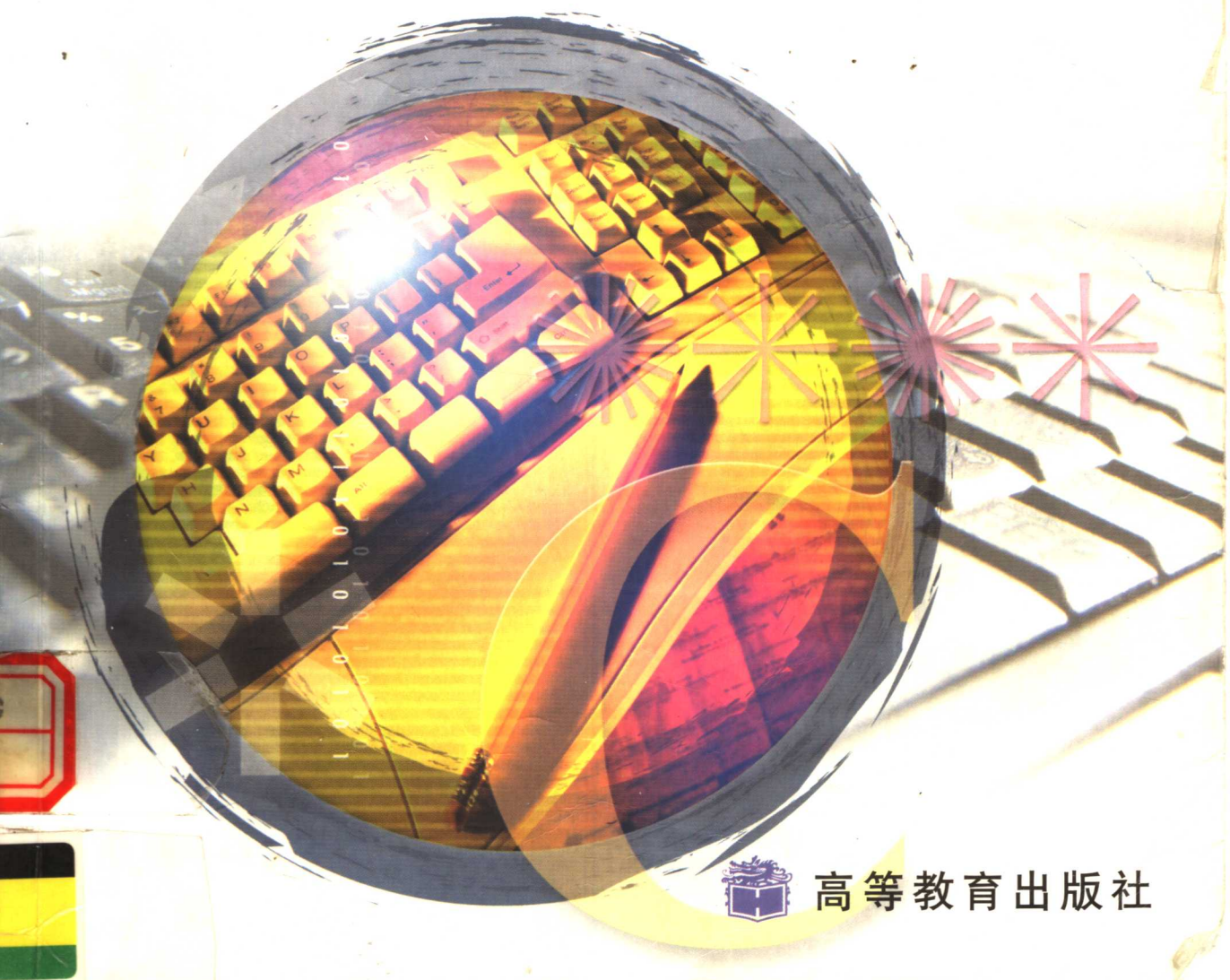


高等职业学校教材

C 语言 程序设计基础

蔡先华 主编



高等教育出版社

高等院校计算机教材

C语言 程序设计基础

蔡先华 主编



高等教育出版社

内容提要

本书以 Turbo C 环境为基础介绍适用于 PC 系列计算机的 C 语言程序设计。全书共分十章, 主要介绍 C 语言概述; 计算机中的数据; 基本输入输出; 程序控制; 函数; 数组; 指针; 结构体与共用体; 文件的打开、读写及关闭; 基本的图形编程方法。书中给出了许多程序实例, 每一章都有学习目标和小结, 并附有较丰富的习题。

本书重点定位在语言使用上, 以程序设计为主线, 对内容做了精心安排, 主要是针对高等职业院校计算机及应用专业的学生编写的, 同时兼顾其他专业学生的使用或供自学者使用, 也可作为没有或只有很少程序设计经验的读者的入门读物或参考书。

图书在版编目(CIP)数据

C 语言程序设计基础 / 蔡先华主编. —北京: 高等教育出版社, 2001.7

高等职业学校教材

ISBN 7-04-008900-3

I. C… II. 蔡… III. C 语言-程序设计-高等教育: 技术教育-教材 IV. TP312

中国版本图书馆 CIP 数据核字 (2001) 第 10543 号

责任编辑 李 波 封面设计 王凌波 版式设计 周顺银
责任校对 陈 荣 责任印制 陈伟光

C 语言程序设计基础
蔡先华 主编

出版发行 高等教育出版社

社 址 北京市东城区沙滩后街 55 号

电 话 010—64054588

网 址 <http://www.hep.edu.cn>

<http://www.hep.com.cn>

邮政编码 100009

传 真 010—64014048

经 销 新华书店北京发行所

印 刷 北京民族印刷厂

开 本 787 × 1092 1/16

印 张 15.5

字 数 360 000

版 次 2001 年 7 月第 1 版

印 次 2001 年 7 月第 1 次印刷

定 价 21.30 元

本书如有缺页、倒页、脱页等质量问题, 请到所购图书销售部门联系调换。

版权所有 侵权必究

前 言

C 语言是程序设计中使用的最广泛的语言之一,无论是计算机专业还是非计算机专业的人士都喜欢使用。几乎所有学校的计算机应用专业都开设与 C 语言相关的课程。有些学校已把 C 语言作为学生学习程序设计的入门语言,这主要是因为 C 语言表达能力强、使用灵活方便、应用面广、目标程序效率高、可移植性好,既具有高级语言的优点,又具有低级语言的许多特点。

本书分为两大部分,共 10 章。第一部分 1~5 章,是 C 语言的基本内容。通过学习,要求掌握 C 语言的基本内容,包括基本数据类型、控制结构、表达式和函数,以及编译预处理和 C 的一般调试技术。学习了这些内容后,应能编写、调试和运行小型的应用程序,并对结构化程序设计方法有所了解,为进一步学习打下基础。第二部分 6~10 章,是 C 语言的高级编程技术,包括高级数据类型:指针、结构体、共用体和文件的概念、相互关系和一般使用。学习这部分内容之后,学生应理解和掌握有关的概念,具备运用高级数据结构和编程技术来编制较大型应用程序的能力。

本书主要是针对高等职业院校计算机及应用专业的学生编写的,同时兼顾其他专业学生的使用。本书具有如下特点:

(1) 本书的学习重点定位在语言使用上,教会学生如何正确运用 C 语言编写程序,所举例题和习题主要是帮助学生学会如何使用 C 语言,对需要较高数学基础的计算机算法问题不作要求。

(2) 可作为学习计算机程序设计的入门语言。本书力图讲清楚各种基本概念和观点。许多学校已把 C 语言作为学习计算机程序设计的入门语言,针对这种情况,在每一章的编写过程中,考虑初学者学习语言的困难,对重点、难点精讲、细讲,提供合适的例题进行分析。

(3) 以提出问题、探索解决问题的方法、编写程序为线索编排内容。好奇心是人的天性,特别是十多岁的学生,急于探索解决问题的方法。通过提出问题,在探索解决问题的方法中讲解程序设计的原理、方法,更有利于提高学生学习积极性。

(4) 通过程序的书写、文字的叙述指导学生如何写出清晰、简洁的程序。C 语言是一种不太严格的语言,稍不注意,就会隐含不易发现的错误。书写规范格式的 C 语言程序有利于发现错误。为了培养学生书写规范格式程序的习惯,书中程序用 ANSI C 提倡的程序写法,同时提醒学生应该如何写而不应该怎么写。

(5) 不死抠每一个细节,而把主要精力放在最基本、常用的部分。语言正朝向简单化、

方便程序设计的方向发展,作为初学者不必死抠语言的细节或过分强调使用技巧。故本书的目标主要放在讲解程序设计的基本技能上。

(6) 为了适应社会的需要,各校都强调学生具有一张文凭,多张证书。许多学校要求学生在学习了 C 语言后能通过全国计算机等级考试的二级考试。因此,本书在编写过程中适当参考了全国计算机等级考试二级考试大纲。

本书由蔡先华任主编,第 1 章、第 8 章 1~4 节、第 10 章由蔡先华编写,第 2 章、第 4 章由江宁升编写,第 3 章、第 7 章、第 8 章 5~9 节由张远编写,第 5 章由江春编写,第 6 章、第 9 章由张卫编写。王乃国老师审阅了全书,提出了许多宝贵意见,在此表示感谢,本书编写过程中得到了唐瑞庭、徐悦等老师的大力支持,在此一并感谢!

由于时间仓促,错漏之处在所难免,恳请广大读者批评指正。

编者

2001 年 5 月

目 录

第1章 概述	1	2.4.4 关系运算与逻辑运算	23
1.1 人与计算机的交流	1	2.4.5 位运算	25
1.2 C 语言的由来与特点	2	2.4.6 sizeof 运算符	27
1.2.1 C 语言的由来	2	2.4.7 复合运算符	28
1.2.2 C 语言的特点	3	2.5 不同类型数据间的转换	28
1.3 Hello C	4	2.6 小结	32
1.4 Turbo C 环境中程序的编译连接 与运行	5	2.7 习题	33
1.4.1 Turbo C 集成环境	6	第3章 基本库函数	36
1.4.2 Turbo C 的基本操作	6	3.1 字符输入输出函数	36
1.4.3 编辑 C 程序	6	3.2 格式输入输出函数	38
1.4.4 编译、连接	7	3.2.1 格式输入函数	38
1.4.5 运行	8	3.2.2 格式输出函数	40
1.5 小结	9	3.3 数学函数及应用	45
1.6 习题	9	3.4 小结	46
第2章 计算机中的数据	10	3.5 习题	47
2.1 计算机中的数制与码制	10	第4章 程序控制	49
2.1.1 数制	10	4.1 选择语句	49
2.1.2 数制转换	11	4.1.1 if 语句的一般格式	50
2.1.3 码制	13	4.1.2 条件运算符	54
2.2 基本数据类型	13	4.1.3 switch 语句	55
2.2.1 基本数据类型及其取值范围	13	4.2 循环程序设计	57
2.2.2 数值型数据的表示及存储	15	4.2.1 goto 语句以及用 goto 语句构成 的循环	57
2.2.3 字符型数据	15	4.2.2 循环结构的基本形式	58
2.3 标识符、常量和变量	17	4.2.3 while 循环	59
2.3.1 标识符	17	4.2.4 do-while 循环结构	60
2.3.2 常量的表示	18	4.2.5 for 循环	61
2.3.3 变量的表示、定义及初始化	19	4.2.6 循环的嵌套	63
2.4 运算符、表达式及表达式的值	20	4.2.7 break、continue、return 和 exit 语句	64
2.4.1 赋值表达式	20	4.3 程序举例	67
2.4.2 算术运算符与算术表达式	21	4.4 小结	71
2.4.3 逗号运算符与逗号表达式	23		

4.5 习题	72	6.3.3 多维数组的应用	140
第5章 函数	79	6.3.4 多维数组的应用举例	140
5.1 函数的定义和函数的声明	79	6.4 数组名作为函数的参数	142
5.1.1 函数的概念	80	6.5 小结	144
5.1.2 函数的定义	82	6.7 习题	144
5.1.3 函数的声明	84	第7章 指针	147
5.2 函数的调用和值的传递	86	7.1 指针的定义及使用	147
5.2.1 函数调用的一般形式	86	7.1.1 指针概述	147
5.2.2 传值调用	88	7.1.2 指针变量的定义及赋值	148
5.2.3 嵌套调用	90	7.1.3 指针的基本运算	150
5.3 函数递归	92	7.2 指针与数组	151
5.4 变量的存储类型及作用域	96	7.2.1 一维数组的指针及数组元素的 访问	151
5.4.1 自动变量 auto 及其作用域	96	7.2.2 二维数组的指针及数组元素 的访问	154
5.4.2 寄存器变量	98	7.2.3 字符串指针	156
5.4.3 外部变量及其作用域	100	7.3 指针作为函数的参数	157
5.4.4 静态变量及其作用域	103	7.3.1 变量的指针作函数参数	157
5.4.5 存储类别小结	105	7.3.2 指向数组的指针变量作函数 参数	160
5.5 内部函数和外部函数	106	7.3.3 字符串指针作函数参数	161
5.5.1 内部函数	106	7.4 指针数组	163
5.5.2 外部函数	107	7.4.1 指针数组	163
5.6 编译预处理	110	7.4.2 指向指针的指针	164
5.6.1 宏定义	110	7.4.3 指针数组作为函数参数	165
5.6.2 文件包含	116	7.5 指向函数的指针	166
5.6.3 条件编译	117	7.5.1 函数的指针	166
5.7 小结	121	7.5.2 指向函数的指针变量	167
5.8 习题	122	7.5.3 用函数指针实现函数的调用	167
第6章 数组	125	7.6 main()函数的参数、函数的类型和 返回值	168
6.1 数组的定义及使用	125	7.7 小结	170
6.1.1 数组变量定义	125	7.8 习题	171
6.1.2 数组的初始化	126	第8章 结构体与共用体	174
6.1.3 数组的使用	127	8.1 结构体类型数据	174
6.1.4 数组的存储实现	128	8.1.1 简单的人事档案问题	174
6.1.5 数组的应用举例	128	8.1.2 结构体变量定义	175
6.2 字符数组和字符串	130	8.1.3 结构体类型变量引用	178
6.2.1 字符数组的定义和初始化	131	8.1.4 结构体类型变量初始化	179
6.2.2 字符串	131	8.1.5 结构体数组	180
6.2.3 字符串的输入与输出	132	8.2 共用体类型数据	183
6.2.4 字符串处理函数	134	8.2.1 共用体类型变量定义	183
6.2.5 字符数组的应用举例	136		
6.3 二维和 multidimensional 数组	138		
6.3.1 多维数组变量定义	138		
6.3.2 多维数组的初始化	139		

8.2.2 共用体类型变量引用方式	184	9.3 文件的读写	213
8.3 枚举类型数据	184	9.3.1 fputc 函数和 fgetc 函数	213
8.4 取个别名——typedef 语句	187	9.3.2 fread 函数和 fwrite 函数	215
8.5 指向结构体类型数据的指针	188	9.3.3 fprintf()函数和 fscanf()函数	217
8.5.1 指向结构体元素的指针	188	9.3.4 其他读写函数	219
8.5.2 指向结构体数组的指针	190	9.4 文件的定位函数	220
8.6 结构体作为函数参数	191	9.4.1 反绕函数 rewind()	220
8.6.1 结构体成员作函数参数	191	9.4.2 随机定位函数 fseek()	221
8.6.2 整个结构体变量作函数参数	192	9.4.3 求当前读写位置函数 ftell()	221
8.6.3 指向结构体的指针作为函数参数	193	9.5 出错检测	222
8.7 结构体作函数返回值	194	9.5.1 ferrord()函数	222
8.7.1 结构体型函数	195	9.5.2 clearerr 函数	222
8.7.2 返回结构体指针的函数	196	9.6 文件系统和流	222
8.8 动态存储分配与内存管理	197	9.6.1 文件系统	222
8.8.1 数据在内存的存储分配	197	9.6.2 流的概念	223
8.8.2 与内存管理有关的函数	197	9.7 小结	223
8.9 牵着手的数据——链表	199	9.8 习题	224
8.9.1 链表概述	199	第10章 Turbo C图形编程基础	225
8.9.2 链表的建立	200	10.1 显示卡及显示器模式的操作	225
8.9.3 链表的查询	203	10.2 字符屏幕操作	225
8.9.4 链表的插入	204	10.2.1 窗口	226
8.9.5 链表的删除	206	10.2.2 基本输入与输出	226
8.10 小结	207	10.2.3 屏幕操作	226
8.11 习题	208	10.2.4 字符属性控制	227
第9章 文件	210	10.3 Turbo C 的图形操作	227
9.1 文件、文件类型指针	210	10.3.1 设置和关闭图形模式	227
9.1.1 文件的概念	210	10.3.2 绘制图形	229
9.1.2 文件类型指针	210	10.3.3 图形模式下的字符输出	230
9.2 文件的打开与关闭	211	10.3.4 演示程序	231
9.2.1 文件的打开函数 fopen()	211	10.4 小结	232
9.2.2 文件的关闭函数 fclose()	213	10.5 习题	232
		附录A ASCII码表	233
		附录B 常用Turbo C 库函数	235

第1章 概述

【学习目标】

本章主要介绍程序设计语言 C 语言程序的基本结构，在计算机上输入、编译、连接、调试和运行 C 程序的基本方法和步骤。通过本章的学习，应掌握程序设计语言的基本概念，了解 C 语言程序的基本特点，熟悉 C 程序的编辑、编译、调试和运行的基本过程。

1.1 人与计算机的交流

什么是程序？什么是程序设计语言？可能是大多数初学程序设计的人首先遇到的问题。语言是人们交流信息的一种媒介，人类使用的汉语、英语、俄语等都是自然语言。这些自然语言是随着人类社会的发展和进步而逐渐形成的，是人类相互之间交流信息的工具。人们用口头语言向别人传递消息、传播所见、表达自己的思想；用书面语言撰写文章、书籍。科学技术的发展使人类制造出了各种各样的机器，人不仅要与人类自身进行交流，还必须同为人类服务的各种机器进行交流。比如人与电视机，人们通过遥控器向电视机发送信息，电视机根据接收的信息完成转换频道、放大音量、调整色彩等操作，并在电视屏幕上向人显示当前状态信息。人与电视机的交流是一种简单的交流。计算机是一种特殊的机器，人们通过与计算机进行交流，让计算机为我们提供各种各样的服务，而人与计算机的交流要远比人与电视机的交流复杂得多。为了让计算机处理不同的问题，人们发明了不同的与计算机交流的语言，人们通过这些语言与计算机进行交流，让计算机为我们服务。早期人们只用机器语言与计算机交流。

从根本上讲，计算机只能接受和处理由二进制代码组成的指令。每一种型号的计算机在设计时由设计者规定好一套指令系统，确定用什么指令来完成什么操作，这种计算机能接受的代码称为机器指令。它是面向机器的。机器语言是机器指令的集合。要让计算机产生一系列的操作，应当写出一条一条的机器指令，为解决某一问题而设计的一段指令序列称为程序。

由于机器语言指令不易记忆，容易出错，人们又发明了符号语言。符号语言比机器语言前进了一步，它用符号代替二进制码。例如用“ADD”代表“加法”，用“MOV”代表

“传送”，一条符号语言指令对应于一条机器指令。但是，计算机不能直接执行符号语言程序，必须一条一条地把它“翻译”成机器指令，人们编制了一种程序让计算机来完成“翻译”工作，这个“翻译”程序称为符号汇编程序或汇编程序。

符号语言和机器语言被称为低级语言，对初学者来说，用符号语言编写程序是一件比较困难的事情。为了让计算机语言不仅计算机可以“懂得”，大多数人也能够懂得它、掌握它，人们创造了“高级语言”，又称“计算机程序设计语言”或“编程语言”。本书后面常把它们简称为“程序语言”或“语言”。它不是面向机器的，而是面向要解决的问题的，即不依赖于具体机器。用高级语言写的程序可以适用于多种类型的计算机。高级语言与人们习惯使用的自然语言和数学语言比较接近，因而人们容易理解和使用。显然，用高级语言写程序比用低级语言写程序容易得多，任何一个具有中学以上文化程度的人都能很快学会和使用它。

目前已经出现了几百种高级语言，它们各有不同的语法规则和用途。使用得比较广泛的语言主要有 BASIC、C、C++、Pascal、Java 等，这些语言通常被认为是常规的语言，因为它们有许多共同的性质。另外还有一些比较特殊的语言，在形式、编程方式等方面与常规语言差别很大，通常相互之间也大相径庭。这些非常规的语言一般各有各的特点或者应用领域，甚至有着特殊的使用人群。属于这类语言有 LISP、Smalltalk、PROLOG、ML 等。这些语言虽然不如常规语言使用广泛，但也是非常重要的。

计算机不能直接接受高级语言程序，必须首先翻译成机器指令。这项工作比汇编语言“翻译”复杂得多。将高级语言程序(称为源程序)转换为机器语言程序(称为目标程序)的工作由“编译程序”来完成。

并不是任何计算机在任何时候都能运行任何一种语言编写的程序，如果想在—台计算机上运行一个 C 语言程序，必须首先保证该计算机有 C 的编译程序。

1.2 C 语言的由来与特点

1.2.1 C 语言的由来

C 语言的产生与 UNIX 系统的发展有密切的关系。UNIX 系统是一个通用的、复杂的计算机操作系统。它的内核最初用汇编语言编写。汇编语言的主要特点是能充分体现计算机硬件指令的特性，生成的代码具有较高的质量。但其可读性、可移植性以及描述问题的能力远不如高级计算机语言。能否研制一种既有汇编语言的基本特性又能克服一般高级计算机语言某些缺点的新语言以进行系统软件的设计呢？

1973 年，美国贝尔实验室的 Dennis Ritchie 成功地设计了称为“C 语言”的程序设计语言，设计的目的是用于书写操作系统和其他系统程序，并首次用来写 UNIX 操作系统。C 语

言在当时的 PDP—11 计算机上运行。它具有汇编语言的一些特性，同时又具有高级语言的特性。20 世纪 70 年代后期，C 语言作为 UNIX 系统上标准的系统开发语言，随着 UNIX 系统的流行而被越来越广泛地接受和应用。以后，C 语言被逐渐移植到包括大型计算机、工作站等在内的其他机型的操作系统上，逐渐成为各种计算机上开发操作系统和复杂软件系统的一种通用语言。1983 年美国标准化协会(ANSI)制订了一套 ANSI 标准，成为现行的 C 语言标准。

目前，在世界上使用最多的微型计算机程序编译系统中，已经有了许多性能卓越的商用 C 语言系统。如 PRICE(原 Borland)公司早期的 Turbo C 系列和后续的 Borland C / C++ 系列 Borland C++ Builder 系列产品；微软(Microsoft)公司的 Microsoft C 和后续的 Visual C / C++ 系列等产品，还有其他一些使用较广泛的 C / C++ 系列产品，如 Watcom C / C++ 系列和 Symantic C / C++ 系列等。

C 语言种类非常多，但从教学入门的角度来看，Turbo C 有许多优点，因此本书以 Turbo C 2.0 为基础介绍 C 语言。掌握了 Turbo C 之后，就会较容易地向其他版本的 C 语言乃至 C++ 过渡。

1.2.2 C 语言的特点

C 语言之所以能够被广泛接受，是由于它有许多的不同于其他计算机语言的特点。C 语言的主要特点包括：

(1) C 语言程序紧凑、简洁。C 语言的定义比较简单，是一个比较小的计算机语言。这里的“小”不是说 C 语言功能小，而是说对于初学者不需要学习很多先修知识就可以开始编写程序，提高了初学者学习的兴趣。C 语言把许多程序设计中需要的功能放在程序库(在 C 语言中称为“函数库”)里实现，例如程序的输入输出功能等，这样就使语言本身比较简洁，编译程序比较容易，写出的程序紧凑。

(2) 使用时方便、灵活。C 语言有多种运算符，多种表达式求值方法，多种描述问题的途径。因此，程序设计者有较大的主动性，使用时更加灵活。

(3) 范围广而丰富的运算符。丰富的运算符使 C 语言的运算类型非常丰富，表达式类型多种多样，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。

(4) 丰富的数据结构。C 语言有 5 种基本数据类型和多种构造类型以及复杂的导出类型，能实现各种复杂的数据结构运算。尤其是指针类型数据，使用起来灵活、多样。

(5) C 语言程序具有较好的可移植性。不作(或稍作)修改就能用于各种型号计算机和各种操作系统。

(6) C 语言是结构化程序设计语言。C 语言具有编写结构化程序所需的结构化控制语句，用函数作为程序模块以实现程序的模块化，是写结构化程序的理想语言，符合现代编程风格要求。

(7) C 语言是高、低级兼容语言。C 语言是比较靠近硬件和系统的语言，它与汇编语言比较接近。C 语言既有面向硬件和系统，像汇编语言那样可以直接访问硬件的功能，又有高级语言面向用户，容易记忆，便于阅读和修改的优点。

(8) C 语言可以写出执行效率很高的程序。人们之所以在许多地方用汇编语言，一个重要原因就是高级语言写出的程序效率相对较低。这样，要开发一些效率要求特别高的程序时只能用汇编语言。有了 C 语言之后情况发生了变化，C 语言可以像汇编语言那样直接访问硬件，可以生成质量很高的目标代码。

同样，C 语言也存在着不足。最初设计 C 语言的主要指导思想是把它作为汇编语言的一个替代品，首先是作为设计者自己用来写操作系统的语言工具，因此强调更多的是灵活性和方便性。这样发展起来的语言对于程序形式要求不很严格，给写程序的人留下了很大余地，使人可以用许多不太合规矩的方式写程序。这样写出的 C 语言程序里会留下了许多不安全因素，C 语言不提供很多帮助，要求写程序者自己保证程序的安全，这对专业程序员来说不是太难的事。但随着 C 语言应用的广泛，使用者越来越多，由于不少人对 C 语言的理解远不如设计者，写出的程序里常常带有隐藏很深的错误。

1.3 Hello C

C 语言写出的程序是什么模样呢？这一定是读者很想知道的。下面来看一个简单的 C 语言程序例子——Hello C。然后从 Hello C 这个最简单的程序出发，解释编写程序的一些基本问题。

```
/* Hello.c is an example of C program */
#include <stdio.h>

main()
{
    printf("Hello C!\n");
}
```

编译并运行这个程序得出结果：

Hello C!

这是一个非常简单的程序，因为是用 C 语言写出来的，所以称为 C 语言源程序，简称“C 程序”。Hello C 共有七行。其中第一行为注释行，用/*和*/括起来的部分是注释。注释可写在程序的任何地方，它只为阅读程序提供方便，没有其他意义。从第二行开始，程序分为两部分。第二行为一部分，它是一个特殊行，说明在这个程序里要用到 C 语言系统提供的一些标准功能，因此需要引用输入输出函数库文件 stdio.h。第三行为空行，目的是与下一部分隔开。后面的四行是程序的基本部分，这部分描述程序所要完成的工作。这里是在屏幕上

输出“Hello C!”。

在这个 C 程序中，`main()` 是主函数，C 程序都是从主函数开始执行的，一个 C 程序有且仅有一个主函数。`main()` 后面部分用一对花括号 `{}` 括起来，它们是函数体的开始与结束，花括号可称为是函数体的界限符，`printf()` 是 C 的一个库函数。

C 语言是一种“自由格式”的语言，除了一些简单的限制外，写程序时可完全根据自己的爱好去书写。如：在哪里加一些空格，在哪里换行等等。不论用什么格式写程序都不影响程序本身的意义。

C 程序的书写虽然没有严格的格式规定，但为了便于阅读，在书写时应按照阶梯缩进式书写。`Hello.C` 虽然很短，但它是按照阶梯缩进式书写格式书写的。我们可以总结出阶梯缩进式的如下特点：

(1) 一般情况下每条语句占用一行。

(2) 程序不同部分用空行隔开。如 `#include <stdio.h>` 后空一行与下一部分隔开。目的是分隔程序中处在同一层次的不同部分。

(3) 不同层次的语句，从该层的起始位置开始缩进。在同一层次中的语句，缩进同样多的字符数。在输入源程序时，可以用空格键或 `TAB` 键调整各行的起始位置。如“`main()`”行的第一个字母与“`printf("Hello C!\n");`”行第一字母相差一定数的空格。

(4) 在程序里增加一些必要的说明信息，以增加程序的可读性。如 `Hello.c` 中注释 `/*Hello.c is an example of C program */` 是说明这个程序是一个简单的 C 程序例子。

采用阶梯缩进格式书写的程序，既充分体现了结构化程序层次清晰的特点，又十分便于阅读和理解程序。初学者因为开始写的一些程序比较小，容易看懂，往往不注意书写格式，但稍大一点的程序如果不注意书写格式往往给自己或别人阅读程序带来麻烦。

1.4 Turbo C 环境中程序的编译连接与运行

C 语言是一种高级程序设计语言，比较容易书写和阅读。用 C 语言写出的程序通常称作“源程序”，Turbo C 源程序代码文件必须以后缀“`.C`”结束，也就是 C 源程序的文件属性为“`.C`”。

由于计算机只能识别和执行具有特定二进制代码形式的机器语言程序，不能直接执行源程序，为使计算机能够完成某个 C 语言程序所描述的工作，必须首先把这个源程序转换成二进制代码表示的机器语言程序，Turbo C 就是一种将 C 源程序转换成二进制代码机器语言的程序语言系统，将 C 源程序转换成二进制代码的机器语言程序分两步进行，第一步进行编译，第二步进行连接。在第一步，由编译程序对源程序文件进行分析和处理，生成相应的机器语言目标模块，由目标模块构成的代码文件被称作“目标文件”，目标文件名以后缀“`.obj`”结束，也就是目标程序的文件属性为“`.obj`”。在第二步，由连接程序将目标文件与其他一些必要组成部分组合在一起，构成一个完整的可执行程序。可执行程序的文件名以

“.exe”结束。

1.4.1 Turbo C 集成环境

Turbo C 不仅是一个快速、高效的编译程序，同时还带有一个易学、好用的集成开发环境。使用 Turbo C 时无需独立的编辑、编译和连接程序，就能编辑和运行 C 程序，所有这些功能都组合在 Turbo C 的集成环境内，并且可以通过主屏幕菜单选择使用这些功能。

在配置有 Turbo C 集成开发环境的计算机系统上，若要启用 Turbo C，只需进入 Turbo C 所在的子目录(Turbo C 的缺省子目录是 TC)，在 DOS 提示符下键入 TC 并按下 Enter 键，即可进入 Turbo C 集成开发环境。此时初启屏幕包括主菜单和版本信息。按任意键，版本信息消失，屏幕上留下如图 1-1 所示的主屏幕。

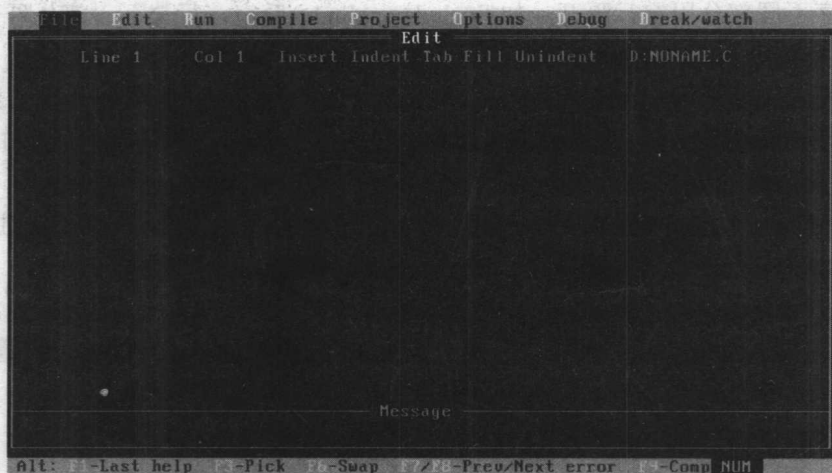


图1-1 Turbo C的主屏幕

1.4.2 Turbo C 的基本操作

同时按下 Alt 键和高亮度的大写字母可选择一个菜单项，例如用热键 Alt+E，则进入编辑状态。也可以用←键或→键把高亮度条移到相应选择项，按下 Enter 键进入相应选择状态。按 Esc 键退出一层菜单。在任何时刻按一下 F10 都将返回主菜单。

1.4.3 编辑 C 程序

在开始编辑 C 程序时，首先应用“Load”装入一个准备编辑的 C 程序，或用“New”编辑新文件。这两个子菜单项都在“File”子菜单中。如图 1-2。

1. 装入已存在的 C 程序

在主菜单状态下，将高亮度条用→键或←键移到左上角的“File”处，然后按 Enter 键，屏幕出现“File”子窗口。在“File”菜单中将高亮度条用↑键或↓键移到“Load”处，按

Enter 键选择此命令,或在出现主菜单窗口时直接按“F3”键,这时出现一个包含“*.C”提示框。用户可输入要编辑的文件名,按 Enter 键即可。也可在出现包含“*.C”提示框时直接按 Enter 键。这时,Turbo C 就会显示出当前目录下的所有后缀为“*.C”文件名,利用↑键或↓键将高亮度条移到需要编辑的文件名处,接 Enter 键后该文件内容即显示在屏幕上,用户可对它进行编辑、修改等操作。

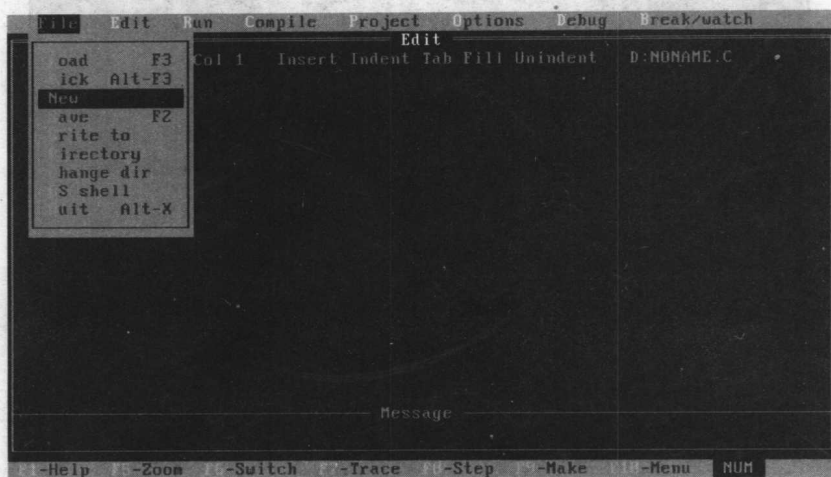


图1-2 Turbo C File子菜单

2. 开始新文件的编辑

在主菜单状态下,进入“File”子菜单,再将高亮度条移到“New”处,按 Enter 键选择此菜单项,即可开始编辑新文件。新文件的文件名默认为“NONAME.C”,此文件可在保存文件时进行改名。另外,也可用“Load”菜单项,在出现“*.C”提示框时,输入当前目录下不存在的文件名,表示文件无内容,是新文件,用户可以通过键盘输入文件内容,进行编辑。

当用“Load”和“New”命令后,Turbo C 自动进入编辑状态。在其他状态情况下,先回到主菜单状态,将高亮度条用←键或→键移到“Edit”处,然后按 Enter 键,系统进入 Turbo C 全屏幕编辑状态。此时编辑窗口顶部的编辑状态行给出正在编辑文件的有关信息。

编辑完成后,可从“File”菜单中选择“Save”命令(将亮条移到 Save 处后,再按 Enter 键)将文件存盘,也可直接按下“F2”键,将文件存盘。在编辑过程中经常保存文件是一个很好的习惯。

1.4.4 编译、连接

在源程序编辑完成后就可以进行编译和连接。编译的目的是生成目标文件(扩展名为 obj)。然后将目标文件进行连接生成可执行文件(扩展名为 exe)。

对一个程序编译、连接的方法是:在文件存盘后,按“F10”键返回到主菜单,然后将亮条移至“Compile”处,按 Enter 键,也可直接按“Alt+C”产生一个子窗口,如图 1-3 所

示。表示编译后将生成 .obj 文件, 连接后生成 .exe 文件。将亮条移到 “Make EXE file” 处(或直接按 F9 键), 则 TC 将对文件进行编译、连接并生成可执行文件。

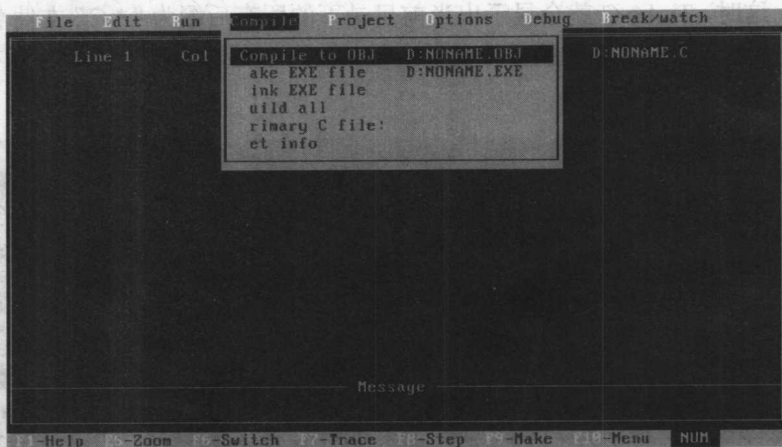


图1-3 Turbo C Compile子菜单

若程序有错, 则在屏幕底部的 Message 窗口显示出错及警告信息。这时可进行修改, 改完后, 再重新进行编译、连接。

1.4.5 运行

编译、连接完一个文件后, 可利用主菜单中的 “Run” 命令或直接按 “Ctrl+F9” 来运行程序。其实, 在源程序编辑完成后, 当用户认为自己的源程序不会有编译、连接错误时, 就可直接用 “Run” 命令或直接按 “Ctrl+F9”, Turbo C 将一次完成从编译、连接到运行的全过程。运行窗口如图 1-4 所示。

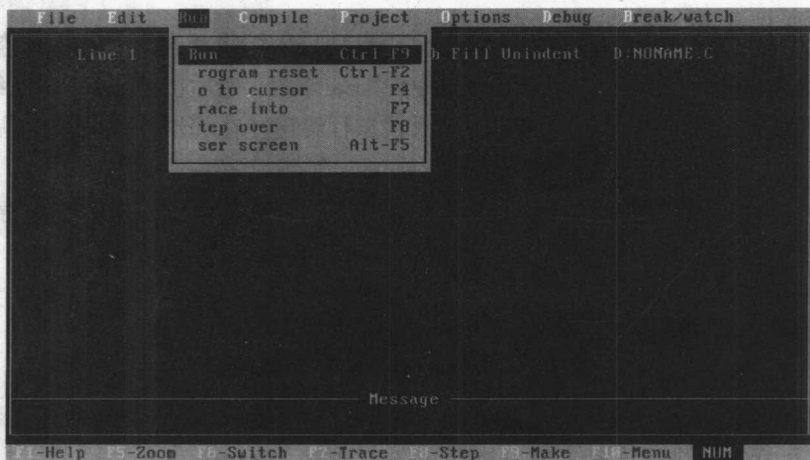


图1-4 Turbo C Run子菜单

程序运行后, 仍回到 TurboC 集成环境。这时若想看运行结果, 可以按 “Alt+ F5” 转到

用户屏，程序运行的结果显示在用户屏上。这时屏幕上的结果是包括本次在内的以前各次运行程序的结果和 DOS 下的操作信息，一般最下边的一组是本次程序的运行结果。看完后可按任意键回到 TurboC 集成环境。

1.5 小 结

本章主要介绍程序、程序设计语言，C 语言的来源及特点，C 语言程序的基本结构。写 C 语言源程序时为了便于阅读，应用阶梯缩进格式书写。本章还简单介绍了 Turbo C 集成环境的应用，装入 Turbo C 源程序的方法，新建源程序的方法，在 Turbo C 集成环境中输入、编译、连接、调试和运行 C 程序的基本方法和步骤。

1.6 习 题

- (1) 简述 C 语言的主要特点。
- (2) 上机运行“Hello C”程序，熟悉 Turbo C 系统的上机方法与步骤。
- (3) 在运行 Turbo C 系统前先运行中文系统，在“Hello C”程序中的一对双引号之间写一句中文，看看程序能否正确编译、运行，输出的结果是什么，是否能正确输出中文信息。
- (4) 在“Hello C”程序中加入一些空格、制表符、换行符，看看在什么情况下会出现问题，什么情况下不会出现问题。