



中等职业教育规划教材
根据教育部中等职业学校新教学指导要求编写

C语言 程序设计

中等职业教育规划教材编写组

► 董世方 主编



中华工商联合出版社
CHINA INDUSTRY & COMMERCE ASSOCIATED PRESS

中等职业学校计算机系列教材

C 语言程序设计

中等职业教育规划教材编写组

董世方 主编

中华工商联合出版社

责任编辑：曹荣 卢俊

封面设计：陈立明

图书在版编目(CIP)数据

C 语言程序设计/董世方主编. —北京：中华工商联合出版社，2006

(中等职业学校计算机系列教材)

ISBN 7-80193-472-5

I. C... II. 董... III. C 语言—程序设计—专业学校—教材 IV. TP312

中国版本图书馆 CIP 数据核字(2006)第 124890 号

中华工商联合出版社 出版、发行

北京东城区东直门外新中街 11 号

邮编：100027 电话：64153909

网址：www.chgslcbs.cn

北京鸿鹄印刷厂印刷

新华书店总经销

787×1092 毫米 1/16 印张：16 260 千字

2006 年 11 月第 1 版 2006 年 11 月第 1 次印刷

ISBN 7-80193-472-5/G·157

定价：20.80 元

中等职业教育系列规划教材

出版说明

为了更好地贯彻《中共中央、国务院关于深化教育改革、全面推进素质教育的决定》精神，全面落实《面向 21 世纪教育振兴行动计划》中提出的职业教育课程改革和教材建设规划，中等职业教育规划教材编写组组织相关力量对实现中等职业教育培养目标、保障重点专业建设的主干课程进行了规划和编写。从 2006 年秋季开始，中等职业教育系列规划教材将陆续出版，提供给广大中等职业学校使用。

中等职业教育系列规划教材是面向中等职业教育的规范性教材，严格按照国家教育部最新颁发的教学大纲编写，并通过了专家的审定。本套教材深入贯彻了素质教育的理念，突出了中等职业教育的特点，注重对学生的创新能力和实践能力的培养。本套教材在内容编排、例题组织和图示说明等方面努力作出创新亮点，在满足不同学制、不同专业以及不同办学条件教学需求的同时，实现教学效果的最优化。

希望各地、各校在使用本套教材的过程中，认真总结经验，及时提出改善意见和建议，使之不断地得到完善和提高。

中等职业教育规划教材编写组

前 言

进入 21 世纪, 计算机技术已经成为推动社会经济飞速发展的重要基础, 也是知识经济时代的代表。中等职业学校在培养合格技术人员的同时, 必须重视计算机技术的教育, 这不仅有利于提高学生的基本文化素质, 也为学生学习其他知识提供了必要的辅助工具。

C 语言由于其功能的强大, 在计算机的各个领域内得到广泛的应用, 从 UNIX、DOS 到 Windows 都使用了 C 语言来进行设计。同时, C 语言结构性强, 在编程领域的涵盖面广, 已成为学习计算机程序语言设计的基础语言, 众多的程序开发课程均以 C 语言为基础。

本书是为中等职业学校编写的教材, 根据教育部 2001 年颁布的《中等职业学校计算机及应用专业 C 语言编程基础教学指导》编写。根据中等职业教育的培养目标, 本课程的教学任务是: 使学生了解和掌握 C 语言的基础知识, 并能用 C 语言进行简单的程序设计, 提高学生的科学文化素质, 培养团结合作的精神, 达到培养高素质劳动者和初中级专门人才的基本要求。

本书适用于中等职业学校计算机及相关专业, 按计算机软件、多媒体应用技术、计算机网络技术及应用等 3 个专业组织编写。在教学内容的编排上, 力求首重提高受教育者的职业能力。

本书在内容的选择上注意贴近应用, 在叙述上尽量做到通俗易懂, 透析基础知识, 注重“任务驱动”, 以学生为主体安排教学内容。全书共分为 10 章, 包括了从程序设计的基础知识、流程控制到指针、结构、文件管理与编译预处理等各方面的知识点。

书中各章都配有相应的习题, 以利于学生对本章内容的巩固。为了便于教学, 同时每章均提供了相应的具有代表性的实例, 通过实例巩固理论知识, 再结合实验教学, 培养学生的创新能力, 将理论与实践紧密联系在一起。

评审专家充分肯定了本书的编写思想、内容选择和编写体例，专家们认为，“教材在内容选择和篇幅上符合教学指导的要求，较好地采用了‘任务驱动’的编写方法。在教学内容的选择和编排上作了一些合理的调整 and 探索，比较适合中等职业学校学生的特点和教学实际”。

由于编者水平有限，书中难免存在一些不足和缺点，恳请广大师生及读者不吝提出批评、指正和改进意见。在此深表谢意。

编 者

目 录

第1章 C语言概述	1	2.3.2 算术运算符与算术表达式	33
1.1 C语言简史	1	2.3.3 逻辑运算符与逻辑表达式	34
1.1.1 C语言的由来	1	2.3.4 关系运算符与关系表达式	35
1.1.2 C语言的发展历史	1	2.3.5 其他运算符	35
1.1.3 C语言简介	2	小结·练习·实验	39
1.1.4 C语言的特点	2	第3章 顺序程序设计	47
1.2 C语言的组成	3	3.1 语句	47
1.2.1 C程序的构成	3	3.2 数据的输入与输出	50
1.2.2 C程序的书写格式	4	3.2.1 字符的输入/输出函数	50
1.3 C程序的编译、链接和运行	4	3.2.2 格式化输入/输出函数	52
1.3.1 C程序的开发过程	4	3.3 顺序结构程序设计	63
1.3.2 Turbo C 开发环境简介	5	小结·练习·实验	65
1.4 算法	9	第4章 选择程序设计	71
1.4.1 算法的概念	9	4.1 概述	71
1.4.2 算法的特性	9	4.2 条件选择语句——if 结构	71
1.5 流程图	10	4.2.1 if 语句的第1种形式—— if 结构	71
1.6 程序设计	10	4.2.2 if 语句的第2种形式—— if-else 结构	72
1.6.1 程序设计	10	4.2.3 if 语句的第3种形式—— if-else if-else 结构	74
1.6.2 结构化程序设计	11	4.3 if 语句的嵌套使用（选学）	76
小结·练习·实验	13	4.3.1 在 if 语句中嵌套具有 else 子句的 if 语句	76
第2章 数据类型、运算符和表达式	15	4.3.2 在 if 语句中嵌套没有 else 子句的 if 语句	78
2.1 常量与变量	15	4.4 多分支选择语句—— switch 结构	79
2.1.1 标识符	15	4.4.1 switch 结构的基本形式	79
2.1.2 关键字	16	4.4.2 带 break 语句的 switch 语句	81
2.1.3 常量	16	4.4.3 switch 语句的嵌套（选学）	83
2.1.4 变量	18	4.4.4 if 语句和 switch 语句 的相互转换	84
2.2 基本数据类型	18		
2.2.1 划分数据类型的必要性	18		
2.2.2 整型数据	19		
2.2.3 实型数据	21		
2.2.4 字符型数据	23		
2.2.5 类型的混合运算	26		
2.3 运算符与表达式	29		
2.3.1 赋值运算符与赋值表达式	29		

小结·练习·实验	85	第7章 数组	142
第5章 循环程序设计	92	7.1 概述	142
5.1 概述	92	7.1.1 数组和数组元素的基本概念	142
5.2 while 和 do-while 循环	92	7.2 一维数组	143
5.2.1 while 循环	92	7.2.1 一维数组的定义	143
5.2.2 do-while 循环	94	7.2.2 一维数组的引用	144
5.2.3 while 循环和 do-while 循环	95	7.2.3 一维数组中元素的地址	146
5.3 for 循环	95	7.2.4 一维数组的初始化	147
5.3.1 for 循环的一般形式	95	7.2.5 一维数组应用举例	149
5.3.2 for 循环的特殊形式(选学)	97	7.3 字符串	154
5.4 循环的嵌套	98	7.3.1 字符串的输入函数 scanf、gets	154
5.5 for 循环与 while 循环的转换	101	7.3.2 字符串的输出函数 printf、puts	155
5.6 break 语句与 continue 语句	102	7.3.3 求字符串长度的函数 strlen	156
5.6.1 break 语句	102	7.3.4 字符串的复制函数 strcpy	157
5.6.2 continue 语句	103	7.3.5 字符串的比较函数 strcmp	158
5.6.3 break 和 continue 的区别	104	7.3.6 字符函数的综合应用	159
小结·练习·实验	106	7.4 字符数组	162
第6章 函数	114	7.4.1 字符数组的定义	162
6.1 概述	114	7.4.2 字符数组的引用	163
6.2 函数的定义	116	7.4.3 字符数组的初始化	163
6.2.1 无参函数的定义	117	7.4.4 字符数组的输入输出	166
6.2.2 有参函数的定义	118	7.5 二维数组(选学)	168
6.3 函数的参数和返回值	119	7.5.1 二维数组概述	168
6.3.1 函数的参数	119	7.5.2 二维数组的定义	168
6.3.2 函数的返回值	122	7.5.3 二维数组中各元素的引用	169
6.4 函数的调用	123	7.5.4 二维数组元素的地址	170
6.4.1 函数调用的一般形式	124	7.5.5 二维数组的初始化	170
6.4.2 函数调用的方式	125	7.6 数组在函数中的应用	171
6.4.3 函数的声明	126	小结·练习·实验	175
6.5 变量的作用范围	128	第8章 指针	182
6.5.1 局部变量	128	8.1 概述	182
6.5.2 全局变量	129	8.2 指针的定义	182
6.6 变量的存储类别(选学)	130	8.2.1 地址的概念	182
6.6.1 自动变量 auto	131	8.2.2 指针的定义	184
6.6.2 静态变量 static	132	8.2.3 指针变量的操作	185
6.6.3 寄存器变量 register	133	8.2.4 指针的运算	190
6.6.4 外部变量 extern	134	8.3 指针在数组中的应用	191
小结·练习	136	8.3.1 指针与一维数组	191

8.3.2 指向数组元素的指针变量.....	194	9.5 位运算符与位运算（选学）.....	223
8.4 指针函数与函数指针.....	195	小结•练习.....	226
8.4.1 指针作为函数参数.....	196	第 10 章 文件（选学）.....	230
8.4.2 函数指针的定义.....	198	10.1 文件概述.....	230
8.5 指针其他用法（选学）.....	199	10.2 文件指针.....	231
小结•实验.....	203	10.3 文件的打开.....	231
第 9 章 编译预处理、结构体和共同体.....	208	10.4 文件的关闭.....	233
9.1 概述.....	208	10.5 文件的读写.....	233
9.2 编译预处理.....	208	10.6 其他一些函数.....	235
9.2.1 宏定义.....	209	小结•练习.....	238
9.2.2 文件包含与条件编译.....	212	附录.....	240
9.3 结构体.....	215	附录一 C 语言的关键字.....	240
9.3.1 结构体的含义.....	215	附录二 C 语言运算符的 优先级和结合性.....	240
9.3.2 定义结构体变量.....	216	附录三 C 常用函数库.....	241
9.3.3 结构体变量初始化.....	218	附录四 常用字符与 ASCII 码对照.....	244
9.3.4 结构体变量成员的访问.....	219		
9.4 共同体（选学）.....	221		

第 1 章 C 语言概述

随着高科技日新月异的发展，我们在日常工作、学习中与计算机接触的机会越来越多，大家都知道，人与人之间通过语言来交流，那么人与计算机怎么打交道呢？这就需要一门特殊的语言——计算机语言。计算机语言是一种人与计算机之间通信的语言，包括编程语言、命令语言、查询语言、定义语言等。本书所讲的 C 语言就是计算机语言中的一种编程语言。

1.1 C 语言简史

C 语言的发展经历了很多过程，下面从它的鼻祖讲起。

1.1.1 C 语言的由来

C 语言诞生于 1972 年，由美国电话电报公司 (AT&T) 贝尔实验室的 D. M. Ritchie 设计，并首先在一台使用 UNIX 操作系统的 DEC PDP-11 计算机上实现。

C 语言的前身是 BCPL 语言。1967 年英国剑桥大学的 Martin Richard 推出 BCPL 语言 (Basic Combined Programming Language)。1970 年贝尔实验室的 K. Thompson 以 BCPL 语言为基础，开发了 B 语言，并用 B 语言编写了 UNIX 操作系统，在 PDP-7 计算机上实现。1972 年贝尔实验室的 D. M. Ritchie 在 B 语言的基础上设计出 C 语言。C 语言既保持了 BCPL 语言和 B 语言的精炼、接近硬件的优点，又克服了它们过于简单的缺点。

最初，设计 C 语言的目的只是为了描述和实现 UNIX 操作系统。目前，C 语言已独立于 UNIX 系统，被移植到大、中、小型计算机上。1978 年 B. Kernighan 和 D. M. Ritchie 合编了经典著作 “The C Programming Language”，这是目前所有 C 语言版本的基础。1983 年美国国家标准化协会 (ANSI) 对 C 语言问世以来的各种版本进行了扩充，制定了 ANSI C。现在流行的 C 语言的版本有：Microsoft C，Turbo C，Quick C，Borland C 等。

1.1.2 C 语言的发展历史

C 语言的演变过程如下所示：

ALGOL60 (1960 年) → CPL (1963 年) → BCPL (1976 年) → B (1970 年) → C (1972 年)

(1) 1972 年贝尔实验室的 D.M.Ritchie 在 B 语言的基础上设计出 C 语言，此时 C 语言只是为了描述和实现 UNIX 操作系统的一种工作语言，一段时间还只是在贝尔实验室内部使用。

(2) 1975 年，UNIX 第 6 版公布后，C 语言的突出优点引起人们的普遍注意。

(3) 1977 年，出现了可移植的 C 语言。

(4) 1978 年, UNIX 第 7 版的 C 语言成为后来被广泛使用的 C 语言版本基础, 即标准 C。

(5) 1983 年, 美国国家标准化协会 (ANSI) 根据 C 语言问世以来的各种版本进行了扩充, 制定了 ANSI C。

(6) 1990 年, 国际标准化组织 ISO 制定了 ISO C 标准, 目前流行的 C 语言编译系统都是以它为标准的。

1.1.3 C 语言简介

当前, 用于程序设计的高级语言数以百计, 其中, C 语言以其强大的功能被称为是真正的、程序员的编程语言。

C 语言虽然是高级语言, 但它也可以完成许多只有低级语言才能完成的、面向机器的底层工作, 因此也被称为“中间语言”。正是由于 C 语言的这些特性, 决定了它成为一种重要的程序设计语言, 在人们所使用的程序中, 大多是由 C 语言编写的, 例如, 使用最多的计算机操作系统 Windows 98/2000 中, 有相当多的部分是由 C 语言编写的。

1.1.4 C 语言的特点

C 语言是结构化程序设计语言之一, 和其他高级程序设计语言相比, C 语言具有以下几个显著的特点。

1. 语言简练

C 语言中常用的关键字有 32 个, 其中和数据类型有关的有 15 个, 和存储类型有关的有 4 个, 用于流程控制的有 12 个, 以及一个运算符 `sizeof`。它压缩了其他高级语言中冗余的部分。程序主要由小写字母组成。

2. 运算符丰富

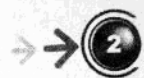
C 语言中提供了 34 种运算符, 可以构成多种表达式, 完成复杂的运算。

3. 数据类型丰富

C 语言中除了有整型、实型、字符型等基本数据类型以外, 还提供了数组、指针、结构体等构造数据类型, 而且能够用来实现更复杂的数据结构, 如线性表、链表、栈、队列、树和图等。

4. 结构化语言

C 语言中有 `if-else`、`while`、`switch` 等结构化控制语句, 并用函数作为程序的模块, 是理想的结构化程序设计语言。



5. 程序设计灵活

C 语言程序书写格式自由，编程的自由度大，语法限制不太严格，如对数组下标越界不作检查，各种类型的变量可以通用。

6. 直接访问物理地址和位运算

允许直接访问物理内存，这使得 C 语言具有汇编语言的一些功能，是其他高级语言不具备的。它还能对数据进行按位操作。

1.2 C 语言的组成

1.2.1 C 程序的构成

例如，一个简单的 C 程序如下：

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

运行本程序将输出以下一行内容：

```
Hello World
```

(1) C 程序由一系列函数组成，这些函数中必须有一个且只能有一个以 `main` 为名的函数，这个函数称为主函数，整个程序从主函数开始执行。在此例中，只有一个主函数而无其他函数。

(2) 程序的第一行中的 `main` 是主函数的函数名，`main` 后面的一对小括号是函数的标志，不能省略。

(3) 第三行的 `printf` 是 C 语言的格式输出函数，它在屏幕上显示出括号内双引号之间的字符串，其中“`\n`”代表换行符。语句末尾用分号“`;`”，它是一个语句结束的标志。

(4) `{}` 括起来的部分称为函数体，一个函数中所有的执行语句都写在函数体中。左大括号“`{`”表示函数体的开始，右大括号“`}`”表示函数体的结束。

通过对以上一个很简单的 C 程序的分析，可以总结出 C 语言源程序的构成，有以下几个要点：

- ① 一个 C 语言源程序由一个或多个函数组成。
- ② 每个源程序必须有且只能有一个 `main()` 函数。
- ③ 除了 `main()` 函数外，程序中还可以有若干个其他的函数。

C 语言程序的执行过程如下：

- ① 程序从 `main()` 函数开始执行。
- ② 其他函数通过调用的方式被执行。
- ③ 程序最后在 `main()` 函数中结束。

1.2.2 C 程序的书写格式

C 语言程序的书写格式比较自由，主要有以下几点要注意：

- (1) 标识符的大小写是有区别的。例如，`a` 和 `A` 表示两个不同的变量。
- (2) C 程序语句用分号 “`;`” 结束，分号是 C 语句的必要组成部分。
- (3) 一行可以写多个语句，一个语句可以分写在几行。
- (4) 可以在程序的任何位置用 “`/*`”、“`*/`” 对程序或语句进行注释。注释主要用于说明变量的含义和程序段的功能等，以提高程序的可读性，对程序的运行不起作用。

1.3 C 程序的编译、连接和运行

1.3.1 C 程序的开发过程

C 程序开发的基本过程如图 1-1 所示。

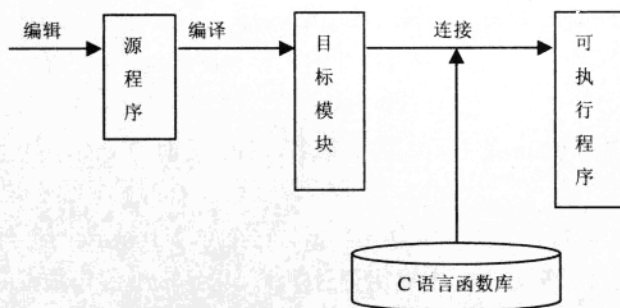


图 1-1 C 程序的开发过程

1. 编辑

选择适当的编辑程序，将 C 语言源程序通过键盘输入到计算机，并以文件的形式存储到磁盘中。源程序文件以 `.C` 作为扩展名。

2. 编译

选择适当的编译程序，将 C 语言源程序转换为目标程序，这一过程称为编译。在编译过程中，可能会发现源程序中的一些语法错误，这时需要重新利用编辑程序来修改源程序，然

后再重新编译。源程序文件进行编译之后形成目标文件. OBJ。

3. 连接

经过编译后生成的目标程序是不能直接执行的,需要经过连接之后才能生成可执行程序。连接是将编译形成的目标文件. OBJ 和库函数、其他目录文件连接形成统一的可执行的二进制文件. EXE。

4. 执行

经过编译、连接之后,源程序文件就生成了可执行文件,运行可执行的二进制文件. EXE,可以得到程序的运行结果。

1.3.2 Turbo C 开发环境简介

Turbo C 是微机上广泛使用的编译系统,它是一个集成开发环境,在这个环境下,可以完成一个 C 程序的编辑、编译、连接、运行的全部过程。

1. Turbo C 的启动

DOS 命令启动方式如下:

C:\>CD\TC<回车> 或 C:\> CD TC <回车>即进入 Turbo C 目录;

C:\TC> TC<回车>即执行 TC.EXE 文件;

启动成功后,即可进入 Turbo C 集成开发环境,如图 1-2 所示。

2. Turbo C 的窗口组成

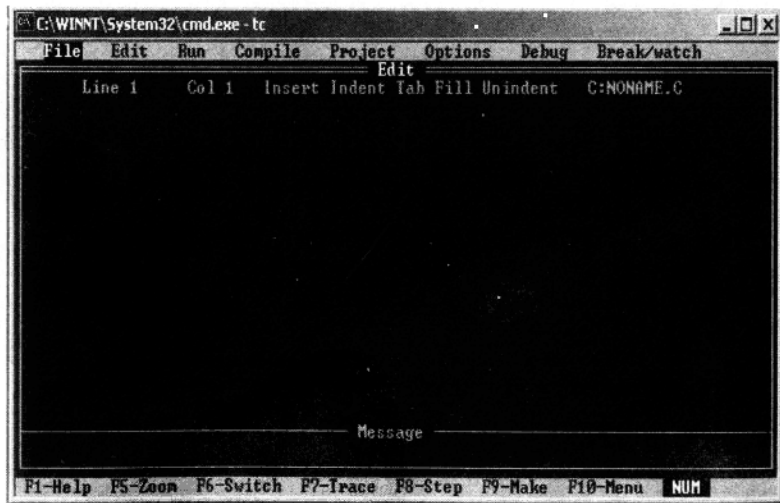


图 1-2 Turbo C 的启动窗口

(1) 主菜单。屏幕的第一行是主菜单，包括了 8 个菜单，各菜单的名称及主要功能如下：

- File 文件操作，载入或保存文件，调入 DOS 和退出 Turbo C 系统。
- Edit 编辑程序。
- Run 运行当前程序。
- Compile 编译、连接当前程序。
- Project 项目管理，管理多文件项目。
- Options 设置系统环境选项。
- Debug 设置调试选项。
- Break/watch 协调跟踪断点/查看内部值。

激活主菜单可以用下面的两种方法之一：

- ① 按 F10 键后，用光标键“→”和“←”选择菜单上的各菜单选项，被选中的项以反相显示，按回车键后，显示该选项的下拉菜单。
 - ② 用 Alt 键配合菜单项名称的大写字母，如按 Alt+F 组合键，File 菜单项将被反相显示。
- 在上面的菜单选项中，除 Edit 外，其他各项均有下拉菜单，这时可用“↑”和“↓”在子菜单中选择具体的命令。

(2) 编辑窗口。屏幕上的第二部分为编辑窗口，用来输入和修改源程序。

(3) 消息窗口。屏幕的第三部分为消息窗口，用来显示程序编译、连接和运行时的情况，如出错信息等。

(4) 常用功能键。屏幕上的最下面一行是常用的功能键：

- F1-Help 帮助
- F5-Zoom 放大或恢复激活的窗口
- F6-Switch 交替激活窗口
- F7-Trace 跟踪
- F8-Step 单步执行程序
- F9-Make 编译并连接
- F10-Menu 激活主菜单

3. 选择用户文件的存放目录

在 Turbo C 主菜单窗口中，用 F10 键和光标从主菜单中选择“File”选项，按回车键，弹出文件下拉菜单，如图 1-3 所示。

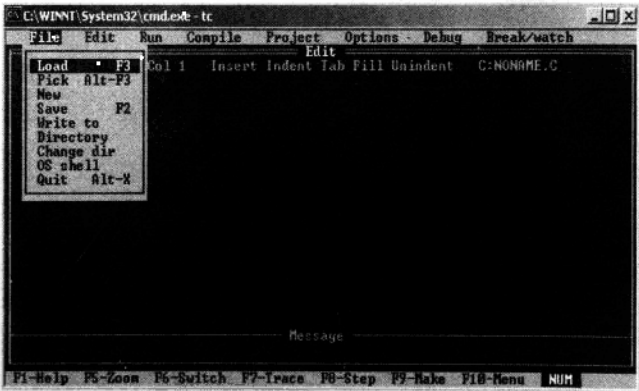


图 1-3 “File” 菜单

在弹出的下拉菜单中选“Change dir”选项并按回车键，则弹出一个子目录提示框，如图 1-4 所示。用户可以在该提示框中输入用户文件的存放目录，输入完后按回车键即可。

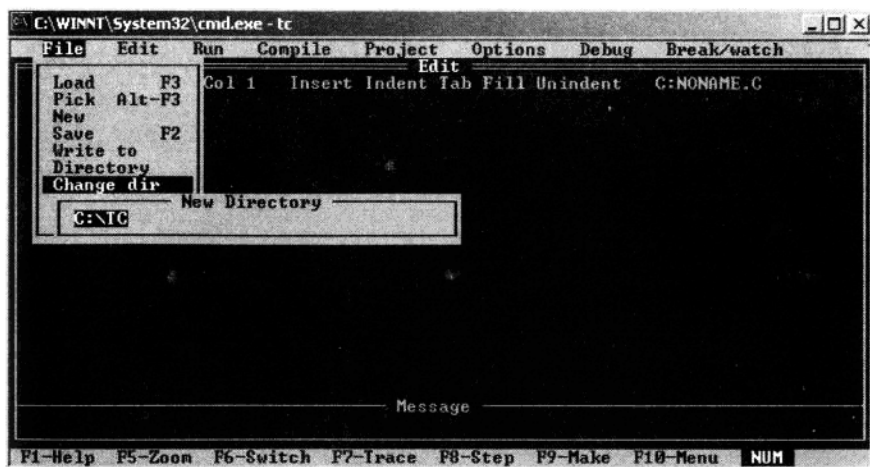


图 1-4 “File/Change dir”菜单

4. 建立工作环境

这里所说的工作环境，是指对 Turbo C 工作状态的设置以及编译程序和连接程序要求的环境变量的设置。具体地说，就是设置编译、连接所需要的文件的目录，以及生成的可执行文件的存放目录。

操作步骤如下：

- (1) 按 F10 键，在主菜单上选择“Options”选项并按回车键，显示如图 1-5 所示。

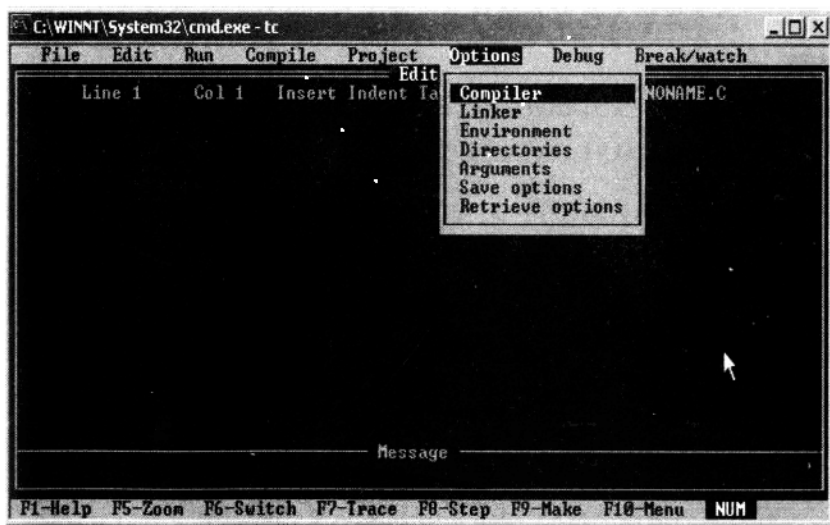


图 1-5 “Options”菜单

- (2) 在“Options”子菜单中选择“Directories”选项并按回车键，将弹出子菜单，如图 1-6 所示。

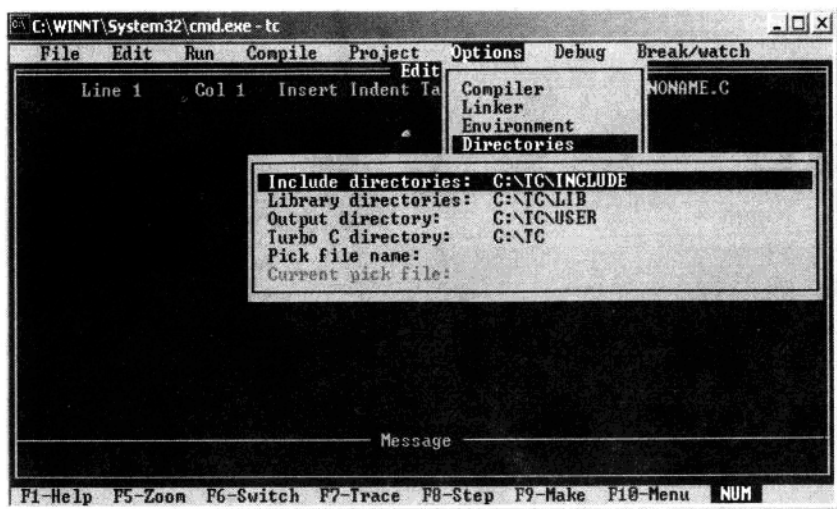


图 1-6 Options/Directories 菜单

子菜单中各选项的含义如下：

“Include directories”选项：说明标准包含文件的目录，即 INCLUDE 所在的位置。输入时应包含文件所在的盘符和路径，这里输入：C:\TC\INCLUDE。

“Library directory”选项：说明 Turbo C 运行时库文件所在的目录，即 LIB 所在的位置。输入时应包含库文件所在盘符和路径，这里输入 C:\TC\LIB。

“Output directory”选项：编译、连接后的可执行文件所存放的目录，即用户的.obj、.exe 文件的存放位置。用户选择输入已存在的目录，若为空，则这些文件被放在当前目录。这里输入 C:\TC\USER。

“Turbo C directory”选项：Turbo C 文件所在的目录，用于 Turbo C 系统寻找配置文件(.TC)和帮助文件(TCHELPT.CH)。这里输入 C:\TC。

5. 编辑源程序

(1) 建立新文件。执行主菜单的“Edit”光标停在编辑区，此时可以输入源程序，并可以利用光标移动键、删除键和插入键等进行程序的编辑。

(2) 保存源程序。执行“File”菜单的“Save”命令，这时，在打开的对话框中输入程序名后，按回车键，可将程序保存在文件中，文件的扩展名为“.c”。

当一个程序运行成功后，要编辑下一个程序时，应先执行“File”菜单的“New”命令，将编辑窗口原有内容清除，才可以输入新的程序。

(3) 修改已保存的旧程序。执行“File”菜单的“Load”命令，在打开的对话框输入已保存的程序名，该程序的内容显示到编辑窗口中，这时可以对其进行修改。

(4) 编译和连接。执行“Compile”菜单的“Compile to OBJ”命令，可对源程序进行编译，形成目标程序.obj。

