

TURBO C TOOLS 6.0

使用手册

〔美〕 Blaise Computing Inc. 编

刘少文 译

电子工业出版社

TURBO C TOOLS 6.0

使 用 手 册

(美) BLAISE COMPUTING INC. 编

刘 少 文 译

電子工業出版社

内 容 提 要

本手册详细介绍了 TURBO C TOOLS 6.0 十余类库函数的使用方法, 列举了若干使用示例。这些库函数在原 TURBO C TOOLS 5.0 版本的基础上, 新增加了虚拟窗口、虚拟选单、编辑器、帮助系统和鼠标器等几类重要函数。本书是 TURBO C TOOLS 6.0 用户的必备使用手册, 也是计算机用户和软件开发人员及大专院校计算机专业师生的参考书。

TURBO C TOOLS 6.0 使用手册

(美)BLAISE COMPUTING INC. 编

刘 少 文 译

责任编辑: 阮 建

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

原子能工业出版社印刷厂印刷

开本: 788×1092 毫米 1/16 印张: 15.25 数: 450 千字

1991 年 5 月第 1 版 1991 年 5 月第 1 次印刷

印数: 6000 册 定价: 12.00 元

书号: ISBN7-5053-1401-7/TP·231

译者前言

Turbo C TOOLS 6.0 是美国 Blaise Computing Inc. 1989 年推出的 Turbo C 工具库, 用于 Turbo C 版本 1.0、1.5 和 2.0。这个工具库提供了丰富的库函数, 如字符串转换、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。其中, 新增的虚拟窗口、虚拟选单、编辑器、帮助系统和鼠标器几类重要的函数使得 Turbo C TOOLS 版本 6.0 在功能上大大优于原 5.0 版本, 这些新的函数能够帮助您设计出更为精致完美的用户界面, 使得应用程序在外观和操作上达到一个新的境界。

本书详细介绍了 Turbo C TOOLS 6.0 中每一个库函数的使用方法, 列举了使用示例。如果您想进一步揭开某个库函数的深层奥秘或对它进行改造, 那么软件上的源文件已经为您大开了方便之门。

本书的翻译和编排得到了于滨、陈铁刚、黄步霞、高银素、戎杰、王荣、王丽红和邱小红等许多朋友的热情关心和帮助, 我在此特向他们表示衷心的感谢。

译者

1990 年 12 月

目 录

第一部分 函数分类及概述

1. 字符串函数 (ST).....	1
2. 屏幕函数 (SC 和 VI).....	1
3. 窗口函数 (WN).....	4
4. 选单函数 (MN).....	7
5. 域编辑 (ED).....	9
6. 帮助系统 (HL).....	12
7. 鼠标器支持 (MO).....	15
8. 键盘函数 (KB).....	17
9. 文件管理函数 (FL).....	19
10. 打印机函数 (PR).....	20
11. 内存管理 (MM).....	20
12. 中断服务支持 (IS).....	21
13. 插入码 (IV).....	24
14. 实用函数和宏 (UT).....	27

第二部分 函数参考说明

1. EDBUFFER	对编辑缓冲区执行编辑动作.....	30
2. EDCHGKEY	加入或修改EDFIELD和WNFIELD所认可的一个按键.....	32
3. EDFIELD	对屏幕上的一个域进行编辑.....	33
4. EDINITKY	安装EDFIELD和WNFIELD所接受的缺省按键.....	36
5. EDREMKEY	删去EDFIELD和WNFIELD所接受的一个按键.....	36
6. EDRETKEY	报告一个按键的编辑动作.....	37
7. EDZAPKEY	删去EDFIELD和WNFIELD所接受的按键队列.....	38
8. FLDOLOCK	对已打开文件的一个文件段上锁或解锁.....	39
9. FLFLUSH	迫使挂起的文件输出写到磁盘上.....	40
10. FLGETDTA	返回磁盘传送地址.....	41
11. FLLOCK	对已打开文件的一个文件段上锁或解锁.....	41
12. FLNORM	验证一个文件名并把它转化为标准形式.....	42
13. FLPROMPT	从标准输入中返回一行, 可以使用提示信息.....	44
14. FLPUTDTA	设置磁盘传送地址 (DAT).....	45
15. FLREMOVOL	从给定的磁盘上删除卷标 (如果有的话).....	46
16. FLRETVOL	报告给定磁盘驱动器上的卷标.....	46
17. FLSETVOL	建立或修改给定磁盘上的卷标.....	47
18. HLCLOSE	释放二叉帮助文件的已有索引.....	48
19. HLDISP	从帮助文件读取一段帮助信息并显示在屏幕上, 供用户浏览.....	48
20. HLLOOKUP	从二叉帮助文件中读取一段帮助信息.....	49
21. HLOPEN	对一个二叉帮助文件建立索引.....	52

22. HLREAD	在视口中显示帮助信息, 供用户浏览	52
23. ISCALL	对软件中断调用中断服务例程进行模拟	55
24. ISCURPRC	返回或设置当前执行的进程	56
25. ISGETVEC	返回一个中断向量	56
26. ISINSTAL	安装一个中断服务例程(ISR)	57
27. ISPREP	预备一个ISR控制块	59
28. ISPUTVEC	设置一个中断向量	60
29. ISREMOVE	摘除一个驻留程序	60
30. ISRESERV	保留ISR所需的动态内存	61
31. ISRESEXT	中止一个程序但保持驻留	62
32. ISSENSE	探测一个已安装的中断服务例程(ISR)	62
33. IVCTRL	报告本程序中插入控制块的地址	63
34. IVDetect	探测已安装的插入函数, 即使它被部分遮蔽	64
35. IVDISABL	使一个插入函数失效	65
36. IVINSTAL	安装一个插入函数	66
37. IVSENSE	探测一个已安装的插入函数是否是可摘除的	68
38. IVVECS	设置或返回插入过滤程序所使用的中断向量	68
39. KBEQUIP	探测键盘环境	69
40. KBEXTEND	选用扩展的或一般的BIOS键盘服务	70
41. KBFLUSH	废弃所有在键盘缓冲区等待的按键	71
42. KBGETKEY	等待读入下一个按键	72
43. KBKCFLSH	通过键控制函数废弃所有的等待按键	72
44. KBPLACE	在键盘缓冲区中放置一个按键	73
45. KBPOLL	通过一个键控制函数查看下一个等待按键	74
46. KBQUERY	从标准IBM控制台读取用户的响应	74
47. KBQUEUE	报告键盘缓冲区总容量及剩余容量	76
48. KBREADY	检查下一个等待按键	76
49. KBSCANOF	返回一个字符的键码	77
50. KBSET	设置移位键的当前状态	77
51. KBSTATUS	报告移位键的当前状态	78
52. KBSTUFF	强行将一个字符串送入BIOS超前键入缓冲区	79
53. KBWAIT	等待并通过键控制函数读取下一个按键	81
54. MMCTRL	读取DOS内存控制块	82
55. MMFIRST	报告第一个内存块的地址	83
56. MMSIZE	报告一个程序的尺寸	83
57. MNCREATE	建立一个包含选单信息的选单结构和窗口	84
58. MNDSPY	在同尺寸视口中显示一个选单	84
59. MNDSTROY	从屏幕上取消一个选单, 废弃其数据结构	85
60. MNHILITE	移动或取消选单亮条及选项说明	86
61. MNITEM	加入、修改或删除一个选项	87
62. MNITMKEY	向选单加入一个选项, 为它分配选择字符	89
63. MNKEY	加入、修改或取消一个选单的键分配	90
64. MNLITEM	加入、修改或取消一个Lotus形式的选项	91

65. MNLITKEY	加入一个Lotus形式的选项, 为它分配选择字符	93
66. MNLREAD	通过Lotus形式的选单读入一个用户响应	94
67. MNMOUSE	加入、修改或删除一个选单认可的鼠标器事件	96
68. MNMSTYLE	设立一个标准选单鼠标器风格	97
69. MNREAD	读取来自选单的用户响应	98
70. MNVDISP	在视口中显示一个虚拟选单	99
71. MOAVOID	在指定区域中隐蔽鼠标	101
72. MOBUTTON	报告鼠标器按钮的按下/释放历史	101
73. MOCHECK	检查最近发生的鼠标器事件	102
74. MOCURMOV	移动鼠标	104
75. MOEQUIP	感知鼠标器驱动程序的存在	105
76. MOGATE	鼠标器驱动程序的人口	105
77. MOGETMOV	报告自上次查询以来物理鼠标器的移动	107
78. MOGRAPH	设置鼠标器图形方式光标	107
79. MOHANDLR	安装或摘除中断处理程序	109
80. MOHARD	设置鼠标器硬件字符方式光标	110
81. MOHIDE	隐藏或显示鼠标	111
82. MOJUMP	设置鼠标器加速阈值	111
83. MOLITPEN	使鼠标器光笔模拟有效或失效	112
84. MOPRECLK	安装或摘除MOCHECK所有的内部例程	112
85. MORANGE	设置鼠标器范围界限	113
86. MORESET	重置鼠标器驱动程序	113
87. MOSOFT	设置鼠标器软件字符方式光标	114
88. MOSPEED	设置鼠标器灵敏度	115
89. MOSTAT	报告鼠标器位置和按钮状态	115
90. PRCANCEL	删除假脱机打印队列中一个或全部文件	116
91. PRCHAR	通过BIOS向打印机发送一个字符	117
92. PRERROR	返回解释来自PR(打印机)函数错误代码的字符串	117
93. PRGETQ	报告假脱机打印队列中的一个文件名	118
94. PRINT	通过BIOS初始化一个打印口	119
95. PRINSTLD	检查驻留式假脱机打印系统PRINT是否已安装	120
96. PRSPOOL	将一个文件提交给假脱机打印系统	120
97. PRSTATUS	通过BIOS报告打印机的状态	121
98. SCAPAGE	显示(激活)一个显示页	121
99. SCATTRIB	用指定的显示属性显示一个字符的拷贝	122
100. SCBLINK	选择前景闪烁或背景亮度	123
101. SCBORDER	设置当前显示屏幕的边界颜色	123
102. SCBOX	用图形字符在屏幕上画一个方框	124
103. SCCHGDEV	切换至彩色或单色显示	125
104. SCCLRMSG	清除屏幕上的消息	126
105. SCCURSET	移动当前显示页上的光标	127
106. SCCURST	返回当前显示页上的光标位置和尺寸	127
107. SCEQUIP	探测显示硬件环境	128

108. SCGETVID	记录整个显示状态	129
109. SCMODE	返回屏幕的显示方式	130
110. SCMODE4	设置方式4色板和背景颜色	130
111. SCNEWDEV	选择和重设显示设备及设置字符行数	131
112. SCPAGE	设置当前显示页	131
113. SCPAGES	返回显示页的数目	132
114. SCPAL1	定义一个EGA、VGA或MCGA色板颜色	132
115. SCPALETTE	定义EGA、VGA或MCGA颜色的整个色板	133
116. SCPCLR	清除当前显示页	134
117. SCPGCUR	设置当前页的光标尺寸	135
118. SCREAD	从屏幕读取一个显示字符及其属性	136
119. SCRESTPG	恢复一个显示页	136
120. SCROWS	返回屏幕的字符行数	137
121. SCSAVEPG	保存一个显示页	137
122. SCSETVID	恢复整个显示状态	138
123. SCTTYWIN	以TTY方式向矩形区域写入一个字符	139
124. SCTTYWRT	以TTY方式向屏幕写一个字符	140
125. SCWRAP	以TTY方式向一个矩形中写入一个字符串, 带有整字换行	141
126. SCWRITE	在屏幕上显示一个字符的多个拷贝	142
127. STPCVT	常用字符串转换	143
128. STPEXPAN	将tab字符转换为空格	144
129. STPJUS	在域中将一个字符串左右对齐或居中	145
130. STPTABFY	用tab字符替换空格	145
131. STPXLT	用翻译表翻译一个字符串	146
132. STSCHIND	查找字符串中的一个字符, 返回它的位置	147
133. UTANSI	探测、关闭或重新开放ANSI.SYS	147
134. UTCHKNIL	报告无效指针赋值, 使程序夭折	148
135. UTCRIT	取得DOS临界段标帜的地址	149
136. UTCTLBRK	设置或返回Ctrl-Break检查的状态	149
137. UTDOSRDY	报告DOS服务是否可用	150
138. UTGETCLK	报告自午夜以来BIOS计时脉冲的个数	150
139. UTINTFLG	开放或关闭硬件中断	151
140. UTMODEL	报告IBM型号和子型号及BIOS版本	151
141. UTMVMEM	不受限制地从内存或向内存任何位置拷贝数据	153
142. UTNORM	使一个指针具有最小的偏移值	154
143. UTNULCHK	探测无效的指针赋值	155
144. UTOFF	返回一个地址的偏移部分	156
145. UTPEEKB	从任意地址读取一个字符	157
146. UTPEEKN	从任意地址读取多个字节的数据	157
147. UTPEEKW	从任意地址读取一个字	157
148. UTPLONG	将一个指针转换为指向20位物理地址的指针	158
149. UTPOKEB	在任意地址存放一个字节的数据	158
150. UTPOKEN	在任意地址存放多个字节的数据	159

151. UTPOKEW	在任意位置写入一个字的数据	159
152. UTSAFPCPY	以确保不跨越段界的方式拷贝数据	160
153. UTSEG	返回任意地址的段部分	161
154. UTSLEEP	暂停处理直至经过几个计时脉冲之后	161
155. UTSPKR	打开或关闭扬声器	161
156. UTSQZSCN	压缩一个屏幕图象	162
157. UTTIM2TK	将时间转换为计时脉冲计数	163
158. UTTK2TIM	将计时脉冲计数转换为24小时制时间	164
159. UTTOFAR	用段和偏移构造一个双字指针	165
160. UTTOFARU	用一个段和偏移构造一个泛双字指针	166
161. UTUNSQZ	还原一个压缩屏幕图象	166
162. VIATRECT	改变屏幕上一个矩形的属性	167
163. VIDSPMSG	显示一条消息	168
164. VIHORIZ	在当前显示页上水平滚动正文列	168
165. VIPTR	将屏幕位置转换成内存地址	169
166. VIRDIRECT	读取屏幕上一个矩形区域的内容	170
167. VIRDSECT	将屏幕的一个区域读入内存中更大的矩形区域	170
168. VISCROLL	垂直滚动当前显示页上的正文行	171
169. VIWRRECT	向当前显示页上一个矩形区域写入数据	172
170. VIWRSECT	显示矩形缓冲区中的一个矩形段	173
171. WNATRBLK	修改窗口中一个矩形块的属性	174
172. WNATRSTR	改变窗口中一片连续位置的属性	174
173. WNATTR	改变当前窗口的属性	176
174. WNCHGEVN	加入或修改WNREAD认可的一个用户响应	176
175. WNCURMOV	建立一个窗口结构	177
176. WNCURPOS	移动当前窗口的光标	178
177. WNCURPOS	返回当前窗口的光标位置	178
178. WNCURSOR	激活一个窗口光标	179
179. WNDSPRAY	在同尺寸视口中显示一个窗口	179
180. WNDSTROY	废弃一个窗口结构	182
181. WNERORR	记录窗口或选单的系统错误	182
182. WNFIELD	对窗口中的一个域进行编辑	183
183. WNGETOPT	读取窗口信息项或状态	185
184. WNHORIZ	水平滚动当前窗口	187
185. WNINITEV	为WNREAD安装缺省的窗口事件	188
186. WNORIGIN	在视口中移动窗口	188
187. WNPRINTF	向当前窗口写入一个格式化的字符串	189
188. WNQUERY	返回经窗口得到的来自用户的字符串	189
189. WNRDBUF	读取当前窗口中一片连续位置的内容	191
190. WNREAD	允许用户在虚拟窗口中浏览	192
191. WNREDRAW	重现显示在当前显示页上的全部窗口	194
192. WNREMEVN	删去WNREAD接受的一个用户响应	195
193. WNREMOVE	从屏幕上取消一个窗口	196
194. WNREVUPD	用显示的数据更新已保存的窗口图象	196
195. WNSCRBLK	在窗口中以任意方向滚动一个矩形区域	197

196. WNSCRLBR	向窗口加入一个滚动箭头	198
197. WNSCROLL	垂直滚动当前窗口	199
198. WNSELECT	选择用于I/O的窗口	199
199. WNSETBUF	为WNPRINTF分配内部缓冲区	200
200. WNSETOPT	设置窗口控制项	201
201. WNSHOBLK	在视口间隙中显示一个窗口数据块	202
202. WNUPDATE	将挂起的输出写入窗口	203
203. WNVDISP	在视口中显示一个虚拟窗口	204
204. WNWRAP	以TTY方式向当前窗口写入字符串, 带有整字换行	205
205. WNWRBUF	向当前视口的一片连续位置写入字符	206
206. WNWRRECT	写入窗口中的一个矩形区域	207
207. WNWRSTR	以TTY方式向当前窗口写入一个字符串	208
208. WNWRSTRN	以TTY方式向窗口写入一个字符串, 带有任选项	209
209. WNWRTTY	以TTY方式向当前窗口写入一个字符	210
210. WNZAPEVN	删除WNREAD认可的窗口事件表	210

第三部分 附录

附录 A 键码和符号	212
附录 B 窗口/选单/帮助系统错误	215
附录 C 全局变量	216
附录 D 显示方式	218
附录 E 屏幕/窗口对应函数	219
附录 F 安装及使用说明	220

第一部分 函数分类及概述

1、字符串函数 (ST)

所有 Turbo C TOOLS 字符串函数处理并返回标准的 C 字符串, 即 char 类型的数组, 字符串的末尾由一个 NUL 字符('\0') 标记。如有必要, 字符串函数使用一个名为 tsize 的参数, 它是目标字符串的容量。目标字符数组必须至少为 tsize 字节长。由于尾随的 NUL 的缘故, 作为结果的字符串的最大长度比 tsize 小 1 个字节。

当使用一个整数值指示字符串中的某个位置时, 0 总是指向字符串的第一个字符, 1 指示第二个, 等等。

当提及空字符串时, 我们指的是长度为零的字符串(即 ""), 请不要与 NUL 字符('\0')混淆, 也不要与指向数据地址 0 的 NULL 指针混淆。(在本书的所有源代码中, NULL 指针写为 "NIL".)

(一) 字符串函数的种类

1. 对字符串的一部分进行填充

STPJUST 在指定的域尺寸中对一个字符串居中或对齐(左对齐或右对齐)。

2. 查找

STSCHIND 在一个字符串中查找一个字符的第一次出现, 返回它的位置。

3. 字符转换

STPXlate 用一个翻译表翻译一个字符串的每一个字符。

STPCVT 执行一种或几种类型的转换: 转换为大写或小写字母, 删除开头、尾随或所有的空白, 将多个连续空白压缩为一个空格, 删除控制字符等。该函数可以任选地使括在引号 (") 或省略号 (') 中的子串不受影响。

4. 处理 tab 字符

STPEXPAN 将 tab 字符扩展为空格。

STPTABFY 将连续多个空格压缩为 tab 字符。

2、屏幕函数 (SC 和 VI)

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下。当需要最快的速度时, 用户还可直接访问显示内存。

(一) 屏幕操作函数的种类

1. 读取屏幕方式信息

SCEQUIP 感知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。

2. 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。

3. 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。

4. 控制/读取光标形状和位置

SCCURST 报告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。

5. 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消息。

VIScroll 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。

6. 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偶对的拷贝。

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。

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

VIATRECT 改变一个矩形区域的属性, 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。

8. 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 正象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。

9. 色板支持

SCPALETT 定义由EGA、VGA或MCGA显示的整个颜色色板。

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：前景闪烁或背景亮度。

10.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)。

11.保存和恢复整个显示状态

让应用程序在结束之后不打扰显示环境，这是一种礼貌的行为，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这个状态。

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、 VIRDIRECT 或 SCSAVEPG	SCATTRIB、 SCWRITE、 VIWRRECT 或 SCRESTPG
屏幕上的属性	SCREAD、 VIRDIRECT 或 SCSAVEPG	SCATTRIB、 VIWRRECT、 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。

12.强制快速屏幕访问

直接显示(VI)函数探测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;
```

3、窗口函数 (WN)

(一) 窗口功能概述

Turbo C TOOLS窗口函数提供了各种字符窗口应用程序所必需的功能和灵活性。系统自动处理许多高级效果,用户为标准应用程序建立简单的窗口也非常容易。每个窗口“出生”时带有一组缺省的特性,用户可以安全地省略许多复杂设置。

下面是Turbo C TOOLS窗口所具有的基本特性:

窗口在显示时可带有多种边界和标题或根本不带有边界。

窗户是虚拟的,也就是说,它可以超出它的显示边界的尺寸,用户可以借助鼠标器或键盘浏览一个更大的数据区。不论窗口是否是当前显示的,它们都具有一组灵活的I/O函数用于向窗口写或从窗口读。除了用于窗口的printf()之外,这些函数还可以控制颜色和属性。

窗口可以被删除,屏幕的原内容得到恢复。然而,您也可以指定窗口为不可删除的,以便于节省本来用来记录原屏幕数据的内存空间。窗口可以相互迭盖,系统自动处理迭盖窗口中的输入和输出操作。

Turbo C的字符窗口可以与Turbo C TOOLS窗口联合使用。

窗口的显示和处理可以在多个显示页上进行。

一个窗口的输出可以被延迟,这样挂起的改变内容可以在一瞬间出现。

每个窗口可以具有自己独立的可控制的光标。

(二) 窗口函数的种类

1.建立和释放窗口结构

WNCREATE 分配并建立一个BWINDOW结构,包括窗口数据区的初始拷贝。

WNDSTROY 释放一个BWINDOW结构以及有关的内部数据结构(不改变屏幕)。

2.显示和删除窗口

WNDSPLAY 用可选的边界显示一个窗口并选择这个窗口为“当前窗口”,用于以后的I/O请求。它还选定该窗口(对于它的显示页上的其它窗口来说)具有活动光标。如果相关的话,Turbo C字符窗口被设置为与新窗口的视口相匹配。

WNREMOVE 从显示页上取消一个窗口,恢复屏幕的原内容,窗口不再与它的显示位置相关联。如果窗口具有活动光标,光标即被关闭。如果相关的话,Turbo C字符窗口被置为整个显示屏幕,光标被关闭。

WNREDRAW 重现显示页上的所有窗口并恢复光标(如果其它进程影响了屏幕,这个功能是很有用的)。

3.虚拟窗口

WNVDISP 在视口中显示一个窗口,该视口可以比窗口数据区小。这个函数与WNDSPLAY很相似。

WNORIGIN 在视口中移动窗口,显示数据区的各个部分。

WNSHOBK 在视口中移动窗口。如果可能的话,强制显示数据区中需要的那一部分。

WNREAD 允许用户借助于鼠标器或键盘在视口中移动窗口。如果窗口未显示,WNREAD即在视口中显示这个窗口,事后取消该窗口。

4.窗口输出

WNPRINTF 以printf()的方式格式化一个字符串,将该字符串写入当前窗口。

WNWRAP 以TIY方式将一个字符串写入当前视口,带有整字换行。

WNWRTTY 以TTY方式向当前窗口写入一个字符。
 WNWRSTR 以TTY方式向当前窗口写入一个字符串。
 WNWRSTRN 以TTY方式向任意窗口写入一个字符串, 带有几个可选项。
 WNWRBUF 将缓冲区写入当前窗口。它不带有整字换行, 不滚动屏幕, 对字符不做特殊处理。
 WNWRRECT 向窗口中一个矩形区域做写入操作。

5.窗口输入

WNQUERY 返回来自用户的输入, 在当前窗口回映所有的按键。
 WNRDBUF 从当前窗口 (不是从用户)读取数据, 写入到缓冲区中。
 WNFIELD 允许用户编辑窗口中的一个域。要想进一步了解有关的内容, 请参见“域编辑 (ED)”一章。

6.滚动和清除

任意一个滚动函数都可以清除一个区域。

WNSCROLL 垂直滚动当前窗口的数据区。
 WNHORIZ 水平滚动当前窗口的数据区。
 WNSCRBLK 垂直或水平滚动窗口中的一个矩形块。

7.控制属性

WNATTR 修改当前窗口数据区的属性(它不修改窗口的缺省属性, 而WNSETOPT可以修改)。
 WNATRBLK 修改窗口中一个矩形块的属性。
 WNATRSTR 修改窗口中一系列连续位置的属性。

8.控制光标

WNCURMOV 移动当前窗口的光标。
 WNCURPOS 返回当前窗口光标相对于该窗口的位置。
 WNCURSOR 选择一个窗口 (对于显示页上所有的窗口来说)使之具有活动光标。

9.控制窗口任选项

WNSELECT 选择一个窗口作为“当前窗口”, 用于以后的I/O请求。如果相关, 该窗口也被作为Turbo C字符窗口而建立。
 WNUPDATE 在可能的情况下完成任何挂起的输出请求。
 WNSETOPT 设置窗口任选项, 如下所示:
 光标外观;
 窗口数据区的缺省属性;
 窗口输出是“延迟的”还是“立即的”;
 窗口是否是“可删除的”;
 虚拟窗口是否自动移动, 使窗口光标在视口中保持可见 (“自动跟踪”);
 视口间隙。
 WNSETOPT也可以迫使窗口数据结构在使用之前被分配。
 WNGETOPT 返回窗口任选项的值或窗口状态的某些参数, 如下所示:
 尺寸、属性或边界;
 位于窗口显示位置的视口及被遮盖区域的尺寸。
 光标尺寸或位置 (相对于窗口数据区或屏幕);
 窗口是否是“延迟的”、“可删除的”, 窗口是否被指定为“当前的”;
 虚拟窗口是否移动使窗口光标在视口中保持可见 (“自动跟踪”);
 视口间隙;
 窗口数据区的任意部分是否被另一个窗口遮盖, 向窗口的输出是否由于这个原

因而被延迟。

WNSETBUF 强行分配 **WNPRINTF** 所使用的缓冲区。

WNEROR 在全局变量 **b_wnerr** 中记录窗口的错误代码。

10. 使用 Turbo C 的字符窗口

WNREVUPD 用当前显示在屏幕上的数据更新当前窗口数据区已保存的拷贝, 这对于使 Turbo C TOOLS 窗口与通过 Turbo C 字符窗口或其它手段显示的数据相同步是很有用的。

11. 为通过 **WNREAD** 进行窗口输入做准备

WNSCRLBR 设置一个窗口的滚动箭头, 这些滚动箭头可以指向水平方向(左右移动)或垂直方向(上下移动)或指向水平垂直两个方向。

WNINITEV 建立一个 **WNREAD** 所感知的键盘和鼠标器事件的缺省表。

WNCHGEVN 加入或修改事件表中的一个事件。

WNREMEVN 从事件表中删除一项。

WNZAPEVN 删除整个事件表。

12. 使用高级窗口特性

13. 显示和更新窗口

在某些应用程序中, 向窗口填充信息的过程可能是比较缓慢的, 用户于是更希望使输出延迟, 直到某个时刻。下面的语句可以做到这一点:

```
wnsetopt(pwindow, WN_DELAYED, 1);
```

下面的语句使窗口成为“立即的”;

```
wnsetopt(pwindow, WN_DELAYED, 0);
```

用 **WNUPDATE** 可以迫使任何挂起的输出送往屏幕。

14. 虚拟窗口

用户可以在比窗口数据区小的视口中显示任意的窗口(视口是显示屏幕上的一个矩形区域, 用于显示窗口数据区的一部分)。要在可能比窗口小的视口中显示一个窗口, 需使用 **WNVDISP** (**WNSPLAY** 与 **WNVDISP** 相同, 不过 **WNSPLAY** 使用与窗口同尺寸的视口)。

按照缺省, 窗口数据区的显示部分是自动控制的, 窗口自动移动, 使得光标位置保持可见(即使窗口光标被关闭)。这个特性称为自动光标跟踪或自动跟踪。下面的语句可以使这个特性失效:

```
wnsetopt(pwin, WN_CUR_TRACK, 0);
```

在关闭之后若想重建自动跟踪, 可使用下面的语句:

```
wnsetopt(pwin, WN_CUR_TRACK, 1);
```

通过使用视口间隙用户还可以提高自动跟踪的性能。间隙使得窗口在移动时让光标位置与视口边沿保持一个间隙。用户可以通过 **WNSETOPT** 分别控制四个边沿的间隙。所有四个边沿间隙的初始值是 0, 也就是说间隙不起作用。当自动跟踪关闭时, 间隙只影响 **WNSHOBK**、**MNREAD** 和 **MNLREAD**。

15. 不可删除的窗口

通常窗口是可删除的, 即在任何时候都可以利用 **WNREMOVE** 从屏幕上取消某个窗口, 恢复屏幕的原内容。每个 **BWINDOW** 结构包含着一些信息, 这些信息保存着屏幕的原内容。

然而, 为了节省内存空间和时间, 用下面的语句可以随时指定一个窗口为不可删除窗口:

```
wnsetopt(pwindow, WN_REMOVABLE, 0);
```

这将释放用于记录原屏幕图象的空间。为节省大部分的内存, 可以在窗口显示之前发出这个函数调用。

16. 控制内存分配

窗口函数利用标准的内存分配函数 **malloc()** 和 **calloc()** 来建立数据结构和缓冲区, 这对于中止并

驻留的程序来说是比较危险的。为此,用户也许希望确保在程序中止并驻留之前使窗口数据结构和缓冲区得到分配。为了做到这一点,在程序中止之前可以对每个窗口执行下面的语句

```
wsetopt(pwindow,WN_PREV_ALLOC,1);
```

如果使用 **WNPRINTF**, 在程序中止之前应激活 **WNSSETBUF**; 如果使用 **WNREAD**, 在程序中止之前需激活 **WNINITEV** 和 **WNSCRLBR**。这些步骤将分配必需的信息项, 因而 **WNSDISPLAY**、**WNPRINTF** 和 **WNREAD** 就不必这样做了。

如果使用了 Turbo C TOOLS 的选单或帮助系统, 请参见“选单函数(MN)”和“帮助系统(HL)”下的“控制内存分配”一节。

4、选单函数 (MN)

(一) 选单函数功能概述

这一章介绍了 Turbo C TOOLS/6.0 最为重要的特性之一: 具有亮条移动和鼠标器支持的选单。系统定义了一个按键动作的标准集, 用户可以重定义、删除或暂时使任意的标准按键分配失效, 也可以定义随意选择一个选项的键。

(二) 选单函数的种类

1. 建立、显示和释放选单

MNCREATE 分配并建立一个 **BMENU** 结构, 包括用于容纳选单的 Turbo C TOOLS 的窗口。

MNSDISPLAY 用可选的边界和标题在屏幕上显示一个选单。如果相关的话, 它还将 Turbo C 字符窗口设置为与选单的视口相匹配。

MNVDISP 在视口中显示一个虚拟选单 (带有可选的边界和标题, 它们可以比选单小。) 如果相关的话, 它还将 Turbo C 字符窗口设置为与选单的视口相匹配。

MNDSTROY 从屏幕上取消一个选单, 释放它的所有内部数据结构。如果选单具有活动光标, 使该光标失效, Turbo C 字符窗口 (如果相关) 被置为全屏。

2. 定义标准选项和按键

MNITEM 向选单中加入一个选项或删除一个已有的选项。

MNKEY 定义一个按键的效果或修改、删除一个已有的按键定义。

MNITMKEY 向选单加入一个选项及一个或多个 ASCII 字符, 这些字符用于将亮条移到这个选项上。

3. 定义 Lotus 形式的选项

MNLITEM 将一个选项连同说明字符串加入到选单中, 在该选项被显亮的同时出现说明字符串。MNLITEM 还可以修改或删除这样的选项。

MNLITKEY 与 MNLITEM 表现相同, 但它还可以定义一个或多个 ASCII 字符, 这些字符用于把亮条移到某选项上。

4. 使用鼠标器的准备工作

MNMSTYLE 为选单选择一个鼠标器的使用方式。

MNMOUSE 向 MNREAD 和 MNLREAD 认可的鼠标器事件表中加入、修改或删除一个选项。

5. 读取用户的响应

MNREAD 通过选单读取一个用户响应。

MNLREAD 通过 Lotus 形式的选单读取用户的响应。

6. 亮条操作

MNHILITE 移动或取消亮条, 可带有或不带有选项的 Lotus 形式的说明字符串。