

1 Turbo C 上机指南

1.1 Turbo C 的安装

Turbo C 由美国 Borland 公司研制生产，是 C 语言的程序开发软件，具有速度快、效率高、功能强等特点，使用非常方便。

Turbo C 对系统要求不高，占用的系统资源很少，对硬件没有过高的要求。在 Turbo C 的编辑环境下没有鼠标器。

Turbo C 是基于 DOS 平台的编译系统，安装 Turbo C 时要求在计算机上安装 DOS 2.0 或以上的版本。

Turbo C 一般放在光盘上或软盘上，也可以直接从网上下载。在光盘或软盘上的 Turbo C 系统中有一个安装程序 `install.exe`，安装时直接运行该程序即可。从网上下载的 Turbo C 系统一般是个压缩文件，在使用之前需要经过解压缩处理。为了能给 Turbo C 系统提供一个汉字处理环境，首先应安装中文操作系统 CCDOS 或 UCDOS，然后再在 CCDOS 或 UCDOS 状态下运行 Turbo C 系统的可执行文件 `TC.EXE`。通过运行该程序就可以进入用户在磁盘（一般是在 C 盘）上安装的 Turbo C 开发环境。

安装的过程完成后，将在用户的磁盘上增加以下子目录和文件。

- (1) C:\TC：其中包括 `tc.exe`、`tcc.exe`、`make.exe` 等执行文件。
- (2) C:\TC\INCLUDE：其中包括 `stdio.h`、`math.h`、`malloc.h`、`string.h` 等头文件。
- (3) C:\TC\LIB：其中包括 `maths.lib`、`mathl.lib`、`graphics.lib` 等库函数文件。
- (4) C:\TC\BGI 和 C:\TC\C：其中包括 Turbo C 运行时所需的信息。

1.2 Turbo C 编译系统部分功能介绍

Turbo C 安装成功后，在 Windows 系统的“开始”菜单中选择“运行”菜单项，键入“cmd”命令即可进入 DOS 状态，在 DOS 提示符“C:\>”下键入：

```
C:\>cd\ccdos✓
```

```
C:\CCDOS>ccdos✓
```

上述命令为编译系统建立一个汉字输入环境。在此环境下再键入：

```
C:\CCDOS>cd\tc✓
```

```
C:\TC>tc✓
```

运行 `TC.EXE` 就可以进入 Turbo C 系统集成编译环境。也可在存放 Turbo C 系统程序的磁盘上直接运行文件 `TC.EXE`，但这种方式不具备汉字输入环境。

1. 主窗口

Turbo C 系统集成环境主窗口如图 1-1 所示。

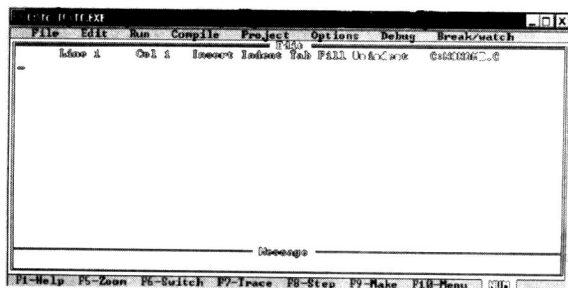


图 1-1

它包括以下几个组成部分：

(1) 主菜单窗口：在 TC 系统屏幕的顶部。它包括 8 个主菜单项：File（文件）、Edit（编辑）、Run（运行）、Compile（编译）、Project（项目）、Options（选项）、Debug（调试）、Break/watch（断点/监视）。每一个主菜单都还有下一级子菜单，分别用来实现各菜单项的具体操作。

(2) 编辑窗口：在主菜单窗口的下面，正上方有 Edit 字样作为标志。编辑窗口的作用是对 Turbo C 源程序进行输入和编辑。用户输入或调用的源程序都在这个窗口中显示，因而编辑窗口占据了屏幕的大部分面积。

在编辑窗口的上部有一行英文：

Line 1 Col 1 Insert Indent Tab Fill Unindent c:\NONAME.C

其中，Line 和 Col 表示当前光标的位置。Line 1 和 Col 1 表示当前光标的位置在编辑窗口中的第 1 行第 1 列。当光标移动时，Line 和 Col 后面的数字也随之改变，它可以方便地显示出光标当前所在的位置。

该行最右端显示的是当前正在编辑的文件名，对新文件自动命名为：NONAME.C。如果从磁盘调入一个已存在的文件，则在该位置上显示的不再是：NONAME.C，而是该文件的名字。

(3) 信息窗口：在屏幕的下部，用来显示编译和连接时的有关信息。在信息窗口上方有“Message”字样作为标志。在编辑源程序时不用此窗口。

(4) 功能键提示行：在屏幕最下方（在信息窗口的下面）。它显示一些功能键的作用。例如：

F1-Help（帮助）：任何时候按 F1 键都会显示帮助信息。

F5-Zoom（分区控制）：如果当前在编辑窗口工作，也就是说编辑窗口是激活的，按 F5 键就不显示信息窗口，它的作用是扩大编辑窗口，以便能容纳和显示较长的源程序。若再按一次 F5 键，就会恢复信息窗口；如果当前信息窗口是激活的，按 F5 键就不显示编辑窗口，它的作用是扩大信息窗口，以便能显示较多的信息，便于用户观看。若再按一次 F5 键，就会恢复原状。

F6-Switch（切换）：按 F6 键就能在信息窗口与编辑窗口之间进行切换，切换后激活窗口中的标题是高亮显示的。

F7-Trace（跟踪）：用于跟踪程序的运行情况。

F8-Step（按步执行）：按一次键执行一个语句。

F9-Make（生成目标文件）：进行编译和连接，生成目标文件，但不进行运行。

F10-Menu（菜单）：回到主菜单，激活第一个菜单（此时以高亮显示）。

为了从主菜单中选择所需要的项目，可以用以下两种方法：

（1）按 F10 键，可以看到屏幕上部菜单中的某个项目处出现亮块，此时，利用左、右光标移动键（→与←）将此亮块移到所要选择的项目位置处，然后按回车键，即出现相应的子菜单，再利用上、下光标移动键（↓与↑）来移动高亮度条，从而选择所需要的功能。

（2）直接按 Alt+主菜单项目中的首字母（分别是 F、E、R、C、P、O、D、B），此时就会出现相应的子菜单。当出现子菜单后，其中子菜单中的某个项目是高亮度显示的，此时可以利用上、下光标移动键（↓与↑）来移动该高亮度条，从而选择所需要的功能。

在主菜单或通过主菜单调用的任意一个子菜单中，按 ESC 键后将直接返回到活动窗口。

下面对各个主菜单进行说明。

2. File 菜单的功能

按 ALT+F 然后回车或按 F10 键都可以选中该主菜单项，在“File”下方将出现一个子菜单窗口，如图 1-2 所示：

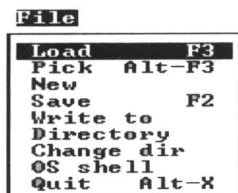


图 1-2

在该子菜单中，有的子菜单项右边还标出了可直接调用该项目的热键。所谓热键，是指系统为执行菜单中某一固定功能而设置的键。通过热键实现某种功能，一般要比从菜单逐层选择更简单直接，但要求用户熟记这些热键。例如，调用子菜单中的“Save”项，可以直接按 F2 键。

下面简要说明“文件”子菜单中各项目的功能：

Load（加载）：装入一个文件，可用类似 DOS 的通配符（如*.C）来进行列表选择。也可装入其他扩展名的文件，只要给出文件名（或只给路径）即可。该项的热键为 F3，即只要在主窗口中按 F3 即可进入该项，而不需要先进入 File 菜单再选此项。

Pick（选择）：将最近装入编辑窗口的多个文件列成表，供用户选择，选择后再装入编辑器，光标放置在上次修改过的地方。其热键为 Alt+F3。若选了“...Load file...”，屏幕上将出现“Load File Name”提示符。

New（新建文件）：所编辑的文件是新的，装入编辑器的默认文件名为“NONAME.C”，存盘时可改名。

Save（存盘）：将编辑器中的文件内容存盘。若存盘的文件名为“NONAME.C”，则系统会询问是否需要改名。其热键为 F2。

Write to（存盘）：将编辑器中的文件内容写入指定的文件中。若该文件已经存在，则导致重写（即该原文件中原来的内容被破坏）。

Directory（目录）：显示目录与所需文件列表（若按回车键，则选择当前目录）。

Change dir (改变驱动器): 显示当前目录, 用户可以改变当前驱动器与目录。

OS Shell (暂时退出) 暂时退出 Turbo C 转到 DOS 状态。在 DOS 状态下用命令 “ EXIT ” 可返回到 Turbo C 。此功能在想运行 DOS 命令但又不想退出 Turbo C 时有用。

Quit (退出): 退出 Turbo C , 返回到 DOS 状态。其热键为 Alt+X。

3. Edit 菜单的功能

“Edit ” 子菜单调用 Turbo C 集成环境中的内部编辑器。在主窗口中, 按 Alt+E 可直接打开该菜单, 若再回车, 则光标出现在编辑窗口, 此时用户可以进行文本编辑。编辑方法基本与 WordStar 相同, 可用 F1 键获得有关编辑方法的帮助信息。

与编辑有关的功能键如下:

- F1 获得 Turbo C 2.0 编辑命令的帮助信息
- F5 扩大编辑窗口到整个屏幕
- F6 在编辑窗口与信息窗口之间进行切换
- F10 从编辑窗口转到主菜单

编辑命令简介:

- PageUp 向前翻页
- PageDn 向后翻页
- Home 将光标移到所在行的并始
- End 将光标移到所在行的结尾
- Ctrl+Y 删除光标所在的一行
- Ctrl+T 删除光标所在处的一个词
- Ctrl+KB 设置块开始
- Ctrl+KK 设置块结尾
- Ctrl+KV 块移动
- Ctrl+KC 块拷贝
- Ctrl+KY 块删除
- Ctrl+KR 读文件
- Ctrl+KW 存文件
- Ctrl+KP 块文件打印

4. Run 菜单的功能

当选中主菜单的 “Run ” 子菜单后, 在 “ Run ” 下方将出现一个子菜单窗口, 其子菜单项也列出了一些对应的热键。如图 1-3 所示。

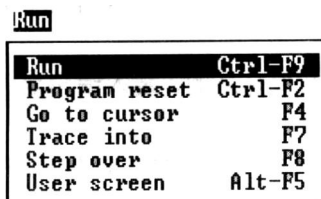


图 1-3

下面简要说明该子菜单中各项目的功能:

Run (运行) 运行由 Project/Project name 项指定的文件名或当前编辑窗口的文件。如

果对上次编译后的源代码未作过修改，则直接运行到下一个断点（没有断点则运行到结束）；否则先进行编译、连接后才运行。其热键为 **Ctrl+F9**。

Program reset（程序重启动）：终止当前调试，释放分配给程序的空间，关闭已打开的文件。其热键为 **Ctrl+F2**。

Go to cursor（执行到）：使程序从运行到编辑窗口中光标所在行。若光标所在行不含可执行代码语句，则显示一个 **Esc** 框作警告。其热键为 **F4**。

Trace into（跟踪进入）：运行当前函数中的下一个语句。若此语句不含调试器可访问的函数调用，则停在下一条可执行语句上；但若此语句含有调试器可访问的函数调用，则停在函数定义的开始。其热键为 **F7**。

Step over（单步执行）：执行当前函数的下一个语句，即使遇到调试器可访问的函数调用也不会跟踪进入下级函数中。其热键为 **F8**。

User screen（用户屏幕）切换到用户屏幕。其热键为 **Alt+F5**。

5. Compile 菜单的功能

选中主菜单的“**Compile**”子菜单后，在“**Compile**”下方将出现一个子菜单窗口。如图 1-4 所示。

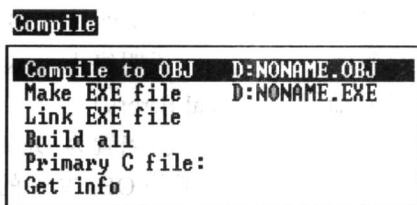


图 1-4

Compile to OBJ（编译生成目标码）本命令将一个 **.C** 源文件编译成 **.OBJ** 文件，同时显示生成的 **.OBJ** 文件名。**.OBJ** 文件由指定的 **.C** 文件生成：如果没有指定 **.C** 文件时，则由上次装入编辑器的 **.C** 文件产生。**Turbo C** 在编译时弹出一个窗口，用于显示编译结果。在编译/组装完成后，按任意键将清除编译窗口。此时，若发现有错误，则转到消息窗口的第一个错误处（高亮度标识）。本命令的热键为 **ALT+F9**。

Make EXE file（生成可执行文件）本命令调用 **Project/Make** 命令来生成 **.EXE** 文件，并显示所生成的 **.EXE** 文件名。**.EXE** 文件依次由下列文件产生：**Project/Project Name** 说明的项目文件；或 **Project C File** 说明的文件；或上次装入窗口的文件。

Link EXE file（连接可执行文件）：把当前 **.OBJ** 文件与库文件（既可以是缺省的，也可以是定义在当前项目 **project** 中的文件）连接在一起，生成 **.EXE** 文件。

Build all（建立所有文件）：重建项目中的所有文件。本命令类似 **Compile/Make EXE File**，它是无条件执行的，而 **Compile/Make EXE File** 只重建那些非过时的文件。本命令首先将所有的 **project** 文件中的 **.OBJ** 文件的日期与时间置为 0，然后再组装。这样，若用 **Ctrl+Break** 键中断了 **Build all** 命令，用 **Compile/Make EXE File** 即可恢复。

Primary C file（主 C 文件）当编译含多个 **.h** 文件的单个 **.C** 文件时，**Primary C file** 命令是很有用的（但并非必要的）。若在编译过程中发现错误，包含错误的文件（**.h** 或 **.C**）将被自动装入编辑器，可对其修改。但必须注意，**.h** 文件只有在已将 **Options/Environment/Message**

Tracking 缺省设置改为 All File 时才能自动装入，而原缺省设置不会自动加载.h 文件。即使.C 文件不在编辑器，但只要按 Alt+F9 键，.C 文件即被重新编译。

Get info (获得信息): Get info 可以给出如下信息：源文件、与当前文件相联系的目标文件、当前源文件、当前源文件的字节数、程序退出码、可用空间等。

6. Project 菜单的功能

当选中主菜单的“Project”子菜单后，在“Project”下方将出现一个子菜单窗口。如图 1-5 所示。

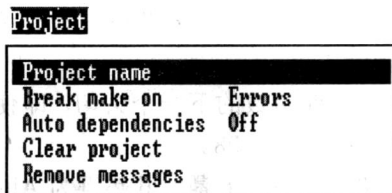


图 1-5

下面简要说明该子菜单中各项目的功能：

Project name (项目名)：选择一个包含将要编译、连接文件的 Project 文件，项目名将是以后要建立的 .EXE 或 .MAP 文件名。典型的项目文件具有 .PRJ 扩展名。

Break make on (中止执行条件)：提供用户说明终止 make 命令的缺省条件，如警告 (Warnings)、错误 (Errors) 和致命错误 (Fatal Errors)。

Auto dependencies (自动依赖) 这是一个开关。当置为 On 时，项目组装 (Project-Make) 自动检查每个项目表中在磁盘上相应 .C 文件的那些 .OBJ 文件的依赖关系。所谓自动依赖关系检查，是指项目组装打开 .OBJ 文件，寻找包含在源代码的那些文件的有关信息。这种信息总是在编译源模块时即被 TC 或 TCC 放进 .OBJ 文件。此时，把每一个组成 .OBJ 文件的源文件的日期 / 时间信息与 .OBJ 中的时间进行比较，若不同，则重新编译 .C 源文件。若 Auto dependencies 开关置为 Off，则不进行这种检查。

Clear project (清除项目)：该命令将清除项目文件名，重置消息窗口 (Message window)。

Remove message (删除消息)：该命令将错误消息从消息窗口中清除掉。

Get info (获得信息) 获得有关信息。

7. Debug 菜单的功能

当选中主菜单的“Debug”子菜单后，在“Debug”下方将出现一个子菜单窗口。如图 1-6 所示。

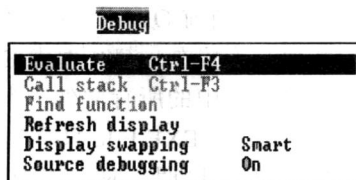


图 1-6

下面简要说明该子菜单中各项目的功能：

Evaluate (计算)：计算变量或表达式值，并显示其结果。

Call stack (调用栈)：本命令显示一个包含调用栈的弹出窗口。调用栈显示程序运行

到正在运行的函数时调用的函数序列。其中主函数 `main()` 在栈底，正在运行的函数在栈顶。调用栈中的每一项显示了函数名以及传递给它的参数值。

Find function（查找函数定义）：显示编辑窗口中每一函数的定义。只有在调试阶段才能使用本命令。

Refresh display（刷新显示）：如果编辑屏幕被重写，使用本命令可以恢复当前屏幕的内容。

Display swapping（显示转换）本命令提供 3 种选择：`Smart`、`Always` 和 `None`。

Smart：一般为缺省值，选择 `Smart` 时，仅当代码被执行时才向屏幕产生输出。

Always：选择 `Always` 时，每个时刻都有一个声明执行。

None：选择 `None` 时，只能用于没有输出的代码，否则输出会改写编辑窗口。

Source debugging（源代码调试）本命令提供 3 种选择：`On`、`Standalone` 和 `None`。

On：选择 `On` 时，调试信息放到 `.EXE` 文件的地方，被调试器使用。

Standalone：选择 `Standalone` 时，调试信息放到 `.EXE` 文件的地方，但不被调试器使用。

None：选择 `None` 时，调试信息不放到 `.EXE` 文件的地方，也不被调试器使用。

8. Break/watch 菜单的功能

当选中主菜单的“`Break/watch`”子菜单后，在“`Break/watch`”下方将出现一个子菜单窗口。如图 1-7 所示。

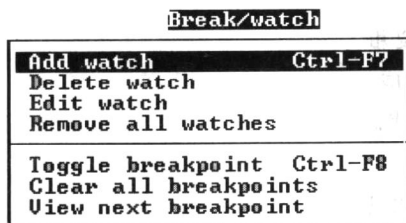


图 1-7

下面简要说明该子菜单中各项目的功能：

Add watch（增加监视表达式）：向监视窗口插入一个监视表达式。

Delete watch（删除监视表达式）：从监视窗口中删除当前的监视表达式。

Edit watch（编辑监视表达式）：选择本命令后，调试器弹出一个含有当前监视表达式的窗口。

Remove all watches（删除所有监视表达式）：将所有监视表达式从监视窗口中删除。

Toggle breakpoint（打开或关闭断点）：设置或去除光标所在断点。

Clear all breakpoints（清除所有断点）：从程序中删除所有断点。

View next breakpoint（显示下一个断点）：将光标移到程序中的下一个断点。

9. Options 菜单的功能

当选中主菜单的“`Options`”子菜单后，在“`Options`”下方将出现一个子菜单窗口。如图 1-8 所示。

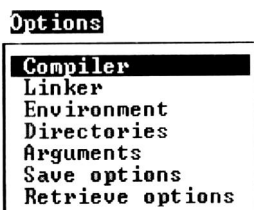


图 1-8

下在简要说明该子菜单中各项目的功能：

(1) Compiler (编译器)：本命令将产生一个子菜单，为硬件配置、存储模式、调试技术、代码优化、诊断消息控制以及宏定义等提供说明。其子菜单项如图 1-9 所示。

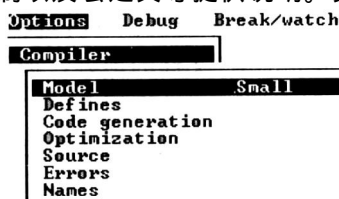


图 1-9

Model：选择存储模式。

Defines：打开一个宏定义框。

Code generation：代码生成。

Optimization：优化用户代码。

Source：处理源代码。

Errors：处理和响应诊断信息。

Names：改变代码、数据等。

其中每一个子菜单项又对应一个子菜单，供用户选择各种功能。详细介绍请参看 Turbo C 用户手册。

(2) Linker (连接器)：本命令将产生有关连接器的设置，其内容如图 1-10 所示。

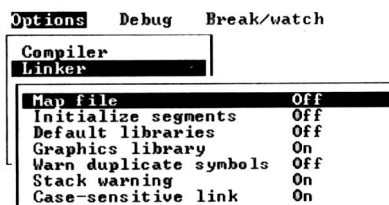


图 1-10

Map file：选择映射文件的类型 On/Off，缺省值为 Off。

Initialize segments：段初始化 On/Off，缺省值为 Off。

Default libraries：缺省库 On/Off，缺省值为 On。

Graphics library：图形库 On/Off，缺省值为 On。

Warn duplicate symbols：警告重复字符 On/Off，缺省值为 On。

Stack warning：堆栈警告 On/Off 缺省值为 On。

Case-sensitive link : 大小写区别连接 ON/Off , 缺省值为 On。

(3) Environment (环境) : 本命令将产生编译环境的设置, 其内容如图 1-11 所示。

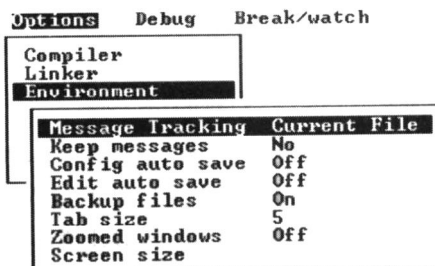


图 1-11

Message tracking: 消息跟踪 Current File/All File/Off , 缺省值为 Current File。

Keep message : 保存消息 Yes/No 缺省值为 No。

Config auto save : 配置自动保存 On/Off , 缺省值为 On。

Edit auto save : 编辑自动保存 On/Off , 缺省值为 Off。

Backup files : 备份文件 On/Off , 缺省值为 On。

Tab size : 制表键大小, 缺省值为 8。

Zoomed windows: 放大窗口 On/Off , 缺省值为 Off。

Screen size : 选择屏幕显示行数。

(4) Directories (目录) : 本命令告诉系统到哪里去寻找编译连接所需要的文件, 生成的可执行文件放到何处, 在哪里查找配置文件、选择文件和帮助文件。其内容如图 1-12 所示。

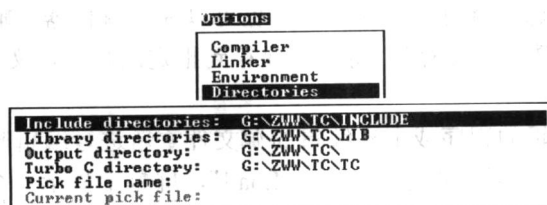


图 1-12

Include directories : 包含目录: C:\TURBOC\INCLUDE; C:\TURBOC\IN。

Library directories : 库目录: C:\TURBOC\LIB。

Output directory : 输出目录。

Turbo C directory: Turbo C 目录。

Pick file name : 选择文件名。

Current pick file : 当前选择文件。

(5) Arguments (参数): 本命令设置允许用户给出运行程序命令行。

(6) Save options (保存选项): 本命令将选择的编译器、连接器环境、调试和选项保存到一个配置文件中 (缺省文件名为 TCCONFIG.TC)。启动时, Turbo C 再到目录 TURBOC 中查找同样的文件。

(7) Retrieve options (恢复选项) 本命令用以加载以前用 Option/Save options 命令保存的配置文件。

1.3 文本编辑

1.3.1 编辑新文件

如果要输入和编辑一个新的 C 程序，应选择“File”菜单中的子菜单项“New”，则编辑窗口被清空，光标定位在第 1 行第 1 列，用户可以开始输入和编辑源程序了。

在编辑过程中除各字符键外，还可以用到 Ins 和 Del 键。Ins 键是一个切换键，用来控制工作状态是否为“插入状态”。这可从屏幕编辑窗口的上方状态行的英文单词 INSERT 来判断：当这个单词出现时，是插入状态；想要取消该状态时，只需按下 Ins 键即可。Del 键是删除光标所在的字符。还可通过上下左右（↑、↓、→、←）移动键来改变光标所在的行和列。

输入程序后应对程序作认真检查，并改正已发现的错误，然后将源程序保存起来。保存文件可用“File”菜单中的子菜单项“Save”或用热键 F2。TC 弹出一个对话框，可在对话框中指定文件路径，如果不指定路径则保存在 C 盘根目录下。保存后，该文件名将出现在编辑窗口的右上角。

1.3.2 编辑已有的文件

在 TC 环境下编辑文本不是十分方便，我们可以借助其他文本编辑器来完成。比如 Windows 中的“记事本”，编辑修改都很方便。用“记事本”编辑时，中英文切换直接在 Windows 的系统托盘中选择就可以完成，同时也可以使用鼠标器。输入完成后，在“记事本”的“文件”菜单中选择“保存”，将文件保存到指定的目录，文件扩展名可以是.TXT，也可以是.C。

对用“记事本”编辑的程序或上次编辑的源文件重新进行编辑时，需要将其从磁盘中调出来。选择“File”菜单项中的子菜单项“Load”，屏幕上显示一个包含*.C 的对话框，要求用户输入准备调入的文件路径和文件名。

如果用户输入的文件名不存在，则屏幕上一片空白，表示文件无内容。如果记不清所要装入的源文件名，可以在目录中查看。在对话框中输入目录后按回车键，则将该目录下的文件显示出来。

1.4 程序的编译、连接及运行

编辑好源程序并存盘后，应当对源程序进行编译、连接和运行。在 Turbo C 集成环境中，进行编译、连接和运行是十分方便的。既可以将编译、连接和运行分 3 个步骤分别进行，也可以将编译和连接合起来作为一步进行，然后再运行；还可以将编译、连接和运行三者合在一起一次完成：既可以对单个文件模块的程序进行编译、连接和运行，也可以一次对多个文件模块的程序进行编译、连接和运行。

1.4.1 编译及连接

1. 对单个文件的编译、连接

对单个文件既可以分别进行编译、连接，也可以一次完成编译和连接。

当分别进行编译和连接时，一定要先进行编译，然后再进行连接。按 F10 键，将亮条移至“Compile”处后回车或者直接按热键 ALT+C 调出编译菜单，选择子菜单项“Compile to OBJ”，此时系统显示出默认的目标文件名，按回车键进行编译，生成一个与源文件同名的目标文件。有了目标文件后，选择编译菜单的子菜单项“Compile/Link EXE file”，按回车键执行连接，在当前目录下生成一个与源文件同名的可执行文件。

如果选择编译菜单的子菜单项“Make EXE file”，按回车键可一次完成编译和连接；或者直接按功能键 F9 进行编译和连接，在当前工作目录下生成一个与源文件同名的可执行文件。

2. 对多个文件程序进行编译和连接

如果一个源程序包含多个文件模块，则应当对各个文件分别进行编译，得到多个.OBJ 文件，然后将这些目标文件以及库函数、包含文件等连接成一个可执行文件。

Turbo C 对多个文件进行编译和连接，是将这些文件组成一个“项目”，建立一个项目文件，在该文件中包含各文件的名字，然后将该项目文件交付编译和连接，就可以得到可执行文件。

具体过程如下：

(1) 在 Turbo C 编辑环境中，输入各源文件的名字。

(2) 选择“File/Write to”将文件保存为以.PRJ 扩展名的项目文件。

(3) 按热键 ALT+P 或者通过 F10 移动亮条到“Project”菜单，选择“Project name”项，按回车键，出现项目名对话框，输入需要进行编译和连接的项目文件名，再按回车键确认。

(4) 按 F9 进行编译和连接，生成各个源文件的目标文件以及一个与项目文件同名的可执行文件。

需要注意的是，为了避免出错，在项目文件中最好用带路径的源文件名。另外在处理完一个多文件程序的编译和连接后，应及时将“Project name”项清空，否则，编译连接时将优先对项目文件进行编译，而不是对编辑窗口的源文件进行编译。

3. 编译和连接过程中的显示信息

(1) 系统没有发现程序中有错误。如果屏幕上出现如图 1-13 所示窗口，表示顺利通过编译。从该编译信息窗口可以得到如下信息：

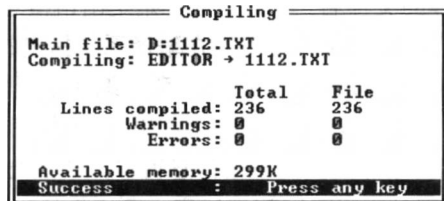


图 1-13

窗口最上方的“Compile”表示是编译阶段的信息。

第 1 行表示主文件名。

第 2 行表示正在编译的文件。

下面几行表示编译的行数、“警告”的次数、“错误”的次数。

倒数第二行表示占用的有效存储空间。

⑥最后一行表示编译成功。用户按任一键以便继续。

如果顺利通过连接，屏幕上就会出现一个连接信息窗口。如图 1-14 所示。

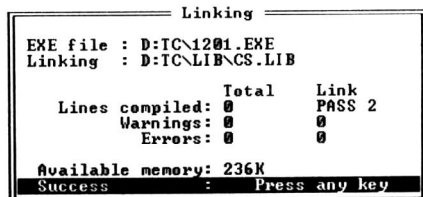


图 1-14

它告诉用户连接后生成的可执行文件名，将目标文件与 C 盘 TC 的 LIB 子目录中的库 CS.LIB 进行连接，在连接过程中未出现警告和错误，则连接成功。按任意键以便继续。

(2) 在编译和连接过程中发现错误。编译信息窗口如图 1-15 所示：

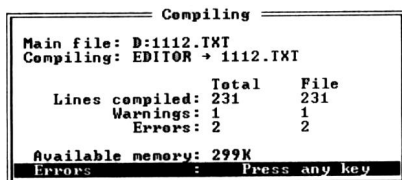


图 1-15

如果是“Warnings”性错误，系统仍然生成目标文件和可执行文件，但程序的运行结果可能出错。如果是“Errors”性错误，编译系统不生成目标文件和可执行文件，必须改正后再重新编译和连接。

按任意键，编译信息窗口消失；在屏幕下部的 Message 窗口中，显示了具体的出错信息。如图 1-16 所示：

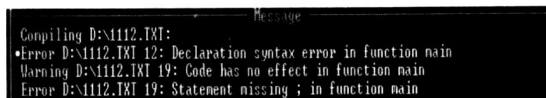


图 1-16

该窗口的第 1 行显示编译的文件名。从第 2 行开始是出错信息，指出出错的行号和可能的错误原因。但要注意，真正有错误的位置可能不在编译系统指出的该行上，一般是在前一行，甚至有可能在前几行。错误原因可以参阅本书第 4 章“编译错误信息”。按 F6 键，显示出错信息，再按一下 F6 键，光标出现在编辑窗口源文件的出错行上，通过上下文分析来改正错误。一个错误修改完后，再按一下 F6 键，回到出错信息窗口，通过光标上下移动选择下一个要修改的错误，同样再按一下 F6 键，光标又回到编辑窗口源文件的出错行继续进行修改。

修改完所有错误后，再进行编译和连接。也可以修改几个错误后进行编译和连接，这是因为有时好像发生了许多错误，而实际上可能是由一个错误所造成的。

4. 停止编译的方式

如果同时提交编译的一个项目文件中的多个文件在编译过程中出现了错误，为了便于修改，允许用户根据需要选择停止编译的方式。

在用户同时提交编译的几个文件中，如果有一个文件出现错误就想停止编译，则按 ALT+P 键到“Project”菜单，选择子菜单项“Break make on”，按回车键后出现一个子窗口，如图 1-17 所示。

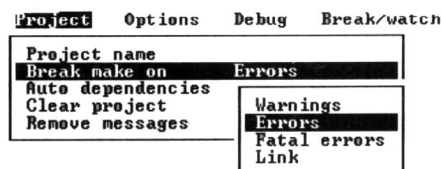


图 1-17

如果选择“Errors”后按回车键，则在编译完一个文件后如发现存在“错误”，则显示有关该文件的出错信息并停止编译，提示用户对该文件进行修改。如果选择“Warnings”，则在编译过程中发现“警告”以上的错误，则显示有关该文件的出错信息并停止编译，提示用户对该文件进行修改。

如果用户希望把所有文件中存在的错误都找出来后，再修改错误，可在“Break make on”子窗口中，选择“Fatal errors”。这样，只有找出所有错误后，才停止编译，用户可统一修改源文件。

“Break make on”子窗口中“Link”的含义是：在进行连接之前停止编译，而不管编译过程中是否遇到错误。

1.4.2 运行

如果在编译、连接过程中未发现错误，系统就会生成目标文件.OBJ 和可执行文件，这时就可以运行了。运行一个程序，有以下几种方法：

1. 在 DOS 环境下运行

如果当前目录就是输出文件.EXE 所在的目录，直接输入可执行文件名，即可运行程序。如果输出文件.EXE 不在当前目录，则要指定路径或切换当前目录。

2. 在 TC 集成环境中运行

按 F10 键选择主菜单“RUN”的子菜单项“RUN”，按回车键，即开始运行。

3. 直接按 CTRL+F9 键运行

CTRL+F9 键的作用包括编译、连接和运行，这是运行 Turbo C 最简便的常用的方法。如果在编译或连接中有错，就将停下来，显示出错信息，等用户修改后再按 CTRL+F9 键重新编译、连接；如不出错，将自动运行。

程序运行后，要查看结果，可按 ALT+X 切换到 DOS 屏幕看程序运行的结果。如果程序中有键盘输入语句，程序运行后自动切换到 DOS 屏幕，用户可从键盘输入数据，输入完成后按回车键，屏幕又从 DOS 屏幕回到 TC 窗口，再通过 ALT+X 切换到 DOS 屏幕查看程序运行的结果。用户在看完运行结果后，想回到编辑窗口继续其他操作，则在 DOS 状态下再次运行 TC，回到 TC 主窗口。

1.5 程序的调试

程序的调试是一项复杂而又细致的工作，它要求程序调试者既善于观察和思考，又要善于总结和积累经验。通过程序调试这一环节，可以进一步提高设计者的程序设计能力。因此，在程序设计的过程中，切不可只注重设计出程序，而不重视程序调试这一重要环节。

进行调试前，要熟悉程序运行环境，在程序设计过程中要为程序调试作好准备。具体来说有以下几点：

(1) 采用模块化、结构化设计方法设计程序。在设计时要按照不同的功能将程序模块化，尽量使每个模块完成单一的功能，并且要将结构化设计的原则贯穿于程序设计的始终。这样设计出来的程序便于调试。

(2) 编写程序时要为调试程序提供足够的灵活性。程序设计虽然是针对实际问题的，但也要充分考虑到调试程序时可能会出现的情况，编写程序时要为调试中临时修改、选择输入数据的形式和个数、改变输出形式等情况提供灵活性。

(3) 根据程序调试的需要，选择并安排适当的中间结果输出和必要的断点。为了判断程序执行过程是否正常，以及观察程序执行的路径与中间结果的变化情况，应该在程序的适当位置输出必要的中间结果，通过这些中间结果可以观察程序的执行情况。

(4) 精心准备调试用例，选用的数据应该具有典型性和代表性。程序在进行编译和连接时，编译系统一般能检查出语法错误，但无法检查出逻辑错误。程序调试的目的主要是检查出程序中的逻辑错误，可以分为静态检查和动态调试。

对程序逻辑结构的静态检查主要包括以下几方面的内容：

- (1) 检查程序中各变量的初始值是否正确。
- (2) 检查分支选择结构或循环结构中的条件是否正确。
- (3) 检查循环结构之间的循环次数是否正确，循环体中是否有改变循环条件的语句。
- (4) 检查各结构之间的嵌套关系是否正确。
- (5) 检查程序中的所有表达式是否正确，表达式中的括号是否配对等。

动态调试是在上机过程中进行调试，常用的动态调试方法有两种：

(1) 按步执行。这种方法的特点是：程序一次只执行一行。每执行完一行后，就停下来，用户可以检查此时各个有关变量和表达式的值，以便发现问题所在。

按 F7 或者通过主菜单“RUN”的子菜单项“Trace into”使程序按步执行。当程序执行到某个包含变量的语句，想要查看一下变量的值是否正确，则按 CTRL+F7 或者选择主菜单“Break/watch”的子菜单项“Add watch”，在编辑窗口出现一个观察数据的输入框。在此框内输入变量名按回车键，输入框消失，在屏幕下部的 Watch 窗口显示出变量的当前值。重复此过程可检查多个变量的值是否正确。

(2) 设置断点。按步执行方法能有效地、一行一行地检查有关变量或表达式的值，但如果程序很长，则难以逐行进行检查。对于一个较长的程序，常用的方法是在程序中设若干断点，程序执行到断点时暂停，用户可以检查此时有关变量或表达式的值。如果未发现错误，就使程序继续执行到下一个断点，如此一段一段地检查。

设置断点的方法是将光标移到某一行上，然后按 Ctrl+F8，此行以颜色条覆盖，作为断点行。如果想取消断点行，则将光标移到断点行上，再按一次 Ctrl+F8 键，颜色条消失，

该行就不再是断点行。运行时遇到断点行暂停，此时，用户可以用单步执行的方法查看有关变量和表达式的值。如果想继续运行，再按一次 **Ctrl+F9** 键即可。调试过程中如果遇到错误，程序将向 DOS 屏幕输出出错信息，用 **ALT+X** 切换屏幕即可查看该信息。

在用按步执行或设置断点方法找错的过程中，还可以使用 TC 的“**Debug**”菜单提供的调试工具。“**Debug**”菜单中的“**Evaluate**”命令不仅可以查看有关变量和表达式的值，还可以修改它们的值，以帮助用户调试程序。

2 上机实验指导

程序设计是一门实践性很强的课程，必须十分重视实践环节。学习 C 语言也一样，不仅要能看懂书上的程序，而且应当熟练地掌握程序设计的全过程，即独立编写出源程序，独立上机调试程序，独立运行程序和分析结果。通过上机实践，加深对所学内容的理解，不断总结经验，做到触类旁通，这对学习其他计算机语言方面的知识也是有帮助的。

C 语言的上机实验一般包括以下几个过程：

- (1) 调出 C 编译系统，进入 C 工作环境。
 - (2) 输入自己编好的源程序。
 - (3) 静态检查输入的程序是否有错。如发现有错，则及时改正。
 - (4) 进行编译和连接。如果在编译和连接过程中发现错误，屏幕上会出现出错信息，参照第 1 章介绍的方法进行改正，直到顺利通过编译和连接。
 - (5) 运行程序，并分析运行结果是否合理和正确。在运行时要注意输入不同的数据来分析结果是否正确。
 - (6) 输出程序清单和运行结果。总结本次调试过程所取得的经验。
- 下面，我们针对《C 语言程序设计教程》的每一章提出了实验要求和实验题目。

2.1 C 程序的运行过程

一、上机要求

1. 熟悉 Turbo C 2.0 的软件环境，掌握使用 TC 进行编辑、编译、连接和运行的方法。
2. 通过运行简单的 C 程序，掌握 C 源程序的特点。

二、上机练习题

1. 运行 Turbo C 2.0 程序，进入 C 程序设计环境

在 Windows 系统的“开始”菜单中选择“运行”菜单项，键入“cmd”命令进入 DOS 状态，在 DOS 提示符“C:\>”下进入 TC 子目录中运行 TC.EXE 程序，即可进入 TC 程序设计环境。

2. 选择工作目录

工作目录即用户文件所在的目录，是程序当前的工作目录。改变当前工作目录可使用菜单命令“File/Change Dir”，然后输入已经存在的目录作为新的当前工作目录。可用“File/Directory”菜单命令显示当前目录下的文件列表。

3. 建立工作环境

建立工作环境的作用是告诉 TC 包含文件、库文件所存放的位置，以及输出文件存于何处。使用菜单命令“Options/Directories”进行设置，按回车键，在弹出窗口内设置新的文件路径，如图 2-1 所示。

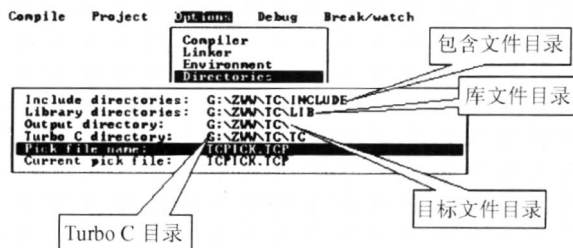


图 2-1

其中，目标文件目录(Output directory)用来存放输出的.OBJ、.EXE 文件，可建立一个新目录来存放输出文件。Pick 文件用来保存用户退出系统时曾经编辑的文件相关信息（文件名、光标位置等），使用 File/Pick ”菜单命令可把最近加载到编辑器的 8 个文件列出来 称为 Pick 表）从而选择这 8 个文件中的任一个进行编辑。此时，Pick 表可以自动恢复上次退出系统的编辑状态，继续程序的开发。在编辑状态下，用 Alt+F6 可以在 Pick 表中的文件之间进行跳转操作，而不必退出系统。如果没有指定 Pick 文件名，可以新建立或使用系统默认的 Pick 文件 Tcpick.tcp。可以有多个 pick 文件，但当前的环境只能使用其中一个。“Pick file name”设置如图 2-2 所示。

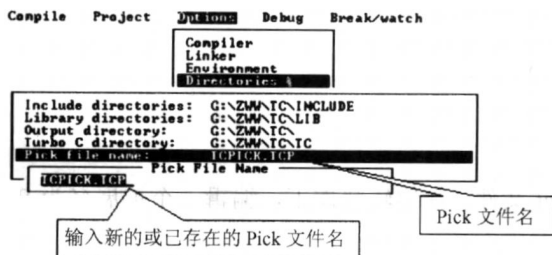


图2-2

4. 保存环境设置

将上面的环境设置保存到一个配置文件中，默认文件名为 Tcconfig.tc。Turbo C 启动时加载已保存的环境设置。

5. 编辑源文件

使用“File”的子菜单项 Load 命令 加载已经存在的 C 源文件。建立新文件，用“File”的子菜单项“New”命令。请输入任一 C 源程序 用 Insert 键在插入和改写状态之间切换 用 Tab 键进行缩进。编辑完毕，用菜单命令“File/Save”或热键 F2 将编辑结果保存到新文件中。

6. 编译、连接

单文件源程序的编译、连接。源程序先经过编译，生成目标文件(.OBJ 文件)然后将目标文件进行连接，生成可执行文件(.EXE)。

方法一：先编译再连接：

Compile to OBJ：将源程序编译成.OBJ 文件。

Link EXE file：将目标文件及库文件连接在一起，生成.EXE 文件。

方法二：一次完成编译和连接：