



清华松岗系列丛书

# Windows 95

# 实用编程与范例

木林森 等 编著



清华大学出版社



7
4

TP316.7  
/MLC/1

# Windows 95 实用编程与范例

木林森 高峰霞 奚红宇 王世航 编著

清华大学出版社

040312

(京)新登字 158 号

**Windows 95 应用编程与范例**

木林森 高峰霞 奚红宇 王世航 编著

本书对 Windows 95 的编程方法和编程技术作了全面系统的介绍。全书内容主要包括:Windows 95 概述、Windows 95 编程基础、消息与消息框、菜单、对话框、图标、光标、位图、绘图、文本输出、标准控件、公用控件、多进程与多线程、动态链接库、内存管理。本书最后一章对 DDE(动态数据交换)、OLE(对象链接与嵌入)、ODBC、多媒体、网络和 Internet 等内容作了简要介绍。此外,作者还精心设计了大量的编程实例。

全书内容新颖、语言浅显、通俗易懂,适用于不同层次、不同需要的读者,并可作为培训班的教材。

**版权所有,翻印必究。本书封面贴有清华大学出版社激光防伪标签,无标签者不得销售。**

JS353/32 书

出版者:清华大学出版社(北京 清华大学校内,邮政编码:100084)

责任编辑:王炼韧

印刷者:清华大学印刷厂

发行者:新华书店总店北京科技发行所

开本:787×1092 1/16 印张:20.25 字数:480 千字

版次:1997年6月第1版 1997年6月第1次印刷

书号:ISBN 7-302-02499-5/TP·1265

印数:0001—6000

定价:36.00 元

# 前 言

Windows 95 是美国微软公司于 1995 年 8 月推出的具有划时代意义的又一力作。Windows 95 的推出是个人计算机发展史上的一个重要里程碑,标志着个人计算环境从 16 位向 32 位的过渡。同以往版本相比,Windows 95 主要具有以下特点:

- Windows 95 是一种全新的多任务、多线程 32 位操作系统,支持 32 位应用程序和设备驱动程序。为了保持兼容性,Windows 95 同时还支持 16 位应用程序和设备驱动程序。因此,Windows 95 混合有 16 位和 32 位代码,从而使其不仅具有较好的兼容性,而且具有较高的性能。
- Windows 95 不再使用 DOS 作为基础环境,从而克服了早期 16 位操作系统所带来的种种烦恼。
- Windows 95 的 32 位文件系统实现了完全的统一,不再受 MS-DOS 文件名长度的限制,可以支持长达 255 个字符的文件名。
- Windows 95 使用虚拟设备驱动程序管理各种各样的系统资源,从而使得常规内存得以增大,性能得以提高。
- Windows 95 在调用所有 API 函数时,可以检查函数参数的有效性,从而减少了程序崩溃的可能性。而且由于 Windows 95 的进程都可以在自身的寻址空间内运行,这样一个进程的崩溃不会导致其他进程的崩溃,因此 Windows 95 系统更加坚固。
- Windows 95 最终放弃了 16 位段内存寻址方式,而采用 32 位平面内存寻址方式,极大地改进了内存管理技术。更为重要的是 Windows 95 可以分配给每个进程多达 4MB 的系统虚拟地址空间,为程序员编写应用程序带来了极大的方便。
- Windows 95 支持独立的消息队列和大量的公用控件(如工具栏和状态栏),提供了鼓励程序员编写更大程序的特性。
- Windows 95 用户界面更加友好,内嵌网络系统和 Internet 访问工具,支持可移动计算,支持即插即用设备,具有强大的多媒体支持功能。所有这些,使 Windows 95 的使用更加方便。

总之,作为 Windows 家族的成员,Windows 95 必将带给您一种全新的感受,使您工作起来更加简便快捷,帮助您梦想成真。

目前国内用户正纷纷转向 Windows 95,全国范围内正兴起一股 Windows 95 热潮。为了使国内广大用户能尽快掌握 Windows 95 的编程方法和编程技术编撰了本书。本书是作者在长期使用 Windows 系列软件的基础上,经过不断总结、提炼和实践才得以完成的。

本书对 Windows 95 编程所应掌握的基本内容和方法作了详细介绍。同时,为了帮助读者加深和巩固对每一部分的理解,作者精心设计了大量的编程实例,并将这些实例在 Visual C++ for Windows 95 上进行了严格的调试和测试。

此外,本书在介绍的过程中,对 Windows 95 与 Windows 3. x 的不同之处作了区别。因此,对于已有 Windows 编程经验的读者,在阅读时要注意这些区别。如果读者在阅读本书之前还未编写过任何 Windows 程序,那么在开始阶段最好要耐心一些,这是因为图形用户界面的编程是绝对值得花时间和精力去研究的。

本书主要由木林森、高峰霞、奚红宇和王世航负责编写,罗丽琼仔细审阅了全书,对全书提出了宝贵的修改意见。参与本书编写工作的还有郭建、陈小武、张玉南、郑妨、林志强。本书的录排工作由郭静负责。

由于时间仓促,加之水平有限,不当之处在所难免,尚望读者批评指正。

作 者

1996 年 9 月

# 目 录

<b>第一章 Windows 95 概述</b> .....	1
1.1 从 Windows 3. x 到 Windows 95 .....	1
1.1.1 Windows 3. x 的弱点 .....	1
1.1.2 从虚拟多任务到多任务 .....	1
1.1.3 维持与 Windows 3. x、DOS 的兼容性 .....	2
1.2 Windows 95 的目标和功能 .....	2
1.3 Windows 95 的结构 .....	3
1.4 与其他操作系统的比较 .....	4
1.4.1 Windows 95 与 Windows 3. x 的比较 .....	4
1.4.2 Windows 95 与 Windows NT 的比较 .....	4
1.5 小结 .....	4
<b>第二章 Windows 95 编程基础</b> .....	5
2.1 常用术语 .....	5
2.1.1 用户界面 .....	5
2.1.2 窗口 .....	5
2.1.3 图标和位图 .....	6
2.1.4 菜单、工具栏、状态栏和对话框 .....	6
2.2 Windows 95 API .....	7
2.3 命名规则 .....	7
2.3.1 函数命名规则 .....	7
2.3.2 变量命名规则 .....	8
2.4 Windows 95 应用程序结构 .....	8
2.4.1 Windows 95 应用程序的基本结构 .....	8
2.4.2 WinMain 函数 .....	11
2.4.3 窗口函数 .....	18
2.5 小结 .....	19
<b>第三章 消息和消息框</b> .....	20
3.1 消息及其处理 .....	20
3.1.1 WM_CHAR 消息 .....	20
3.1.2 WM_LBUTTONDOWN 和 WM_RBUTTONDOWN 消息 .....	23

3.1.3	WM_PAINT 消息 .....	25
3.1.4	WM_HSCROLL 和 WM_VSCROLL 消息 .....	27
3.1.5	WM_TIMER 消息 .....	28
3.2	消息框及其处理 .....	29
3.3	编程实例 .....	30
3.4	小结 .....	35
<b>第四章</b>	<b>菜单 .....</b>	<b>36</b>
4.1	菜单与菜单项 .....	36
4.2	定义菜单 .....	36
4.2.1	资源描述文件 .....	36
4.2.2	创建菜单 .....	37
4.3	在应用程序中访问菜单 .....	38
4.3.1	创建窗口类时指定菜单 .....	39
4.3.2	创建窗口时指定菜单 .....	39
4.4	处理菜单输入 .....	40
4.5	从应用程序中控制菜单 .....	41
4.5.1	使菜单项有效或无效 .....	41
4.5.2	检取和清除菜单项 .....	41
4.5.3	其他菜单项操作 .....	42
4.6	增加菜单加速键 .....	42
4.6.1	创建加速键表 .....	43
4.6.2	为菜单项增加加速键文字 .....	43
4.6.3	装入加速键表 .....	44
4.7	建立动态菜单 .....	45
4.8	范例程序 .....	46
4.9	小结 .....	52
<b>第五章</b>	<b>对话框 .....</b>	<b>53</b>
5.1	对话框的使用 .....	53
5.1.1	控件 .....	53
5.1.2	模态对话框 .....	53
5.1.3	非模态对话框 .....	54
5.2	对话框处理 .....	54
5.2.1	对话框窗口函数 .....	54
5.2.2	显示对话框 .....	55
5.2.3	创建对话框 .....	55
5.2.4	范例程序 1 .....	58

5.3	使用非模态对话框.....	62
5.3.1	处理非模态对话框与处理模态对话框的区别 .....	63
5.3.2	范例程序 2 .....	64
5.4	小结.....	69
<b>第六章</b>	<b>图标、光标和位图.....</b>	<b>70</b>
6.1	图标.....	70
6.1.1	图标的定义 .....	70
6.1.2	使用系统提供的图标 .....	70
6.1.3	使用自定义的图标 .....	70
6.1.4	在对话框中显示图标 .....	73
6.2	光标.....	73
6.2.1	使用系统光标形状 .....	73
6.2.2	使用自定义的光标形状 .....	74
6.3	位图.....	76
6.3.1	创建位图 .....	76
6.3.2	定义位图资源 .....	77
6.3.3	显示位图 .....	77
6.4	范例程序.....	80
6.5	小结.....	84
<b>第七章</b>	<b>绘图与文本输出 .....</b>	<b>85</b>
7.1	设备文本.....	85
7.1.1	访问输出设备 .....	85
7.1.2	设备文本属性 .....	87
7.2	虚拟窗口.....	87
7.2.1	虚拟窗口方法 .....	87
7.2.2	虚拟窗口的实现过程 .....	87
7.3	绘图.....	90
7.3.1	坐标系 .....	90
7.3.2	画图工具 .....	90
7.3.3	图形处理 .....	92
7.3.4	绘图范例程序 .....	95
7.3.5	其他绘图函数.....	101
7.4	映射模式 .....	103
7.4.1	约束映射模式.....	103
7.4.2	非约束映射模式.....	104
7.4.3	设置映射模式.....	105



7.4.4	定义窗口区域	105
7.4.5	定义视口	106
7.4.6	设置视口原点	106
7.4.7	映射模式范例程序	106
7.5	文本处理	114
7.5.1	设置文本和背景颜色	114
7.5.2	设置背景显示模式	114
7.5.3	获取文字属性	115
7.5.4	使用字体	121
7.6	小结	130
<b>第八章</b>	<b>标准控件</b>	<b>131</b>
8.1	列表框	131
8.1.1	列表框的定义	131
8.1.2	列表框风格	132
8.1.3	列表框处理	132
8.1.4	列表框的其他内容	139
8.2	编辑框	141
8.2.1	编辑框的创建	141
8.2.2	编辑框处理	142
8.3	滚动条	142
8.3.1	创建滚动条	143
8.3.2	处理滚动条	143
8.3.3	滚动条处理代码示例	145
8.4	复选框	146
8.4.1	创建复选框	146
8.4.2	处理复选框消息	148
8.4.3	复选框切换	149
8.4.4	复选框初始化	149
8.5	静态控件	150
8.5.1	LTEXT 控件	150
8.5.2	CTEXT 控件	150
8.5.3	RTEXT 控件	150
8.5.4	GROUPBOX 控件	151
8.6	单选按钮	151
8.6.1	创建单选按钮	151
8.6.2	处理单选按钮消息	152
8.6.3	其他处理内容	153

8.6.4	示例程序	153
8.7	标准控件补充说明	160
8.7.1	标准控件的创建	160
8.7.2	标准控件的处理	162
8.8	小结	162
<b>第九章</b>	<b>公用控件</b>	<b>163</b>
9.1	公用控件的种类	163
9.2	公用控件使用基础	164
9.2.1	公用控件的头文件	164
9.2.2	公用控件的初始化	164
9.2.3	公用控件的使用方法	164
9.3	工具栏	165
9.3.1	工具栏创建函数及其说明	165
9.3.2	工具栏位图的创建	167
9.3.3	工具栏处理过程	167
9.4	工具栏提示	169
9.4.1	工具栏提示的创建	169
9.4.2	工具栏提示产生的消息及其结构	169
9.4.3	工具栏提示处理	170
9.4.4	范例程序	171
9.5	上下控件	177
9.5.1	上下控件的创建与使用	177
9.5.2	上下控件消息	178
9.5.3	使用上下控件	179
9.5.4	旋转控件	180
9.5.5	旋转控件的使用	180
9.6	轨道条控件	181
9.6.1	轨道条风格	181
9.6.2	发送轨道条消息	182
9.6.3	处理轨道条通知消息	182
9.6.4	轨道条创建	183
9.6.5	轨道条的使用	183
9.7	进展条控件	184
9.7.1	发送进展条消息	184
9.7.2	进展条的创建	185
9.8	状态栏控件	185
9.8.1	创建状态栏	185

9.8.2	状态栏消息	185
9.8.3	使用状态栏	186
9.9	应用范例	188
9.10	制表控件	196
9.10.1	创建制表控件	196
9.10.2	发送制表控件消息	197
9.10.3	制表通知消息	198
9.11	树形查看控件	199
9.11.1	创建树形查看控件	199
9.11.2	发送树形查看消息	199
9.11.3	树形查看通知消息	202
9.12	公用对话框	203
9.13	小结	204
<b>第十章</b>	<b>文件输入和输出</b>	<b>205</b>
10.1	Windows 3. x 的基本文件操作	205
10.2	Windows 95 文件操作特点	206
10.2.1	基本操作	206
10.2.2	长文件名	206
10.2.3	文件时间	206
10.3	I/O 函数的详细讨论	207
10.3.1	CreateFile()函数	207
10.3.2	ReadFile()和 WriteFile()函数	210
10.3.3	SetFilePointer()函数	211
10.4	文件加锁和反加锁	212
10.5	有关文件的其他信息	213
10.5.1	获取文件属性	213
10.5.2	设置文件属性	214
10.5.3	获取文件大小	214
10.5.4	获取文件类型	215
10.5.5	获取文件的所有信息	215
10.5.6	获取文件全名	216
10.5.7	获取驱动器信息	216
10.5.8	获取卷册信息	217
10.5.9	创建临时文件	218
10.6	文件监视	218
10.7	GetOpenFileName()和 GetSaveFileName()函数	220
10.8	应用范例	223

• VII •

10.9	小结	231
<b>第十一章</b>	<b>多进程和多线程</b>	<b>232</b>
11.1	多进程	232
11.1.1	进程的概念	232
11.1.2	创建一个进程	232
11.1.3	终止一个进程	234
11.1.4	多进程的需求	235
11.1.5	多进程范例程序	235
11.2	多线程	243
11.2.1	多线程的概念	243
11.2.2	创建一个线程	244
11.2.3	终止一个线程	244
11.2.4	挂起和恢复一个线程	245
11.2.5	多线程的需求	245
11.2.6	多线程范例程序	246
11.3	同步	255
11.3.1	同步的场合	255
11.3.2	同步对象的类型	256
11.3.3	使用信号灯实现同步	257
11.3.4	其他同步机制函数	261
11.4	小结	261
<b>第十二章</b>	<b>动态链接库</b>	<b>262</b>
12.1	动态链接库的概念	262
12.2	Windows 3.x 动态链接库范例程序	263
12.3	Windows 95 与 Windows 3.x 动态链接库的区别	267
12.3.1	调用 DLL 中的函数	267
12.3.2	可执行文件的 DEF 文件	269
12.4	动态链接库的建立	269
12.4.1	库模块定义文件	269
12.4.2	DLL 初始化	271
12.4.3	共享数据	273
12.5	小结	275
<b>第十三章</b>	<b>内存管理</b>	<b>276</b>
13.1	内存寻址模式	276
13.2	全局堆和局部堆	277

13.2.1	局部堆	277
13.2.2	全局堆	278
13.3	使用 C 语言内存分配函数	280
13.4	堆内存	280
13.5	内存映像文件	281
13.6	虚拟内存	281
13.7	小结	281
<b>第十四章</b>	<b>在 Windows 95 上实现其他功能</b>	<b>283</b>
14.1	动态数据交换	283
14.1.1	基本概念	283
14.1.2	DDE 消息	285
14.1.3	DDE 管理库(DDEML)	286
14.2	对象链接和嵌入	290
14.2.1	部件对象模型	290
14.2.2	统一格式数据传输	291
14.2.3	复合文档	292
14.2.4	OLE 自动化	292
14.2.5	OLE 控件	293
14.3	多媒体	294
14.3.1	多媒体数据格式	294
14.3.2	播放函数	295
14.3.3	多媒体声音服务	295
14.3.4	多媒体控制接口	295
14.3.5	范例	296
14.4	网络和 Internet	303
14.4.1	编程接口	303
14.4.2	Internet	304
14.5	ODBC	306
14.5.1	ODBC 体系结构	306
14.5.2	访问数据库	307
14.6	小结	311

# 第一章 Windows 95 概述

**本章学习目的:**

本章将学习 Windows 95 的主要功能特点,学习 Windows 95 与 Windows 3. x、Windows NT 的区别。

## 1.1 从 Windows 3. x 到 Windows 95

Windows 95 是 Microsoft 公司继 MS-DOS, Windows 3. x 和 Windows NT 之后推出的新一代操作系统。Windows 95 博采众长,融合了其他操作系统的许多先进技术,达到了较高的技术境界。

作为一个真正的 32 位多任务、多线程操作系统,Windows 95 不再使用 DOS 作为基础环境,而且克服了早期 16 位操作系统所带来的种种烦恼。

### 1.1.1 Windows 3. x 的弱点

Microsoft 公司 1990 年推出的 Windows 3. 0,由于界面友好,操作方便,内存管理突破了 MS-DOS 的 640KB 的限制,又具有动态数据交换和丰富的 API,因而受到了广大用户的青睐。在随后推出的 Windows 3. 1 中,由于引入了对象链接和嵌入(OLE)、TrueType 字型等新技术,因此获得了空前的成功。但是,Microsoft 公司清楚地知道 Windows 3. x 的症结所在。而且随着计算机技术的不断发展,Windows 3. x 的弱点也日益暴露。例如,在 Windows 3. x 中,除原有 DOS 应用程序外,所有 Windows 应用程序都无优先权的协同多任务的运行机制。

Windows 95 的推出,将取代 MS-DOS 和 Windows 3. x 成为新一代操作系统的主流。Windows 95 具有许多新的技术和特征能满足更多用户的需要。在 API 方面,Windows 95 不仅丰富了 Windows 3. x 原有的 API,而且几乎所有的 API 都是 32 位的,这样就能够充分发挥和利用 32 位保护模式芯片的功能。

### 1.1.2 从虚拟多任务到多任务

从 Windows 3. x 过渡到 Windows 95,二者最主要的区别之一就在于多任务的实现方式。Windows 3. x 在进行多任务切换时,使用的是非优先方法,即虚拟多任务处理方式。操作系统本身没有调度功能,Windows 3. x 任务必须手工将控制返回给调度程序,这样使得

一个应用程序有可能独霸 CPU。从编程的角度来说,由于 Windows 3. x 程序保留了对 CPU 的控制直至其自身决定放弃它,因此可能在出现错误的任务时将 CPU 独占,从而导致死机。

与此相反,Windows 95 具有多任务的调度功能,使用的是优先分时式任务切换。在这种方式下,正在运行的任务可以被 Windows 95 自动中断,将 CPU 调度给下一个任务(如果还有任务的话)。这样可以防止 CPU 被某个任务独占。在程序设计人员的眼中看来,这无疑是在向前迈了一大步。

### 1.1.3 维持与 Windows 3. x、DOS 的兼容性

与 Windows 3. x 和 DOS 兼容是 Windows 95 的主要设计目标之一。虽说 Microsoft 使 Windows 95 内核脱胎换骨,焕然一新,但并未割断与 MS-DOS 及 Windows 3. x 的兼容性。在 Windows 95 上可以运行四种类型的应用程序: DOS 应用程序、Windows 3. x 应用程序、Windows NT 应用程序以及专门为 Windows 95 编写的 Windows 95 应用程序。Windows 95 不仅配备有 32 位 API(Win32),而且还配备有 Windows 3. x 用的 16 位 API(Win16)。

此外,Microsoft 公司在开发 Windows 95 时,就要求做到用与 Windows 3. x 同等程度的硬件资源来运行 Windows 95,从而为 Windows 95 的推广奠定了良好的基础。

## 1.2 Windows 95 的目标和功能

Microsoft 公司在设计 Windows 95 时,针对 MS-DOS、Windows 3. x 存在的问题、用户的要求和市场的动向,进行了慎密的研究,制定出 Windows 95 的目标和实现这些目标应具备的功能。

#### (1) 使用更加方便

- 支持即插即用(Plug and play)设备
- 用户界面更友好
- 支持长文件名
- 内嵌连网功能

#### (2) 各种功能都有提高

- 提供完全的多任务功能
- 更有效地利用内存
- 提高系统的坚固性
- 实现可移动计算
- 提高打印速度
- 提高多媒体性能
- 增加 MS-DOS 应用程序的可用内存

#### (3) 保持兼容性

- 同现有 MS-DOS 和 Windows 应用程序保持兼容
- 同现有硬件兼容
- 具有和 Windows 3. x 一样或更高的性能
- 安全、简便的升级和过渡

## 1.3 Windows 95 的结构

Windows 95 的结构可以从以下几个方面加以介绍:

### 1. 混合有 16 位码段和 32 位码段

Windows 95 虽然被称为 32 位操作系统,但在系统中仍残留一些 16 位码段。这是从兼顾兼容性和性能这两个方面出发而采取的措施。既要用少的内存,又要有高的兼容性,还要实现更高的性能,用一种方式很难全部满足,16 位/32 位码段的结合无疑是一个明智之举。

### 2. 可以运行 MS-DOS, Win16, Win32 应用程序

对 MS-DOS 应用程序的支持基本上和对 Windows 3. x 的支持一样,都是通过利用 386 以上处理器的特权方式虚拟机功能来实现的。

### 3. 单独的 MS-DOS 应用方式

对于直接访问硬件等系统资源的应用程序,单独的 MS-DOS 应用方式是非常有用的。

### 4. 运行 Win32 应用程序

同运行 Win16 应用程序相比,运行 Win32 应用程序具有以下优点:具有优先权的多任务功能,具有 32 位的 Win32 API,支持长文件名,每一应用程序有自己的消息队列,有平面线性地址空间,能进行存储器保护。

### 5. 多任务和多线程

在 Windows 95 中,实现了完全的多任务功能,此外还使用名为线程的调度链接单位。所谓线程(thread)是可执行代码的不可拆散的单位,所有进程必须具有至少一个线程。MS-DOS 和 Win16 应用程序使用各自的单一线程,而 Win32 应用程序则可执行多线程,这意味着每个进程可能具有其自身的两块或多块在同时运行,这样就可编写出效率非常高的程序。

### 6. 32 位文件系统

Windows 95 文件系统实现了完全的统一,并进行了 32 位码段化,所有文件系统由 IFS 进行一元化管理。其最明显的特征是文件名的长度不再受 MS-DOS 八字符文件名/三字符扩展名的限制,可以长达 255 个字符。

### 7. 虚拟设备驱动程序(VxD)

VxD 用于管理各种各样的系统资源,这可以提高系统性能,因为一些占用常规内存的驱动程序也被进行了 VxD 化,使得可用的常规内存增大。

### 8. 系统更加坚固

Windows 95 在调用所有 API 时,可以检查参数的有效性,使得 Win16 应用程序崩溃的概率大大降低。当然 Win32 应用程序由于其进程都可以在自身的寻址空间内运行并与其他进程隔离开来,所以当一进程崩溃时,其他进程将不受影响。也就是说一个出现了错误的



程序不会造成整个系统的失效,那么 Windows 95 与 32 位应用程序也更不容易崩溃了。Windows 95 在设计时就力图提高各部分的坚固性。

## 1.4 与其他操作系统的比较

### 1.4.1 Windows 95 与 Windows 3. x 的比较

从用户角度来看,Windows 95 主要是在四个方面与 Windows 3. x 不同,即用户界面的改变、窗口风格的改变,增加新控件成分和 DOS 不再是必需的。

从程序设计人员角度来看,Windows 95 和 Windows 3. x 之间主要有两个区别。首先 Windows 95 支持 32 位平面寻址和虚存,而 Windows 3. x 使用的是 16 位分段寻址;第二个主要区别在于多任务的实现方式。

### 1.4.2 Windows 95 与 Windows NT 的比较

Windows NT 是 Microsoft 公司推出的基于 Windows 的操作系统。Windows NT 与 Windows 95 在许多方面都一样,例如,都支持 32 位平面寻址,都支持基于线程的多任务,都支持基于控制台的用户界面。但是 Windows 95 不是 Windows NT,Windows NT 重视可移植性,支持各种 CPU,而 Windows 95 则专门针对 Intel 386 以上的 CPU,以利于提高性能;Windows NT 支持 OS/2 和 POSIX 应用程序的执行,而 Windows 95 不支持;Windows NT 需要大量内存(至少 12~16MB),而 Windows 95 的目标则是在 4MB 的 386 机器上也能工作。

总之,Windows NT 着眼于服务器等用途,而 Windows 95 则注重使大多数 Windows 3. x 的用户和初学用户不必对机器升级也能舒适地使用。

## 1.5 小结

Windows 95 是一个 32 位的多任务、多线程操作系统,Windows 95 支持两种方式的多任务:基于进程的多任务和基于线程的多任务。Windows 95 支持全 32 位寻址和虚存,很好地利用了保护模式芯片的特性,使得程序设计人员解除了要处理分段内存的负担。与 Windows NT 相比,Windows NT 侧重于作服务器,而 Windows 95 侧重于在 Intel 386 系列机器上发挥更大的作用,满足更多普通用户的要求。