

■ 计算机教学通用教材

C
YUYAN
SHEJI
CHENGXU

C 语言程序设计

张向利 马杰 宣士斌 周德新 编



GUANGXI NORMAL UNIVERSITY PRESS

广西师范大学出版社

计算机教学通用教材

C YUYAN CHENGXU SHEJI

C 语言程序设计

张向利 马杰 宣士斌 周德新 编

广西师范大学出版社

· 桂林 ·

图书在版编目 (CIP) 数据

C 语言程序设计 / 张向利等编. —2 版. —桂林: 广西师范大学出版社, 2004. 2

ISBN 7-5633-2391-0

I. C… II. 张… III. C 语言—程序设计—教材
IV. TP312

中国版本图书馆 CIP 数据核字 (2004) 第 004323 号

广西师范大学出版社出版发行

(桂林市育才路 15 号 邮政编码: 541004)
(网址: <http://www.bbtpress.com.cn>)

出版人: 萧启明

全国新华书店经销

广西民族印刷厂印刷

(广西南宁市明秀西路 53 号 邮政编码: 530001)

开本: 787 mm × 1 092 mm 1/16

印张: 17 字数: 390 千字

2004 年 2 月第 2 版 2004 年 2 月第 1 次印刷

印数: 0 001 ~ 5 000 册 定价: 24.00 元

如发现印装质量问题, 影响阅读, 请与印刷厂联系调换。

总 序

随着我国科学技术水平的发展,计算机的应用已日益深入到经济建设的各个部门、各个领域和千家万户。当今国际间的竞争归根到底是人才的竞争,因此培养学生面向现代化、面向世界、面向未来的观点,已经成为各个高等院校的共识。在人才培养规格上,所培养的人才是否具有较强的计算机应用能力,愈来愈为社会各界所关注与重视。

计算机课程已列为高等学校各类专业的必修课,实施计算机教学,目的在于推动计算机知识的普及,促进计算机技术的推广应用,使学生掌握计算机的基础知识和基本操作技能,为推进国民经济和社会信息化服务,为实现富民兴国的新跨越服务。

原广西壮族自治区教育委员会于1993年7月组织了部分计算机专家,参考北京、上海、四川、陕西、浙江、江苏等地的经验,制定了广西普通高校计算机等级考试大纲和样题,并组成了计算机教材编写委员会。1994年起,举办广西普通高校计算机等级考试。其后于1997年并入全国高等学校指导委员会,1994年起,举办广西普通高校计算机等级考试。其后于1997年并入全国高等学校计算机等级考试(又称全国高校联考)。1998年广西教育厅成立了广西高等学校计算机教学指导委员会,作为我区高校计算机教学研究、指导的咨询机构。到2000年为止,广西共有45所学校参加该考试,包括全区所有普通高校和部分中专学校。全国高校计算机等级考试广西考区从广西高校的实际出发,紧密结合教学需要,不断更新教学内容,编写新版教材,改进考试管理办法,开发考试的计算机管理软件,保证了考试的科学性、客观性、公正性,促进了高校计算机教育的普及、计算机设备的投入和计算机师资队伍的建设。所编写的教材除了在全区高校、部分中专学校广泛使用之外,还被其他部门采用。

本套系列教材的编写顺应了当今计算机发展的总趋势,与考试科目相对应,可满足高校学生和各行各业人员学习和掌握计算机应用知识的需要。2000年前已经出版的教材包括:《计算机应用基础》、《计算机应用基础 For Windows》、《FORTRAN77 程序设计》、《PASCAL 程序设计基础》、《TRUE BASIC 程序设计》、《Visual BASIC 程序设计》、《FOX-BASE 应用基础》、《C 语言程序设计基础》、《微机原理及应用》、《计算机软件技术基础》等。最近陆续编写出版的教材有《计算机文化基础 For Window2000》、《Visual FoxPro 应用基础》、《计算机等级考试指南》、《C 语言程序设计》、《C 语言程序设计习题解答与实验指导》等。按照计算机一、二、三级水平考试的要求,本着由浅入深、由表及里的原则,本套系列教材将一步步地引导读者学习和掌握计算机的基础知识、基本理论和基本技能,更好地适应今后的工作需要。

参与各册教材编写的作者,都是长期工作在计算机教学、科研和实验室第一线的计算机专家、教师和教育管理工作。他们在长期的实践中积累了丰富的教学经验。因此本套教材具有层次分明、重点突出、概念明确、实例丰富、使用方便等特点。各册教材在每章后面附有练习与思考题,书末附有上机实验及附录。一册在手,必使你感到运用自如,受益匪浅。

本套系统教材的编写,得到了许多高校和有关专家的大力协助和支持,在此一并表示衷心的感谢。

由于编著者水平有限,本系列教材中的欠缺和不足之处在所难免,欢迎读者批评指正。

车芳仁

前 言

高级语言程序设计课程,曾经采用过 BASIC、FORTRAN、PASCAL 等计算机语言作为程序设计的教学语言。由于 C 语言的优良特性和它的广泛应用,使多数人认为应该首选 C 语言作为教学的语言。当然,对于大学低年级学生,特别是非计算机专业的学生来说,由于缺乏计算机的基本知识,因此首次学习 C 语言有一定的难度。本教材在编写时充分考虑了这一点,在讲清基本概念的基础上,通过大量实例使学生通过应用来掌握 C 语言的基本知识。对于 C 的语法,不求过全、过细,使学生能较快地掌握 C 语言的主要内容并拥有初步程序设计的能力,这是本教材编写的指导思想。

全书共分九章,编写分工如下:周德新第二、八章;张向利第一、三、四章;宣士斌第七、九章;马杰第五、六章。全书由周德新统一审定。

为配合教学,其配套参考书《C 语言题解》一书包含习题解答、部分全国和地方二级 C 语言考题、及有价值的例题分析。

本教材参考并部分引用了国内已有的 C 语言教材,对于他们的劳动,我们表示感谢。

由于编写时间仓促,不足之处,请专家和读者指正。

编者
2004 年 1 月

目 录

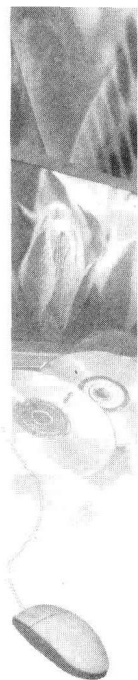
第一章 C 语言概述	(1)
1.1 C 语言的出现及特点	(1)
1.1.1 计算机语言介绍	(1)
1.1.2 C 语言的特点	(2)
1.2 简单 C 语言程序示例	(2)
1.3 C 语言程序的编译和运行	(5)
1.3.1 C 语言程序运行步骤	(5)
1.3.2 Turbo C 的上机过程	(5)
1.4 算法	(7)
1.4.1 算法的概念	(7)
1.4.2 流程图及 N-S 流程图	(9)
习题一	(9)
第二章 数据类型、运算符和表达式	(11)
2.1 概述	(11)
2.2 常量、变量和标识符	(12)
2.2.1 标识符	(12)
2.2.2 常量	(13)
2.2.3 变量	(13)
2.3 整型数据	(14)
2.3.1 整型常量	(14)
2.3.2 整型变量	(15)
2.3.3 整型数据的分类	(15)
2.3.4 整数在内存中的存储形式	(16)
2.4 实型数据	(17)
2.4.1 实型常量	(17)
2.4.2 实型变量	(18)
2.5 字符型数据	(19)
2.5.1 字符型常量	(19)
2.5.2 字符变量	(20)

2.6 类型转换	(21)
2.6.1 隐式(自动)类型转换	(21)
2.6.2 强制类型转换	(22)
2.7 算术运算符和算术表达式	(22)
2.7.1 基本的算术运算符	(22)
2.7.2 运算符的优先级、结合性和算术表达式	(23)
2.8 赋值运算符和赋值表达式	(24)
2.8.1 赋值运算符和单纯赋值表达式	(24)
2.8.2 复合的赋值表达式	(25)
2.8.3 赋值运算中的类型转换	(25)
2.9 自加、自减运算符和逗号运算符	(26)
2.9.1 自加运算符(++)和自减运算符(--)	(26)
2.9.2 逗号运算符和逗号表达式	(27)
2.10 关系、逻辑运算符及关系和逻辑表达式	(28)
2.10.1 C语言中的逻辑值	(28)
2.10.2 关系运算符和关系表达式	(28)
2.10.3 逻辑运算符和逻辑表达式	(29)
2.11 位运算	(30)
2.11.1 位运算符	(30)
2.11.2 位运算符的运算功能	(31)
2.12 应用举例	(33)
习题二	(35)
第三章 语句和程序控制	(39)
3.1 程序的三种基本结构及流程图表示	(39)
3.2 C语言语句	(41)
3.3 数据的输入和输出	(42)
3.3.1 putchar 函数	(42)
3.3.2 getchar 函数	(43)
3.3.3 printf 函数	(43)
3.3.4 scanf 函数	(46)
3.4 if 语句	(48)
3.4.1 if 语句的两种基本形式	(48)
3.4.2 if 语句的嵌套	(52)
3.4.3 条件运算符与条件表达式	(55)
3.5 switch 语句	(56)
3.6 for 语句	(58)
3.7 while 语句	(64)
3.8 do-while 语句	(66)
3.9 break 语句与 continue 语句	(68)
3.9.1 break 语句	(68)

3.9.2	continue 语句	(70)
3.10	goto 语句	(72)
3.11	程序举例	(73)
	习题三	(85)
第四章	函数与程序结构	(90)
4.1	函数及其定义	(91)
4.1.1	无参函数的定义	(91)
4.1.2	有参函数的定义	(91)
4.1.3	空函数	(92)
4.2	函数的参数及函数的值	(92)
4.2.1	函数的参数	(92)
4.2.2	函数的值	(94)
4.3	函数的调用	(94)
4.3.1	函数调用的一般格式	(94)
4.3.2	函数调用的方式	(94)
4.3.3	被调用函数的说明	(95)
4.3.4	函数的嵌套调用	(96)
4.4	递归	(97)
4.5	C 语言程序的一般结构	(97)
4.6	变量的作用域与存储类型	(98)
4.6.1	变量的作用域	(98)
4.6.2	变量的存储类型	(99)
4.7	C 语言预处理程序	(102)
4.7.1	宏定义	(102)
4.7.2	文件包含	(104)
4.7.3	条件编译	(105)
	习题四	(107)
第五章	数组	(111)
5.1	数组的概念	(111)
5.2	一维数组	(112)
5.2.1	一维数组的定义	(112)
5.2.2	一维数组元素的引用	(113)
5.2.3	一维数组的初始化	(114)
5.2.4	一维数组的程序应用举例	(115)
5.3	二维数组	(117)
5.3.1	二维数组的定义	(117)
5.3.2	二维数组元素的引用	(118)
5.3.3	二维数组的初始化	(118)
5.3.4	程序举例	(120)
5.4	数组在函数调用中的应用	(121)

5.4.1	数组元素作为函数实参	(122)
5.4.2	一维数组名作为函数参数	(122)
5.4.3	用二维数组名作函数参数	(125)
	习题五	(126)
第六章	指针	(132)
6.1	指针与地址	(132)
6.1.1	计算机内存的基本组织	(132)
6.1.2	内存中的数据存储单元和地址	(133)
6.1.3	变量地址与指针	(134)
6.2	指针变量的定义	(135)
6.3	指针变量的使用	(137)
6.3.1	指针变量的引用	(137)
6.3.2	指针变量的运算	(139)
6.3.3	指针变量作为函数参数	(145)
6.4	数组与指针	(148)
6.4.1	用指针访问一维数组元素	(148)
6.4.2	一维数组名与指针	(149)
6.4.3	多维数组和指针	(152)
6.4.4	指向数组的指针变量	(158)
6.4.5	用指向数组的指针作函数参数	(161)
6.4.6	指针数组	(168)
6.4.7	main 函数的命令行参数	(172)
6.5	指向函数的指针	(174)
6.5.1	用指向函数的指针变量调用函数	(175)
6.5.2	用指向函数的指针变量作为函数的参数	(177)
6.6	返回指针值的函数	(178)
6.7	指向指针的指针	(180)
6.8	复杂数据说明	(184)
6.9	程序举例	(189)
	习题六	(194)
第七章	字符串	(200)
7.1	字符、字符数组和字符串	(200)
7.2	字符串的输入与输出	(201)
7.2.1	字符串标准输入输出函数	(201)
7.2.2	gets 函数	(202)
7.2.3	puts 函数	(203)
7.3	字符串与指针	(204)
7.4	字符串处理	(206)
	习题七	(208)

第八章 结构体、联合体和枚举类型	(210)
8.1 概述	(210)
8.2 用 typedef 说明一种新类型名	(211)
8.3 结构体类型	(212)
8.3.1 结构体类型的说明	(212)
8.3.2 结构体变量的定义	(213)
8.3.3 结构体变量的初始化	(214)
8.3.4 引用结构体变量中的数据	(215)
8.3.5 函数调用时结构体变量的数据传递	(217)
8.3.6 利用结构体变量构成链表	(220)
8.4 共用体	(225)
8.4.1 公用体类型的说明和变量定义	(226)
8.4.2 共用体变量的引用	(226)
8.5 枚举型	(228)
8.6 应用举例	(229)
习题八	(234)
第九章 文件	(240)
9.1 文件的概念	(240)
9.2 文件指针	(241)
9.3 文件的打开和关闭	(242)
9.4 文件的读写	(243)
9.4.1 文件的字符输入和输出	(243)
9.4.2 文件的字符串输入和输出	(246)
9.4.3 文件的格式化输入和输出	(248)
9.4.4 文件的块输入和输出	(250)
9.5 文件的定位	(252)
9.5.1 rewind() 函数	(252)
9.5.2 fseek() 函数和随机读写	(253)
9.6 返回文件位置指针值	(254)
9.7 文件的出错处理	(254)
习题九	(255)
附录	(256)
附录 1 C 语言的常见关键字	(256)
附录 2 常用字符与 ASCII 代码对照表	(256)
附录 3 运算符的优先级和结合性	(257)
附录 4 库函数	(258)
参考文献	(262)



第一章 C 语言概述

人与人之间需要交流,人与动物之间需要交流,人与机器之间也需要交流。人与人之间可以通过自然语言(例如:汉语、英语)等进行交流,人与动物之间可以通过身体语言等方式进行交流,那么人与机器之间如何交流呢?其实,现实中人与机器的交流是普遍存在的,例如:我们通过遥控器可以控制电视机播放节目,可以通过计算机键盘、鼠标等工具命令计算机做什么,而计算机又通过显示器等将结果告诉我们。无论是人类,还是动物、机器,任何两个独立的个体 A、B 之间能够进行交流的前提大体有下面几种:

(1) A、B 双方共同掌握了一种或多种交流方式。

(2) A、B 之间存在第三者 C。C 既可以与 A 交流,也能够同 B 交流,在 A、B 之间充当翻译。

在此,我们主要讨论人如何与计算机进行交流。我们假设计算机和人一样有耳朵能够听、有大脑能够思维、有嘴能够说话,那么我们人类就能够使用我们的自然语言与计算机交流。但是,科学发展到现在,我们常用的一般的计算机还不能达到这种程度,我们人类要同计算机交流,命令它完成我们的指令,还需要掌握一种机器能够理解(或通过翻译理解)的语言。C 语言就是这样的一种语言,我们掌握了 C 语言,就可以通过编写 C 语言程序,命令计算机完成我们的指令,例如科学计算、事务处理等。

1.1 C 语言的出现及特点

1.1.1 计算机语言介绍

我们人类要同计算机交流,命令它完成我们的指令,需要掌握一种机器能够理解的语言。因为机器只能识别高、低电平,所以我们最初是通过 0、1 代码[分别代表高(低)、低(高)电平]来书写指令的,这种语言我们称为第一代语言,又称为机器语言。

由于用 0、1 代码书写指令存在很多不便,例如:出错率高,可读性差等。于是人们在机器语言中加了一些容易理解的符号来代替纯粹的 0、1 代码,例如:用 ADD 表示加法,用 SUB 表示减法等,这些符号被称为助记符。这种语言我们称为第二代语言,即汇编语言。虽然汇编语言比机器语言容易理解,但开发程序的工作量仍然很大,且可移植性较差。

随着程序规模的不断增大,人们越来越需要与特定计算机系统无关的、更接近于人类自然语言的计算机语言来编写程序,这时就出现了第三代语言,称为高级语言。例如:C、



PASCAL、BASIC、FORTRAN 等。下面介绍 C 语言出现的历史。

C 语言的前身是 ALGOL 语言和 B 语言。20 世纪 60 年代曾经流行一种称为 ALGOL 60 的语言,因为其不能满足发展的需要,英国人马丁·理查德(Martin Richards)在 ALGOL 语言的基础上发明了 BCPL 语言(简称 B 语言)。1972 年贝尔实验室的布朗·W. 卡尼汉和丹尼斯·M. 利奇又在 BCPL 语言的基础上发明了 C 语言,并用 C 语言改写了 Unix 操作系统。1983 年美国标准化协会制订了 ANSI C 标准。目前,PC 机上流行的 C 语言有 Turbo C, Quick C, Microsoft C 等版本,教学中普遍用的是 Turbo C 语言。

1.1.2 C 语言的特点

C 语言由于它的下述特点,被广泛地应用于各个领域的程序设计中。

(1) 语言简洁:C 语言只有 32 个关键字,提供了很多语法供程序员尽量简化程序。

例如:用 $b = a++$ 代替 $b = a, a = a + 1$;

(2) 运算符丰富:共提供了 34 种运算符。运算符多,运算对象的种类也多。

(3) 数据结构丰富:提供了整型、实型、字符型、枚举型、数组、结构体、共用体、指针、空等 9 种以上的数据类型。

(4) 语法限制不太严格。

(5) 允许直接访问物理地址,能进行位操作,能实现汇编语言的大部分功能,可以直接对硬件进行操作。

(6) 生成目标代码质量高,程序执行效率高,C 语言的目标代码的效率仅比汇编语言低 15%。

(7) 可移植性好,基本不做修改就能用于各种型号的计算机和各种操作系统。

1.2 简单 C 语言程序示例

下面介绍几个简单的 C 语言程序,帮助我们初步了解编程的方法和 C 语言的特性。

例 1.1 求两个整数之和。

方法 1:

```
main()  
{  
    printf("100 + 200 = 300\n");  
}
```

运行程序,输出:

100 + 200 = 300

程序中的 main 表示主函数,C 语言的程序必须有一个 main 函数,main 后面的小括弧里存放函数的参数,为空表示不带参数(带参数的 main 函数参照第 6 章)。大括弧括起部分为 main 函数的函数体部分(详细说明参考第 4 章),函数体部分包含说明部分和语句部分,每条说明或每个语句后都有一个分号。本程序的函数体部分包含了一条语句,它的功能是输出双引号中的字符串。“\n”表示回车换行,即输出完本行后将光标移到下一行开始位置。

对于方法 1 所列出的程序,它仅仅完成了 $100 + 200 = 300$ 的显示功能,实际上 $100 + 200 = ?$ 的计算是编程人员算出的,而不是计算机算出的。我们再分析下面的方法。

方法 2:

main()

```
{
    int a,b,sum;      /* 第 1 行:a,b 表示两个相加的数,sum 表示和 */
    a = 100;          /* 第 2 行:给 a 赋值为 100 */
    b = 200;          /* 第 3 行:给 b 赋值为 200 */
    sum = a + b;       /* 第 4 行:求 a + b 并将结果赋值给 sum */
    printf("100 + 200 = %d\n",sum);    /* 第 5 行:输出结果 */
}
```

运行程序,输出:

100 + 200 = 300

程序的函数体有 5 行(第 1 行是说明部分,其余为语句部分),第 1 行定义变量 a,b,sum 为整型数据,第 2 行给 a 赋值为 100,第 3 行给 b 赋值为 200,第 4 行求 a + b 并将结果赋值给 sum,第 5 行输出结果。第 5 行的“%d”表示“十进制整数类型”,程序运行输出时在该位置输出一个十进制整数,本程序中为双引号外的第一个参数 sum 的值(有关 printf 的详细信息参见第三章)。

对于方法 2 所列出的程序,它仅仅完成了 100 和 200 相加的功能,对于任意两个数相加还不能完成。我们再分析下面的方法。

方法 3:

main()

```
{
    int a,b,sum;
    scanf("%d,%d",&a,&b);
    sum = a + b;
    printf("%d + %d = %d\n",a,b,sum);
}
```

本程序的函数体中有 4 行(第 1 行是变量说明,其余为语句),其中用第 2 行 scanf("%d,%d",&a,&b)代替了方法 2 的第 2、3 行。scanf 是 C 语言提供的一个标准输入函数(有关 scanf 的详细信息参见第三章),本程序中它的功能是把键盘输入两个整数分别送给 a 和 b。

程序运行时,用户从键盘输入:

100,200↵

则程序输出:

100 + 200 = 300

若用户从键盘输入:

300,400↵

则程序输出:

300 + 400 = 700

总之,本程序可以完成任意两个整数的加法。

通过上述 3 个不同的程序,我们不仅要了解程序的基本结构,还要了解一些编程的基



本方法。

例 1.2 求 3 个数中的最大数。

```
main() /* 主函数 */
{
    int a,b,c,d; /* 定义变量 a,b,c,d */
    scanf("%d,%d,%d",&a,&b,&c); /* 输入 3 个数送给 a,b,c */
    d=max(a,b,c); /* 调用函数 max,求 a,b,c 中最大的一个,然后放置于变量 d 中 */
    printf("The maximal value is %d\n",d); /* 输出 d 的值 */
}

int max(x,y,z) /* 定义函数 max,求 x,y,z 中的最大值 */
int x,y,z; /* 函数的形式参数 x,y,z 定义为整型 */
{
    int m; /* 定义 max 函数中用到的变量 m 为整型 */
    if ((x>=y)&&(x>=z)) m=x;
    if ((y>=x)&&(y>=z)) m=y;
    if ((z>=x)&&(z>=y)) m=z;
    return(m); /* 返回 m 的值,该值通过 max 带回给 main 函数中作为 max(a,b,c)
    的结果值 */
}
```

程序运行时,用户从键盘输入:

100,200,300↵

则程序输出:

The maximal value is 300

本程序包含了两个函数:主函数 main 和被调用的函数 max。其中 max 的功能就是求 x,y,z 的最大值。程序执行时首先从 main 函数开始,第一步:通过 scanf("%d,%d,%d",&a,&b,&c) 输入 3 个数送给 a,b,c;第二步:通过 d=max(a,b,c) 调用函数 max,进入函数 max 中执行 max 函数体语句,执行完后返回 d=max(a,b,c) 处,并将求出的最大值赋值给 d;第三步:通过 printf("The maximal value is %d\n",d) 输出 d 的值。(有关函数的详细信息参见第四章)。从上面的几个例子中可以看出 C 语言的一些结构特点,归纳起来有下面几点:

(1) C 语言程序是由一个或多个函数构成的,有且仅有一个主函数(main 函数)。

(2) 无论程序中有多少函数,程序的执行总是从 main 函数开始的,即 main 函数是整个程序的入口。

(3) 函数由函数的首部和函数体两部分组成。首部包含函数类型、函数名、函数形式参数名、形式参数定义,如例 1.2 中的

```
int max(x,y,z)
```

```
int x,y,z;
```

就是 max 函数的首部。它的各项对应如下:

int	max	(x,y,z)	int x,y,z;
函数类型	函数名	函数形式参数名	形式参数定义

函数体部分包括说明部分和语句部分,是由最外层的大括弧括起的部分。说明部分可有可无,例 1.1 中的方法 1 就没有说明部分,方法 2 的说明部分为 `int a,b,sum`。语句部分可以没有,也可以为 1 条或多条,例如,例 1.1 中的方法 1 就只有一条语句,而前面介绍的其他程序都有多条语句。

(4) C 语言书写格式自由,一行内可以写几条语句,一条语句也可以分写在多行。

(5) 每条语句和说明的后面必须有一个分号。

(6) 可以用 `/*...*/` 在程序的任何地方做注释。注释部分可以用汉字也可以用英语写,主要是为了便于理解,对编译和运行不起作用。例如,例 1 的方法 2 中就使用了注释对程序进行说明。建议在编写有价值的程序时尽量使用注释。

1.3 C 语言程序的编译和运行

1.3.1 C 语言程序运行步骤

获得 C 语言程序的运行结果需要经过下面几个步骤:

1. 编辑源文件

本阶段完成程序代码的录入和修改。修改完成后可将源程序以 `.c` 为扩展名存盘。编辑 C 程序可以采用 Edit、记事本、vi (Unix 操作系统下的编辑工具) 等,通常采用 C 语言提供的集成环境进行编辑。例如,可以采用 Turbo C 进行编辑。

2. 编译源文件

本阶段完成源程序的语法分析、查错以及翻译源程序生成以 `.obj` 为扩展名的目标程序。目标程序为机器语言,但是还不能执行。

3. 连接、执行程序

本阶段将目标程序与其他目标程序或库连接装配,生成以 `.exe` 为扩展名的可执行程序。可执行程序为机器语言,而且可以执行。C 语言的源程序变为可执行程序的过程如图 1.1 所示:

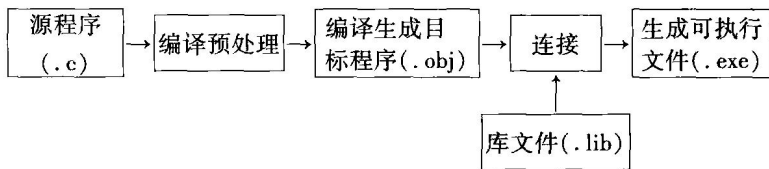


图 1.1 C 语言的源程序变为可执行程序的过程

1.3.2 Turbo C 的上机过程

Turbo C 是一种广泛使用的 C 语言编译系统,它将程序的编辑、编译和调试集成在一个环境中,使用方便、效率高、功能强。下面主要介绍 Turbo C 的上机过程。

1. 系统配置要求

操作系统: DOS

硬盘容量: 约 2M

运行空间: 约 400K RAM

Turbo C 要求很小的系统资源,几乎所有的微机都可以使用。



2. 安装 Turbo C

如果安装盘或光盘上有 install.exe 文件,运行 install  命令就可以安装。若不是可安装盘,可将文件拷贝到下面对应的目录下。

3. 进入 Turbo C 环境

(1) 由 DOS 平台进入 Turbo C

C:\TC > TC.exe  (或其他安装路径)

(2) 由 Windows 平台进入 Turbo C,有下面几种方法:

方法 1:进入 MS-DOS 方式,使用 DOS 命令。

方法 2:通过我的电脑或资源管理器找到 tc.exe,并且双击运行 Turbo C。

方法 3:创建快捷方式,将 tc.exe 放到桌面,方便进入。

4. 建立 Turbo C 工作环境

如果安装采用 Turbo C 的默认目录结构,则可省略该项操作;否则,需要建立 Turbo C 的工作环境:按 Alt + O 键进入 Options 菜单,选择 Directories 项,修改下面的各项路径为安装路径:

Include directories: C:\TC\INCLUDE

包含文件默认目录为 C:\TC\INCLUDE,修改为实际安装路径。

Library directories: C:\TC\LIB

库文件默认目录为 C:\TC\LIB,修改为实际安装路径。

Turbo C directories: C:\TC

Turbo C 的默认工作目录,一般用于存放源程序文件、*.obj 文件、*.exe 文件。可修改也可不修改。

修改完成后,选择 Options 菜单下的 Save Options,保存所做配置。

5. Turbo C 工作环境介绍

(1) 文件操作 FILE:该菜单下可以完成创建新文件(new)、打开已存在文件(Load, Pick),保存文件(Save)、另存文件(Write to)、改变当前工作目录(Change dir)、退出 Turbo C(Quit)等功能。(2) 编辑操作 EDIT:完成插入/修改、查找/替换等对源文件的编辑功能。(3) 编译连接 COMPILE:可完成源程序的编译(Compile to OBJ)、连接(Link EXE File)、一次完成编译和连接(Make EXE File)等功能。(4) 执行 RUN:可完成程序运行(Run)、终止调试(Program Reset)、执行到光标处(Go to cursor)、跟踪执行(Trace into)、单步执行(Step over)和显示程序输出结果等功能。

6. 退出 Turbo C

方法 1:按 Alt + X 热键。

方法 2:按 Alt + F,然后按 Q 键。

7. 调出帮助(Help)信息

方法 1:按 F1 键。

方法 2:按 Ctrl + F1 键。

8. 常用热键

基本操作:

F10——调用主菜单

Alt + F——进入 FILE 菜单



Alt + C——进入 COMPILE 菜单
Alt + R——进入 RUN 菜单
Alt + O——进入 OPTIONS 菜单
F2——存盘
F3——打开
Alt + F9——Compile 编译
Ctrl + F9——Run 运行
Alt + F5——User Screen 查看执行结果用←↑→↓键
PgUp, PgDn——上下翻页
Ctrl + PgUp, Ctrl + PgDn——文件首尾
Home——行头
End——行尾
Delete——删除
Insert——插入
Backspace——退格
Ctrl + Y——删除光标所在行
块操作：
Ctrl + KB——块头标记
Ctrl + KK——块尾标记
Ctrl + KC——块拷贝
Ctrl + KV——块移动
Ctrl + KY——块删除
Ctrl + KH——块隐藏
窗口操作：
F5——激活窗口缩放
F6——激活窗口的切换
F8——Step over 单步调试
F7——Trace into 跟踪调试
F4——Goto Cursor 执行到光标处
Ctrl + F7——Add Watch 在观察窗中输入表达式
Ctrl + F8——Toggle Breakpoint 设置或清除断点
Ctrl + F2——Program Reset 终止调试

1.4 算法

1.4.1 算法的概念

计算机是用于表示和处理数据的工具,用计算机处理实际问题的过程,我们称之为“计算机解题”。计算机解题的基本步骤可以这样描述:

(1) 计算机解题的第一个任务就是对问题的分析讨论和对数据的定义过程,通过该