

高级C环境下窗口系统 开发与应用

梁满贵 编译
梁满昌

中国科学院希望高级电脑技术公司

高级C环境下窗口系统开发与应用

梁满贵 编 译
梁满昌

中国科学院希望高级电脑技术公司

一九九〇年十月

版权所有
翻印必究

★北京市新闻出版局

准印证号：891446

★订购单位：北京8721信箱资料室

★邮 码：100080

★电 话：2562329

★乘 车：320、302、332、路车

至海淀黄庄下车

★办公地点：希望公司大楼101房间

前 言

随着计算机技术的不断发展,人们对开发软件的速度,以及软件产品的效率和人机界面的“友好”性有了新的要求。为满足这种需求已推出不少工具性的软件,特别是各种窗口软件,已成为应用软件程序员编制高质量软件的有力工具。

典型的窗口软件有microsoft窗口系统和IBM Top View窗口系统。这些窗口系统是作为操作系统的一种扩充,用户程序可通过系统功能调用完成窗口操作。其优点是通用性强,可支持各种程序语言。然而通用性是以效率和灵活性为代价的。随着用户对软件质量的要求进一步提高,人们希望为某种语言配专用窗口系统。

C语言窗口系统(Windows for C,简称为WFC)就是一种新的专为C语言而设计的窗口软件。WFC不是操作系统的一种扩充,而是C语言程序的一部分。同以往的窗口系统相比具有以下优点:(1)使用灵活。窗口系统由C语言的一些库函数组成成分可方便地对WFC进行修改、裁剪,以便在空间(存贮量)和时间(CPU时间)上达到最后合理的安排;(2)窗口系统不驻留内存,所以使用窗口的进程和不使用窗口的进程之间无任何干扰;(3)窗口系统和C语言用户程序结合成一个执行文件,可适用于多用户操作系统。基于这些优点,我们翻译了《高级C环境下窗口系统开发与应用》这本书。它不仅可为程序员使用WFC提供方便,而且可为自己设计窗口软件提供有用的设计思想。

由于时间仓促,加之译者水平有限,一定会有不少错误,希望读者指正。

最后对翻译和校对过程中给予大力协助的陈复兴,杜绍娟,李震三位同志表示感谢。

高级C环境下窗系统开发与应用

编译 梁满贵 梁满昌

编辑 玉 兰 人 华

北京市新闻出版局

准印证号: 891446

订 购: 北京8721信箱资料部

邮 码: 100080

电 话: 2562329

乘 车: 320.302.332路车至海淀黄庄下车

办公地点: 希望公司大楼101房间

1990年10月第一次印刷

目 录

用户注意事项.....	(1)
关于本手册.....	(1)
前版本用户须知.....	(2)
第一章 WFC概述.....	(8)
§ 1.1 WFC的特性.....	(8)
§ 1.2 演示程序.....	(10)
第二章 WFC入门.....	(12)
§ 2.1 磁盘内容.....	(12)
§ 2.2 程序的编译和连接.....	(12)
§ 2.3 使用物理属性.....	(14)
§ 2.4 一些 BM兼容机与TOPView和MS窗口系统不兼容.....	(14)
第三章 窗口系统的基本内容.....	(15)
§ 3.1 直接显示和内存文件显示.....	(15)
§ 3.2 第一个窗口系统程序: "Hello, world".....	(15)
§ 3.3 编窗口程序的六个步骤.....	(16)
§ 3.4 进入VCS调试系统.....	(18)
§ 3.5 WFC的常用缩写.....	(18)
§ 3.6 改变缺省值和使用选择项.....	(19)
§ 3.7 控制输出形式: 属性.....	(22)
§ 3.8 显示和清除窗口.....	(25)
§ 3.9 控制输出位置.....	(25)
§ 3.10 写入窗口.....	(28)
§ 3.11 读窗口中的内容.....	(32)
§ 3.12 读键盘.....	(32)
§ 3.13 清除屏幕和窗口.....	(34)
§ 3.14 在全屏幕上用窗口函数.....	(34)
§ 3.15 打印"Hello, world"程序的另一种编法.....	(35)
§ 3.16 窗口系统的实例.....	(36)
第四章 WFC的示范程序.....	(38)
第五章 VCS调试和错误处理系统.....	(44)
§ 5.1 概述.....	(44)
§ 5.2 使用VCS调试系统.....	(45)
§ 5.3 在自己的函数中实现VCS错误系统程.....	(46)
§ 5.4 调试状态对速度和内存的影响.....	(48)

§ 5.5	内存完整性检查是怎样进行的	(48)
§ 5.6	全局错误码和错误信息	(49)
§ 5.7	编写和装错误处理器	(49)
§ 5.8	模型检查	(51)
第六章	控制颜色: 逻辑和物理属性	(52)
§ 6.1	逻辑属性系统	(52)
§ 6.2	使用物理属性	(55)
§ 6.3	确定和改变视频方式	(58)
第七章	建立和显示内存文件	(60)
§ 7.1	内存文件概述	(60)
§ 7.2	建立内存文件	(60)
§ 7.3	把文本写到内存文件	(62)
§ 7.4	在窗口中察看内存文件	(65)
§ 7.5	修改内存文件	(66)
§ 7.6	向磁盘上写内存文件: mf-wr ()	(68)
§ 7.7	清除和释放一个内存文件	(69)
§ 7.8	关于内存结构和内存管理的技术资料	(69)
第八章	帮助文件、菜单和后台缓冲器	(73)
§ 8.1	使用一个上托帮助文件	(73)
§ 8.2	建立并显示上托菜单	(74)
§ 8.3	管理一个后台缓冲器	(77)
第九章	高级课题应用和实例	(79)
§ 9.1	多个文件的窗口观察	(79)
§ 9.2	向窗口输入输出信息	(81)
§ 9.3	移动存储及恢复窗口图象	(84)
§ 9.4	高亮度显示和改变属性	(86)
§ 9.5	用窗口来格式化打印文本	(86)
§ 9.6	绘图函数	(87)
§ 9.7	使用和修改系统的全局变量	(87)
§ 9.8	字符串应用	(88)
§ 9.9	其它应用	(88)
§ 9.10	开发应用	(89)
§ 9.11	WINDOW结构的技术资料	(90)
第十章	改变系统信息	(94)
§ 10.1	系统提示信息存放在哪里	(94)
§ 10.2	访问系统信息	(94)
§ 10.3	改变系统信息	(95)
§ 10.4	减小可执行文件的大小	(95)
§ 10.5	编译器的内存限制	(96)

第十一章 兼容性问题	(97)
§ 11.1 MS窗口与IBM TOPView的兼容性.....	(97)
§ 11.2 利用别的显示适配器.....	(99)
§ 11.3 鼠标器支持.....	(100)
§ 11.4 与可重入性应用的兼容性.....	(102)
§ 11.5 与UNIX、VMS以及其它终端系统的接口程序.....	(102)
附录A: 嵌入文件	(105)
附录B: WFC库函数	(127)
附录C: 定义和缩写	(127)
附录D: 错误码	(198)
附录E: 常见的编程错误	(204)

用户注意事项

过去我们修改了代表窗口系统和内存文件基本内容的数据结构元素，以后我们还将这样进行下去。用户应采取适当措施把版本更新带来的困难减小到最低限度。

我们郑重声明不要初始化静态和外部的系统数据结构。系统提供的函数和宏允许对需要访问的系统结构元素赋值。如果使用了这些系统函数和宏，对于以后结构元素的变化，用户只要重新编译原程序，而不需作其它任何改动。如果由于实际需要而违背了这条规则，可以用下面所提供的办法把将来可能产生的问题减小到最低限度。

- 开发和使用你自己按结构元素符号名编写的函数和宏，而不要对外部的或静态的系统结构初始化。
- 使用了结构初始化，要把它们归到一个嵌入文件中，并用 `#include` 语句把它加在 `main ( )` 之前。
- 要避免在子程序中初始化静态的系统结构。

对系统结构的任何修改都会给用户带来诸多不便，因此我们不会轻易这样做。然而计算机领域在日新月异地变化，VCS窗口系统也必须做相应的修改，以便在竞争中处于不败之地。

内部子程序

VCS窗口系统包含一部分不许用户使用的内部子程序。窗口系统的内部子程序名称都是以前缀下划线开头。禁止使用带这样前缀的名称，以免与系统文件发生冲突。

关于本手册

本手册介绍如何使用WFC版本4.1。

WFC的新用户注意事项：

手册的前四章介绍WFC的基本内容，以后几章在使用具有更大灵活性和控制能力的选项及特点方面进行了探讨。

WFC的老用户注意事项：

在“对前版本用户的重要说明”一节介绍了WFC版本4.1对WFC老版本程序有什么影响，并说明了如何查找版本4.1新特点。

WFD用户注意事项：

第三章讲述了如何使用WFC的基本内容，并对如何使用VCS调试系统和错误处理系统（第五章），如何用物理和逻辑属性控制颜色（第六章）和如何改变系统提示信息（第十章）也有很大的参考价值。

下面粗略地总结一下手册里各章和附录的内容：

第一章对WFC的各特点作了扼要说明，并介绍了各个演示程序。

第二章介绍启动WFC的方法。介绍了WFC系统盘内容和把WFC代码嵌入你的程序的方法。

第三章扼要说明WFC的基本内容，它一开始就介绍了使用窗口系统的六个步骤和使用WFC须知的大概内容。如果你想更进一步地了解WFC请参阅以后几章。若你想用WFC按某种方法去开发软件时，则可以把本章作为一个出发点。

第四章提供了一个用第三章介绍的函数编写的演示程序。

第五章介绍了如何控制颜色和WFC的逻辑属性系统。使用逻辑属性能编写出在彩色系统和单色属性黑白系统的基础上控制颜色的程序。本章也描述了通过使用物理属性控制颜色的方法。

第七章介绍如何建立和打开与内存文件相联系的窗口，用内存文件，你能检查和显示存储在磁盘上的文本文件，并且能获取供以后察看的实时数据。程序可内部地建立和编辑内存文件。本章还包括内存文件的结构和内存管理方面的技术资料。

第八章说明如何用内存文件存贮上托式帮助文件和菜单，以及后台缓冲器这些操作过程中的实时信息。

第九章介绍了几个高级技术，其中包括在多个窗口里显示多个文件，窗口的覆盖，消除窗口中的内容，移动，存贮和恢复窗口图象，活动字符，高亮度文本，打印窗口内容和绘图功能。并且还讨论了字符串和其它一些功能，其中也包含结构WINDOW的有关技术信息。

第十章说明了如何改变系统提示信息 and 错误提示信息。

第十一章讨论了WFC、IBM TOPView和Microsoft windows的兼容性，其它显示适配器，鼠标系统，UNIX、VMS和其它一些多用户系统，和可重入性的应用。

表格和清单这部分内容包括了正文中引用的所有表格和程序清单。

附录A列出了几个嵌入文件。它们是：wfc.h, att-glob.h, Computer.h, wfc_defs.h, vfc-glob.h和wfc_stru.h。

附录B按使用的种类和字母列出了各种函数清单。

附录C提供了在调用窗口系统函数时所使用到的变量和助记符的定义。

附录D是系统错误码和它们的含义。

附录E列出了常见WFC用户的编程错误，在编程出现错误时，使用附录E作为查找错误来源的第一步。

前 版 本 用 户 须 知

版本4.0到4.1的变化

当您使用WFC和WFD版本4.1时，必须重新编译你以前的代码，因为WFC在结构上作了很大的变化并增加了一些参照已被代替和改名函数的宏，所以重新编译是必要的。

主要的变化如下：在有一个变化的简要解释之后，都指出在手册中如何去查找更进一步信息的说明。

- 更改系统嵌入文件名，库文件名和函数名。
- 引入新的命名系统。
- 函数重新命名。
- 内存文件结构重新命名。
- 修改内存文件的内存管理。

- 取代函数def--fr()
- 新增加的函数和功能
- 系统结构的变化
- 修改的结构WINDOW
- VCS调试和错误处理系统
- 支持鼠标
- 获的系统信息和系统错误信息的简化方法
- 模板定义函数
- 改进的可重性应用的兼容性
- 增加了把窗口名设置在窗口边界的选择项
- 引进了判断窗口是否设置在屏幕上的新宏。
- 按第一字母选择的菜单项
- 在系统嵌入文件中取消NULL的定义
- 函数menu2()改变了当前菜单项的属性

更改系统嵌入文件库文件名和函数名

要使WFCIII版本的源程序能在新版本的WFC环境下运行，用户必须对新命名的嵌入文件，库文件和系统函数作一些相应的处理。具体处理说明如下：

重新命名系统文件

几个嵌入文件是按WFD标准进行重新命名的。新命名的嵌入文件如下：

旧名称	新名称
bios.h	wfc.h
window.h	wfc_glob.h
vextern.h	wfc_xtrn.h

对于WFD的用户，并且源程序仅参考了高级WFD嵌入文件(wfd.h和wfd_glob.h)，不必重新命名任何文件。

假如用户希望继续在程序使用老名称，可按下面几步去做。

(1) 后备一份WFC软件，然后把工作盘上的嵌入文件换成旧名。

(2) 把wfc.h中的语句#include vextern.h修改为#include wfc_xtrn.h。

另一种方法是建立一个仅仅嵌入wfc.h和windows.h的嵌入文件bios.h，在此window.h中要嵌入wfc_glob.h。有了这些文件，你不需修改源程序就能很顺利地编译子程序。

嵌入旧的嵌入文件

一些函数和宏已重新命名(详见以下)；然而WFC4.1版本仍然支持旧定义。这一任务是由wfc_hist.h中的宏定义命令#define来完成的为了编译包含旧的定义的程序，在主程序中必须嵌入wfc_hist.h这一文件。具体作法是在#include wf.h之前加上宏定义命令#define WN_HISTORY。

如下：

```
#define WN_HISTORY    /*嵌入旧定义*/
#include <wfc.h>
#include <wfc_glob.h>
```

重新命名窗口系统库文件

新命名的WFC库文件名称具有WFCx.lib的形式，x是编译系统使用的内存模型的助记符，lib是编译系统库文件的扩展名。

新命名的系统

随着系统版本的不断更新，我们已清醒地认识到以前给函数命名时作了一些不妥的并且是矛盾的选择。为了使我们自己和我们的新用户不受过去错误选择的影响，我们介绍了一种新方法去命名新函数，并想办法努力把这种对现有用户的影响减小到最低限度。

为了把所有具有密切关系的函数都集中到附注部分并简化记忆，我们采用了以下的方法命名函数。

- 函数名的第一部分应包含最重要的标识符，例如，所有的内存文件都是以“mf”开头。
- 尽量使用一致的缩写，因为DOS的文件名最多只能有8个字符，这样许多缩写比 我们想象的更加难懂，并且这些缩写还不可能自始至终保持一致。

新文件都是按照这种新规则进行命名的，当介绍某一部分的新函数时，我们将重新命名现有的文件以便使它们和新命名的系统保持一致。

修改函数名

WFC 4.1版本支持所有被替换和重新命名的函数。我们或提供一些宏把源程序转换为新名称下的程序，或已把函数作入库中。在附录B中被替换和重新命名的函数当作旧函数进行注意并指出如何查找新名称。

我们将在以后的版本中删除由等价的新函数支持的替代函数。因此建议用户以后编程序时应尽量使用新命名的函数。

下面列出一些被重新命名的函数

旧名称	新名称
def_fr	mf_def (函数本身也有变化)
di_file	mf_rd
di_st	rd_st
file_lnp	mf_rup
free_fil	mf_clr
s_tbfmsg	se_mfmsg
s_keyloop	se_keyloop
sctl_file	mf_sctl
sti_free	mf_rwrpl
sw_mfile	sw_mf
v_file	v_mf
vs_file	vs_mf

重新命名内存文件结构

旧名称	新名称
FREC	MFILE
FRECPtr	MFILEPTR

FLINE

MFLINE

FLINEPTR

MFLINEPTR

老的文件类型名是由包含在wfc_hist.h文件中的#define语句支持的，我们建议用户在程序中尽量使用新名称。

其它一些名称的变化

重新命名的一些出错代码和参数如下：

旧名称	新名称
出错码：FILETOOBIG	MFILETOOBIG
FROWNED	MFROWNED
FROWTOOBIG	MFROWTOOBIG
参 数：CENTER_TXT	CENTER_TEXT
LEFT_TXT	LEFT_TEXT
RIGHT_TXT	RIGHT_TEXT

修改内存文件

除了上面列举的名称变化外，内存文件系统也作了一些重要变化，所以前版本的用户都应参阅第七章。其中主要的变化有：

函数def_fr()的更换

函数def_fr()是定义内存文件新函数。它与原定义内存文件函数相比有很大的变化。新函数可为内存文件结构分配内存，并当一个函数所占的内存不需要时，允许释放。

新增加的函数和功能

下面的函数是新增加的：

mf_rwins()	在内存文件中插入一行
mf_rwdel()	在内存文件中删除一行
mf_wr()	把内存文件写到磁盘上

两个内存释放函数：

■ mf_free() 函数释放被文件或文件有关的数组和结构所占的内存，当这一函数被调用后，内存文件所占的所有内存都将释放。函数mf_free对由def_fr()命名的内存文件不起作用。

■ 函数mf_clr(mfp) 释放被内存文件所占的内存，这个函数取代了函数free_fil()，函数free_fil()并不真正释放文件所占的内存，而只是把相应的内容清除而已。

函数mf_rwins()和mf_rwrpl() (取代了函数sti_file()) 支持往内存文件的字符串加入换行符的功能。

省略的源程序

本版中我们没有提供与di_file()和v_file()两函数相对应的源程序。当内存文件系统初具规模时，我们曾介绍这些函数的源程序。为了用户能增加功能，这些函数现在已是新系统不可分割的一部分了。这就是说我们不能单独列出它的源程序了。

系统结构的变化

如果你已编写了一些使用系统结构的汇编程序，那么你就需按这些结构的变化对它们进行修改。请参阅系统盘上的wfc_stru.h文件就能进一步了解现在的系统结构。

结构WINDOW的变化

访问结构字符元素的函数已被改成访问整数变量。使用整数能避免由编辑器的字或字节定位选择引起的问题。当结构中没有字符时已编译的程序按字节或字定位都能运行。我们做了一些如下的变化:

■ 窗口结构成员 `wn.att`, `wn.bdratt`, `wn.page`, `wn.setsw` 和 `wn.popup` 现在都是整型而不是字符型。

■ 下列函数中的一些变量现在只接收整数而不接收字符:

函数	改变为接收整型的参数
<code>*rd-st (fp, stp, q, brk_ch,</code>	
<code>tab_q)</code>	<code>brk_ch</code>
<code>mv_rws (qlines, dir, &wn)</code>	<code>dir</code>
<code>v_bar (row_size, col_size,</code>	
<code>r_begin, c_begin, ch,</code>	
<code>attrib, &wn, &bdr)</code>	<code>ch</code>
<code>r_w(ch, q, &wn)</code>	<code>ch</code>
<code>r_9ch(ch, q, &wn)</code>	<code>ch</code>
<code>v_rw (ch, q, &wn)</code>	<code>ch</code>
<code>*formerly di_st ()</code>	

由于上述和其它一些结构的变化, 用户需要重新编译以前的程序。然而, 对于受影响的函数中的整型变量倒不需要修改。当不同类型的字符变量作为函数变量在编译时将被转换成整型变量, 所以一个字符变量类型和具有相同值的整型变量经过编译后是等效的。

VCS调试系统和错误处理系统

我们提供了5个调试和错误处理的辅助功能以帮助用户查找程序中出错的来源。

■ 错误跟踪系统保留以前被调用并有错的函数的记录, 当发现一个错误时这种记录清单就会显示在屏幕上。

■ 内存完整性检查, 帮助检查和确定内存的覆盖。

■ 每当查到出错时, 就置一个全局错误码。

■ 每当函数发现出错时, 就能安装并调用错误处理器

■ 支持ANSI标准函数原型

本手册的第五章将详细介绍VCS调试和错误处理系统。

支持鼠标

凡是具有把鼠标运动转换为存贮在键盘缓冲区中的字符驱动器的任何鼠标都与WFC兼容。有关WFC支持鼠标详细介绍请参阅第十一章。

所有系统信息都可以修改

所有的WFC提示信息 and 出错信息都可以动态地修改, 这就改变了WFC与其它语言的兼容性功能。对如何获取和改变系统信息的说明请参阅本手册的第十章。

窗口模板的定义函数

窗口可以用模板来定义。模板指的是所有函数调用中没有确定的结构成员的值。用户可以建立任意多个模板来使用。参阅第三章的窗口定义函数。

与可重入性应用的兼容性

以前版本的VCS窗口系统在某特定环境下会引起可重入性应用的问题。现在如果中断程序采取适当措施,可重入性问题可以毫无问题地处理(参见第十一章的“与可重入性应用的空”的一节)。

为确定窗口名的位置而设置的新选择项

现在窗口名可以放在边界各种不同的地方,用户可以用宏`sw_namelocation()`确定其位置。参阅第三章的“窗口的命名”。

判断窗口是否在屏上的宏

我们提供了一个确定窗口是否在屏幕上的宏:

```
isset_wn (wnp)
```

当窗口在屏幕上时,返回值是TURE,否则返回值是FALSE。

用菜单项的第一字符选择

在函数`menu2()`中我们设置了一个按或不按第一字符选择菜单项的宏:

```
sc_mucharselect (ON/OFF)
```

缺省值是ON

在嵌入文件中不再定义NULL

在系统的嵌入文件中不再定义NULL,我们通常把NULLP用作空数据指针把NULLFP用作空函数指针。

当前菜单项的属性修改

函数`menu2()`中的当前选择字段已从原来的LREVERSE修改为LCHOICEA,在单色屏幕上,我们看不到有什么变化,但在彩色显示器上,当前选择字段就有明显的不同。

第一章 WFC概述

本章叙述C窗口系统(WFC)的特性和功能。通过列举WFC的特性,本章将指导用户使用本手册的其他内容。同时本章对演示程序做了解释,用户可以使用这些演示程序作为学习工具,也可以作为自己程序中的子程序。

§ 1.1 WFC的特性

WFC提供了一系列窗口函数以简化普通屏幕显示工作。

WFC具有如下功能:

- 大小不受限制的窗口和文件
- 水平, 竖直滚动
- 上托窗口, 菜单和求助文件
- 用模板定义窗口功能
- 状态行管理
- 后台缓冲器
- 快速屏幕变化
- 节省内存使用
- VCS调试和错误处理系统
- 与Microsoft Windows和IBM TOPView的兼容性
- 与UNIX和VMS的兼容性
- 逻辑视频属性
- 色彩控制
- 高亮度
- 窗口名
- 具有字绕回和自动滚动的字符串输出
- 格式化字符串输出
- 打印窗口
- 键盘输入
- 读屏幕属性和字符
- 可变化的系统提示信息
- 支持IBM增加型图象适配配置
- 支持鼠标器
- 子程序库含有80多个建筑式的子程序

窗口管理: WFC对建立窗口的个数没有限制。每一个窗口都由一个C语言的结构控制。这个结构包含了管理窗口所需的全部信息。窗口结构的引入代替了在函数调用时所需的很长

的参数表。这样就减少了错误，简化了编程并节省了时间。

显示文件：为把ASCII码文件读入内存并通过窗口显示，WFC提供了一些子程序。付键盘解释器允许在文件内水平，竖直滚动。这些子程序在给用户提供帮助文件服务时特别有用。一些上下文关系敏感的帮助系统也很容易完成。

上托式窗口：窗口可以是上托到或写到屏幕上。在窗口结构中简单地设置一个开关就可以完成上托窗口。底层屏幕内容将自动为上托窗口保留并存储。

上托式菜单：WFC为上托式菜单提供了马上就可使用的子程序。菜单可以比窗口大。使用光标键来选择菜单项，选择项将用高亮度显示，也可以用菜单项的首写字母来选择。为了能够修改以满足个人需要，这些子程序的源程序也一并提供。

模板定义功能：可以使用模板来定义窗口。模板指定了所有在函数调用中没有指定的结构成员的值。用户可以建立任何数量的模板。模板使得建立窗口既方便又灵活。

状态行管理：状态行可以简单地作为单行窗口。字符串输出函数的特殊选择简化了更新状态行内容的工作。

后台缓冲器：WFC提供了一些功能用于管理和显示内存文件。这些文件可以任意大小并可动态更新。这些功能提供了比用虚拟屏幕来捕捉和显示实时信息更好的方法。

显示速度：屏幕显示是通过WFC自己的接口子程序来完成的，这些子程序在不失清晰性的情况下可以高速显示。

内存使用：窗口最初需要的内存仅是窗口结构需要的内存大约50字节。只有在需要时才使用内存。所有的函数都做成单个模块，所以只是那些需要的模块才被连接在可执行程序里。由于有效的设计，编码非常紧凑。具体地说，如果一个程序充分利用WFC的子程序库。程序只不过增加20到30K字节的内存需要。只有当窗口内容需要后台存储时，例如当一个上托式窗口需要显示时，才为窗口缓冲器分配内存，并且只分配够现时需要的内存量；不再使用时，缓冲器内存将释放。

VCS调试和错误处理系统：这个调试系统帮助用户找到错误的根源，为此提供了5项调试和查错帮助项。他们分别是：(1) 一个错误跟踪系统，这个系统保存一份错误发生前和发生时已调用过的函数记录。在检查出错误时这个记录将显示在屏幕上。(2) 内存完整性检查，此项提供检查和定位内存覆盖的帮助。(3) 查到一个错误都给定一个全局错误代码。(4) 可以安装一个错误处理器，这样一旦查到错误就可以调用这个处理器。(5) 支持ANSI标准的函数原型。

与Microsoft Windows和Topview的兼容性：WFC和WFD与Microsoft Windows和IBM TOPView具有完全的兼容性。用WFC来作屏幕输出和键盘输入的程序可以在Microsoft Windows和IBM TOPView里操作并在其环境下运行。使用VCS建立的程序可在TOPView和Microsoft系统下运行。用户不需要购买Microsoft的IBM的程序员工具包或在用户程序中引用的特殊代码。WFC自己处理兼容性，因为它时刻观察着Microsoft Windows和IBM TOPView的现状并对自己的屏幕处理作适当调整以便与Microsoft Windows和IBM TOPView的要求相统一。

与UNIX和VMS操作系统的兼容性：对DOS, XENIX, UNIX和VMS等操作系统，VCS窗口系统都有适应的版本。这些版本在源程序上具有兼容性。

逻辑视频属性：专业程序员喜欢在WFC中使用逻辑视频属性。可以只编写单个程序就

能使用 BM增强型图象适配器的全部彩色能力，也能使用标准的彩色/图象适配器的部分彩色功能；而且，当使用单色显示器时，程序仍能正常运行而显示结果完全清晰可读。

色彩控制：为了对 BM PC系列微机的色彩功能具有完全的控制能力，WFC提供了一些函数。窗口边界和窗口设置的背景颜色来清除整个窗口。

高亮度：窗口中一部分内容进行高亮度显示可以很容易用一个函数实现，这个函数只改变最高亮度显示部分的颜色属性而不改变其它文本内容。

向窗口中写：WFC有各种的字符串和字符输出函数。有各种开关控制字绕回、自动在全屏幕上滚动、自动更新光标位置、屏幕光标放置、自动清除到行末等。

格式化输出：和语句printf（）等价的格式化字符串输出函数也由WFC提供了。WFC的屏幕输出比C语言编译器提供的使用DOS功能调用的输出子程序printf（）更快。

打印窗口：窗口中的内容可以拷贝到标准的ASCII码文件也可由打印机输出。打印窗口内容由打印窗口函数用特殊的格式打印文本。

读键盘：WFC提供了读键盘函数，每次调用都能从键盘读入一个 BM扩展代码。也可以检查键盘缓冲器以确定是否有键被敲过。

读屏幕：WFC允许用户读屏幕内容，可以单个地读字符和属性，或者一次调用就可读出窗口中指定部分的字符属性内容。

支持 BM增强型图象适配器：如果IBM EGA卡是正在使用的显示适配器的话，WFC将设置一个全局变量。这样，当EGA卡被激活的时候，在彩色模式下的屏幕输出和单色显示适配器的情况下一样快。

可以改变的系统提示信息：系统提示信息可以通过一个全局变量来动态地改变。这就改善了WFC适应其他语言的能力。

支持鼠标器：只要鼠标器具有一个驱动器把鼠标器运动翻译成存放在键盘缓冲器中的字符，就可以在WFC中使用。WFC中已经有了支持Microsoft鼠标器和PC鼠标器的控制程序。

程序建筑块：WFC提供了一个程序库，它具有80多个建筑式的子程序。这样用户可以在提供的模块上修改或重新组合几个子程序以便满足用户要求。

§ 1.2 演示程序

手册中除了指导教学程序和编程实例外，还有几个组合的子程序，不需修改就可完成普通的窗口管理工作。这些子程序包括在演示程序中，一是演示他们的用途，二是作为使用窗口函数的教学工具。以下子程序提供了源程序：

- demo_wm——把多个ASCII文件读入内存并在多个窗口中显示他们，还为用户提供用光标付键盘控制来察看文件。光标控制子程序还有完整的C语言源程序。
- dem_menu——把一个ASCII码文件读入内存，并为用户提供调出一个菜单再选择其中一项的功能。使用光标付键盘命令在菜单中移动，这个菜单可以比显示菜单的窗口大。所指的菜单项以高亮度且反显示。选择结束后，菜单自动移走并且被以前的屏幕内容代替。选择的菜单项在状态行上出现。
- dem_cmov——演示WFC的色彩管理，窗口存贮和移动功能。在屏幕上，彩色窗口快速移动功能。图象的移动在游戏软件中是很有用的，但是窗口存贮和复位的高速度