Microsoft Microsoft Microsoft

WINDOWS NT 技术内幕

[美] HELEN CUSTER
程渝荣 译
王继中 校

清华大学出版社

Windows NT

技 术 内 幕

[美] HELEN CUSTER 著

程渝荣 译

王继中 校

清华大学出版社

Inside Windows NT

Helen Custer

(京)新登字 158 号

Windows NT 技术内幕

[美]HELEN CUSTER 著

程渝荣 译

王继中 校

清华大学出版社出版

北京 清华园

清华大学印刷厂印刷

新华书店总店科技发行所发行

☆

开本:787×1092 1/16 印张:17.5 字数:410 千字

1993 年 7 月第 1 版 1993 年 7 月第 1 次印刷

印数:0001—4000

ISBN 7-302-01229-6/TP・467

定价:35.00 元

本书谨献给 Windows NT 开发小组的全体成员，他们中很多人为设计、构造该操作系统作出了巨大牺牲。

愿 Windows NT 长久运行！

序

1965年，我以数学学士学位，并以物理为副科毕业于高等学院。当时有一种势不可挡的愿望想成为一名工程师，做点事情，于是我作为一个材料测试工程师在特拉华州维尔明顿(Wilmigton，Delaware)的Dupont公司工作。大约过了非常乏味的一年之后，我被借到数学与统计小组，并指定为Scott造纸公司正在开发的新的泡沫处理过程建立一个计算机仿真模型。使用那些从不按我要求工作的机器是丢面子的，但仅在六个月之内我就被吸引住了，那些在出学校时我曾经躲开的东西——计算机，变成了我一生的职业。

不久以后，我转到Dupont的工程部，在那里我可以全时编程。Dupont公司有一个开发联机计算机系统应用程序的小组。我参加这个小组的真实动机是想更接近计算机。事实上，我想实现一个操作系统。在这个小组中，我很幸运地在与项目有关的若干独立的实时系统上工作，编写调度各种任务及监视系统活动的中央控制程序以及实际应用程序的代码。

很快我就发现要想得到实现一个真实操作系统机会的唯一途径是加入一个以计算机为经营对象的公司。于是1971年我离开了Dupont公司，转到马萨诸塞州(麻省)梅纳德(Maynard，Massachusetts)的DEC公司工作。结果，我还是等了相当长的时间才进入了操作系统行业。我根本不知道我是否有幸能在一生中开发几个操作系统；对任何人来说，能开发一个操作系统也是极少有的机会。

我的第一个操作系统项目是建造一个称作RSX-11M的实时系统，它运行在DEC的PDP-11 16位小型机系列上。当时，我们的目标似乎非常雄心勃勃，我们被要求建立一个多任务操作系统，它能运行在32KB内存上，并具有层次文件系统、应用程序交换、实时调度以及一系列开发用实用程序。这个操作系统及一些实用程序准备在PDP-11平台的全部系列上运行，从很小的系统一直到PDP-11/70，PDP-11/70具有内存映象硬件并支持最多到4MB的内存量。

对于RSX-11M的开发过程，我有许多美好的回忆。我做了一个橡皮图章，刻着“容量就是目标”，并在每封项目信函上都盖此戳以确保所有程序设计员及产品负责人都懂得要达到我们的目标是何等重要。我们也了解了条件汇编语言的功能(那时将高级语言用于操作系统还处于摇篮时代)，不论何时只要有人增加了一个功能，我们就把它变成系统生成的一个选项。

在开发RSX-11M时，我们大部分时间都花在用工程方法解决内存的问题上。因为系统要在32KB上运行，我们创建了一种内存预算方法，将可用内存平分给操作系统和公用程序。这样只给公用程序留出了16KB，并导致花了很长时间去调整覆盖结构以使许

多 RSX-11M 系统程序能达到可接受的性能。

尽管 RSX-11M 有某些很苛刻的容量及速度限制，但在我所工作过的系统中，也许它是最容易开发的系统。它包括重新实现一个已存在的系统，但我们可以自由更改或替代其编程接口，只要修改应用程序最少量的源代码就能通过重新汇编或重新编译。RSX-11M 在 1973 年问世，即我们开始建造它的十八个月之后。事实证明它是非常成功的，并且有助于使 PDP-11 成为当时最流行的 16 位小型机。

PDP-11 比大型机提供了更好的性能价格比，部门一级就能买得起。它和同时代其它流行的小型机一起导致了计算机产业"小型化"的第一次浪潮。小型化的企图是将大型机的应用程序"下放"到小型机系统中。由于许多大型机程序都比 PDP-11 能容纳的要大，因此几乎是同时 DEC 就起来反击 Gordon Bell 认为(PDP-11)小型机的计算机结构已过时的最重要理由：缺乏足够的寻址位。

出于这种需求，VAX 结构诞生了。它变成 70 年代后期最流行的结构之一，并一直保持到整个 80 年代。VAX 结构提供了 32 位虚拟寻址空间并消除了将程序塞进似乎总在减小的虚拟地址空间的需求。

我第二次开发一个操作系统的机会随着 VAX 而到来。我很荣幸被选中来领导研制这个用于 VAX-11 结构的操作系统的工作。这一工作的结果是 VMS 操作系统。

VMS 是 DEC 公司的第二个通用分时系统，是专门为 VAX 结构开发的系统。然而，因为 VAX 结构是在 PDP-11 得到巨大成功之后发展起来的，这次就强制要求为应用程序提供比源码级更高的兼容性。因此，VAX-11 结构中包括了一个 PDP-11 兼容方式。在这种方式下，PDP-11 的指令可以直接由硬件执行。在当时一个操作系统支持多种"兼容"环境是不可思议的。尽管 RSX-11M 并不是 PDP-11 操作系统中最著名的，(令人惊奇的是 DEC 在同一时刻前后竟有不少于 10 个用于 PDP-11 的操作系统!)但它还是被选作在 VAX 上模仿 PDP-11 兼容方式的操作系统接口。这一决定可能对公司以外的许多人来说毫无意义，但是 RSX-11M 具有大量的应用程序开发工具，具有最多的通用操作系统功能，支持多任务，还有一个适于扩展的文件系统结构。最终，订购到的软件包可立刻在 VAX-11 系统上运行 RSX-11M 二进制码；它使 RSX-11M 可以直接安装，并且其中的文件可以在 RSX-11M 兼容方式程序和本机 VMS 程序之间存取和共享。

从技术发展前景看，我们在 VMS 上所犯的最大错误是没有用高级语言来编写它。当时，我们有一组非常有经验的汇编语言编程人员，由于苛刻的容量限制和没有一种适合操作系统开发使用的高质量的编译器，因此，为了保证在市场需求时间之内交付系统，我们采用了汇编语言。现在回顾过去所发生的情况，要做出应该用高级语言编写 VMS 的决定仍然是很困难的。(无疑，技术上看来是正确的事情在财政上并不总是该做的最好的事情。)

80 年代初，当小型机正忙于吸收大型机及其他新应用程序的优点时，两个主要的技术出现了：个人计算机(PC)和工作站。完成 VMS 项目之后，我花了几年时间研制编译器，而后领导一个小组研制 DEC 的第一台 Micro VAX 工作站——Micro VAXI。

类似 Micro VAX 的工作站为像计算机辅助设计(CAD)这样的应用程序提供了单一的、高性能的设计环境;而 PC 则支持面向个人生产活动的商业性应用程序,如电子数据表格和字处理——两个非常成功的早期 PC 产品。虽然工作站相对贵一些,但个人计算机必须使小企业买得起。

为了满足价格目标,最初的 PC 是用 8 位,而后是用 16 位处理器建造的。它们像 RSX-11M 一样受到限制。这就要求部分程序员和操作系统设计者做出很大的努力去适应这种限制。硬件资源是如此贫乏,以致于操作系统主要只能用于处理很少的低级的硬件函数及提供一套文件系统库。但是个人计算机提供了一些小型机不具备的东西——一个独立软件开发者可以大量销售其程序的市场。结果,运行在 PC 机上、且开发出其各种性能的应用程序的广度和多样性实在令人吃惊。

80 年代中期,微处理器达到了 32 位寻址,工作站很快就利用了这一能力。但是,由于已经有了非常大量的、已安装的个人计算机及其应用程序,简单地把所有应用程序软件搬进另一种计算机,然后重新编译、重新连接并不容易。PC 的最终用户根本没有程序的源代码,他们要求二进制上的兼容性。

1988 年夏,我接到 Microsoft 公司 Bill Gates 一个有意思的电话,他问我是否愿意去讨论一下在 Microsoft 公司为个人计算机建造一个新操作系统的问题。当时,我对个人计算机并不很感兴趣,但我认为这是一个见到 Bill 并讨论他已有设想的机会。Bill 所要提供的就是建立另一个操作系统的机会,一个可移植的操作系统,并且涉及人们在使用个人计算 机运行关键任务的应用程序时所关心的问题。对我而言,这意味着创建另一个操作系统的机会!

最后 Bill 说服了我,使我相信这是一个我不该错过的机会。1988 年 10 月,我来到 Microsoft 公司并开始组建开发新操作系统的队伍。当时我还没意识到,这将是我所从事的最雄心勃勃的操作系统项目。

我们的系统设计目标包括可移植性、安全性、符合 POSIX、兼容性、可缩放性(支持多处理器)、可扩充性以及易于国际化。在所有这些目标中,最难达到、也是对系统结构最具有深远影响的是兼容性。几十万套 PDP-11 系统已经卖出,而几千万台个人计算机在使用中!好象这还不够,我们需要兼容地支持三种不同的 16 位操作环境,并且增加新的 32 位能力来解决个人计算机应用程序所遇到的类似 PDP-11 所受到的虚拟地址限制的问题。为了要使它更完善,我们还要支持被称为 POSIX 的 UNIX 标准接口规范。

现在,也就是大约四年之后,我们已处于将这系统——Windows NT 带进市场的边缘。Helen Custer 从这个系统设计开始时就开始写这本书,随着我们的设计日趋成熟,这本书也不断地修改以跟踪这个操作系统的结构。这是一项艰巨的任务——随着设计的发展不断更新,编写或重写书中的各章。虽然系统是我们设计的,但 Helen 是捕捉设计精髓的人,她让更多的人,远超过认真实现这个操作系统的人,能理解这些设计。正因如此,我们要感谢 Helen。

要感谢所有为 Windows NT 设计做出过贡献的人是不可能的。我必须说我并没有设

计 Windows NT——我只不过是这个系统设计者中的一员。当你读这本书的时候，会介绍到一些其他设计者，但不是全部。这是整个小组集体努力的结果，并且包括了几百人年的努力。也许最重要的贡献是由测试并增强了这个系统的人们所做出的，没有他们的努力，Windows NT 不可能达到它已经达到的质量水平。

我希望读者能喜欢这本书，就像我们喜欢设计这个系统一样。

Dave Culter
Windows NT 开发部负责人

前 言

自1989年我写这本书以来，已经走过了漫长的道路。在接受这一任务时，我对全身心地投入到操作系统的理论、设计、实现以及其它新的知识中毫无准备。开始写作之前，我又重读了 Tracy Kidder 的《新机器之魂》《Soul of a New Machine》一书，以期找到灵感及某种意义上的同路人——至少是一个已走过与我将要走的路相似的人。在许多方面，Windows NT 的构造就是 Kidder 书中所述硬件构造的一个软件版本，我想，我的工作有点象 Kidder 一样。

象创建计算机一样，创建一个操作系统是极少数工程师才能得到的机遇。大多数操作系统工程师穷尽毕生的精力去增强或修改现存的操作系统，或是设计永远不会被采用或投放市场的新操作系统。计算机公司常有的失败以及财政、管理方面的困难使它们在项目完成之前就被迫取消，那些没有完成的系统通常在市场上并不受欢迎或不受重视，因为现有的应用程序要求永远支持原有的系统。很少作家有机会写一本象本书一样的书，其中记录了一个重要的新操作系统的设计，因此写这本书是一种不寻常的特权。

本书的背景材料并不新，其中大部分以前都以多种形式描述过，并且通常写得都比我所拼凑的几页更为雄辩有力。然而，我的目的并不是写一本已在其它书中出现的有关操作系统原理的教科书，而是要将 Windows NT 置于现存系统的体系中。尽管我没有赘述常常隐含在实现背后的复杂推理，但我还是试图提供一些影响 Windows NT 最终格局的操作系统的历史及研究概貌。

这本书不是为操作系统的设计者们编写的，他们可能需要的是提供比这本书更详细的 Windows NT 内部的情况。然而，这本书的读者对象是那些对计算机已有一些了解，但为了编写更好的应用程序或简单地解开被称为操作系统的那个黑盒之谜而需要了解该系统的内部设计的人。

《Windows NT 技术内幕》在 Windows NT 最后完成之前几个月就写完了，因此，这本书中描述的一些特性也许最终没有出现在首次发表的版本中；有些可能被推迟到后续发表的版本中；而有些则可能被彻底地抛弃了。然而，我试图提供一个 Windows NT 的长远概貌，而不描述太多渺茫的希望或侧重于那些很可能将被改变的实现细节。这里描述的一切要么已经在系统中，要么是已存在的、但被扣压以等待进一步测试，或等到用已有的软件产品适当混合来补充之。对有些专题作了必要的省略，这是因为或许这些专题是在开发的后期引入的，或许这些专题可能在其它地方陈述过。还有一些专题，如安全性及每一子系统的内部设计，都被缩写了。一个明显的例子是 Win 32 子系统，它在第五章“Windows 与保护子系统”中进行描述，但其内部细节会出现在另外一卷书中。这本书没有详细阐述 Win 32 API——因为其他作家已经开始在写专门的书。本书将集中描述 Windows NT 的设计和 Win 32 及其它 API 环境是怎样“插进”NT 执行体的。

没有必要从头至尾逐页阅读这本书；这本书的结构使读者可以先读前两章，然后转向所感兴趣的任何专题。但由于技术词汇及理论互相关联的原因，从头至尾地读这本书会增进读者对某一领域的理解。

在过去的三年中，我曾经与许多人交谈过，启发他们开口，倾听他们的意见，并与他们争论过，所有这些人都值得我一谢。我最感谢的是 Dave Cutler，他本想写这本书，但还是把写此书这一不可多得的机会给了我。他在技术及编排方面的意见也是极有价值的。

我也向 Lou Perazzoli 表示极大的感谢，他是在我写作过程中唯一读完我所写的每一页草稿的人，甚至在他的日程安排不可能周转的情况下，他仍然尽力完成了这一任务。没有 Lou 的协助与支持，这本书就不会存在。

特别感谢 Ron Burk 和 Gary Kimura 为我推荐了合适的框架结构，这样我才能随着计划的进展把我所收集到的大量信息组织到一起。找到一个编辑框架并将这样一个复杂多样性的系统信息紧凑编写进去，曾经是写这本书的最大困难。

还要感谢那些软件工程师，他们允许我自由地借用他们的技术规范中的文字，而且当我试图从一个偏离他们的角度来反映他们的观点时，他们十分耐心。尽管如果这本书由他们来写可能不一样，但是这本书确实是他们的书；它记录了四年中他们的欢乐、忧虑、挫折与灵感的由来。与他们一起工作并分享这唯一的体验是一种特权，也是一种挑战。除了前面列出的以外，特别要感谢 Darryl Havens、Steve Wood、Mark Lucovsky、Jim Kelly、Scott Ludwig、Mattew Felton、Mark Zbikowsky、Chandan Chauhan、Chuck Lenzmeier、Mary Hutten、Asmus Freytag、Dave Thompson、Larry Osterman、Sanjay Jejurikar、David Gilman、Robert Reichel、Chad Schwitters、Bryan Willman、Eric Kutter、Lee Smith、Steve Rowe、Paul Leach 、Bruce Hale、Roberta Leibovitz、Gregory Wilson、David Treadwell、Sudeep Bharati、Chuck Chan、Manny Weiser、Leif Pederson、Dan Hinsley 、Bob Rinne、David McBride、Richard Barth、John Balciunas、Rick Rashid、Therese Stowell、Dave Hart、Matthew Bradburn、Cliff Van Dyke、David Thacher、Jane Howell、Lorelei Seifert、Bob Muglia 和 Paul Maritz 提供的技术、编排或精神上的支持。

我个人要感谢 Callie Wilson 在内部分发这本书，也要感谢 Carl Stork 在这本书原稿已完成的消息泄漏出去后为我抵制干扰。很高兴能与 Microsoft 出版社的工作人员合作，其中包括 Nancy Siadek、Jeff Carey、Deborah Long、Judith Bloch、Connie Little、Katherine Erickson、Peggy Herman、Jean Trenary、Barb Runyan、Kim Eggleston、Wallis Bolz 和 Dean Holmes，感谢他们满足了这一困难的出版日程并且沉着地处理了这本大而细的书中的错综复杂的事情。

我还要向 Microsoft 图书馆的工作人员致谢，我从那里得到了我用作背景及参考资料的所有文章及书籍。每当我提出棘手的请求时，他们从未拒绝过我；当我借阅书籍过期时，他们也从未对我发过火。我也向我在堪萨斯大学时的操作系统课的老师 Daniel Canas 致以迟到的感谢，他激起了我对操作系统的兴趣并教给我研究的价值。

贯穿于这本书中，读者将会看到 Windows NT 的设计者与实现者的名字。许多名字都被省略了，但这种省略是随机的，仅仅反映出操作系统的某些部分未在本书中予以描述，或者是对某一特定部分有太多的贡献者以致于不能逐一提及。尽管本书说 Dave Cut-

ler 主要是 NT 内核的开发者，但他作为 Windows NT 的总设计师及最多产的编码者之一，提供了许多代码或至少是操作系统几乎每一部分的方向。

设计良好的操作系统都具有某种美感，在其实现的表面、无尽的细节之下包含着可理解的 内在体系。我写这本书的目标就是审视大量的软件并将所有细节都削去以揭示出其内在的体系。这种艰难冒险的怪论也许从一则简短的轶事中可以得到最好的说明：

一天下午，我坐在 Lou Perazzoli 的办公室里，他正给我讲述虚拟内存系统中的一个部件——工作集剪裁中的入与出（几乎是逐字地）。在他解释时，我专心致志地听并在心里构成他的描述的摘要，那或许将适合写入本书中。当他讲述结束后，我从我的观点出发，简要概括了他所说的，然后问他："我说的对吗？"他真诚地回答："对，那正是我们所做的部分工作。"

这本书体现了在详细的实情与有序美之间的平衡作用。因此，这本书"准确"记录了开发者所做的"部分工作"。我感谢他们让我共享他们头脑中的内涵，改写这些内容时的任何错误都归咎于我。

Helen K. Custer

1992 年 9 月

译 者 序

1993 年 5 月 24 日 Microsoft 公司正式推出了 Windows NT。Windows NT 是 32 位的操作系统，是 Microsoft 公司 Windows 产品系列中的最新成员。

自 Microsoft 公司着手开发 NT 以来，引起了计算机界广泛地注视。Windows NT 并不是替代目前 DOS 上流行的 Windows，而是为了满足高档、单用户桌上工作站平台；满足局部区域网络超级服务器，或者主干计算机系统的需要。NT 和 DOS 上的 Windows 都支持图形用户接口和 Win32 应用程序接口子集。但是，NT 具有良好的可移植性；安全性达到美国政府 C2 级安全标准；每个应用程序都可以使用单独的 32 位受到保护的地址空间，克服了 DOS 上 640K 瓶颈的限制；NT 支持对称多处理结构；支持多线程程序；集成了网络；采用 16 位标准字符集的单一代码方法来支持国际字符集；提供了性能优良、可靠的文件系统，等等。

《Windows NT 技术内幕》一书译自 HELEN CUSTER 所著的《INSIDE WINDOWS NT》。本书介绍了 NT 的设计目标、思想、方法及其结构和实现技术。由于 Windows NT 及 Windows 都是重要的操作系统环境，为了了解和掌握 NT 技术，我们翻译了本书。

参加本书翻译的还有温以德、孙全康、周立、王涵、何菁、冯焱、娄军、吕缨等同志。全书由王继中教授审校，在此对王继中教授及本书编辑表示感谢。

译 者

1993 年 6 月

目 录

第 1 章

使　命

在操作系统的世界中，进步的车轮缓缓而行。操作系统需要花多年时间才能开发出来。一旦完成，它们仍然是无生命的，直到能够利用其潜力的应用程序被开发出来。即使在应用程序问世后，人们还必须通过文档、培训及实践去学会如何使用它们。这就与通常为操作系统开发应用程序时的延迟联系起来了，它意味着普通用户通常拥有并使用的是10 或 20 年前的操作系统技术。

当操作系统等待接受时，硬件技术却向前迈进了。具有更快的处理器、更多内存以至多个处理器的计算机变成了常事，这样操作系统开发者就要急忙扩展其已有系统以利用新的硬件性能。。

Intel 80386 和 80486 芯片与许多其它流行处理器一道被称为复杂指令集计算机(CISC)。它们的主要特点是具有大量的机器指令，每条指令都很精细并且有用。在过去的几年中，Intel 已经在处理器的速度及功能方面取得了很大进展，其他制造商已经在 Intel 的 CISC 技术基础上开发出了多处理器机器。

在 80 年代中期，硬件产业创造了另一类被称作精简指令集计算机(RISC)的处理器结构。RISC 芯片与 CISC 芯片的主要区别在于 RISC 芯片只提供少量的简单机器指令，因为简化了指令集，RISC 处理器可以在提高了的时钟速度下运行并且取得很快的执行时间。

在 CISC 和 RISC 竞争场中，有希望的处理器技术正迅速出现。Microsoft 认识到要利用这些及其它硬件的先进特性，就需要为 90 年代研制出一种新的操作系统——一种可以移植并能轻易地从一种硬件平台移到另一种硬件平台的操作系统。尽管 Microsoft 与 IBM 在 80 年代创造了 OS/2 操作系统，但 Microsoft 认为那个系统有许多缺点，最明显的一条是 OS/2 不能移植。它是用汇编语言写成的，运行于单处理器——Intel 80286 计算机上。Microsoft 没有为 OS/2 系统软件欢呼，而是决定重新建立一个新的、可移植的操作系统。

1.1　90 年代的操作系统

1988 年秋，Microsoft 聘请了 David N. Cutler("Dave")来领导一个新的软件开发行动：创建 90 年代的 Microsoft 的操作系统。著名的小型计算机系统设计师[1]Dave 很快就组织了一支工程师队伍来设计 Microsoft 的新技术(NT)操作系统。1989 年初，Bill Gates 和 Microsoft 的主要决策者们一起审查了 Dave Cutler 小组所定义的操作系统规范，他们的计划为新操作系统确定了如下几项主要市场需求：

可移植性　硬件进步发展很快而且常常是不可预计的，例如，RISC 处理器就代表了

与传统CISC技术的不同。用一种可移植语言来写NT会使它很方便地从一种处理器结构移植到另一种。

多处理及可伸缩性 应用程序应该能利用现在可用的各种计算机的特性，例如，具有多个处理器的计算机已经出现在市场上了，但几乎没几个现有的操作系统能充分利用它们。使NT成为一个可伸缩的、多道处理的操作系统，会使用户能在单处理器计算机与多处理器计算机上运行同一个应用程序。在最佳情况下，用户可以全速同时运行几个应用程序，计算量大的应用程序也可通过将其工作分配到几个处理器上而提高性能。

分布式计算 随着80年代个人计算机的增加，计算的性质也不可避免地改变了。以前使用单一大型主干机作为整个公司服务的地方，而现在小的、较为便宜的微机到处可见，并且成了普通员工的标准工具。增强的网络功能允许较小的计算机互相通信，常常共享如磁盘空间或处理能力(如文件服务器、打印服务器或计算服务器的形式)这样的硬件资源。为了适应这种变化，NT系统的开发者应将网络功能直接装进操作系统，而且还应提供一种工具，它使应用程序能分布在多个计算机系统上工作。

POSIX承诺 在80年代中期到后期，美国政府部门开始定义POSIX为政府计算合约的认证标准。POSIX虽然是被不确切地定义为"基于UNIX的可移植操作系统界面"的首字母缩略语，其实它代表了UNIX类型的操作系统界面的国际标准集。POSIX标准(IEEE标准1003.1—1988年)鼓励制造商实现兼容的UNIX风格界面，以使编程者很容易将其应用程序从一个系统移到另一个系统。为了满足政府的POSIX认证要求，NT将设计成提供一个可选的POSIX应用程序执行环境。

政府认证的安全性 除了对POSIX的承诺之外，美国政府还为政府应用程序规定了计算机安全准则。达到某种政府批准的安全级就使操作系统可以在那一竞争场中竞争。当然，那些限定的能力中的多数是多用户系统所拥有的特性。安全性准则定义了一些必要的能力，如保护一个用户的资源不受他人侵犯、建立资源配额以防止一个用户得到全部系统资源(如内存)。

NT安全性的初始目标是所谓的C2级，这是由美国国防部定义的，即提供"自由决定的(需要知道)保护并且通过引入监视功能对用户及其行为负责"[2]。这就意味着系统资源的拥有者有权决定谁能存取资源，而操作系统能检测出何时数据被存取及被谁存取。美国政府的安全级从D级(最不严格)定到A级(最严格)，其中B级和C级分别有几个子级。尽管NT最初会写成支持C2安全级，将来的增强版本会满足更高安全级的、更严格的要求。

由于存在这些市场需求，NT开发小组就有了它的使命：为90年代创造Microsoft的操作系统。最初，这一计划还要求NT具有OS/2风格的用户接口，而且将OS/2应用程序编程接口(API)作为主要的编程接口提供给用户，然而在系统开发的过程中，Microsoft Windows 3.0版走俏市场，而且与没有争取到大量用户的OS/2相反，立即取得了成功。

Microsoft在认清了这个市场要求及考虑了增强并支持两种不兼容的操作系统所带来的复杂性之后，决定改变其路线，将其精力引向一个单一的、一致的操作系统策略中。这个策略就是创建一个基于Windows的操作系统家族，这一家族能适用于从最小的笔记本式机到最大的多处理器工作站的各种计算机。Windows NT作为下一代Windows系统被