

第一篇 结构与功能篇

Windows NT 操作系统结构与功能的探讨

个人计算机自 80 年代蓬勃发展以来,就使人们的生活与工作带进了信息化和自动化的世界。到 90 年代,由于计算机硬件在运算能力与各种功能上的增强及软件在网络、通信及多媒体技术上的突破,使得个人计算机与工作站的应用进入一个全新的纪元。在个人计算机的领域里,微软以其 Windows 在市场上成功的声势下,紧接着推出了 Windows NT,也在计算机领域引起相当大的震撼。由于微软的 Windows NT 是新一代的 PC 操作系统,其功能与特性相当于 UNIX、VMS 等中大型操作系统。预测此举将使个人计算机的发展进入一个新的里程。

本篇针对 Windows NT 的系统结构与功能作深入的探讨。全文共分为四章,其内容包括:

第一章 Windows NT 的市场分析。本章主要讨论 Windows NT 在现在与未来个人计算机市场上所扮演的角色及对国内厂商的影响。

第二章 Windows NT 的功能与特性。本章详细地介绍了 Windows NT 操作系统的结构,以帮助设计者了解 Windows NT 的功能与特性。

第三章 Windows NT 与 NT Advanced Server 下的网络环境。本章特别强调 Windows NT 的网络结构及与 NT Advanced Server 所构成的网络系统。

第四章 Windows NT 与新一代个人计算机的硬件结构。本章介绍目前最新的个人计算机硬件结构以及 Windows NT 作为服务器或高档工作站(High-End Workstation)的操作系统所需的硬件设备。

第一章 Windows NT 的市场分析

1.1 市场背景

一个新的操作系统是否成功,除考虑技术方面因素外,也必须考虑有相关的市场与时机的配合。从下列计算机工业的发展趋势,我们可以看出 Windows NT 及其应用在目前个人计算机软硬件市场的角色与契机。

1.1.1 个人计算机运算能力迅速增强,但价格却持续下降

Windows NT 的市场是定位在高档个人计算机或服务器(Server)的操作系统。目前在 PC 及网络服务器中,最广泛使用的硬件平台是 80486 为处理器(CPU)的计算机。但由于新一代的 RISC CPU,如,DEC Alpha、MIPS R4X00 与最新推出的 Apple、IBM、Motorola 共同开发的 PowerPC 等,及新一代的 CISC CPU,如 Intel Pentium,均可以 Windows NT 下执行,其执行速度为 486 的二至四倍。因此,以高档的 PC 作为硬件平台,加上 Windows NT 作为操作系统,可提供使用者一个更理想的应用环境。

图 1-1 表列出了目前 CPU 厂商及其联盟的计算机厂商。读者可看出 CPU 厂商间的竞争的相应激烈的。正因为如此,此类新一代的 CPU 在价格上也不断地下降。其中 CPU 的降价更使得计算机厂商有了更具竞争力的机种,例如,IBM Motorola 与 Apple 合作开发的 PowerPC 计算机,DEC 的 Alpha AXP,及众多以 Intel Pentium 为 CPU 的各厂牌 PC 与 Multiprocessor(MP)机器。

The Who's Who of RISC Technology

RISC	PowerPC	HP-PA	Super Sparc	R4X00	Alpha
Designer	IBM & Motorola	HP	Sun	MIPS	DEC
Chip Makers	IBM Motorola	HP Hitachi Samsung	T. I. LSILogic Cypress Fujitsu Philips, ...	NEC Toshiba Siemens LSI Logic IDT, ...	DEC
Major Allies	Apple, Bull Cetia, IBM Taiwan New PC (TNPC) Consortium	HP, Samsung Prime, Swquola Control Data Hitachi Convex	Toshiba, RDI Tadpole Sun, Seiko Tatung Hyundai Goldstar...	Ace, Acer Olivetti Siemens DEC, SGI NEC, Sony CDC, ...	Digital Kubota Cray

图 1-1 CPU 厂商与计算机厂商间的联盟关系

1.1.2 高档 PC 与服务器需求量不断上升

上述 CPU 与计算机厂商之所以竞争如此激烈,主要原因是目前及未来高档 PC 与服务器的需求将为计算机市场带来新的契机。全世界约有一亿二千万台 PC,其中 20% 的使用者为 Power Users。所谓 Power Users 即程序开发者,多媒体使用者,图形工作站使用者,及一些特殊用途需要(Mission Critical)的使用者。这些 Power Users 对目前的软硬件环境并非十分满意。他们的需要包括:Performance 更好的硬件,更能发挥硬件功能的系统软件,更容易使用的环境,功能更完备的网络功能,以及能与既有的系统共存等条件。而 Windows NT 的目录即是希望能满足这 20%(约两千四百万)的 Power Users 的需求。

除了 Power Users 处,全世界也有数百万各种类型服务器在 PC 网络上。依据 International Data Corp. 的调查,1994 及 1995 年以 PC 作为平台的服务器将以每年增加一倍的速度增长。以 Windows NT 为基本担任系统的各种 Server 应用软件,例如:MS SQL Server、NT Advanced Server、SNA Server 及许多从 UNIX 移植到 Windows NT 的 Application Servers 等,将为服务器市场提供更多的选择机会。

1.1.3 Windows 使用者界面广受欢迎

自 1989 年 Windows 3.0 推出,至现在 Windows 3.1 的广泛使用,微软的 Windows 几乎已成为标准的 PC 使用者界面。由于 Windows NT 的使用者界面与 Windows 相同,因此 PC 使用者非常容易上机操作 Windows NT。此外,在主从式运算为结构的小型化(Downsizing)过程中,采用 Windows 应用程序作为 Client 端的软件已是大势所趋。因此,以 Windows NT 为操作系统平台的 Server 软件与 Windows 搭配能为用户提供一致的操作环境。正因为如此,一些原本欲采用 UNIX 为 Server 平台的客户,纷纷转而采用 Windows NT。这一趋势也造成更多的中大型系统软件移植到 Windows NT。这也正好弥补了目前 Windows NT 在主从式运算中仍缺少稳定及功能强大的中大型 Server 软件的弱点。

1.1.4 中型化市场已渐趋成熟

所谓小型化,是指将诸如 IBM、DEC、WANG 等大系统的工作移植到以区域网络为主的主从式结构的平台上。据美国 Dataguest 所做的统计,1992 年美国 1,000 家大公司中,47% 的公司宣布已完成小型化工作;25% 的公司仍未进行中型化;剩余的 28% 则正在规划与进行之中。

采用小型化作为公司计算机化解决方案的主要因素是价钱便宜。就大企业而言,采用大型专门系统每年的维护与软件租赁费往往为数千万元台币。若走向开放式的主从式运算结构,则可能只需花费数百万元。除价格便宜外,软硬件设备使用开放结构,不再局限于单一的厂牌,就会享受最新的计算机技术与多元化的应用软件。至于仍未走向小型化的公司,其主要考虑因素为一些特殊安全性与可靠性的需求,在目前开放式的主从运算环境中仍无法找到适当的解决方法。

就台湾市场而言,不论技术与需求均比美国慢一至两年。但目前已有许多大企业走向小型化。此外,台湾的工商业是以中小型企业为主,开放式的主从运算环境应是这些公司计算机化非常理想的选择。

1.2 Windows NT 的机遇

从 Microsoft 的角度来看,上述的市场背景及趋势是创造目前 Windows NT 上市的最佳因素与时机。如图 1-2 所示,新一代个人计算机的技术与市场前景看好,若仍以目前的 DOS 与 Windows 作为操作系统,已无法满足高档 PC 使用者的需求。目前的 Windows 操作系统中网络、图像图形与多媒体件的处理上,功能尚属不足并且须搭配各种 Third-party 软件来使用。因此,需要新的操作系统以满足未来个人计算机在多媒体与网络上的应用。

Driving Force of Windows NT Market

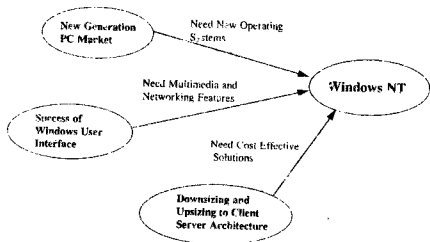


图 1-2 Windows NT 的市场契机

此外,以开放式的主从运算结构作为小型化的解决方案已经成熟。所需要的是价格便宜且功能强大的操作环境以满足目前资源短缺但业务量不断扩充情况下的各种应用。

Microsoft 此时推出 Windows NT,正好迎合了目前计算机市场的趋势及使用者的需求。Windows NT 是一全新的操作系统,其中结合了许多 UNIX 与大系统的技术,抛开了传统 DOS 的束缚,与 Microsoft 下一代操作系统的主力产品。Windows NT 中建立了丰富的网络功能,图形处理与多媒体功能,使 Windows NT 在高档 PC 与网络应用上具有强大的竞争力。最后,Windows NT 与外围配合的软硬件也比目前 UNIX 与 Novell 的解决方案便宜许多。这正是目前各公司考虑使用 Windows NT 的主要原因。传统 UNIX 及大系统的应用软件与开发工具一般数十万至数百万台币。若采用 Windows NT,此类软件只需数万至数十万。

举 Database Server 为例,以 UNIX 为平台的各著名的 Database 均需一百多至二百多万台币(假设上百个使用者)。若使用以 Windows NT 为平台的 Microsoft SQL Server 则仅需三、四十万。此外,传统以 UNIX 作为 Server 平台,以 Windows 或 DOS 作为 Client 端使用者界面的解决方案,由于 Client 端与 Server 端分属不同厂牌操作系统,往往需在 Client 端装

Drivers(例如 TCP/IP、Net Library 等)或增加 Gateway 转换软件。此种方式增加了 Client 端软件的费用,并且会降低整体运行效率。若 Client 端的数量庞大,所需费用也相当可观。但采用 Windows NT 作为 Server 的平台,至少 Client 端与 Server 端的通信无须额外的费用。在与 UNIX 及其他系统的连接上,也可通过 Windows NT 内建立的各种通信界面作传递。

当然 Windows NT 在新一代计算机市场中也有其较弱的一些环节。例如,作为一个新的产品,使用者是否能接受仍有待时间的验证。此外,Windows NT 也需面对众多强有力的竞争者,例如:Novell NetWare、IBM OS/2 与各厂商的 UNIX 操作系统。其中 Novell NetWare 占全世界近 70% 的文件服务器市场,OS/2 在 IBM 的推广下也不容忽视,UNIX 更是目前各类 Application Servers 软件的主要平台。环顾左右,Windows NT 的确面临着一场硬仗。

但 Windows NT 也并非等闲之辈。Microsoft 带其 Windows 3.X 具有 PC Client 操作系统的优势,所推出之 Windows 家族(即 Windows 3.1、Windows For Workgroup、Windows NT 与 NT Advanced Server)可提供一致的使用者界面与连接环境。此外,Windows NT 的 Symmetric Multiprocessing(SMP)功能与正起飞 MP 市场可相互搭配也是 Windows NT 吸引人之处。在功能方面,Windows NT 具备 Preemptive Multitasking 的功能并支持 Multiprocess 与 Multithread 的执行能力。Windows NT 内建有 TCP/IP、NetBEUI、SPX/IPX、AppleTalk 与 DLC 等不同的网络协议层,以及 Windows Sockets、NetBIOS、Named Pipe、Mailslot、WNet 与 RPC 等通信界面。此处,以 Windows NT 为平台的软件开发环境更是非常丰富,例如:Win32 SDK(Software Development Kit)、Visual C++ for NT、NT DDK(Device-driver Development Kit)、MDK(Multimedia Development Kit)、ODBC(Open Database Connectivity) SDK 及 WOSA(Windows Open Services Architecture)中所包含的各种界面与 APIs。综观上述之功能与特性,Windows NT 应足以作为新一代 Power Client 与 Server 的操作系统。

在应用软件方面,目前许多知名的软件已经或正在转移至 Windows NT 平台上,例如:AutoCAD、MS SQL Server、IBM's CICS、SunSelect's Wabi、Computer Associates'UniCenter... 等。相信将有更多的应用软件会不断地转移至 Windows NT。目前 Windows NT 在 Power Client 的应用上已渐渐地被接受,但在 Application Server 的应用仍有待更进一步地得到市场验证。

1.3 国内厂商的机会

对国内一般计算机与通信业者而言,Windows NT 应用技术层次远高于传统 MS DOS 与 MS Windows,所需投入的 Effort 较大。但 Windows NT 的应用软硬件开发与国外同期,可掌握商机。对国内从事多媒体人而言,Windows NT 为极佳的多媒体软件平台。Microsoft 的目标就为多媒体,故国内多媒体产品必须与 Microsoft 产品兼容。Windows NT 应用技术与在其他技术配合方面,许多原本在 PC 上无法开发的软硬件技术,可因 Windows NT 的出现而获得实现,有利于擅长 PC 软硬件的国内厂商。另外,将中大型软件移植到 Windows NT 上,可使软件更 Scalable,例如:分散式应用软件、网络软件、UNIX 应用软件等。但是,有一点国内软件厂商应该注意。那就是 Microsoft 本身也开发应用软件,例如:有关网络的 LAN Manager,有关 MIS 的 SQL Server、DCA/MS Communication Server,有关办公室自动化的 MS Office、Project、Mail、Scheduler,有关 Groupware 的 Windows For Workgroup 等。Micr-

soft 所开发的应用软件自然会借助 Windows 与 Windows NT 成为各应用软件领域的主流。这也将成为全球各软件厂商的威胁。然而,从反方向看 Micorsoft 软件,大都为 Package 式的中型软件,因此国内软件人员应避免开发同类的 Package,应开发 Add-on 软件以增加硬件的附加价值及开发 High-end 应用系统。

1.4 结 论

从上述的市场背景及趋势,我们可看出微软此时推出 Windows NT,不论在硬件的配合或市场的需求上均处于相当有利的地位。台湾是全世界最主要的 PC 硬件产地,近年来当局也在大力提倡通信业,厂商应把握现有基础,提高技术,开发适合市场需要的软硬件产品,为台湾在这新来临的契机中创造最佳的利益。

第二章 Windows NT 的功能与特性

Windows NT 在 Microsoft 的产品中被定为高档(High-end)个人计算机与服务器(Server)的操作系统。虽然许多 Windows NT 的功能与特性已出现在 OS/2、NetWare 与 UNIX 等操作系统中,但没有任何系统能像 Windows NT 一样连接如此丰富的功能。这些内建的 Windows NT 的功能提供给个人计算机软件一个更广阔的发挥空间。Windows NT 的使用者可享受到与原 Windows 3.1 一致的操作方式与更强的系统功能。程序开发人员也可利用这些功能开发许多以往在 PC 上无法实现的软件。

本章将深入探讨 Windows NT 的主要功能与特性,以帮助读者了解 Windows NT 是否为一适合您的操作系统。Windows NT 的功能主要可从下列项目做探讨:

- 均衡式多重处理(Symmetric Multiprocessing,简称 SMP)能力
- 多种硬件结构的支持
- 多样的保护子系统(Protected Subsystems)
- Windows NT Executive
- 内建的网络能力

2.1 均衡式多重处理功能

均衡式多重处理所代表的意义:中断式的多工(Preemptive Multitasking),多重执行单元(Multi-thread)、多重程序(Multi-process)及多重处理器(Multi-Processor)等性质。也就是在 Windows NT 下可容许多个 Processes(想当于执行文件)同时执行。每一个 Process 包含多个 Threads(相当于 Subroutines),而 Threads 平均分布在一个或多个 Processors 上执行。执行的方式则是利用 Time Sharing 的方式由这些 Threads 共同分享有限的 Processor(或 CPU)处理时间。

2.1.1 Preemptive 与 Non-preemptive 的多重处理

所谓中断式的多工,简单地讲就是每隔一段时间,执行中的 Task 会被硬件中断并将 CPU 的控制权交给其他的 Task 断续执行(参见图 2-1)。此处的 Task 可为整个的程序或一段类如 Subroutine 的程序。在 Windows NT 的定义里,Task 属于后者,称为 Thread。

至于硬件如何中断执行中的程式,可参见图 2-2。图中 Windows NT 的多工以 Thread 为基本单元。每一个 Process 在开始执行时,其本身即为一个 Thread,称为 Main Thread。在执行过程中,Main Thread 可产生其他的 Threads,而这些 Threads 可产生新的 Threads,依此类推。所有的 Threads,不论是属于应用程序或操作系统本身,可共同分享 CPU 的资源。其中 Threads 的产生与结束由通过 Windows NT System Services 中的 Process Manager 来控制。至于 Threads 间执行顺序与在不同 CPU 上执行的同步(Synchronization)工作则由 Windows NT Kernel 负责。有关 System Services 与 Kernel 将在 2.3 节中详细叙述。

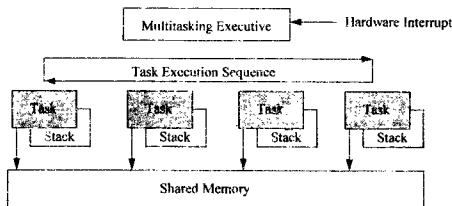


图 2-1 Windows NT 的多工环境

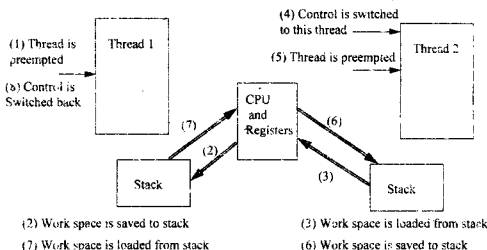


图 2-2 中断式的多工处理(Preemptive Multitasking)

另一种多工方式称为非中断式多工(Non-preemptive Multitasking)。所谓非中断式多工,如图 2-3 所示,Thread 程序必须主动放弃 CPU 的控制权,其他 Thread 才有机会执行。Windows 3.1 即为此类型的多工。而 Windows NT 中也保留此功能。当 Windows 程序执行 GetMessage、MessageBox、PeekMessage 与 DialogBox 系列的 APIs 时,调用的 Thread 会自动放弃 CPU 的控制权。

Threads 间 CPU 控制权的转换过程称为 Context Switching。其动作是将被中断的 Thread 的执行状况存在属于 Thread 的 Stack 中,并把下一个执行 Thread 的环境从相对的 Stack 中取出。Context Switching 所产生的 Overhead 对于实时系统而言固然重要,但不至于成为系统的瓶颈。此 Overhead 因操作系统与使用的 CPU 而不同。以 25MHz 80386CPU 为例,一般使用于工业计算机之多的操作系统中最佳可达 33 μ S。至于 Windows NT,目前仍缺

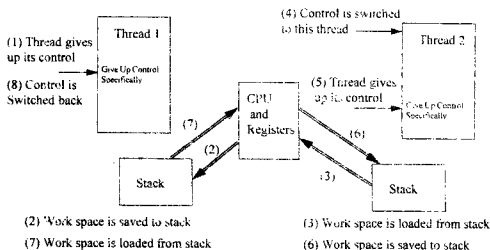


图 2-3 非中断式的多工处理(Non-Preemptive Multitasking)

少这方面数据。但以 Windows NT 所需的硬件配备来看,此 Overhead 应不至于影响 Windows NT 的 Performance。

在 Windows NT 系统中,中断式与非中断式的多工可同时存在,互不冲突。中断式的多工使 Threads 间的执行得以均衡,而非中断式的多工使 Thread 在空闲时,可主动将 CPU 让给其他 Threads 执行,以提高系统整体的 Throughput。在中断式的多工环境中,每一个 Thread 一次执行的时间为 10 至 20Milliseconds(视硬件结构而定),称为一个 Time Slice。当 Thread 放弃其执行权时,其剩余的 Time Slice 即交给下一个 Thread 执行,如图 2-4 所示。

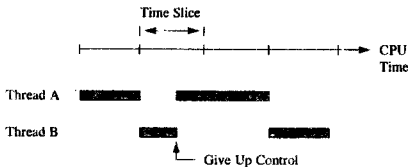


图 2-4 中断式多工环境下 CPU 的时间分配

2.1.2 Thread 与 Process 的特性

图 2-5 中,程序中的主程序为 Main Thread, Main Thread 产生了数个 Threads。其中有些 Threads 共用同一段 Code(亦即 Thread body),但为不同的 Instances。每一个 Thread 均有对应的 Stack,以储存 Thread 的执行环境,称为 Thread's Context。Thread's Context 包含诸

如 Program Counter, Stack Pointer, Register Set 及属于此 Thread 的 Local Variables 等数据。一个程序的 Process 包含了此程序中所有的 Threads 及定义此 Process 的一些数据结构。以下是 Process 与 Thread 在 Windows NT 环境中的特性。

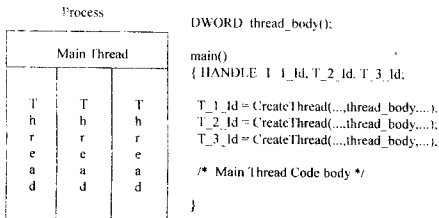


图 2-5 Threads 的产生

1. 每一个 Process 拥有独立的地址空间(Address Space), 为 4G Bytes, 但其中 2GB 由系统所使用, 真正程序仅能控制在 2GB。Process 中的 Threads 则共享此 2GB 的地址空间。
2. 一般而言, Process 均为独立的程序, 可能由不同的厂商所开发, 因此彼此间无同步问题。其执行顺序由系统所设定。而 Threads 间的同步由程序设计者所控制, 其执行顺序也可由程序本身来设定。
3. 由于 Windows NT 提供多重处理器的功能, 程序的 Threads 可在不同的 CPU 上执行, 提高平行处理的效益。Windows NT 运用 Load Balance 技术使平行处理与 Inter-processor Communication 的效能尽可能地达到最佳化。
4. 在 Windows NT 的环境中, 每一个 Process 均有其执行的 Priority, 称为 Base Priority。Windows NT 提供 32 等级的 Priority (如图 2-6), 分为两类, 一类为 Real-time Priority (16~31), 另一类为 Variable Priority (1~15)。Priority 0 由系统使用。其中 Real-time Priorities 供 CPU 立即反应的 Real-time Thread 使用, Variable Priorities 供一般 Thread 使用。Process 中的每一个 Thread 也有其 Base Priority, 其等级范围在 Process 的 Base Priority 上二级与下二级之间。真正 Thread 执行时的 Priority 称之为 Dynamic Priority, 其范围由 Thread 的 Base Priority 至第 31 级, 视 Process 的 Priority 种类而定。系统可依 Thread 当时的执行状况动态的调整其 Priority。例如: 当 Thread 在处理使用者输入时, 为求快速回应使用者, Thread 的 Priority 便被提高。当 Thread 在作长时间的计算时, 其 Priority 便被降低。Process 与 Thread 的 Priority 也可由程序中控制。读者可参考 SetPriorityClass 与 SetThreadPriority 等 APIs。

Windows NT 处理 Thread 的执行顺序(Scheduling)采用 Round-robin Algorithm, 即高 Priority Threads 优先执行, 完毕后才会轮到低 Priority Threads 执行。同

等级 Priority 的 Threads 则同步执行。低 Priority 的 Threads 在等待被执行时,其 Priority 会自动被提高以确保每一个 Thread 都有机会使用 CPU。

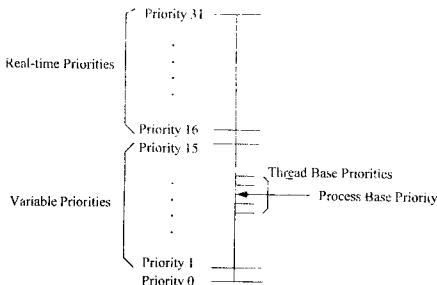


图 2-6 Windows NT 中 Threads 的 Priorities 等级

5. Threads 可共享 Global 数据结构与变量,作为彼此沟通的通道,而 Processes 必须通过系统提供的服务(例如:Win32 API 的 Pipe、DDE 及 Shared Memory 等方式)来完成。

在多重处理器(MP)的结构方面,Windows NT 所采用的 SMP 结构,将每一个 Processor 均视为相同。Windows NT Kernel 负责将 Threads 平均分布在各个 Processor 上执行。至于哪一个 Thread 分配哪一个 Processor,并无特定的安排。如图 2.7 所示,在 Windows NT 的环境中,无论是操作系统的 Threads 或应用程序的 Threads 都平均分布在数个 Processors 上。各类型的 I/O 也是由每一个 Processor 共同处理。

另一种多重处理器结构称为非均衡式多重处理(Asymmetric Multiprocessing,简称 ASMP),参见图 2-8。其原理是设定特定的 CPU 作特定的工作,例如专门负责执行操作系统的 CPU、I/O CPU 及应用软件使用的 CPU。ASMP 结构通常须在软硬件的搭配上作特别的处理,例如 NetWare 即采用 ASMP 结构。由于 NetWare 的市场很大,因此有专门为 NetWare 而设计的 MP 机器。至于 SMP 与 ASMP 哪个 Performance 好,乃是仁者见仁,智者见智。一般而言,ASMP 结构下的硬件与操作系统间必须经过测试(Tuning)方可发挥好的功效。正因为如此,采用 ASMP 结构的计算机在与各种操作系统搭配上不如 SMP 来得更有弹性。此外,当执行应用程序的 Processor 数量增加时,ASMP 结构下的操作系统不但无法分享此增加的资源,并有可能因负荷过重而造成瓶颈。比较起来,在扩充性与资源使用的效率上,SMP 比 ASMP 为好。

作者曾参加 Windows NT 与 MP 机器在实际应用上的测试工作,所测的其中一台机器为国产 Acer Altos 17000 主机,含四种 Pentium CPU,128M Bytes 的 RAM,4G Bytes RAID

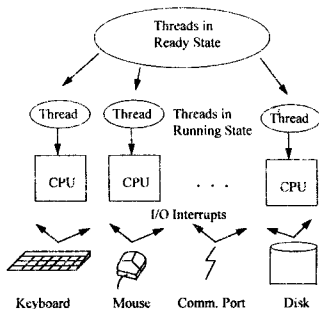


图 2-7 Symmetric Multiprocessing (SMP) 框架

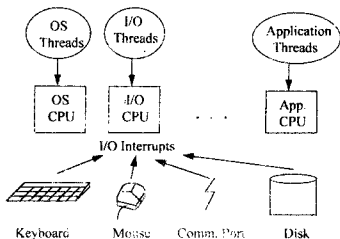


图 2-8 Asymmetric Multiprocessing (ASMP) 结构

Level 的硬件。测试软件包括 MS SQL Server 与多组 Win32 多工程序。在测试中,我们从 Windows NT 所提供的 Performance Monitor 中可以看出四个 CPU 的使用效率相当平均。由于 Windows NT 本身即包含一百多个 Threads,并同时执行,因此每个 CPU 所分配到 Thread 的数量差别不大,故可以均衡地处理。至于 CPU 的数量需要多少方能达到最佳的使用效率则须视所执行软件的负荷而定。一般而言,当 CPU 的使用率长时间地超过 80% 时,

可考虑增加 CPU 的数量。因此在选购 MP 机器时,也应考虑未来的扩充性。

2.2 多种硬件结构的支持

所谓硬件结构,此处是指以各种 CPU 为平台所设计的计算机。Windows NT 支持的 CPU 种类包括 Intel 486 及 Pentium, MIPS R4X00, DEC Alpha 与 IBM 及 Motorola 的 PowerPC。由于各种 CPU 的特性与执行指令不同,因此所设计出的计算机在应用上也有差别。例如 Intel 的 CPU 为 Windows NT 最早使用的处理器,适用于一般桌上型及高档个人计算机。MIPS R4X00 则是图像处理的能力好。DEC Alpha 则是强调其 IC 的设计与体积,适用范围涵盖 PC 与迷你级计算机。至于 PowerPC 则提供价格较便宜效能也不错解决方案。对 Windows NT 而言,它可在不同厂牌、不同结构上执行,提供应用软件一致的界面,增加软件的可移植性(Portability)。Windows NT 应用软件想在不同硬件平台上执行,虽必须有不同的版本,但软件的原始码(Source Code)仍是一致的。也就是说在 Pentium 机器上执行的程序,移到 PowerPC 机上,只要从新 Compile 后即可执行 Native Mode。

除了支持各种 CPU 外,Microsoft 欲使 Windows NT 能广泛在不同厂牌的计算机上执行,而开发了一套移植 Windows NT 的工具,称为 Hardware Abstraction Layer (HAL) Development Kits。Windows NT 的 HAL 分隔了 Windows NT 操作系统与硬件(参见图 2-9)。HAL 为一 DLL,内含一组数据结构与 Subroutines。此数据结构(例如,Loader Parameter Block)定义计算机与操作系统间的界面。

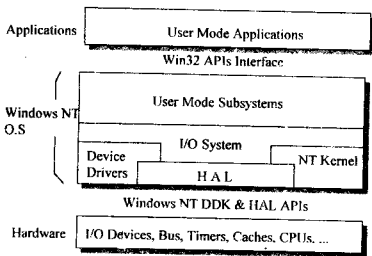


图 2-9 Windows NT Hardware Abstraction Layer (HAL)

HAL 中所提供的 Subroutines 被 NT Kernel 与 Device Drivers 所调用,以执行硬件相应的动作。因此,只要提供不同的 HAL, Windows NT 便可在各种硬件平台上执行。

除了计算机厂商可依据 HAL 将 Windows NT 移植到各种计算机上外,外国设备的厂商如:鼠标、图像扫描器、打印机以及各种网络通信设备等,可依照 Microsoft 的 NT Device-

driver Development Kits(NT DDK)来开发专用的 Drivers,NT DDK 包含了各种 Drivers 的范例、规格(如 NDIS、VGA 等)及调试工具。Microsoft 为了达到 Windows NT 能广泛地被硬件所支持,在与硬件的连接上花了许多功夫。传统的 Device Drivers 由于缺乏标准,往往造成不同厂商的硬件不兼容的问题。Microsoft 制定统一的 DDK,以解决此方面的问题。

2.3 多样的保护子系统

Windows NT 本身包含有许多子系统(Subsystems),又称为 Server Process(参见图 2-10)。每一个子系统为一 Process,拥有自己独立的地址空间。子系统对外的通信均被保护,其他 Process 不经过一定方式无法进入,因此称其为保护子系统(Protected Subsystems)。当某个子系统清除时,并不影响其他子系统的执行。此清除的子系统也可重新启动。子系统又称之为 Server Process,其主要的工作是提供服务给 Client Processes 及其他的子系统。Windows NT 包括下列子系统:

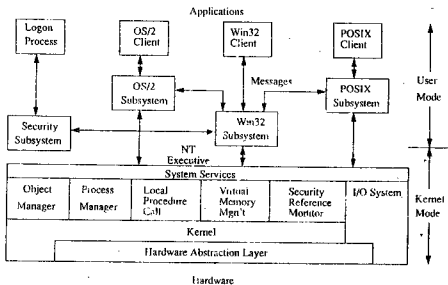


图 2-10 Windows NT 的子系统(Subsystems)

1. 安全子系统(Security Subsystem):其功能为管理整个系统的安全,例如 Event Log、Error Log、Access Control 等,并提供服务给 Logon Process。此 Logon Process 提供多用户(Multi-user)的特性。使用者在进入 Windows NT 时,必须先以 Administrator、User 或 Guest 等身份 Logon。不同身份具备不同等级的使用权限。Windows NT 本身拥有一个数据库,记录各种使用者的身份和 Configuration。不同的使用者可对系统设定自己的 Configuration。Windows NT 在确认身份后,便会将所属的 Configuration 展现给使用者。
2. OS/2 子系统:此子系统目的在于支持 OS/2 应用程序的执行。凡 OS/2 1.3 版或以前的应用软件(仅限于 Character-based)均为 Binary Compatible。所谓 Binary Compatible

ble, 即指程序无需重新 Compatible 便可在 Windows NT 上执行。

- Win32 子系统: 此子系统为 Windows NT 中最重要的 Subsystem, 主要提供服务给 Win32 Clients, DOS 与 Win16 的应用软件, 并且均为 Binary Compatible。Win32 子系统也提供 I/O Redirection 服务给 OS/2 子系统与 POSIX 子系统, 以维持统一管理的人机界面。有关 Win32 子系统的介绍, 请参见 2.3.1 节。
- POSIX 子系统: 其功能在于支持 POSIX Client 的执行。凡 UNIX 的应用软件(仅限于 Character-based), 依据 POSIX (Portable Operating System Interface for UNIX) 的标准, 则为 Source Code Compatible, 即程序经 Re compile 后, 便可在 Windows NT 上执行。Windows NT 提供此系统的意图在于抢占部分 UNIX 市场。

上述的各子系统与 Client Processes 之间及子系统彼此之间以信息传递(Message Passing)方式来沟通。此方式是通过 System Services 的 Local Procedure Call 完成(参见图 2-11)。Clients 由传递 Messages 向子系统要求服务, 而不直接调用系统的核心部分, 避免程序间不正确的调用。Windows NT 的这种 Client-Server 结构, 使得操作系统具有良好的扩充性。由于 Client Process 均通过子系统来操作, 而子系统又为独立的 Process, 因此只要不断地扩充或增加子系统而不需修改 Windows NT 的 Kernel, 便可使操作系统提供更多的服务。此 Client-Server 结构提供了操作系统的安全性, 但是在效率上必须付出一些代价。不过对于一个高档操作系统而言, 此负荷可由硬件设备补偿, 故无须多虑。

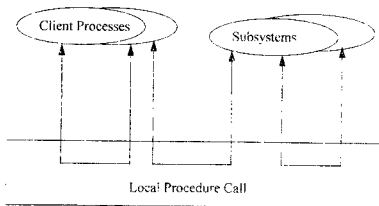


图 2-11 Processes 通过 Local Procedure Call 执行 Message 传递

2.3.1 Win32 子系统

Win32 子系统仍延续 Windows 3.1 的概念, 分为下列五个部分:

- Console: 负责管理 Text Windows(例如 DOS Prompt)中字符的输出/输入。
- Windows Manager: 相当于 Windows 3.1 的 USER.DLL, 称为 USER32.DLL。Windows Manager 负责管理所有的视窗显现, 应用程序的输出与输入, 以及 Clipboard 与程序间的数据传输。此外, Windows Manager 还支持 APIs 供应用程序处理视窗画面。Windows Manager 在管理视窗显示上, 追踪视窗的大小、颜色、内容重叠的状态。

当使用者改变视窗的显示状态时, Windows Manager 则会通知应用程序作适当的处理与重画动作。

3. Graphics Device Interface(GDI):此部分也是 DLL,称为 GDI DLL。GDI 包含一组丰富的 APIs 供 Windows Manager 与应用程序调用,以执行绘图的工作。GDI 除了提供线、图形、符号、字形与多媒体数据的显现功能外,并具备复杂的图形运算能力。例如,Bezier Curve,Bitmap 图形的旋转与 Matrix Transformation 等复杂的数学运算,均可通过 GDI 作有效的处理。GDI 并不直接控制硬件,而是调用 Graphics Device Drivers。
4. Graphics Device Drivers:Graphics Device Drivers 通过调用 NT Device Drivers 来控制硬件的动作。如此,GDI 与 NT Device Drivers 间各自独立,对于未来 APIs 与 Drivers 的扩充均有帮助。
5. Operating System Functions:此部分的 DLL 称为 Kernel32.DLL,负责所有的 I/O 处理,系统 Objects 操作(将于 Object Manager 中说明),Thread 操作,Threads 同步处理,存储器管理与资源安全管理等工作。此 DLL 负责调用 NT Executive,为 Windows NT 子系统与 NT Executive 间的枢纽。

2.3.2 DOS 与 Win16 应用程序的执行环境

如前所述的 Win32 子系统支持 DOS 与 Win16 应用程序的执行。其执行方式是通过虚拟的 Virtual DOS Machine(VDM)与 Windows On Win32(WOW)(如图 2-12 所示)。VDM 为一 Win32 的 Process,模拟 DOS 的执行环境。因此每一个在 Windows NT 下执行的 DOS 程序均为一个独立的 Process。当使用者开启一个 DOS Prompt 时,Win32 子系统则产生一 VDM 供此 DOS 应用程序使用。在 Windows NT 环境下,可同时存在多个 VDM,并以中断式的多工方式执行。在这些 VDM 中执行的 DOS 程序,彼此间互传数据,并可通过 Clipboard 与 Windows 程序互相通信。

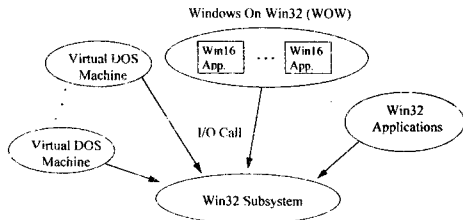


图 2-12 Virtual DOS Machine(VDM)与 Windows On Win32(WOW)

由于每一个 VDM 均具有独立的地址空间,因此 DOS 程序与 Drivers 必须包含在每一个 VDM 中。换句话说,每一个 VDM 是一个虚拟的 DOS 操作系统,模拟 X86 CPU 的执行环境。

至于 Win16 应用程序的执行,则是通过 WOW。它与 VDM 不同之处是所有的 Win16 程序只在单一的 WOW 环境下执行。WOW 环境也是 Win32 的一个 Process,当使用者开启第一个 Win16 程序时,Win32 子系统即产生一个 VDM Process,此 VDM 并将 WOW 环境装入。此 WOW 环境包含了 Windows 3.1 Kernel 与 DOS 程序以模拟 Win16 的执行环境。

由于 Win16 为一非中断的多工环境,WOW 在此方面需要做特殊的处理。每当使用者开启一个 Win16 程序时,此开启动作首先由 Win32 子系统接到,Win32 子系统便传送一 Message 给 WOW,WOW 随即将此 Win16 程序码装入并产生一个 Thread 来执行此程序。虽然 Windows NT 为中断式多工处理,但 Win16 的 Threads 在执行时仍采取非中断式多工方式,以便与 Windows 3.1 兼容。如图 2-13 所示,WOW 只有一个 System Message Queue 供所有 Win16 程序使用。当某一个 Win16 程序处理 Message 时,其他的 Win16 程序则停止执行,直到执行中的 Win16 程序处理完 Message 后,其他 Win16 程序方有机会执行。但由于每一个 Win16 程序均为一个 Thread,而 Windows NT 的中断式多工处理也是以 Thread 为单位。因此每一个 Win16 程序仍会被 NT Kernel 中断(Interrupt)。只是中断后下一个轮到执行的 Win16 程序若仍在等待其他 Win16 程序交出执行权时,则 WOW 便自动将此段 Time Slice 分配给正在执行的 Win16 程序,以便其快速完成 Message 的处理(参见图 2-14)。此种安排的程序的方式让所有的 Threads,不论 Win32 Threads 或 Win16 Threads,均公平地分享 CPU 时间。此外,Win16 的非中断式多工处理模式也不会影响整体的效率。

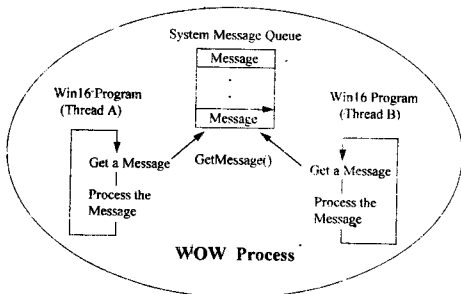


图 2-13 Win16 程序在 WOW 中执行的情况