

实用热处理 模拟技术

王顺兴 刘勇 编著



实用热处理模拟技术

王顺兴 刘 勇 编著



机械工业出版社

本书介绍了渗碳模拟技术、渗碳工艺 CAD、一维热处理模拟技术、组织转变量计算的数学模型和计算方法及全部源程序。

本书选用 Turbo C 语言为编程语言。围绕热处理模拟的实际需要,对 C 语言进行了简洁明了的介绍。程序设计充分体现了结构化设计的特点,化整为零,逐个击破。精心设计了包括中文界面、下拉菜单、参数输入、模型选择、文件管理等各种实用函数,给出了全部源程序。

本书内容深入浅出,通俗易懂,围绕主题逐步深入,源程序系统完整,彻底解决了热处理工程技术人员编写大型、复杂应用程序的困难。本书可供从事金属热处理的工程技术人员使用,亦可供材料科学与工程学科的师生参考。

图书在版编目(CIP)数据

实用热处理模拟技术/王顺兴,刘勇编著. —北京:
机械工业出版社, 2002. 1
ISBN7-111-09452-2

I. 实... II. ①王... ②刘... III. 热处理—模拟
IV. TG15

中国版本图书馆 CIP 数据核字 (2001) 第 071054 号

机械工业出版社 (北京市百万庄大街 22 号 邮政编码 100037)

责任编辑: 余茂祚

封面设计: 姚 毅 责任印制: 付方敏

北京市密云县印刷厂印刷·新华书店北京发行所发行

2002 年 1 月 第 1 版·第 1 次印刷

787mm×1092mm 1/16·12 印张·字数 290 千字

0001—3000 册

定价: 21.00 元

凡购本书, 如有缺页、倒页、脱页, 由本社发行部调换
本社购书热线电话 (010) 68993821、68326677—2527

前 言

随着新世纪的到来,信息革命的浪潮也汹涌而来。计算机技术已成为工业化时代向信息化时代转变的重要技术杠杆,且以惊人的速度改变着人们的生产方式、工作方式、思维方式、学习方式、交往方式乃至生活方式。毫无疑问,计算机的发展与普及,将成为人类科学和文化发展的又一个里程碑。

计算机技术已经深入到制造业的各个过程,大大地改变了传统制造业的面貌。热处理作为机械制造业中十分重要的生产工艺,对保证产品质量起决定作用。传统的热处理是以“理论—实验”为基础的。在理论指导下,靠经验制定热处理工艺,通过反复试验才能确定最终工艺。这种方法不仅周期长,而且人力、财力浪费严重。计算机的发展与普及,为改变这种现状提供了可能。有限差分法、有限元法、计算传热学、热弹塑性理论、相变理论等为改变这种现状提供了坚实的基础。

计算机的应用范围几乎是无所不在,无所不能。计算机在热处理中的应用也是如此。本书仅就热处理过程中的数值模拟作为选题,包括渗碳模拟、温度场和组织模拟。对应力和变形的模拟问题不予考虑。在计算模型中,包括了四个一维模型和三个二维模型。

在内容上,详细介绍了数学模型、计算方法和程序设计三方面内容。

对数学模型,力求从基本问题入手,对物理参数的测试方法做必要的介绍和讨论,并给出至少一组参数值,这样读者就可以进行模拟了。渗碳模型只介绍四种一维模型。温度场模拟除了四种一维问题外,还介绍了四棱柱、圆柱体和空心圆柱体三种二维问题。

在计算方法上,本书选用了简单、易懂、编程容易的有限差分法。

本书的特点之一是给出了完整的计算程序。对绝大多数从事热处理的工作者来说,热处理模拟的最大难点是编程问题。在现有的参考文献中,一般都没有给出源程序,这就给绝大多数读者(专门从事模拟研究者除外)进一步了解带来了困难,非常不利于这一技术的普及和应用。而本书给出了完整的源程序,弥补了这项缺陷。

在编程语言上,选用相对容易理解、移植性好的 Turbo C 语言。程序设计力求做到简洁、规范、通用和便于维护。即使读者对计算模型或计算方法进行修改,也只需对极少数程序进行修改。对 C 语言较熟悉的读者可跳过第 1 章从第 2 章看起,能用 C 语言熟练编程的读者可从第 3 章看起。

本书的特点之二是完整地介绍了提高模拟准确性的思路和方法。为了得到某种可靠的计算公式,首先通过分析研究建立一个比较符合实际(因为完全符合实际的模型还无法建立)的计算模型;根据该模型进行试验,求出模型中的最佳参数;最后就可以用计算模型(含有关参数)进行计算了。这种方法可以称为:模型——试验——应用模型。试验的目的有两个:一是检验模型是否正确,如果不正确,改进模型;二是通过试验得到模型中的参数。如果按照这种方法解决热处理过程中的模拟问题,原则上就不能直接应用现有的物理参数,如比热容、热导率、扩散系数、相变动力学数据(如 TTT 或 CCT 图),尽管这些数据是非常准确的。当使用已有的物理参数(不是根据模型——试验——应用模型这种方法得到的)进行模拟计

IV

算的结果与实际有较大误差时，要么修改模型，要么修改参数值，二者必取其一。本书选用后一种方法——在书中常称为对参数的校正。由于这种方法还在研究中，有关物理参数的校正程序还不够完善，所以本书未给出这方面的程序。

本书在编写过程中得到了洛阳工学院有关领导的大力支持，在此我们表示衷心感谢。我们还要特别感谢机械工业出版社第三编辑室的余茂祚教授级高工对本书的出版所付出的辛勤劳动。

由于作者水平有限，遗漏或不妥之处在所难免，对本书的缺点和错误之处，敬请读者批评指正。另外，需要源程序（已使用 C 语言编程）或可执行文件（未使用 C 语言编程）的读者可与作者联系。

地址：洛阳工学院材料科学与工程系，邮编：471039

电话：0379-4231331， 手机：13017646162

E-mail: wsx@mail.lyit.edu.cn

王顺兴

目 录

前言

第 1 章 C 语言简介..... 1

1.1 C 语言概述..... 1

1.1.1 C 语言产生的背景..... 1

1.1.2 C 语言的特点..... 2

1.1.3 C 语言程序的格式..... 2

1.1.4 PC-DOS 下 Turbo C 上机步骤..... 3

1.2 终端设备上的输入输出函数..... 6

1.2.1 Turbo C 库函数..... 6

1.2.2 字符输入输出函数 `getche()`
和 `putchar()`..... 6

1.2.3 字符串输入输出函数 `gets()`
和 `puts()`..... 7

1.2.4 格式化输入输出函数 `scanf()`
和 `printf()`..... 7

1.3 变量..... 12

1.3.1 标识符..... 12

1.3.2 数据类型..... 12

1.3.3 变量的说明(定义)..... 13

1.3.4 局部变量..... 14

1.3.5 形式参数..... 17

1.3.6 全程变量..... 17

1.3.7 存储类型说明..... 18

1.3.8 数组..... 20

1.3.9 结构..... 22

1.4 常量..... 24

1.5 运算符和表达式..... 25

1.5.1 运算符的分类..... 26

1.5.2 算术运算符及表达式..... 26

1.5.3 关系和逻辑运算符及表达式..... 27

1.5.4 赋值运算符及表达式..... 28

1.5.5 类型转换..... 28

1.6 语句及流程控制..... 30

1.6.1 程序设计中的三种基本结构..... 30

1.6.2 顺序执行语句..... 30

1.6.3 `if` 语句..... 31

1.6.4 `switch` 语句..... 32

1.6.5 `break`、`exit` 和 `continue` 语句..... 33

1.6.6 `for` 语句..... 34

1.6.7 `while` 语句..... 35

1.6.8 `do...while` 语句..... 35

1.7 函数..... 37

1.7.1 返回语句..... 37

1.7.2 函数作用域规则..... 39

1.7.3 函数的形参及调用..... 39

1.7.4 函数递归..... 40

1.7.5 标准函数的使用..... 41

1.7.6 自己建立函数库..... 41

1.8 指针..... 42

1.8.1 指针概念..... 42

1.8.2 指针变量说明及运算..... 43

1.8.3 指针与数组..... 44

1.8.4 指针与函数..... 47

1.8.5 指针与结构..... 51

第 2 章 实用函数设计..... 54

2.1 键盘响应..... 54

2.1.1 `bioskey()`..... 54

2.1.2 键盘响应函数 `specialkey()`
和 `specialkey1()`..... 54

2.2 Mouse 的消息响应..... 55

2.2.1 `mouse_IQR()`..... 55

2.2.2 `init_mouse()`..... 56

2.2.3 `put_mouse_position()`..... 57

2.2.4 `rightb_pressed()`..... 57

2.2.5 `leftb_pressed()`..... 57

2.3 初始化图形系统..... 58

2.3.1	initgraph()	58		和 feof()	72
2.3.2	cleardevice()	58	2.9.3	fscanf()和fprintf()	72
2.3.3	closegraph()	58	2.9.4	fread()和fwrite()	72
2.3.4	winitgraph()	58	2.9.5	坐标参数的输入输出函数 get-	
2.4	将数字转换为字符串	58		coorpar()和 savecoorpar()	73
2.4.1	itoa()	59	2.9.6	曲线颜色输入输出函数 getcolor-	
2.4.2	stpcpy()	59		par()和 savecolorpar()	75
2.4.3	strcat()	59	2.10	图形下的人机对话与提示	76
2.4.4	fitoa()	59	2.10.1	图形模式下屏幕的保存和恢复	76
2.5	图形下的字符输出	60	2.10.2	显示模板 board()	77
2.5.1	settextstyle()	60	2.10.3	提示信息的输出函数	78
2.5.2	settextjustify()	60	2.10.4	警惕性提示信息的输出与	
2.5.3	outtext()和 outtextxy()	60		选择函数 dialogue()	79
2.6	图形下的汉字输出	61	2.10.5	通用的数值输入函数	
2.6.1	open()	61		setparameter()	81
2.6.2	lseek()	61	2.10.6	x、y 坐标参数的输入函数	
2.6.3	read()	62		x_coor()和 y_coor()	84
2.6.4	putpixel()	62	2.10.7	颜色参数输入函数	
2.6.5	close()	62		setcolorpar()	85
2.6.6	outl6hz()	62	2.10.8	设置坐标颜色函数	
2.7	常用画图库函数	64		coor_color()	87
2.7.1	getmaxx()和 getmaxy()	64	2.10.9	设置曲线颜色函数	
2.7.2	setviewport()和			curve_color()	88
	clearviewport()	64	2.11	图形下的菜单设计	88
2.7.3	setpalette()、setcolor()和		2.11.1	主菜单条的显示与选择函数	
	setbkcolor()	64		select_main_menu()	89
2.7.4	setfillstyle()和 floodfill()	65	2.11.2	下拉菜单的显示与选择函数	
2.7.5	setlinestyle()	65		select_menu()	90
2.7.6	line()、lineto()、linerel()		2.11.3	文件菜单的显示与选择函数	
	和 moveto()	66		file_menu()	91
2.7.7	rectangle()、bar()和 bar3d()	66	2.11.4	坐标菜单的显示与选择函数	
2.7.8	circle()、arc()和 ellipse()	66		coor_menu()	93
2.8	一个通用的画坐标函数	67	2.11.5	形体及形体参数	93
2.8.1	画图原理	67	2.11.6	退出函数 wexit()	95
2.8.2	strlen()	68	2.12	模拟主函数	95
2.8.3	画坐标函数 coordinate()	68	2.13	画曲线和刷新曲线函数	97
2.9	外存储器上的输入输出	71	2.13.1	画曲线函数 curve()	97
2.9.1	fopen()和 fclose()	71	2.13.2	刷新曲线函数 renovate()	99
2.9.2	rewind()、fseek()、ftell()		2.14	目录函数和查找文件	99

2.14.1 findfirst()和findnext()	99	4.2.4 I型分布碳势设计	125
2.14.2 unlink()	100	4.2.5 II型分布碳势设计	127
2.14.3 searchpath()	100	4.2.6 III和IV型分布碳势设计	128
2.14.4 输入文件名	100	4.2.7 考虑最后降温的渗碳工艺设计	128
2.14.5 查找文件	101	4.3 渗碳工艺 CAD 程序设计	129
第3章 渗碳模拟技术	104	4.4 用渗碳工艺 CAD 程序校对 物理参数	134
3.1 渗碳数学模型	104	第5章 一维问题热处理模拟 技术	137
3.1.1 扩散方程	104	5.1 热处理模拟技术概述	137
3.1.2 初始条件	106	5.1.1 热处理数值模拟的基本特点	137
3.1.3 边界条件	106	5.1.2 研究现状及存在问题	137
3.1.4 渗碳工艺参数模型	106	5.2 变温相变动力学模型	145
3.1.5 讨论	107	5.2.1 假设与动力学方程	145
3.1.6 渗碳模型小结及参数输入函数 set_car_par()	111	5.2.2 等温转变过程	146
3.2 差分方法	112	5.2.3 变温冷却转变	149
3.2.1 差分基本概念	113	5.2.4 举例	149
3.2.2 差分方程	113	5.3 一维温度场计算模型及 差分方程	150
3.2.3 三对角方程的求解方法	115	5.3.1 计算模型	150
3.3 渗碳模拟函数 carburization()	116	5.3.2 差分方程	151
3.4 碳浓度分布函数 carbon_distribution()	118	5.4 温度和组织模拟计算程序	152
3.5 各种因素对渗碳模拟 结果的影响	120	5.4.1 参数输入及预处理	152
3.5.1 形体和原始含碳量的影响	120	5.4.2 模拟计算程序	153
3.5.2 温度和碳势的影响	121	5.5 温度场分析	157
3.5.3 物理参数的影响	121	5.5.1 温度—时间关系分析程序	157
3.5.4 物理参数的校正原理	122	5.5.2 冷却速率分析	160
第4章 渗碳工艺 CAD	124	5.5.3 温度分布分析程序	162
4.1 渗碳工艺设计的内容和目标	124	5.5.4 模拟举例	163
4.1.1 渗碳工艺设计的内容	124	5.6 组织分析	165
4.1.2 渗碳工艺设计的目标	124	5.6.1 组织转变量与时间关系	166
4.2 渗碳设计方法	125	5.6.2 组织分布	166
4.2.1 对碳势的约束条件	125	5.6.3 温度—组织转变量	167
4.2.2 对温度的约束条件	125	第6章 二维问题热处理模拟 技术简介	170
4.2.3 碳浓度分布基本类型和 描述方法	125	6.1 二维热处理模拟模型	170
		6.1.1 二维模型的一般形式	170

VIII

6.1.2 三个最简单的二维模型	171	附录 B 本书编的实用函数索引	178
6.2 最简单的二维模型差分方程	172	参考文献	180
附录 A 本书使用的库函数索引	176		

第 1 章 C 语言简介

在 PC 机领域，很多著名的软件都是用 C 语言编写的，如：DBASE、FoxBASE、wordstar 等等。C 语言已成为软件开发者最主要的编辑语言之一以及第一流专业程序员最常用的语言。C 语言有如此大的魅力是由其特点和功能所决定的。很多人将 C 语言称为“低级语言的高级形式”或“汇编语言的速记形式”。C 语言的创始者 Ritchie 称 C 语言是一种通用的、较低级的程序设计语言。由此可知，C 语言是一种集高级语言与汇编语言的特点于一体的中级语言。

本章仅仅介绍 C 语言中的常量和常数、运算符和表达式、语句及流程控制、函数和指针等基本内容，以便为阅读后续各章的程序打下基础。

1.1 C 语言概述

1.1.1 C 语言产生的背景

C 语言是 1972 年由美国 Dennis Ritchie 设计，并首次在一台使用 UNIX 操作系统的 DEC PDP-11 的计算机上实现的。它是由一种早期的编程语言 BCPL 发展演变而来的，BCPL 语言目前仍在使用，但主要在欧洲。1963 年，剑桥和伦敦大学为了使 ALGOL 语言得到实用，在此基础上构造了 CPL (Combined Programming Language)，尔后，剑桥大学的 Martin Richards 于 1967 年又对 CPL 进行浓缩，保留较好的基本特性，设计出了 BCPL (Basic Combined Programming Language)，由于它规模小，便于实现和学习，且最大的特点是移植性好，所以目前欧洲一些国家仍在使用。

特别值得一提的是，在当时，计算机中的系统软件都是用汇编语言编写的，尽管当时已经有了高级程序设计语言，但在编写像操作系统这样与硬件密切相关的系统时，高级语言是无能为力的，而使用像汇编这样低级的、非结构化的语言编写较大的系统，其可靠性是无法保证的。比如为 IBM360 机器编写的操作系统，花费 5 千人年，规模为几百万条指令，但在以后的每个版本中总有上千个错误。其他一些较大的系统也都如此，这就是当时所谓的“软件危机”。当时也已萌芽了一些结构化程序设计的思想。20 世纪 60 年代末，N. Wirth 提出了 Pascal 程序设计语言，较好地体现了结构化程序设计的一些原则，但是由于高级语言就是让用户摆脱对硬件环境的依赖，因而对操作系统这样的软件，高级语言尚无法实现。长期从事系统软件设计的 Ken Thompson 和 Dennis M. Ritchie 谙知这一矛盾，他们试图寻求一种解决的方法，因此 Ken Thompson 在 1970 年又对 BCPL 进一步优化，在 PDP-11/20 上实现了一个 B 语言（取 BCPL 的第一个字母），并用它重又编写了 UNIX 操作系统及大部分的实用程序。但是由于 B 语言是一种无类型面向机器字的语言，所以在描述各种数据结构时是很困难的。另外，由于 B 语言最后产生的是解释执行的代码，运行速度较慢。

鉴于这一原因，1972 年 Dennis M. Ritchie 又在 B 语言的基础上添加了许多数据类型，并为其编写了一个实用的编译程序，取 BCPL 的第二个字母，称为 C 语言。随之用 C 语言重新编写了 UNIX 系统，从此使 UNIX 系统产生了强大的生命力。由此可知，C 语言作为一种中级语言恰恰是它的长处所在，它正好解决了用高级语言实现系统软件困难、而用汇编语言实现

时可靠性及移植性都差这一矛盾。

1.1.2 C 语言的特点

C 语言以其简洁、灵活、表达能力强，产生的目标代码质量高，可移植性好为其基本特点而著称于世，详细归纳起来，它具有以下六个特点：

(1) C 语言兼容了其他计算机语言的一些优点，其程序结构紧凑、简洁、规整，表达式简练、灵活、实用。用 C 语言编写的程序可读性强，编译效率高。

(2) C 语言具有丰富的数据类型。在系统软件中，特别是操作系统中，对计算机的所有软件、硬件资源要实施管理和调度，这就要求有相应的数据结构作为操作基础。C 语言具有五种基本的数据类型：字符型、整型、浮点型、双精度型、无值型和多种构造数据类型（数组、结构、联合、枚举）以及复杂的导出类型。C 语言还提供了与地址密切相关的指针及其运算符。指针可以指向各种类型的简单变量、数组、结构和联合，乃至函数等。C 语言还允许用户自己定义数据类型。

(3) C 语言具有丰富的运算符，多达 44 种。丰富的数据类型与丰富的运算符相结合，使 C 语言具有表达灵活和效率高等特点。

(4) C 语言是一种结构化程序设计语言，特别适合于大型程序的模块化设计。C 语言具有编写结构化程序所必需的基本流程控制语句。C 语言程序是由函数集合构成的，函数各自独立，并且作为模块化设计的基本单位。C 语言的源文件，可以分割成多个源程序，分别进行编译，然后连接起来构成可执行的目标文件，为开发大型软件提供了极大的方便。C 语言还提供了多种存储属性，使数据可以按其需要在相应的作用域起作用，从而提高了程序的可靠性。

(5) C 语言是处于汇编语言和高级语言之间的一种中间型程序设计语言，常被称为中级语言。它把高级语言的基本结构和汇编语言的高效率结合起来。因此，它既具有高级语言面向用户、可读性强、容易编程和维护等特点，又具有汇编语言面向硬件和系统、可以直接访问硬件的功能，C 语言的运行效率可以与汇编语言媲美。

(6) C 语言具有较高的可移植性。在 C 语言的语句中，没有依赖于硬件的输入输出语句，程序的输入输出功能是通过调用输入输出函数实现的，而这些函数是由系统提供的独立于 C 语言的程序块，从而便于硬件结构不同的计算机之间实现程序的移植。

由于 C 语言具有上述诸多特点，因此近年来 C 语言迅速得到了广泛的普及和应用。特别是在微处理机和微型计算机的软件开发以及各种软件工具的研制中，使用 C 语言的趋势日益增大，呈现出 C 语言可能取代汇编语言的发展趋势。

本书选用 C 语言作为编程语言还有以下几点考虑：

(1) 对本书的大多数读者来说，对计算机语言不是十分精通，所以用 C 语言还是比较容易理解的（本书中主要用 C 语言中容易掌握的部分就够了）。

(2) 对于已经或准备用 C++ 等面向对象编程的读者来说，C 语言程序可以直接在 C++ 编译器上进行编译，只需将源文件的扩展名更名为 .cpp 或 .cxx。

1.1.3 C 语言程序的格式

任何一种程序设计语言都具有特定的语法规则和一定的表达形式。按照一定的格式和构成规则书写程序，不仅可以使程序设计人员和使用程序的人员易于理解，更重要的是，把程序输入到计算机时，计算机能充分识别，并且正确执行。

在研究 C 语言程序的格式前, 先来看一个用 C 语言编写的求半径为 R 的圆面积的程序。

```
#include "stdio.h"
#define PI 3.14159
void main(void)
{
    float R,S;
    clrscr();
    printf("输入半径值:");
    scanf("%f",&R);
    S=PI*R*R;
    printf("圆面积=%f",S);
}
```

输入半径值: 5

圆面积=78.539749

图 1-1 程序运行结果

该程序运行结果如图 1-1 所示(这是屏幕的左上角部分)。运行程序时先在屏幕上显示“输入半径值:”的提示, 用键盘输入 5, 再按 Enter 键, 接着显示“圆面积=78.539749”, 程序结束。

由上面的程序可以看出, C 语言具有以下格式特点:

(1) C 语言程序习惯上使用英文小写字母书写, 也可以用大写字母, 但是大写字母在 C 语言中通常作为常量(如 PI)或其他特殊用途(如表示半径的 R 和圆面积的 S)来使用。应该注意的是, C 语言对大小写是区分的, 因此 count、Count、COUNT 在 C 语言中代表不同的变量。

(2) C 语言程序是由一条条语句组成的, 每个语句都具有规定的语法格式和特定的功能。上面程序中, float R,S 是变量说明语句; printf() 语句是输出的函数调用语句; scanf() 语句是输入变量数值的函数调用语句; S=PI*R*R 是赋值语句。

(3) C 语言语句不使用行号。

(4) C 语言程序使用“;”号作为语句间的分隔符。

(5) C 语言程序中, 一个语句可以占用多行, 一行也可以有多个语句。不过为了使程序易于阅读, 一般一个语句占一行。

(6) C 语言程序中使用大括号对“{}”来表示程序的结构层次范围。一个完整的程序模块要用一对大括号括起来, 以表示该程序模块的范围。

(7) 为了增加程序的可读性, 可以使用适量的空格和空行。但是, 变量名、函数名和 C 语言保留字中间不能加空格。

(8) 在 C 语言程序中可以加注释, 对初学者建议用中文注释。如果在 C 集成开发环境下输入, 在 Dos 提示符下先运行 PDOS95, 可进入中文平台。

(9) 程序的输出语句中也可以用汉字, 如上例中所示。使用汉字的优点是显而易见的, 但这增大了内存的占用量, 可能会给大型软件带来加载困难的危险, 但由于现在的内存都很大了, 一般来说也不必担心, 所以建议在程序中多用汉字提示。

1.1.4 PC-DOS 下 Turbo C 上机步骤

Turbo C 是美国 Borland 国际公司在 IBM-PC 机上实现的高效、优化的交互式 C 语言集成开发环境。它是集编辑、编译、连接装配功能于一体, 并提供了内含式上下文敏感的帮助功

能。它的优点在于用户不需要脱离该环境就可以将上述的操作过程完成，在操作过程中遇到问题时可以通过帮助信息解决。在 Windows 操作系统下，先进入 MS-DOS，然后运行 PDOS95（如果不使用汉字不要运行 PDOS95），进入 Turbo C，最后键入 TC，按 Enter 键，即可进入 Turbo C 开发环境，如图 1-2 所示。



图 1-2 Turbo C 开发环境

1.1.4.1 编辑建立 C 语言源程序

在进入 TC 后按 F3 键或选择 File 菜单（如图 1-3 所示）中的 Load 项，系统会提示输入一个文件名，如果你输入的是一个新文件名，这时就会出现一个空的编辑屏，用户就可以输入或修改源程序。如果输入的是一个已有的文件名，就会将原有的文件调入。也可以选择 File 菜单中的 New 项，选择后出现一个名为 noname 的空编辑屏。如果进入了 Windows95 中文 DOS 方式，用 Ctrl+空格键进行中英文之间的切换。有关编辑的更多说明请参阅书后的参考书。

1.1.4.2 编译及连接

1. 单个源文件的编译与连接 在 DOS 状态下启动 TC 后，调入已建好的源程序文件（用 File 菜单中的 Load 命令），如果是刚在 TC 环境下建立的源程序则不需要此步骤。当源程序出现在编辑框里后，有几种方法可以完成编译连接。

(1) 选择 Run 菜单（按 Alt+R）中的 Run 命令（如图 1-4 所示）。该命令将编译、连接和运行一次完成。用这种方法的缺点是，如果出现警告，不提示就直接运行。

(2) 选择 Compile 菜单（按 Alt+C）中的 Compile to OBJ 命令（如图 1-5 所示），则系

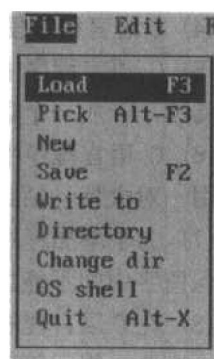


图 1-3 File 菜单

统就会对编辑框中的源程序进行编译，如果源程序的词法语法正确，就会产生一个与源文件名相同但扩展名为“.obj”的目标文件。然后再选择 Compile 菜单中的 Make EXE file 或 Link EXE file 命令对编译产生的 OBJ 文件进行库程序连接装配，最终生成一个文件名与源程序文件名相同，扩展名为“.exe”的可执行文件。

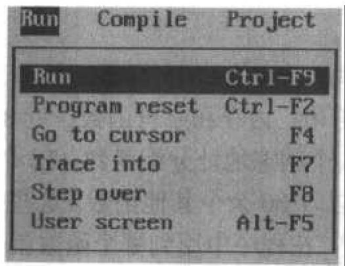


图 1-4 Run 菜单

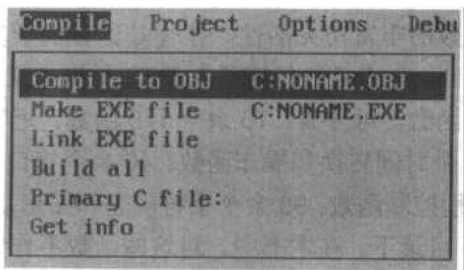


图 1-5 Compile 菜单

2. 多个源程序文件的编译连接 对于一个比较小的程序，源程序放在一个文件内用上述方法进行编译、连接是没问题的。但是，对于一个大的程序，往往需要几个人合编，即使一个人编写，也不可能放在一个源文件中；即使放到了一个源文件中，也存在一些问题，比如增加调试的难度，增加调试时间等。实际上，在用 C 语言编程过程中，最重要的是编好每一个能完成一定功能的子函数。只要每一个函数正确了，调用方法正确，一般就不会出问题。对存放在几个源文件中的程序进行编译连接有两种方法。如果所有的源文件合在一起并不十分大，那么在主函数（main() 函数—每一个程序中必须有一个主函数）所在的源文件中用包含文件命令#include 将其他文件包含进去，然后按照编译单个源文件的方法进行编译连接。例如在 main() 所在文件中加上

```
#include "w1.c"
#include "w2.c"
```

就如同在 main() 函数所在文件中加上了 w1.c 和 w2.c 的内容一样。第二种方法是将每一个源文件先分别进行编译，成功后再由连接装配程序将各自产生的目标文件连接装配成一个可执行程序。这一过程在 TC 中是靠产生一个工程文件来实现的。如上例中 main() 所在文件名为 w.c，首先用编辑器建立一个工程文件，文件名的后缀为“.prj”其内容如下：

```
w.c
w1.c
w2.c
```

然后选择 Project(按 Alt+P)中的 Project name 命令(如图 1-6 所示),系统会提问要工程文件名,输入工程文件名,如 w。最后用 Compile 中的 Make EXE file 命令连接成可执行文件。

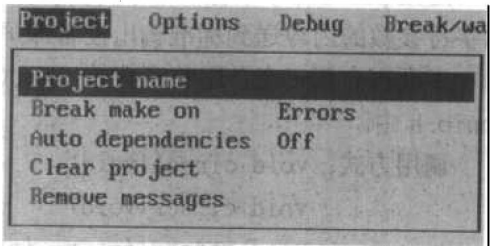


图 1-6 Project 菜单

1.1.4.3 程序的运行

在 DOS 命令状态下或汉字操作系统状态下直接敲入可执行文件名即可。

1.2 终端设备上的输入输出函数

1.2.1 Turbo C 库函数

C 语言编译程序本身实际上是比较容易编制的，然而编制库函数的工作量却很大。在 C 语言函数库中定义了非常丰富的各类函数，并且其中许多函数必须与操作系统连接。正是标准库函数的丰富与灵活性，才使 C 语言优于其他语言。

Turbo C 库函数包括了由 ANSI 标准定义的标准库、已有的 UNIX 标准函数和 Turbo C 所补充的函数。可分为：70 余个 I/O 函数；近 50 个字符串、内存和字符函数；近 30 个数学函数；15 个时间转换和操作函数；40 余个接口函数；15 个动态地址分配函数；12 个目录函数；7 个过程控制函数；90 余个字符屏幕和图形功能函数以及 30 余个其他函数。这些函数都存放在 \lib 目录下。在本书中，对这些函数不作专门介绍，在程序中用到哪个函数就介绍哪个函数。所介绍的函数在附录中列有索引，以便读者快速查找。其他函数的介绍可查阅有关专著。

在 Turbo C 库中有许多函数要与它们自己专门的数据类型、变量和函数的原型一起工作，你的程序也必须访问这些数据、变量和函数原型。这些变量类型和函数原型由编译程序提供的首标文件（头部文件）定义。在任何一个使用这些特定函数的文件中，涉及这些函数的首标文件必须嵌入（用 #include）该文件中，如 #include “stdio.h”。

对每一个库函数，需要掌握其调用方式、函数功能、每个参数的意义和该函数原型所在的首标文件。

1.2.2 字符输入输出函数 getche() 和 putchar()

由于终端设备是用户上机过程中使用最频繁的输入输出设备，故在操作系统中将它们定义为标准输入输出设备，即键盘是标准输入设备，显示器为标准输出设备。体现在程序设计中，就是当使用输入输出函数进行数据传输时，不指明具体设备即表示对标准输入输出设备进行操作。

getche() 和 putchar() 是两个用于单个字符输入输出的标准函数。它们的函数原型在 stdio.h 中。

getche() 没有参数，引用该函数时机器将等待用户从输入设备上键入一个字符并返回其值，同时将接收到的字符自动送回到屏幕的当前光标位置上。

getche() 的一个变体——getch() 函数十分有用，它与 getche() 的惟一差别是不将读入的字符回显在屏幕上。利用这一点来编程序可以避免不必要的显示。例如当程序运行时要求输入一个字符作为口令；在程序结束前按一个任意键再结束（以便看结果）等。

函数 putchar() 带有一个字符型参数，它在功能上与 getche() 相反。它被调用时将所带的字符参数的内容送到标准输出设备即屏幕的当前光标位置。

下面再介绍三个常用的函数 clrerr()、clrscr() 和 gotoxy()。这三个函数的原型都在 conio.h 中。

```
调用方式: void clrerr(void)
           void clrscr(void)
           void gotoxy(int x, int y)
```

函数 clrerr() 清除当前字符窗口中从当前光标位置到该行结束的所有字符，光标位置保持不变。

函数 `clrscr()` 清除整个字符窗口，并且把光标定位在左上角 (1, 1) 位置。当使用字符屏幕时先调用该函数较好。

函数 `gotoxy()` 将字符屏幕上的光标移动到 `x, y` 所指定的位置上，如果其中一个坐标值是无效的，则光标不移动。其中 `x` 是列号，通常取值范围为 1~80；`y` 是行号，通常取值范围为 1~25。

1.2.3 字符串输入输出函数 `gets()` 和 `puts()`

调用方式: `char *gets(char *str)`

`int puts(char *str)`

这两个函数的原型在 `stdio.h` 中。

函数 `gets()` 用来从键盘读入一串字符，并把它们送到 `str` 所指向的字符数组中去。用该函数可以从键盘上接收多个字符并用回车键结束输入过程。回车符并不属于这串字符，由一个空 (‘\0’) 字符放在字符串的最后来代替回车符。打错字符时，可以用光标左移键在回车之前进行修改。如果不成功则返回一个空指针。

函数 `puts()` 把它的字符串变量写在屏幕上并换行。

例 1 下面的程序运行时输入“王顺兴”和“M”，结果如图 1-7。

```
#include "stdio.h"
void main(void)
{
    char str[50], ch;
    clrscr();
    gotoxy(20, 5);
    printf("输入姓名:");
    gets(str);
    gotoxy(20, 6);
    printf("输入性别(M 或 W):");
    ch=getche();
    gotoxy(20, 10);
    printf("姓名: ");
    puts(str);
    gotoxy(20, 11);
    printf("性别: ");
    putchar(ch);
    getch(); }
```

输入姓名: 王顺兴

输入性别 (M 或 W): M

姓名: 王顺兴

性别: M

图 1-7 例 1 结果

1.2.4 格式化输入输出函数 `scanf()` 和 `printf()`

前边的 4 个函数用于字符或字符串的输入输出，但是一般用户在程序中处理的数据类型往往是不同的，比如要输入的是整数或实数，字符或字符串。格式化输入输出函数就是用来实现不同类型数据的输入输出。这两个函数的原型都在 `stdio.h` 中。

1.2.4.1 格式化输出函数 `printf()`

该函数带有参数，其一般的引用格式为：

```
printf("控制字符序列", 参数 1, 参数 2, ...)
```

此函数中参数的第一部分，即双引号括起来的控制字符串是不可少的，它表示在控制字符序列的控制下，对后边的参数进行计算及转换，按指定格式在屏幕上输出结果。

说明：

- (1) 控制字符后边的参数用“,”号分开，它们可以是空、常数、变量或表达式。
- (2) 控制字符串有三种情况：

1) 不含“%”号的普通字符串，这时后边的参数为空，结果是将字符串照原样输出，如例 1 中的 4 个 printf() 语句。在字符串中还可以包括一些特殊的字符符号，它们必须跟在反斜线的后边，称其为转义字符或反斜线字符常量，它们用来表示非图形字符或在使用中容易产生二义性的字符。表 1-1 给出了 C 语言定义的转义字符。其中最常用的是换行符\n。

表 1-1 转义字符表

转 义 字 符	含 义	ASCII 码（十六进制）
\n	换行符	0A
\r	回车符	0D
\b	退格符	08
\t	制表符	09
\f	换页符	0C
\\	反斜线\	5C
\"	双引号"	22

2) 转换说明（格式命令）。转换说明是以百分号%打头后跟一个转换字符组成。表 1-2 给出了 printf() 使用的转换说明。

表 1-2 printf() 使用的转换说明

转 换 说 明	动 作
%d	将参数转换成带正负号的十进制整数输出
%u	将参数转换成无正负号的十进制整数输出
%o	将参数转换成无正负号的八进制整数输出(无前导零)
%x	将参数转换成无正负号的十六进制整数输出(无前导零)
%c	将参数看作单个字符输出
%s	将参数按原字符串输出
%f	将参数转换成[-]mmm. nnn 形式输出
%e	将参数转换成[-]m. nnnnE[±]nn 形式输出
%g	将参数转换成 f 或 e 格式中较短的一个输出
%%	表示百分号本身

例 2 转换说明实例。

```
#include "stdio.h"
```