Microsoft Windows 95

开发者必读

Microsoft Windows® 95 developer's Guide

这是在Windows 95下开发32位应用程序的权威著作！

[美] Stefano Maruzzi 著
翟炯 石秋云 译
石祥生 校

所附光盘提供了100多个可借鉴的范例

- 关键的系统信息存储在Registry数据库中
- 灵活而完美的存储管理技术
- 强有力的线程、消息、对话框的处理能力

ZD PRESS

電子工業出版社
PUBLISHING HOUSE OF ELECTRONICS INDUSTRY

Microsoft Windows 95 开发者必读

Microsoft Windows 95 Developer's Guide

〔美〕 Stefano Maruzzi 著

翟 炯 石秋云 译

石祥生 校

ZIFF-DAVIS ZD PRESS
電子工業出版社

038803

内容简介

本书全面系统地介绍了在 Windows 95 下开发各种应用程序所必需的基本和先进的编程技巧，其范围包括用户界面特点（如弹出菜单、窗口、对话框、特性表、菜单条等）；通过拖放操作实现外壳集成；处理存贮器管理、异常和 DLL 的各种新方法；图形设备接口；类、控制、资源文件、多线程、进程间通信等。本书着眼于将作者丰富的编程实践经验介绍给读者。附带的 CD－ROM 内包含了 100 多个范例，既可以加深对各种编程技巧的理解，又可以直接用于读者所编写的应用程序。全书对 Win16 和 Win32 的各种编程方法进行了比较，将在 Windows 95 编程中容易犯的各种错误介绍给读者，以便减少读者在学习过程可能走的弯路。本书是开发 Win32 应用程序的一本理想指导书。

Microsoft Windows 95 开发者必读

The Microsoft Windows 95 Developer's Guide

［美］ Stefano Maruzzi 著
翟 炯 石秋云 译
石祥生 校

特约编辑：张成全

*

电子工业出版社出版
北京市海淀区万寿路 173 号信箱（100036）
电子工业出版社发行 各地新华书店经销
北京顺义天竺颖华印刷厂印刷

*

开本：787×1092 毫米 1/16 印张：47.25 字数：1145 千字
1997 年 2 月第一版 1997 年 2 月第一次印刷
印数：3,000 册 定价：88.00 元
ISBN 7-5053-3839-0/TP.1652
著作权合同登记号
图字：01-96-0712

目 录

第一章　Win32 软件开发

- Microsoft Windows 的发展过程
- 操作平台的识别
- 32 位编程简介
- Windows 的硬件要求
- 系统信息的管理
- 占先多任务对开发的影响
- Windows 3.x 的多任务
- 异步输入方式
- 存储器管理
- 异常处理程序
- 存储器练习

微软公司(Microsoft)的 Windows 是该公司为配备 x86 微处理器的 Intel 个人计算机推出的第一个多任务操作系统，其第一个版本是在 1985 年 11 月推出的 Windows 1.01 版，在那个时候，几乎所有的个人计算机都使用 Intel 8088/86 微处理器。第一个 80286 系统是 IBM PC AT，是在 1984 年 8 月推出的。

当时，Windows 并没有立即得到广泛的采用，其主要原因是固有的硬件限制和 DOS 操作系统的有限功能尚能适应当时的需求。其它方面的原因包括：图形分辨率和处理器速度不够高；具有 320×200 像素和 4 种颜色的 CGA 标准由于成本高而不能得到广泛的普及；与八十年代后期强大的工业实力相比，软件生产几乎以一种不成熟的艺术方式进行。Windows 已经连续地推出了很多版本，其中的每一个版本都具有更强的功能和更好地使用现有硬件的能力。

目前的情况已发生了根本性的变化，个人计算机已配备了 80486 和 Pentium 处理器，80386 很快就变得过时了，台式计算机可以轻而易举地处理 800×600 像素的图形分辨率，使用至少 16 种颜色。事实上，大量的各种系统可以处理更高的图形分辨率和 256 种颜色，甚至高达一百万种颜色。鼠标器已成为每一台新型个人计算机的标准和不可分割的组成部分。数百兆字节的硬盘已是司空见惯。硬件性能不断地提高，但价格却不断地降低，其结果，使得个人计算机的使用几乎深入到了每一个领域和占领了所有的市场。

1992 年 4 月推出了 Microsoft Windows 3.1，在 1993 年 8 月又推出了 Microsoft Windows NT 3.1，紧接着在 1993 年 12 月份，微软公司正式宣布开发一种新的 32 位版本的 Windows 环境。在 Windows NT 3.1 版环境中，第一个出现的是 32 位 API(应用编程接口)，但对这种 Win32 的广泛接受只有在普及使用 Microsoft Windows 95 以后才能成为现实。这种 32 位版本的程序最初命名为 Chicago，后来又更名为 Microsoft Windows 95，在本

书中将称它为 Win95。

Microsoft Windows 的发展过程

最初，开发一个 Windows 应用程序就意味着要使用一种高级编程语言（一般情况下总是采用 C 语言）和与该环境的软件开发包（SDK）进行交互操作，这些功能统称为应用编程接口（API）。

当 Windows API 随着 Windows NT 的出现而发展成 32 位时，就将它称为 Win32，而老版本的 API 则称为 Win 16。表 1.1 全面地介绍了 Microsoft Windows 的发展过程。

表 1.1 Microsoft Windows 的发展过程

产品	推出日期	码长	API	应用程序
MS Windows 1.x	1985.11	16 位	Win16	16 位
MS Windows/386	1987.9	16 位/32 位	Win16	16 位
MS Windows 2.x	1987.12	16 位	Win16	16 位
MS Windows 3.0	1990.5	16 位/32 位	Win16	16 位
MS Windows 3.1	1992.4	16 位/32 位	Win16	16 位
MS Windows for Workgroups 3.1	1992.11	16 位/32 位	Win16	16 位
MS Windows for Workgroups 3.11	1993.11	16 位/32 位	Win16	16 位
MS Windows 3.11	1993.12	16 位/32 位	Win16	16 位
MS Windows NT 3.1	1993.8	32 位	Win32	32 位/16 位
MS Windows NT AS 3.1	1993.8	32 位	Win32	32 位/16 位
MS Windows NT Workstation 3.5	1994.10	32 位	Win32	32 位/16 位
MS Windows NT Server 3.5	1994.10	32 位	Win32	32 位/16 位
MS Windows NT Workstation 3.51	1995.7	32 位	Win32	32 位/16 位
MS Windows NT Server 3.51	1995.7	32 位	Win32	32 位/16 位
MS Windows 95	1995.8	32 位/16 位	Win32	32 位/16 位

在该表中，码长栏是指构成 Microsoft Windows 环境的程序的特点，API 栏是指软件开发人员在相应的平台上开发本机应用程序时可以使用的应用编程接口，应用程序栏是指每一种 Microsoft Windows 版本所支持的程序的特征。

Win32 代表了 API 总体质量的最重要改进，特别是随着 Microsoft Windows 95 的推出，它将日益成为一种普及的目标平台。Microsoft 公司打算推出一种供 Win32 使用的统一开发工具包，它可以适用于 Windows NT 和 Windows 95 支持的所有硬件平台。

在今后的几年中，将会完全合并 Microsoft 的二个 32 位 API 子集。目前，实现 NT 和 Win95 中的 Win 32 时是存在一些差别的，必须非常小心地去处理这些差别。自从 1992 年 7 月第一次公开 Windows NT 以来，它就全面地支持 Win 32，后来当 1993 年 8 月推出 Win-

dows NT 时，又重新进行定义。而另一方面，Windows 95 却将它的作用范围限制于一种 Win 32 子集，该子集曾经被称为 Win 32c。然而，已经要求 Microsoft 的工程师们将新的 Win 32 API 功能增添到 Win 95 中，而这些功能在NT 中是没有的。至于所以要进行这种开发的主要原因是在 Microsoft Windows 95 中采用了面向对象的用户接口。在将来，即使是 NT 也将包含 Microsoft Windows 95 对 Win 32 的所有扩充内容，在它的下一个版本（目前命名为 Microsoft Cairo）中将要做这方面的工作。表 1.2 对 Win 32 的两个当前版本的各个方面进行了比较。

表 1.2 Microsoft Windows 95 和 NT 当前版本的 Win 32 API 比较

Win 32 特点	MS Windows 95	MS windows NT
32 位内存管理	√	√
文件映射	√	√
联网	√	√
OLE 2.x	√	√
邮件槽和命名管道	√（仅客户机方面）	√
Win 32 线程	√	√
先进的 GDI	√（大多数）	√
远程过程调用	√	√
MS Windows 95 UI	√	只有公共控制
GDI 变换		√
事件登录		√
安全性		√
Unicode		√

在以前，API 总是与特定版本的 Microsoft Windows 紧紧捆绑在一起的；目前，Win32 具有若干种形式，每一种形式都有细小的差异，以适应这二种操作系统。这就是最新的 Windows 编程方法所具有的最有创造性的一个特点。

操作平台的识别

到目前为止，还没有涉及与 Win 32s 有关的内容，这是 Win 32 的另一个子集，其目的是开发一种 32 位的程序，通过相应的支持组成部分，它甚至可在 Microsoft Windows 3.x 系统中运行。这种建议引起了这样一个问题：Win 32 的开发工作应该放在哪一个目标平台上？要回答这一问题，首先要开发一小段简单的应用程序，名为 Platform，将它作为探讨 Win32 API 的一种方法。该应用程序的目的是将基础操作系统告诉用户。

有二个函数可以知道当前正在运行的是哪一个版本的 32 位 Windows 系统，它们是：GetVersion()和 GetVersionEx()。最好专门使用第二个函数，因为它比第一个函数更清晰和更富有成效。下面是这二个函数的语法：

```
DWORD GetVersion(void);
DWORD GetVersionEx(LPOSVERSIONINFO osvi);
```

虽然 GetVersionEx()依赖于数据结构 OSVERSIONINFO，但二个函数都能返回一个 DWORD(双字)。*GetVersion*()在返回值中包含的信息允许你去检测运行该程序的 32 位平台，但只有在执行一小段操作并提取出标识其差别的有关位后才有可能，如下列程序段所示：

```
...
DWORD dwVer;
WORD wHi;
int i;

dwVer=GetVersion();
wHi=HIWORD(dwVer);
i=wHi >> 14;
switch(i)
{
    case 0:
      strcpy(szString,"MS Windows NT");
      break;

    case 1:
      strcpy(szString,"MS Windows 3.x-Win32s");
      break;

    case 3:
      strcpy(szString,"MS Windows 95");
      break;
}
...
```

通过检查返回的高字中的最后二位就能进行识别，该最后二位的二进制值是 00 就表明是 Windwos NT，是 01 就表明是 Win 32s，是 11 则表明是 Win 95。在另一方面，GetVersion Ex()则在 dwPlatformId 中直接返回所需的信息，如下所示：

```
OSVERSIONINFO osvi;
DWORD version;
osvi.dwOSVersionInfoSize=sizeof(osvi);
GetVersionEx(&osvi);

switch(osvi.dwPlatformId)
{
    case VER_PLATFORM_WIN32s:
    {
        strcpy(szString,"MS Win32s on MS Windows 3.1");
    }
        break;
```

```
        case VER_PLATFORM_WIN32_WINDOWS:
        {
            strcpy(szString,"MS Windows 95");
        }
            break;

        case VER_PLATFORM_WIN32_NT:

            strcpy(szString,"MS Windows NT");
        }
            break;
}
...
```

通过运行 Platform（图 1.1），你立刻就会对 Windows 95 的某些新特点感兴趣。你肯定会注意到一个彩色光标、新的菜单外观和 Windows 的其它结构成份，以及可以覆盖客户区的背景位图。在下面各章中将对这些特点和其它的一些特点进行解释。

供本书所用的光盘中，程序清单 Listing 1.1 包含了 Platform 的源程序，它着重于能运行 Windows 32 位程序的 Win 32。

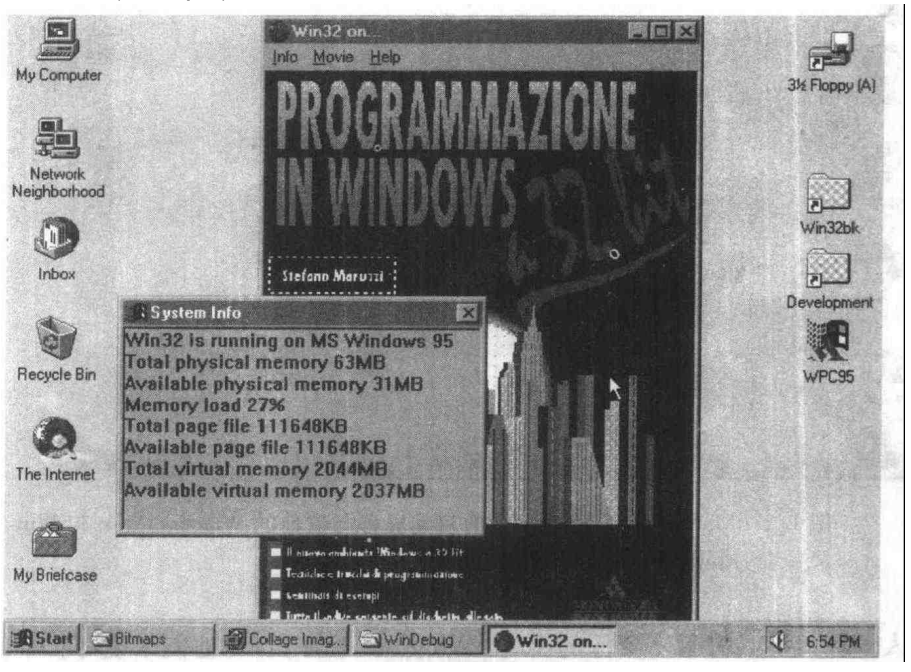

图 1.1　在 MS Windows 95 中运行的 Platform

32 位编程简介

尽管存在着 Win 32s，而且它在不断地发展，但 Microsoft 公司支持的主要的 32 位平台是 MS Windows 95，其次是 Windows NT. Microsoft Win 32 是构造具有下列特点的程序的起点：

- 执行应用程序时硬件设备的完全独立性
- 图形用户接口
- 在 Microsoft Windows 95 和 Microsoft Windows NT 之间的透明可移植性，对于支持 Microsoft Windows NT 的 RISC 硬件平台也是可移植的。
- 面向对象的用户接口
- 高性能的占先多任务和多线程
- 先进的多媒体支持(声音、图像、视频等)
- 利用"对象连接和嵌入"(OLE)技术的多应用对象倾向

这种对 Win 32 的简要综述很容易地就指出了在进行 32 位开发时的各种创新可能性。Win 32 可以不必借助于 MS-DOS 系统服务就能直接被控制和处理个人计算机硬件资源的操作系统使用。在图 1.2 中，引导以后就可立即看到 Windows 95 的面向对象的用户接口。

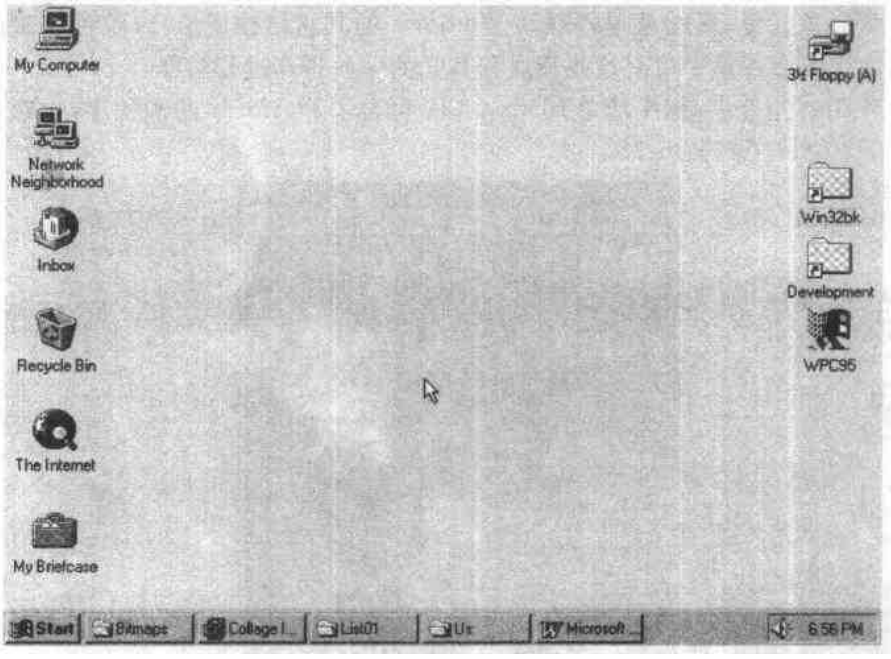

图 1.2 Microsoft Windows 95 的启动屏幕与以前的 Windows 3.x 和 Windows NT 具有明显的差别

在个人计算机界具有很多资深的 16 位 Windows 程序员，对他们来说，向 32 位编程的过渡是一个很好的机会，他们可以重新审查所有的程序清单，甚至可以通过增添新的功能而得到增强。然而，对于很多新的程序开发人员，将要求他们进行 Windows 编程，这是由于 32 位版本 Windows 的性能和新的 Win 32 API 的能力决定的。无论你的背景是什么，向 Win32 的过渡都需要一个学习过程。

最后，Win 32 真正意味着占先的多任务、多线程和线性寻址的存储器管理，Windows 95 的新接口将强制你对你的程序的外观和行为重新进行定义，这往往意味着要对管理用户交互的逻辑进行修改。