

北京希望电脑公司技术丛书

Microsoft Windows 软件开发技术基础

(上册)

夏洪山 编著
林志坚



海洋出版社

前 言

随着微型计算机的日益普及,研制和开发适用于普通用户易学好用的高效率微型计算机软件成为当今软件技术发展的重要特点之一。

Microsoft Windows 这个多窗口多任务软件系统,正是将这些特性融于一体最为成功的代表之一。它的操作命令图形化,意形结合;用户界面一致而友好,使操作简便;独立于设备,使软件兼容性强;支持多任务多道作业,使机器效率显著提高。更为重要的是,Windows 3.0 以上版提供保护方式的内存管理,使系统运行空间突破 640K RAM 禁锢;丰富多彩的图形特征已成为基于图形的软件界面标准。

然而,Windows 的易学好用,并没有使 Windows 软件开发的难度减轻。由于它完全不同于 DOS^①,许多新概念使得通常的 DOS 程序设计方法似乎无能为力。Windows 的新颖功能如此诱人,使得个人计算机的应用对用户的要求愈来愈低。但是,Windows 中种类繁多的把柄(handle),千变万化的信息(message),难以理解的事例(instance),捉摸不定的内存分段(segment)等等,却把软件开发人员搞得混淆不清,不知何处下手,才能开发出一个图文并茂的 Windows 应用软件。笔者正是针对这个目的而编写本书。本书将从 Windows 系统的基本概念开始,逐步引入各种概念;从编制只显示一个窗口的最简单的程序入手,系统地介绍实现各种功能的程序设计方法和软件开发技巧。希望通过阅读本书的有关章节,开发 Windows 应用软件过程中通常碰到的问题,都将能够迎刃而解。

本书虽然是专为 Windows 软件开发而写,但是对于极想了解 Windows 系统内部运行机制的读者同样具有帮助。当然,本书的读者最好具有阅读 C 语言程序的基础。如果了解 Intel 80386 系列 CPU 的结构,将更容易理解 Windows 的动态特性和内存(尤其是保持方式下的)管理策略。本书中的每一道程序,至少说明一种 Windows 功能的实现方法和开发步骤。书中的所有范例程序都是在 Windows 3.0 版下经过实际运行通过的源程序代码。如果读者打算亲手试一试这种开发过程!Windows 3.0 版基本系统,Windows SDK 3.0 版,Microsoft C 5.1 或 C 6.0 都是必不可少的基本软件。如果增加一个显示器或显示终端,调试程序就更为方便有效。

全书共分十八章介绍,分为上下两册。上册包括第一章至第十一章的内容,下册为第十二章至第十八章的内容。第一章简要介绍 Microsoft Windows 的发展过程,Windows 的各种特性与构造,与 DOS 的区别和内在联系,以及 Windows 的基本概念。第二章介绍创建窗口的的基本方法与开发步骤,以及有关信息的处理。第三章介绍如何在窗口用户区中显示文本和绘制图形,并介绍滚行杠的应用。第四章的主要内容为应用程序的键盘接口以及字符集的选择。有关对鼠标操作产生的几种信息的处理,将在第五章中介绍。第六章介绍 Windows 系统时钟的三种使用方法,以及系统时钟的实时性分析。在第二章的基础上,第七章介绍 Windows 系统中各种控制键(control)的应用,并介绍颜色的控制。Windows 的 RAM 管理是第八章的主要议题,同时还详细介绍实模式(real mode)地址和保护模式(protected mode)地址的形成,Windows 的三种 RAM 模式的内存结构,内存分段技术,内存的分配与管理。第九章介绍 Win-

① 本书中所用的“DOS”一语,泛指各个人计算机制造商提供的与 MS-DOS 兼容的微型计算机磁盘操作系统。

dows 资源,包括标象、光标、位图和字符串的创建与引用,以及图元文件做资源的应用。第十章介绍字符清单和位图清单的创建,键盘、鼠标与清单的虚拟键盘接口,以及各类清单的处理。会话框是另一类窗口,它有独特的用途。第十一章主要介绍形式会话框与无形式会话框的创建,以及在文件操作中的应用。第十二章介绍 Windows 的图形设备接口(GDI)概念,以及在六种映射方式下和不同度量单位时的坐标系变换与显示方法。第十三章介绍 Windows 的点、线、弧等基本图形应用。位图与图元文件的创建及其应用,位图的光栅操作,颜色特性等将在第十四章中介绍。第十五章着重介绍字体的种类、字体特性及其在文本中的应用,其中包括字体的创建。第十六章介绍动态数据交换的实现,包括数据传递区(clipboard)的产生与应用,以及 DDE 技术。第十七章分析 2.0 版与 3.0 版 Windows 动态库的区别和它们的创建方法,同时讨论动态库的应用。第十八章介绍 Windows 中打印机如何经过 SPOOLER 和 Escape 功能实现打印,并重点介绍分块(Banding)打印技术。

本书由夏洪山组稿,与林志坚共同编写。第一章至第十七章的内容编写,程序编制与调试,以及全书插图的绘制由夏洪山完成。林志坚承担了第十八章内容的编写。在本书的编写过程中,得到上海科普出版社胡名正先生的指导,《工业控制计算机》杂志编辑部张开元先生的热情支持。全书文字的录排由张筱婕小姐负责完成。对本书的出版、印刷和发行,北京希望电脑公司秦人华女士给予了鼎力支持。值此对他们表示衷心感谢。

由于编者水平所限,书中错误难免,敬请读者不吝指正。

编者

一九九一年五月

于江苏省计算技术研究所

目 录

第一章 Windows 和应用软件开发	1
第一节 Windows 的发展	1
第二节 基本概念	2
1.2.1 “视觉界面”	2
1.2.2 用户界面的一致性	3
1.2.3 视觉界面的由来	4
1.2.4 多任务特性	4
1.2.5 Windows 的实模式内存管理	5
1.2.6 与设备无关的图形界面	6
1.2.7 Windows 下运行 DOS 应用程序	7
1.2.8 Windows 的缺点	7
第三节 Windows 的结构分析	7
1.3.1 DOS 和应用程序之间的理论关系	8
1.3.2 DOS 和应用程序之间的实际关系	9
1.3.3 Windows 和应用程序	9
1.3.4 “面向目标的程序设计”	11
1.3.5 多窗口概念	12
1.3.6 信息系统	12
第四节 Windows 应用软件开发	16
1.4.1 8086 系列 CPU 结构	16
1.4.2 Windows 函数调用	18
1.4.3 Pascal 调用规则	18
1.4.4 动态链接过程	18
1.4.5 可执行文件格式	19
1.4.6 内存管理初探	20
1.4.7 把柄	20
1.4.8 windows.h	20
1.4.9 变量类型转换	23
1.4.10 符号规则	24
第二章 多窗口的创建	26
第一节 源代码的编译	26
2.1.1 编译开关	28
2.1.2 模块定义文件和链接	29
2.1.3 产生 EXE 文件	30
第二节 创建窗口	31
2.2.1 窗口类别	33
2.2.2 创建窗口	35

2.2.3	窗口函数 WndProc	37
2.2.4	关于 WndProc	37
2.2.5	默认信息处理	39
2.2.6	应用举例	39
2.2.7	信息循环	42
2.2.8	非先入性多任务	45
2.2.9	窗口的特性	46
2.2.10	固定大小的窗口	50
2.2.11	一种类别, 多个窗口	53
第三节	上托窗口和子窗口	56
2.3.1	上托窗口	56
2.3.2	上托窗口和父窗口	60
2.3.3	子窗口	60
2.3.4	应用举例	61
第三章	用户区的正文显示	68
第一节	绘画与刷新	69
3.1.1	WM_PAINT 信息	69
3.1.2	有效和无效矩形区	70
第二节	GDI 基础	71
3.2.1	设备输出缓冲区	71
3.2.2	获取设备输出缓冲区把柄方法之一	71
3.2.3	PAINTSTRUCT 结构	72
3.2.4	获取设备输出缓冲区把柄方法之二	74
3.2.5	TextOut 函数	74
3.2.6	系统字体	75
3.2.7	字符尺寸	76
3.2.8	正文度量分析	76
3.2.9	格式化正文	77
3.2.10	正文显示	79
3.2.11	显示空间问题	82
3.2.12	用户区尺寸	82
第三节	滚行杠	84
3.3.1	两种类型的滚行杠	84
3.3.2	滚行范围和滚行位置	85
3.3.3	滚行杠信息	85
3.3.4	滚行杠的应用	86
3.3.5	关于鼠标	96
第四章	键盘	97
第一节	键盘基础	97

4.1.1	键盘驱动程序	97
4.1.2	键盘信息的筛选	98
4.1.3	输入光标	99
4.1.4	键盘输入和字符	99
第二节	键盘信息	99
4.2.1	系统键和非系统键	100
4.2.2	lParam 变量	100
4.2.3	虚拟键代码	102
4.3.4	Shift 状态	104
4.2.5	键盘信息的使用	105
第三节	键盘接口	105
4.3.1	WM_KEYDOWN 信息	106
4.3.2	发送信息	107
第四节	字符信息	112
4.4.1	WM_CHAR 信息	114
4.4.2	死字符信息	115
第五节	键盘信息	115
第六节	Windows 字符集	119
4.6.1	OEM 字符集	119
4.6.2	ANSI 字符集	120
4.6.3	字符集与字体	120
第七节	其它语言的键盘	120
4.7.1	字符集应用	120
4.7.2	DOS 与字符集	121
4.7.3	数字小键盘的使用	122
第五章	鼠标器	123
第一节	鼠标器基础	123
5.1.1	鼠标键	123
5.1.2	鼠标器驱动程序	124
第二节	用户区鼠标器信息	124
5.2.1	鼠标器信息的处理	125
5.2.2	POINT, RECT 和 lParam	129
5.2.3	Shift 键的处理	130
5.2.4	鼠标器的连续按键	131
5.2.5	连续按键的间隔时间	131
第三节	非用户区鼠标器信息	133
5.3.1	瞬时测试信息	133
5.3.2	信息链	135
第四节	应用程序中的瞬时测试	135

5.4.1	应用举例	135
5.4.2	键盘与鼠标器	139
5.4.3	键盘接口的应用	141
5.4.4	子窗口的瞬时测试	144
第五节	捕获鼠标光标	150
5.5.1	鼠标信息的应用	150
5.5.2	鼠标器光标	153
5.5.3	StretchBlt 函数	154
第六章	定时器与系统时钟	156
第一节	定时器基础	157
6.1.1	定时器驱动程序	157
6.1.2	非异步定时器信息	158
第二节	定时器的使用	159
6.2.1	方法一	159
6.2.2	方法二	164
6.2.3	方法三	169
第三节	用于状态报告的定时器	169
6.3.1	巧用标象	172
6.3.2	程序标象	172
6.3.3	保持标象	173
6.3.4	可用内存空间	173
第四节	定时器的应用——时钟	174
第五节	Windows 标准时间	179
第七章	子窗口类控制框	180
第一节	键类别	181
7.1.1	创建子窗口	184
7.1.2	子窗口向父窗口发信	185
7.1.3	父窗口向子窗口发信	186
7.1.4	回答键	186
7.1.5	检查框	187
7.1.6	指示键	188
7.1.7	成组框	188
7.1.8	键中的正文修改	189
7.1.9	修改键的正文	189
7.1.10	可见键和使能键	189
7.1.11	控制键和输入光标	190
第二节	控制键和颜色	191
7.2.1	颜色定义	191
7.2.2	系统颜色	192

7.2.3	控制键颜色	193
7.2.4	WM_CTLCOLOR 信息	194
第三节	静态类控制框	196
第四节	滚行杠类别	197
7.4.1	颜色的应用	198
7.4.2	自动键盘接口	204
7.4.3	窗口子类别	204
7.4.4	背景着色	205
7.4.5	滚行杠着色	206
7.4.6	多事例处理	207
第五节	编辑型控制框	207
7.5.1	编辑型控制框	210
7.5.2	编辑型控制框通知码	211
7.5.3	编辑型控制框的应用	211
7.5.4	编辑型控制框中的信息	212
第六节	列表框	213
7.6.1	列表框格式	213
7.6.2	列表框中的字符串	214
7.6.3	列表框中项的选择与提取	215
7.6.4	列表框信息	216
7.6.5	列表框的应用	217
7.6.6	列文件	220
7.6.7	Windows 下的 head 程序	221
第八章	内存管理	226
第一节	内存结构	226
8.1.1	基本内存结构	228
8.1.2	EMS4.0 内存结构	229
8.1.3	标准模式的内存结构	230
8.1.4	386 增强型内存结构	231
第二节	内存组织	232
8.2.1	全局内存	232
8.2.2	局部内存	234
第三节	内存模式	235
第四节	内存的分段	237
8.4.1	固定段和可移动段	237
8.4.2	多重代码段	238
8.4.3	段移动问题	239
8.4.4	段属性	240
第五节	程序段的移动与再入	242

8.5.1	远程函数的特殊处理	242
8.5.2	Windows 应用程序的执行	245
8.5.3	MakeProcInstance 的作用	246
8.5.4	Windows 动态库	247
8.5.5	远程函数的返回	247
第六节	应用程序中的内存分配	248
8.6.1	内存块加锁	249
8.6.2	全局内存函数	250
8.6.3	可放弃内存块	253
8.6.4	巨型全局内存块	254
8.6.5	局部内存的分配	255
8.6.6	私有数据段加锁	257
8.6.7	内存把柄与指针	258
8.6.8	C 语言的内存分配函数	259
第九章	标象、光标、位图和字符串	261
第一节	编译资源	261
第二节	标象与光标	262
9.2.1	标象位图和光标位图	265
9.2.2	色彩设计规则	267
9.2.3	标象与光标的几个问题	267
9.2.4	光标文件和标象文件格式	268
9.2.5	标象把柄	270
9.2.6	标象的使用	271
9.2.7	不同光标的应用	272
第三节	资源与内存	273
9.3.1	位图	273
9.3.2	位图文件格式	274
9.3.3	位图和刷	275
第四节	字符串	278
9.4.1	字符串资源的应用	279
9.4.2	MessageBox 和字符串	279
9.4.3	字符串资源与内存空间	280
第五节	用户自定义资源	281
第十章	清单与虚拟键盘	287
第一节	清单	287
10.1.1	清单结构	287
10.1.2	清单模型	288
10.1.3	清单的使用	290
10.1.4	清单和信息	291

10.1.5	上托清单的应用	294
10.1.6	清单格式	298
10.1.7	清单的定义方法	298
10.1.8	系统清单	300
10.1.9	ChangeMenu 函数	303
10.1.10	其它清单命令	304
10.1.11	创建清单的其它方法	306
第二节	位图清单	310
10.2.1	清单位图	314
10.2.2	内存设备输出缓冲区	315
10.2.3	文本位图	315
10.2.4	位图的比例缩放	317
10.2.5	链接清单	318
10.2.6	加入键盘接口	318
第三节	虚拟键盘	319
10.3.1	虚拟键盘的使用	319
10.3.2	安排虚拟键盘的几条规则	319
10.3.3	虚拟键盘转换表	320
10.3.4	虚拟键盘转换表的装入	322
10.3.5	键盘代码转换	322
10.3.6	虚拟键盘转换信息	323
10.3.7	清单与虚拟键盘的应用举例	324
10.3.8	使能清单项	328
10.3.9	处理清单选择	329
第十一章	会话框	332
第一节	形式会话框	332
11.1.1	创建会话框	332
11.1.2	会话框模型	335
11.1.3	会话框函数	337
11.1.4	输出会话框函数	338
11.1.5	会话框函数的调用	338
11.1.6	会话框格式	340
11.1.7	定义控制框	341
11.1.8	复杂会话框	343
11.1.9	会话框中的控制框	348
11.1.10	OK 和 Cancel 按钮	351
11.1.11	Tab 键和成组框	352
11.1.12	会话框内的绘图操作	354
11.1.13	用于会话框的其它函数	355

11.1.14 用户自定义控制框	355
第二节 信息框	361
11.2.1 说明性信息框	362
11.2.2 提示信息	363
第三节 文件操作	364
11.3.1 OpenFile 函数	364
11.3.2 文件的 I/O	366
11.3.3 内部文件 I/O 函数	367
11.3.4 内部字符串函数	369
11.3.5 Open 和 Save 会话框	369
11.3.6 DlgDirList 和 DlgDirSelect 函数	374
11.3.7 读取有效文件名	376
11.3.8 列表框函数的应用	376
第四节 无形式会话框	388
11.4.1 形式会话框和无形式会话框的区别	389
11.4.2 无形式会话框的应用	390
11.4.3 窗口与会话框的区别	395
第十二章 GDI——图形设备接口	401
第一节 GDI 基本原理	401
第二节 设备输出缓冲区	402
12.2.1 获取设备输出缓冲区把柄	402
12.2.2 获取设备输出缓冲区信息	404
12.2.3 GetDeviceCaps 函数的应用	405
12.2.4 设备尺寸	414
12.2.5 有关颜色问题	415
12.2.6 设备输出缓冲区属性	416
12.2.7 保存设备输出缓冲区	416
第三节 映射方式	419
12.3.1 设备座标与逻辑座标	420
12.3.2 设备座标系	421
12.3.3 显示区与窗口	421
12.3.4 MM_TEXT 映射方式	423
12.3.5 公制单位映射方式	426
12.3.6 “自由”映射方式	428
12.3.7 映射方式的应用	433
第十三章 Windows 图形基础	437
第一节 点	437
第二节 线	437
13.2.1 系统笔	439

13.2.2	笔的创建、选择和删除	440
13.2.3	设备依赖性	443
13.2.4	点间空隙问题	443
13.2.5	画图方式	444
13.2.6	ROP2 和颜色	448
第三节	实心图的画法	449
13.3.1	边框	451
13.3.2	Arc 应用	454
13.3.3	扇形图及其三角学	458
13.3.4	多边形及其填充方式	460
13.3.5	填充内部区域	461
第四节	刷与位图	464
13.4.1	位图刷的创建和使用	465
13.4.2	刷的排列	468
第五节	矩形、区域和剪辑	470
13.5.1	矩形	470
13.5.2	区域的创建与绘图	472
13.5.3	矩形区域的剪辑操作	473
13.5.4	区域的应用	474
第六节	GDI 函数的应用	478
第十四章	位图操作和图元文件	488
第一节	目标位图和文件位图	488
14.1.1	创建位图	488
14.1.2	单色位图格式	490
14.1.3	彩色位图格式	491
14.1.4	位图文件	491
14.1.5	位图的尺寸	492
第二节	内存设备输出缓冲区	492
第三节	位块传送	493
14.3.1	PatBlt 函数	494
14.3.2	位块传送坐标	496
14.3.3	BitBlt 位块传送	497
14.3.4	显示位图的函数	498
14.3.5	ROP 代码的应用	499
14.3.6	内存设备输出缓冲区的应用	501
14.3.7	颜色转换	504
14.3.8	映射方式转换	505
14.3.9	StretchBlt 函数的位图缩放功能	505
14.3.10	位块传送函数的综合应用	506

第四节	图元文件	510
14.4.1	内存图元文件	510
14.4.2	磁盘图元文件	512
14.4.3	图元文件的应用	513
14.4.4	资源图元文件	515
14.4.5	图元文件结构	518
14.4.6	关于图元文件	519
第十五章	文本输出和字体	522
第一节	简单文本输出	522
15.1.1	文本输出函数	523
15.1.2	文本的设备输出缓冲区属性	524
15.1.3	系统字体的使用	525
15.1.4	加灰字符串	525
15.1.5	GrayString 的使用	527
第二节	字体的背景	527
15.2.1	字体类型	527
15.2.2	字形	529
15.2.3	字体资源文件	530
15.2.4	关于“点”	531
15.2.5	逻辑英寸	532
15.2.6	字体结构	532
15.2.7	“逻辑 twips”映射方式	533
第三节	字体的创建, 选择和删除	534
15.3.1	逻辑字体结构	535
15.3.2	字形映射算法	539
15.3.3	可变字体的处理	539
第四节	字体选择	540
第五节	格式化文本	550
15.5.1	单行文本排列	550
15.5.2	文本段落	551
第十六章	数据动态交换	561
第一节	数据传递区	561
16.1.1	数据传递区标准数据格式	561
16.1.2	文本交换	562
16.1.3	读取文本	563
16.1.4	数据传递区的操作	564
16.1.5	数据传递区的打开和关闭	565
16.1.6	位图交换	565
16.1.7	图元文件交换	571

第二节	数据传送区的应用.....	575
16.2.1	使用多项数据	575
16.2.2	滞后复制技术	576
16.2.3	用户自定义数据格式	578
第三节	数据传递区观察程序.....	580
16.3.1	数据传递区观察程序链	580
16.3.2	数据传递区观察程序函数和信息	581
16.3.3	数据传递区观察程序	583
16.3.4	数据传递区中数据的装入与保存	586
第四节	数据动态交换.....	598
16.4.1	DDE 的组织	599
16.4.2	DDE 的初始化	599
16.4.3	数据交换“链”.....	600
16.4.4	数据动态交换的应用	600
第十七章	动态链接与动态库.....	607
第一节	基本概念.....	607
17.1.1	动态链接	608
17.1.2	EXEHDR 程序	608
第二节	动态库的构造.....	609
17.2.1	动态库程序	610
17.2.2	动态库初始化	613
17.2.3	动态库的出口	615
17.2.4	动态库的 DEF 文件	616
17.2.5	动态库源代码的编译	616
17.2.6	动态库函数的调用	616
17.2.7	远程函数前序代码	622
17.2.8	回调函数	623
第三节	近程指针与远程指针.....	624
第四节	动态库的其它限制.....	628
第五节	动态库的链接.....	629
第六节	输入库的应用.....	630
第七节	动态库在调试中的应用.....	631
第八节	没有输入函数的动态链接.....	632
第九节	动态库的数据段.....	633
第十节	资源动态库.....	633
第十八章	打印机和 Escape 函数的应用	639
第一节	假脱机打印技术.....	639
第二节	打印机设备输出缓冲区.....	642
18.2.1	获取 CreateDC 参数	643

18.2.2	打印机信息	645
18.2.3	DeviceMode 调用	651
18.2.4	打印机的图形缩放功能	652
第三节	打印技术基础	653
18.3.1	Escape 函数	653
18.3.2	打印机走纸控制	654
第四节	图形和文本的打印	656
18.4.1	打印控制核	658
18.4.2	设置异常终止函数	659
18.4.3	AbortProc 的应用	661
18.4.4	打印的异常终止	661
18.4.5	交互打印	664
18.4.6	出错处理	673
第五节	分块打印	674
18.5.1	分块技术初步	674
18.5.2	分块打印与异常终止函数	677
18.5.3	屏幕打印	681
18.5.4	键盘输入的截获	687
第六节	打印机和字体	689

1A
8-2 (15)

第一章 Windows 和应用软件开发

Microsoft Windows 是一个操作环境，一般在 IBM 个人计算机或兼容机上 DOS 3.0 版或更高版操作系统下即可运行。

对用户而言，Windows 提供了一个基于图形的多任务多窗口环境。在这个环境下能够运行为 Windows 编写的程序，以及一些现有的为 DOS 设计的程序，专门为 Windows 而设计的程序都具有一致的外部特征和命令结构。

对于程序开发者而言，Windows 提供了丰富的库程序，允许开发者易于实现清单、会话框、滚动杠以及组成友好的用户界面的其它图形元素。Windows 还支持一个用途广泛的图形程序设计语言，以及使用可选字体的格式化文本。程序设计者能够以与设备独立的方式对待键盘、鼠标、视频显示器、打印机、系统时钟和 RS-232 通讯口。Windows 程序可以在不同的硬件配置下运行。

第一节 Windows 的发展

Windows 最初由 Microsoft 公司于 1983 年 11 月颁布，在时隔二年之后的 1985 年 11 月，推出了 Windows 的最初版 Windows 1.03 版（1986 年 8 月），那是一种美国国内版。为了同欧洲版具有一致性，进行了少量改进。Windows 1.03 版包含用于欧式键盘的驱动程序，并增加了显示设备和打印机的驱动程序。Windows 1.04 版（1987 年 4 月）增加了主要用于支持 IBM PS/2 的行视频模式显示设备的相应功能。

在最初推出 Windows 的同时，其它一些软件制造商同样也推出了能够在多窗口和图形环境下运行的其它软件系统。这种环境的成功都取决于环境的质量和在该环境下运行应用程序的通用性。尽管在这方面 Windows 进展顺利，逐渐成为 PC 窗口环境的标准，然而它真正的发展是 1987 年 4 月 Microsoft 和 IBM 宣布推出 OS/2 和 OS/2 显示管理程序（Presentation Manager）。

OS/2 是在 IBM / Microsoft 联合开发协议下开发的，用于微处理器是 Intel 80286 和 Intel80386 保持方式的操作系统。当然，程序设计者也可以开发字符方式的应用程序，并不需要 OS/2 的显示管理程序。但是，显示管理程序为新的应用提供了一个较好的环境。OS/2 的显示管理程序在 IBM PC 系统应用结构（SAA）中起着重要作用，它是一项包括建立整个 IBM PC 系统的一致性用户界面在内的长期战略。

在 1987 年 11 月，Windows 2.0 版问世。为了与 OS/2 显示管理程序的用户界面一致，Windows 2.0 版对窗口的外部特征和用户界面做了相应的改变。最明显的变化是“重迭”窗口的使用，这与 1.01 版的“平面”式界面截然不同。同样，2.0 版还对键盘、鼠标接口进行了改善，包括对清单和会话框性能的加强。

Windows 划时代的发展是 1990 年 5 月 3.0 版的推出。它与以往任何一版都不同，首先是用户界面，显得操作与概念更形象化。最重要的突破是，Windows 3.0 版支持 Intel80286, 80386 和 80486CPU 的保护模式，使用户程序可以超过常规 640K RAM 的运

行空间，为未来的 PC DOS 奠定了基础。

第二节 基本概念

Windows 为用户和程序设计者提供了许多方便，这是传统的 DOS 环境下很难做到的。Windows 采用了许多新概念，这对程序开发者来说就有一个“术语刷新”过程。

1.2.1 “视觉界面”

用来描述 Windows 系统的术语看起来深奥难懂，但它们实际上是一些非常简单的概念。举世公认 Windows 提供了一个友好的“视觉界面”。这在某种意义上可以理解为 Windows 的用户能在屏幕上看到有意义的目标。

DOS 不具有视觉界面。就象早期销售的个人计算机不带键盘和显示器一样，用户看不到 DOS 的任何东西。在 DOS 装入计算机后，显示器上出现的仅仅是一个提示符，如图 1-1 所示。这个提示符实际上只告诉用户一件事：默认驱动器是 C 盘，等待用户键入命令。

C>

图 1-1 DOS 提示符

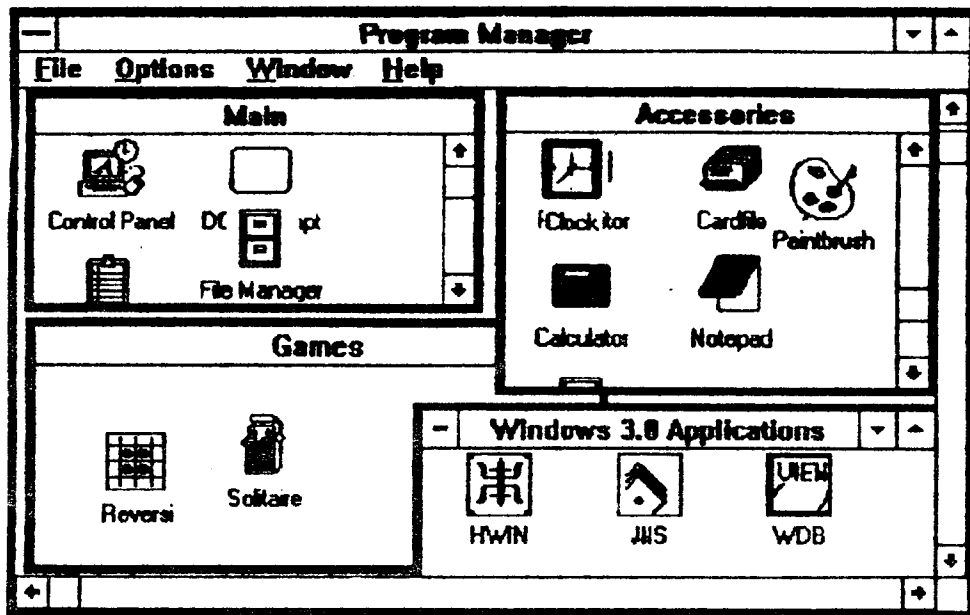


图 1-2 Windows 的初始屏幕