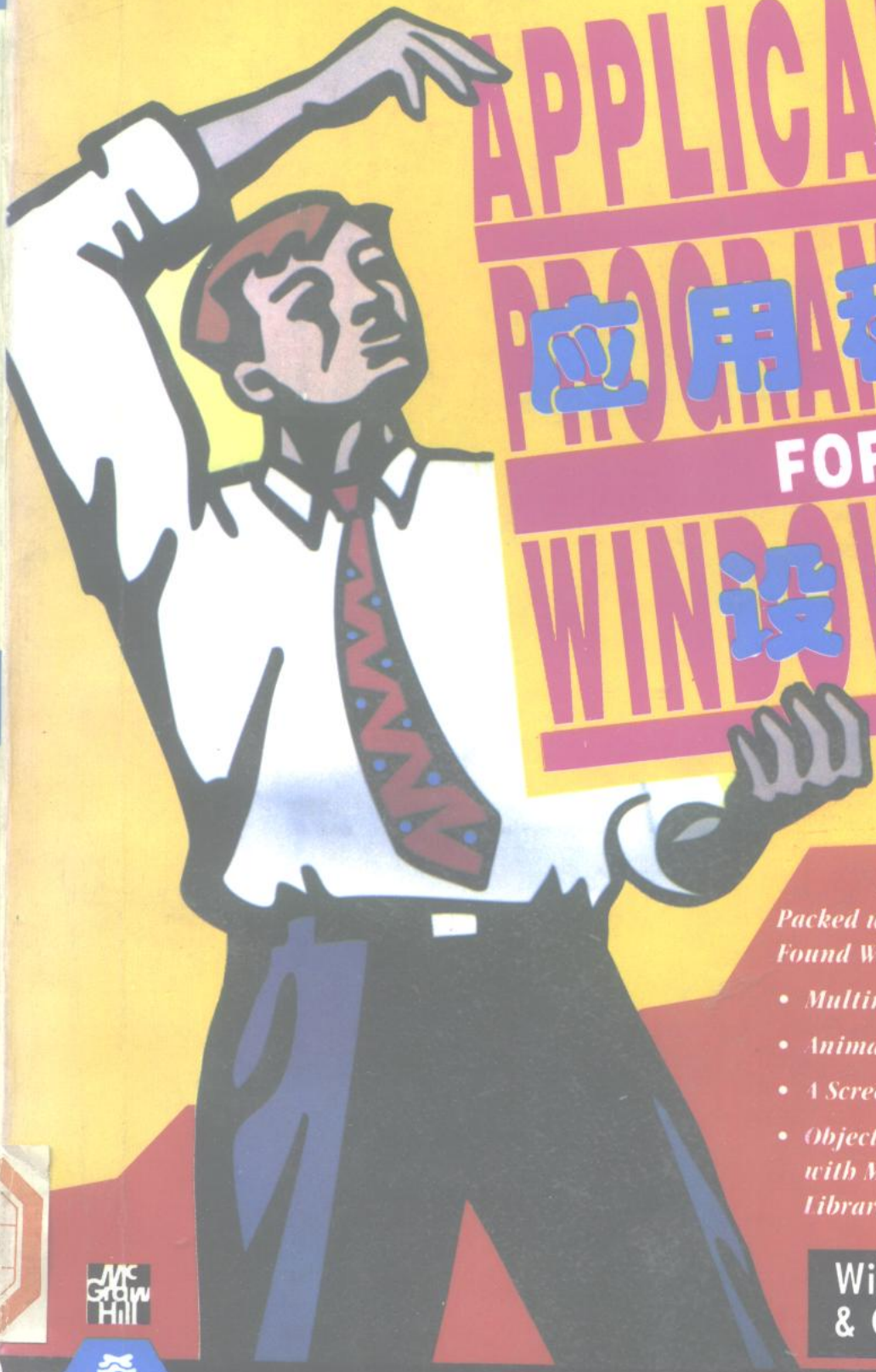
APPLICATION PROGRAMMING FOR WINDOWS NT

应用程序设计



INCLUDES 3.5" DISK

Packed with Detailed Sample Code Found Within the Book, Including:

- *Multimedia Sound*
- *Animation Techniques*
- *A Screen Saver Application*
- *Object-Oriented Programming with Microsoft Foundation Class Libraries*

William H. Murray III
& Chris H. Pappas

McGraw Hill

希望

海洋出版社

Application Programming for Windows NT

Windows NT 应用程序设计

William H. Murray Ⅲ
Chris H. Pappas
著

于春燕
谢小兵
熊可宜
译
审校

海洋出版社

1995年·北京

（京）新登字 087 号

内 容 简 介

本书首先详细解释和定义了 Windows NT 的术语、操作环境和各种功能，接着通过简单的程序来说明这些定义的概念，简单的概念被连在一起构成复杂的应用程序。

本书不仅适用于 Windows NT 编程的新手，也是 C/C++或 Windows 3.1 编程员不可缺少的参考书。

需要本书的用户，请直接与北京海淀 8721 信箱书刊部联系，邮政编号：100080，电话：2562329。

版 权 声 明

Windows NT 应用程序设计

William H. Murray Ⅲ
Chris H. Pappas 著
于春燕 译
谢小兵 熊可宜 审校

海洋出版社出版发行（北京市复兴门外大街 1 号）
施园印刷厂印刷
开本：787×1092 1/16 印张：33.75 字数：790 千字
1995 年 5 月第一版 1995 年 5 月第一次印刷
印数：1—5000

ISBN 7—5027—4093—7/TP・253
定价：58.00

简　介

Windows NT 提供了一个跨进新一代 32 位图形程序的美好坦途。也许 Windows NT 更令人期待或需要。它使程序或软件开发人员彻底摆脱 16 位 DOS 的内存桎梏。现在是学习 Windows NT 编程技术的时候了。现在是面向未来的时候了。

通过购买此书，已经说明你对 Windows NT 环境有兴趣并需要发掘你在这方面的编程潜力。

假定你已经成功地安装了 Windows NT 和 Microsoft C/C++的编译器以及 Windows NT 环境的 SDK，这些工具是用来建立 Windows NT 应用程序的。为了彻底地理解本书中的应用程序，一些 C/C++的编程经验是必需的。为得到更多的编程方面的帮助，请参见 Osborne McGraw－Hill 公司出版的《Microsoft C/C++ 7：The Complete Reference》或者《Borland C++ Handbook》Third Edition，作者是 Pappas 和 Murray。

本书的篇幅是这样安排的。前几章通过解释和定义 Windows NT 的术语。操作环境和各种功能使你逐步入门。接着通过简单的程序来说明这些定义和概念。随着你进入最后的几章，简单的概念被连在一起构成许多复杂的程序。

如果你是 Windows NT 编程的新手，请从本书的第一章开始。每一章都是一个比前一章更高的新的台阶。前几章的例子都非常短小。我们只是想说明一些编程概念，而无需用大量的程序代码来扰乱你的思路。当你读到最后几章时，程序变得越来越复杂，这是因为我们可以用前面的模块作为构件来构造更复杂的程序。坚持下去——你看完本书后你就会对你所能做的事感到惊奇。

如果你是资深的 C/C++或 Windows 3.1 编程员，你就可以用更快的速度通读本书，学到有关 Windows NT 新的特点，例如多媒体资源、新的函数等等。

多么激动人心的时刻啊！你正处在一个编程环境和个人体验的新境界。

致　谢

首先，最重要的是我们要感谢 Osborne McGraw－Hill 公司的高级编辑 Jeff Pepper 对本书的帮助和指导。同时还要感谢编辑 Frances Stack 积极协调本书编撰过程中的不同的活动；感谢你在本书的录入和排版方面做了大量的工作。还要感谢 Ann Pharr 使我们能相互沟通。我们感觉到我们是在与 Osberne 公司的最优秀的人员合作。

我们还要感谢在 Microsoft 公司推进该项目的那些职员。特别要感谢的是那些奉献了才智，创建出如此优秀的 Windows NT 软件的程序员。用户将会赞誉你们提供了强大、易于使用的操作系统。

目　录

第一部分　理解 Windows NT 概念

第二部分 向程序中加入资源

第三部分 开发优质的应用程序

第四部分 高级编程技术

第一部分

理解 Windows NT 概念

第一章　WINDOWS——过去、现在和未来

你将开始你的个人编程生涯最激动的新时代！这是因为 Windows NT 为你提供了一个坚实的基础。Microsoft 的新一代操作系统引入了许多开发人员曾经梦想的高精技术。

例如，由于 Windows NT 是一个完整的 32 位操作系统，因此你的应用程序不再需要使用令人困惑、经常冲突的内存管理模式。加上许多新的图形函数，如 Bezier 曲线，完整的 32 位图形例程将极大地提高程序的性能。另外，由于 Windows NT 引入了新的 Unicode 标准，使得程序员开发的程序可以在世界各地的不同语音版本上运行。Microsoft 以她的承诺和业已证明的能力为软件开发做了可靠的保证。

对 Windows NT 的认可始于对这个产品目标的了解。Windows NT 操作系统是 Microsoft 公司为适应下个世纪程序员需要的高档计算平台而设计的。Windows NT 是为开发人员、最终用户、公司管理信息系统(MIS)管理员、网络服务器管理员而设计的。

这个 32 位的、抢先多任务的操作系统集易用性、Windows 图形界面和先进的操作系统技术于一体。由于它运行在基于 Intel 80x86 微处理和 RISC 结构上，Windows NT 极大地释放了先进的 PC 硬件结构所蕴含的潜力。它甚至还支持对称多处理结构。

当 Windows NT 安装到你以前购买的硬件上时，并不意味着你以前的软件投资作废。Windows NT 支持所有 MS－DOS 和 Windows 应用程序的运行。

Microsoft 对向上兼容性的承诺使 Windows NT 对未来的应用程序开发仍是一个安全的平台。Windows NT 的硬件接口使它能充分利用所有的 I/O 技术，从 CD 声频到实时视频，并且能适应未来的发展。

Microsoft Windows 的巨大成功已经证明一个真理：使人们比以前更有创造力。

1.1　过去——从 MS－DOS 到 Windows

Windows NT 无疑是这个时代最活跃的最终用户环境和开发环境。然而，要全面地了解 Windows NT，你需要回顾它的历史。下面是基于 PC 硬件和软件革命性变化的一个简略回顾。

历史回顾

我们从一个普通人对计算机的印象开始。20 世纪 40 年代，计算机还在使用硬拷贝设备(用现在的标准来衡量，它就是古董啦！)，它需要最终用户翻阅大量的打印输出。自然，这非常浪费时间，而且对于个人在成千上万张硬拷贝中查找其所需的少量数据也相当困难。另外，这些古老的输出设备在容量，产生图形输出等方面也非常落后。

为了能进行简单的图形输出，计算机使用了阴极射线管(CRT)系统，该系统最早是由 MIT(马塞诸塞州立大学)为研究飞机的控制和其稳定性而于 1950 年研制出的。之所以把这个显示设备与计算机联系起来，是由于当时需要缩短用户输入和计算机输出之间的时间。

50 年代飞机也安装了 SAGE 空中防御系统，它把雷达信号转变成粗糙的计算机产生的图

像，这个新的图形系统也是第一个用光笔在屏幕上选择符号的系统。

60 年代初期，一位 MIT Ph. D 候选人通过开发 Sketchpad 画线系统改进了图形输入。这个系统允许用户在屏幕上各点之间通过光笔连线来作图。这个图形系统不仅可以画线，也可以画建筑平面多边形，简化了复制简单物体的复杂图形的生成。

早期的 CRTs 可以在显示屏幕的任意两点间画一条直线。然而，由于图像很快就消失，它又不得不在每秒钟内重画几次。60 年代，这个用来存储线的端点的存储器和用于快速重画线的硬件都非常昂贵。比如，1965 年 IBM 推出第一个适用于这种图形显示的 CRT 产品。单独的显示部分就要 100000 美元，由此你可以理解为什么当时这种设备不普及。

三年后，Tektronix 开发出了第一个存储管 CRT。这种类型的 CRT 具有保持画面的能力，直到你不再需要它为止。由于这种显示的构造的改变，昂贵的存储器和重画所用的硬件都可以省掉，从而使显示部分的价格降到 15000 美元。以这个价格，Tektronix 显示马上取得了成功。

70 年代，由于存储器和硬件逻辑器件的费用戏剧性地降低，图形开发环境得到了巨大的推动。这些变化导致了存储器、激光扫描等快速发展，从而可以显示出真实效果的带有灰度和彩色的图像。

到了 80 年代，显示监视器不再是数字的。比如，来自 IBM 显示图形阵列(VGA)的输出就是模拟的。为保证和现有环境兼容，它可以支持所有的以前的显示模式。因此，单色、彩色图形适配器(CGA)和增强型图形适配器(EGA)模式都可以在 VGA 适配器上使用。

将来会出现标准的、高分辨率的、带有大彩色调色板的监视器。随着这些新技术的开发利用，也许将来写 Windows NT 应用程序将无须代码。

软件从 BIOS 10H 到 Windows

要理解图形软件开发的历史过程，首先让我们花点时间讨论一下 BIOS(基本输入/输出子系统)。构成任何一个小系统的是一组 BIOS 例程；它们被存储在 ROM(只读存储器)中。这些子例程提供了计算机同其标准硬件的一个接口，这些标准硬件包括时钟、键盘、软件和硬盘以及与 Windows NT 相关的硬件，还有显示子系统。

显示 BIOS 例程由一组执行基本显示任务的简单函数组成，比如往屏幕上写字符串，刷新屏幕，改变颜色等等。

中断 10H

以前，要完成任何实时图形处理，程序员必须用汇编语言编写汇编程序来访问这些 BIOS 例程。要访问 BIOS 的显示部分，使用 8086 系列微处理器的程序员必须发出一个 10H 中断。

今天，ROM BIOS 支持多个在执行中断 10H 时被访问的显示输入/输出功能。这些功能都已经被编号；在执行中断 10H 之前，程序员必须先把你希望的功能的号码放到适当的寄存器中。比如，ah 中。

当中断执行时，微处理器上的其它寄存器可能含有由 BIOS 例程传送过来的其它参数。如果被调用的中断 10H 返回数据给程序，则它一样是把数据放在微处理器上的一个或多个寄存器中。然而，这个寄存器的基、参数传送规程根本上还是由你用的汇编语言程序决定(注意：由于 Windows NT 是一个完整的操作系统。所以 Windows NT 应用程序要调用新函数以替代 MS—DOS 中断的调用)。

高级语言

很可能你第一次接触计算机图形用的是 BASIC 语言。你可能已经用过像 LINE，CIRCLE、COLOR 等这样的命令。接着你可能转向学习其它语言，比如 Pascal。这个语言会给你的程序更多的结构，但在图形能力上没什么进步。

随着 Borland 的 Turbo Pascal 的引入，程序员有了一个惊人丰富的图形环境。从 Turbo Pascal 4.0 开始，许多复杂的特征，比如视口、剪裁板、用户自定义填充模式，三维条等等都可以使用。

另一种广受欢迎的语言是 C/C++，由于它把汇编语言和高级语言结合在一起，所以它被公认为最好的语言。C/C++提供给程序员汇编语言的硬件访问能力和强大的逻辑结构。C/C++程序员现在可以访问整个用于图形应用程序的非常复杂的图形例程库。

并行处理

在这个简单的历史回顾中，我们忽略了一个方面，那就是关于并行图形应用程序的研究。在此以前编写的任何程序都独占整个显示子系统，包括所有寄存器、存储器和显示设备。

随着用户对软件和硬件和期望的提高，要求程序能够实现并行运行。用户在编辑一封信之前，不再等待数据库存储结束——他们希望这两件事能同时做。这个需求引出了像 Quarterdecks' DESQview 和 Microsoft Windows 那样的产品——它给用户多任务的能力。

比多任务能力更重要的是 Windows 提供给用户的图形设备接口(GDI)。通过使用 Windows GDI 函数，程序员可以写出一个对用户看起来很熟悉的应用程序(它有菜单、滚动条、消息框等)。另外，应用程序还可以改变其界面大小、移动、变成图标、成为背景，甚至还可以和其它程序一起运行。最后，GDI 函数还允许一个应用程序与另一个通讯，虽然许多应用程序还未允许充分利用这个强大的性能。

这个关于软件和硬件发展的简短的历史概述说明软件和硬件的发展变化是如此迅猛。今天，用户和应用程序开发人员想有一个具有多任务能力并且不依赖于硬件的面向图形的用户界面。Windows 就适合于这种潮流。

1.2 现在——WINDOWS NT

如果你过去用过 Windows 的 DOS 版本，你可以问自己一个问题：除了 Window 3.x 提供我们的东西之外，我还需要什么？和 Windows 的 DOS 版本一样，NT 仍有很多地方可以改进。

Windows NT 的历史

Windows NT 准确的设计时间是 1988 年初。开发组由 Dave Cutler 领导，并汇集了许多具有设计 Windows，UNIX，VMS 和 OS/2 经验的出色的软件工程师。Windows NT 首次成功的启动是于 1989 年末在一个 Intel i860 上实现的。1990 年初，Microsoft 决定 Windows NT 操作系统基于 Windows－based 32 位 API(应用程序界面)上运行。Micrsoft 把其设计的硬件要求定位于 Intel 386/486 和 RISC 结构。

什么是 Windows NT

和传统的 MS－DOS 操作系统相比，Windows NT 操作系统为用户和程序员提供了相当多的好处。虽然三个主要性能(面向图形的用户界面，多任务和不依赖于硬件)都不是新的，但它的革新在于它把这三个性能结合在一起，构成单独一个微处理器操作系统。

图 1－1 说明了 Windows 3.1，Windows for Workgroups，Windows NT，和 Windows NT LAN Manager 之间的关系。我们可以注意到，Windows NT 包含了所有 Windows 3.1 和 Windows for Workgroups 的性能，并能运行这些环境上的应用程序。

下面的讨论将解释一些使 Windows NT 成为下世纪最好的开发平台的特点。

标准的用户界面

在 Windows NT 提供的三个主要性能中，标准的面向图形的用户界面是最引人注目的，当然，对用户来讲也是最重要的。这种统一的用户界面使用了图片或图标(icon)来表示驱动器、文件、子目录和许多操作系统命令和动作。图 1－2 显示了一个典型的 Windows NT 文件管理器。

程序通过标题条来标识，许多基本文件操作可通过鼠标点菜单来完成。多数 Windows NT 程序都有一个键盘接口和一个鼠标接口。虽然 Windows NT 程序的多数函数可以由键盘控制，但对许多任务来说，使用鼠标往往很容易。

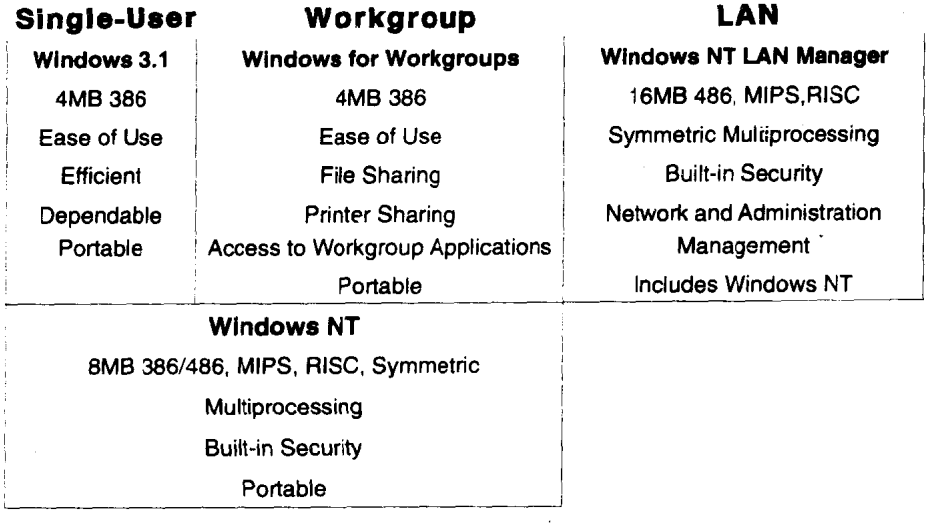


图 1－1　Windows 3.1，Windows for Workgroups，Windows NT 和 Windows NT LAN Manager 之间的内部关系

由于 Windows 程序都很“相像”，所以用户不再需要花费很长时间来学习如何使用新的应

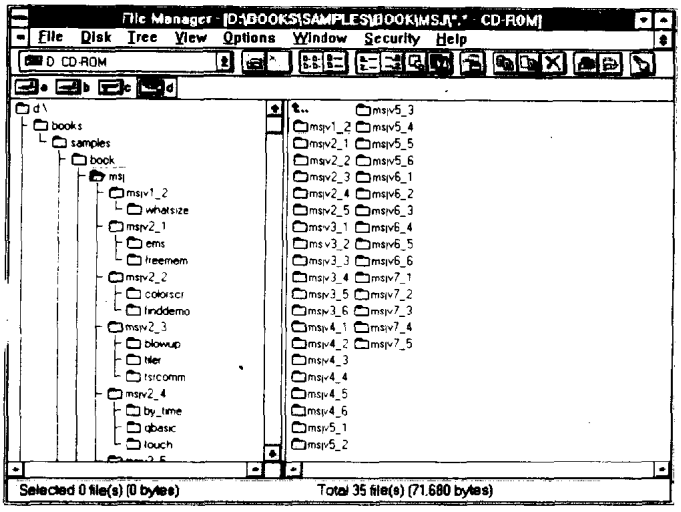


图 1－2　Windows NT 文件管理器

用程序。图 1－3(来自 Windows NT Paintbrush)和图 1－4(来自 Windows NT Notepad)说明了这种"相像";注意看命令 File 和 Edit 以及 Maximize/minimize 按钮。

对于程序员,这种统一的界面可通过使用 Windows NT 的内部子例程来构造菜单和对话框得到。所有菜单都有相同的键盘和鼠标接口,这是因为 Windows NT 处理这些工作,而并非应用程序在管理键盘和鼠标。

多任务

一个多任务操作系统允许用户有几个应用程序或同一应用程序的几个实例并行运行。图 1－5 显示了几个 Windows NT 应用程序,每个应用程序占用了屏幕的一块矩形窗口。在任何时候,用户都可以在屏幕上移动窗口,改变窗口的大小,切换应用程序,和交换窗口间的信息。

这个例子(图 1－5)显示了并行运行 4 个程序时的情况。Windows 3. x(Win16)和 Windows NT(Win32)间最主要的差别之一就是 Windows NT 是真正的占先(premptive)多任务系统。虽然图 1－5 在 Windows 3. x 和 Windows NT 中看起来是相同的,但其在后台所做的事情则完全不同。

使用 Windows 的 DOS 版本时,你可以装载几个应用程序,但在任意时刻他们中只有一个可以真正地使用处理器。区别开一个正在处理的任务和一个仅仅在运行的任务是很重要的。在 Windows NT 下,处理器自动地在各应用程序间切换,而无需支持正在执行的程序释放处理器控制权。

一个应用程序的其中一个状态叫活动态。一个活动的应用程序就是一个正在接受用户输入的程序。正如只有一个应用程序在给定的某一时间片内能够被运行,尽管可以运行多个并行的任务,但在某一时间内只能有一个活动的应用程序。分配处理器的时间是 Windows NT 的职责。Windows NT 通过输入和消息队列来控制处理器的共享。

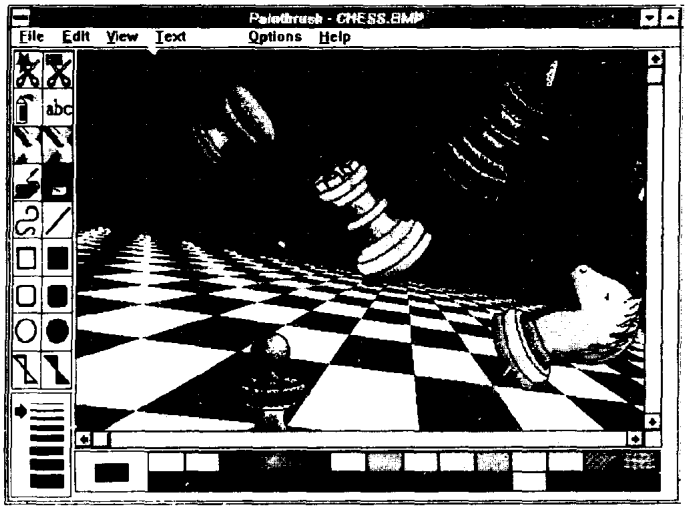


图 1－3　Windows NT Paintbrush

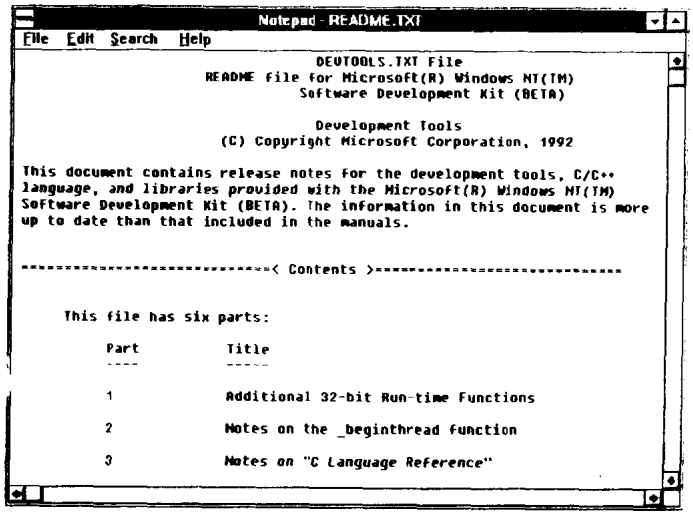


图 1－4　Windows Notpad

在多任务操作系统之前，应用程序独占计算机的所有资源，包括输入/输出设备、内存、显示器，甚至CPU 本身。然而，在 Windows NT 下，所有这些有用的资源都必须共享。例如，一个标准的 C＋＋程序不再能够访问不被系统或程序使用的所有内存。

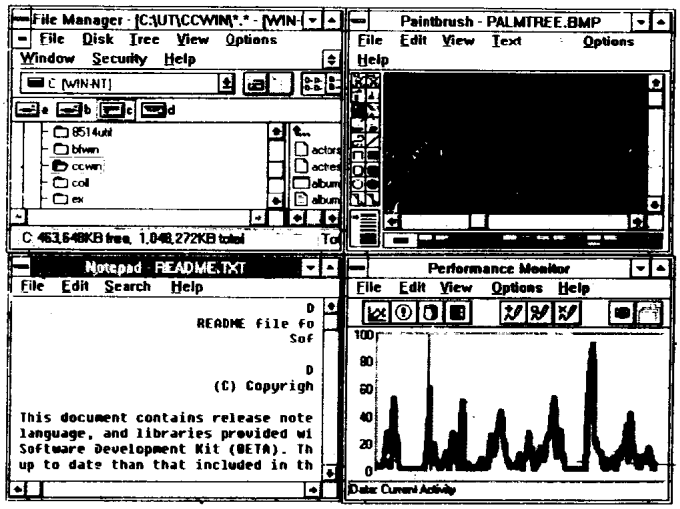


图 1—5　并行运行 4 个 Windows NT 应用程序

内存管理

内存是 Windows NT 下的最重要的共享资源之一。由于同一时期运行多个应用程序，它们之间就必须协同共享内存，以防止耗尽资源。另外，当新的程序运行和旧的程序结束时，内存变得非常零散。Windows NT 可以通过移动内存中的代码和数据块，将非常零散的空闲内存空间合并。

Windows NT 允许应用程序超过内存的限制，也就是说，一个应用程序的代码可以超过某一时刻的空闲内存大小。Windows NT 能够从内存中丢弃一部分代码并在以后再从程序的执行文件中将其重新载入内存。

在某些情况下，一个用户可以同时运行某一程序的几个实例。为了节省空间，Windows NT 共享相同的代码，运行在 Windows NT 下的应用程序甚至可以共享其它.exe 文件中的例程。这些包含共享例程的文件我们称之为动态连接库。

Windows NT 采用了一种在运行时连接程序和动态连接库中例程的机制(Windows NT 本身就是一组动态连接库集)。为了实现这个目标，Windows NT 采用了一种新的.exe 文件格式，我们称之为“new executable”格式。这些文件包含 Windows NT 所需的管理代码和数据段以及执行动态连接的信息。

输入队列

正像内存是 Windows NT 下的共享资源一样，键盘和鼠标输入也是共享的资源。C/C++程序不再能够通过 getchar 函数来直接读到键盘的输入。

在 Windows NT 下，一个应用程序不能直接从键盘或鼠标上读入数据。而是由 Windows NT 接受所有的键盘、鼠标和定时的输入而形成一个系统队列。这个队列的职责是将系统队列