变量	164
10.2.2 指针的运算	165
10.2.3 指针作为函数参数	167
10.2.4 指针与一维数组	168
10.2.5 指针与二维数组	171
10.2.6 字符串的指针和指向字符串的指针变量	175

10.3 知识与技能扩展.....	178
10.3.1 指针数组.....	178
10.3.2 二级指针.....	179
10.3.3 带形式参数的 main() 函数.....	181
10.3.4 指向函数的指针.....	182
10.3.5 指针型函数.....	183
10.4 典型案例.....	184
小结.....	187
习题.....	187
第 11 章 结构体、共用体及枚举类型.....	192
11.1 引例分析.....	192
11.2 基本知识与技能.....	193
11.2.1 结构体.....	193
11.2.2 结构体型数组.....	197
11.2.3 结构体型指针.....	198
11.3 知识与技能扩展.....	200
11.3.1 动态存储分配.....	200
11.3.2 链表.....	202
11.3.3 共用体.....	204
11.3.4 枚举类型.....	206
11.4 典型案例.....	208
小结.....	211
习题.....	212
第 12 章 文件 I/O.....	218
12.1 引例分析.....	218
12.2 基本知识与技能.....	219
12.2.1 文件概述.....	219
12.2.2 文件的打开与关闭.....	221
12.2.3 文件的读/写.....	222
12.3 知识与技能扩展.....	228
12.3.1 文件位置指针操作函数.....	228
12.3.2 文件的随机读/写.....	229
12.4 典型案例.....	230
小结.....	233
习题.....	234
附录 A C 语言的关键字及其用途.....	236
附录 B 常用字符的 ASCII 表.....	237
附录 C 扩充字符 ASCII 表.....	238
附录 D 运算符和结合性.....	239
附录 E C 库函数.....	241
参考文献.....	247

第 1 章 C 语言概论

本章目标

本章首先介绍了 C 语言的起源与发展,阐述了 C 语言的特点,然后重点介绍了 C 语言程序的结构组成和上机过程及相关知识。通过本章的学习,读者应重点掌握以下内容:

- C 语言基本结构。
- C 语言程序执行过程。
- C 语言程序的集成开发环境。

1.1 基本知识与技能

1.1.1 概述

1. C 语言的起源与发展

1972 年美国的 Dennis Ritchie 设计发明了 C 语言,并首次在 UNIX 操作系统的 DEC PDP-11 计算机上使用。它由早期的编程语言 BCPL (Basic Combined Programming Language) 发展演变而来。在 1970 年,AT&T 贝尔实验室的 Ken Thompson 根据 BCPL 语言设计出较先进的并取名为 B 的语言,最后引导了 C 语言的问世。

随着微型计算机的日益普及,出现了许多 C 语言版本。由于没有统一的标准,使得这些 C 语言之间出现了一些不一致的地方。为了改变这种情况,美国国家标准学会 (ANSI) 为 C 语言制定了一套 ANSI 标准,成为现行的 C 语言标准。

早期的 C 语言主要用于 UNIX 系统。由于 C 语言的强大功能和各方面的优点逐渐为人们认识,到了 20 世纪 80 年代, C 语言开始应用于其他操作系统,并很快在各类大、中、小和微型计算机上得到广泛应用,成为当代最优秀的程序设计语言之一。

2. C 语言的特点

C 语言发展如此迅速,而且成为最受欢迎的语言之一,主要因为它具有强大的功能。归纳起来, C 语言具有下列特点:

(1) C 语言是中级语言。它把高级语言的基本结构和语句与低级语言的实用性结合起来,常被称为中级语言。C 语言可以像汇编语言一样对位、字节和地址进行操作,它也可以直接访问内存的物理地址,进行位 (bit) 一级的操作,还可实现对硬件的编程操作,因此 C 语言既可用于系统软件的开发,也适合于应用软件的开发。

(2) C 语言是结构式语言。C 语言是以函数形式提供给用户的,这些函数可方便地调用,

并具有多种循环、条件语句控制程序流向,从而使程序完全结构化。按模块化方式组织程序,层次清晰,易于调试和维护。C 语言的表现能力和处理能力极强。

(3) C 语言功能齐全。C 语言具有丰富的运算符和数据类型,便于实现各类复杂的数据结构,并引入了指针概念,可使程序效率更高。另外,C 语言也具有强大的图形功能,支持多种显示器和驱动器,而且计算功能、逻辑判断功能也比较强大,可以实现决策目的。

(4) C 语言适用范围大。C 语言还有一个突出的优点就是适用于多种操作系统,如 DOS、UNIX,也适用于多种机型,可广泛地移植到各类型计算机上,从而形成了多种版本的 C 语言。

总之,C 语言简洁、紧凑、实用、方便、移植性好、执行效率高、处理能力强、结构化程度高,但对编程人员要求较高,较难掌握,不够安全。

3. C 语言版本

目前最流行的 C 语言有以下几种:

- (1) Microsoft C 或称 MS C。
- (2) Borland Turbo C 或称 Turbo C。
- (3) AT&T C。

这些 C 语言版本不仅实现了 ANSI C 标准,而且在此基础上各自作了一些扩充,使之更加方便、完美。

1.1.2 C 语言源程序的基本结构

1. 范例介绍

为了说明 C 语言源程序结构的特点,先看以下几个程序。这几个程序由简到难,体现了 C 语言源程序在组成结构上的特点。虽然有关内容还未介绍,但可从这些程序中了解到组成一个 C 源程序的基本部分。

【例 1.1】显示指定的内容。

源程序:

```
void main()
{
    printf(" This is the first program \n");
}
```

程序说明:

(1) main()是主函数的函数名,表示这是一个主函数。每一个 C 源程序都必须有且只能有一个主函数(main()函数)。

void 表示 main()函数无返回值(这一点将在第 8 章详细讲述)。

(2) printf()是 C 语言函数库提供的标准函数,该函数的功能是把要输出的内容送到显示器显示,可在程序中直接调用。

【例 1.2】求输入数的正弦值。

源程序:

```
#include <math.h>    /*include 称为文件包含命令,也把扩展名为.h 的文件称为头文件*/
#include <stdio.h>
void main()          /*定义主函数*/
{                    /*主函数开始*/
    double x,s;      /*定义两个实数变量,以被后面程序使用*/
```

```

printf("input number:\n");           /*显示提示信息*/
scanf("%lf",&x);                     /*从键盘获得一个实数 x*/
s=sin(x);                             /*求 x 的正弦, 并把它赋给变量 s*/
printf("sine of %lf is %lf\n",x,s);   /*显示程序运算结果*/
}                                     /*主函数结束*/

```

程序说明:

(1) 在 void main()之前的两行称为预处理命令(详见后面)。这里的 include 称为文件包含命令,其意义是把尖括号<>内指定的文件包含到本程序中,成为本程序的一部分。被包含的文件通常是由系统提供的,其扩展名为.h,因此也称为头文件或首部文件。

(2) C 语言的头文件中包括了各个标准库函数的函数原型说明。在本例中,使用了三个库函数:输入函数 scanf(), 正弦函数 sin(), 输出函数 printf()。sin()函数是数学函数,其头文件为 math.h(),因此在程序的主函数前用 include 命令包含了 math.h。scanf()和 printf()是标准输入/输出函数,其头文件为 stdio.h,在主函数前也用 include 命令包含了 stdio.h。

(3) C 语言规定对 scanf()和 printf()这两个函数可以省去对其头文件的包含命令,所以在本例中可以删去第二行的包含命令#include。

(4) 在例题的主函数体中又分为两部分,一部分为说明部分,另一部分为执行部分。说明是指变量的类型说明。C 语言规定,源程序中所有用到的变量都必须先说明,后使用,否则将会出错。说明部分是 C 源程序结构中很重要的组成部分。本例中使用了两个变量 x、s,用来表示输入的自变量和 sin()函数值。

(5) 说明部分的后四行为执行部分或称为执行语句部分,用以完成程序的功能。

① 执行部分的第一行是输出语句,调用 printf()函数在显示器上输出提示字符串,要求操作人员输入自变量 x 的值。

② 第二行为输入语句,调用 scanf()函数,接收键盘上输入的数并存入变量 x 中。

③ 第三行是调用 sin()函数,并把函数值送到变量 s 中。

④ 第四行是用 printf()函数,输出变量 s 的值,即 x 的正弦值。

(6) 语句 printf("sine of %lf is %lf\n",x,s);其中%lf 为格式字符,表示按双精度浮点数处理。它在格式串中出现两次,对应 x 和 s 两个变量。其余字符为非格式字符,则照原样输出在屏幕上。

2. C 语言程序结构特点

一个完整的 C 语言程序应符合以下几点:

(1) C 语言程序是以函数为基本单位,整个程序由函数组成。一个较完整的程序大致由包含文件(一组#include <*.h>语句)、用户函数说明部分、全程变量定义、主函数和若干子函数组成。在主函数和子函数中又包括局部变量定义、程序体等,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。函数说明部分包括对函数名、函数类型、形式参数等的定义和说明;函数体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) 语句是由一些基本字符和定义符按照 C 语言的语法规则组成的,每个语句以分号结束。

(4) C 程序的书写格式比较自由。一个语句可写在一行上,也可分写在多行内。一行内可

以写一个语句，也可写多个语句。注释内容可以单独写在一行，也可以写在 C 语句的右面。

(5) 一个 C 语言源程序可以由一个或多个源文件组成。

(6) 一个源程序不论由多少个文件组成，都有一个且只能有一个 `main()` 函数，即主函数。

(7) 注释部分包含在 “/*” 和 “*/” 之间，在编译时它被 Turbo C 编译器忽略。

注意初学者易犯错误：

(1) 遗漏分号：`printf()` 等后忘记加 “;”。

(2) 遗漏花括号或者多用花括号：花括号总是成对出现的。

(3) 忘记使用 & 号：输入时指明地址要加上此符号。

3. 书写程序时应遵循的规则

从书写清晰，便于阅读、理解、维护的角度出发，在书写程序时应遵循以下规则：

(1) 一个说明或一个语句占一行。

(2) 用 {} 括起来的部分通常表示程序的某一层结构。

(3) 低一层次的语句或说明应缩进若干格后书写，使阅读更加清晰，增加程序的可读性。

在编程时应力求遵循这些规则，以养成良好的编程风格。

4. C 语言程序的成分

C 语言程序是由语句组成的，而语句是由词汇构成的，每个词汇由字符构成，即：字符 → 词汇 → 语句 → C 语言程序。

(1) C 语言的字符集。字符是组成语言的最基本的元素。C 语言字符集由字母、数字、空格、标点和特殊字符组成。在字符常量、字符串常量和注释中还可以使用汉字或其他可表示的图形符号。

(2) C 语言词汇。在 C 语言中使用的词汇分为以下几类：标识符、关键字、运算符、分隔符、常量、注释符等。

① 标识符：在程序中使用的变量名、函数名、标号等统称为标识符。除库函数的函数名由系统定义外，其余都由用户自定义。C 语言规定，标识符只能是由字母 (A~Z, a~z)、数字 (0~9)、下画线组成的字符串，并且第一个字符必须是字母或下画线。

② 关键字：关键字是由 C 语言规定的具有特定意义的字符串。用户定义的标识符不应与关键字相同。C 语言的关键字分为以下几类：

- 类型说明符。用于定义、说明变量、函数或其他数据结构的类型，如前面例题中用到的 `int`、`double` 等。
- 语句定义符。用于表示一个语句的功能。`if else` 就是条件语句的语句定义符。
- 预处理命令字。用于表示一个预处理命令，如前面各例中用到的 `include`。

③ 运算符：C 语言中含有相当丰富的运算符。运算符、变量与函数一起组成表达式，表示各种运算功能。运算符由一个或多个字符组成。

④ 分隔符：在 C 语言中采用的分隔符有逗号和空格两种。逗号主要用于类型说明和函数参数表中，分隔各个变量。空格多用于语句各单词之间，作间隔符。在关键字、标识符之间必须要有一个以上的空格符作间隔，否则将会出现语法错误，例如把 `int a;` 写成 `inta;`，C 编译器会把 `inta` 当成一个标识符处理，其结果必然出错。

⑤ 常量：C 语言中使用的常量可分为字面常量、符号常量。

⑥ 注释符：C 语言的注释符是以 “/*” 开头并以 “*/” 结尾的串。在 “/*” 和 “*/” 之间

的即为注释。编译程序时，不对注释作任何处理。注释可出现在程序中的任何位置，用来向用户提示或解释程序的意义。在调试程序中对暂不使用的语句也可用注释符括起来，使编译跳过不作处理，待调试结束后再去掉注释符。

(3) 程序语句。C 程序是由若干条语句组成，语句是程序的基本书写单位和执行单位。其语句可分为以下五种：表达式语句、声明语句、空语句、复合语句及流程控制语句。

1.1.3 C 语言程序的执行过程

1. 源程序、目标程序、可执行程序的概念

程序：为了使计算机能按照人们的意志工作，就要根据问题的要求编写相应的程序。程序是一组计算机可以识别和执行的指令，每一条指令使计算机执行特定的操作。

源程序：程序可以用高级语言或汇编语言编写，用高级语言或汇编语言编写的程序称为源程序。C 语言源程序的扩展名为“.c”。源程序不能直接在计算机上执行，需要用编译程序将源程序翻译为二进制形式的代码。

目标程序：源程序经过编译程序翻译所得到的二进制代码称为目标程序。目标程序的扩展名为“.obj”。目标代码尽管已经是机器指令，但是还不能运行，因为目标程序还没有解决函数调用问题，需要将各个目标程序与库函数连接，才能形成完整的可执行程序。

可执行程序：目标程序与库函数连接，形成完整的可在操作系统下独立执行的程序称为可执行程序。可执行程序的扩展名为“.exe”（在 DOS/Windows 环境下）。

2. C 语言程序的上机步骤

输入与编辑源程序→编译源程序→产生目标代码→连接各个目标代码、库函数→产生可执行程序→运行程序，如图 1-1 所示。

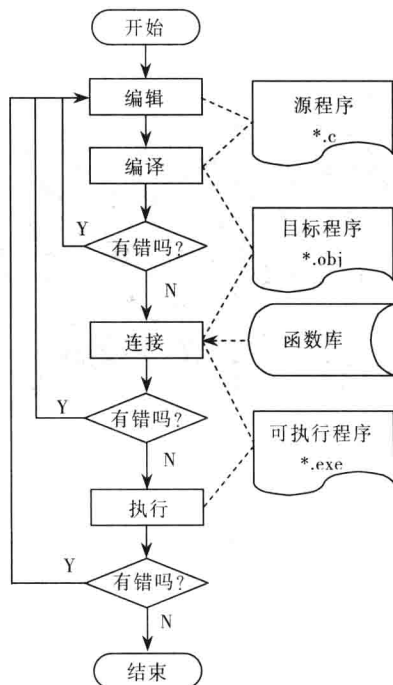


图 1-1 C 语言程序的上机步骤

1.1.4 Turbo C 2.0 语言集成开发环境

针对不同的平台有相应的集成开发环境：Turbo C 作为在 DOS 和 Windows 上学习 C 语言的常用开发工具，适用于初学者；Visual Studio 中 Visual C++ 是以 Windows 平台开发的一个主流的可视化 C 语言开发环境，现在已经升级到.NET 版本；GCC 是 UNIX 平台上主要使用的 C 语言开发工具，嵌入式系统的开发常用 GCC 的交叉编译器来完成。本书主要以 Turbo C 2.0 集成开发环境来作为 C 语言开发工具。

1. Turbo C 2.0 简介和启动

Turbo C 是美国 Borland 公司的产品，Borland 公司是一家专门从事软件开发、研制的公司。该公司相继推出了一套 Turbo 系列软件，如 Turbo BASIC、Turbo Pascal、Turbo Prolog 等，这些软件很受用户欢迎。该公司在 1987 年首次推出 TurboC 1.0 产品，其中使用了全然一新的集成开发环境，即使用一系列下拉式菜单，将文本编辑、程序编译、连接以及程序运行一体化，大大方便了程序的开发。1988 年，Borland 公司又推出 Turbo C 1.5 版本，增加了图形库和文本窗口函数库等，而 Turbo C 2.0 则是该公司 1989 年出版的。Turbo C 2.0 在原来集成开发环境的基础上增加了查错功能，并可以在 Tiny 模式下直接生成.com（数据、代码、堆栈处在同一 64 KB 内存中）文件，还可对数学协处理器（支持 8087/80287/80387 等）进行仿真。

Borland 公司后来又推出了面向对象的程序软件包 Turbo C++，它继承发展 Turbo C 2.0 的集成开发环境，并包含了面向对象的基本思想和设计方法。

Turbo C 2.0 是 DOS 操作系统支持下的软件，在 Windows 环境下，可以在 DOS 窗口下运行。

假定在 D 盘的 TC 子目录下安装 Turbo C 2.0 系统，在 TC 下的 LIB 子目录中存放库文件，在 ITC 下的 NCLUDE 子目录中存放所有头文件。

在 Windows 环境下，也可以选择“运行”命令，然后输入 d:\tc\tc 即可，也可以在 tc 文件夹找到 tc.exe 文件，然后用鼠标双击该文件也可进入 Turbo C 2.0 集成开发环境，如图 1-2 所示。



图 1-2 Turbo C 2.0 主窗口

2. Turbo C 集成开发环境的常见操作

(1) 菜单的使用（选中、打开或执行）：Alt+菜单选项的快捷字母激活某菜单，利用方向键进行菜单项的选择，也可按【F10】功能键，移动方向键选中所需项目。选中项目后按【Enter】键即可打开或执行该项目。

(2) 源程序的输入、修改和保存：利用功能键和编辑键对源程序进行编辑，利用【F2】键和相应的菜单项来保存文件。

- (3) 程序编译并执行:【Ctrl + F9】组合键。
- (4) 程序执行结果查看:【Alt + F5】组合键。
- (5) 退出 TC 开发环境:【Alt + X】组合键。

3. 开发由单一文件构成的 C 程序

以例 1.1 的源程序为例来展示一般 C 语言程序,要经过编辑源程序文件(.c)、编译生成目标文件(.obj)、连接生成可执行文件(.exe)和执行四个步骤。

进入 Turbo C 主窗口后,按【F3】键,即可在随之出现的框中输入文件名,文件名可以带“.c”,也可以不带(此时系统会自动加上)。输入文件名后,按【Enter】键,即可将文件调入,如果文件不存在,就建立一个新文件(也可用下面例子中的方法输入文件名)。系统随之进入编辑状态,就可以输入或修改源程序了,源程序输入或修改完毕以后,按【Ctrl+F9】组合键,则立即进行编译、连接和执行,这三项工作是连续完成的。具体的操作步骤如下:

(1) 启动 Turbo C。双击执行 TC.EXE 文件,使系统进入 Turbo C 集成开发环境(见图 1-2),并建立一个名为 fg.c 的文件。这时,系统进入 Turbo C 编辑环境。

(2) 处理前初始化。设定用户文件的存放目录:依次执行主菜单 File 和其对应的 Change dir 子菜单,然后在子目录提示框中输入用户文件的存放目录,如 D:\USER。

设定编译连接和可执行文件的目录(用户的.obj 和.exe 文件的位置):依次执行主菜单 Options 选项和其对应的 Directories 子菜单,然后在 Output directory 选项中输入输出文件的存放目录,如 D:\USER。此时这些文件被放在当前目录。

说明标准包含文件的目录(INCLUDE 的位置):依次执行主菜单 Options 选项和其对应的 Directories 子菜单,然后在 Include directories 选项中输入标准包含文件的存放目录,如 D:\TC\INCLUDE。

说明运行库文件的目录(LIB 的位置):依次执行主菜单 Options 选项和其对应的 Directories 子菜单,然后在 Library directories 选项中输入库文件的存放目录,如 D:\TC\LIB。

说明 Turbo C 文件的目录(Turbo C 系统的位置):依次执行主菜单 Options 选项和其对应的 Directories 子菜单,然后在 Turbo C directory 选项中输入 Turbo C 文件(配置文件.TC)和帮助文件(TCHELP.TCH)的所在目录,如 D:\TC。

初始化是可选操作,一般在第一次编程时进行。前两项初始化根据用户的需要可随意指定路径,但要求对该路径要有读/写权限,后三项初始化是根据 Turbo C 所安装的位置所作的说明,所以对某一系统而言它们的值是一定的。

(3) 通过键盘输入程序。

```
void main()
{
    printf(" This is the first program \n");
}
```

则该程序进入计算机存储器。

(4) 程序存盘。为防止意外事故丢失程序,最好将输入的程序存储到磁盘中。在编辑窗口下,可直接按【F2】键或按【F10】键,再按【F】键进入 File 菜单,最后按【S】或【W】键将文件存盘。存盘时屏幕最底行会显示:

```
saving edit file
```

(5) 编译一个程序。对源程序进行编译有两种方法:直接按【Alt+F9】组合键即可;按【F10】

键返回主菜单, 选择 Compile 命令, 屏幕显示 Compile 下拉菜单, 从下拉菜单中选择 Compile to .OBJ 命令, 按【Enter】键。

进入编译状态后, 屏幕会出现一个编译窗口, 几秒钟后屏幕显示一闪烁信息:

Success: press any key

表示编译成功。此时可按任意键, 编译窗口消失, 光标返回主菜单。

如果编译时产生警告 Warning 或出错 Error 信息, 这些具体错误信息会显示在屏幕下部的信息窗中, 必须纠正这些错误。对源程序进行修改, 重新进行编译。

(6) 运行程序。源程序经编译无误后, 可以投入运行。具体操作如下:

① 如果当前还在编辑状态, 可按【Alt+R】组合键, 再选择 RUN 命令即可。

② 按【Ctrl+F9】组合键。

程序投入运行时, 屏幕会出现一个连接窗口, 显示 Turbo C 正在连接和程序所需的库函数。按【Alt+F5】组合键查看, 此时屏幕被清除, 在顶部显示“This is the first program”字样。再按任意键, 即可回到 TC 主屏幕。

4. 开发运行由多文件构成的 C 程序

程序由 master.c 文件和 max.c 文件组成, 主函数放在 master.c 文件中, 另一函数放在 max.c 文件中, 且主函数所在文件中没有包含另一函数所在的文件。

【例 1.3】由多文件构成的 C 程序。

```
/*master.c*/
int max(int a,int b);           /*函数说明*/
void main()                     /*主函数*/
{
    int x,y,z;                  /*变量说明*/
    printf("input two numbers:\n");
    scanf("%d%d",&x,&y);        /*输入 x,y 值*/
    z=max(x,y);                 /*调用 max() 函数*/
    printf("maxmum=%d",z);      /*输出*/
}
/*max.c*/
int max(int a,int b)            /*定义 max() 函数*/
{
    if(a>b) return a;
    else return b;              /*把结果返回主调函数*/
}
```

(1) 启动 Turbo C。

(2) 输入每个源文件 master.c 和 max.c。

(3) 构造 Project 文件。

在编辑状态下, 输入以下内容:

D:\USER\master.c

D:\USER\max.c

(4) 选择需编译、连接的 .PRJ 文件。

选择 Project 主菜单下的 Project name 命令, 输入 Project 文件名 (如 mypro.prj)。

(5) 编译、连接并运行程序, 与单个文件的方法相同。

1.2 知识与技能扩展

1.2.1 在 Visual C++ 6.0 开发环境下运行 C 程序

Visual C++ 6.0 是 Microsoft 公司推出的使用非常广泛的可视化编程环境,主要用以编写 Windows 窗口应用程序。它提供了强大的开发能力。无论是简单的 Windows 程序、绘图程序,还是最新的 Internet 应用程序、复杂的 ODBC 数据库应用程序以及 AutoCAD 应用开发程序,Visual C++ 6.0 都能轻松地胜任。本节将利用 Visual C++ 6.0 开发一些小的 C 程序。

在 Visual C++ 6.0 运行 C 程序的步骤如下:

(1) 启动 Visual C++ 6.0: 在 Windows 系统下选择“开始”→“程序”→Microsoft Visual Studio 6.0→Microsoft Visual C++ 6.0 命令。打开 Microsoft Visual C++ 6.0 主窗口,如图 1-3 所示。

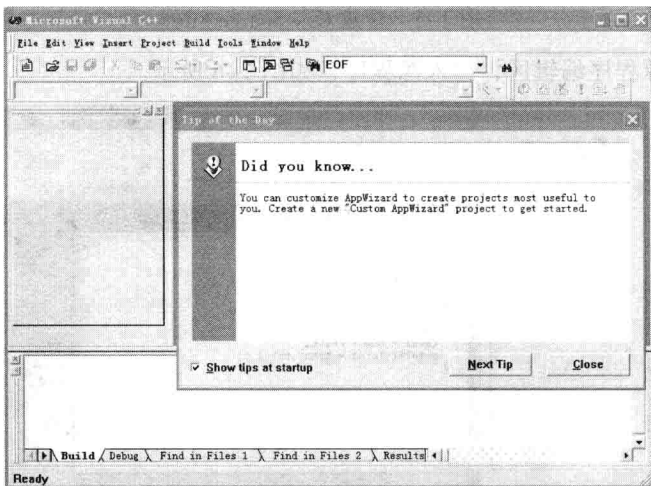


图 1-3 打开 Visual C++ 6.0 主窗口

(2) 在图 1-3 中单击 Close 按钮。

(3) 选择菜单栏中的 File→New 命令,弹出如图 1-4 所示的对话框。

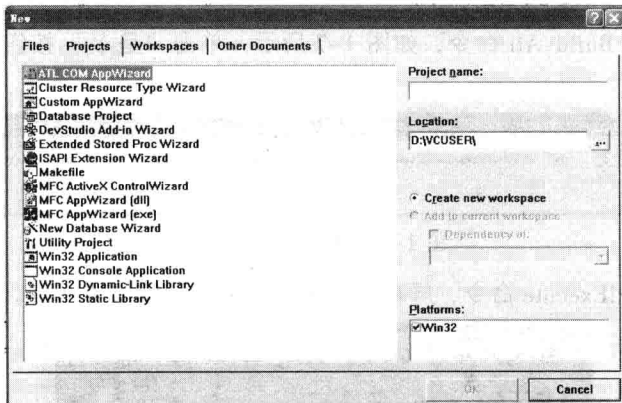


图 1-4 New 对话框