



ZHONGDENGZHIYEAJIAOYU
JISUANJIXILIEJIAOCAI

· 中等职业教育计算机系列教材 ·

C语言程序设计 实用教程

刘体斌 主编

陈昌兵 胡宗祥 谭 剑 副主编



电子科技大学出版社

中等职业教育计算机系列教材

语言程序设计实用教程

刘体斌 主编

陈昌兵 胡宗祥 谭 剑 副主编

电子科技大学出版社

内 容 简 介

本书根据中等职业技术学校计算机教材大纲的要求,结合中等职业技术学校学生的特点,设计的理论知识编写而成。其主要内容有:C语言的基本概念、数据类型、语法及其语句规则、及其调用、指针、文件操作等。另外,结合作者从教的经验体会编写了大量与书中内容紧密联系书中内容浅显易懂、便于接受,并在每章后附有习题和上机实习,供学习者练习、复习和巩固知识。

本书可作为中等职业技术学校教材,还可作为计算机爱好者的自学参考书。

声 明

本书无四川省版权防盗标识,不得销售;版权所有,违者必究,举报有奖。举报电话:(026241146 3201496

中等职业教育计算机系列教材

C语言程序设计实用教程

刘体斌 主编

陈昌兵 胡宗祥 谭 剑 副主编

出 版: 电子科技大学出版社(成都建设北路二段四号 邮编: 610054)
责任编辑: 朱 丹
发 行: 新华书店
印 刷: 西南冶金地质印刷厂
开 本: 787×1092 1/16 印张 14.75 字数 358千字
版 次: 1999年10月第一版
印 次: 1999年10月第一次印刷
书 号: ISBN 7—81065—233—8/TP•130
印 数: 1—4000册
定 价: 17.00元

前 言

C语言是一种通用的程序设计语言，已经被广大的计算机工作者普遍接受和使用。可这样说，没有一个编程者是不懂C语言的。现在许多流行的语言，诸如C++、Java等语言基础都是C语言。另外，C语言具有直接操纵计算机硬件能力的这一特点也增添了C语言的实用性。因此，C语言有必要被广大计算机爱好者所认识、所掌握。

本书面向的对象主要是中等职业技术学校计算机专业的学生，因此在安排教学内容上根据学习者的层次，力求浅显易懂，对某些理论上难以理解的内容没有作深入探讨，避免学习者对所学内容产生畏难情绪。

本教材以美国国家标准C语言为基本内容，结合Turbo C编译系统，全面、系统地介绍语言及其程序设计的方法。全书共分9章。第一章为绪论，介绍C语言的发展、C程序的基本结构及其调试C程序的集成环境——Turbo C 2.0的使用方法。第二章和第三章介绍C语言的基本数据类型、运算符及其各种表达式。第四章介绍C语言提供的各种语句，逐步掌握程序设计方法。第五章介绍C语言的复杂数据类型，如数组、结构体、共用体、枚举和自定义数据类型等。第六章介绍C语言中比较难以理解的指针数据类型，指针这种数据类型是C语言的精华所在，因此，对指针掌握的程度是验证学习者是否学好C语言的标志。第七章介绍模块化程序设计，讲解C语言函数的定义及其调用方法。第八章介绍C程序中变量的不同存储方法及其编译预处理命令的一些用法。第九章介绍C语言关于文件处理的一些库函数的使用方法，并在学习帮助初学者利用这些函数实现一些DOS命令的功能。最后在附录中给出C语言的各种运算符表和常见库函数的用法。

本书由刘体斌编写第一、二、六、七、九章及其附录；陈昌兵编写第四、五章；胡宗洋编写第三章；谭剑编写第八章。全书由刘体斌统稿。电子科技大学出版社朱丹、吴艳玲女士对本书的编写提出了各种有益的建议。另外，邓晖小姐对本书的录入、排版给予了极大的支持，在此表示诚挚的感谢。

由于编写时间较紧，编者水平有限，书中若有不妥之处，请读者提出宝贵意见。

编者

1999年8月于成都

目 录

第一章 绪 论	1
1.1 C语言概述	1
1.1.1 C语言的发展	1
1.1.2 C语言的特点	2
1.2 C语言程序的基本结构	3
1.3 C程序的编辑、编译、链接和运行	5
1.3.1 源程序的编辑	5
1.3.2 源程序的编译	6
1.3.3 目标文件的链接	6
1.3.4 运行	6
1.4 Turbo C 2.0	6
1.4.1 安装方法和主要文件	6
1.4.2 Turbo C 2.0集成环境的使用	8
1.5 习 题 一	12
1.6 上机实习一	13
第二章 基本数据类型、数据的输入/输出	15
2.1 基本数据类型	15
2.1.1 常量和变量	15
2.1.2 基本数据类型	16
2.2 整型数据的表示方法	17
2.2.1 整型常量	17
2.2.2 整型变量	17
2.2.3 整型常量的类型	20
2.3 实型数据的表示方法	20
2.3.1 实型常量	20
2.3.2 实型变量	20
2.4 字符型数据的表示方法	21
2.4.1 字符常量	22

2.4.2	字符变量.....	23
2.4.3	字符串.....	24
2.5	输出函数.....	24
2.5.1	printf()函数.....	25
2.5.2	putchar()字符输出函数.....	29
2.6	输入函数.....	30
2.6.1	scanf()函数.....	30
2.6.2	getchar()字符输入函数.....	34
2.7	习 题 二.....	36
2.8	上机实习二.....	38
第三章 基本运算符及 表达式.....		40
3.1	表 达 式.....	40
3.1.1	表达式概念.....	40
3.1.2	表达式的值.....	41
3.1.3	语句和表达式的关系.....	41
3.2	算术运算符及其算术表达式.....	41
3.3	赋值运算符及其赋值表达式.....	44
3.4	关系运算符及其关系表达式.....	45
3.4.1	关系运算符.....	45
3.4.2	关系表达式.....	46
3.5	逻辑运算符及其逻辑表达式.....	46
3.5.1	逻辑运算符.....	46
3.5.2	按位逻辑运算符.....	48
3.6	其他运算符的运用.....	50
3.6.1	自增运算符和自减运算符.....	50
3.6.2	复合赋值运算符.....	52
3.6.3	逗号运算符.....	52
3.6.4	条件运算符.....	53
3.7	习 题 三.....	54
3.8	上机实习三.....	57
第四章 控 制 语 句.....		58
4.1	程序基本控制结构.....	58
4.1.1	顺序结构.....	58
4.1.2	分支结构.....	59
4.1.3	循环结构.....	60
4.2	复 合 语 句.....	60
4.3	if条件分支语句.....	61

4.3.1	一般格式	62
4.3.2	嵌套的if语句	63
4.4	switch多分支语句	65
4.5	循环语句	68
4.5.1	while循环语句	69
4.5.2	do~while循环语句	70
4.5.3	for循环语句	72
4.5.4	循环语句嵌套	74
4.6	break、continue、goto语句	75
4.6.1	break语句	75
4.6.2	continue语句	76
4.6.3	标号与goto语句	77
4.7	习 题 四	78
4.8	上机实习四	83
第五章 数组及其他数据类型		84
5.1	一 维 数 组	84
5.1.1	一维数组的定义	84
5.1.2	一维数组的初始化	85
5.1.3	一维数组元素的引用	86
5.1.4	应用举例	86
5.2	二 维 数 组	89
5.2.1	二维数组的定义	89
5.2.2	二维数组的初始化	90
5.2.3	二维数组元素的引用	91
5.2.4	多维数组	91
5.2.5	应用举例	92
5.3	字 符 数 组	92
5.3.1	字符数组的定义和初始化	93
5.3.2	字符数组的输入与输出	94
5.3.3	字符串处理函数	95
5.3.4	应用举例	98
5.4	结 构 体	98
5.4.1	结构体类型和结构体类型变量的定义	100
5.4.2	结构体类型变量的使用	101
5.4.3	结构体变量的初始化	102
5.4.4	结构体数组	102
5.5	共 用 体	103
5.5.1	共用体类型及共用体类型变量的定义	103

5.5.2	共用体类型变量的使用.....	104
5.6	枚举类型.....	105
5.6.1	枚举类型及枚举类型变量的定义.....	105
5.6.2	枚举类型变量的使用.....	106
5.7	typedef自定义类型.....	106
5.8	习 题 五.....	108
5.9	上机实习五.....	112
第六章	指 针	114
6.1	指针与指针变量.....	114
6.1.1	地址与指针.....	114
6.1.2	变量的指针.....	115
6.1.3	指针变量的定义及赋值.....	116
6.2	指 针 运 算.....	116
6.2.1	取地址运算“&”与取内容运算“*”.....	116
6.2.2	指针与整数的加减运算.....	117
6.2.3	指针的关系运算.....	118
6.3	指针与数组.....	119
6.3.1	一维数组的指针.....	119
6.3.2	二维数组的指针.....	121
6.3.3	字符串指针.....	123
6.4	指针数组和指向指针的指针.....	126
6.4.1	指针数组.....	126
6.4.2	指向指针的指针.....	128
6.5	指针与结构体.....	130
6.5.1	结构体指针变量.....	130
6.5.2	指向结构体数组的指针.....	131
6.5.3	结构体指针数组.....	132
6.6	指针与链表.....	134
6.6.1	链表.....	134
6.6.2	内存的动态分配和释放.....	135
6.6.3	链表结点在C语言中的表示.....	136
6.6.4	链表的操作.....	137
6.7	习 题 六.....	143
6.8	上机实习六.....	149
第七章	函 数	151
7.1	函数定义和调用.....	151
7.1.1	函数定义.....	151

7.1.2	函数调用.....	153
7.2	函数的递归调用.....	158
7.3	函数与数组.....	162
7.3.1	数组元素作函数的参数.....	162
7.3.2	数组名作为函数的参数.....	162
7.4	函数与指针.....	164
7.4.1	指针作为函数的参数.....	164
7.4.2	返回值是指针的函数.....	164
7.4.3	指向函数的指针.....	166
7.4.4	函数与链表.....	168
7.5	命令行参数.....	171
7.6	习 题 七.....	173
7.7	上机实习七.....	179
第八章 变量存储类别及编译预处理命令		181
8.1	变量的存储类别.....	181
8.1.1	局部变量和全局变量.....	181
8.1.2	动态存储和静态存储.....	184
8.2	编译预处理.....	188
8.2.1	宏定义.....	188
8.2.2	文件包含.....	191
8.2.3	条件编译.....	191
8.3	习 题 八.....	194
8.4	上机实习八.....	197
第九章 文 件 操 作		198
9.1	文件与流概念.....	198
9.2	打开、关闭文件函数.....	199
9.2.1	打开文件函数fopen().....	200
9.2.2	关闭文件函数fclose().....	200
9.3	标准流式文件.....	201
9.4	文件读写函数.....	202
9.4.1	fputc()和fgetc()函数.....	202
9.4.2	fputs()和fgets()函数.....	205
9.4.3	feof()函数.....	205
9.4.4	fread()函数和fwrite()函数.....	207
9.4.5	格式输出函数fprintf()和格式输入函数fscanf().....	210

9.5	文件定位函数.....	212
9.5.1	rewind()函数.....	212
9.5.2	fseek()函数.....	212
9.5.3	ftell()函数.....	213
9.6	习 题 九.....	214
9.7	上机实习九.....	216
附录一	C语言运算符一览表.....	218
附 录 二		220
181	命令型及过程型数据库系统分类与特点.....	1.1.1
181		1.1.2
181		1.1.3
181		1.1.4
181		1.1.5
181		1.1.6
181		1.1.7
181		1.1.8
181		1.1.9
181		1.1.10
181		1.1.11
181		1.1.12
181		1.1.13
181		1.1.14
181		1.1.15
181		1.1.16
181		1.1.17
181		1.1.18
181		1.1.19
181		1.1.20
181		1.1.21
181		1.1.22
181		1.1.23
181		1.1.24
181		1.1.25
181		1.1.26
181		1.1.27
181		1.1.28
181		1.1.29
181		1.1.30
181		1.1.31
181		1.1.32
181		1.1.33
181		1.1.34
181		1.1.35
181		1.1.36
181		1.1.37
181		1.1.38
181		1.1.39
181		1.1.40
181		1.1.41
181		1.1.42
181		1.1.43
181		1.1.44
181		1.1.45
181		1.1.46
181		1.1.47
181		1.1.48
181		1.1.49
181		1.1.50
181		1.1.51
181		1.1.52
181		1.1.53
181		1.1.54
181		1.1.55
181		1.1.56
181		1.1.57
181		1.1.58
181		1.1.59
181		1.1.60
181		1.1.61
181		1.1.62
181		1.1.63
181		1.1.64
181		1.1.65
181		1.1.66
181		1.1.67
181		1.1.68
181		1.1.69
181		1.1.70
181		1.1.71
181		1.1.72
181		1.1.73
181		1.1.74
181		1.1.75
181		1.1.76
181		1.1.77
181		1.1.78
181		1.1.79
181		1.1.80
181		1.1.81
181		1.1.82
181		1.1.83
181		1.1.84
181		1.1.85
181		1.1.86
181		1.1.87
181		1.1.88
181		1.1.89
181		1.1.90
181		1.1.91
181		1.1.92
181		1.1.93
181		1.1.94
181		1.1.95
181		1.1.96
181		1.1.97
181		1.1.98
181		1.1.99
181		1.1.100

第一章 绪论

[学习目标]

- 1.了解C语言的发展,掌握C语言及C程序的特点;
- 2.熟练掌握TC系统集成环境的使用,编辑、调试、运行第一个C语言程序。

1.1 C语言概述

1.1.1 C语言的发展

C语言是国际上流行的一种适合于编写系统软件的高级程序设计语言,正是由于C语言的这一特殊的特点,它正被越来越多的人所接受,并被许多学校选为计算机语言的初学语言。

C语言的产生和发展与优秀的UNIX操作系统有着十分密切的联系,在此有必要对UNIX系统作一个简单的介绍。UNIX起源于60年代末,它是美国贝尔实验室为DEC公司的PDP-7型计算机研制和开发的操作系统。最初选用的程序设计语言是汇编语言,由于汇编语言的难以移植、难以调试、编程困难等诸多原因,贝尔实验室的工程师肯·汤普逊(Ken Thompson)等工作人员尝试发明新的适合编写系统软件的计算机语言。他们在借鉴早期编程语言BCPL(Basic Programming Language)的基础上,研制出一种新型的“B”语言,并用此语言将UNIX系统重写。随着UNIX系统的推广以及后来的人们不断将B语言本身作了改进和完善,提出了一种用结构化程序设计思想进行程序设计的新型语言——C语言。C语言在其版本的不断演化中得到改进和完善。1983年,美国国家标准化协会(ANSI)根据C语言问世以来各种版本对C语言的发展和扩充,制定了新的标准,称为ANSI C。1987年,ANSI又公布了新标准——87ANSI C。本书的绝大部分内容都是基于该标准编写的。

C++语言是在C语言的基础上,在80年代后期研制成功的一种面向对象的程序设计语言,它正显示出越来越强大的生命力和灿烂的发展前景。但鉴于本书面向的读者是中等职业技术学校的学生及非计算机专业的学生,因此,有关面向对象程序设计的知识在本书就不作阐述。

1.1.2 C语言的特点

自计算机发明以来,控制计算机工作的编程语言有许多,但是流行的语言却不多。C语言作为一种最具有长久生命力、最适合编写系统软件的计算机语言,总有优越于、不同于其他语言的特点。它的特点是多方面的,总体考虑有以下几点:

1.C语言数据类型丰富

C语言除了整型、字符型、浮点型等基本数据类型外,还允许用户定义诸如数组、结构、共用、枚举、指针等高级数据类型来描述复杂的数据结构。特别是C语言的指针类型,功能强大,灵活多变,大大简化了程序结构,提高了程序的运行效率,是C语言的重要特色之一。当然,这一部分也是C语言中较难掌握的内容,希望读者学习到这部分内容时,应引起重视。

2.C语言具有丰富的运算符

C语言的运算符非常丰富。其他语言中的一些语法规则,甚至一些语句,在C语言中也将它作为运算符来处理。例如,赋值运算符、括号等。C语言有直接求变量地址的运算符,也有直接实现二进制运算的运算符。正是由于C语言运算符的多样性,使得C语言中的表达式多样化,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。有时看似一个非常复杂的问题,用C语言可能仅仅是一个简单的表达式。因此,希望初学者应正确理解C程序中的一些表达式的含义。

3.C语言是一种结构化程序设计语言

C语言提供的分支、循环和模块的基本结构和语句,便于实现由顶向下、逐步求精的结构化程序设计方法。程序可读性好,便于调试和维护。C语言的主要结构成分是函数,程序以函数为模块有利于把整体程序分割成若干个相对独立的功能模块,同时也为程序模块之间的相互调用及数据传递提供了许多方便,这一特点有利于大型软件模块化,为多人共同开发大型软件的软件工程技术提供了强有力的支持。

4.C语言能生成高效率的目标代码

C语言具有丰富的运算符和直接控制计算机硬件的能力,在必要时可以直接对字节、位、甚至内存地址和寄存器进行操作;C语言也可以直接嵌入汇编语言模块代码;C语言的编译系统也注重了目标代码的效率问题。基于以上因素,用C语言编写的程序产生的目标代码短、运行速度快,在各种高级语言中,它的效率最高。即使有的系统在设计初期,基于程序员的知识,可能是用其他高级语言编写的,但是最后却常常由于系统运行速度慢、达不到用户的要求而将系统用C语言进行改写。

5.C语言允许直接访问物理地址,可以实现汇编语言的大部分功能

C语言可以直接访问物理地址,能进行位操作,可以直接对硬件进行操作。因此,C语言既具有高级语言的功能,又具有低级语言的许多功能。C语言的这种双重性,使得

功的系统描述语言，又是通用的程序设计语言。也正是基于这种双重的特点，C语称为“中级语言”，意为兼有高级语言和低级语言的特点。

1.2 C语言程序的基本结构

任何一种程序设计语言都具有特定的语法规则和规定的表达方式。一个程序只有严格按照语言规定的语法和表达方式编写，才能保证编写的程序在计算机中能够被正确执行；在编写程序时，要求程序员遵循良好的程序设计风格，便于其他读者阅读和理解。

用C语言编写的程序称为C语言源程序。一个C语言源程序文件由一个或多个函数构成，函数是C程序的基本构成单位。每个C程序必须有一个名为main的主函数，程序的执行总是从主函数开始的。

我们通过两个例子来说明C程序的基本结构。

下面1

例1- *程序DH1.C,在屏幕上显示:How are you!*/

```
main()
{
    printf("How are you!\n");
}
```

运行结果为:

运行 How are you! ↵

C语言源程序非常简单，仅由一个main()主函数构成。在主函数中仅仅只有一个语句，该语句是调用C语言中提供的库函数printf()形成的，printf()函数的功能是向屏幕输出句，该句的数据和信息，每个语句末以分号作为语句的结束标志。

相关的源程序中的第一行是C语言的注释，C语言的注释是由“/*”和“*/”括起来的任何文字，它可以出现在程序中的任何地方。C语言中的注释功能用来说明程序段的功能、变量的字，以及程序员认为应该向程序阅读者说明的任何内容。在将C程序编译成目标代码时，作用 的注释都将被忽略。C语言注释的多少不影响目标代码的效率。注释可以使程序清晰所有，便于调试、修改、维护，便于程序员之间相互交流与协作。但是需要注意的是：注释标志不允许嵌套。例如，下面的注释形式是非法的：

解释1- /*C语言注释/*非法1*/请注意*/

/*C语言注释非法2*/请注意*/

1 2

程序DH2.C,求最大值/

```
main()
{
    int i,j,z;
    scanf("i=%d,j=%d",&i,&j);
```

```

    z=max(i,j);
    printf("max=%d\n",z);
}
/*求两整型数据的最大值*/
int max(x,y)
int x,y;
{
    int m;
    m=x>y?x:y;
    return(m);
}

```

运行结果为:

输入: i=10,j=16

输出: max=16

该C语言源程序由两个函数构成,从主函数的第一个语句开始执行。首先定义变量,然后输入变量*i*和*j*的值。`scanf()`函数与`printf()`函数相对应,是从终端给量输入值。第三个语句表示主函数调用用户自己定义的`max()`函数,将*i*、*j*参数的`max()`函数的*x*、*y*参数,执行`max()`函数中的语句,通过`return`语句返回*x*、*y*的最给*z*变量,最后将求得的最大值在屏幕上输出显示。执行完主函数的所有语句,执行完毕。

从上述两个例子我们可以了解到C语言的函数有以下3种:

(1)主函数`main()`。每个程序中必须有一个主函数,也只能有一个。在运行时,主函数中的第一条语句开始执行,执行完最后一条语句后,整个程序就执行完毕。

(2)用户自己编写的函数(也称为自定义函数或子程序)。程序中用户自定义函数不限,当然也可以没有。实际上,一个C语言应用程序通常都包含若干用户自定义,灵活地定义若干功能单一、结构简单的程序模块(也即函数),能提高程序的可读性,于调试和维护。

(3)标准函数(也即库函数)。C语言提供了若干个不同功能的标准库函数,便于编写各种应用程序。C语言的开发者已经将许多常用的算法和功能编写成函数,然后译后生成的目标文件统一存放在函数库中。这样,程序员在编写程序时如果需要实现功能,而库函数中具有完成这些功能的函数,那么就不必自己再编写程序实现了,直接调用相应的库函数就可以了。但是需要注意的是,由于这些库函数的目标程序是相关的文件中,所以需要在引用这些库函数的程序中指明这些文件的位置。实现用C语言中的`#include`预编译命令来实现的。

各种版本的C语言提供的库函数的数目也各不相同,一般都在数百个以上。4. 一些常用的库函数 具体介绍,并且要求读者掌握,对于其他的库函数只要求读者了书将具体实践中,如果需要引用一些库函数,可以查看相关资料,掌握这些库函数的具体解方法。

现在,我们可以把书写C程序的基本要点归纳如下:

①C程序习惯使用小写英文字母，常量和特殊用途的符号可以用大写字母。C语言分大小写字母，如果将主函数名main写成MAIN，程序将不能运行。

②C语言程序不存在程序行概念。一行中可以写多个语句，有时也可以将一个语句写多行上。分号是一个语句的结束标记，多个语句之间必须用分号间隔。

③用一对花括号标志一个程序功能块的开始和结束。为了提高程序的可读性，一般总让每一对花括号按列对齐。

④在程序中增加适当的空格或空行，可以将各语句组按其功能和嵌套关系进行缩排，程序的模块和复合关系变得明显，增加程序层次的清晰性，以培养用户良好的程序设计风格。

⑤充分利用注释功能。一个好的、有使用价值的源程序都应当加上必要的注释，以增加程序的可读性。

1.3 C程序的编辑、编译、链接和运行

计算机是一种传输、存储和处理信息的电子设备，其基本工作过程包括：数据的输入、数据在计算机中的处理、数据的输出。数据是信息的载体，计算机无论处理什么类型的数据，都是通过执行人所编制的程序来完成预定目标的。因此，用计算机解决一个实际问题，首先要进行程序设计，其过程是：

(1)分析问题；

(2)找出解决问题的算法和确定适合的数据结构；

(3)选用一种高级语言描述算法，具体编写解问题的程序。

C语言和其他编译型高级语言一样，其运行一所有要经过4个步骤，即：源程序的编辑、源程序的编译、目标程序的链接、可执行程序的运行，这一过程可用如图1-1所示的流程图描述。

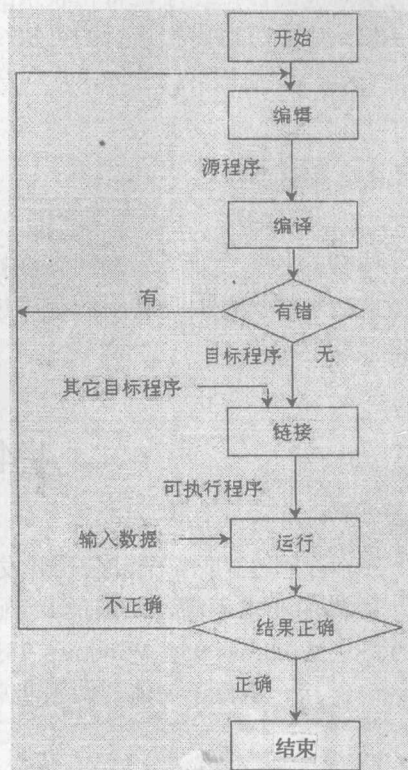


图 1-1

1.3.1 源程序的编辑

程序员使用一种编辑软件将自己编写的程序输入到计算机中并以文本文件的形式存储起来，文件的后缀名为“.C”，以备程序员在调试和修改时能够识别是C语言程序。现有的编辑软件很多，如常用的Edit、Word、WPS等，大家也可

以利用任何一个程序开发平台来编辑，后面我们将着重介绍基于C语言的TC集成开发平台。

1.3.2 源程序的编译

编译是指借助于对应的编译程序将编辑好的源程序翻译成二进制文件的过程，生成的这种二进制文件也称为目标文件。编译的作用是对程序员编辑的源程序逐行进行语法和语义的检查，例如，使用的括号不配对、变量没有定义、语句格式不正确等等。当在编译时发现错误，编译程序将给出相应的提示信息，指导程序员对源程序作出相应的修改，然后再重新进行编译，直至将源程序的错误修正完为止。源程序编译后生成的目标文件以“.OBJ”为后缀。

1.3.3 目标文件的链接

编译后生成的目标文件是可以重新定位的程序模块，此时不能够直接运行。链接就是借助于对应的链接程序，将相联系的目标文件以及系统提供的相应的支持函数、模块链接在一起，生成可以直接运行的可执行文件的过程。在DOS系统中，生成的可执行文件一般以“.EXE”为后缀。

1.3.4 运行

生成可执行文件后，就可以在操作系统下运行，以达到预期的运行结果。若结果并非预期的那样，此时还得对源程序进行检查、修正、重复编译、重复链接这些过程，直至取得预期的结果。

1.4 Turbo C 2.0

Turbo C 2.0是Borland公司开发的C语言编译软件包，功能相当完善。它的安装、运行对于硬件的要求不高，一般的286配置就可以了；它对于操作系统也没有什么选择，流行的DOS、Windows 95、Windows 98操作系统皆可运行。Turbo C 2.0的集成环境是集编辑、编译、链接、运行于一体，使用非常方便，成为初学者使用的最流行的C语言编译器。

1.4.1 安装方法和主要文件

Turbo C 2.0的所有内容存储在两张高密软盘上，一般将Turbo C 2.0安装在单机硬盘上

或服务服务器上，只要运行第一张软盘上的安装程序install.exe，再根据其相关提示顺序选择即可。

Turbo C 2.0的内容很多，以下文件是最常用到的：

1. 头文件

头文件的内容包括库函数、全局变量的说明、宏和结构的定义等，头文件的内容可以通过预编译命令调用。头文件一般放在TC目录中的include子目录下。表1-1对常用的头文件的功能作一简要说明。

表 1-1

头文件	文件说明	头文件	文件说明
ALLOC.H	存储分配管理	MATH.H	数学计算
ASSERT.H	调试说明	MEM.H	内存管理
BIOS.H	BIOS调用	PROCESS.H	过程处理
CONIO.H	字符终端显示	SETJMP.H	远程调用控制
CTYPE.H	字符判断	SHARE.H	共享文件常数处理
DIR.H	目录处理	SIGNAL.H	ANSI信号处理
DOS.H	DOS调用	STDARG.H	标准参数管理
ERRNO.H	出错信息定义	STDDEF.H	标准定义
FCNTL.H	控制常数	STDIO.H	标准输入/输出
FLOAT.H	浮点处理	STDLIB.H	标准库函数
GRAPHICS.H	图形处理	STRING.H	字符串处理
IO.H	IO处理	TIME.H	时间处理
LIMITS.H	数值上下限	VALUES.H	常数说明

2. 库文件

Turbo C 2.0系统提供了许多经常用到的一些函数，这些函数不再需要程序员自己编写程序，只要给出正确的调用格式直接引用这些函数即可，这类函数就是库函数。库函数分装在若干个库文件中，库文件存放不同编译模式下的库函数的目标代码。例如，CS.LIB库文件存放着小模式下所有库函数的目标代码。但是，有关图形处理的库函数单独存放在GRAPHICS.LIB库文件中。

3. TC.EXE

TC.EXE是Turbo C 2.0的集成编辑环境文件。该环境集编辑、编译、链接、运行这些过程为一体，是程序员编辑、调试源程序的一个很好的平台。该平台的使用方法在后面我