



普通高等学校职业教育教改示范教材

C语言程序设计

孙琳 张江城 主编 杨文 主审



中国轻工业出版社

普通高等学校职业教育教改示范教材

C 语言程序设计

孙 琳 张江城 主编
杨 文 主审

图书在版编目 (CIP) 数据

C 语言程序设计/孙琳, 张江城主编. —北京: 中国轻工业出版社, 2005. 8

普通高等学校职业教育教改示范教材

ISBN 7-5019-4975-1

I. C... II. ①孙...②张... 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 (2005) 第 071845 号

责任编辑: 王 淳

策划编辑: 王 淳 责任终审: 孟寿萱 封面设计: 邱亦刚

版式设计: 马金路 责任校对: 李 靖 责任监印: 胡 兵

出版发行: 中国轻工业出版社 (北京东长安街 6 号, 邮编: 100740)

印 刷: 北京公大印刷厂

经 销: 各地新华书店

版 次: 2005 年 8 月第 1 版 2005 年 8 月第 1 次印刷

开 本: 787 × 1092 1/16 印张: 19.5

字 数: 432 千字

书 号: ISBN 7-5019-4975-1/TP · 077 定价: 28.00 元

读者服务部邮购热线电话: 010—65241695 85111729 传真: 85111730

发行电话: 010—65141375 85119845

网 址: <http://www.chlip.com.cn>

Email: club@chlip.com.cn

如发现图书残缺请直接与我社读者服务部联系调换

41284J4X101ZBW

前 言

C 语言是一种结构化、可编译的通用程序设计语言。它具有功能丰富、表达能力强、使用灵活、应用面广、可移植性好等特点,它既具有高级语言的优点,又具有低级语言的特点,可用于系统软件和应用软件的开发。目前在许多高等院校,不仅计算机专业开设了 C 语言课程,而且非计算机类专业也开设了本课程。全国计算机等级考试、全国计算机应用技术证书考试(NIT)和全国各地区组织的大学生计算机统一考试都将 C 语言列入了考试范围。

本书是在多年教学改革与实践的基础上,依据现代计算机编程技术发展的需求,以及学生的实际情况,本着“注重基础,讲清原理,注重实践,提高能力,深入浅出,便于理解”的原则编写此书。本书内容以应用为目的,注重实用性,先进性,尽量删繁就简,使教材重点突出,概念清楚,实用性强。

本书在内容安排上,以培养学生的应用能力为目的,将理论知识的讲授与技能训练有机结合,融为一体,使能力培养贯穿于整个教学过程。每章都有实训内容,内容丰富、实用,使学生通过学习逐步提高应用能力和综合素质。

本教材有如下特点:

1. 本书以 C 语言基本内容为主线,精心组织安排内容结构。各章深入浅出,难点分散,内容充实,安排合理,衔接自然。

2. 每章后有“本章小结”,配有一定量的习题并附答案,便于教师教学与读者自学。习题靠近全国计算机等级考试类型及难度,一些例题、习题取自等级考试真题,为读者参加全国计算机等级考试打下基础。

3. 每章后有上机实验。教材组织的出发点应是重视操作,重视操作的途径则应将“验证性”实验向“探索式”实验转变。本书注重由浅入深的引导,首先向学生提几个为什么,为什么这个语句要设在这里?这个语句起什么作用?等等,引导他们去分析程序,得到结果。然后要学生自己提问,自己分析,培养提高他们的分析设计能力及逻辑思维能力。

4. 本书利用一节的篇幅简单介绍了 Turbo C 2.0 的操作过程,对教师教学及初学者上机操作有很大帮助。

5. C 语言的图形和用户界面技术是编写实用软件必不可少的,需要掌握相关库函数的使用。此部分虽不属于 C 语言必备的内容,但对想进一步学习的读者却有很大帮助。本书最后一章介绍了图形和用户界面技术,可作为各学校的选学内容及读者的自学内容。

本书可作为高等院校各专业 C 语言程序设计课程的教材,也可供广大编程技术人员和 C 语言自学者参考。同时,本书内容尽量贴近全国计算机等级考试,也可作为全国计算机等级考试参考用书。

本书根据辽宁信息职业技术学院、湖北三峡职业技术学院、哈尔滨职业技术学院、辽宁机电职业技术学院、郑州牧业工程高等专科学校、辽阳职业技术学院、湖北轻工职业技术学院等院校教学实践编写而成,其中孙琳和张江城任主编,第一、二章由张春波编写,第三、四章由王辉编写,第五、六章由王兰编写,第七章由王艳芳编写,第八、九、十、十三章由孙琳编写,第十一、十二章由徐刚编写。最后由孙琳、张江城对全书进行了统稿,杨文、徐翠娟、李灵佳进行了审稿。在此书编写中得到了窦玮、张冬青、王宇、黄冬梅、王慕堂、李峰等老师的指导意见,同时在本书编写过程中,编者参考了目前国内比较优秀的关于 C 语言程序设计方面的资料,在此谨向有关作者表示感谢。

各院校师生如发现书中存有错处,请与中国轻工业出版社王淳编辑联系,以便我们及时修订。

电子信箱: wchun12@163.com

编者

2005 年 4 月

目 录

第一章 C 语言概述	(1)
第一节 绪论	(1)
第二节 C 语言的程序结构	(2)
第三节 Turbo C 的操作过程	(6)
第四节 上机实验	(10)
第一章习题	(10)
第二章 数据与运算	(12)
第一节 标识符	(12)
第二节 常量和变量	(13)
第三节 整型数据	(15)
第四节 实型数据	(17)
第五节 字符型数据	(18)
第六节 变量赋初值	(21)
第七节 运算符与表达式	(21)
第八节 类型转换	(26)
第九节 上机实验	(27)
第二章习题	(29)
第三章 顺序结构程序设计	(34)
第一节 程序设计的三种基本结构	(34)
第二节 C 程序的语句	(36)
第三节 标准库函数	(37)
第四节 数据输出	(38)
第五节 数据输入	(41)
第六节 上机实验	(45)
第三章习题	(47)
第四章 选择结构程序设计	(50)
第一节 关系运算符及关系表达式	(50)
第二节 逻辑运算符及逻辑表达式	(51)
第三节 条件运算符和条件表达式	(53)
第四节 if 语句	(54)
第五节 switch 语句	(58)
第六节 程序举例	(60)
第七节 上机实验	(62)
第四章习题	(64)
第五章 循环结构程序设计	(69)
第一节 while 语句	(69)

第二节	do-while 语句	(71)
第三节	for 语句	(74)
第四节	循环的嵌套	(78)
第五节	continue 语句	(80)
第六节	break 语句	(81)
第七节	goto 语句	(82)
第八节	程序举例	(82)
第九节	上机实验	(86)
第五章习题		(87)
第六章 数组		(92)
第一节	一维数组	(92)
第二节	二维数组	(97)
第三节	字符数组	(100)
第四节	上机实验	(109)
第六章习题		(110)
第七章 函数		(116)
第一节	函数概述	(116)
第二节	函数的定义和返回值	(117)
第三节	函数的调用	(119)
第四节	函数的声明	(121)
第五节	函数的递归调用	(123)
第六节	数组名作函数的参数	(125)
第七节	变量的作用域和存储类型	(129)
第八节	内部函数和外部函数	(138)
第九节	如何运行一个多文件的程序	(140)
第十节	函数举例	(141)
第十一节	上机实验	(142)
第七章习题		(144)
第八章 指针		(147)
第一节	指针的基本概念	(147)
第二节	指针变量	(148)
第三节	指针与数组	(157)
第四节	字符串的指针和指向字符串的指针变量	(166)
第五节	指针与函数	(170)
第六节	指针数组和指向指针的指针	(173)
第七节	上机实验	(178)
第八章习题		(181)
第九章 编译预处理		(188)
第一节	宏定义	(188)
第二节	文件包含	(192)
第三节	条件编译	(194)

第四节 上机实验	(195)
第九章习题	(197)
第十章 结构体与共用体	(199)
第一节 定义结构体类型	(199)
第二节 结构体变量	(201)
第三节 结构体数组	(204)
第四节 结构体类型指针	(207)
第五节 链表及其操作	(212)
第六节 共用体	(219)
第七节 枚举类型	(222)
第八节 用 typedef 定义类型	(223)
第九节 上机实验	(224)
第十章习题	(227)
第十一章 位运算	(233)
第一节 位运算符和位运算	(233)
第二节 位段	(235)
第十一章习题	(237)
第十二章 文件	(239)
第一节 C 语言文件的概念	(239)
第二节 文件类型指针	(240)
第三节 文件的打开和关闭	(241)
第四节 文件的读写	(243)
第五节 文件的定位	(248)
第六节 出错检测函数	(250)
第七节 上机实验	(250)
第十二章习题	(252)
第十三章 图形和用户界面技术	(255)
第一节 文本窗口的程序设计	(255)
第二节 图形显示技术	(257)
第三节 汉字显示技术	(261)
第四节 菜单设计	(268)
第五节 上机实验	(278)
各章习题参考答案	(281)
附录	(295)
附录一 ASCII 码表	(295)
附录二 运算符的优先级和结合性	(296)
附录三 Turbo C 2.0 库函数	(297)
参考文献	(304)

第一章 C 语言概述

C 语言是当前国际上广泛流行的计算机程序设计语言。它既可以用来编写系统软件，也可以用来编写应用软件。通过本章的学习，读者能够：

- 了解 C 语言产生的背景和发展现状；
- 了解 C 语言的基本特点；
- 掌握 C 程序的基本结构和一般形式；
- 初步掌握 Turbo C 集成开发环境的使用和在 Turbo C 中编辑、调试和运行一个 C 程序的具体步骤和方法。

第一节 绪 论

一、C 语言的产生和发展

C 语言是目前极为流行的程序设计语言之一，它既具有高级语言的功能又具有汇编语言的一些特性。它是 1972 年由美国 BELL 实验室的 D. Ritchie 和 B. Kernighan 等人首先推出的。之后，C 语言又经过不断改进逐步得到完善。直到 1978 年 B. Kernighan 和 D. Ritchie（简称 K&R）合著了影响深远的著作《The C Programming Language》（C 程序设计语言），首次系统介绍了 C 语言，建立了所谓的 C 语言的 K&R 标准，它一度成为 C 语言的事实标准。

最初，C 语言是为了开发多用户的 UNIX 操作系统而由美国 BELL 实验室研制出来的。最早的 UNIX 是用汇编语言编写的。由于汇编语言可移植性和可读性较差，K. Thompson 决定开发一种高级语言来描述 UNIX。1970 年 K. Thompson 在 PDP-11/20 机器上开发出了 B 语言，并用 B 语言重新写了 UNIX 的绝大多数实用程序。B 语言的主要思想来源于 BCPL 语言。而 BCPL 是剑桥大学的 M. Richards 基于 CPL 语言在 1967 年发表的一种语言。但是 B 语言过于简单，功能有限。1972 ~ 1973 年，D. Ritchie 在 B 语言的基础上研制了 C 语言。C 语言既保持了 BCPL 和 B 的“面向机器”的特征，又克服了它们过于简单，数据无类型的缺点。1973 年，K. Thompson 和 D. Ritchie 两人合作将 UNIX 的 90% 以上代码用 C 语言改写了一遍（即 UNIX V5）。

1977 年，出现了不依赖于具体机器的 C 语言编译文本，这就推动了 UNIX 操作系统迅速在各种机器上实现。随着 UNIX 的日益广泛使用，C 语言的优点引起人们的普遍注意，同时也得到迅速推广。C 语言和 UNIX 操作系统相辅相成。此后，C 语言的发展非常迅速，各种版本的 C 语言相继涌现出来。由于没有统一的标准，使得这些 C 语言之间出现了一些不统一的地方。为了改变这种情况，美国国家标准化协会（ANSI）于 1983 年制定了一套标准，称为 ANSI C（标准 C），成为各种 C 语言版本的基础。

C 语言本身也在发展，20 世纪 80 年代中期，出现了面向对象程序设计的概念，BELL 实验室的 B. Stroustrup 博士借鉴了 Simula 67 中类的概念，将面向对象的语言成分引入 C 语言中，设计出了 C++ 语言，C++ 语言赢得了广大程序员的喜爱，不同的机器不同的操作系统几乎都支持 C++ 语言。同时，C++ 语言也得到了国际标准化组织（ISO）的认可。目前，国际标准化组织已对 C++ 语言实现标准化。

目前微机中使用的 C 语言版本有很多，比较经典的有 Turbo C、Borland C、Microsoft C

等。近年来，又推出了包含面向对象程序设计思想和方法的 C++，它们均支持 ANSI C，本书主要介绍 Turbo C，同时兼顾各种版本的通用性和一致性。

二、C 语言的特点

C 语言之所以能存在和发展，具有生命力并成为程序员的首选语言之一，是因为它具有如下特点：

(1) 语句简练、紧凑，语法规则少，使用方便、灵活。

(2) 语言功能丰富，它不仅提供了丰富的运算符、数据类型，还提供了各种功能强大的系统函数。使程序员能轻松地实现各种复杂的数据结构和运算。

(3) C 语言是一种结构化程序设计语言，具有实现结构化程序设计的控制语句，以及具备抽象功能并体现信息隐蔽思想的函数，可以实现程序的模块化设计。

(4) C 语言既具有高级语言的通用性及易写易读的特点，又具有汇编语言的“位处理”、“地址操作”等能力。这使得 C 语言不仅能像其他高级语言一样用于应用软件的编写，还能像汇编语言一样用于系统软件的编写。

(5) C 语言具备较高的可移植性。C 语言本身不依赖于机器硬件系统，从而便于在不同种机器中实现程序的移植。

(6) 编译后生成的代码质量高，运行速度快。

第二节 C 语言的程序结构

一、简单的 C 语言程序结构

用 C 语言编写的源程序，简称 C 程序。C 程序是基于函数结构而构成的，它由一个或多个函数按照一定的规律组合而成。其中必须有一个名为 main 的主函数，程序的执行总是从这里开始的。

下面先介绍 3 个简单的 C 程序，通过这 3 个 C 程序再分析归纳 C 程序的一般结构。

(一) 简单的 C 程序

例 1.1：在屏幕上输出一行文本信息。

```
main()                      /* 主函数 */
{
    printf("How are you!\n"); /* 在屏幕上输出一行文本信息并换行 */
}
```

运行结果如下：

How are you!

说明：

(1) 本程序的功能是在屏幕上输出一行文本信息：How are you!

(2) main()为主函数名。每个 C 程序都必须有一个 main()函数。

(3) 大括号“{}”是函数体界定符，位于大括号{.....}中的内容为函数体，每个函数都必须用一对大括号将函数体括起来。

(4) 函数体中 printf()是一个函数，括号()中的内容是这个函数的参数，括号()后加

了一个分号“;”即构成了一个函数调用语句,也称之为输出语句。其目的是将引号中的内容 How are you!原样输出。printf()为 C 语言的标准输出函数,是系统提供的库函数。

(5) 语句后面有一个分号“;”,这是 C 语言的语句结束符。

(6) “\n”是转义字符(关于转义的概念,将在第二章第五节中讲述),其具体的含义是:换行,或者说,将当前位置移到下一行开头。在本例中的作用是:在输出 How are you!后自动回车换行。

(7) /*.....*/是注释语句,用来帮助读者阅读程序,在程序编译运行时/*和*/之间的内容是不起作用的,注释语句可写在程序中的任何位置。

例 1.2: 编写一 C 程序,计算并输出两数之和。

```
main()                /* 主函数 */
{
    int a,b,sum; /* 定义 3 个整型变量 */
    a=210; /* 给变量 a 赋值 */
    b=342; /* 给变量 b 赋值 */
    sum=a+b; /* 计算 a+b 的值并送到变量 sum 中保存 */
    printf("sum is %d",sum); /* 输出文字 "sum is"和变量 sum 的值并换行 */
}
```

运行结果如下:

```
sum is 552
```

说明:

(1) 在 main()函数中首先定义了 3 个整型变量 a、b、sum。

(2) 语句“a=210; b=342;”对变量 a、b 进行赋值。

(3) 语句“sum=a+b;”计算 a+b 的值并将它送给 sum 变量。

(4) printf()是一个函数,在它的最后加了一个“;”号构成了函数调用语句,其功能是根据要求输出 sum,即将符号 sum is 运算结果 552 一起输出。大家可以看到,本例中的 printf()的格式与上例中 printf()的格式有些区别。其实,上例中的输出函数的形式是本例的一种特例。也就是说,本例的形式就是标准输出格式的一般形式。至于()中的组成部分,%d 的含义等等,将在第三章第四节详细讲述。

例 1.3: 从键盘上输入两个整数,比较两个数的大小,并输出较大者。

```
main()                /* 主函数 */
{
    int a,b,c;                /* 定义整型变量 a,b,c */
    printf("Enter Two Numbers: "); /* 输出提示信息 */
    scanf("%d,%d",&a,&b); /* 从键盘接收 2 个整数并送到变量 a,b 中 */
    c=max(a,b);                /* 调用 max()函数,将得到的值赋给变量 c */
    printf("max=%d\n",c); /* 输出文字 "max=" 和变量 c 的值,并换行 */
}

int max(int x,int y) /* 定义值为整型的函数 max(), x,y 是整型形式参数 */
{
    int z;                /* 定义 max()函数中用到的整型变量 z */
```

```

if(x>y) z=x;      /* 比较 x,y 的大小, 将大的送 z */
else z=y;
return(z);        /* 将 z 中的值由函数名 max 带回调用处 */
}

```

运行结果如下:

```
Enter Two Numbers:32, 28✓
```

```
max=32
```

说明:

(1) 本程序包括两个函数, 主函数 `main()` 和用户自己定义的函数 `max()`。`max()` 函数的作用是将 `x` 和 `y` 中较大者的值赋给变量 `z`, 并由 `return` 语句将 `z` 中的值返回调用处。

(2) `main()` 函数中第一个 `printf()` 函数输出一行提示信息: `Enter Two Numbers:` 提示用户从键盘输入两个整数。

(3) `scanf()` 是一个标准输入函数, 它完成 `a, b` 两个变量的输入工作。语句中的 “&” 表示 “取地址”, 此处 `scanf()` 函数的作用是从键盘上输入两个数 32 和 28, 并将这两个数分别送到由 `&a` 和 `&b` 所标识的地址单元中, 使得变量 `a` 的值为 32, 变量 `b` 的值为 28; 语句中 “%d” 的含义与例 1.2 相同, 只是现在用于 “输入”, 它指定输入的两个数据按十进制整数形式输入。在这里, 请注意两数之间要根据输入格式的要求用逗号分隔。关于具体的用法请见第三章第五节。

(4) 语句 “`c=max(a,b);`” 为调用函数 `max()`, 求得 `a, b` 中的最大值并将其赋给变量 `c`, 在调用时将实际参数 `a, b` 的值传送给函数 `max()` 的形式参数 `x, y`。

(5) `main()` 函数中的第二个 `printf()` 函数在将 “`max=%d`” 输出时, 其中的 “%d” 由 `c` 的值代替。

本例中所用到的函数调用、形参和实参等概念, 这里只是做简单的解释, 在以后的章节中会详细介绍。通过上述几个简单的例子, 下面总结 C 程序的基本结构。

(二) C 程序的结构

通过观察上述几个简单的 C 程序, 可以总结归纳出 C 程序的一般形式如下:

<code>main()</code>	<code>/* 主函数 */</code>	
{		} 主函数 <code>main()</code> 的函数体
变量定义序列		
语句序列		
}		
<code>sub1()</code>	<code>/* 自定义函数 sub1 */</code>	
{		} 自定义函数 <code>sub1()</code> 的函数体
变量定义序列		
语句序列		
}		
...		
<code>subn()</code>	<code>/* 自定义函数 subn */</code>	
{		} 自定义函数 <code>subn()</code> 的函数体
变量定义序列		
语句序列		

二、C 程序的结构特点

(一) C 程序是由函数构成的

一个完整的 C 源程序有且只能有一个主函数，可以包含有多个其他函数。主函数有固定的名字 `main()`，任何 C 程序都是从 `main()` 函数开始执行的。C 程序可以包含一个 `main()` 函数和若干个其他函数。函数是 C 程序的基本单位。被调用的函数可以是系统提供的库函数（如：输入函数 `scanf()`，输出函数 `printf()`，绝对值函数 `abs()` 和正弦函数 `cos()`），也可以是用户根据需要自己编写的可以完成某些特定功能的函数。事实上，C 程序中的函数即相当于其他语言中的子程序。C 语言中没有子程序的概念，但 C 语言的库函数是非常丰富的，ANSI C 就可提供 100 多个库函数，Turbo C 和 MS C 4.0 可提供 300 多个库函数。C 语言的这种特点为结构化、模块化的程序设计和实现提供了方便。

(二) 一个函数由函数首部和函数体两个部分组成

1. 函数首部

函数的第一行，按可能出现的先后次序依次包括函数类型、函数名、参数类型、函数参数（形参）名几个部分。且参数类型及参数名要用一对圆括号（）括起。例 1.3 中的函数 `max()` 的首部如下：

<code>int</code>	<code>max</code>	<code>(int x,</code>	<code>int y)</code>
↑	↑	↑	↑
函数类型	函数名	形参类型及形参名 x	形参类型及形参名 y

2. 函数体

函数首部下面的大括号 {.....} 括起来的部分。函数体一般包含两个部分，即“声明部分”和“执行部分”。如例 1.3 中的 `max()` 函数：

```
int max (int x,int y)
{
    int z; /*变量声明部分*/
    if(x>y)z=x; /*执行部分*/
    else z=y;
    return(z);
}
```

(1) 变量声明部分 用来定义程序中所用到的变量，如例 1.3 中 `main()` 函数中的

```
int a,b,c;
```

和 `max()` 函数中的

```
int z;
```

当然，有的函数可以没有变量声明部分，比如例 1.1 中的 `main()` 函数。

(2) 执行部分 由若干语句组成，完成对数据的运算及各种处理。

(三) `main()` 函数的执行

一个 C 程序总是从 `main()` 函数开始执行，在 `main()` 函数中结束。`main()` 函数可以放在程序中的任何位置。

(四) 书写格式

C 程序的书写格式自由，一行内可以写一个或几个语句，一个语句也可以分写在多个行

上。C 程序没有行号，每个语句和数据定义必须用一个分号结尾。

(五) 函数化

C 语言本身没有输入输出语句。输入和输出的操作是由库函数如 `scanf` 和 `printf` 等来完成的，C 程序对输入输出实行“函数化”。

(六) 关于注释

用 `/*...*/` 对程序任何部分进行注释，以增加程序的可读性。

第三节 Turbo C 的操作过程

一、Turbo C 的运行环境和安装

前面已经给出了一些用 C 语言编写的程序，但它们是不能直接运行的，因为计算机只能识别和执行由 0 和 1 组成的二进制的指令，而不能识别和执行用高级语言编写的程序。为了使计算机能执行高级语言所写的程序，必须先用一种称为“编译程序”的软件，把程序翻译成二进制形式的“目标程序”(object program)，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，形成可执行的程序才能被机器所执行。相对于目标程序，用高级语言编写的程序称为“源程序”。在此将介绍利用 Turbo C 2.0 编译程序，把 C 语言源程序编译连接生成可执行程序 (*.exe 文件)。假设 C 源程序名为 `a1.c`，其编辑、编译、连接、执行过程如图 1-1 所示。具体操作步骤如下：

(1) 编辑源程序，并以扩展名 `.c` 的文件存盘。

(2) 对源程序进行编译，将源程序转换为扩展名为 `.obj` 的目标程序，但目标程序仍不能运行。若源程序有错，必须予以修改，然后重新编译。

(3) 对编译通过的源程序连接，即加入库函数和其他二进制代码生成可执行程序。连接过程中，如出现“未定义的函数”等错误，则必须修改源程序，并重新编译和连接。

(4) 执行生成的可执行代码，若不能得到正确的结果，则必须修改源程序，重新编译、连接。若能得到正确结果，则整个编辑、编译、连接、运行过程顺利结束。

下面介绍 Turbo C 2.0 集成开发环境的安装与设置。

(一) 安装 Turbo C 2.0

Turbo C 2.0 是在微机上广泛应用的一个 C 程序开发环境，它具有方便、直观、易用的界面和丰富的库函数。它把程序的编辑、编译、连接和运行等操作全部都集中在一个界面上进行，使用非常方便。

为了使用 Turbo C 2.0 集成开发环境，必须先把 Turbo C 2.0 程序安装在机器的磁盘目录中，例如安装在 C 盘根目录下的 `TC2` 子目录下。

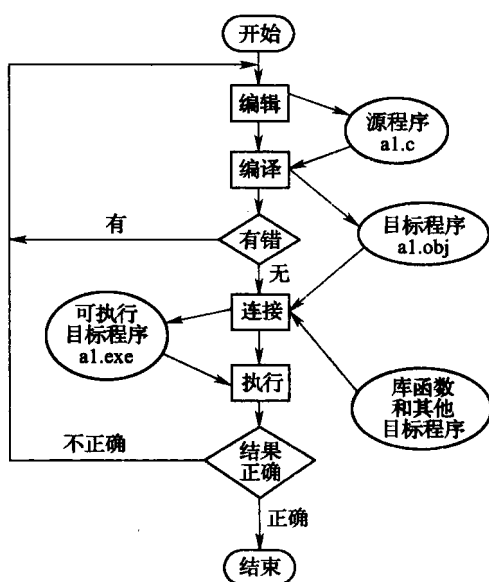


图 1-1 C 程序编辑、编译、连接、执行过程

(二) 建立工作目录

为了方便管理用户的程序文件和维护 Turbo C 2.0 的运行环境,应在机器的磁盘上建立用户自己的工作目录,以便用来存放自己所开发的源程序文件,例如,在 C 盘根目录下建立子目录 C:\myc。

(三) 设置工作目录

按如下步骤操作:

启动 TC, 进入 TC 编辑界面。

按下功能键 F10, 这时光条就会跳到主菜单。

用左、右方向键移动光带, 定位于“File”菜单, 按回车键。

用上、下方向键移动光带, 定位于“Change dir”菜单项, 按回车键, 进入编辑框。

编辑工作目录“C:\myc”, 如图 1-2 所示。



图 1-2 设置工作目录

(四) 设置编译环境

按如下步骤操作: 如图 1-3 所示。



图 1-3 设置编译环境

按下功能键 F10, 这时光条就会跳到主菜单。

用左、右方向键移动光带, 定位于“Options”菜单, 按回车键。

用上、下方向键移动光带, 定位于“Directories”子菜单, 按回车键, 进入编辑框。

编辑框中可分别设置:

Include directories: Turbo C 的包含文件所在目录

Library directories: Turbo C 库函数所在目录

Output directory: 输出目录

Turbo C directory: Turbo C 所在目录

Pick file name: “环境设置”的保存文件(绝对路径)

(五) 保存“环境设置”文件

按如下步骤操作:

按下功能键 F10, 这时光条就会跳到主菜单。

用左、右方向键移动光带, 定位于“Options”菜单, 按回车键。

用上、下方向键移动光带, 定位于“Save options”菜单项, 按回车键, 进入编辑框。

编辑框中输入*.TC 文件的绝对路径, 如图 1-4 所示。



图 1-4 保存“环境设置”

二、Turbo C 的使用

Turbo C 是一个在 DOS 环境下运行的集成开发环境。我们将通过本章的例 1.1 来讲解 Turbo C 中编辑、编译、连接和运行一个 C 程序的步骤。

(一) 编辑—C 语言源程序文件

Turbo C 启动后, 在 Edit 状态下输入源程序代码。输入完毕后, 选择 File/Save, 将程序命名为 a1.C, 回车, 将源程序文件存盘保存。如图 1-5 所示。

一般情况下, 在输入完源程序代码后, 应立即将其存盘, 以防止在调试过程中出现死机等故障而导致程序文件的丢失。

(二) 编译、连接源程序文件

选择 Compile/Compile to OBJ 或 (Compile/Build all) 或按 F9 键, 编译源程序, 生成目标代码文件 a1.OBJ 和可执行文件 a1.EXE, 如果程序有错误, 则需要修改源程序, 再重新进行

此过程；如果程序没有错误，则编译后屏幕显示如图 1-6 所示。



图 1-5 编辑一个 C 程序

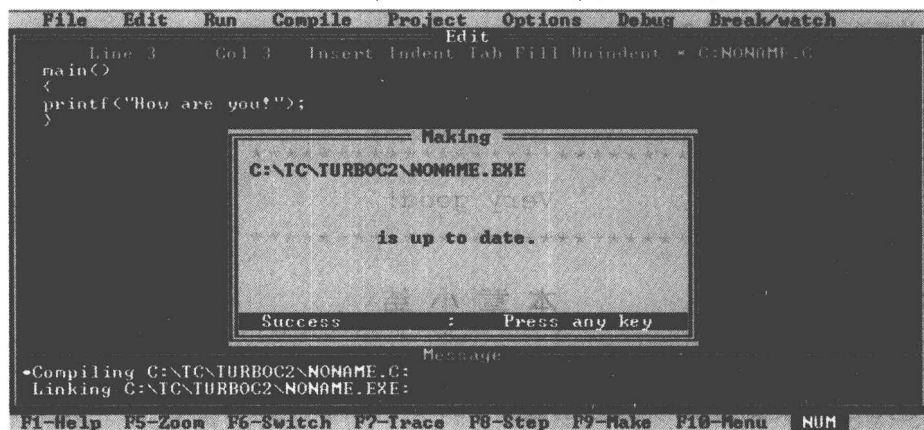


图 1-6 编译、连接

(三) 运行可执行文件

选择 RUN/RUN 或 Ctrl+F9 运行该程序，按 Alt+F5 就可以看到运行结果。如果程序需要输入数据（如例 1.3）则应从键盘输入所需数据，然后程序会接着执行，输出结果。按任意键返回 TC 集成环境窗口。如果发现运行结果不对，要重新修改源程序。

可以用 Alt+X，脱离 Turbo C，回到 DOS 命令状态。此时可以用 DOS 命令显示源程序和运行程序：

C:\MYC>TYPE a1.c✓(列出源程序清单)

C:\MYC>a1✓(执行目标程序 my.exe)

三、Turbo C 的块操作命令

在编辑过程中，可使用块操作命令：

Ctrl+KB(块首定义)，Ctrl+KK(块尾定义)，

Ctrl+KC(块复制)，Ctrl+KY(块删除)，

Ctrl+KV(块移动), Ctrl+KH(块删除)

另外还有一快捷键命令: Ctrl+Y(删除光标所在的一行)

第四节 上机实验

一、实验目的

1. 了解所用的计算机系统的基本操作方法, 学会独立使用该系统;
2. 了解在该系统上如何编辑、编译、连接和运行一个 C 程序;
3. 通过运算简单的 C 程序, 初步了解 C 源程序的特点。

二、实验内容

学习从开机开始的操作, 包括如何建立子目录, 使用操作命令列目录。

调用编辑程序, 输入第一章例 1.1 程序, 并进行编译和运行。应了解所用的系统用什么命令进行编译和连接运行的。编译和连接后所得到的目标程序的后缀(扩展名)是什么。

输入并运行第一章例 1.2 程序。

输入并运行第一章例 1.3 程序。了解如何在运行时向程序变量输入数据。

运行一个自己编写的程序。参照本章例题, 编写一个 C 程序, 输出以下信息:

```
*****  
Very good!  
*****
```

本章小结

本章简单介绍了 C 语言产生的背景、发展现状和基本特点, 并详细介绍了 C 程序的基本结构和一般形式及 Turbo C 集成应用环境的使用和在 Turbo C 中编辑、调试和运行一个 C 程序的具体步骤和方法。

第一章习题

一、选择题

- 1 一个 C 程序的执行是从 ()。
A) 本程序的 main 函数开始, 到 main 函数结束
B) 本程序文件的第一个函数开始, 到本程序文件的最后一个函数结束
C) 本程序的 main 函数开始, 到本程序文件的最后一个函数结束
D) 本程序文件的第一个函数开始, 到本程序 main 函数结束
- 2 以下叙述正确的是 ()。
A) 在 C 程序中, main 函数必须位于程序的最前面
B) C 程序的每行中只能写一条语句
C) C 语言本身没有输入输出语句
D) 在对一个 C 程序进行编译的过程中, 可发现注释中的拼写错误
- 3 以下叙述不正确的是 ()。