

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

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

第三章 文件操作	52
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
FLLOCK 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
第四章 帮助系统程序设计	75
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
第五章 用 C 进行中断服务子程序的设计	128
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
第六章 内存驻留程序设计的插入码		161
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
第七章 高级键盘管理程序设计		194
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
第八章 内存管理高级程序设计	220
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
第九章 选单程序设计	223
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
MNCREATO	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
MNHILITO	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

MINVALMNO	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
第十章 鼠标器编程	333
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器的移动	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	设置鼠标器敏感度	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
MOUSEHAN.C	鼠标中断处理子程序演示程序	364
第十一章 打印机编程	368
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
PRINIT	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
第十二章 屏幕(视频)编程		376
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
SCPAL1	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
✓ SCITYWIN	以 TTY 方式向矩形区域写入一个字符	422
✓ SCITYWRT	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
VIATRECT	改变屏幕上一个矩形的属性	430
✓ VIDSPMSG	显示一条消息	432

✓ vidirect	直接从或向视屏适配读取或写入长方形的字符	432
✓ 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	451
第十三章 字符串处理		453
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
STPIABFY	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
第十四章 实用函数和宏		463
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 IMB 型号和子型号及 BIOS 版本	473
UTMOV MEN	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
第十五章 窗口系统高级程序设计		498
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
WNATRLBK	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
WNCREAT0	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
WNSPLAY	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
WNEROR	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_EVENTI	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
WNRESPRV	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
WNUPDATE	将挂起的输出写入窗口	606
WNVALEVO	验证窗口事件	607
WNVALNOO	验证窗口节点	607
WNVALWIO	验证 BIWINDOW 结构	608
WNVDISP	在视口中显示一个虚拟窗口	608
WNWRAP	以 TTY 方式向当前窗口写入一个字符串, 常有整字换行	612
WNWRBUF	向当前视口中的一片连续位置写入字符	614
WNWRRECT	写入窗口的一个矩形区域	616
WNWRSTR	以 TTY 方式向当前窗口写入一个字符串	618
WNWRSTRN	以 TTY 方式向窗口写入一个字符串, 带有任选项	619
WNWRTTY	以 TTY 方式向当前窗口写入一个字符	623
WNWRTTYX	写一字符到一 TTY 方式的窗口	623
WNZAPEVN	删除 WNREAD 认可的窗口事件表	624
WNZAPSEN	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

第一章 Turbo C 工具源程序的使用说明

本章的主要内容

- 函数包介绍
- 软件和硬件要求
- 支持的编译器版本和内存模式
- 警告和使用注意事项
- 错误及解决方法提取源代码
- 打印源代码
- 从源代码中消除 tab
- 重建函数库
- 重建示例程序
- C 和汇编模块及宏函数库构造说明

函数包介绍

函数包中的函数能用于 IBM PC 系列的 Turbo C 程序中。它们提供：

- 屏幕和键盘的快速、多功能控制；
- 创建、显示和使用多窗口的窗口管理；
- 使用移动高亮条的用户界面的菜单；
- 帮助系统使产生正文的过程自动化；
- 支持与标准 Microsoft 兼容的鼠标器驱动器模块接口的函数；
- 完全用 C 语言编写中断服务程序的能力。这些程序处理硬件和软件中断，并调用 DOS 服务；
- 插入码：按一热键，或在给定的时间间隔里调度 C 函数执行的能力；
- 创建和删除驻留在内存中的程序(也叫 TSR 应用)的能力；
- 常驻伪脱打机制的界面；
- 具有强大功能的常规字符串函数。

Turbo C TOOLS 经过精心设计，支持许多易使用的重要的功能。整个源代码组装在一起，使用户能调整此程序包适应自己的需要。另一方面，用户使用 Turbo C TOOLS 时不必考察其源代码，除非有意这么做。若用户需要调整其中的功能以适应自己的特殊需要，将会看到 Turbo C TOOLS 的源代码是易于理解、修改和安装的。

Turbo C 是软件开发的优秀编译器。写得好的 Turbo C 程序是可靠的、有效的、可读性好的、易于维护的。并且 Turbo C 代码量少，运行速度快。然而，许多功能需要利用 IBM PC 环境中的高级硬件和软件特性才能实现，标准 Turbo C 库不支持这些高级功能。而 Turbo C TOOLS 支持许多附加的高级功能。

函数的分类

函数根据各工具解决的问题或相关问题集分类。共有 15 类：

- | 前缀 | 类别 |
|-----|----------|
| ST | 字符串函数 ✓ |
| ✓SC | 屏幕处理 ✓ |
| ✓VI | 直接视频访问 ✓ |
| GR | 图形 |
| ✓WN | 窗口管理 ✓ |
| ✓MN | 菜单 ✓ |
| HL | 帮助系统 |
| MO | 鼠标器支持 |

- KB 键盘控制
- FL 文件和目录管理
- PR 打印机和 PRINT.COM 伪机打印机界面
- MM 存贮管理
- IS 中断服务支持
- UT 实用函数和宏

每个函数的名字以分类字母为前缀。例如，屏幕处理类(前缀 SC)包括在屏幕上面框的函数 scbox()。象所有 Turbo C TOOLS 函数一样，调用它时使用它的小写字母名字，它在手册中和其他文档中的名字为大写 SCBOX()。

虽然，每类函数解决一个特定领域的问题，但 Turbo C TOOLS 中有些函数依赖于别的函数。我们力图使各类函数自包含，以便你不用研究整个产品，就可以学习使用其中的一类函数。然而，有些领域的问题不可避免的需要别的领域的知识。例如，为了充分发掘窗口管理函数的功能，有必要了解屏幕处理函数的功能和限制。

软件需求

有些 Turbo C TOOLS 函数是用汇编语言写的。一般不需汇编器，除非修改这些特殊函数。若一定要这样做，建议用 Microsoft 的 3.00 以上版本的宏汇编包。我们编译 Turbo C TOOLS 汇编语言时也使用它。

为了推广和维护目标模块库(.lib)，需要有与 Microsoft 库管理软件 LIB 兼容的库管理程序。这也由 Microsoft 宏汇编包提供。

为了推广可执行 .exe 文件，我们用到了 Turbo C 编译中的 TLINK 程序。

还需要 DOS 2.00 或更高版本，尽管有些函数只适用于 3.00 以上版本，但大部分函数在 DOS 的所有版本中的行为相同。需要 DOS 3.00 以上版的函数在说明中特加以强调。

硬件需求

许多 Turbo C TOOLS 程序能在运行 Microsoft 操作系统(MS-DOS) 2.00 以上版本的机器上正确执行。(IBM PC-DOS 是 Microsoft MS-DOS 的改写版。)事实上，ST、FL、MM 或 IS 类中的程序都不依赖于任何特定的 IBM 硬件特性。

支持的编译器版本和内存模式

这个版本的 Turbo C TOOLS 支持 Turbo C 版本 1.00、1.50 和 2.00。

Tiny(T) 模式：Turbo C TOOLS 不正式支持 T 模式，然而 S 模块库(TCT_T2S.LIB)中的很多模块在 T 模式程序中可以正确地工作。

Huge(H) 模式：许多使用 H 模式的程序可以使用 L 模式库(TCT_T2L.OBJ)，特别是如果这些程序不包含中断服务例程(ISR)或插入码。然而，如果需要支持大于 65,535 字节的数据对象或者程序工作失常，则应使用 H 内存模式重新编译这个库。参见下面的“重建函数库”。

下面的每个文件都针对于特定的内存模式，同一个文件持 Turbo C 1.00、1.50 和 2.00 几种版本，但在下面的“警告和使用注意事项”中列出了几个例外。下表的栏目解释如下：

内存模式	函数库	“原始窗口”	COMPILER.MAC	“不支持鼠标器”
S	TCT_T2S.LIB	NW_T2S.OBJ	COMP_T2S.MAC	BNOMOUSE.OBJ
M	TCT_T2M.LIB	NW_T2M.OBJ	COMP_T2M.MAC	BNOMOUSE.OBJ
C	TCT_T2C.LIB	NW_T2C.OBJ	COMP_T2C.MAC	BNOMOUSE.OBJ
L	TCT_T2L.LIB	NW_T2L.OBJ	COM_T2L.MAC	BNOMOUSE.OBJ

内存模式：它影响程序中数据和代码的最大长度以及数据和代码指针的特性。在 Turbo C 集成环境(TC.EXE)中，通过指明 Options/Compiler/Model 开关可以指定内存模式。缺省的模式是 S，除非您保存了一个不同的配置。对于命令行版本(TCC.EXE)，利用 /m 命令行开关能够指定内存模式。缺省值是 S 模式。

函数库：它们是所有 Turbo C TOOLS 函数的已编译的版本。要想把它们与您的程序相链接，需在编译器、链接器命令行或 project 文件中指明适当的库文件。

口操作与 Turbo C 字符窗口结合起来。这个目标文件需要 Turbo C 版本 1.50 或更高版本。

COMPILER.MAC 文件：它为用汇编语言写的 Turbo C 模块指明内存模式。如果重新汇编任何汇编模块，首先应将所需的文件拷贝到 COMPILER.MAC。

“不支持鼠标器”文件：与 BNOMOUSE.OBJ 相链接的程序不包含鼠标器函数 MOCHECK、MOHIDE 和 MOPRECLK。如果您的程序使用选单、WNREAD 或帮助系统并且您不希望使用鼠标器，则用 BNOMOUSE.OBJ 链接可以使程序小一些。请注意，同一个 BNOMOUSE.OBJ 文件支持所有的内存模式。

警告和使用注意事项

ISR 中的栈检查——对于 Turbo C 版本 2.00 和更高版本，在 T、S 和 M 内存模式下的中断服务例程 (ISR) 或插入函数中不允许做栈溢出检查，在所有受到影响的 ISR 和插入函数所调用的函数中也禁止做栈检查。(注意：仅当对程序进行编译时设置了 0/C/Code/Test Stack 选项(在集成环境中)或指明了 -N 命令行任选项时才会出现栈检查。)

TURBO C 1.0 与 FLPROMPT 及 MOCHECK——如果在 Turbo C 1.0 下程序使用 FLPROMPT 或 MOCHECK，那么您应该在版本 1.0 下重新编译 FLPROMPT.C 或 MOCHECK.C，将它们与程序的 .OBJ 文件相链接。(MOCHECK 能够检测鼠标器的点按动作，用于 HLDISP、HLREAD、MNREAD 和 WNREAD。如果程序与 BNOMOUSE.OBJ 链接，则 MOCHECK 被抑制。)

在 DOS 版本 2 下的插入码——在 DOS 版本 2 下插入调度程序不设置当前进程，因而在 DOS 版本下，后台程序中的插入函数应避免内存分配。另外，这些插入函数在每次被激活时应该打开和重新关闭它们的文件。

在非 IBM DOS 上的插入码——如果在安装插入函数时指明了 IV_DOS_NEED，则调用程序在中断 0x24(严重错误)正在进行的情况下将延迟激活插入函数。中断 0x24 忙标志的位置在不同 DOS 版本下有所不同。

滚动单色显示页——BIOS 滚动功能通常检查光标原位置来取得新空行的属性。Enhanced Graphics Adapter 在 Monochrome Display 下支持多个显示页。对于超过页 0 的单色显示页，BIOS 滚动操作可能会出现不正常的情况，新空行的属性将从页 0 而不是从活动页取得，这会影响 SCTTYWIN、SCTTYWRT 和 VISCROLL。

异常错误及解决方法

程序异常中止：如果启动模块不能为堆栈和 near 堆分配足够的空间，这时会出现程序异常中止的情况。建议在对全局变量 “_stklen” 和 “_heaplen” 进行初始化时试用一个小一些的值。

字符显现的颜色不正确：当 MNDSTROY 和 WNREMOVE 将 Turbo C 字符窗口设置为全屏时，它们并不改变字符的颜色，这样，以后通过 Turbo C 字符窗口函数输出的字符所显示的颜色可能是以前所选窗口的颜色。

鼠标器操作期间出现异步通讯口拥塞：由于 Microsoft mouse 通过硬件中断进行工作，所以鼠标器的移动和按钮的按下及释放有可能在数据到达串口时引起拥塞错误。在这种情况下首先应测试一下不用 Turbo C TOOLS 时是否会出现拥塞。在使用 HLDISP、HLREAD、MOCHECK、MNREAD 或 WNREAD 时将安装 MOCATCH，这个内部函数在一段相当长的时间内将关闭中断。

提取源代码

库的源代码以两个自展开的文档程序 TCTSRC1.EXE 和 TCTSRC2.EXE 的形式提供。后面列出了这两个文件的全部内容。

要想提取一个文件的内容，首先应将当前驱动器和目录设置为待放置展开文件的驱动器和目录，然后执行自展开文档文件，如下例所示：

```
a:tctsrc1
```

要想有选择地提取文档中的文件，可以使用多义文件名，如下例如示。它提取出所有窗口 (WN) 函数的源代码：

打印源代码

Turbo C TOOLS 经过了很好的设计和文档说明, 容易理解, 你不必参阅源代码。然而, 你也许希望能很容易地打印代码。

为了打印源代码, 我们提供了一个叫 CONCAT 的程序, 它产生格式化的打印文件。你会发现它是对你的软件库的有益补充。格式化打印文件的每页的顶部是文件名、最后修改日期和时间以及页号。每页的第二行显示 CONCAT 格式化此文件的时间。每个输出行以一个换页字符(^L)结束。

为了执行 CONCAT, 键入命令:

```
Concat /oofile /ppsize /mmarg /npnum /ddebug
```

其参数的意义如下:

/oOfile 是由 ofile 产生的格式化打印文件的名称。若此文件已存在, 则被冲掉, 若此文件不存在, 则创建之。若没有给出输出文件名, 则在当前目录上打开 concat.prt 文件。

/pPsize 是每页打印的行数。每页的头部需要 4 行(一个空白行, 2 个头信息行, 和另一个空白行); 因此最小的可接受值是 5。缺省值是 64, 对标准 66 行来说, 允许在每页的顶部有两个空白行。

/mMarg 是每个打印行前的空格数。缺省值是 0。最大值是 10。

/nPnum 是开始页号。文件的各页顺序编号。每个文件的开始页号的缺省值是 1, 如果/n 没有指定值, 则页编号顺序从 1 开始, 但是再也不为每个文件重新设置开始页号。

/dDebug 是纠错信息送到其中的某文件的名称。如果此开关没有指定一文件名, 则自动创建文件 concat.dbg。若没有指定此开关, 则不产生纠错文件。

所有的参数都是任选项。如果用到了开关, 则参数可以任何次序出现。如果没有用到开关, 则参数以上述顺序解释。开始字符的斜杠(/)可以用减号(-)代替。

例如, 下面两命令中的每个都能产生输出文件 foo.prt, 其格式是每页 64 个输出行, 左边的空边为 10 个字符宽, 首页号为 1:

```
concat foo.prt 64 10 1
```

```
conct foo.prt /n /m 10
```

执行 CONCAT 时, 屏幕上显示一欢迎信息, 然后出现提示符 > >。这时你可以键入一个文件的全路径名。此步骤可以重复任意次。一空反应(仅敲 FNTER 键)结束 CONCAT。

输出文件构造出来后, 可以用 DOS PRINT 命令打印之, 或拷贝到 prn(打印机)设备上。

CONCAT 展开 tab 字符成 8 个字符宽的空格, 此约定在 DOS PRINT 和 TYPE 命令中也用到。

找不到输入文件时显示一警告信息, 但是整个过程仍继续下去。CONCAT 在输出文件打不开或有错误写到输出文件中时才结束。

从源代码中取消 TAB

源模块将连续的空格压缩为 tab 字符, 每隔八个字符一个 tab 站位, 并以这种形式保存起来, 这里提供的 EXPAND.COM 工具程序可以将 tab 扩展回复为空格, 这对于从帮助源文件中取消 tab 是很有用的。

EXPAND 是一个过滤程序, 它从你在输入设备中取得输入, 将输出写至标准输出。然而, 您使用时可以重新定向标准输入和输出, 如下例所示:

```
expand <wnexampl.c>wnexampl.ntb
```

结果文件 WNEAMP.NTB 中的 tab 字符被扩展为空格串。

重建函数库

系统提供了一个批文件 BUILDLIB.BAT 和一个库管理程序响应文件 LIBRESP, 使得重建 Turbo C TOOLS 函数库的工作更为容易。批文件作了一些假设。如果这些假设不适合您的环境, 您应该修改这个批文件, 使之适合您的系统。这些假设是:

所有源文件(.C 和 .ASM)及 LIBRESP 文件在当前目录中;

BEGINASM.MAC 和 COMP_T2?.MAC 在 MAC 子目录中;

C 语言头文件(.H)在 \TURBOC\INCLUDE 目录中;

Turbo C 命令行编译器(TCC.EXE)、Turbo Assembler (TASM.EXE) 和 Turbo Librarian