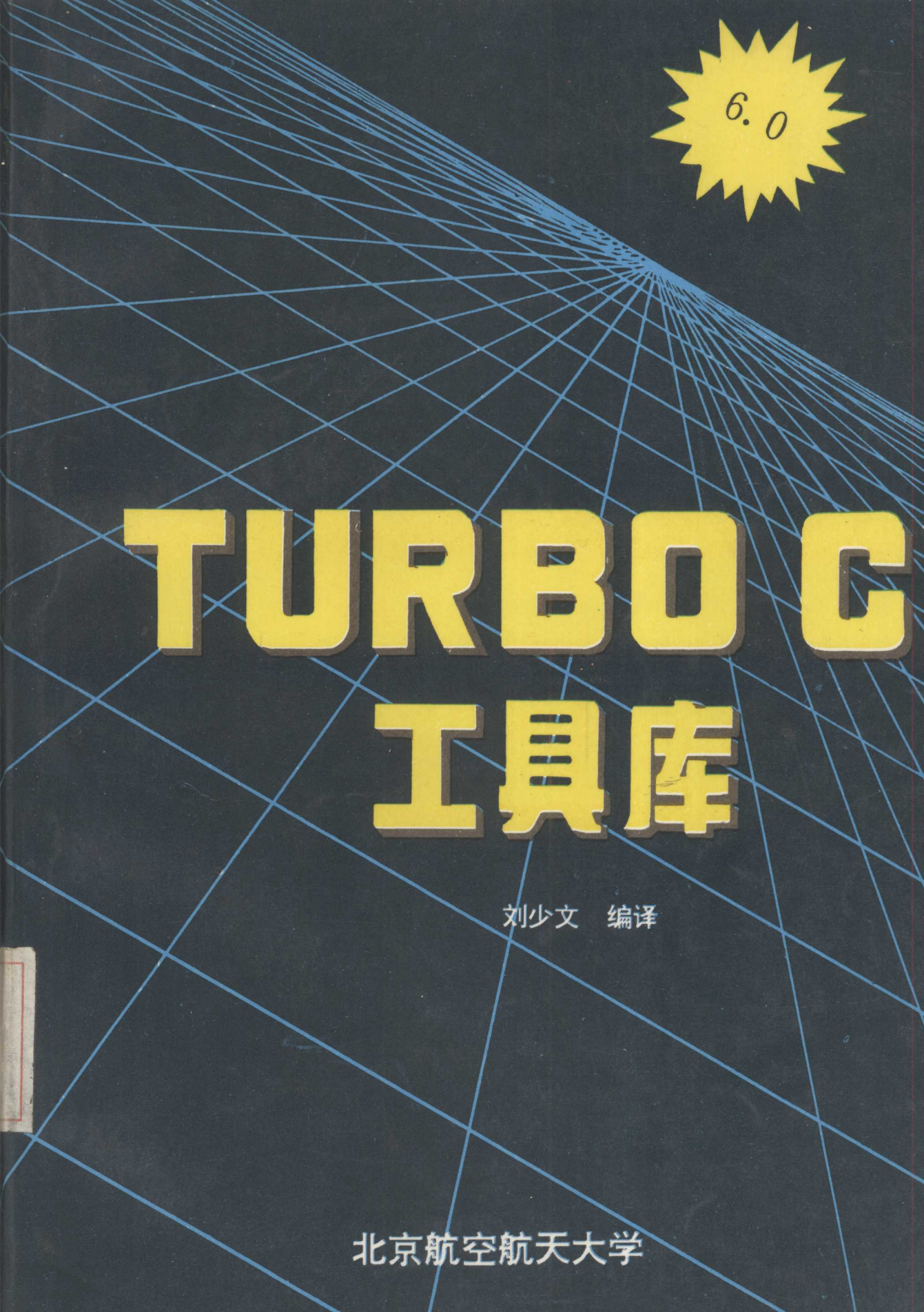
6.0
TURBO C
工具库
刘少文 编译
北京航空航天大学

TURBO C 工具库

刘少文　编译

北京航空航天大学

译 者 前 言

Turbo C TOOLS 6.0 是美国 Blaise Computing Inc. 在 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，那么发行盘上的源文件已经为您大开了方便之门。

本书的翻译和出版得到了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。黄步霞、高银素两位小姐为本书的排版做了一定的工作，戎杰、王荣也付出了劳动。我要特别感谢王丽红小姐，她以特有的耐心和细心自始至终参与了本书的排版和整理工作。另外，于滨、陈铁刚二先生对我的工作给予了各方面的支持。我非常感激我的朋友邱小红，没有她的鼓励和帮助，本书就不可能以现在的面貌出现在读者面前。

译者

1990 年 12 月 29 日

目 录

第一部分 函数分类及概述

第二部分 函数参考说明

第三部分　附录

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 tarsize 的参数，它是目标字符串的容量。目标字符数组必须至少为 tarsize 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 tarsize 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"。）

字符串函数的种类

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）。

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。

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。

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。

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。

屏幕操作函数的种类

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。

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

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。

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。

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。

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。

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。

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。

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

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;
```

窗口函数（WN）

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。

窗口函数的种类

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。）

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。）

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。

WNSHOBLK 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。

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 TTY 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。

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。

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。

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。

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。

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。

WNERROR 在全局变量 b_wnerr 中记录窗口的错误代码。

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。

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。

使用高级窗口特性

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。

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。（ WNDSPLAY 与 WNVDISP 相同，不过 WNDSPLAY 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 WNSHOBLK、MNREAD 和 MNLREAD。

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。

控制内存分配