

TURBO C 语言屏幕绘图

李兰友 王俊省 编
王强如 朱文君

电子工业出版社

(京)新登字 055 号

内容提要

本书是一本 Turbo C 绘图的入门书。全书共分五章,第一章 Turbo C 绘图基础,详细介绍基本操作并讲入绘图世界;第二章介绍基本绘图函数、填充函数的用法,并由程序例进行说明;第三章介绍 Turbo C 绘图程序;第四章介绍二维、三维图形几何变换程序及应用例;第五章介绍 Turbo C 绘图在科学技术中的应用。全书提供了近 100 个实用程序供读者参考。

本书适用于一般科学技术人员,计算机绘图爱好者,亦可作为大专院校师生及各类培训班的教学参考书。

TURBO C 语言屏幕绘图

李兰友 王俊省 编
王强如 朱文君
责任编辑 徐云鹏

*

电子工业出版社出版(北京市万寿路)
电子工业出版社发行 各地新华书店经销
北京顺义李史山胶印厂印刷

*

开本:787×1092 毫米 1/16 印张:6.125 字数:198 千字

1993 年 10 月第 1 版 1993 年 10 月第 1 次印刷

印数:8000 册 定价:5.20 元

ISBN7-5053-2130-X/TP·549

0000112

前 言

Turbo C (2.0 版本) 提供了一个包含有 70 多个图形函数的独立的图形库, 运用 TurboC 的绘图函数, 可画出彩色的线、弧、圆、椭圆、矩形、扇形、二维及三维直方图、多边形以及由这些基本图形组成的工程图形及艺术图案; 可以用 11 种预定义的模式或自定义的模式来填充任何有界的区域。这为绘制丰富多彩的图形和电脑创艺提供了许多便利和手段; 除了绘图函数和填充函数外, 图形库还提供了一些屏幕、视口、图象及图素函数, 颜色设置、控制函数、状态查询函数及错误处理函数等, 使 Turbo C 具有很强的图形功能、深为计算机绘图和 CAD 技术人员所欢迎。

本书是一本 Turbo C 绘图的入门书。全书共五章。第一章 Turbo C 绘图基础, 介绍基本操作并进入绘图世界; 第二章介绍基本绘图函数, 填充函数的用法, 由程序例进行说明; 第三章介绍 Turbo C 绘图程序, 第四章为二维、三维图形几何变换程序及应用例, 第五章介绍 Turbo C 绘图在科学技术中的应用。本书提供近 100 个实用程序, 全部程序存放在一张软盘上, 可供读者使用。适用机种为 IBM PC/XT、PC/286 或兼容机, 用高档微机也可运行本书程序。

参加本书编写的还有许溪根同学及天津理工学院刘印华老师。由于编者水平所限, 不足之处敬希读者教正。

编 者

1991.8.

目 录

第一章 Turbo C 绘图基础	(1)
§ 1.1 Turbo C 系统文件和建立	(1)
§ 1.2 Turbo C 主屏幕	(3)
§ 1.3 菜单命令及用法	(5)
§ 1.4 Turbo C 图形库函数简介	(10)
§ 1.5 Turbo C 绘图源程序的输入	(13)
§ 1.6 使用 CGA 卡绘图	(15)
第二章 图形函数及应用程序例	(19)
§ 2.1 绘图函数	(19)
§ 2.2 填充图形函数	(25)
§ 2.3 视口、屏幕图形存取和文字	(30)
第三章 Turbo C 绘图程序	(34)
§ 3.1 图形程序设计	(34)
§ 3.2 直线图案	(40)
§ 3.3 曲线图案	(42)
§ 3.4 图案生成小型程序包	(47)
第四章 图形几何变换程序及应用	(49)
§ 4.1 二维图形几何变换	(49)
§ 4.2 二维图案特殊变换	(54)
§ 4.3 三维图形变换程序	(59)
§ 4.4 美术图集设计综合程序	(62)
第五章 Turbo C 绘图在科学计算中应用	(70)
附录 Turbo C 库函数分类	(85)

参 考 文 献

- [1] 长江贞彦, “C 语言による パソコンCAD・CG”, 工业调查会, 1986.
- [2] 赤松義幸, “パソコン图形处理テクニック”, 诚文堂新光社, 1985.
- [3] 王俊省等, “Turbo C 程序设计 400 例”, 电子工业出版社, 1991.
- [4] 梅 邨等, “C 语言程序设计及应用” 电子工业出版社, 1988.
- [5] 平田邦男, “Turbo C による科学计算入门” 共立出版株式会社, 1989. 3.
- [6] 潘金贵, 沈默君, “Turbo C 程序设计技术”, 南京大学出版社, 1989

第一章 Turbo C 绘图基础

§ 1.1 Turbo C 系统文件和建立

一、Turbo C 系统文件

Turbo C 2.0 版本的系统文件分别放在六张配给盘上:

1# 盘: 装配及帮助文件

2# 盘: 集成开发环境

3# 盘: 命令行及实用程序

4# 盘: 系统库文件

5# 盘: 嵌入文件和库

6# 盘: 例子、图形驱动程序和字体文件等

各配给盘文件名及内容如表 1.1 所示。

表 1—1 Turbo C 软盘文件配置

盘 号	文 件 名	内 容
1# 装配及帮 助文件	INSTALL .EXE	Turbo C 装配程序
	README .COM	README.COM 文件
	TCHelp .TCH	Turbo C 帮助文件
	THELP .COM	THELP.COM 文本文件
	README	README.COM 应用正文
2# 集成开 发环境	TC .EXE	Turbo C 编译程序集成环境版本
	TCCONFIG .EXE	转换配置文件的程序
	MAKE .EXE	工程管理文件
	GREP .EXE	Turbo GREP 文件
	TOUCH .COM	修改文件日期、时间的程序
3# 命令行及 实用程序	TCC .EXE	Turbo C 命令行版本编译程序
	CPP .EXE	Turbo C 预处理程序
	TCINST .EXE	TC.EXE 安装程序
	TLINK .EXE	Turbo C 连接程序
	HELPME! .DOC	询问帮助文件
4# 系统库 文件	COS .OBJ	小模式启动代码
	COT .OBJ	极小模式启动代码
	COL .OBJ	大模式启动代码
	MATHS .LIB	小模式数学库
	MATHL .LIB	大模式数学库
	CS .LIB	小模式运行时刻库
	CL .LIB	大规模运行时刻库
	EMU .LIB	8087 仿真程序库
	GRAPHICS .LIB	图形库
	FP87 .LIB	8087 库
5# 嵌入文 件库	TLIB .EXE	库管理程序
	ALLOC .H	说明存储管理函数(allocation 和 deallocation 等)
	ASSERT .H	定义 ASSERT 调试宏
	BIOS .H	说明调用 IBM-PC ROM BIOS 例行程序所用的各种函数
	CONIO .H	说明调用 DOS 控制台 I/O 例行程序所用的各种函数
	CTYPE .H	包含字符分类和字符转换宏(如 isnlphat 和 toascii)使用的信息
	DIR .H	包含为使用目录和路径名进行工作的结构、宏和函数
	DOS .H	定义各种常量,并给出为满足 MSDOS 和特殊的 8086 及调用需要的说明
	ERRNO .H	为出错代码定义常量助记码
	FCNTL .H	(定义同库子程序)open 连接所需要的符号常量
	FLOAT .H	包含浮点例行程序的参数
	I/O .H	包含了低级 I/O 例行程序的结构和说明
	LIMITS .H	包含环境参数及编译时刻的限制信息和整型量的上下界

盘 号	文 件 名	内 容
	MATH .H	为数学函数说明原型,还定义了宏 HUGB-VAL 说明 matherr 和 -matherr 函数使用的例外结构
	MEM .H	说明存储操作函数(这些函数中有许多在 string.h 中定义)
	PROCESS .H	包含为了 spawn...和 exec...函数的结构和说明
	SETJMP .H	定义由 longjmp 和 setjmp 函数使用的一个类型 jmp-bur,并且说明函数 longjmp 和 setjmp
	SHARE .H	定义用于使用共享文件的函数的参数
	SIGNAL .H	定义常量 SIG-IGN 和 sig-dfl,并且说明 saignal 和 gsignal 函数
	STDARG .H	定义读写说明为可按受可变参数的参数表所使用的宏
	STDDEF .H	定义几种公用的数据类型和宏
	STDIO .H	定义标准的 I/O 程序包所需的类型和宏,这个程序包在 K&R 中有定义,并且在 UNIX 系统 V 中得到扩充,定义标准的 I/O 的预定义流 stdin,stdout 和 stderr. 并且说明流级 (stream-level) I/O 子程序。
	STDLIB .H	说明几个公用的例行程序,转换子程序,查找/排序子程序和其他的杂函数
	STRING .H	说明一些串操作和存储操作的库函数
	GRAPHICS .H	为图形函数说明原型
	TIMB .H	定义被时间转换函数 asctime、函数 localtime 和 gmtime 填好的一个结构,定义由函数 ctime、difftime、gmtime、localtime、和 stime 使用的一个类型,还提供了这些函数的原型。
	VALUES .H	定义重要的常量,包括依赖于机器值,提供了与 UNIX 系统 V 的兼容性。
	COC .OBJ	紧凑模式启动代码
	COM .OBJ	中模式启动代码
	MATHC .LIB	紧凑模式数学库
	MATHM .LIB	中模式数学库
	CC .LIB	紧凑式运行时刻库
	CM .LIB	中模式运行时刻库
6# 例子, 图形驱 动程序 和字体 文件等	UNPACK .COM	组合文件予处理程序
	OBJXREF .COM	生成交叉目标文件
	COH .OBJ	特大模式启动代码
	MATHH .LIB	特大模式数学库
	CH .LIB	特大模式运行时刻库
	GETOPT .C	命令行中语法分析程序选择项
	HELLO .C	TURBO C 程序例
	MATHERR .C	处理数学库例外的源代码
	SSIGNAL .C	函数 SSIGNAL 和 GSIGNAL 的源代码
	CINSTXER .EXE	1.5 到 2.0 版升级程序
	TNIT .OBJ	与 TURBO C prolog 连接的初始化模块
	BGI .ARC	启动源代码及相关文件的组合文件
	EXAMPLES .ARC	含多个 C 程序例的组合程序
	MCALC .ARC	Micro CALL 的源程序说明体的组合文件

二、Turbo C 的安装

在计算机系统上建立 Turbo C 时,需将有关文件由配给盘复制到工作盘和硬盘上。

将 Turbo C 建立在硬盘上用户用起来是很方便的,在硬盘上建立 Turbo C 时,使用 1# 盘提供的安装程序 (INSTALL) 是很方便的。具体操作步骤如下:

1. 将系统置于 DOS 状态下;屏幕提示符为:

C>—

2. 将标签为 INSTALATIN DISK 的源盘插入 A 驱动器,键入 A:,回车,屏幕提示符为:

A>—

3. 键入 INSTALL,回车,屏幕显示 Turbo C 版本号等,按回车键。屏幕显示提示:

“Enter the SOURCE drive use:”

按 A 键,回车。

4. 屏幕提示安装功能菜单,选择 “Install Turbo C on Hard Drive”,回车。

5. 屏幕显示 Turbo C 安装子目录。这时按 F9 键, 开始安装。

6. 根据装配程序提示, 顺序将配给盘逐一更换。

安装过程中可按 F1 键寻求联机帮助, 它将提示您应如何做。安装完毕可按 ESC 键。

如果硬盘上已安装了 Turbo C 1.5 版本, 可以用 INSTALL 装配程序将其升级为 2.0。

安装完毕后, 可检查一下是否已建立完毕, 操作为:

1. 在 DOS 状态下, 在 C>提示符后键入:

C>CD\TC ✓

2. 屏幕仍显示 C>提示符, 键入:

C>TC✓

3. 屏幕显示版本信息, 则表示系统已建好, 可以使用。

§ 1.2 Turbo C 主屏幕

在 DOS 提示符下键入 tc, 按回车键, 即可将 Turbo C 装入内存。这时, 屏幕显示版本信息(图 1.1)

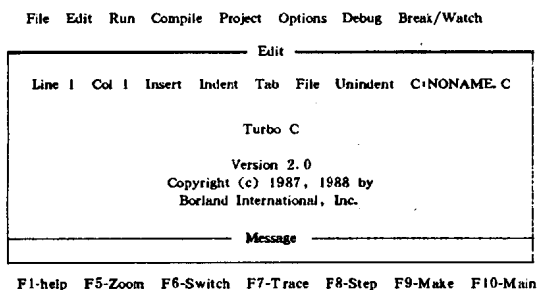


图 1.1

按任意键, 版本信息消失, 呈现 Turbo C 主屏幕, 光标在屏幕左上角 File 处。

主屏幕由主菜单行, 编辑窗、信息窗及快速参考行等四部分组成。

一、主菜单

主屏幕顶端一行为主菜单。主菜单有 8 个选项:

File Edit Run Compile Project Options Debug Break/watch	
File	处理文件(装入、存盘、选择、建立、换名和写盘。目录操作(列表、改变工作目录)及退出程序调用 DOS。
Edit	建立编辑源程序。
Run	运行程序。

Compile	编译, 生成目标文件及可执行文件。
Project	允许说明程序里包含那些文件的管理条目。
Options	允许用户选择编辑器选择项(如存贮模型、编译任选项、诊断及连接任选项等)和定义宏指令。
Debug	调试程序。检查改变变量值, 查找函数, 程序运行时查看调用栈。
Break/Watch	增加、删除、编辑表达式及设置、清除、执行至断点。

在主菜单中有 Edit 是单项菜单命令, 使系统进入编辑状态, 其它条目被选择时, 均有下拉菜单或选择项。

在系统内, 按 F10 键, 则返回主菜单。

二、编辑窗

在主菜单内选择 Edit, 按回车键, 或在系统内按 Alt-E 键, 则系统进入编辑状态, 光标在窗口左上角, 窗口顶部双线且 Edit 高亮度, 表明编辑窗口是活动窗口, 可以建立和编辑源文件。

编辑窗口顶端状态行给出了有关正在被编辑文件的信息, 光标在文件中位于何处等。

Line Col Insert Indent Tab Fill	
Unindent C: NONAME. C	
各项含意为:	
Line n	光标位于文件第几行
Col n	光标位于文件第几列
Insert	插入模式开关
Indent	自动缩进开关
Tab	制表开关开启
Fill	Tab 模式为 on 时, 编辑器将用制表及空格符优化每一行的开始
Unindent	当光标在一行中的第一个非空字符上时, 或在非空行上时, 退格键回退一级
C:NONAME. C	当前驱动器名(C:), 文件名(此时为 NONAME)和扩展名(. C)

三、信息窗

信息窗在编辑窗下部, 在编译和调试源程序时, 用于查看诊断信息。源文件被编译编辑时所有的警告和错误都列在信息窗内, 同时在编辑窗口中指出错误在源文件中的位置。

四、功能键提示行

对任一菜单和窗口, 屏幕底部有一行功能键提示行, 提供在当前状态下功能键的提示。主菜单下功

能键提示行为:

F1-Help F5-zoom F6-Switch F7-Trace F8-Step
F9-Make F10-Menu

按 Alt 键保持几秒钟, 提示行将描述 Alt 键与那些键联用时执行什么功能

Alt: F1-Last Help F3-Pick F6-Swap F1/F8-Prev/Next Error F9-Compile

在编辑状态下功能键提示行为:

F1-Help 打开一个帮助窗口, 提供有关编辑命令的信息
F5-Zoom 扩大编辑窗至整个屏幕
F7-Trace 在调试模式下一次执行一程序, 跟踪到函数内部
F8-Step 在调试模式下一次执行一程序, 但不跟踪到函数内部
F9-Make 编译连接 EXE 文件
F10-Main 从编辑窗口返回主菜单

信息窗口时功能键提示行功能为:

F1-Help 打开帮助窗, 概述 tc 错误跟踪的特点
F5-Zoom 将信息窗扩大到整个屏幕
F6-Swith 激活编辑窗口
F7-Step 在源文件调试模式下允许一次执行一行, 跟踪到函数内部
F8-Make 生成 Make.EXE 文件
F10-Main 返回主菜单。

为了更快地熟悉 Turbo C 集成开发环境, 这里先介绍一些操作键的功能, 以便能更快掌握 Turbo C 的操作。

1. 适用于整个 TC 系统的操作键是

(1) 按 F1 键可获得有关当前位置的帮助信息。按 F1 键后即调用了 Turbo C 帮助系统, 帮助窗口解释当前条目的功能。按 ESC 键即可退出帮助。

(2) 按 F5 键、可放大活动窗口或缩小活动窗口。

(3) 按 F6 键进行开关活动窗口(循环切换编辑窗与信息窗)。

(4) 按 F10 键, 活动窗口和菜单互相切换。

(5) 按 Alt-F6 键, 可改变窗口内容。

(6) 按 Alt 键和主菜单命令首字符(F, E, R, C, P, O, B)调用主菜单命令。

例如, 在系统的任何位置按 Alt-E 即进入编辑状态。

2. 在各个菜单内的操作键为:

(1) 按 ESC 键退出菜单。

(2) 按 F10 键从任意菜单级回到以前的活动窗

口。

(3) 用左右箭头键可选择主菜单项。

(4) 用上、下箭头键进行菜单项选择。

在介绍各菜单之前, 先熟悉一下为执行菜单某一固定功能而设的键是非常有益的, 这些键称为快捷键, 不论在集成环境的任何位置, 只要一按这些快捷键, 则可调用相应的功能。Turbo C 快捷键如表 1-2 所示。

表 1-2 Turbo C 快捷键

键	功 能
F1	打开帮助窗口
F2	编辑器中文件存盘
F3	装入文件至内存
F4	运行程序到光标所在行
F5	放大、缩小活动窗口
F6	开关活动窗口
F7	在调试模式下运行程序, 跟踪到函数内
F8	在调试模式下运行程序。跳过函数执行“Make”
Ctrl-F1	调用有关函数上下文帮助
Ctrl-F2	重新运行程序
Ctrl-F3	显示调用栈
Ctrl-F4	计算表达式
Ctrl-F7	增加监视表达式
Ctrl-F8	断点开关
Ctrl-F9	运行程序
Alt-F1	显示上次访问的帮助
Alt-F3	选择装入文件
Alt-F6	开关活动窗口里的内容
Alt-F7	定位上一错误
Alt-F8	定位下一错误
Alt-F9	将编辑器内文件译成 .OBJ 文件
Alt-B	转到 Break/watch 菜单
Alt-C	转到 Compile 菜单
Alt-D	转到 Debug 菜单
Alt-F	转到 File 菜单
Alt-O	转到 Options 菜单
Alt-P	转到 Project 菜单
Alt-R	转到 RUN 菜单
Alt-X	返回 DOS

五、调试器和监视窗

Turbo C 集成开发环境, 内含有一个帮助查找程序中错误的集成调试器。调试器能够使程序在任何地方停止运行, 以便用户检查、修改变量的值。

当集成开发环境的调试器运行时, 监视窗口代替信息窗口, 屏幕由顶部的编辑窗口和下部的监视窗口组成, 按 F6 键可切换这两个窗口。

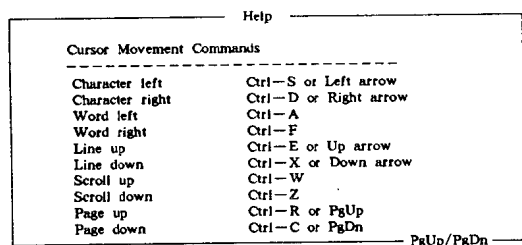
监视窗口含监视表达式及每个表达式的当前值, 因为其值可能发生变化, 监视窗口给用户提供了跟踪程序运行时一些重要表达式值的手段。

用户可以用编辑窗口内所用的编辑命令来编辑监视窗口中的表达式。

六、帮助

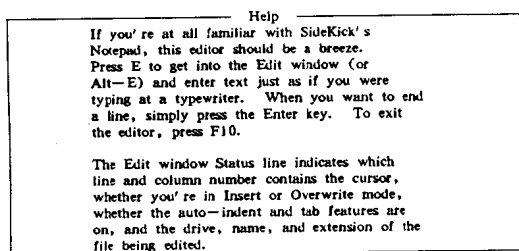
Turbo C 提供三种形式的联机帮助：一是各种状态下屏幕底行的功能键提示行，帮助用户选择在该状态下可以使用的功能键帮助。

第二种联机帮助是按 F1 键，可显示总帮助



按 PgDn 可得下一页主帮助。再按 F1 键，屏幕显示帮助索引，用箭头键选择，按回车键，则显示该目录的详细说明。

按 Alt-F1 键，可获得编辑状态下有关函数的帮助说明。例如，在编辑窗口按 Alt-F1，则屏幕显示：



按 PgDn 可获得其他帮助。

§ 1.3 菜单命令及用法

一、文件菜单

在主菜单下选择 File、按回车键，或按 Alt-F 键，屏幕显示文件下拉菜单（图 1.2）

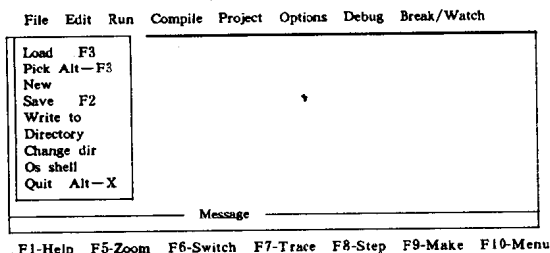
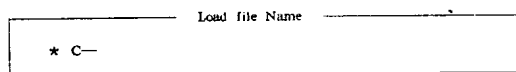


图 1.2

文件菜单各选择项应用如下：

1. Load

装入一个文件到内存。选择时屏幕显示提示框：



按空格键，彩框消失。键入盘名、文件名及扩展名(.C)，按回车键，装入后回到主屏幕，显示装入文件名及内容。

2. Pick

从先前装入编辑窗口的各个文件中选择一个并装入。

3. New

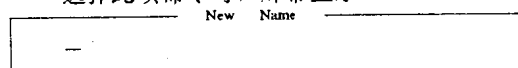
说明文件是一个新文件，用户已进入编辑窗编辑新文件，新文件名默认为 NONAME.C。

4. Save

将编辑窗内文件存盘，如果文件名为 NON-AME.C 则询问是否要改名。这时写入新文件名按回车键即可；如果仍以原文件名存盘，可按回车键。

5. Write to

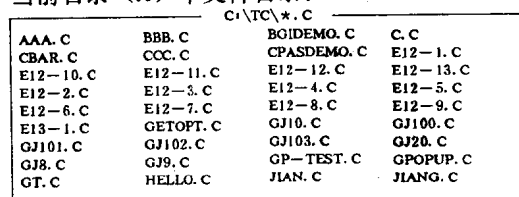
选择此项命令时，屏幕显示



此时将一个文件写到新文件名下，或复盖一个已存在的文件。

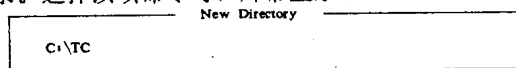
6. Dircroxy

显示用户需要的目录和文件集。例如，显示 C 盘当前目录 (tc) 下文件目录：



7. Change dir

显示当前目录，并允许修改指定的驱动器和目录。选择该项命令时，屏幕显示。



8. Os shell

暂时退出 Turbo C 到 DOS，按 Exit 可返回 Turbo C。

9. Quit

退出 Turbo C 返 DOS。相应快捷键为 Alt-x。

二、编辑

在主菜单中按 E 或 ESC，即可到编辑状态，编辑命令调用内部编辑器，对源文件进行编辑、或建立新的源文件。

用户常用编辑命令如表 1—3。

表 1—3 编辑程序命令

类 别	功 能	默 认 键
基 本 光 标 移 动 命 令	字符左 字符右 字左 字右 上行 下行 上滚 下滚 上一页 下一页	Ctrl-S 或 Left Ctrl-D 或 Right Ctrl-A Ctrl-F Ctrl-B 或 Up Ctrl-X 或 Down Ctrl-W Ctrl-Z Ctrl-R 或 PgUp Ctrl-C 或 PgDn
快 速 光 标 移 动 命 令	行头 行尾 窗口头 窗口底 文件头 文件尾 块头 块尾 上次光标位置	Ctrl-QS 或 Home Ctrl-QD 或 End Ctrl-QE Ctrl-QX Ctrl-QR Ctrl-QC Ctrl-QB Ctrl-QK Ctrl-QP
插 入 与 删 除 命 令	插入模式 插入行 删除行 删除到行尾 删除光标左边的字符 删除光标处的字符 删除光标右边的字符	Ctrl-V 或 Ins Ctrl-N Ctrl-Y Ctrl-QY Ctrl-H 或 backspace Ctrl-G 或 Del Ctrl-T
块 命 令	标记块开始 标记块结束 标记单个字 复制块 删除块 隐藏/显示块 移动块 从盘中读块 将块写回磁盘	Ctrl-KB Ctrl-KK Ctrl-KT Ctrl-KC Ctrl-KY Ctrl-KH Ctrl-KV Ctrl-KR Ctrl-KW
其 他	异常结束操作 制表模式 制表 自动缩进和最佳填充 定界符配对 控制字符前缀 查找 查找替换 查找标记 调用主菜单 装入文件 退出编辑、不保存文件 重复上次查找 恢复行 保存并编辑 置标记	Ctrl-U Ctrl-OT Ctrl-L 或 Tab Ctrl-OI 或 Ctrl-QI Ctrl-Q, Ctrl- [或 Ctrl-Q, Ctrl] Ctrl-P Ctrl-QF Ctrl-QA Ctrl-QN F10 F3 Ctrl-KD 或 Ctrl-KQ Ctrl-I Ctrl-QL Ctrl-KS 或 F2 Ctrl-KN

光标操作命令如表 1—4 所示。

表 1—4 光标操作命令键

	命 令 键	光 标 操 作
光 标 操 作	Ctrl-A	移到光标左边字的第一个字符
	Ctrl-S	光标左移一格
	Ctrl-D	光标右移一格
	Ctrl-F	移到光标右边字的第一个字符
	Ctrl-E	移上一行
	Ctrl-R	移上一屏
	Ctrl-X	移下一行
	Ctrl-C	移下一屏
	Ctrl-W	屏幕下滚一行，光标不动
	Ctrl-Z	屏幕上滚一行，光标不动
	PgUp	屏幕及光标上滚一屏
	PgDn	屏幕及光标下滚一屏
快 速 光 标 操 作	Ctrl-QS 或 Home	移到当前行的第一列
	Ctrl-QD 或 End	移到当前行的尾端
	Ctrl-QE	移到屏幕的顶部
	Ctrl-QX	移到屏幕的底部
	Ctrl-QR	移到文件第一个字符
	Ctrl-QC	移到文件最后一个字符
		(前缀 Ctrl-Q 加上字母 B、K、P 可以使光标跳到正文中的一些特殊点。)
	Ctrl-QB	将光标移到由 Ctrl-KB 设置的块开始标记
	Ctrl-QK	将光标移到由 Ctrl-KK 设置的块结束标记
		这两个命令即便在该块未显示或块结束/开始标志未设置时也有效(参见“隐藏/显示块”)
	Ctrl-QP	移到上一命令前光标的最后位置，在查找或查找/替代操作执行后，希望光标回到执行前的最后位置时，这个命令特别有用

三、运行菜单

按 Alt-R，屏幕显示运行菜单（图 1.3）

运行菜单命令运行程序，开始和结束调试。它有 6 个选择项。

Run 运行程序

Program reset 程序重新启动

Go to cursor 从执行长条运行到编辑窗口光标所在行

Trace into 跟踪进入到函数

Step over 单步执行，一次一条语句

User screen 显示转入用户屏幕，按任意键返回

时屏幕显示示例为：

Compiling		
Main file: E12-6.C		
Compiling: EDITOR E12-6.C		
	Total	File
Lines compiled:	393	393
Warnings:	0	0
Errors:	0	0
Available memory	265K	
Success	Press any key	

此时系统对源文件进行编译，编译完毕后屏幕显示

File	Edit	Run	Compile	Project	Options	Debug	Break/Watch
		Run	Ctrl—F9				
		Program reset	Ctrl—F7				
		Go to cursor	F4				
		Trace into	F7				
		Step over	F8				
		User screen	Ctrl—F5				
Message							

ALT: F1-Last help F3-Pick F6-Swap F7/F8-Prev/Next error F9-Compile

图 1. 3

在选择 Run 或直接按 Ctrl-F9 键，程序则运行，运行

Linking		
EXE file: E12-6.EXE		
Linkings: LIB\CS.LIB		
	Total	File
Lines compiled:	0	PASS 2
Warnings:	0	0
Errors:	0	0
Available memory	202K	
Success	Press any key	

系统对编译完的文件（.OBJ 的文件）进行连接，如

无错误, 则文件 (.C, .OBJ 和 .EXE) 由系统自动存盘, 并在屏幕上显示运行结果。

四、编译菜单

编译菜单选择项可将源文件译成 .OBJ 文件, 生成 .EXE 文件, 重建、设置、运行或显示上次编译的信息。

编译菜单如下 (图 1.4)

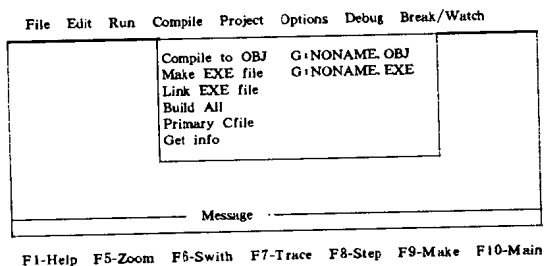


图 1.4

各选择项功能简介如下:

Compile to OBJ 将源文件 (.C 文件) 译成目标文件 (.OBJ 文件), 显示生成文件名。编译时, 弹出一个窗口, 显示编译结果。编译/组装 (Make) 完毕之后, 按任意键, 编译窗口消失。如出错, 则转到信息窗第一条错误上。

Make EXE File 生成可执行文件。

Link EXE File 连接执行文件。将当前 .OBJ 文件与库文件连接在一起, 生成 .EXE 文件。

Build All 建立所有文件。

Primary c File 主 C 文件。

Get Info 获得信息。开辟一个显示窗口显示源文件、与当前文件有关的源文件名及当前源文件名等内容。

五、Project 菜单

本菜单命令可将多个源文件及目标文件合起来。按 Alt-P 键, 屏幕显示 (图 1.5) 为

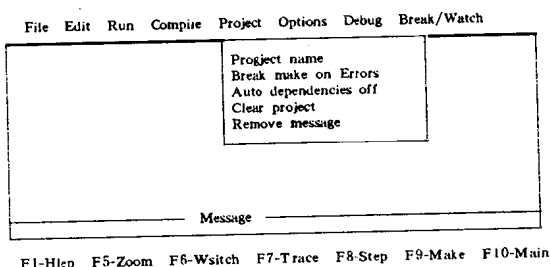


图 1.5

各菜单选择项功能如下:

Project name 选择一个包含将要编译、连接的文件名的目标文件, 项目名也将是以后建立的 .EXE 及 Map 文件名。

Break mak on 允许用户说明在编译一具有

警告、出错、致命错的文件名, 或在链接之前停止。

Clear project 清除项目名重建信息窗。

六、选择项菜单 (options)

选择项菜单包括决定集成化环境如何工作的设置。选择项菜单如下 (图 1.6)

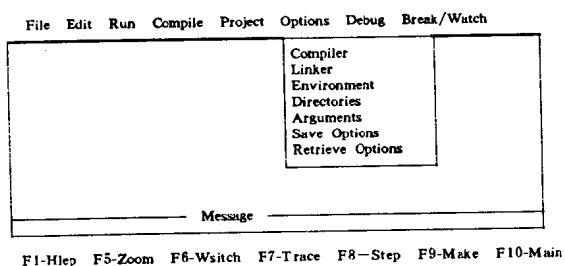


图 1.6

选择项菜单功能概略说明如下:

1. Compiler 编译器

Compile 被选择后, 将弹出一下拉菜单 (图 1.7)

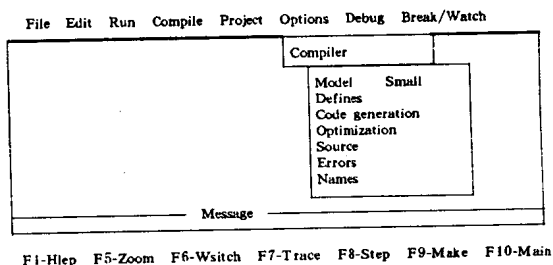


图 1.7

可进行存贮模式 (Model), 宏定义, 代码生成、优化, 源代码处理、出错处理和命名等功能。

其中, Model 选择项用来设置 Turbo C 的各种存贮模式。某项被选择时, 则又弹出一下拉子菜单, 显示极小模式 (Tiny), 小模式 (Small), 紧凑模式 (Compact)、中模式 (Medium)、大模式 (Large) 和特大模式 (Huge) 等选择项。

2. Linker 链接器

Linker 被选择时屏幕出现一下拉菜单 (图 1.8)

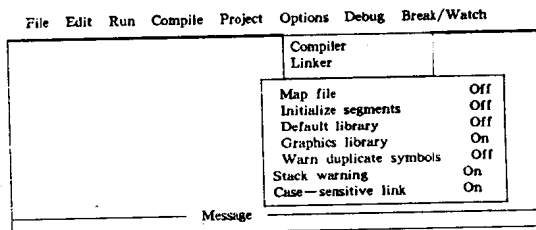
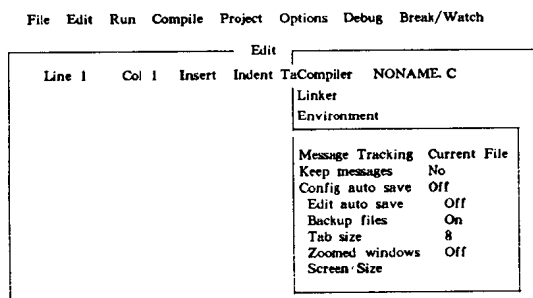


图 1.8

本下拉菜单处理有关连接程序的选择项。

3. Environment

条目让用户自备份在编辑器里的源文件，裁减 Turbo C 工作环境以适应程序的需要。选用该选择项时，屏幕显示下拉菜单（图 1.9）



ALT: F1-Help F3-Zoom F6-Switch F7-Trace F8-Step F9-Make F10-Menu

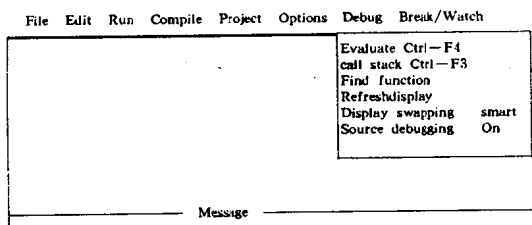
图 1.9

Environment 的菜单项实现消息跟踪、保存消息、备份文件、放大窗口等功能。

Options 菜单的其它选择项命令功能请参阅有关手册。

七、调试菜单

Debug 菜单控制集成调试器除断点和监视表达式以外的条目。按 Alt-D，屏幕显示 Debug 下拉菜单（图 1.10）



F1-Help F3-Zoom F6-Switch F7-Trace F8-Step F9-Make F10-Main

图 1.10

Evaluate 计算变量或表达式并显示其值；

Call stack 显示一包调用栈的弹出窗口。
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。

Find function 显示编辑窗口中某一函数定义；

Source Debugging 完成源代码调试：开关为 on 时，连接的程序可用集成调试器和单独的 Turbo C 调试器调试；开关设为 standalone 时，只能用 Turbo C 调试器；若设为 None 时，两种调试器均不能使用。

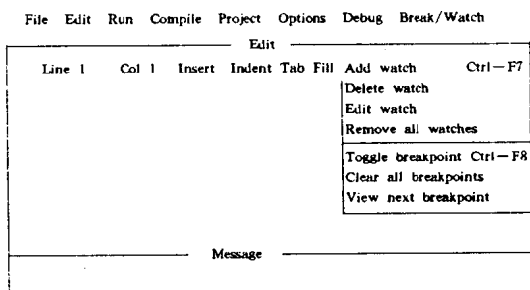
Display Swapping 显示转换。

Refresh Display 刷新显示器。

八、Break/watch 菜单

Break/watch 菜单命令控制断点及监视表达式。

按 Alt-B 屏幕显示下拉菜单（图 1.11）



F1-Help F3-Zoom F6-Switch F7-Trace F8-Step F9-Make F10-Menu

图 1.11

菜单选择项：

Add Watch 增加监视表达式

Delete watch 删除监视表达式

Edit Watch 编辑监视表达式

Remove All watches 删除所在监视表达式

Toggle Break Point 打开或关闭断点

Clear All Break Watches 清除所有断点

View Next Break Point 显示下一个断点

九、调试程序

程序一般不是一次运行成功。一个新程序几乎总是包含一些错误，发现错误并修改错误使之得到正确结果的过程即为程序调试。

Turbo C 集成环境内含有一集成调试器，它允许您控制程序运行，可在任意点上终止程序运行，一次执行一条语句，还可以在运行中观察数据的变化。这里仅介绍最基本的程序调试操作。

1. 设置和使用断点

调试程序时，可以在程序中的一行或几行上设断点，当程序执行到断点处时，会停下来暂不执行有断点那一行的第一条语句，并将调试控制返回给用户。这时，用户可以做显示表达式的值，或改变变量的值，或逐行运行程序等工作。

断点设置的操作为：

(1) 在编辑窗口将光标移到欲设置断点的那一行上；

(2) 按 Ctrl-F8 键，或选用 Break/Watch 菜单的 Toggle Breakpoint 命令，则设置断点的一行置高亮度。

(3) 在程序运行中一旦遇到断点就会暂时停止，且返回编辑状态。在编辑源文件时，断点始终附着原来设置的地方。

(4) 清除断点。清除断点仍用 Break/Watch 菜单的 Toggle Breakpoint 命令。

2. 单步执行程序

单步执行程序可按 F7 键或 F8 键。每按一次执行一行程序。一般通常使用 F8 键进行单步调试。每执行一行停下来查看执行结果。操作如下：

(1) 按 F8 键，屏幕显示编译窗，编译后链接，最后执行长条显示在程序第一行。

(2) 按 F8 键，每按一次执行一行。

(3) 按 Alt-F5 键观察程序结果，按任意键返回编辑屏幕。

3. 计算表达式

按 Alt-D 键并选择 Evaluate 命令，或按 Ctrl-F4 则屏幕显示测试器窗口：

Evaluate
Result
New Value

窗口内含三个段：计算段 (Evaluate)，结果段 (Result) 和新值段 (New Value)

调试时使用该窗口的操作：

(1) 按 F8 键，使执行长条移到程序第一行。

(2) 将光标用光标移动键移到欲观察变量名处。

(3) 按 F8 键单步执行到该行时，按 F4 键，显示测试窗，计算段内显示光标处变量名，按回车键，则在结果段显示该变量的当前值。

(4) 如在新值段给变量赋新值，重复上述步骤，则在结果段显示赋新值后变量的当前值。

注意，此时屏幕源程序并未改变。

程序调试时常使用快捷键如表 1—5 所示：

表 1—5 程序调试常用命令键

键	功 能
F4	程序运行到光标所在行
F7	运行当前函数中的下一条语句
F8	运行当前函数中的下一条语句但不跟踪进函数
F5	放大缩小活动窗口
F6	循环激活编辑、信息、监视窗口。
Alt-F5	转入用户屏，按任意键返回
Alt-F6	若编辑窗是活动窗，则转上次装入编辑器的文件，若下部窗是活动窗，则进行信息窗、监视窗的切换。
Ctrl-F9	运行程序
Ctrl-F8	设置断点或删除断点
Ctrl-F7	增加一监视表达式
Ctrl-F4	计算 C 表达式
Ctrl-F2	结束当前调试节

§ 1.4 Turbo C 图形库函数简介

Turbo C 提供了一个包含有 70 多个图形函数的独立的图形库，其中包含有 7 类函数：

1. 图形系统控制
2. 绘图及填充
3. 屏幕及视口管理
4. 正文输出
5. 颜色控制
6. 错误处理
7. 状态查询

一、图形系统控制

图形控制函数如表 1—6 所示。

表 1—6 图形系统控制函数

函数名	功 能
closegraph	关闭图形系统
detectgraph	检查硬件以决定使用何种图形驱动程序，并推荐一种模式
graphdefaults	将所有的图形系统变量重新设置为它们的缺省值
graphfreemem	释放图形内存；定义自己的子程序
graphgetmem	分配图形内存；定义自己的子程序
getgraphmode	返回当前的图形模式
getmoderange	返回特定驱动程序的最高及最低的有有效模式
initgraph	初始化图形系统，并将硬件置于图形模式
installuserdriver	将一个用户添加的设备驱动程序加到 BGI 设备驱动程序表中
installuserfont	装入一个图形程序所不知的笔划字体文件
registerbgidriver	登记一个被接过的或用户加载的驱动程序，用于链接
restorecrtmode	恢复初始的（先于 initgraph 的）屏幕模式
setgraphbufsize	设定一个内部图形缓冲的大小
setgraphmode	选择图形模式，清屏，并恢复所有的缺省值

图形系统控制函数 initgraph 用于初始化图形系统，将硬件置于图形模式；detectgraph 来选择一个驱动程序及模式；getdrivename 找出驱动程序的名字，用 getmodename 找出模式名；用 Setgraphmode 改变图形模式；用 closegraph 关闭图形系统等。

二、绘图与填充

Turbo C 2.0 版本提供了 25 种绘图填充函数，如表 1—7 所示

表 1—7 绘图与填充

函数名	功 能
绘图	
arc	画一个圆弧
circle	画一个圆
drawpoly	画一个多边形
ellipse	画一个椭圆弧
getarccords	返回上次调用 arc 或 ellipse 后的坐标
getaspectratio	返回当前图形模式纵横比
getlinesettings	返回当前的画线格式、模式及宽度
line	从 (x0, y0) 到 (x1, y1) 画线
linereel	从当前位置 (CP) 到与 CP 有一定距离的某点画线
moveto	将当前位置 (CP) 移到 (X, Y)
moverel	将当前位置 (CP) 移动一段距离
rectangle	画矩形
setaspectratio	改变缺省的纵横比率校正系数
setlinestyle	设置当前行宽及格式
填充	
bar	画条形图并填充
bar3d	画 3 维直方图并填充
fillellipse	画椭圆并填充
fillpoly	画多边形并填充
floodfill	填充一个有界区域
getfillpattern	返回一个用户定义的填充模式
getfillsettings	返回当前填充模式及颜色
pieslice	画扇形并填充
sector	画椭圆形饼状图形并填充
setfillpattern	选择用户定义的填充模式
setfillstyle	设置填充模式及颜色

使用 Turbo C 的绘图函数, 可以画出彩色的线、弧、圆、椭圆、矩形、扇形、2 维及 3 维的条形 (直方) 图、多边形及由这些图形组成的规则或不规则的图形; 可以用 11 种预定义的模式或自定义的模式来填充任何有界区域; 可以控制画线的粗细、格式及当前位置 (CP) 的定位。

arc, circle, drawpoly, ellipse, line, linereel, lineto 及 rectangle 等用于画线段或未填充的图形; 用 floodfill 填充图形, 或用 bar, bar3d, fillellipse, fillpoly, pieslice 及 sector 等将绘图/填充合为一体。setlinestyle 设定画线 (及被填充图形的边界线) 的粗细、虚实、以及所定义的一些画线模式。setfillstyle 选择一个预定义的填充模式, 或用 setfillpattern 自己定义填充模式。moveto 可以将当前位置 (CP) 移到指定处, 而 moverel 则将其相对移动一定的距离。getlinesettings 可以获得当前线段的格式及宽度; getfillsettings 可以获得当前的填充模式及颜色; getfillpattern 可以得到用户定义的填充模式; getaspectratio 可以得到纵横比率, 用 getarccords 则

可获得上次画过的弧或椭圆的坐标, 若所画的圆有失真, 可用 setaspectratio 来校正。

三、屏幕及视口管理函数

除了绘图函数以外, Turbo C 图形库还提供一些管理屏幕、视口、图象及像素的函数。如表 1—8 所示:

表 1—8 屏幕及视口管理函数

	函数名	功 能
屏幕管理	cleardevice	清屏 (活动页)
	setactivepage	设置图形输出的活动页
	setvisualpage	设置可见的图形页数
视口管理	clearviewport	清除当前视口
	getviewsettings	返回当前视口的信息
	setviewport	设置图形输出的当前视口
图象管理	getimage	将指定区域的位图象存入内存中
	imagesize	返回存储屏幕上某一矩形区域所需的字节数
	putimage	将上次所存的位图象送到屏幕上
像素管理	getpixel	取 (x, y) 处的像素色
	putpixel	在 (x, y) 处画一像素

调用 cleardevice 可以立即清除全屏幕, 并将在视口中的当前位置置于原点处, 保持其他的图形系统设置 (线段、填充、正文格式、模式、视口设置等) 不变, 系统可以有一到八个屏幕页缓冲, 即内存中按点存储各屏的整屏图象的区域, 缓冲区的数目取决于图形卡。函数 setactivepage 可以指定某页是当前活动页 (图形函数的输出页), setvisualpage 可以指定某页是当前可见页 (屏幕所显示的页)。

在图形模式下, 用 setviewport 可在屏幕上定义一视口。视口的位置用屏幕绝对坐标定义, 并且可以将剪裁状态设为 on (有效) 或 off, clearviewport 用于清除视口。getviewsettings 找出当前视口的屏幕绝对坐标及剪裁状态。

getimage 可获得屏幕上图象的一部分, 用 imagesize 可计算在内存中需多少字节来存储所取的图象, 可用 putimage 将所存的图象回送到屏幕上 (任意地方); 可用 getpixel (返回一个给定像素的颜色) 和 putpixel (用给定的颜色画一个像素) 来控制各个像素的颜色。

四、颜色函数

颜色控制函数如表 1—9 所示:

表 1—9 颜色控制函数

	函数名	功 能
取 颜 色 信 息	getbkcolor	返回当前的背景色
	getcolor	返回当前的绘图色
	getdefaultpalette	返回调色板定义结构
	getmaxcolor	返回在当前图形模式下可获得的最大颜色值
	getpalette	返回当前的调色板及其大小
颜 色 设 定	getpalettesize	返回调色板查找表的大小
	setallpalette	按指定的颜色改变所有的调色板
	setbkcolor	设置当前的背景色
	setcolor	设置当前的绘图色
	setpalette	按指定的参数来改变某一调色板的颜色

图形屏幕由一个象素阵列组成。每个象素都在屏幕上产生一个(带色的)点,象素的数值并不直接决定真实的颜色、而是调色板(palette)的颜色表的一个索引。调色板的各项与给定的象素值相对应,保存某个象素真实的色彩信息。getmaxcolor 将最大有效象素(size-1)返回给当前的图形驱动程序及模式。

在讨论 Turbo C 的图形函数时常用到当前绘图颜色、填充颜色及象素颜色等,实际上,这个颜色是一个象素的数值,它是对调色板一个索引值。只有调色板才能决定在屏幕上显示真正的颜色是什么。通过改变调色板、就可以改变在屏幕上显示的实际色彩,尽管象素的值(绘图色,填充色等)并未改变。

背景颜色总是对应于象素值 0。当某块区域被清成背景色时,那块区域的象素值就被置成 0。绘图色是画线时象素所置的数值。可用 setcolor(n)来选择一个绘图色,n 是当前调色板的一个有效象素值。

五、图形模式下正文输出函数

图形模式下正文输出函数如表 1—10 所示。

表 1—10 图形模式正文输出函数

函 数	意 义
gettextsettings	返回当前的正文字体、方向、大小及对齐方式
outtext	发送一个字符串到屏幕上的当前位置(CP)
outtextxy	发送一个字符串到屏幕上的指定位置
registerbgiFont	注册一个被链接过的或用户安装的字体
settexitjustify	设置用于 outtext 和 outtextxy 的对齐方式
settexitstyle	设置当前的正文字体、格式及字符的放大率
setusercharsize	设置笔划字体的高宽比例
textheight	以象素形式返回一个字符串的高
textwidth	以象素形式返回一个字符串的宽

输出正文用 outtext 和 outtextxy 函数;settextjustify 函数控制输出正文的对齐方式;settextstyle 选择字符字体、方向及大小。使用 gettextsettings 可

获取当前正文设置,setusercharsize 能修改笔划字体字符的高宽比例。

六、图形模式下错误处理函数

图形模式下错误处理函数为:

grapherrormsg 返回指定错误代码的错误信息串

graphresult 返回最后一次出错误图形操作的错代码

如果调用图形函数时出现错误,就要置内部错误代码。用 graphresult 来检索最后报告出错的图形操作的出错误代码。出错误代码及其相应的信息串如表 1—11 所示。

表 1—11 出错误代码及其相应的信息串

错误代码	graphics errors 常 量	对应的错误信息串
0	grOk	No error(无错)
-1	grNoInitGraph	(BGI) graphics not installed (use initgraph)((BGI)图形未安装(使用 initgraph))
-2	grNotDetected	graphics hardware not detected (图形硬件未找到)
-3	grFileNotFound	Device driver file not found (设备驱动文件未找到)
-4	grInvalidDrive	Invalid device driver file(无效的设备驱动文件)
-5	grNoLoadMem	Not enough memory to load driver(没有足够的内存来装入驱动程序)
-6	grNoScanMem	Out of memory in scan fill(在扫描填充时内存不够)
-7	grNoFloodMem	out of memory in flood fill (在填充时内存不够)
-8	grFontNotFound	Font file not found(字体文件未找到)
-9	grNoFontMem	Not enough memory to load font (装载字体时内存不够)
-10	grInvalidMode	Invalid graphics mode for selected driver(对于选定的驱动程序图形模式是无效的)
-11	grError	Graphics error(图形错误)
-12	grIOerror	Graphics I/O error(图形 I/O 错误)
-13	grInvalidFont	Invalid font file(无效的字体文件)
-14	grInvalidFontNum	Invalid font number(无效的字体号)
-15	grInvalidDeviceNum	Invalid device nymber(无效的设备号)
-18	grInvalidVersion	Invalid version of file(无效的文件版本)

当且仅当图形函数报告一个错误时,错误返回码才变化。只有当 `initgraph` 执行成功之后,或调用了 `graphresult`, 错误返回码才重置为零。因此,一般想知道图形函数返回了什么错误代码,应先将 `graphresult` 的值存入一个临时变量中,然后再检测它。

七、状态查询

Turbo C 的每一类图形函数至少有一个状态查询函数。每个 Turbo C 图形查询函数只返回一个代表所查信息的数值或用相应的信息填写一个结构而不返回数值。状态查询函数如表 1—12 所示。

表 1—12 状态查询函数

函数名	功 能
<code>getarc</code>	返回上次调用 <code>arc</code> 或 <code>ellipse</code> 的坐标信息
<code>getaspectratio</code>	返回图形屏幕的纵横比
<code>getbkcolor</code>	返回当前的背景色
<code>getcolor</code>	返回当前的绘图色
<code>getdrivername</code>	返回当前图形驱动程序名
<code>getfillpattern</code>	返回用户定义的填充模式
<code>getfillsettings</code>	返回当前填充模式及颜色
<code>getgraphmode</code>	返回当前图形模式
<code>getlinesettings</code>	返回当前线段格式、模式和粗细
<code>getmaxcolor</code>	返回当前最高的有效像素值
<code>getmaxmode</code>	返回最大的模式给当前的驱动程序
<code>getmaxx</code>	返回当前 x 向的分辨率
<code>getmaxy</code>	返回当前 y 向的分辨率
<code>getmodename</code>	返回一个给定驱动程序的模式名
<code>getmoderange</code>	返回一个给定驱动程序的模式范围
<code>getpalette</code>	返回当前的调色板及其尺寸
<code>getpixel</code>	返回(x,y)处的像素颜色
<code>gettextsettings</code>	返回当前正文字体、方向、尺寸及对齐方式
<code>getviewsettings</code>	返回当前视口的信息
<code>getx</code>	返回当前位置(CP)的 x 坐标
<code>gety</code>	返回当前位置(CP)的 y 坐标

图形系统控制类的查询状态函数有 `getgraphmode`, `getmaxmode` 及 `getmoderangs`。第一个返回一个代表当前图形驱动程序模式的整数,第二个将模式范围返回给一个给定的驱动程序,第三个返回一个给定图形驱动程序所支持的模式范围。`getmaxx` 和 `getmaxy` 返回当前图形模式的最大 x, y 屏幕坐标。

绘图与填充状态查询函数有: `getarc`, `getaspectratio`, `getfillpattern`, `getfillsettings` 及 `getlinesettings`。`getarc` 用上次调用 `arc` 或 `ellipse` 的坐标来填充一个结构; `getaspectratio` 则返回当前模式下的纵横比率,图形系统用它来保证所画的圆不失真; `getfillpattern` 则返回当前的用户定义的填充模式; `getfillsettings` 则用当前填充模式及颜色来填充一个

结构; `getlinesettings` 用当前线段格式、线宽及线的模式来填写一个结构。

屏幕管理及视口管理的状态查询函数中,可用 `getviewsettings` 查找视口定义后其屏幕绝对坐标及剪口是否有效; `getx`、`gety` 返回当前位置的 x, y 坐标; `getpixel` 返回一个指定像素的颜色。

在图形正文模式中, `gettextsettings` 用诸如当前字符字体,正文显示方向,字符放大系数等信息填写一个结构。

颜色控制查询函数有 `getbkcolor`、`getcolor` 和 `getpalette`。`getbkcolor` 返回当前的背景色, `getcolor` 返回当前绘图色, `getpalette` 则用当前的绘图调色板及调色板常量来填写一个结构。

`getmodename` 及 `getdrivername` 分别返回一个给定驱动程序的名字及一个当前图形驱动程序的名字。

§ 1.5 Turbo C 绘图源程序的输入

一、进入 Turbo C 绘图世界

在 IBM PC 系列微机上, Turbo C (1.5 以上版本)为用户提供了一个功能很强的图形库,这个图形库支持 CGA、EGA、VGA 等多种显示卡的各种显示模式,提供了画点、线、圆、椭圆、多边形、输出各种字体、大小的字符等多种原语,为开发高质量的含有图形显示的程序软件提供了许多方便。

1. 打开 Turbo C 绘图世界的大门

在 C 语言程序中使用绘图系统,除了在源程序中指出嵌入文件 `<GRAPHICS.H>` 外,在连接成可执行文件时,还需要用图形库文件 `GRAPHICS.LIB`。如果使用的是 Turbo C 1.5 版本在进行一般的系统安装时,没有将 `GRAPHICS.LIB` 文件接入模式库中。因此,需先用 `TLIB.EXE` 文件将 `GRAPHICS.LIB` 文件连接进入模式库。

`TLIB.EXE` 是 Turbo C 主目录下的一个库管理程序,它能将一个库加到另一个库中。利用 `TLIB.EXE` 将 `GRAPHICS.LIB` 文件添加到 Turbo C 库的操作如下:

(1) 在 C> 提示符下键入 `CD\TURBOC\LIB`, 按回车键;

(2) 依次键入如下五条命令:

```

\TURBOC\TLIB CS+GRAPHICS.LIB
\TURBOC\TLIB CC+GRAPHICS.LIB
\TURBOC\TLIB CM+GRAPHICS.LIB
\TURBOC\TLIB CL+GRAPHICS.LIB
\TURBOC\TLIB CH+GRAPHICS.LIB

```

将 `GRAPHICS.LIB` 文件分别添加到 Turbo c 的五