



现代商务电脑实用系列丛书

中文 Windows 3.2 实用指南

袁源 肖晓 等 编著



Version 3.2

海天出版社

Windows 3.2

现代商务电脑实用系列丛书

中文

Windows 3.2

实用指南

袁源 肖晓 等 编著

海天出版社
(中国·深圳)

策划编辑 旷 昕
责任编辑 祝匡三 旷 昕
封面设计 张 音

内 容 简 介

Windows 是基于图形的优操作环境, 它把单击式控制、弹出式菜单, 以及运行 Windows 应用程序的能力及在 MS DOS 下运行标准应用程序的能力结合在一起, 使它成为个人机上最流行的操作平台。中文 Windows 不仅继承了西文 Windows 的全部功能, 而且增加了中文输入法和中文字体, 从而使它在国内迅速普及开来。

本书介绍中文 Windows 3.2 的使用与操作, 特别适用办公、商务以及家庭处理的初、中级用户阅读。

现代商务电脑实用系列丛书 中文 Windows 3.2 实用指南

袁 源 肖 晓 等 编著

海天出版社 出版

(中国·深圳)

海天出版社发行 (新华书店经销) 南京东海印刷厂印刷

开本 787mm×1092mm 1/16 印张 21 字数 522 千字

1996 年 9 月第 1 版 1996 年 9 月第 1 次印刷

印数 1—15000册

ISBN 7-80615-487-6/T·33

定价: 36.00 元

前 言

Microsoft 公司 1990 年推出的 Windows 软件,具有友好的窗口界面,使千千万万的用户很快学会了使用计算机。对广大非计算机专业人员来说,要记住许许多多的计算机命令(如 DOS 操作系统命令)是十分难堪的事情。方便操作是计算机软件普及,从而推动计算机普及的首要问题。

对国内的广大计算机爱好者来说,中文 Windows 无疑更具有诱惑力。有了中文 Windows,读者就能用中文写信、制表、记事及用中文字体打印文件。为此,中文 Windows 在延续西文 Windows 的全部功能的基础上增加了中文输入法和中文字体,从而使 Windows 在国内迅速普及开来。

Windows 3.2 对 Windows 3.1 进行了完善和提高,中文 Windows 3.2 在显示字体和中文输入方面增加了更多的选择,即显示字体更美观,输入方法更多,增加了如郑码和智能 ABC 输入方法等。

Windows 3.2 是目前使用最广泛,性能最稳定的产品,几乎所有国内 Windows 用户都在使用它,因此,帮助更多的读者熟悉和掌握中文 Windows 3.2 的操作,提高应用技巧,便是本书的主要目标。本书中错误与疏漏之处望广大读者批评、指正。

目 录

前言

1	Windows 入门	1	程序管理器窗口	45		
	历史展望	1	程序组	46		
	从 BIOS 10H 软件到 Windows 软件	2	组的建立改名和删除	48		
	Windows 是什么	3	在程序组之间移动(选定程序组)	50		
	标准用户界面	4	组窗口的打开和关闭	51		
	多任务	4	组窗口的重新排列	52		
	内存管理	6	程序项	51		
	队列输入	6	在组中建立程序项	54		
	消息	7	删除程序项	60		
	设备的独立性	7	复制程序项	60		
	动态连接库	9	移动程序项	62		
	“新可执行”格式	10	程序项图标的改变	63		
	MS-DOS 的应用程序	10	应用程序的启动	64		
	对窗口的增强	11	在组中启动程序项	64		
	Windows 软件的主要特点	11	启动程序项并使之以图标方式运行	64		
	基本操作	14	运行应用程序时将程序管理器缩成 图标	65		
	鼠标操作	14	程序项自动启动	65		
	键盘操作	14	从应用程序中退回到程序管理器	66		
	创建热键	18	退出程序管理器	67		
			保存程序管理器中的设定值	67		
			退出程序管理器	68		
2	Windows 基础	19	4	Windows 游戏	69	
	简单介绍	19		接龙游戏	69	
	Windows 的启动	20		启动接龙程序	69	
	Windows 启动后的窗口及应用图标	20		接龙规则说明	70	
	基本组成	21		移动纸版的方法	72	
	窗口结构以及分类	21		有关接龙游戏的一些技巧	73	
	图标的结构以及分类	24		修改接龙游戏的选项	73	
	对话框的结构以及分类	25		扫雷游戏	75	
	菜单	28		扫雷游戏窗口简介	76	
	操作	31		有关扫雷游戏的一些技巧	78	
	窗口操作	31		扫雷游戏的选项	79	
	图标的操作	34				
	滚动条的操作	34				
	菜单的操作	35				
	对话框的操作	37				
	帮助系统	42	5	Windows 附件	81	
	进入帮助系统	42		时钟应用程序	82	

启动时钟应用程序	82	将字符映射表中选定的字符插入到文 件中	123
移动时钟的位置和改变时钟的大小	82	退出字符映射表	124
改变时钟的显示	83		
改变时钟显示的字体	84		
时钟的显示的其他一些改变	84		
计算器应用程序	85	6 画笔	125
两种计算器的转换	86	画笔的启动	126
普通计算器使用方法	86	直接打开画笔	126
科学计算器使用方法	88	启动画笔的同时打开一个已存在的图 形文件	126
科学计算器中的统计功能	89	打开和存储画笔文件	127
日历	90	打开画笔文件	127
启动日历	90	存储文件	128
日历和月志的窗口介绍	91	画笔的画面简介和操作步骤简介	130
日历窗口的使用	93	画笔窗口画面简介	130
使用月志	100	在画笔中绘制图形的基本步骤	131
日历文件的打开、删除和存储	102	绘图区的控制	131
打印日历文件	103	选择绘图区的大小	131
退出日历应用程序	105	清除绘图区的内容	132
卡片盒	105	在屏幕上显示绘图区的更多部分	132
卡片盒的启动	105	移动画面	133
卡片盒窗口简介	105	颜色控制	134
卡片的填写	107	前景颜色和背景颜色的作用	134
卡片的增加	108	在调色板中选择前景和背景颜色	134
卡片的排列形式	109	线宽的选择	135
卡片资料的查找	109	绘图工具的选择和使用	136
卡片中文字的编辑	111	剪切工具的使用	137
在卡片盒中查找	112	喷雾工具的使用	139
在卡片中的编辑图画	112	文字工具的使用	140
恢复和删除卡片	114	橡皮擦工具的选择和使用	140
记事本	115	滚筒绘图工具的使用	144
记事本的启动	115	刷子绘图工具的使用	145
记事本窗口简介	115	直线工具的使用	146
在记事本中输入文字	116	曲线绘图工具的使用	147
选定记事本中的文字	117	矩形绘图工具的使用	148
在记事本中复制、移动和拷贝	117	圆/椭圆绘图工具的使用	149
打开记事中的文字	117	多边形工具的使用	150
在记事本中显示时间和日期	118	进一步输入文字	151
在记事本中查找文字	118	图形编辑	153
打印记事本中的资料	120	删除图形	153
退出记事本	121	粘贴画面	153
字符映射表	121	移动图形	154
字符映射表的启动	121	复制剪切块	155
字符映射表窗口简介	122	延伸移动剪切块	156
在字符映射表中选择字体	122		

在画笔中粘贴	157
图形水平和垂直翻转	158
缩小与放大	159
倾斜图形	160
反相显示	160
进一步改变绘图区的显示	161
放大观察绘图区	161
缩小观察和编辑绘图区	161
放大图形中的某一部分编辑	162
光标位置显示	163
生成特定颜色	164
特殊颜色的生成	164
特殊颜色的保存和再次调用	165
图形打印	165
打印设置	165
图形打印	166
退出画笔	167

7

文件管理器	168
启动文件管理器	168
文件管理器启动窗口	169
主窗口	169
文件管理器中的目录窗口	170
目录窗口中的图标	171
目录窗口的使用	172
改变驱动器	172
改变目录	174
展开目录窗口中的目录	174
展开某个目录的完整分支	176
折叠目录窗口	176
标识展开和折叠的目录	177
选定文件	178
目录窗口大小的改变	180
排列文件管理器中的图标	181
改变目录窗口的显示方式	182
改变文件显示的信息	184
显示文件的全部信息	184
显示文件的部分指定信息	185
目录内容的排序	185
显示文件不同类型	187
在目录窗口中字体的改变	187
窗口中的状态栏显示	189
将改变的内容保存起来	189

新建目录窗口	190
在多个目录窗口中选定某个目录窗口	190
在多个目录窗口中关闭某个目录窗口	191
排列目录窗口	192
在文件管理器中处理文件和目录	193
建立目录	193
查找文件或目录	194
文件或目录的移动	195
文件或目录的复制	196
将文件或目录复制到剪贴板	197
文件或目录的删除	198
文件操作中的信息	198
改变文件或目录的名称	199
文件的属性	200
文件与应用程序的关联	201
在文件管理器中启动应用程序	204
磁盘操作	204
格式化磁盘	204
制作系统盘	205
标识磁盘	206
退出文件管理器	206

8

控制面板	207
控制面板简介	208
控制面板中图标的启动	208
颜色图标的使用 改变窗口显示的颜色	210
选择 Windows 所使用的颜色组	210
改变颜色组	210
保存颜色组	212
删除颜色组	213
定义一种新颜色	213
将定义的颜色用到 Windows 中	214
桌面选择——改变 Windows 的外观	215
选择桌面图案	216
观察所选桌面图案的效果	216
修改已有的桌面图案	218
增加自己的桌面图案	219
删除桌面图案	220
显示或改变壁纸	221
定义应用程序快速切换的方法	222
暂时离开的时候——设定屏幕保护	223
设定网格在小	227
改变图标的间距	227

窗口边缘宽度的改变	228
光标闪烁快慢的改变	228
打印机的安装和配置	229
打印机驱动程序的安装	229
选定打印机端口	231
打印机的配置	232
选定打印机的“超时”选项	233
选择默认打印机	233
删除已安装的打印机	234
字体的设置	234
增加字体	235
TrueType 字体	236
删除字体	237
键盘速度调整	238
鼠标器设置	239
改变鼠标的跟踪速度	239
改变鼠标的按键位置	239
改变鼠标的双击速度	240
鼠标轨迹选定框	240
设定 Windows 系统的通信端口	240
指定不同的显示格式	241
基本设定	242
日期格式	243
“国别设定-日期格式”对话框	243
时间格式	244
货币格式	244
数字格式	245
安装驱动程序	246
新设备的驱动程序安装到 Windows	
中	246
将无用的驱动程序从 Windows 中删除	
.....	247
设置各种声音	248
声音设定	248
声音或警告的打开或关闭	248
退出控制面板	249

9

书写器	250
书写器的启动	250
启动书写器的同时打开一个已存在的	
文件	252
书写器窗口	252
文本输入	253

打开已有文件	253
打开书写器格式文本	253
打开非书写器格式文本	254
存储文件	255
使用书写器的“另存为”命令	255
使用书写器的“保存”命令	256
文本中移动	257
在文件中移动	257
移动在文件中的插入点	257
移动到指定页	258
选定文本	258
查找文本	261
查找文本	261
继续查找	261
查找特殊符号	263
重复查找上次查找内容	263
修改文本	264
查找并且替换	264
删除文本	265
复制文本(同一文件中)	266
移动文本(同一文件中)	267
复制文本(不同文件中)	268
改变文件或段落的外观	269
改变字形	269
上标和下标字体	270
改变字体的大小	270
改变行距	271
改变行对齐方式	272
改变跳格键设定	273
使用标尺来改变文档	274
在书写器中使用图形	276
嵌入图形到书写器文件中	276
复制图形到书写器文件中	277
在书写器中移动图形	277
改变图形的大小	278
重新定义页号	279
文件的自动分页	279
文件的人工分页	280
设定页眉或页脚	280
打印书写器文件	282
打印机设置	282
打印文档	282
退出书写器	283

10	打印管理器	285
	在使用打印管理器之前	286
	使用打印管理器打印	286
	启动打印管理器	286
	打印管理器的窗口和查看打印队列 ...	287
	打印方法	288
	暂停和恢复打印	288
	删除打印队列的文件	289
	改变打印速度	290
	控制打印管理器信息显示	291
	改变打印队列顺序	291
	显示日期/时间和文件大小	292
	打印到文件	292
	退出打印管理器	293

11	Windows 中文输入方法和字体	295
	汉字输入方法的安装	295
	汉字输入方法的选用	296
	汉字输入方法的删除	296
	汉字输入方法的启动	297
	最简单的方法	297
	切换汉字输入方法	297
	使用“输入方法”选项选用汉字输入 方法	298
	汉字输入方法的使用	298
	输入方法的设置功能	301
	拼音输入方法的设置	301
	Windows 中的码表输入方法	304
	Windows 3.2 中的各种字体	305

“字体”对话框	305
Windows 3.2 中的各种字体	307

12	MS-DOS 方式及 Windows 高级使用 ...	308
	在 Windows 中启动 MS-DOS 方式	308
	MS-DOS 方式窗口	308
	使用 MS-DOS 方式在 Windows 中作为 窗口显示	308
	MS-DOS 窗口中的控制菜单	309
	移动 MS-DOS 窗口	309
	改变 MS-DOS 窗口的大小	310
	改变 MS-DOS 窗口中显示的字体大小	310
	改变 MS-DOS 方式初始设置	311
	复制 MS-DOS 方式下的屏幕内容	311
	整屏复制	312
	窗口复制	312
	作标记将屏幕内容的一部分复制到剪 贴板中	312
	退出 MS-DOS 方式	313
	PIF 编辑器	314
	启动 PIF 编辑器	315
	改变 PIF 编辑器的方式	315
	指定标准方式下的 PIF 选项	316
	指定增强方式下的 PIF 选项	318
	386 增强方式	322
	进入 386 增强方式对话框	323
	386 增强方式对话框中的选项	323
	设置虚拟内存	324

1

Windows 入门

Microsoft Windows 是 Microsoft 基于图形的优秀操作环境。它将单击式控制、弹出式菜单、专为运行 Windows 编写的应用程序的能力及在 DOS 环境下运行标准应用程序的能力结合在一起。

历史展望

在 20 世纪 40 年代,计算机使用的是硬件拷贝设备(按今天的标准它是过时的),要求用户对大量的输出结果进行检查。为了调查飞机的稳定性和控制性,于 1950 年在 MIT(麻省理工学院)建立了该系统,最早计算机使用了阴极射线管(CRT),缩短了用户输入和计算机输出的时间,这就促使现代公共显示设备与计算机相连接。

20 世纪 50 年代飞机也促进了 SAGE 防空系统的发展,它将雷达的可视信号转换为计算机产生的图像。这个系统也是第一次利用光笔在显示屏上选择符号。

20 世纪 60 年代早期,一个 MIT 博士的候选人开发了草图直线绘制系统。这个系统可使用户通过对屏幕上光笔的选择来绘制草图。该绘图系统能画出直线或结构封闭的图形,因此简化了表达简单物体的复杂图形设计。早期的 CRT 能在显示屏上任两点间画出一条直线。然而,因为图像容易很快消失,所以它不得不每秒重复绘画几次。在 20 世纪 60 年代,这需要有一个用存储直线末端点的内存及快速重画直线的硬件(这两样都非常昂贵)。例如,在 1965 年,IBM 为这种图形显示引进了第一批大量生产的 CRT,单买显示器就用了将近 10 万美元,价格昂贵是这段时间内没有更多的人安装该设备的原因。三年以后,Tektronix 开发了第一批 CRT 存储管,这种类型的 CRT 能保存草图直至用户不再需要为止。由于设计了该显示管,昂贵的内存和重画图硬件就被淘汰了,从而将显示器花费降到了 1.5 美元万,凭这个价钱,Tektronix 显示器立即成功了。

20 世纪 70 年代,由于在内存和硬件逻辑单元花费上的急剧降低,促进了图形开发环境进一步发展。这种开发增强了扩展内存光栅扫描显示器,可产生逼真的阴影和色彩图案。

到了 20 世纪 80 年代末,显示监视器不再是数字量。IBM 视频图形

数组(VGA)的输出为模拟量,它保持了与先前应用程序环境的兼容性,以前的所有视频方式都受到支持。单色或有色图形适配器(CGA)和增强型图形适配器(EGA)的模式都可通过 VGA 适配器复制。例如,一个模拟的单色显示器与 VGA 相连,那么彩色就会转变为灰色。此外,用 VGA 还可得到下面几个新的模式:

- 640×480 图形(2 种色彩和 16 种色彩)
- 720×100 文本(单色和彩色)
- 360×400 文本(16 种色彩)
- 320×200 图形(256 种色彩)

以后还产生了 XGA 和 SVGA 监视器——具有更高的分辨率和更大的彩色调色板的监视器;并且,利用这些产品,用户不需要再改变已编写的 Windows 程序的源代码。

从 BIOS 10H 软件到 Windows 软件

为了解图形软件的发展历史,我们先介绍一下 BIOS(基本输入/输出)子程序。每个 IBM PC,PS/2 和兼容机上都固化有一套 BIOS 子程序,它们存储在 ROM(只读存储器)上。这些存储的子程序给具有标准特征的硬件提供了一个界面,包括时钟、键盘、软盘和硬盘及 Windows 的视频系统相关的硬件。视频 BIOS 程序由一套可执行基本视频程序工作的简单工具组成,像往荧幕写入字符串、清除字符和改换颜色等等。

中断 10H

过去为实现实时图形,程序员不得不编写一个汇编语言程序来访问同样也用汇编语言编写的 BIOS 程序。为访问 BIOS 的视频部分,使用 8088 处理器的程序员不得不发出中断 10H 命令。

现在 ROM BIOS 支持几种视频 I/O 函数,每种函数都通过执行一个中断 10H 来访问。这些函数被编了码;在执行一个中断 10H 前,程序员必须在相应的寄存器如 ah 中设置需要的子功能代码。

当执行中断时,微处理机上的其它寄存器可能保存有要送到 BIOS 程序去的其他参数。如果中断 10H 函数要调用返回程序的数据,那么它通过将数据留在一个或多个微处理器寄存器中就可以实现。不过,基于寄存器的参数传递主要用在汇编语言程序中。

高级语言

说到软件,读者可能对计算机的初步体验是使用 BASIC 命令: LINE, CIRCLE, COLOR 等;然后读者可能还学过其他语言如 Pascal, 它可以提高程序结构性,但不能提高图形能力。随着 Borland Turbo Pascal 的推出,程序员有了一个相当丰富的绘图环境。从 Turbo Pascal 4.0 开始,像视频器、剪裁、用户定义填充模式和立体条等许多复杂功

能都能实现。

另一种受人欢迎的开发语言是 C/C++，它在两个领域中实现得很好(汇编语言和高级语言的结合)。C/C++ 提供给程序员汇编语言的硬件存取能力以及高级语言的库逻辑结构特征。C/C++ 程序员在开发专用于图形的应用程序时，直接利用高级图形库子程序。

前面的讨论忽略了对并行图形应用程序的讨论。到目前为止所写的每个程序都使用了整个视频系统，包括各种寄存器、内存及显示装置等。

随着用户对软、硬件期望的提高，对能同时运行的程序的需求量也增大了。用户不再满足于他们编辑字符前等待数据库完成排序——他们希望同时做好这两项工作。这种需求在产品上帮助用户。像 Quarterdeck 公司的 DESQ View 和 Microsoft Windows(这些产品使用户能完成多任务)。更为重要的是 Windows 提供了图形设备界面(GDI)。

利用 Windows 的 GDI 函数，程序员能编出用户熟悉的应用程序(用菜单、滚动条和信息框等)。此外，当其他程序正在运行的同时，能对应用程序重设尺寸、移动、图标化、转进后台等。最为重要的是，GDI 函数允许一种应用程序与另一种应用程序进行通讯，尽管大多数应用程序还不能充分利用这种特性。

对软、硬件发展的短暂历史回顾解释了两者变化迅速的程度。用户和应用程序开发者今天所需要的是一个具备面向图形的用户界面、多任务能力，以及硬件独立性的环境。在这些方面，Windows 已经相当成熟了。

Windows 是什么

Microsoft Windows 是在 MS-DOS 下运行的一个操作环境。它是一个以图形为基础、具有多任务的窗口环境。所有专门为 Windows 编写的程序都有一致的形式和命令结构，以使新的 Windows 应用程序较易掌握。

对于应用程序的开发，Windows 提供了大量的内建子程序，使用户很容易实现弹出式菜单、滚动条、对话框、图标以及许多其它对用户友好的图形界面。刚开始使用 Windows 时，用户可使用新的对话控制、菜单类型及“所有者绘图控制”。通过使用 Windows 提供的扩展图形编程接口，能容易地将应用程序格式化，并以各种字体和间距输出文本。

Windows 也允许应用程序的设计人员以一种与设备无关的方式对待视频显示器、键盘、鼠标、打印机、串行口以及系统计时器。这可以使一个应用程序在不同硬件配置的情况下正常运行。

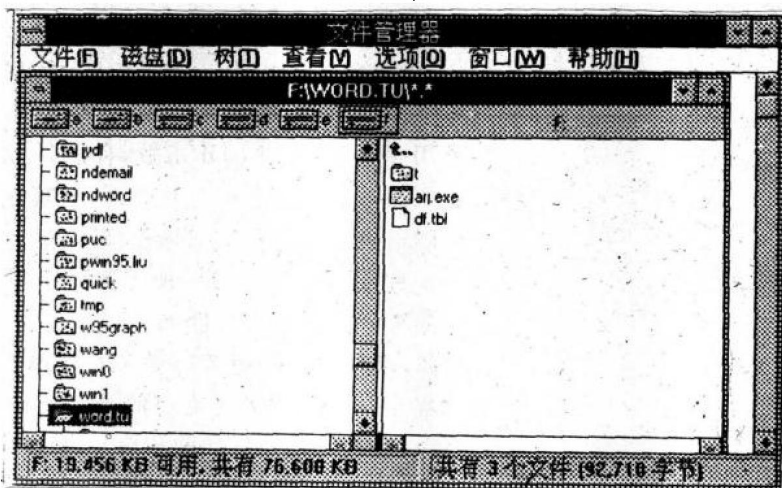
和传统的 MS-DOS 环境相比，Windows 操作环境给用户和程序员带来了相当多的好处。三个主要性能(面向图形的用户界面、多任务以及硬件独立性)已明显不新颖了，需要的是将这三种能力结合到单一的

微机操作环境中。

标准用户界面

在 Windows 提供的三种主要特性中,标准化的与面向图形的用户界面是最重要的,当然对用户也是最重要的。一般的用户界面使用图形或图标来表示驱动器、文件、子目录和多个操作系统命令及功能。图 1.1 为一个典型的 Windows“文件管理器”的窗口。程序由标题条来识别,许多基本的文件操作功能通过用鼠标单击菜单来访问。大多数 Windows 程序既有键盘界面又有鼠标界面。尽管大多数 Windows 程序的功能能通过键盘控制,但使用鼠标在很多方面更为方便。

图 1.1
Windows“文件管理器”的窗口



因为所有程序都有相似的“外表”,所以用户不需要再花费很多时间去学习使用一个新的应用程序。图 1.2(Windows“画笔”的屏幕)和图 1.3(Windows“书写器”的屏幕)说明了这个相似性。注意公共的“文件”和“编辑”选项及“最大化/最小化”按钮。对程序员来说,一致的用户界面通过使用直接固化在 Windows 中的子程序来实现菜单和对话框。所有菜单有相同的键盘和鼠标界面,由 Windows 而不是应用程序来处理这项工作。

多任务

多任务操作系统允许用户同时运行几个应用程序,或同一应用程序同时发出几种请求。尽管一些人仍怀疑多任务在微机上是否必要,但像 Borland 的 Sidekick 和 Norton 实用程序这样的 TSR(终止并驻留)应用程序的成功证明了它们的必要性。图 1.4 给出了几种平铺的 Windows 应用程序,每一程序在屏幕上都占用一个矩形窗口,用户能随时在屏幕上移动窗口、改变尺寸、切换不同应用程序以及交换窗口间的信息等。

图 1.4 的例子表示四个并发运行的进程。它们中仅有一个能在任

图 1.2

Windows“画笔”的屏幕

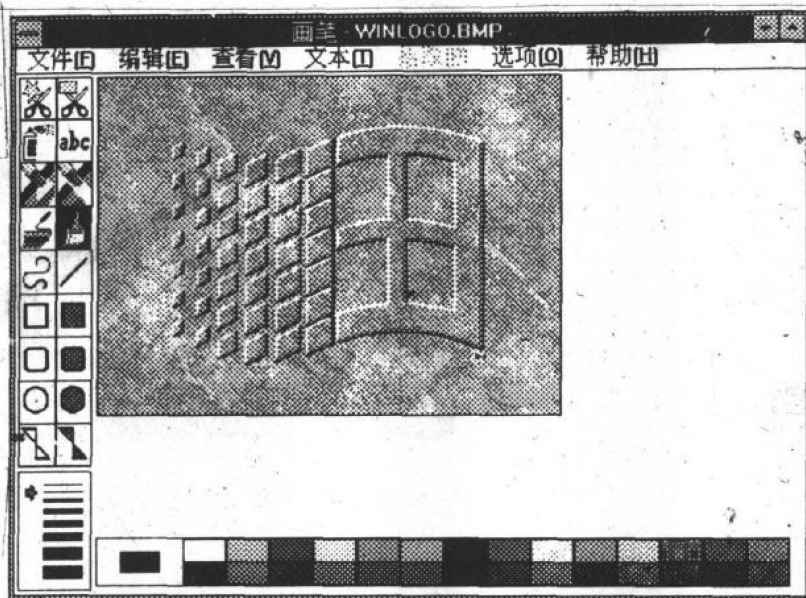
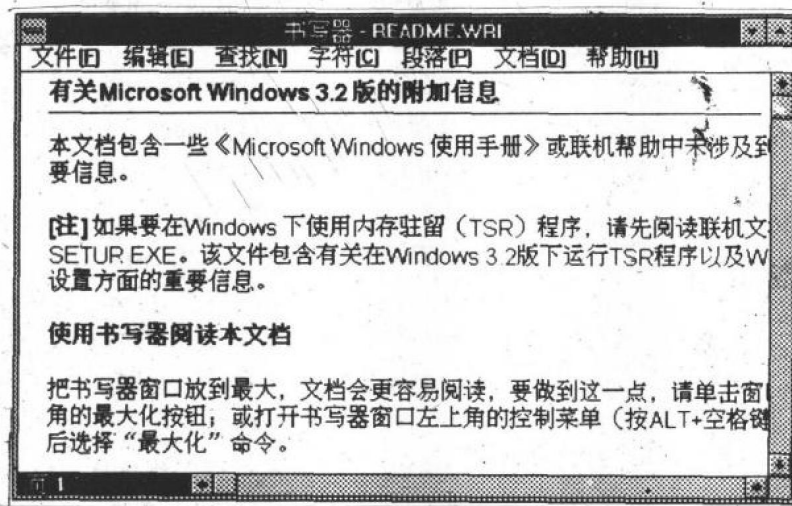


图 1.3

Windows“书写器”的屏幕

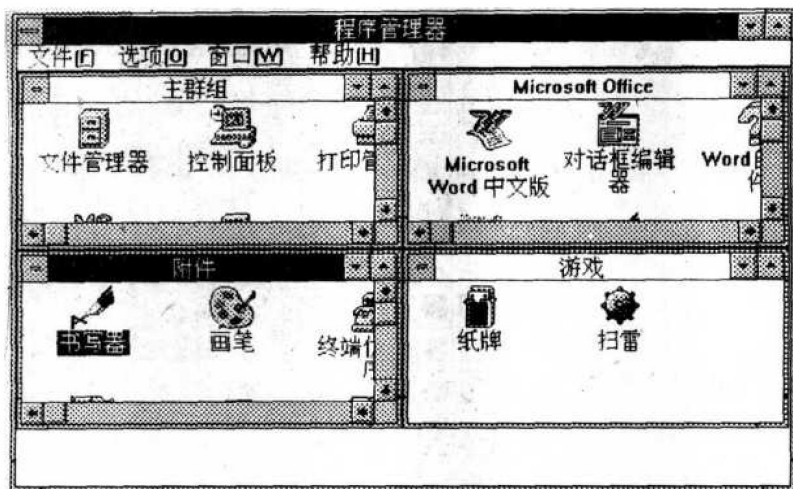


一给定时间使用处理器。在处理任务和运行任务之间的区别是很明显的。一个应用程序很可能处在第三种状态，即所谓的活动状态。一个活动的应用程序能够接受用户的输入。正如在任一时刻只能运行一个应用程序一样，某一时刻也只有一个活动应用程序。不过，存在大量的同时运行任务。Windows 负责分配微处理器时间，它通过使用排队输入和消息来控制对处理器的共享。

在多任务操作系统以前，应用程序已全部控制了计算机资源，包括输入输出设备、内存、视频显示，甚至 CPU 本身。然而，在 Windows 环境，所有这些重要资源都必须共享。例如，对于系统，程序本身和任何 TSR 程序遗留下来的内存，标准 C/C++ 程序不再全部占用。

图 1.4

平铺的 Windows 应用程序



内存管理

在 Windows 中,内存是最重要的共享设备。当不只一个应用程序在同一时刻运行时,为避免它用完内存,这些程序必须协调共享内存。此外,当旧程序结束新程序开始时,内存就会变得不连续。Windows 可通过移动内存中的代码和数据块来优化空闲存储空间。

Windows 允许应用程序“重复使用”内存。也就是说,一个应用程序可以有比一次放入内存的代码数还要多的代码。Windows 能够从内存中放弃一些代码,然后从程序的 .EXE 文件中重新装入代码。

在其他环境下,用户在并行运行程序时,可能会有几个实例或者拷贝。为了节省空间,Windows 共用相同的代码。在 Windows 环境下运行程序时甚至能共享在其他 .EXE 文件中的程序。包含这些共用程序的文件称作动态链接库(dynamic link library)。Windows 包含在运行时将动态链接库中的例程与程序相连接的机制(Windows 本身就是一系列动态链接库)。为方便起见,Windows 程序使用了一个新的 .EXE 文件格式,叫做“新可执行”格式。这些文件包括 Windows 管理代码段、数据段及执行动态连接的信息。

队列输入

正如内存是 Windows 环境下的共用资源一样,键盘及鼠标输入也一样。C/C++ 程序不能再通过 getch 函数直接从键盘读入。在 Windows 环境下,一个应用程序不能显式地从键盘或鼠标等进行输入;相反地,Windows 接受所有从键盘、鼠标和计时器的输入,这就是所谓的系统队列。通过从系统队列到应用程序队列的拷贝,队列将输入重定向到相应的应用程序中去,这时,当应用程序开始处理输入时,它就能从队列读入并将信息分配到相应的窗口。

输入按称为输入消息的统一格式提供。所有的输入消息指明了系

统的时间、键盘状态、压下键的扫描码、鼠标位置(如果有的话)以及那个设备产生该消息的信息等。

任何键盘、鼠标及计时器信息都有统一的格式并以同样的方式处理。此外,对每一个消息,Windows 都提供一个独立设备的虚拟键码来识别键(它不在乎是何种类型的键盘),相关设备的扫描码可由键盘以及键盘上其他键的状态产生,像 Num Lock, ALT.SHIFT 和 CTRL。

因为键盘和鼠标是一个共享资源,所以键盘和鼠标在 Windows 环境下提供每个运行程序的所有输入。利用键盘,各种输入消息直接到了当前活动窗口;然而,鼠标消息则要不同对待,它们被送往鼠标下的窗口中。

计时器消息与键盘和鼠标非常相似。Windows 允许程序设置计时器,以便它的其中一个窗口能收到周期间隔消息。这个消息直接送到程序的消息队列中。当程序调用某个 Windows 函数结束时,也会有其他消息输入程序的信息队列中去。

消 息

Windows 的消息系统是用于在一个多任务环境中传递信息的根本结构。从用户应用程序的观点来看,消息可以看作是产生了一些感兴趣的事件的通知,它可能需要也可能不需要一个特殊动作。这些事件可能由用户产生,如移动鼠标、改变窗口尺寸或选择菜单。不过,这种事件也可能由应用程序产生。例如,图形电子表格已完成了重算工作,如果需要修改饼形显示图;这时,应用程序就会给自己一个“修改窗口”的消息。Windows 也可能产生消息,像“关闭”这种情况。在这种情况下,Windows 就会通知每个应用程序关闭。最终的重要消息来源应该是仪器的监控程序,如指明主要化学进程已达到指定温度。不论是何种消息来源,程序都必须采取相应措施。

当考虑 Windows 中的消息作用时必须牢记两个主要方面。首先,Windows 通过消息系统来实现它的多任务特性。消息系统使 Windows 能够在几种不同应用程序间共用处理器。每当 Windows 送一个消息给应用程序时,它也给予了处理机的时间。实际上,应用程序在收到消息时才能访问微处理器。Windows 消息的第二个作用是它能使应用程序对该环境下发生的事件作出反应,不管这些事件是由应用程序本身产生的,还是由其它正运行的应用程序、用户或 Windows 产生的。当事件发生时,Windows 就会注意到它,并将一个相应消息分配到所关心的应用程序中。所以,以最简单的水平来看,有人可能会说 Windows 应用程序的主要任务就是处理消息。

设备的独立性

由 Windows 提供的第三个主要功能是硬件设备的独立性。前面已经看到了应用程序对硬件设备影响变化是何等迅速,尤其当硬件独立

性与视频显示相联系时,读者就会明白硬件独立性的好处。这一节将集中讲解如何实现设备的独立性。

Windows 使程序员从不得不解释各种可能变化的监视器、打印机、输入设备这些工作中解脱了出来。目前,一个非 Windows 的应用程序必须编写各种设备的驱动程序。很明显,在软件开发上这是一个很大的缺点。

一般地,为使一个应用程序能在任何打印机上打印,程序员为每个程序都装备了不同的驱动程序。这就要求软件公司反复不断地编写相同设备的驱动程序。例如,一个 IBM Proprinter 为 WordPerfect 编写两个驱动程序:一个给 Wordstar,一个给 Ami 等。

在 Windows 环境中,每个设备驱动程序——不论是为视频显示、打印机、键盘,还是鼠标,只需编写一次,勿需自己编写一套完整的驱动程序,而只由硬件公司为系统编写一个驱动程序。Microsoft 有许多 Windows 驱动程序,Windows 没有提供可从硬件厂家获取。例如,其中一种 Windows 通用的增强型(HPPCL.DRV)驱动程序是支持外部的字符卡的,设备独立性已用到了惠普激光打印驱动程序中。当安装了 Windows 后,在当前系统下每个设备都有一个驱动程序。应用程序不论何时命令打印或返回,Windows 都通过相应的驱动程序传送输出。将所需驱动程序直接与系统结合避免了在程序上进行的大量重复工作。

对于应用程序的设计员来说,没有什么事比这更容易了。应用程序是与 Windows 相作用而不是和一些特殊设备,它不需要知道连接的是什么样的打印机。当应用程序要求画一个矩形填充图时,Windows 就会在最后的“3-D Surrealistic Animation Holographic SilkScreener”上执行该命令。采取相同的约定,每个设备的驱动程序都能为每个 Windows 应用程序服务。程序员节省了时间,用户也不用担忧是否每个新的应用程序都能支持它们所喜欢的光笔。

Windows 通过指定硬件应具有的性能来实现设备的独立性,和 Windows 提供的 SDK(Software Development Kit)子程序一起,提供了保证程序能正确运行的最小性能。若不考虑复杂性,每个 SDK 子程序都能将它自己分解成设备所需的最小操作单位。例如,并不是每个绘图机都能画出一个椭圆。不过,作为一个应用程序开发者,即使绘图机没有绘制椭圆的能力仍能使用 SDK 子程序来画椭圆,因为每个与 Windows 连接的绘图机必须有画直线的能力,所以 Windows 能将椭圆分解成一系列小直线。

谈到输入,Windows 能定义一套最简单的程序来保证应用程序能获得唯一有效的、预定义过的输入。Windows 事先已定义好了一套合法击键的集合;换句话说,Windows 已定义了一个 Windows 应用程序可能接收的击键。这与 IBM PC 键盘产生的效果是一致的。如果厂家能生产出一套键盘,所含键不存在于 Windows 的可接收键中,那么厂家