

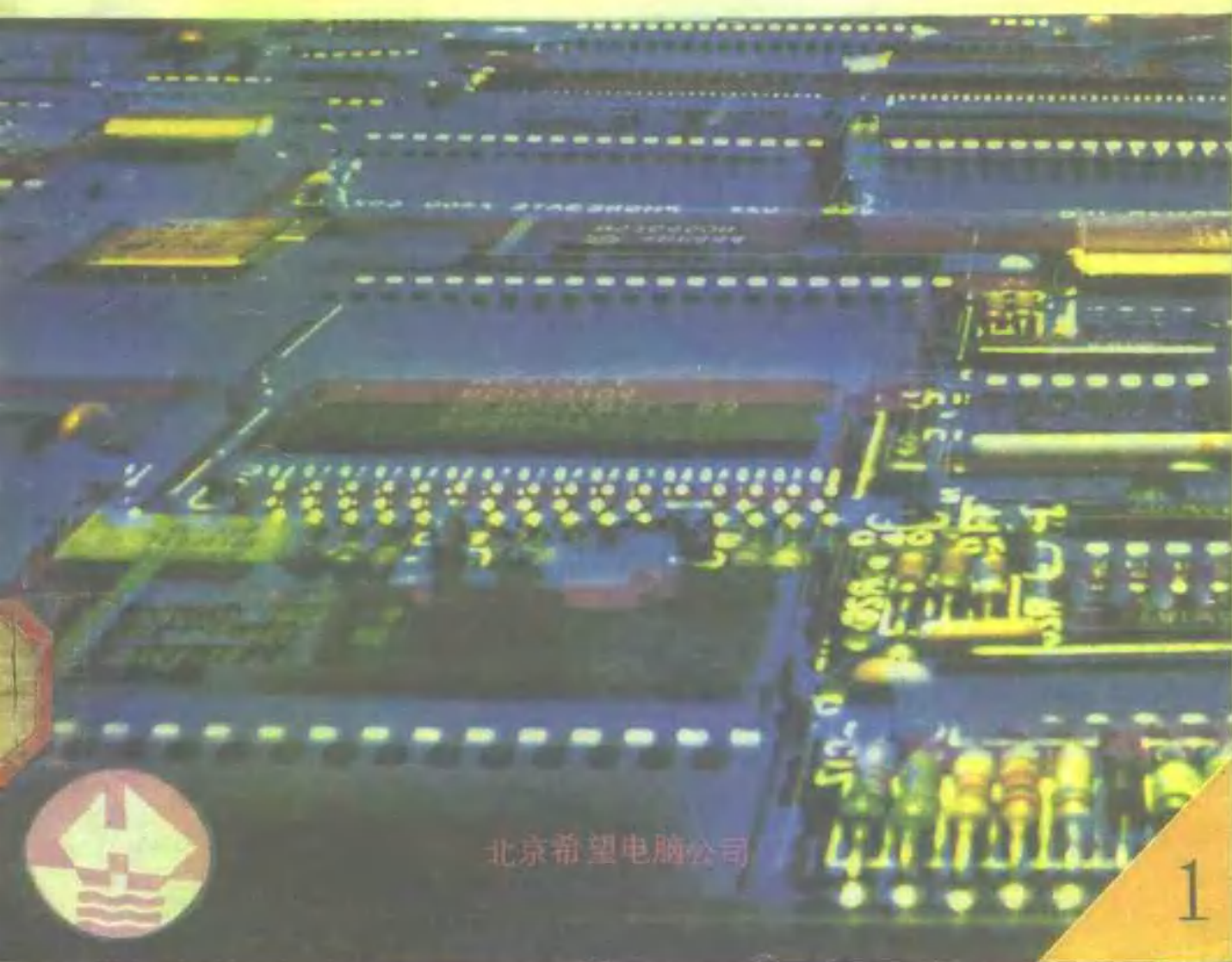
# HOPE

6.0 版

## Turbo C TOOLS 源程序剖析

## Turbo C 高级程序设计实例

(共二册)



北京希望电脑公司

1

L302-2

1

354254

# Turbo C TOOLS 6.0 源程序剖析

## Turbo C 高级程序设计实例(上)

李 文 编译

- 域编辑编程扩展
- 扩展的文件操作
- 帮助系统程序设计
- 用 C 进行中断服务子程序的设计
- 内存驻留程序设计的插入码
- 高级键盘管理程序设计
- 内存管理高级程序设计
- 选单程序设计
- 鼠标器编程
- 打印机编程
- 屏幕(视频)编程
- 字符串处理
- 实用函数和宏
- 窗口系统高级程序设计
- 创建扩展函数库



中国科学院希望高级电脑技术公司

一九九一年七月

# 前言

Turbo C TOOLS 6.0 在 5.0 版本的基础上, 提供了更丰富的函数。本书详细介绍了 Turbo C Tools 6.0 中每一个函数的使用方法, 介绍的函数比在手册中介绍的函数还要多。并提供了字符串转换、屏幕操作、窗口、选单、编辑器、帮助系统、鼠标器、键盘、文件、打印机、内存管理、中断服务、插入码等所有函数的源程序, 其中包括珍贵的虚拟窗口、虚拟选单、编辑器、帮助系统和鼠标器几类重要的函数的源程序。

通过函数包, 本书还介绍了用 Turbo C 进行字符串转换、屏幕操作、窗口、选单、编辑器、帮助系统、鼠标器、键盘、文件、打印机、内存管理、中断服务、插入码和内存驻留程序(TSR)程序设计的要点和技巧。

函数之多, 俯首拾来, 皆成程序。这些函数能够帮助用户设计出赏心悦目, 而且功能强大的程序。源程序能让您进一步探索这些函数或对它进行有效的开发, 仔细研读这些子程序, 能更好地使用这些函数, 开发自己的实用函数包, 极大地提高实际的编程水平。

## 本书的组织

本书分成十三章, 和 A、B 和 C 三个附录。

第一章 Turbo C 工具扩展源程序的使用说明

第二章 域编辑编程扩展

第三章 文件操作

第四章 帮助系统程序设计

第五章 用 C 进行中断服务子程序的设计

第六章 内存驻留程序设计的插入码

第七章 高级键盘管理程序设计

第八章 内存管理高级程序设计

第九章 选单程序设计

第十章 鼠标器编程

第十一章 打印机编程

第十二章 屏幕(视频)编程

第十三章 字符串处理

第十四章 实用函数和宏

第十五章 窗口系统高级程序设计

附录 A 使用的汇编头文件

附录 B 创建功能强大的函数库的批命令

附录 C 函数包使用的头文件

在本书的编译、整理过程中, 得到了多方面的关心和帮助, 特别是得到了希望公司经理秦人华同志的大力支持和热情鼓励, 在此编者表示衷心的感谢。

# 目 录

## 前言

第一章 Turbo C 工具源程序的使用说明 .....	1
函数包介绍 .....	1
函数的分类 .....	1
软件需求 .....	2
硬件需求 .....	2
支持的编译器版本和内存模式 .....	2
警告和使用注意事项 .....	3
异常错误及解决方法 .....	3
提取源代码 .....	3
打印源代码 .....	4
从源代码中取消 TAB .....	4
重建函数库 .....	4
重建示例程序 .....	5
C 和汇编模块及宏 .....	5
函数库建造说明 .....	7
利用数包的程序设计 .....	7
编程风格、编译警告和头文件 .....	7
修改 Turbo C TOOLS .....	8
源代码约定 .....	8
插入码函数程序设计示例 .....	8
第二章 域编辑编程扩展 .....	12
域编辑函数的种类 .....	12
数据项和编辑 .....	12
编辑键定义 .....	12
编辑操作和缺省的按键分配 .....	12
域编辑函数源程序 .....	14
EDBASE 编辑和返回一个用户响应 .....	14
EDBUFFER 对编辑缓冲区执行编辑动作 .....	22
EDCHGKEY 加入或修改 EDFIELD 和 WNFIELD 所认可的一个按键 .....	30
EDFIELD 对屏幕上的一个域进行编辑 .....	32
EDINITKY 安装 EDFIELD 和 WNFIELD 所接受的缺省按键 .....	34
EDREDUCE 把连续的空白转换成单个空格 .....	36
EDREMKEY 删去 EDFIELD 和 WNFIELD 所接受的一个按键 .....	37
EDRETINF 返回当前光标和视屏信息 .....	38
EDRETKEY 报告一个按键的编辑动作 .....	39
EDSETCUR 设置光标的尺寸和位置 .....	40
EDWRAP 向编辑区写入字符时, 带有整字换行 .....	41
EDWRECT 向屏幕写入一长方形图形 .....	42
EDZAPKEY 删去 EDFIELD 和 WNFIELD 所接受的按键队列 .....	43
编辑函数程序设计示例 .....	43
KEYCTRL.C 键控制函数的样例 .....	44
ENTRYEDT.C 是面向窗口多编辑域的程序 .....	45

<b>第三章 文件操作</b> .....	<b>52</b>
文件管理函数(FL) .....	52
文件管理函数的种类 .....	52
维护卷标 .....	52
杂类 .....	52
文件操作函数源程序和使用参考 .....	52
FLDOLOCK     对已打开文件一个文件段上锁或解锁, 必要时等待 .....	52
FLFLUSH     迫使挂起的文件输出写到磁盘上 .....	54
FLGETDTA     返回磁盘传送地址 .....	55
FLOCK        对已打开文件的一个文件段上锁或解锁 .....	55
FLNORM       验证一个文件名, 把它转化成标准形式 .....	56
FLPROMPT     从标准输入中返回一行, 可以使用提示信息 .....	66
FLPUTDTA     设置磁盘传送地址(DTA) .....	67
FLREMOVOL    从给定的磁盘上删除卷标(如果有的话) .....	67
FLRETVOL     报告给定磁盘驱动器上的卷标 .....	69
FLSETVOL     建立或修改给定磁盘上的卷标 .....	71
文件操作程序设计示例 .....	73
<b>第四章 帮助系统程序设计</b> .....	<b>75</b>
帮助系统(HL) .....	75
帮助函数的功能 .....	75
设计帮助窗口 .....	75
编写帮助源文件 .....	75
控制内存分配 .....	77
帮助函数源程序 .....	77
HLCLOSE     释放二叉帮助文件的已存索引 .....	78
HLDISP     从帮助文件读取一段帮助信息, 显示在屏幕上, 供用户浏览 .....	78
HLFRINDX    释放在给定节点以下的所有帮助索引节点 .....	79
HLLOOKUP    从二叉帮助文件中读取一段帮助信息 .....	80
HLPAS2C     转换一个 Pascal 的字符串为 C 的格式 .....	88
HLREAD     在视口中显示帮助信息, 供用户浏览 .....	89
帮助程序示例 .....	94
<b>第五章 用 C 进行中断服务子程序的设计</b> .....	<b>128</b>
概述 .....	128
ISR 的一般用法 .....	128
建立 ISR .....	128
驻留程序 .....	128
调用步骤 .....	128
删除 .....	129
按惯例过滤中断 .....	129
从 ISR 中的特殊退出 .....	129
扩展函数源程序 .....	130
ISCALL     对软件中断调用中断服务例程进行模拟 .....	130
ISCURPRC    返回或设置当前执行的进程 .....	134
ISGETVEC    返回一个中断向量 .....	134
isdispat    中断服务子程序调度程序 .....	135
ISINSTAL    安装一个中断服务例程(ISR) .....	144
ISPREP     预备一个 ISR 控制块 .....	145
ISPUTVEC    设置一个中断向量 .....	147
ISREMOVE    去除一个驻留程序 .....	148

ISRESERV	保留 ISR 所需的动态内存	149
ISRESEXT	中止一个程序但保持驻留	150
ISSENSE	检测一个已安装的中断服务例程(ISR)	151
	使用中断子程序的程序设计	152
CTLBRK.C	说明 control-break 处理子程序的实现和使用	156
CRITERR.C	说明严重错误处理子程序和使用	157
<b>第六章 内存驻留程序设计的插入码</b>		<b>161</b>
	概述	161
	插入码实用函数	161
	建立一个插入函数	161
	调度	161
	调用步骤	162
	重新调度	162
	摘除	162
	高级用法	162
	使用扩展键盘 BIOS	162
	暂时使一个插入函数失效	163
	选择插入过滤程序	163
	防止异步通讯拥塞	163
	插入码扩展码源程序	163
IVCTRL	报告本程序中插入控制块的地址	163
IVDETECT	检测已安装的插入函数, 即使它被部分覆盖	176
IVDISABL	使一个插入函数失效	181
IVINSTAL	安装一个插入函数	182
IVSENSE	检测一个已安装的插入函数是否是可删除的	192
IVVECS	设置或返回插入过滤程序所使用的中断向量	192
<b>第七章 高级键盘管理程序设计</b>		<b>194</b>
	增强键盘	194
	键盘函数的功能	194
	键盘输入	194
	处理超前键入缓冲区	194
	处理移位键	194
	使用增强键盘	194
	使用键控制函数	194
	取得键码	195
	高级特性: 键控制函数	195
	调用步骤	195
	高级键盘管理函数源程序和使用参考	195
KBEQUIP	检测键盘环境	196
KBEXTEND	选用扩展的或一般的 BIOS 键盘服务	197
KBFLUSH	废弃所有在键盘缓冲区中等待的按键	198
KBGETKEY	等待读入下一个按键	198
KBKCFLSH	通过键控制函数废弃所有的等待按键	199
KBPLACE	在键盘缓冲区中放置一个按键	200
KB POLL	通过一个键控制函数查看下一个等待按键	202
KBQUERY	从标准 IBM 控制台读邓用户的响应	203
KBQUEUE	报告键盘缓冲区总容量及剩余容量	206
KBREADY	检查下一个等待按键	207
KBSCANOF	返回一个字符的键码	209

KBSET	设置移位键的当前状态	210
KBSTATUS	报告移位键的当前状态	211
KBSTUFF	强行将一个字符串送入 BIOS 超前键入缓冲区	212
KBWAIT	等待并通过键控制函数读取下一个按键	214
CMKEY.C	键盘宏程序示例	215
<b>第八章 内存管理高级程序设计</b>		<b>220</b>
内存管理函数		220
内存管理函数源程序		220
MMCTRL	读取 DOS 内存控制块	220
MMFIRST	报告第一个内存块的地址	221
MMSIZE	报告一个程序的尺寸	222
<b>第九章 选单程序设计</b>		<b>223</b>
选单函数功能概述		223
选单函数的种类		223
建立、显示和释放选单		223
定义标准选项和按键		223
定义 Lotus 形式的选项		223
使用鼠标器的准备工作		223
读取用户的反应		223
亮条操作		223
高级选单特性		223
按键动作和亮条移动		224
缺省的键分配		224
控制内存分配		225
选单扩展函数源程序和使用参考		225
MNATR	改变菜单在项的属性	225
MNCREATE	建立一个包含单信息的选单结构和窗口	227
MNCREAT0	分配并创建一个菜单结构	227
MNDEFKEY	增加约束菜单的缺省键	228
b_mndefkey	包含 MNDEFKEY 使用的缺省键表	229
MNDLITMS	释放选项表中所有项的内存	231
MNDLKEYS	释放键表中使用的所有的键的内存	232
MNDSPLAY	在同尺寸视口中显示一个选单	232
MNDSTROY	从屏幕上取消一个选单, 废弃其数据结构	233
MNFINDSL	找到给出起始坐标的菜单的第一可选项	234
MNHILIT0	移动或删除菜单的高亮条或项描述	236
MNHILITE	移动或取消选单亮条及选项说明	238
MNITEM	插入、修改或删除一个选项	239
MNITMKEY	向选单加入一个选项, 为它分配选择字符	244
MNKEY	加入、修改或取消一个选单的键分配	245
MNLITEM	加入、修改或取消一个 Lotus 形式的选项	249
MNLITKEY	加入一个 LOTUS 形式的选项, 为它分配选择字符	250
MNLREAD	通过 Lotus 形式的选单读入一个用户响应	252
MNREAD	从给定的开始行和列使用亮条读取菜单	253
MNMCHITM	匹配一说明条件的菜单项	274
MNMCHKEY	匹配一说明的键	275
MNMOUSE	加入、修改或删除一个选单认可的鼠标器事件	277
MNMSTYLE	设立一个标准选单鼠标器格式	281
MNREAD	读取来自选单的用户响应	284

MNVALMNO	检查 BMENU 结构的有效性	306
MNVDISP	在视口中显示一个虚拟选单	307
示例程序		308
源程序(MENU.C)		308
源程序(PULLMENU.C)		321
<b>第十章 鼠标器编程</b>		<b>333</b>
鼠标器事件的种类		333
访问鼠标器状态		333
控制鼠标器位置		333
控制鼠标的外观		333
对鼠标器硬件中断的反应		333
其它鼠标器操作		333
处理鼠标器中断		334
调用屏蔽: 相关事件组		334
调用步骤		334
鼠标器编程函数源程序函数使用参考		334
MOAVOID	在指定区域中隐藏鼠标	335
MOBUTTON	报告鼠标器按钮的按下/释放历史	336
mocatch	捕获鼠标器按钮的按下与释放	338
MOCHECK	检查最近发生的鼠标器事件	342
MOCURMOV	移动鼠标	348
MOEQUIP	检查鼠标器驱动程序的存在	349
MOGATE	鼠标器驱动程序的人口	350
MOGETMOV	报告自上次查询以来物理鼠标的移动	351
MOGRAPH	设置鼠标器图形方式光标	351
MOHANDLR	安装或摘除中断处理程序	352
MOHARD	设置鼠标器硬件字符方式光标	355
MOHIDE	隐藏或显示鼠标	356
moinst	无条件安装鼠标器中断处理子程序	357
MOJUMP	设置鼠标器加速阈值	357
MOLITYPEN	使鼠标器光笔模拟有效或失效	358
MOPRECLK	安装或删除 MOCHECK 所有的内部例程	359
MORANGE	设置鼠标器范围界限	360
MORESET	重置鼠标器驱动程序	361
MOSOFT	设置鼠标器软件字符方式光标	361
MOSPEED	设置鼠标器灵敏度	362
MOSTAT	报告鼠标器位置和按钮状态	363
movars	一些 MO 函数使用的全程变量:	
	b_mocatch, b_momask, b_modispat, b_mohanmask	363
MOUSEHANC	鼠标中断处理子程序演示程序	364
<b>第十一章 打印机编程</b>		<b>368</b>
BIOS 打印机接口		368
与 PRINT 程序的接口		368
打印机控制函数源程序		368
PRCANCEL	删除假脱机打印队列中一个或全部文件	368
PRCHAR	通过 BIOS 向打印机发送一个字符	369
PRERROR	返回解释错误代码的字符串, 这些错误代码来自 PR(打印机)函数	370
PRGETQ	报告假脱机打印队列中的一个文件名	371
PRINT	通过 BIOS 初始化一个打印口	373



PRINSTLD	检查驻留式假脱机打印系统 PRINT 是否已安装	373
PRSPool	将一个文件提交给假脱机打印系统	374
PRSTATUS	通过 BIOS 报告打印机的状态	375
<b>第十二章 屏幕(视频)编程</b>		
屏幕操作函数的种类		376
读取屏幕方式信息		376
选择显示设备和方式		376
在显示页之间切换		376
控制/读取光标形状和位置		376
清除和滚动		376
常规的屏幕写入		376
写入一个矩形区域		377
屏幕读取		377
调色板支持		377
对直接视频访问的支持		377
保存和恢复整个显示状态		377
强制快速屏幕访问		378
屏幕(视频)控制函数源程序		378
SCAPAGE	显示(激活)一个显示页	378
SCATTRIB	用指定的显示属性显示一个字符的拷贝	379
SCBLINK	选择前景闪烁或背景亮度	379
SCBORDER	设置当前显示屏幕的边界颜色	380
SCBOX	用图形字符在屏幕上画一个方框	381
SCCHGDEV	切换至彩色或单色显示	384
SCCLRMSG	清除屏幕上的消息	386
SCCURSET	移动当前显示页上的光标	387
SCCURST	返回当前显示页上的光标位置和尺寸	387
SCEQUIP	检测显示硬件环境	388
SCGETVID	记录整个显示状态	397
SCMODE	返回屏幕的显示方式	397
SCMODE4	设置方式 4 色板和背景颜色	399
SCNEWDEV	选择并设置显示设备, 设置字符行数	400
SCPAGE	设置当前显示页	408
SCPAGES	返回显示页的数目	408
SCPALI	定义一个 EGA、VGA、或 MCGA 色板颜色	410
SCPALETT	定义 EGA、VGA 或 MCGA 颜色的整个色板	411
SCPCLR	清除当前显示页	413
SCPGCUR	设置当前页的光标尺寸	414
SCREAD	从屏幕读取一个显示字符及其属性	416
SCRESTPG	恢复一个显示页	417
SCROWS	返回屏幕的字符行数	418
SCSAVEPG	保存一个显示页	419
SCSETVID	恢复整个显示状态	420
SCTTYWIN	以 TTY 方式向矩形区域写入一个字符	422
SCTTYWRT	以 TTY 方式向屏幕写一个字符	425
SCWRAP	以 TTY 方式向一个矩形中写入一个字符串, 带有整字换行	426
SCWRITE	在屏幕上显示一个字符的多个拷贝	430
VIARECT	改变屏幕上一个矩形的属性	430
VIDSPMSG	显示一条消息	432

vidirec0	直接从或向视屏适配读取或写入长方形的字符	442
VIHORIZ	在当前显示页上水平滚动正文列	445
VIPTR	将屏幕位置转换成内存地址	447
VIRDRECT	读取屏幕上一个矩形区域中的内容	448
VIRDSECT	将屏幕的一个区域读入内存中更大的矩形区域	449
VISCROLL	垂直滚动当前显示页上的正文行	450
VIWRRECT	向当前显示页上一个矩形区域中写入数据	450
VIWRSECT	显示矩形缓冲区中的一个矩形段	450
<b>第十三章 字符串处理</b>		<b>453</b>
字符串函数的种类		453
对字符串的一部分进行填充		453
查找		453
字符转换		453
处理 tab 字符		453
字符串处理扩展函数源程序		453
STPEXPAN	将 tab 字符转换为空格	456
STPJUST	在域中将一个字符串左右对齐或居中	458
STPTABFY	用 tab 字符替换空格	459
STPXlate	用翻译表翻译一个字符串	461
STSCHIND	查找字符串中的一个字符, 返回它的位置	462
<b>第十四章 实用函数和宏</b>		<b>463</b>
指针和地址		463
内存传送		463
探测指针错误		463
数据压缩		463
ANSISYS 支持		463
程序环境信息		463
口 I/O		464
时钟访问		464
关闭或开放中断		464
扬声器控制		464
算术计算		464
在长字节、字、字节和半字节之间转换		464
设置范围界限		465
数据类型操作		465
各种输出		465
实用函数源程序		465
UTANSI	检测、关闭或重新开放 ANSISYS	466
UTCHKNIL	报告无效指针赋值, 使程序夭折	469
UTCRT	取得 DOS 临界段标志的地址	470
UTCTLBRK	设置或返回 Ctrl-Break 检查的状态	470
UTDOSRDY	报告 DOS 服务是否可用	471
UTGETCLK	报告自午夜以来 BIOS 计时脉冲的个数	471
UTINIFLG	打开或关闭硬件中断	472
UTMODEL	报告 IBM 型号和子型号及 BIOS 版本	473
UTMOVMEN	不受限制地从内存或向内存任何位置拷贝数据	475
UTNORM	使一个指针具有最小的偏移值	477
UTNULCHK	检测无效的指针赋值	477
UTOFF	返回一个地址的偏移部分	478

UTPEEKB	从任意地址读取一个字节	479
UTPEEKN	从任意地址读取多个字节的数据	479
UTPEEKW	从任意地址读取一个字	479
UTPLONG	将一个指针转换为指向 20 位物理地址的指针	480
UTPOKEB	在任意地址存放一个字节的的数据	480
UTPOKEN	在任意地址存放多个字节的数据	480
UTPOKEW	在任意位置写入一个字的数据	481
UTSAFCPY	以确保不跨越段界的方式拷贝数据	481
UTSEG	返回任意地址的段部分	482
UTSLEEP	暂停处理直至经过几个计时脉冲之后	482
UTSPKR	打开或关闭扬声器	483
UTSQZSCN	压缩一个屏幕图象	484
UTTIM2TK	将时间转换为计时脉冲计数	489
UTTK2TIM	将计时脉冲计数转换为 24 小时制时间	489
UTTOFAR	用段和偏移构造一个双字指针	490
UTTOFARU	用一个段和偏移构造一个泛双字指针	491
UTUNSQZ	还原一个压缩屏幕图象	491
实用程序函数设计示例		493
<b>第十五章 窗口系统高级程序设计</b>		<b>498</b>
窗口功能概述		498
窗口函数的种类		498
建立和释放窗口结构		498
显示和删除窗口		498
虚拟窗口		498
窗口输出		498
窗口输入		499
滚动和清除		499
控制属性		499
控制光标		499
控制窗口任选项		499
使用 Turbo C 的字符窗口		500
为 WNREAD 进行窗口输入做准备		500
使用高级窗口特性		500
显示和更新窗口		500
虚拟窗口		500
不可删除的窗口		500
控制内存分配		500
窗口扩展函数源程序		501
WNATRBLK	修改窗口中一个矩形块的属性	501
WNATRSTR	改变窗口中一片连续位置的属性	502
WNATTR	改变当前窗口的属性	504
WNCHGEVN	加入或修改 WNREAD 认可的一个用户响应	504
WNCREATE	建立一个窗口结构	505
WNCOVER	如果被一给定的长方形区域遮盖, 则标为低层窗口	506
WNCREATU	分配和创建一窗口结构	507
WNCURMOV	移动当前窗口的光标	509
WNCUPRPOS	返回当前窗口的光标位置	510
WNCURSOR	激活一个窗口光标	511
WNDSPLAY	在同尺寸视口中显示一个窗口	512

WNCURTRK	移动窗口的光标, 如果必要的活调整数据区源 .....	513
WNDSTROY	废弃一个窗口结构 .....	515
WNERROR	记录窗口或选单的系统错误 .....	516
WNFIELD	对窗口中的一个域进行编辑 .....	516
WNFORGET	从屏幕位置脱离窗口, 但并不清除屏幕 .....	518
WNGETIMG	读取屏幕长方形区域的图象 .....	519
WNGETOPI	读取窗口信息项或状态 .....	521
WNHIDE	删除窗口但仍与视频显示设备和显示页联连 .....	524
undisp	暂时删除(不显示)屏幕上由一长方形区域遮盖的窗口 .....	527
cocovr	标志非覆盖的窗口 .....	528
upcovr	检查高层窗口是否覆盖长方形区域 .....	529
redisp	重新显示指定窗口以上的被暂时隐藏删除的窗口 .....	530
WNHORIZ	水平滚动当前窗口 .....	531
WNINITEV	为 WNREAD 安装缺省的窗口事件 .....	532
WNCMOVE	移动窗口的光标 .....	535
WNSETWIN	设置编译器 native 文本窗口的尺寸 .....	536
WNCHKDM	检查编译器的 native 文本窗口的尺寸和大小 .....	537
WNGETATR	得到当前 native 文本窗口的属性 .....	537
WNSETATR	设置当前 native 文本窗口的属性 .....	538
WNNEEDUP	标志一窗口为无用, 如果可能进行更新 .....	538
WNNUPBLK	标志一窗口为无用, 如果可能更新一部分 .....	539
WNORIGIN	在视口中称动窗口 .....	540
WNOVRLAP	报告两个长方形区域是否覆盖 .....	542
WNPADD	增加窗口到在一设备一或显示页上显示的窗口的链表上 .....	542
WNPGRM	从显示在一设备或显示页上的窗口链表上删除 .....	543
WNPIMBLK	输出窗口的一部分到屏幕 .....	544
WNPOLL	查询属于特定集的键盘或鼠标事件一次 .....	546
WNPRINTF	向当前窗口写入一个格式化的字符串 .....	548
WNPUTBOR	沿着长方形区域显示一边界 .....	549
WNPUTSEN	显示窗口的 sensor 表 .....	552
WNQUERY	返回经窗口得到的来自用户的字符串 .....	555
WNRDBUF	读取当前窗口中一片连续位置的内容 .....	558
WNREAD	允许用户在虚拟窗口中浏览 .....	560
POLL_EVENT1	查询窗口的鼠标或键盘事件一次 .....	566
WNREDRAW	重现显示在当前显示页上的全部窗口 .....	571
WNREMEVN	删去 WNREAD 接受的一个用户响应 .....	573
WNREMOVE	从屏幕上取消一个窗口 .....	575
WNRESRV	恢复窗口先前的屏幕内容 .....	575
WNRETEVN	从窗口列表中返回一窗口事件记录 .....	576
WNRETINF	返回窗口当前光标和尺寸的信息 .....	578
WNREVUPD	用显示的数据更新已保存的窗口图象 .....	579
WNSCRBLK	在窗口中以任意方向滚动一个矩形区域 .....	582
WNSCRLBR	向窗口加入一个滚动箭头 .....	584
WNSCROLL	垂直滚动当前窗口 .....	589
WNSELECT	选择用于 I/O 的窗口 .....	589
WNSETBUF	为 WNPRINTF 分配内部缓冲区 .....	590
WNSETCUR	设置当前窗口的状态、尺寸和位置 .....	591
WNSETOPT	设置窗口控制项 .....	592
WNSHOBLK	在视口间隙中显示一个窗口数据块 .....	597
WNSHOBLK	在窗口视区内显示一长方形区域的数据 .....	600

WNUNHIDE	重新显示隐藏的窗口 .....	603
WNUUPDATE	将挂起的输出写入窗口 .....	606
WNVALEV0	验证窗口事件 .....	607
WNVALNO0	验证窗口节点 .....	607
WNVALW0	验证 WINDOW 结构 .....	608
WVDISP	在视口中显示一个虚拟窗口 .....	608
WVWRAP	以 TTY 方式向当前窗口写入一个字符串, 常有整字换行 .....	612
WVWRBLF	向当前视口中的一片连续位置写入字符 .....	614
WVWRRECT	写入窗口的一个矩形区域 .....	616
WVWRSTR	以 TTY 方式向当前窗口写入一个字符串 .....	618
WVWRSTRN	以 TTY 方式向窗口写入一个字符串, 带有任选项 .....	619
WVWRFTY	以 TTY 方式向当前窗口写入一个字符 .....	623
WVWRFTYX	写一字符到一个 TTY 方式的窗口 .....	623
WVZAPEVN	删除 WVREAD 认可的窗口事件表 .....	624
WVZAPSEN	释放窗口的 sensor 列表 .....	625
附录 A 使用的汇编头文件 .....		626
COMP T2S.MAC .....		626
COMP T2M.MAC .....		626
COMP T2C.MAC .....		627
COMP T2L.MAC .....		627
COMP T2H.MAC .....		628
BEGINASML.MAC 编译依赖的符号和宏 .....		628
附录 B 创建功能强大的函数库批命令 .....		634
创建库文件的 LIB 程序的响应文件(LIBRESP) .....		636
附录 C 函数包使用的头文件 .....		639
BEDIT.H 域编辑函数的头文件 .....		639
BFILES.H 文件和目录函数的头文件 .....		642
BGENWIN.H 禁止提供编译器本地文本窗口支持的头文件 .....		644
BHELP.H 帮助函数的头文件 .....		644
BINTERV.H 插入函数的头文件 .....		649
BINTRUPT.H 中断服务函数的头文件 .....		653
BKEYBRD.H BIOS 键盘函数的头文件 .....		656
BKEYS.H 定义键码的头文件 .....		659
BLAISE.H 所有函数的头文件 .....		676
BMEM.H 内存管理函数的头文件 .....		676
BMENU.H 菜单函数的头文件 .....		677
BMOUSE.H 鼠标函数的头文件 .....		684
BNATVWN.H 支持编译器本地文本窗口的头文件 .....		688
BPRINT.H 支持打印机管理的头文件 .....		688
BSCREENS.H 屏幕函数的头文件 .....		689
BSTRINGS.H 字符串函数的文件 .....		694
BUTIL.H 实用函数的头文件 .....		694
BVIDEO.H 直接存取视屏硬件函数的头文件 .....		702
BWINDOW.H 窗口函数的头文件 .....		703

## 第十二章 屏幕(视频)编程

Turbo C TOOLS 的屏幕操作函数提供了一个调用 BIOS 视频服务的高级接口,因而也提供了一个与 IBM 显示硬件的接口。用户可以选取许多 I/O 函数,将它们用于标准的显示方式或显示页。当需要最快的速度时,用户还可直接访问显示内存。

### 屏幕操作函数的种类

351073

#### 读取屏幕方式信息

- SCEQUIP 识别所安装的各种不同的显示适配器以及安装时设置的可选参数及开关。
- SCMODE 报告当前的显示设备、当前方式、列数和活动的(即当前可见的)显示页。SCROWS 报告当前显示设备和方式所支持的正文行的数目。
- SCPAGES 报告当前显示设备在当前方式下所支持的显示页的数目。
- SCGETVID 报告当前显示设备的全部状态:方式、当前页、活动页、行、列及光标。

#### 选择显示设备和方式

- SCNEWDEV 选择一个显示方式,基于所需方式重置合适的设备(于是清除屏幕)。它可以选择 25-、30-、43-或 50-行方式。
- SCCHGDEV 切换至给定的显示适配器。(请注意下面提到的重要限制。)
- SCSETVID 将显示适配器恢复成原先由 SCGETVID 所记录的状态,包括方式、显示页和光标。

#### 在显示页之间切换

- SCAPAGE 显示(激活)当前显示设备上的所需页。它不改变光标的尺寸。
- SCPAGE 选择给定的当前设备上的一个显示页,将该页作为 Turbo C TOOLS 屏幕 I/O 函数使用的页。它不显示(激活)任何显示页。

#### 控制/读取光标形状和位置

- SCCURSOR 设置当前显示页上的光标位置和尺寸。
- SCCURSET 移动当前显示页上的光标。
- SCPGCUR 在当前显示页是活动(即可见)的情况下设置光标的尺寸。

#### 清除和滚动

- SCPCLR 清除当前显示页。
- SCCLRMSG 清除当前显示页上的一条信息。
- VISROLL 使当前显示页上的一个矩形区域上滚一页或下滚一页。
- VIHORIZ 使当前显示页上的一个矩形区域左移或右移。

#### 常规的屏幕写入

- SCATTRIB 从当前显示页上的光标位置开始写入一个字符和属性 9 对的拷贝。
- SCWRITE 在当前显示页上的光标位置写入一个字符的多个拷贝,原有的属性不变。
- SCTTYWRT 向当前显示页(如果是活动的)写入一个字符并移动光标。回车、换行、响铃和退格以电传(TTY)方式处理。必要时屏幕发生滚动。
- VIDSPMSG 用给定属性在当前显示页上的指定位置显示一条消息。

SCCLRMSG 清除当前显示页上的一条消息。

## 写入一个矩形区域

VIWRRECT 用一个缓冲区的内容填充当前显示页上的一个矩形区域, 该函数的任选项可以对属性进行控制。

VIWRSECT 类似于 VIWRRECT, 但 VIWRSECT 可以显示更大的矩形缓冲区中的一段。

VIATRECT 改变一个矩表区域的属性, 其中的字符不变。

SCTTYWIN 以 TTY 方式(正如 SCTTYWRT)写一个字符, 但所有输出及滚动均限定在一个矩形区域内。

SCWRAP 以 TTY 方式写入一个字符串(正象 SCTTYWIN), 但它可以“整字换行”, 这样能够避免在区域的行之间把一个字折断。

SCBOX 利用特殊的字符方式下的画框字符用单线或双线画一个方框。

SCRESTPG 利用 SCSAVEPG 保存的一个压缩拷贝恢复一个显示页。

## 屏幕读取

SCREAD 报告当前显示页光标位置的字符及其属性。

VIRDRECT 读取显示在一个矩形区域中的字符, 可带有或不带有属性。

VIRDSECT 类似 VIRDRECT, 但 VIRDSECT 可以将数据读入更大矩形区域的一段中。

SCSAVEPG 以压缩格式保存当前显示页的内容。

## 调色板支持

SCPALETT 定义由 EGA、VGA 或 MCGA 显示的整个颜色调色板。

SCPAL1 定义由 EGA、VGA 或 MCGA 显示的一个颜色。

SCMODE4 选择用于显示方式 4 (320X200 四色图形)的色板及背景颜色。

SCBORDER 设置边界颜色。

SCBLINK 选择在 Enhanced Graphics Adapter 下属性位 7 的作用: 前景闪烁背景亮度。

## 对直接视频访问的支持

VIPTR 返回一个 far 指针, 该指针指向当前显示页上的一个给定位置(在显示内存中)。

## 保存和恢复整个显示状态

让应用程序在结束之后不干扰显示环境, 这是一种良好的行为, 有时甚至是必须的。这就需要保存显示状态, 在程序退出前小心地恢复状态。

Turbo C TOOLS 提供了一些函数来完成这些任务。SCGETVID 和 SCSETVID 读取和恢复显示方式及光标尺寸而 SCSAVEPG 和 SCRESTPG 记录并恢复所显示的正文数据。

下面是函数的列表, 这些函数保存和恢复显示环境的特定参数。

保存和恢复显示状态的函数

	读取	恢复
当前显示设备	SCMODE	SCNEWDEV
方式	SCMODE	SCNEWDEV
正文行数	SCROWS	SCNEWDEV
活动显示页	SCMODE	SCAPAGE
光标位置	SCCURST	SCCURSET
光标尺寸	SCCURST	SCPGCUR
屏幕上的正文	SCREAD、 VIRDRECT SCSAVEPG	SCATTRIB、 SCWRITE、 VIWRRECT 或 SCRESTPG

屏幕上的属性	SCREAD,	SCATTRIB,
	VIRIRECT 或	VIWRRECT,
	SCSAVEPG	VIATRECT 或
		SCRESTPG

由于 BIOS 未记录某些显示状态, 所以有时不可能获取这些参数, 例如色板。

## 强制快速屏幕访问

直接显示(VI)函数检测 Color/Graphics Adapter 的存在, 计算操作的时间, 防止视频干扰(“下雪”)。然而在某些情况下用户有时希望避免出现延迟。例如, 有些显示适配器可以模拟 CGA 而不会出现雪花, 它们并不需要特殊的计时工作。有时雪花不是一个问题。

要想使 VI 函数工作得尽可能快而不考虑干扰问题, 可使用下面的语句:

```
#include <bvideo.h>
b_vifast = 1;
```

下面的语句将 VI 函数重置为一般的方式, 避免 CGA 上的干扰。

```
b_vifast = 0;
```

## 屏幕(视频)控制函数源程序

### SCAPAGE 显示(激活)一个显示页

```
#include <bcreens.h>
int scapage(int page);
page 待显示(激活)的页号。
(返回) 实际显示的显示页。
```

SCAPAGE 设置活动的显示页, 即显示该页。它不改变光标尺寸, 也不打开或关闭光标。

SCAPAGE 检查当前的显示方式。如果 page 超出范围, 则该值被强置为合适的值: 负页号置成 0, 过大的正页号置为可用的最大页号, 实际显示的页号作为函数值返回。

用 SCPAGES 可以查出当前设备在当前方式下能够支持多少页。SCPAGE 指示 Turbo C TOOLS 将屏幕输出送到一个给定页, 不论它是活动的(显示的)还是不活动的。

源程序(SCAPAGE.C):

```
#include <dos.h>
#include <bcreens.h>
int scapage(page)
int page;
{
    int max_page;
    union REGS inregs, outregs;
    max_page = scpages() - 1; /* Maximum page value */
    utbound(page, 0, max_page)
    if (max_page) /* Switch active page only if */
    { /* this device supports more */
        inregs.h.ah = 5; /* than one. */
        inregs.h.al = (unsigned char) page;
        int86(16, &inregs, &outregs);
    }
    return(page);
}
```



## SCATTRIB

### 用指定的显示属性显示一个字符的拷贝

```
#include <bscreens.h>
int scattrib( int fore,
             int back,
             char ch,
             unsigned cnt);
```

fore 前景显示属性。  
back 背景显示属性。  
ch 待显示的字符。  
cnt 待显示的 ch 拷贝的数量。  
(返回) 返回值总为 0。

SCATTRIB 从当前光标位置开始显示字符 ch 的 cnt 份拷贝,光标保持不动。拷贝在当前页(b\_curpage)上以属性(fore,back)显示,字符以一行接一行的方式写入。

如果写入时超出屏幕的末端,或(在图形方式下)写入时超出当前的正文行,这时有可能出现未预料到的结果。SCATTRIB 不滚动屏幕。

#### 源程序(SCATTRIB.C):

```
#include <dos.h>
#include <bscreens.h>
int scattrib(fore,back,ch,cnt)
int fore;
int back;
char ch;
unsigned cnt;
{
    union REGS inregs,outregs;
    if (cnt)
    {
        inregs.h.ah = 9;
        inregs.h.al = ch;
        inregs.h.bh = (unsigned char) b_curpage;
        inregs.h.bl = (unsigned char) utnybbyt(back,fore);
        inregs.x.cx = cnt;
        int86(16,&inregs,&outregs);
    }
    return(0);
}
```

## SCBLINK

### 选择前景闪烁或背景亮度

```
#include <bscreens.h>
int scblink(int option);
option SC_BLINK (1) 使属性位 7 控制前景闪烁, SC_INTENSITY(0)
使属性位 7 控制背景亮度。
(返回) 返回的错误代码。可能值包括:
SC_NO_ERROR (0) 成功。
SC-BAD_OPT (1) 当前显示适配器不支持这个操作。
SCBLINK 控制在 EGA、VGA 和 MCGA 字符方式下属性位 7 的作用。
```