OsborneMcGraw-Hill

WINDOWS 3.1

编程实例详解

Windows 3.1 Programming

INCLUDES 3.5"DISK

[美] William H. Murray, Ⅲ和Chris H. Pappas 著

房贺祥　寇国华　等译

McGraw-Hill

電子工業出版社

Windows 3.1 Programming

Windows 3.1 编程实例详解

〔美〕William H. Murray，Ⅲ和 Chris H. Pappas 著

房贺祥 寇国华 等译

电子工业出版社

(京)新登字 055 号

内 容 简 介

本书由浅入深由简易到复杂，通过剖析若干 Windows 程序实例，详细讲解了 Windows 的编程方法。本书首先解释和定义了 Windows 的术语、环境及其各种功能。然后从编写简单的 Windows 程序开始，讲解了控制 Windows 窗口，设计图标、光标和位图，开发菜单和键盘加速器、对话框、使用字体，设计饼图、棒图以及折线图，设计动画等。本书还特别通过若干实例，讲解了使用面向对象的程序设计方法开发 Windows 程序。附录中提供了 Windows API 函数、Microsoft 基类库以及 Borland 面向对象的 Windows 类库项。借鉴书中的实例，可生成自己的 Windows 程库。

本书是学习和开发 Windows 软件的必备工具书，也可作为大中专院校师生的参考用书。

参加本书翻译的还有吕梅、唐红、周明慧、吴丽雯、王新联、张彤、范庆年、姜小玉、殷韶、顾红、苏金树等。

Windows 3.1 Programming
〔美〕William H. Murray, Ⅲ 和 Chris H. Pappas 著
Osborne McGraw-Hill 1992 年出版

Windows 3.1 编程实例详解
房贺祥 寇国华 等译
责任编辑 路 石

*

电子工业出版社出版(北京市万寿路)
电子工业出版社发行 各地新华书店经销
河北省望都县印刷厂印刷

*

开本：787×1092 毫米 1/16 印张：38.25 字数：979 千字
1995 年 1 月第一版 1995 年 4 月第一次印刷
印数：6000册 定价：78.00元(含盘)
ISBN 7-5053-2894-8/TP・963
著作权合同登记号图字：01-1995-171

目　录

序　言

我们生活在一个多么激动人心的时代！Windows 3.1 提供了一条通往下一代图形应用的令人振奋的道路。或许还没有任何其它产品象 Windows 3.1 那样受到热切的期盼。Windows 3.1 将软件开发人员从 DOS 的内存限制下解放出来。现在是学习 Windows 编程技术和培养读者自己进入未来的时刻了。

购买本书和所附带的盘，说明读者对 Windows 3.1 环境感兴趣，而且需要在这个领域发展编程能力。

我们假定读者在计算机中已安装 Windows 3.1、Microsoft C/C++(V7.0 或其以上版本)编译器或 Borland C/C++(V3.1 或其以上版本)编译器以及与上述编译器相关的 Windows 工具，且有一定的 C 或 C++编程经验。

本书是这样编排的：前面的章节解释和定义 Windows 的术语、操作环境和各种功能，以此为读者打下基础，并用简单的程序来说明这些定义和概念。当读者阅读到后面的章节时，这些简单的概念可以和后面的内容相联系以组成更复杂的程序。

如果读者初学 Windows 编程，则从本书的第一章开始阅读。本书的每一章都在前一章所提供内容的基础上提高一步。前几章的实例尽可能短小，这是由于笔者想给读者讲述编程的概念而不是堆积大量的程序编码。读者越往后看，程序实例将变得越复杂。这是因为笔者以前面的程序作为模块以进行更专业的程序设计。坚持到底！当读者读完本书后，将会非常惊奇地发现自己能干什么！

如果读者是一位熟练的 C/C++或 Windows 程序员，则可以迅速地将本书浏览一遍，以学习有关 Windows 3.1 新的特点，诸如实型字模技术和多媒体声音等。此外，一定要看一看附带盘上每一子目录中的 README 文件，以便获得额外的帮助和建议。

这是一个多么激动人心的编程时代——读者正进入一个全新的编程环境！

致　谢

首先要感谢 Osborne/McGraw - Hill 公司所有的同仁在本书准备出版过程中所给予的帮助，对每一项计划他们都给予了专业性指导。作者特别感谢 Frances Stack 女士，在本书出版过程中，她给予了特殊的关注和指导。

还要特别感谢 Microsoft 公司的同仁在软件和技术问题上所给予的帮助。Windows 3.1 是一个稳定的产品，该产品建立在经得起时间检验的基础之上。祝贺 Microsoft 提供了卓越的操作环境和 Windows 产品。

这本书是在两台 50MHz Dell 450DE 80486 计算机上起草和编辑的。该计算机系统包括 16MB 内存、一个 320MB 硬盘驱动器、一个 SVGA 监视器以及一个 Microsoft 鼠标器。书稿是用 Microsoft 的 Windows Word 程序编辑的。所有的开发工作都是用 Microsoft WINDOWS 3.1 完成的，程序用 Microsoft C/C++(V7.0)和 Borland C/C++(V3.1)编译器编译和测试的。所涉及的其它软件还有 Microsoft Windows 的 SDT(Software Development Toolkit，软件开发工具包)和 Borland 的 AFT(Application Framework Toolkit，应用框架工具包)。感谢硬件和软件厂商，将其提供的产品集成得如此完美。

第 1 章　Windows 引言

Microsoft Windows 是 Microsoft 公司以图形为基础的操作环境的顶峰。该环境将指向并选中控制、弹出式菜单以及专为 Wirdons 编写的应用程序，同样也可作为标准的应用程序与 DOS 运行的能力结合在一起。

1.1　Windows 是什么

Microsoft Windows 是一个可在 MS－DOS 下运行的操作环境，也是一个以图形为基础的多任务窗口环境。所有专为 Windows 编写的程序均具有一致的外观和命令结构，这使得很容易掌握新的 Windows 应用程序。

对于开发应用程序而言，Windows 提供了大量的嵌入式子例程，使得很容易实现弹出式菜单、滚动条、对话框、图标以及许多其它的用户友好图形界面的功能。从 Windows 3.0 开始，用户已经可以利用先进的对话框控制、菜单类型和“用户绘图控制”(owner－draw controls)功能。利用 Windows 提供的扩展图形编程语言，很容易规范化一个应用程序，并输出各种字模和色调的文本。

Windons 还允许应用程序开发人员以各种与设备无关的方式处理视频显示、键盘、鼠标器、打印机、串行口和系统计时器等。这就使得同一应用程序可在不同的硬件配置下运行。

随着 Windows NT(New Technology)32 位操作系统的发放，Windons NT 用户将可以使用先进的全优先权以及基于(thread－based)多任务的功能。这种新操作系统的设计，除适用于 80386 和 80486 处理器外，还特别适合于 RISC 结构。

Windows NT 是实现先进计算环境(Advanced Computing Environment ACE)设想的关键部件。它的设计成功，为以多处理器为基础的系统提供了一种开放式的标准的先进计算环境。Windows NT 对两种平台提供完全的支持：以 386/486 为基础的 PC 机和以 MIPS RISC 为基础的系统。

1.2　历史回顾

20 世纪 40 年代后期，计算机使用按今天的标准看来，非常古老，需要用户有大量打印输出的硬拷贝设备。使用阴极射线管(CRT)的首批计算机中的一台是由 MIT(麻省理工学院)在 1950 年研制的，其目的是研究飞行器的稳定性和控制问题。促使今天看来很普通的显示设备与计算机相连的原因是，需要缩短用户输入和计算机输出的时间。飞行器的研究也刺激了 20 世纪 50 年代赛其(SAGE)空中防御系统的研制，该系统可将雷达上的反射脉冲转换为粗糙的计算机产生的图象。这一系统也是第一个使用光笔在显示屏幕上选择符号的系统。

发展到 20 世纪 60 年代初，一名麻省理工学院的物理学博士生研制了线型绘图系统(Sketchpad line－drawing)。该系统可使用户通过将光笔指向屏幕上的点的方式绘图。这样，图形系统就可以在各光点间绘制直线或多边形。由此，简化了复杂图形的制作和简单目标的复

制。

早期的CRT能够在显示屏幕上任意两点间画一条直线。然而，由于图象消失得非常快，所以需要每秒钟重画多次。在60年代后期，这种绘图方式需要存储直线终点的内存和迅速重画直线的硬件都非常昂贵。例如，在1965年，IBM引进了第一次大量生产的这类用于图形显示的CRT，仅显示器本身每台造价就高达10万美元。读者很容易地理解为什么当时这种设备安装得不多。

三年以后，泰克(Tektronix)公司研制了第一台存储管型CRT。这种CRT可以保留所画内容，直到用户不再需要为止。由于这种显示器可以省去昂贵的内存和重画硬件，使得每台显示器的价格降到1万5千美元。靠这种价格，泰克公司的显示器很快就成为最成功的产品。

由于70年代存储器和硬件逻辑电路的价格急剧下降，所以图形开发环境进一步得到了改善，由此导致了大量生产存储器-增强型光栅扫描显示器。光栅扫描显示器能够产生看起来十分逼真的渐变而带彩色的图象。

到80年代后期，显示所用的显示器不再是数字式的。IBM视频图形矩阵(VGA)的输出是模拟信号，不过仍然保持了与以往应用环境的兼容性，即支持所有以前的视频方式。单色、彩色图形适配器(CGA)和增强图形适配器(EGA)方式可通过VGA适配器相连，例如，模拟单色显示器若连接至VGA适配器，则彩色将被转换为不同的灰度。另外，几种新的显示方式也适用于VGA适配器：

640×480图形方式(2色或16色)

720×400文本方式(单色或彩色)

360×400文本方式(16色)

320×200图形方式(256色)

将来会生产XVGA和SVGA显示器，这种显示器具有更高的分辨率和更大的彩色调色板(Palettes)。而且，借助恰当编写的Windows程序，源程序代码无需作任何改动，就可以使用这些即将诞生的研究成果。

1.2.1 从BIOS 10H到Windows软件

为理解图形软件的历史发展过程，让我们利用一点时间来讨论一下BIOS(基本输入/输出系统)。在每台IBM PC以及PS/2微机中的BIOS是一组保存在ROM(只读存储器)中的程序。这些程序为标准硬件特性提供接口，这些硬件包括时钟、键盘、软盘和硬盘，以及与Windows有关的视频子系统等。

视频BIOS程序包含一组执行基本视频编程任务的简单工具，如将字符串写到屏幕上、请除屏幕、改变颜色等等。

1.2.2 中断10H

从历史上看，为实现图形处理，程序员必须编写汇编语言程序以访问也是由汇编语言写的BIOS程序。为访问BIOS的视频部分，使用Intel8086处理器系列的程序员必须使用中断10H。

现在，ROM BIOS支持若干种视频输入/输出功能，通过执行中断10H可以访问其中的每

一种功能。这些功能均有其编号，在执行中断 10H 之前，程序员必须要将所需要的功能号放置在适当的寄存器中，例如 AH 寄存器。

当执行中断时，微处理器中的其它寄存器可能包含传递到 BIOS 程序中的其它参数。如果调用的中断 10H 功能要将数据返回程序，则通过将数据放在一个或多个微处理器的寄存器中即可实现。这种以寄存器为基础的参数传递协议主要用在汇编语言程序中。

1.2.3 高级语言

谈到软件，很可能读者最初用计算机图形的经验是使用 BASIC 的命令，如：LINE、CIRCLE、COLOR 等等。其后可能转而使用其它语言，如 Pascal。这可能使得程序更为结构化，但图形处理能力却没有什么增强。

随着 Borland 的 Turbo Pascal 的引入，程序员获得了极为丰富的图形环境。从 Turb Pascal 4.0 开始，提供了诸如视窗、剪裁、用户可定义的填充图案、三维柱体以及许多其它复杂的功能。

另一种流行的开发语言是 C/C++。该语言被认为是汇编语言和高级语言的最佳结合。C/C++ 为程序员提供了汇编语言对硬件的可访问性和高级语言所具有的强有力的逻辑结构。C/C++ 程序员现在可以使用整个复杂的图形程序库，而这些程序以前是为增强型图形应用而保留的。

在简短的历史回顾中，我们还没有讨论任何图形应用。到目前为止所编写的每个程序都是完全占用整个视频子系统，包括所有的寄存器、存储器和显示设备。

但是正如用户所期望的那样，随着硬件和软件的发展，人们要求不同程序能够同时运行。用户已不再满足于在编辑一封信以前要等待数据库完成分类操作，而希望同时做这两件事。

这种用户的需求推动了诸如 Quarterdecks' DESQview 和 Microsoft Windows 等产品的推出，这些产品为用户提供了多任务处理能力。更为重要的是 Windows 提供了图形设备接口(GDI)。

使用 Windows 的 GDI 功能，程序员可以编写用户看来比较熟悉的应用程序(具有菜单、滚动条、消息框等等)。此外，还可以改变应用程序的大小，移动，以图标表示，并在其它程序运行时将应用程序推入后台。更为重要的是 GDI 功能允许一个应用程序与另一个应用程序通信，尽管多数应用程序并不使用这一非常先进的强大的能力。

通过上述对硬件和软件发展的简短回顾，可以看出这两者变化得多么快。用户和应用程序开发人员今天所需要的是，面向图形的用户界面、多任务处理能力以及与硬件无关性。在这方面，Windows 已经成熟了。

1.3 Windows 可以做什么

与传统的 MS-DOS 环境相比，Windows 操作环境为用户和程序员都提供了相当的优越性。单独来讲，三种基本的能力(面向图形的用户界面、多任务处理以及与硬件无关性)并不是新内容。而富有创造性的是，努力将这三种能力综合在一台微机的操作环境中。

1.3.1 标准的用户界面

在 Windows 所提供的三种基本能力中，标准的面向图形的用户界面是最引人注意的，当然对用户来讲也是最重要的。统一的用户界面使用图片或图标来表示驱动器、文件、子目录以及许多操作系统命令和操作。图 1-1 显示了典型的 Windows File Manager(文件管理器)窗口的式样。程序由标题条以及许多可通过鼠标器选中菜单访问的文件操作功能组成。大多数 Windows 程序具有键盘和鼠标器两种接口。虽然多数 Windows 程序功能可通过键盘控制，但对于很多任务而言，使用鼠标器常常更为容易。

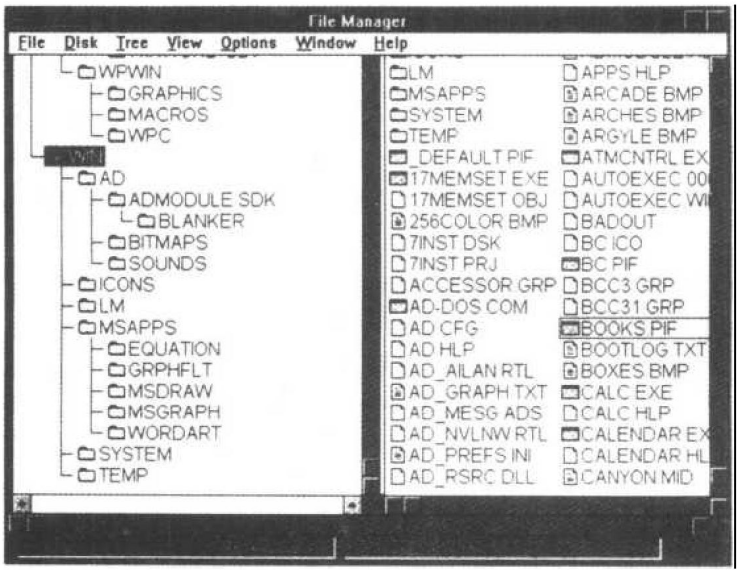

图 1-1 Windows File Manager

用户不必花费大量时间去学会使用一种新的应用程序，因为所有的 Windows 程序都有同样的“外观”。图 1-2(Windows Paintbrush，即画笔中的一个屏幕)和图 1-3(Windows Write，即书写器中的一个屏幕)说明了这种相似性。注意普通的 File 和 Edit 选项以及最大化和最小化(Maximize/minimize)按钮。对程序员而言，这种一致的用户界面是通过使用直接嵌入 Windows 中用以构造菜单和对话框的子例程获得的。所有的菜单具有相同的键盘和鼠标器界面，因为是 Windows，而不是应用程序操纵作业。

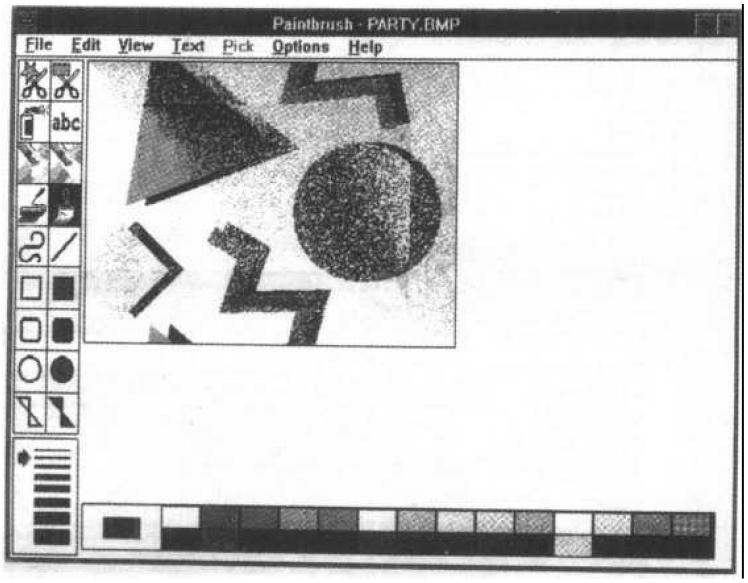

图 1－2　Windows Paintbrush

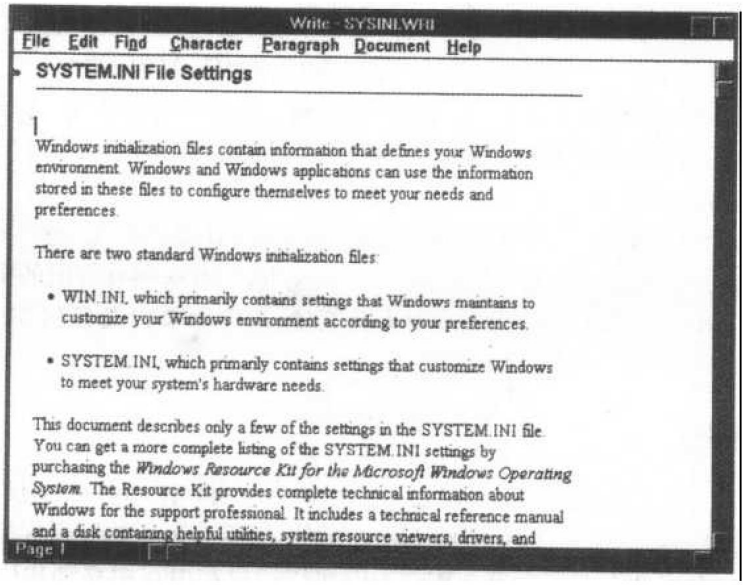

图 1－3　Windows Write

1.3.2　多任务处理

多任务处理操作系统允许用户可以有多个应用程序或同一应用程序多种处理同时运行。虽然一些人仍然怀疑是否有必要在一台微机上执行多任务处理，但很显然，诸如 Borland 的 Sidekick 和 Norton Utilities 一类的 TSR(terminate－and－stay－resident)的成功证明这是需要的。图 1－4 显示了几种拼接后的 Windows 应用程序，每个程序占用屏幕的一块矩形窗口。任何时候用户都可在屏幕上移动窗口，改变窗口大小，并且在不同的应用程序之间进行转换，还可以在窗口之间交换信息。

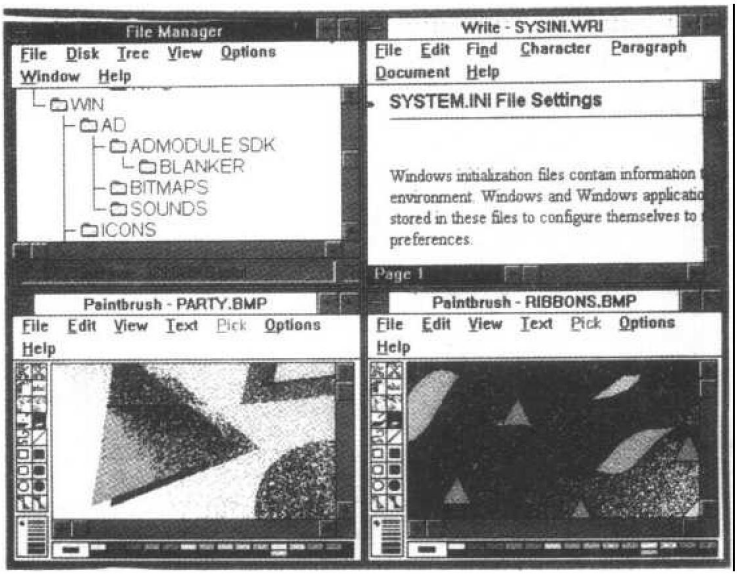

图 1-4 几种拼接后的 Windows 应用程序

尽管图 1-4 显示了 4 个同时运行的进程，但在任一给定的时间，只能有一个进程使用处理器。区别正在进行处理的任务和正在运行的任务是很重要的。另外，一个应用程序也可能处于称为激活态的第三态。一个激活应用程序就是受到用户注意的应用程序。在任一给定时刻只能有一个应用程序被处理，于是在每一时刻只能有一个激活应用程序。然而，可能会有若干同时运行的任务。划分微处理器的时间片是 Windows 的任务。Windows 通过输入和消息队列控制共享微处理器。

在多任务操作系统以前，应用程序可完全控制所有的计算机资源，包括输入和输出设备、存储器、视频显示，甚至 CPU 本身。然而在 Windows 下，所有这些有价值的资源必须共享。例如，一个标准的 C/C++程序不再访问所有未被系统、程序本身以及任何 TSR 程序所占用的存储器。

1.3.3 内存管理

内存是 Windows 下重要的共享资源之一。在多个应用程序同时运行的情况下，各应用程序之间必须通过协调共享内存，以免资源不足。另外，随着多个新的应用程序的启动和旧的应用程序的终结，内存可能会变得零碎。Windows 可以通过在内存中移动代码块和数据以合并空余的内存空间。

Windows 允许应用程序覆盖调用(Over-commit)内存。这就是说，一个应用程序所包含的代码可比一次能实际装入内存的容量大。Windows 可从内存中去掉代码，其后再由程序中的 EXE 文件重新装入。

在其它情况下，用户可能有同一程序的若干情况或拷贝在同时运行。为保存空间，Windows 共享代码。在 Windows 下运行的程序甚至能共享位于其它 EXE 文件中的例程。含有这些共享例程的文件称为动态连接库(dynamic link libraries)。Windows 包含有在运行期间连接程序与动态连结库中例程的机制(Windows 本身就是一组动态连结库)。为了方便，Windows 程序使用一种新的 EXE 文件格式，这种文件格式称为“新的可执行”格式。这些文件包括 Win-