



电脑报 东方工作室

时尚编程

与

CC++
实习指导

- 主 编 周属衡
- 副主编 陈光海
- 编 著 周属衡 王晓蓉
纪良浩 李 崇

重庆出版社

TP312C
173

时尚编程

与

++
实习指导

- 主 编 周属衡
- 副主编 陈光海
- 编 著 周属衡 王晓蓉
纪良浩 李 崇



北京服装学院图书馆



00210917

▲ 重庆出版社

目 录

第 1 章 C 语言与 TURBO C	1
1.1 C 语言的主要特点	1
1.2 TURBO C 常识	1
1.2.1 Turbo C 2.0 的主要特点	1
1.2.2 Turbo C 2.0 安装、运行的基本条件	2
1.2.3 安装	2
1.2.4 启动 TC	3
第 2 章 编辑、运行 C 程序	4
2.1 TC 系统主屏	4
2.1.1 信息行的含义和作用	5
2.1.2 TC 集成系统功能键与快捷键	6
2.2 在 TC 环境建立、运行、调试 C 程序	7
2.2.1 典型 C 程序源文件编辑技巧	7
2.2.2 C 程序的运行	10
2.2.3 C 程序的调试与纠错	11
2.2.4 C 程序建立、保存及常用块操作	16
第 3 章 例程分析及习题	18
3.1 编程思想及基本输入输出实习	18
3.1.1 目的要求	18
3.1.2 例程分析	18
3.1.3 实习内容	24
3.2 选择结构程序设计实习	25
3.2.1 目的要求	25
3.2.2 例程分析	25
3.2.3 实习内容	34
3.3 循环结构程序设计实习	35
3.3.1 目的要求	35
3.3.2 例程分析	35
3.3.3 实习内容	44

第1章 C语言与 Turbo C

C语言问世于70年代初,它也是计算机技术人员与计算机交流的一种语言,当时主要在小型计算机上应用。与计算机硬件相比,计算机软件的发展要漫长一些,直至二十世纪八十年代后期即1987、1988年PC机广为流行时,美国Borland公司推出了Turbo C 2.0的C语言集成开发系统,至二十世纪九十年代中、后期,也就是DOS鼎盛时期,C语言最受程序员青睐,并逐步被我国各高校定为必修的程序设计课。学好C语言程序设计,上机实验是必不可少的重要环节。为此,我们有必要对C语言和Turbo C的特点有足够的认识、了解。

1.1 C语言的主要特点

C语言与其他编程语言相比,有以下主要特点:

(1) 简洁且结构清晰

由于C语言通常由许多函数组成,便于编程的模块化和结构化。显然,这样设计编制的程序简洁清晰,易读性强。

(2) 表达能力强

C语言不仅提供了对计算机运行细节的操作方法,还提供了强大的功能库,使程序员可快速地编制出希望的程序,精确地控制计算机按照自己的意愿工作。可以说,很容易用C语言表达出希望计算机要作的事。

(3) 高效率的编译性语言

该语言生成的目标代码质量高,运行速度快。对于较大的程序,C语言的源代码可分开存放在多个文件中,编译后再链接在一起,形成计算机可执行文件。

(4) 通用性好

几乎各类通用计算机的各种操作系统都配有C语言编译系统,以至使你编写的程序基本上不需作改动即可在计算机绝大部分系统中使用。如你在IBM的PC上编制的C语言程序,为了在苹果机上运行,只需将原来的C程序用苹果机上的C语言编译器编译一次即可,无需再学习苹果机的C语言编译器。

1.2 Turbo C 常识

Turbo C 2.0(以下简称TC)是在DOS操作系统下的C语言编译系统的最终版本。以后的发展是C++和VC++。

1.2.1 Turbo C 2.0 的主要特点

(1) 集成强有力的综合开发环境,不仅无需在TC集成开发系统与操作系统之间切换,就能很方便地进行C语言程序的编辑、编译、链接、运行,而且还提供单步执行、单步跟踪、断点设置、表达式监视和求值等强有力的程序调试工具和Turbo Debugger

独立调试程序；

(2) 没有 80X87 协处理器芯片，可对 80X87 进行浮点仿真，也能实现复杂、快速的浮点运算；EMS（扩展内存规范）和相应函数用作编辑缓冲区分配内存更快；

(3) 能随时方便地获得 C 语言功能库的详细解释；

(4) 生成的目标代码优于一般的 C 编译、链接器 20~30%；

(5) 高级图形库中新增了许多函数（如：Signal 和 Raise 函数、-emit-函数）等，还有可安装的驱动程序和字体；

(6) 支持命令行上的通配符“*”和“？”；支持 Long double 常数和变量；支持并自动检测 CGA、EGA、VGA 等多种类型显卡；可自动进行块缩进、回退及优化填充；可链接生成运行速度快的小模式.COM 文件等；

(7) 可直接在 C 语言的程序中嵌入汇编语言；MAKE 实用程序实现自动依赖关系检查；还能直接调用系统功能等。

1.2.2 Turbo C 2.0 安装、运行的基本条件

Turbo C 2.0 安装、运行需要以下基本条件：

- (1) 最好是 PC/AT（即 80286）以上的（包括 PS/2）系列及兼容微机；
- (2) 至少需要 448K 的 RAM；硬盘空间不少于 3M；一个 3.5 英寸的软驱；
- (3) 任何 80 列、25 行单色或彩色显示器；
- (4) 起码需要 DOS 2.0 版本支持，最好是 DOS 3.2 及以上版本。

Turbo C 2.0 能对 80X87 协处理器芯片进行浮点仿真，所以并不一定需要 80X87 芯片，但如果系统中有该芯片，将能大大加快浮点运算的速度，从而加快程序的执行。

1.2.3 安装

初推出 Turbo C 2.0 时，软盘只有 5 1/4 英寸的 360K 双面低密磁盘，Turbo C 2.0 的全部文件只好分做在 6 张未加密的软盘上。在 5 1/4 英寸的 1.2M 双面高密磁盘上市后，仅用 2 张盘即将全部安装文件存放。

拿到 Turbo C 2.0 源盘后，通常先作一备份，用备份盘去工作，将源盘安全存放，以免丢失或损坏源盘数据。

通常是用源盘中提供的“INSTALL.EXE”安装程序来完成 TC 集成系统的完整安装。若只需用基本功能，不需完整安装，也可按不同的需要把源盘上的文件拷贝到工作软盘或硬盘中即可。下面仅对完整安装作介绍。

将 Turbo C 2.0 第一张含有 INSTALL.EXE 文件的源盘插入到 A 驱动器中，把当前驱动器改为 A，然后在 DOS 的提示符下（如：“A:\>”）键入 INSTALL，然后按回车键。

也可将所有的文件拷贝到硬盘的一个临时目录下，键入 INSTALL，按回车安装。

INSTALL 运行后，首先显示欢迎界面的第一屏幕，按回车，切换为提示让输入源文件所在驱动器和路径的第二屏幕。正确输入后，出现如图 1-1 Turbo C 2.0 必要安装信息的第三屏幕。第三屏幕中间框中列有三项选择，其意分别为：

- ❖ 把 Turbo C 2.0 各部分安装到一硬盘（相应的各子目录下，并根据各子目录创建

Turboc.CFG 文件)

- ❖ 把当前的 Turbo C 1.5 版升级为 2.0 版 (并传送当前 TC.EXE 中的配置选择项)
- ❖ 在软盘系统中建立 Turbo C 2.0 (必须先准备好三张格式化了的一套空盘)

每次运行 INSTALL 时,都让你安装一种存贮模式的 Turbo C,如果想安装几种模式,就得有几套盘,每一套对应一种模式用户可以根据自己的实际需要选择其中的一项。

以后按屏幕提示进行,便能成功完成安装。

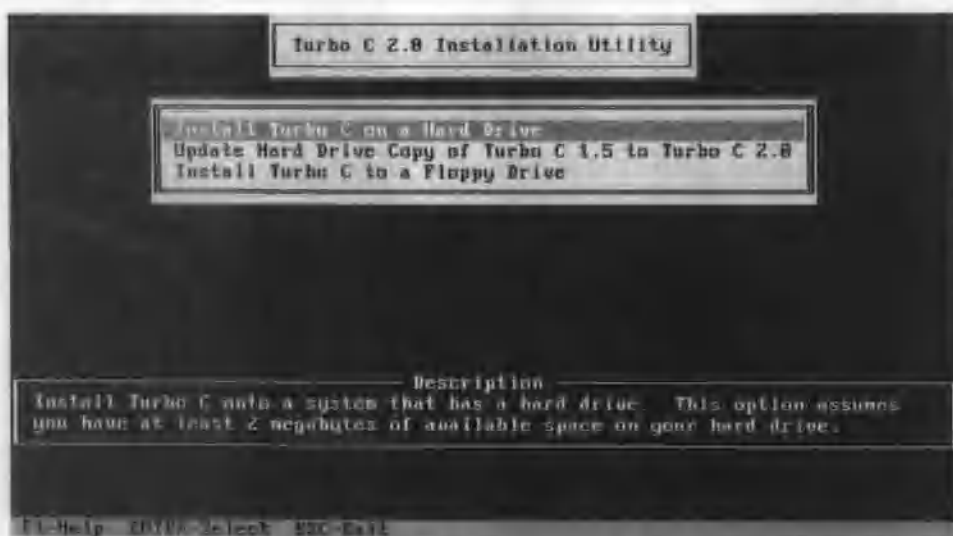


图 1-1 Turbo C 2.0 必要安装信息

1.2.4 启动 TC

安装完 Turbo C 2.0 以后,进入 TC 子目录,在 DOS 提示符下键入 TC、回车,即可加载 Turbo C 集成开发系统,进行 C 程序设计操作。

通常启动 Turbo C 集成开发系统的操作如下:

C:\TC>TC <回车> (书上添加阴影的“C:\TC>”表示系统提示,不从键盘上键入,以下全同)

该操作只键入 T、C 两个字符(不分大小写、中间没有空格),然后按回车键即可。

为了避免计算机病毒侵害等原因,高校计算机 C:盘多数安装有写保护,此时最好以下面步骤启动 Turbo C 集成开发系统:

C:\>cd: <回车> (工作在 C 盘时作此操作,否则省去此步)

D:\>cd:c:\tc\tc <回车> (编辑的 C 程序自动存在 D 盘根目录,不会锁机)

不然,编写的程序很容易被丢失或是保存为乱码。也可将“d:”转换为“a:”或其他没有写保护的盘符。

第2章 编辑、运行 C 程序

Turbo2.0 (简称 TC) 不仅仅只是一个快速高效的编译程序,而是集编辑程序、调试程序和其它许多易学易用的实用程序为一体的综合软件系统,常称“TC 集成环境”可以用它来建立(编辑)、运行、调试各式各样的 C 程序。

2.1 TC 系统主屏

加载启动 TC 后,显示 TC 主屏和 TC 版本信息框,击任一键,版本信息即消失(任何时候按 Shift-F10,即可再显示初始 TC 主屏和 TC 版本信息框)。参见图 2-1。



图 2-1 主屏及版本信息

任何时候, TC 主屏中的编辑窗口与消息窗口(或监视窗口),只有一个窗口处于激活状态。而处于激活状态的窗口,其标题是高亮度、而标题名两侧为双线条,参见图 2-2。



图 2-2 活动窗口与非活动窗

结合图 2-1 可见, TC 主屏由: 主菜单、信息行、编辑窗口、消息窗口 (或监视窗口)、功能键提示行四部分组成。

为了尽快让读者学会在 TC 集成环境下编辑、运行、调试 C 语言程序的基本方法和技巧, 养成良好的编程风格和习惯, 在此先介绍信息行的含义和作用, 再介绍系统功能键和主菜单中最常用的快捷键应用。若需要进一步系统、深入学习掌握 Turbo C2.0 系统集成环境的知识和应用, 请参见“附录 I TC 2.0 主屏详述”。

2.1.1 信息行的含义和作用

主屏编辑窗口中的第一行, 显示一些当前编辑程序的信息, 称为信息行。参见图 2-3。

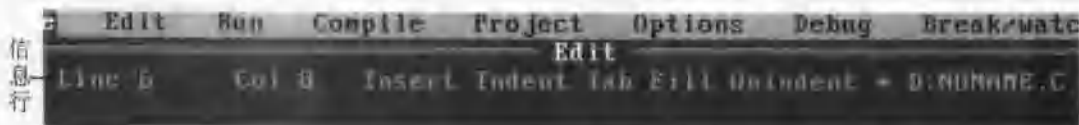


图 2-3 编辑窗口状态栏信息

信息行各项的含义如下:

(1) Line 6 和 Col 8

标示当前光标所在的行和列 (6 行, 8 列)。其数值 6 和 8, 将随光标位置的变化而变化。

(2) Insert

当信息行中有“Insert”显示时, 源程序编辑操作处于插入状态, 即插入为 On。此时:

- 键入的任何字符将插入光标的左侧;
- 若光标在该行代码的末尾回车, 将在该行的下面插入 1 空行;
- 若光标在该行代码的第 1 个字符回车, 将在该行的上面插入 1 空行;
- 若光标在该行的任意代码字符处回车, 则将从该行光标对应字符起拆分为的 2 行插入下面。没作其他操作时, 按退格键可以还原拆分。

• 按控制键区的 Insert 键或小键盘区 (当 Num Lock 灯关闭时) 的 Ins 键, 或按 Ctrl-V (不分大小写), 信息行将隐去“Insert”的显示。

当信息行中有隐去“Insert”时, 源程序编辑操作处于改写状态, 即插入为 Off。此时:

- 键入的任何字符将替换光标正对的代码;
- 无论光标在编辑窗口的任何位置回车, 光标都转到下 1 行的首代码或下 1 行的首列。

不能插入空行:

• 按控制键区的 Insert 键或小键盘区 (当 Num Lock 灯关闭时) 的 Ins 键, 或按 Ctrl-V (不分大小写), 信息行将再次显示“Insert”。

(3) Indent Tab Fill

Indent 自动缩格, 按 Ctrl+O+I 切换 (表示先按下 Ctrl 键不放, 然后再先击 O 字符键, 再击 I 字符键即可。下描述方式雷同)。显示 Indent 时, 自动缩格为 On, 换行编辑时, 光标将定位在上 1 行首字符的同 1 列, 方便于结构化程序设计。

Tab 控制键, 按 Ctrl+O+T 切换。信息行显示有 Tab 时, 每按 1 次 Tab 键, 光标自动缩进 8 列 (系统默认值, 可通过 Options/Environment/Tab Size 重新设置) 递增; 当关闭信息行 Tab 显示时, 每按 1 次 Tab 键, 光标在 1、5、12 列上循环。

Fill 只有在 Indent 和 Tab 同时为 On 时才有此项供用户设置,按 Ctrl-O-F 切换。编辑时优化每行开头的空格。

(4) Unindent

自动对齐,按 Ctrl+O+U 切换。当其为 On,在换行编辑中不小心退格缩进过多时,按退格键(回车键上方),光标即刻与上 1 行首字符所在列对齐;再按退格键,将与上 1 模块首字符所在列对齐。即当光标在文本行的首字符或在空行上时,每按 1 次退格键,文本行的文本或空行的光标将回退 1 级。

(5) D:NONAME.C

此位置显示正在编辑的文件名及存放它的驱动器盘符。新文件未命名时为 NONAME.C,若已命名或编辑已经命名的文件,信息行则显示相应的文件名,不再显示 NONAME。

(6) *

当文件被修改而又未存盘时,在文件名前面会出现星号,提示编者注意修改后的存盘。

2.1.2 TC 集成系统功能键与快捷键

熟练掌握 Turbo C 集成开发系统方便而快捷的功能键和快捷键应用,将大大提高 C 语言的编程效率,充分展现自己综合编程能力。特将 TC 主要功能键和快捷键的相应功能介绍如下:

(1) TC 集成系统主要功能键

在 TC 的任何环境,主要功能键的相应功能见表 2.1.1。

表 2.1.1 主要功能键之功能对应表

功能键	相应功能
F1	获得有关当前位置(运行、编译等)的帮助信息
F5	放大/缩小活动窗口
F6	在 Edit / Message 窗口之间切换为活动或非活动状态
ALT+F6	改变窗口里的内容(消息窗口和监视窗口来回选择,或者当前文件和前面的文件间切换)
F10	在活动窗口与任一主菜单之间切换(若在主菜单的子菜单级也直接返回活动窗口,再按 F10 则只返回该子菜单的主菜单而不是相应的子菜单)
Alt+[主菜单首字符(包括 F、E、R、C、P、O、D 和 B)]	进入指定的主菜单。例如,在系统里的任何位置按 Alt+E 就进入编辑窗口 Edit 菜单项;Alt+F 则表示进入 File 文件菜单项
左、右光标键	从一下拉菜单移到另一个下拉菜单
上、下光标键	弹出下拉菜单或移动任意级菜单中的光条
ESC	退出菜单直接返回活动窗口或退出任意一级子菜单返回上一级菜单
Enter	确认,执行光标或光带选项

注意:书中的 Alt+F6 表示先按下 Alt 键不放,再轻击一下 F6 功能键;Alt+[主菜单首字符],如调用 Edit 主菜单为 Alt+E,即先按下 Alt 键不放,然后轻击一下主菜单首字符 E 字母键。以后类似的描述(包括 Ctrl+[功能键或字母键]等)操作方法全同,不再赘述。

(2) TC 快捷键 (热键)

热键是 TC 系统为执行菜单某一固定功能而设置的键, 它为 TC 的应用提供了方便而快捷的操作。

例如: Alt+X, 它提供退出 TC 集成系统返回 DOS 的快捷方式。

Alt+F9, 弹出“Compiling”信息框, 编译源代码且只生成.OBJ 文件, 不运行程序。

F9, 弹出“Linking”信息框, 编译、连接生成.OBJ 和.EXE 文件, 但不运行程序。

Ctrl+F9, 编译、连接生成.OBJ 和.EXE 文件, 并且运行程序。

Alt+F5, 查看程序运行结果 (等同于调用 Run 主菜单中的 User Screen 命令)。其功能是从 TC 主屏转换到显示用户程序输出结果的屏幕。启动 TC, 通常进入 TC 主屏, 即显示有 Edit 和 Message (或 Watch) 两个窗口的屏幕。当按下“Alt+F5”热键便转到用户程序结果屏幕, 击任意键 (Caps Lock、Shift、Ctrl、Alt、及窗口键除外) 返回 TC 主屏。

可见, 热键能简化应用 TC 集成系统的操作, 提高编辑、调试程序效率, 在编程时应充分运用 Turbo C 2.0 热键及功能键。

注意: 无论在 TC 集成开发系统的任何环境, 只要一按功能或热键, 相应的功能即被调用。但有一例外: 当系统出现红色警示框时, 必须按下警示框中指定的键, 待警示框退出后, 热键方才起作用。

2.2 在 TC 环境建立、运行、调试 C 程序

任何计算机程序语言都要求用户按相应系统规定的格式和语句编写成文本类型的程序文件, 然后由相应系统编译、连接、运行。文本类型的程序文件通常称作源文件 (或源代码)。由于多种原因, 程序员编写的源文件常常会出现代码、逻辑、语法等各种类型的错误, 这需要通过调试、运行来排除。

2.2.1 典型 C 程序源文件编辑技巧

为了使读者尽快掌握程序的编辑、运行、调试的基本方法, 有利于增强读者深入、系统学习的兴趣, 下面特以“手把手”的描述方式叙述, 相信读者很快会感受到学有所成的愉悦心情。先看例程 2.1。

【例 2.1】字符串“Hello”的不同输出方式。

```
/* MCP1.C
   This is my first C program */
#include <stdio.h>
main ( )
{ char a,b,c,d,e;
  a='H';b='e';c='l';d='o';e='!';
  putchar ( a );putchar ( b );putchar ( c );
  putchar ( c );putchar ( d );putchar ( e );putchar ( '\n' );
  printf ( "The result of this program: %c%c%c%c%c%c%c\n",a,b,c,c,d,e );
}
```

(1) 例程语句简释

例程的 1、2 行是注释行，用以对程序的说明。规范的程序设计，都有大量的注释行。所有注释行，在编译时均不产生代码，不影响程序的运行。注释由起始符“/*”和终止符“*/”括起来，既可在语句前，也可在语句后，还可跨行，十分灵活。这也是 C 语言的特点。

第 3 行为预处理命令行。C 程序中若调用了其某个库函数，就必须用“# include”预处理命令，通知编译器将包含该函数说明的包含文件，嵌入源程序文件中。由于这个例程中，调用了包含在“<stdio.h>”头文件中说明的输出函数“Puchar”，所以本行命令进行了嵌入说明。其头文件两侧的尖括号，可用双引号替换。

第 5 行到第 9 行是命名为 main 的函数体。main 函数称为主函数。每个程序都必须有一个 main 函数，不管其在何位置，程序都要从它开始执行。main 函数后面的括号，是函数名必不可少的一部分（如本例程的第 4 行）。这样编译器才知道 main 是一个函数，而不是一个单词，不然，编译器就不能生成相应的代码。

在 C 程序中，函数是能够完成一定功能的语句集合，是程序编译、运行的基本单位。

例程中 5 行的左花括号和第 10 行的右花括号，将函数的语句部分（即函数体）括起来，这是 C 函数要求的形式。C 程序还允许从一个函数中调用别的函数。因此，通过函数的调用可以构成更大的程序。

第 5 行定义 5 个字符变量，第 6 行给变量赋值，第 7、8 两行向终端输出字符及控制，第 9 行为格式输出语句，让计算机将双引号里的文本及变量内容按格式输出，其中反斜杠和字符 n 是换行控制符，其作用是让计算机在输出文字内容后，换一行显示。各语句最后的分号表示各语句的结束。下面需学习建立、保存程序的方法，编辑中的一些技巧。

(2) 程序编辑技巧与存储

启动 TC，选择主菜单 File 中 New 菜单项（或直接按“Esc”键），光标出现在编辑区中，表明已经可以在编辑器中进行程序编写。

◆ 照例程的格式录入到第 7 行的“putchar (a) ;”。观察信息行中有无*号。

◆ 按 F2 键，第 1 次保存程序，都将弹出如图 2-4 所示的“Rename NONAME”更名文本框，只是其中的盘符及目录路径“D:\TOOLS\GRABBER\”会因工作目录的不同而不同。

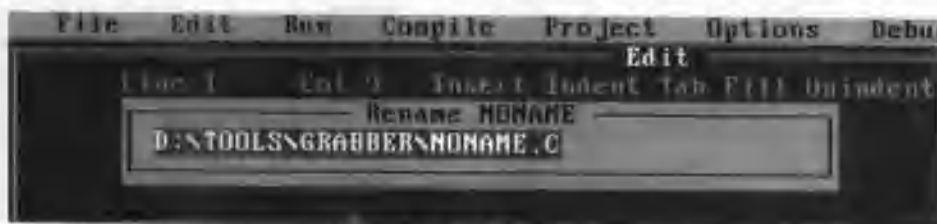


图 2-4 更名文本框

◆ 若只更改文件名，利用箭头键，将光标移至要更改的文件名的某个字符，删除需更改的所有字符，键入新文件名（如例程 2.1，在文本框中删除 NONAME.C 键入 MCP1）回车即可。

注意: (1) 此时源文件名不分大小写; 不必输入扩展名, 系统会自动添加。

(2) 注意搜索箭头键、退格键、Delete 键的使用规律, 尽量减少击键次数和更改的字符数。

(3) 此后编辑源文件, 再按 F2 键, TC 系统源信息行中的 * 号隐去外不再显示任何变化信息, 但已将 MOD1.C 文件刷新保存。

• 若还需更改目录路径, 通过左、右光标键, 将光标移至要更改的目录; 若还需更改盘符, 则可按退格键或 Delete 键, 将光标一步退到位, 键入新盘符、路径、文件名后, 击 Enter 键 (此时信息行中的 * 号将隐去)。

• 若此时的光标在第 7 行的末尾, 可先按 Ctrl+K+K 键, 设置块尾。将光标移至第 7 行的字符 p 再按 Ctrl+K+B, 设置块首, 完成了对 “putchar (a);” 的块设置。

• 按 End 键, 将光标跳移至行末尾, 按 Ctrl+K+C, 完成块复制。将光标移至复制的 “putchar (a);” 括号中, 把字符 a 更改为字符 b。

• 再按 End 键, 将光标跳移至行末尾, 按 Ctrl+K+C, 完成第 2 次块复制。同样的方法, 将括号中的字符 a 更改为 c。将光标跳移至行末尾, 按 Enter 键, 便转入新的第 8 行, 按 Ctrl+K+C, 更改括号中字符等, 用同样的方法步骤, 完成其他的块复制与字符更改。

注意: 设置块首、块尾可不分先后, 而块的设置与操作形式多样, 参照 “2.1.6.C 程序建立、保存及常用编辑技巧”, 摸索规律, 灵活应用, 提高编程效率。

• 照例程 2.1 完成第 9、10 行的录入编辑, 观察信息行中又显示出 * 号; 按 F2 键, 注意主屏中有无存盘更名文本框出现? 再观察信息行, 又将隐去 * 号。为防止停电等意外情况丢失程序, 最好经常按 F2 键, 及时将源程序新内容存盘。完成源程序编辑后的主屏参见图 2-5。

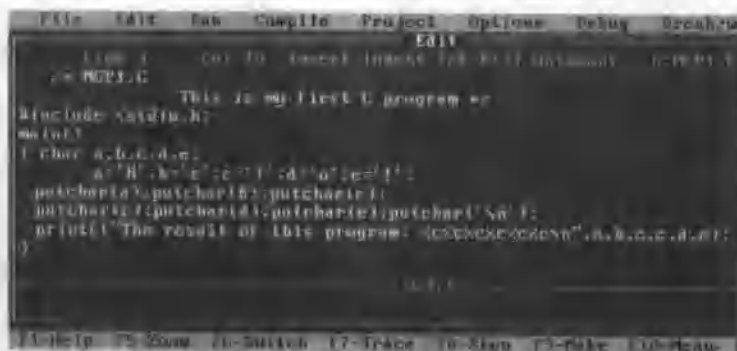


图 2-5 简单的 C 程序主屏

(3) 简单 C 程序组成部分

预处理命令; 主函数名; 函数体起始符; 函数体;
函数体终止符; 可放置在任意位置的注释。

(4) 编写 TC 源程序注意事项

1) TC 严格区分大小写, 程序一般用小写; 同一变量或函数名, 如果前后大小写不一致, 则编译的时候将出现编译错误。

2) 语句结尾一般必须要用分号作为终止符, 否则 TC 不会知道该语句结束。

- 3) 每个程序(项目)必须有一个并且只能有一个称作主函数的 `main()` 函数。
- 4) 每个程序体必须有一对花括号括起来。
- 5) 注释部分包含在注释起始符和注释终止符之间。

2.2.2 C 程序的运行

TC 系统提供了两种运行程序的方式操作。一种是保留了传统的方式:先编译,后链接,再运行,分散操作完成;另一种方式是集编译、链接、运行为一体,只需执行一个菜单项或一个复合热键即可完成。

(1) 先编译、后链接、再运行方式

编译:按热键 `Alt+F9` 或选择主菜单 `Compile` 中的 `Compile to OBJ` 菜单项。屏幕上将出现“Compiling”消息窗口,显示被编译的主(源)文件所在盘符、路径、文件名等信息。当编译消息窗口底部兰色信息条显示 `Success:Press any key` 且 `:Press any key` 在不停地闪烁,表示编译成功,按任意键返回编辑窗口,编译消息窗口消失。此时,在源文件目录下,将生成一个与源文件同名但扩展名为 `OBJ` 的文件(如例程 2.1 即为: `MCPI.OBJ`)。

若程序中有错误,编译时,将产生警告、错误等信息。例如我们将例程中第 3 行的“#”号去掉,编译时,编译窗口中提示出错信息统计,底部兰色信息条显示 `Errors:Press any key`,按任意键,编译消息窗口消失,而编辑窗口和消息窗口中分别有一条先带,指示程序中的错误位置和错误说明。按回车、按 `Esc` 键,去掉编辑窗口顶行的红色警示信息条,纠正所有错误,重新编译才能成功。

链接:按热键 `F9` 或选择主菜单 `Compile` 中的 `Make EXE file` 菜单项。屏幕上将出现“Linking”消息窗口。链接消息窗口底部兰色信息条显示 `Success:Press any key` 表示链接成功,按任意键,返回编辑窗口,链接消息窗口消失。主屏消息窗口显示链接成功的扩展名为 `EXE` 的文件名及其所在的盘符和路径。此时,在源文件目录下,将生成一个与源文件同名但扩展名为 `EXE` 的文件(如例程 2.1 即为: `MCPI.EXE`)。

与 C 语言本身具备的特点一样,TC 集成系统也十分灵活自如。按传统方式,只有编译生成 `OBJ` 文件后才能链接,可 TC 集成系统在没有生成 `OBJ` 文件的情况下操作链接,TC 集成系统会自动先补作编译,再作链接。

经编译、链接操作后,并没运行程序,所以在 DOS 显示方式的用户屏上,看不到程序的运行结果。

运行:由于此时 TC 已生成脱离自身系统、能在 DOS 环境下独立运行的 `EXE` 文件,要运行程序,可选择主菜单 `File` 中的 `Os shell` 菜单项,然后在 DOS 提示符下,直接键入程序文件名(如例程 2.1,在 DOS 提示符“`D:\MAYDR>`”键入 `MCPI`)回车,即可运行程序,并能看到运行的结果。

(2) 集成运行方式

集成运行方式(有的也称直接运行方式)的操作是按 `Ctrl+F9` 复合热键或选择主菜

单 Run 中的 Run 菜单项；如果源程序输入无错而采用集成方式运行程序时，通常只见屏幕闪烁一下，又回到运行前的主屏。但此时，TC 已完成编译、链接、运行操作，并且用户屏上还有程序运行的结果。如果源程序已经作过存盘操作，那在默认和指定的目录下将有如下三个文件（以例程 2.1 为例）：

MCPI.C 源程序文件：用户用 C 语言编写的程序文本。其扩展名为 C，在 DOS 提示符下，键入 "TYPE MCPI.C"，可在屏幕上显示此文本的内容。

MCPI.OBJ 目标文件：存放 TC 编译程序产生的中间代码，其扩展名为 .OBJ。源程序被翻译成目标程序后，不能直接运行，必须链接。链接的目的是使程序成为在计算机上可以执行的最终形式。在这一阶段，从系统程序库来的程序与目标程序链接，链接的结果称为执行程序。不能用 DOS 的 TYPE 命令显示文件内容。

MCPI.EXE 执行文件：由 TC 链接程序生成的直接执行的文件。在 DOS 提示符下，键入 MCPI 再回车，用户屏幕会有运行结果。

要查看以上三个文件，可选择主菜单 File 中的 Os shell 菜单项，暂时进入 DOS 状态屏幕，键入 DOS 命令：dir mcpi.* 计算机便列出这三个文件。键入 exit 命令后，击 Enter 键，计算机又回到 TC 主屏幕。

如果源程序输入有错，集成运行将显示编译错误消息窗口（完全同于图 2-7），并终止集成系统运行。只有纠正所有错误，集成运行才能通过。

(3) 查看运行结果

只要按 Alt+F5 复合热键，就会从 TC 主屏切换到用户程序结果屏幕。当程序集成方式运行成功后，此时，便能在用户屏幕上看到程序运行的结果。再按任意键，又切换回 TC 主屏。

例程 2.1 的运行结果参见图 2-6。其中：

第 7 行输出 "Hel" 且不换行，第 8 行接着输出 "lo!"；8 行中的 putchar ('\n') 为换行操作。

第 9 行输出 "The result of this program : Hello!"。

第 7 行输出 — 第 8 行输出

```

Hello!
The result of this program: Hello!
—

```

第 9 行输出

图 2-6 例程 2.1 的运行结果。

2.2.3 C 程序的调试与纠错

一个程序设计好后，通常难免会有些错误，查找和修改程序中的错误是令人头痛的事，这是初学者常感棘手的问题，但也是初学者理解和牢固掌握 Turbo C 的最好老师。因此，Turbo C 集成开发环境提供了一个调试装置，使这一工作容易了许多。程序的调

试可以直接在开发环境中完成。

(1) 程序的编译及错误提示

TC 集成系统编译 (按 Alt+F9) 时, 会弹出图 2-7 的编译消息框并将“编译器”和“链接器”发出的消息, 收集到一个缓冲区中, 然后显示在消息窗口中。这样在编译调试源代码的同时, 还能同时看到这些消息。现通过【例 2.2】再结合图 2-8 学习 C 程序的编译、调试方法。

【例 2.2】输入 1 个 16 进制数, 输出其对应的 10 进制数。

说明: 为了有利于学习, 特意在程序中设置了一些错误, 在错误行用注释语句作了标明, 但读者先不用理会, 这样, 在学习编译时印象会更深刻。

```
/* MCP2.C
```

```
Input HEX output DECit */
```

```
#include <stdio.h>
```

```
main ( )
```

```
{
```

```
int num;
```

```
printf ( "please input num of HEX:" ); /* ( 7 行) Take the p of printf */
```

```
scanf ( "%x",&Num ); /* ( 8 行) 'Num' displace 'num' */
```

```
printf ( "num of DEC is: %x\n", num ); /* ( 9 行) 'x' displace 'd' */
```

```
/* ( 10 行) Take the */
```

按 Alt+F9 键编译 MCP2.C, 编译信息框将显示有 Errors (错误) 信息和 Warnings (警告) 信息, 如图 2-7。

注意: 在编辑运行两个以上的 C 程序时, 最好先按 F9 键 (即先编译、连接) 后再运行程序, 这样才能保证编译、运行的是当前主屏上的程序。

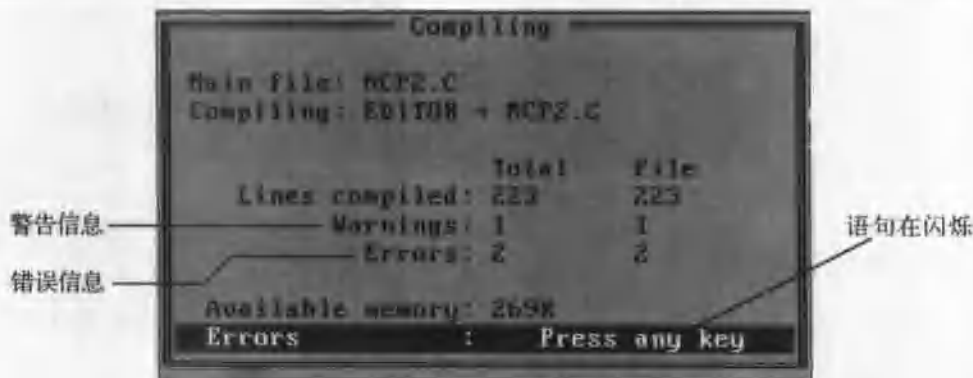


图 2-7 编译信息框

当编译信息框底行出现“Errors : Press any key”提示时，按空格键，消息窗口立即被激活（参见图 2-8），灰色光带出现在第 1 个错误或警告提示上，同时编辑窗口中也有 1 光带，它标志着编译器给出的错误或警告在源代码中的相应位置。这时可用上、下方向键移动消息窗中光带，编辑窗中的光带也随着跟踪源代码中错误发生的位置。第 2 条信息太长，消息窗口显示不下，可按左、右光标键，水平滚动提示信息。

Trubo C 的错误信息由 4 部分组成，以第 1 条为例，4 部分的组成参见图 2-8 图注。



图 2-8 消息窗口中错误信息的组成

如消息窗口中的第 1 个错误（第 2 行）：

第 1 部分：错误性质：Errors：致命错误；

第 2 部分：出错文件名及路径；

第 3 部分：错误发生在第八行；

第 4 部分：错误类型为 main 函数中的“Num”没有定义。

(2) 排除语法错误

在修改语法错误过程当中，经常需要增加、删除正文，编辑器将记住这些修改，并且依然能正确定出错误位置，因此没有必要记住增、删的正文行和错误发生行的行号。

将光带置于第 3 个错误信息上，按下回车键，光标移到编辑窗口中错误产生处，同时编辑窗口顶部红色错误状态行显示错误类型信息，原来我们写丢 1 个右花括号，现加上“}”。修改下一错误，只要按 Alt+F7 键，编译器就会在编辑窗口中将光标移至前一个错误所在的位置，同理按 Alt+F8 键，可移至下一个错误。

修改下 1 错误，将第 8 行 Num 的大写 N 改为小写的 n。从消息窗口中的第 3 行知道，在编辑窗口第九行还有一个警告提示，可是根据前面所学知识分析，应该没有错误，重新编译一下程序，这条警告信息没有了。问题出在哪里，原来源代码中的一个错误“Num”把编译器弄糊涂了，产生了多余的消息。此时，编译窗口显示编译成功的信息。

下面需要作的是链接。按 F9 功能键，链接信息框将提示有错误信息。按任意键，链接信息框消失，而消息窗口显示如图 2-9 所示的另一错误，是链接类型错误，因为找

不到 “_rprintf” 函数。下划线是 Turbo C 对函数的内部表示方法。



图 2-9 消息中的链接错误信息

这是由于源程序第 7 行中将 printf 写为 rprintf, 而 C 语言的函数库中, 没有 rprintf 函数, Turbo C 在编译链接过程中找不到该函数, 所以显示错误信息。纠正后再链接, 链接信息框提示链接成功。

从以上调试中, 我们可以看出 Turbo C 对 C 语言程序的处理过程: 首先, 让语言编译器寻找程序中的语法错误, 并显示在消息窗口中, 如果没有语法错误, 则生成程序的目标代码, 然后进行程序目标代码与库函数的链接过程。如果在链接过程中, 找不到所需要的函数, 则显示链接错误信息, 若没有链接错误, 则编译器生成程序的执行文件。

(3) 排除逻辑错误

例 2.2 经过以上调试修改, 编译、链接均已通过, 下面便可运行程序观察结果。

按 Ctrl+F9, 程序第 1 次运行提示输入 1 个十六进制数。输入 “0x11” (十六进制数 11) 后按回车键, 再按 Alt+F5, 观察程序输出结果是 11 而不是正确的 17 (参见图 2-10)。

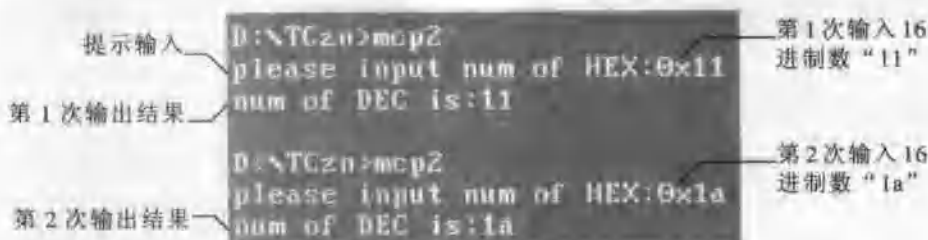


图 2-10 运行结果错误

再按 Ctrl+F9, 第 2 次运行, 输入十六进制数 “0x1a”, 再按 Alt+F5 键, 程序的输出仍然是 1a 而不是正确的十进制数 26, 参见图 2-10。可见程序运行结果出现了错误。此时可用 Turbo C 集成调试器来找出程序的错误。从这个过程中, 我们可以看出, 程序修改完语法错误以后, 虽然编译能顺利通过, 却还是不按要求运行, 这因为程序可能有逻辑错误, 这种错误常称为运行错误。这时编译器的错误检查器就无助于发现错误位置。此时, 可用以下方法调试。

(4) 单步跟踪调试排错

Turbo C 有一个集成调试器, 可以跟踪运行错误。先单步运行程序, 在快速运行到指定的语句处暂停, 如例程 2.2 的第 9 行。按 Ctrl+F4, 通过集成调试器 Debug 主菜单的 Evaluate 菜单项, 检查变量 num 的值 (参见图 2-11)。甚至可以改变 num 变量的值, 即在 New value 框中, 键入合法值, 以便看程序的反映。

Turbo C 的集成调试器, 是源程序的调试器, 也就是说调试器是以 C 语言的语句行