

Die Windows NT Architektur

Die hohen Ansprüche, die Microsoft an Windows NT von Anfang an stellte, ließen sich nicht über eine konventionelle Betriebssystemarchitektur erfüllen. Um ein Höchstmaß an Kompatibilität, Portabilität, Stabilität und Performance zu erreichen, wurde ein neuer Ansatz entwickelt, der in den folgenden Abschnitten erläutert wird.

Windows NT besteht aus mehreren voneinander mehr oder weniger abhängigen Teilsystemen, die selbständig spezielle Aufgaben übernehmen. Die einzelnen Komponenten entsprechen bildhaft gesprochen einem Zahnrädchen in einem Uhrwerk. Nur durch eine koordinierte Zusammenarbeit können die Aufgaben des Betriebssystems effizient erfüllt werden.

Der Name NT steht für New Technology. Der ursprüngliche Name stand jedoch für Northern Telecom. Diesen Namen hat Microsoft jedoch schnell in New Technology umgetauft (die Rechte an diesem Namen sollen Microsoft große Geldsummen gekostet haben).

Architekturkonzept von Windows NT

Windows NT ist ein Betriebssystem, was nach dem Client / Server Modell entworfen wurde. Bevor auf das Client / Server Modell eingegangen wird, soll noch ein Überblick auf die anderen Ansätze, ein Betriebssystem zu entwerfen, gegeben werden.

Monolithischer Aufbau

Bei einem monolithischen Aufbau werden eine Reihe von Prozeduren realisiert, die sich gegenseitig aufrufen. Diese Prozeduren haben unbedingten Zugriff auf Systemdaten und Hardware. Da diese Systeme störanfällig sind und nicht gemäß C2 Sicherheit kontrolliert werden können, konnte Windows NT nicht in monolithischer Bauweise erstellt werden.

Schicht Aufbau

Dieser Ansatz geht von übereinander geschichteten Modulen aus. Die einzelnen Module beinhalten Funktionen, die von höheren Schichten aufgerufen werden können.

Client / Server Aufbau

Bei diesem Modell werden verschiedene Dienste von einzelnen Modulen zur Verfügung gestellt, z. B. Dienste zur Speicherverwaltung, Prozeßerzeugung oder Prozeßsteuerung bei Multiprozessor-Betrieb. Jeder der Dienste wird als Server bezeichnet und kann von verschiedenen Clients abgefragt werden. Diese Server laufen im User-Modus und haben so keinen direkten Zugriff auf Speicher und Hardware. Alle Server überprüfen in einer Schleife, ob ein Client einen ihrer Dienste anfordert. Die Clients, bei denen es sich z. B. um Anwendungen handelt, benutzen

diese Dienste, indem sie Nachrichten mit den zu bearbeitenden Aufgaben an den Systemkernel senden. Der Systemkernel, der im Kernel-Modus läuft, leitet diese Anfragen an die entsprechenden Server weiter. Nachdem die Server die Aufgabe ausgeführt haben, leitet der Kernel die Antwort an den Client zurück.

Windows NT beinhaltet sowohl Bestandteile des Schichten-, sowie des Client/Server Modells. Der Teil von Windows NT, der im Kernel Modus läuft, wird als Executive bezeichnet. Hier finden Komponenten wie die Speicherverwaltung, die Objektverwaltung, die Ein/Ausgabe, das Dateisystem, die Netzwerktreiber, die Prozeßkommunikation und Teile des Sicherheitssystems ihren Platz.

Der HAL

Microsoft verfolgte von Anfang an das Konzept, ein Betriebssystem zu entwickeln, das auf unterschiedlichen Plattformen eingesetzt werden konnte. Um eine Unabhängigkeit vom dem zugrundeliegenden Prozessor und dem Bustyp zu erreichen, hat Microsoft zwei wichtige Voraussetzungen geschaffen. Zum einen wurde das gesamte Betriebssystem vollkommen in C geschrieben, so daß die wesentlichen Programmteile leicht portiert werden können, zum zweiten wurde festgelegt, daß alle Elemente des Betriebssystems, die sich auf eine bestimmte Hardwareplattform beziehen, in einen gesonderten Bereich einfließen müssen, den sogenannten HAL = Hardware Abstraction Layer. Möchte nun ein Entwicklungsteam Windows NT auf eine neue Plattform portieren, so muß im Endeffekt „nur“ die C-Programme rekompiliert und eine neue HAL generiert werden. Der HAL dient als Schnittstelle für die meisten Betriebssystemprozesse und übernimmt die Kommunikation mit der Hardware. Nur wenige Bestandteile des Kernels und die Gerätetreiber kommunizieren direkt mit der Hardware, ohne den HAL zu nutzen.

Die verschiedenen HAL Typen

HAL DLL	Rechnertyp
HALAST.DLL	AST Manhattan SMP
HALSP.DLL	Compaq Systempro oder 100% kompatibel
HALCBUS.DLL	Corollary C-Bus Architektur
HALMCA.DLL	IBM PS/2 oder anderer MCA PC
HALMPM.DLL	Micro Channel Multiprozessor PC
HALAPIC.DLL	MPS Uniprocessor PC
HALNCR.DLL	NCR System 3000 Model 3360/3450/3550
HALOLI.DLL	Olivetti LSX5030/40
HAL.DLL	Standard PC (Single Prozessor HAL)
HAL486C.DLL	Standard PC mit C-Step i486
HALWYSE.DLL	Wyse Serie 7000i Model 740MP/760MP

Ein wenig bekannte Tatsache ist, daß sich die HAL Versionen von den Prozessorvarianten unterscheiden. Eine Einprozessor-HAL unterscheidet sich von einer Multiprozessor-HAL. Microsoft stellt drei verschiedene Varianten von NT zur Verfügung. Eine Einprozessorversion, eine Multiprozessorversion und eine Test-(Debug)-Version. Neben den unterschiedlichen HALs haben Einprozessor- und Multiprozessorsysteme auch unterschiedliche NTOSKRNL-Images. Die Einzelprozessor-Version verfügt nicht über den Code für eine vernünftige

Ausführung von NT auf einem Multiprozessorsystem. Die Testversion von Windows NT liefert die Möglichkeit, sowohl auf Einprozessor- als auch auf Multiprozessor-Systemen zu laufen. Diese Version läuft jedoch sehr instabil. Wollen Sie Ihre Single-Prozessor Version auf eine Multiprozessor-Version aufrüsten, so haben Sie die Möglichkeit ein Programm aus dem Windows NT Resource Kit zu nutzen (UPTOMP.EXE), welches eine Aufrüstung von Single- auf Mehrprozessorsystem ohne Neuinstallation ermöglicht. Bedenken Sie jedoch hierbei, daß es sich um ein Tool handelt, welches gravierende Änderungen an Ihrem System vornimmt und von Microsoft nicht offiziell unterstützt wird. Die ausgetauschten Dateien werden in das Verzeichnis UPTOMP.OLD unter %SYSTEMROOT%\SYSTEM32 kopiert.

Der Aufwand für eine Portierung von Windows NT auf eine neue Plattform kann gewaltig schwanken. Eine Portierung innerhalb einer Prozessorfamilie (z.B. Intel 86) ist natürlich wesentlich einfacher als eine Umsetzung auf eine vollkommen andere Prozessorfamilie (z.B: MIPS, Alpha etc.).

Welche Probleme das unter Umständen bewirkt, wird dadurch deutlich, daß HP den eigenen Prozessor erst einmal modifizieren mußte, damit eine Portierung von NT überhaupt erst möglich ist.

Microsoft hat Ende 1996 die Entwicklung und Portierung von Windows NT auf den Prozessor von MIPS eingestellt.

Die Entwicklung von Windows NT für den Power PC wurde von Microsoft auch bereits vor einiger Zeit eingestellt, so daß sich die vielgerühmte Hardware-Portabilität von Windows NT derzeit auf die Intel- und die Alpha Plattform beschränkt.

Multiprozessor-Nachrüstung bei Windows NT

Bei Windows NT führen mehrere Möglichkeiten zum Ziel eine vorhandene Single Prozessor-Installation auf eine Multiprozessor-Version ubzudaten.

Der einfachste Weg besteht darin, einfach eine zweite CPU auf das vorhandene Multiprozessor-Motherboard zu setzen. Voraussetzung hierfür ist, daß die Installation von Windows NT bereits mit diesem Multiprozessor-Motherboard durchgeführt worden ist.

In einem solchen Fall installiert Windows NT nämlich bereits die für Mehrprozessorbetrieb notwendigen Module, unter anderem spezielle Versionen des Kernels, des HAL und einigen Systembibliotheken. Sie erkennen und nutzen einen hinzugekommenen Prozessor automatisch.

Wer aber erst später auf ein Mehrprozessor-Motherboard wechselt, die anderen Komponenten wie Hostadapter, Platten und dergleichen aber nicht anrührt, kann eine vorhandene NT-Installation trotz neuen Boards in der Regel weiterhin booten.

Ein Utility namens „UPTOMP“ aus dem Windows NT Resource Kit stellt ein Single-Prozessor-System auf SMP-Betrieb um: das Programm ist außerdem vielerorts im Internet erhältlich. Doch Vorsicht: Die erste UPTOMP-Version war fehlerhaft und

vergaß, die Datei win32k.sys zu kopieren (ein Blick in die Datei UPTOMP.INF verschafft Gewißheit, ob es sich um eine alte Version handelt). Das Programm UPTOMP.EXE ersetzt folgende Dateien aus dem Windows NT System:

- ?? Hal.dll
- ?? Ntoskrnl.exe
- ?? Ntkrnlmp.exe
- ?? Ntdll.dll
- ?? Kernel32.dll
- ?? Winsrv.dll
- ?? Win32k.sys

Die aktuelle Datei UPTOMP.INF erhalten Sie über den FTP-Server bei Microsoft.

Für die Durchführung der Prozessorüstung sind noch einige weitere Schritte notwendig:

Kopieren Sie die Dateien UPTOMP.INF und UPTOMP.EXE in ein neues Verzeichnis auf den Datenträger. Kopieren Sie die neue Datei UPTOMP.INF in das Verzeichnis. Kopieren Sie die Datei TXTSETUP.SIF von der Original Installations-CD in das neu angelegte Verzeichnis. Kopieren Sie alle Dateien des aktuellen eingespielten Service Packs in das Verzeichnis und starten von dort das Programm UPTOMP.EXE. Bei der Frage nach dem HAL-Pfad geben Sie den Pfad zum SP-Verzeichnis an. Wählen Sie die richtige HAL aus. Nach dem Kopieren der Daten muß der Rechner neu gestartet werden.

Oder . . .

Basteln ist angesagt

- ?? Stellen Sie fest, welchen HAL Sie benötigen
- ?? Installieren Sie NT auf den gleichen Rechner in ein anderes Verzeichnis
- ?? Im %SYSTEMROOT%\SYSTEM32 Verzeichnis benennen Sie folgende Dateien um:

- NTOSKRNL.EXE
- HAL.DLL
- KERNEL32.DLL
- NTDLL.DLL
- WINSRV.DLL
- WIN32K.SYS (nur erforderlich bei NT 4.0)

- ?? Kopieren Sie die Datei NTKRNLMP.EXE und den entsprechenden HAL von der Windows NT CD oder dem zuletzt eingespielten SP in das %SYSTEMROOT%\SYSTEM32 Verzeichnis und benennen Sie die Dateien NTOSKRNL.EXE und HAL.DLL um.

- ?? Kopieren Sie die Datei NTDLL.DLL von der Windows NT CD oder dem zuletzt benutzen Service Pack in das %SYSTEMROOT%\SYSTEM32 Verzeichnis.
- ?? Von einem Command-Prompt aus, expandieren Sie die Dateien KERNEL32.DL_, WIN32K.SY_ und WINSRV.DL_, welche sich auf der Windows NT CD befinden, und kopieren diese in das %SYSTEMROOT%\SYSTEM32 Verzeichnis.
- ?? Starten Sie den Computer neu
- ?? Wenn Windows NT startet, erscheint auf dem Bildschirm der Hinweis auf den Multiprozessor-Kernel.

Als letztes probates Mittel, ein NT unter geänderten Hardwarevoraussetzungen wieder in Betrieb zu nehmen, stellt ein Update dar: Startet man das NT-Setup Programm nach erfolgtem Umbau, bietet es normalerweise ein Update bestehender NT-Installationen an, sofern es solche auf der Festplatte findet. So kann man eine Installation auf einem nachträglich eingebauten Zwei-Prozessor-Board selbst dann wiederbeleben, wenn sich zum Beispiel der Typ des SCSI-Adapters ändert.

Ein Makel der SMP-Unterstützung von Windows NT am Rande: Wenn eine Installation einmal umgestellt wurde, läuft sie nicht mehr auf Ein-Prozessor-Boards, sehr wohl aber auf Dual-Boards, auch wenn auf diesem nur ein Prozessor steckt. Wer wieder weg will von SMP-Betrieb, muß manuell die gesicherten Dateien restaurieren oder sich des zuvor beschriebenen Tricks bedienen.

Wer test halber Windows NT mit nur einem Prozessor betreiben will, braucht die CPU nicht auszubauen, sondern bedient sich des nicht dokumentierten Eintrags /ONECPU in der Datei BOOT.INI.

Exkurs

Upgradeproblem NT

Allein die Grafikkarte unter NT ist kein Problem. Neue Hardware rein, Windows NT im VGA Modus starten, Treiber wechseln und den Rechner neu starten.

Auch Netzwerkkarten sind in der Regel unter NT kein Problem. In der Systemsteuerung die alte Netzwerkkarte entfernen, Rechner neu starten, Netzwerkkarte einbauen und neuen Treiber über die Systemsteuerung installiere. Allerdings gehen dabei sämtliche Einstellungen und Protokollbindungen verloren, so daß Sie sich vorher die Einstellungen notieren sollten.

Bei SCSI-Controllern wird es erheblich kritischer. Ein passabler Weg ist folgender: SCSI Treiber deinstallieren, NT Herunterfahren und anschließend über die Installationsdisketten die Installation „reparieren“. Beim Wechsel von Symbios Logic Controllern auf Adaptec hilft bei den bekannten Kompatibilitätsproblemen nur das Backup des Systems und anschließende Neuinstallation.

Der Austausch eines Motherboards endet für NT meist tödlich, weil die für das alte Board installierten Treiber meist nicht mehr mit dem neuen Motherboard funktionieren. Da hilft oft nur ein komplettes Backup und anschließende Neuinstallation.

Exkurs

Unter Windows NT können Sie mit Hilfe der Registry oder Applikationen von Drittanbietern festlegen, welche Applikationen auf welchen Prozessoren in einem Multiprozessorsystem laufen sollen.

Der NT Kernel

Der Windows NT Kernel oder auch der Kern übernimmt die Steuerung des Prozessors und ist somit plattformunabhängig. Der HAL und der NT Kern gehören untrennbar zusammen. Teile des Kerns, der durchgängig in Assembler programmiert wurde, befinden sich in dem HAL. Zu den Aufgaben, die der Kernel übernimmt, gehören die Bearbeitung von Interrupts und Ausnahmen, die Bereitstellung von Schnittstellen und Objekten, die Synchronisation von Prozessen und die Steuerung der Threads.

Der Windows NT Kernel und der Hardware Abstraction Layer (HAL) wurden auf optimale Interrupt- und Ereignis- Verteilung hin konzipiert. Der Kernel ermöglicht die Interrupt Verteilung für das gesamte restliche System. Der Kernel kann jeden, der von Windows NT unterstützten 52 Interrupts für seine Funktion verwenden. Der Kernel reserviert jedoch nur acht Interrupts für sich selbst. Die nächsten 24 Interrupts werden den Hardware-Interrupts, die den HAL benutzen, zugeordnet.

Windows NT behandelt Interrupts auf preemptiver Basis. Wenn ein Interrupt angefordert wird, werden alle kleineren (die mit der geringeren Priorität) Interrupts unterbrochen und die Interrupts mit der höheren Priorität bedient. Hierzu wird der Prozeß-Manager von Windows NT verwendet.

Verwendung der IRQs

Interrupt	Definition
IRQ 31	Hardware Error Interrupt
IRQ 30	Powerfail Interrupt
IRQ 29	Inter-Prozess Interrupt
IRQ 28	Clock Interrupt
IRQ 12-27	Diese Interrupts werden den traditionellen Hardware IRQs eines PCs zugeordnet
IRQ 4-11	Diese Interrupts werden normalerweise nicht genutzt
IRQ 3	Software Debugger Interrupt
IRQ 0-2	Reserviert für Software Interrupts um die Arbeit von Device Drivern und Komponenten der Executive zu priorisieren

Das Prioritäten-Modell von Windows NT beinhaltet 32 Priority Levels, von denen 16 für das Betriebssystem und die Real Time Prozesse reserviert werden.

Sie müssen sich hierbei darüber im klaren sein, daß ein Unterschied zwischen Real Time Prozessen und Real Time Anwendungen (die Windows NT ja nicht unterstützt), besteht.

Prozesse, Threads und Fibers

Windows NT ist ein Multitasking- Multithreading-Betriebssystem.

Prozesse ermöglichen in einem Multitasking-Betriebssystem das quasi gleichzeitige Ablaufen von mehreren Applikationen. Prozesse stellen aber vergleichsweise hohe Anforderungen an die Systemressourcen, wenn sie gestartet werden. Zudem sind strenge Sicherheitsmechanismen notwendig, um die Prozesse klar und eindeutig voneinander zu trennen, die Prozesse in separaten Adreßräumen laufen zu lassen und eine Kommunikation der Prozesse untereinander über das LPC-Facility zu ermöglichen.

Threads dagegen ermöglichen ein quasi gleichzeitiges Ablaufen von Programmbestandteilen innerhalb einer Applikation. Das macht nur dann Sinn, wenn das Erzeugen von Threads wesentlich weniger Aufwand bereitet, als das anlegen eines Prozesses.

Unter einem Thread hat man sich nichts anderes vorzustellen, als einen Teil des Programmflusses, der separat vom Scheduler des Betriebssystems verwaltet wird. Dazu muß jeder Thread seinen Zustand definieren können, der durch einige Objekte klassifiziert wird:

- ?? einen eigenen Stack
- ?? eine Priorität für das Scheduling
- ?? einen Befehlszeiger
- ?? einen kompletten Registersatz
- ?? spezieller Speicher, der für den Thread lokal zur Verfügung gestellt wird
- ?? zugehörige Semaphore für die Synchronisation von Threads

Alle anderen Ressourcen der zugehörigen Prozesses teilt der Thread mit den anderen Threads. Dazu gehören der Speicher, die offenen Handles auf Objekte, Prozesse der Interprozesskommunikation, Pipes und Synchronisationsobjekte wie Semaphore und die typischen Umgebungsvariablen.

Ein weiteres Argument für die Verwendung von Threads ist die Multiprozessorfähigkeit von Windows NT.

Kommt ein NT-System mit mehreren Prozessoren zum Einsatz und sollen die Applikationen gleichmäßig auf alle CPUs verteilt werden, muß die Applikation in mehrere Threads aufteilbar sein. Denn die kleinste Einheit, die Windows NT verwalten kann ist nicht der Prozess, sondern der Thread.

Generell ist es nicht sinnvoll, eine Anwendung in mehr Threads aufzuteilen, als CPUs in einem SMP System zur Verfügung stehen. Doch darf man nicht vergessen, daß es sich nicht bei allen Programmen um die reine CPU-Ausnutzung dreht. Das Scheduling der Threads übernimmt das Betriebssystem. Dabei verfolgt es die Vorgaben, die ihm die Prioritäten der Threads auferlegen.

Die Berechnung der aktuellen Prioritätsstufe geht nach folgendem Muster: zuerst nimmt das Betriebssystem den Wert für die Prioritätsklasse des Prozesses, in dem die Threads erzeugt werden. Dazu addiert es die Prioritätsebene des Threads. Liegt

die Priorität über dem Wert 11, kommen die Threads sogar dem Betriebssystem in die Quere - was die Zuteilung der Ressourcen angeht.

Ein weiterer Faktor, der die Priorität eines Threads bestimmt, ist der Priority-Boost. Diesen Wert legt das Betriebssystem fest. Sinn dieses Priority-Boosts ist es, auch Threads mit geringerer Priorität einen Boost zu verpassen, sprich, das dieser Prozess auch einmal von der CPU bedient werden kann.

Threads sind sehr leichtgewichtige Strukturen. Noch schlanker als Threads sind Fiber, die es seit dem Service Pack 3 für Windows NT 3.51 gibt. Ein Fiber ist ein Thread, dem keine Rechenzeit zugewiesen wird. Vielmehr sind es parallel ablaufende Prozeduren in einem Thread-Objekt, auf die vom Programmierer manuell die Rechenzeit des Threads verteilt wird.

Real Time ? -oder nicht ?

Die Frage, ob Windows NT ein Real Time Betriebssystem ist oder ein Prozess in Real Time ausgeführt werden kann, läßt sich mit einem klaren Nein beantworten.

Wenn man von den Definitionen von Real Time Betriebssystemen ausgeht, ist Windows NT kein Hard Real Time Betriebssystem, sondern eher ein Soft Real Time Betriebssystem. Windows NT ist ein Vielzweck Betriebssystem das die Fähigkeit hat, für schnelle Antwortzeiten zu sorgen, aber es ist nicht als Hard Real Time Betriebssystem konzipiert worden.

Windows NT ist ausgehend von den Definitionen eher ein Soft Real Time Betriebssystem. Ein Soft Real Time Betriebssystem zeichnet sich durch reduzierte Latenzzeiten sowie schnelle Antwortzeiten aus, ohne das „faire“ Arbeitsverhalten (preemptives Multitasking) zu vernachlässigen.

Soft-Real-Time Betriebssysteme zeichnen sich durch eine Latenzzeit von ca. 10-100 ms aus, was für wirkliche Realtime-Anforderungen nicht ausreicht.

Echte Realtime-Betriebssysteme hingegen zeichnen sich durch Latenzzeiten von unter 1 ms aus. Der Bereich zwischen 1 ms und 10 ms ist fließend und teilweise definitionsabhängig.

Windows NT arbeitet mit preemptiven Multitasking und offeriert gar Prioritätsebenen, die die Kennzeichnung „Realtime“ tragen. Doch für deterministische Reaktionszeiten eignet sich Windows NT nicht.

Um nun die harten Echtzeitanforderungen erfüllen zu können und zugleich das weit verbreitete Windows-NT Umfeld nicht verlassen zu müssen, treten Realtime-Erweiterungen für Windows NT auf den Plan. Dabei hilft die modulare Architektur von Windows NT und die Unterstützung von Subsystemen.

Um Echtzeit-Anwendungen unter Windows NT zu ermöglichen, wurde ein neues API geschaffen, das sich sehr stark an das Win32-API anlehnt und so die Spezifikationen von Windows NT nicht all zu stark durcheinander bringt, daß RTAPI - Real Time Application Programming Interface. Beide APIs greifen auf eine leichte

Modifikation der HAL um Funktionsänderungen wie Timer, Taktgeber und Interrupt-Verwaltung zu. Der Win32 Prozeß im Umgebungssystem nutzt das RTAPI, daß auf RTAPI-Treiber für den Echtzeitprozeß und das Echtzeit-Subsystem im Kernel-Modus zugreift.

Der echtzeitfähige Thread-Manager bietet 128 verschiedene Prioritätsstufen für die Echtzeit-Threads an, und kontrolliert alle Aufgaben im Zusammenhang mit der Prioritätszuteilung.

Zuordnung von Prozess-Prioritäten

Obwohl das System die Fähigkeit besitzt, alle 32 Prozess-Ebenen direkt zu verwenden, können Sie nur begrenzt Prioritäten setzen.

Standardmäßig haben alle durch Benutzer und Administrator ausgeführten Anwendungen eine Basispriorität von 8(normal).

Benutzer können Applikationen mit einer Priorität 4(low) oder 13(high) ausführen. Ein Administrator kann sogar eine Priorität 24(realtime) für die Ausführung einer Anwendung erhalten.

Seien Sie jedoch vorsichtig bei der Vergabe solch hoher Prozess-Prioritäten, weil es sein kann, daß das System dann gar nicht mehr reagiert. Folgende Syntax ist auf der Kommandozeilenebene notwendig, wenn Sie die Prioritäten für die Ausführung eines Programmes ändern wollen:

Start /low /normal /high /realtime Anwendung

Prioritätsklassen von Prozessen

Beschreibung	Wert
Idle	4
Normal	9
Normal (bei Hintergrundverarbeitung)	7
High	13
Realtime	24

Prioritätsklassen von Threads

Beschreibung	Wert
Idle	1
Lowest	-2
Below Normal	-1
Normal	0

Above Normal	+1
Highest	+2
Time Critical	15

Dispatcher-Objekte

Die verschiedenen Dispatcher-Objekte haben eins gemeinsam: Sie besitzen einen Signalstatus, der den Zustand „signalisiert“ oder „nicht signalisiert“ haben kann.

Zu den Dispatcher-Objekten zählen:

Dispatcher-Objekt	Beschreibung
Ereignis	Das Auftreten eines Ereignisses wird vermerkt. Mit Auftreten des Ereignisses kann eine bestimmte Aktion erfolgen
Mutant	Mutant-Objekte werden in der Regel im Benutzer-Modus eingesetzt. Die Mutant-Objekte sind dafür zuständig, die vom Programmierer festgelegte Synchronisation von Threads untereinander durchzuführen
Mutex	Die Mutex-Objekte erfüllen die gleiche Aufgabe wie die Mutant-Objekte, jedoch werden diese im Kernel-Modus ausgeführt
Semaphore	Die Semaphore sorgt dafür, daß nur so viele Threads zum dem Prozess durchgelassen werden, wie dies von dem Prozess festgelegt worden ist
Thread	Ein Thread ist die kleinste ausführbare Einheit innerhalb eines Prozesses (einer Anwendung)
Zeitgeber	Der Zeitgeber wird dazu verwendet um abgelaufene Zeiten zu protokollieren und Timeout-Operationen auszuführen
Abschnitt	Abschnitte stellen in der Regel kurze Befehlsfolgen dar

Exkurs

Bei konstanter Manpower gilt: je weniger Code, um so weniger Fehler. Einige Orientierungsgrößen: NT 4.0 hat etwa 15 Millionen Zeilen Code. Schätzungen für Windows NT 5.0 belaufen sich auf etwa das doppelte. Ein typisches, kommerzielles Unix benötigt zwischen 2 und 10 Millionen Zeilen Code, ein Echtzeitkern nur etwa 10.000 bis 100.000 Zeilen Code. Der pSos Kern, der Software Component Group, eine der wenigen Stücke Software, die mit Garantie für Fehlerfreiheit ausgeliefert werden, hat gar nur 5.500 Zeilen Code. Natürlich hat das seinen Preis: Die Größenordnung liegt bei ca. 1 DM/Byte. Demnach müßte ein fehlerfreies NT ca. 30 Millionen kosten, ein fehlerfreies Unix hingegen zwei bis zehn Millionen DM. Das würde die leidige Diskussion Unix versus NT beenden.

Die Gerätetreiber

Die Gerätetreiber entsprechen ihren Pendants aus anderen Betriebssystemen. Sie kümmern sich um die Steuerung einzelner Hardwarekomponenten wie etwa ein CD ROM Laufwerk. Sie sind natürlich auch hardwareabhängig, aber sie gehören prinzipiell nicht zum Betriebssystem und sind deshalb auch nicht in das HAL Konzept eingebunden worden. Es versteht sich von selbst, daß

Hardwarekomponenten, die auf verschiedenen Plattformen eingesetzt werden können, unterschiedliche Treiber benötigen.

Der NT Executive

Die Aufgabe der NT Executive ist es, grundlegende Betriebssystemfunktionen zur Verfügung zu stellen, auf denen die Subsysteme aufsetzen können. Dazu gehören folgende Aufgaben:

- ?? Management des Dateisystems
- ?? Management des Speichers
- ?? Management der Prozesse
- ?? Management der Objekte
- ?? Eingabe-/Ausgabesteuerung (I/O Management)
- ?? Management der Sicherheit
- ?? Unterstützung der Prozesskommunikation

Die Subsysteme nutzen die Funktionalität des NT Executive und bereichern sie um spezifische Komponenten an. Der NT Executive steuert sämtliche Aktivitäten des Betriebssystems. Über ihn werden alle Speicher- Festplatten und sonstige Ressourcenzugriffe gesteuert. Für die verschiedenen Aufgaben stehen dem Executive verschiedene Manager zur Verfügung, deren Arbeit er koordiniert.

Der NT Executive wird auch als der Kernel Mode Bestandteil von Windows NT bezeichnet, denn er läuft im privilegierten Modus in Ring 0. Damit ist er vor Programmfehlern in den Subsystemen, die im Ring 3 ausgeführt werden, vollkommen geschützt. Diese strikte Trennung ist eine wesentliche Voraussetzung für einen stabilen Betrieb, birgt aber einen gewissen Mehraufwand in sich, da ein Wechsel zwischen den Ringen des Prozessors eine Reihe von Prozessen veranlaßt, die logischerweise auch den Prozessor beanspruchen. Windows NT wurde so konzipiert, daß möglichst wenige Wechsel notwendig sind.

Der Objekt Manager

Der Objekt-Manager dient zur Verwaltung der Objekte der Executive. Dazu muß man wissen, daß Objekte unter Windows NT der Darstellung sämtlicher Systemressourcen dienen, deren Aufbau für die Systeme, die im Anwendermodus ablaufen, verborgen bleiben. Objekte sind beispielsweise Hauptspeicherbereiche, Dateien und Verzeichnisse, Grafikkarten und andere Geräte.

Sinn und Zweck des Objekt Managements ist es, einem Prozeß, der im Anwendermodus abläuft, die verschiedenen Ressourcen in einheitlicher Form zur Verfügung zu stellen.

Die Aufgabe, die der Objekt Manager dabei übernimmt, ist es, Objekte zu erzeugen und zu verwalten und am Ende auch zu löschen.

Der Objektmanager führt Objektmanagementaufgaben wie die Identifikation und die Verweiszählung aus. Wenn eine Anwendung eine Ressource öffnet, dann findet der Objektmanager entweder das zugehörige Objekt oder er generiert ein neues. Statt einen Objektzeiger an die Anwendung zurück zu senden, die die Ressource öffnete,

sendet er einen erkennbaren Identifikator, den man Handle nennt. Für die Anwendung, welche die Ressource öffnete, ist der Wert des Handles einmalig, nicht jedoch systemweit für die verschiedenen Anwendungen. Die Anwendung benutzt eine Behandlungsroutine um die Ressource in aufeinanderfolgenden Operationen zu identifizieren. Sobald die Anwendung das Objekt nicht mehr benötigt, schließt sie den Handle. Der Objektmanager benutzt den Verweiszähler, um zu verfolgen, wie viele Teile des Systems darunter Anwendungen und Teilsysteme der Ausführungsschicht, auf ein Objekt zugreifen, das eine Ressource darstellt. Wenn der Verweiszähler auf Null geht, dann ist das Objekt, das die Ressource darstellt, nicht mehr in Gebrauch und der Objektmanager löscht das Objekt (aber nicht notwendigerweise die Ressourcen).

Objekte der Executive (Ausschnitt)

Objekt der Executive	Definiert durch	Verwendung
Access Token	Sicherheitssystem	Identifizierung eines Anwenders
Datei	I/O Manager	Instanz einer geöffneten Datei oder eines I/O Gerätes
Event	Systemdienste	Ankündigung eines Systemereignisses
Port	LPC-Nachrichtenschnittstelle	Ziel einer Nachricht zwischen zwei Prozessen
Prozeß	Prozeß-Manager	Zu startendes Programm inkl. Adreßraum und Ressourcen, die benötigt werden
Semaphore	Systemdienste	Zähler, der den Zugriff eines Threads auf eine Ressource beschränkt
Thread	Prozeß-Manager	Ausführbare Einheit in einem Prozeß
Timer	Systemdienste	Zähler für die Zeitmessung

Der Prozeß-Manager (preemptives Multitasking)

Der Prozeß-Manager ist verantwortlich für die Verwaltung der Prozesse und deren Threads. Sobald durch ein Subsystem ein Prozeßstart verlangt wird, wird der Prozeß-Manager aktiv und übernimmt die gesamte Steuerung und Abwicklung dieses Prozesses. Das bedeutet, daß der Prozeß-Manager auch für die Initialisierung und auch die Beendigung von Threads verantwortlich ist, genauso wie er auch die Beendigung des Prozesses anstoßen muß. Sämtliche Informationen über die aktiven Prozesse und deren Threads werden im Prozeß-Manager verwaltet.

Der Ein-/Ausgabe-Manager

Der I/O Manager hat die Aufgabe, den einzelnen Subsystemen eine einheitliche Schnittstelle für die Eingabe und Ausgabe von Daten zu bieten. Er sorgt auch ein Stück weit für die schon erwähnte Geräteunabhängigkeit.

Der I/O Manager übernimmt eine Schlüsselrolle in Windows NT, da eine saubere und geordnete Interaktion mit den peripheren Geräten von entscheidender Bedeutung für die Funktionalität eines Betriebssystems ist.

Als Grundlage des I/O Managers dient das IRP (I/O Request Package), das die einzelnen Komponenten untereinander austauscht.

Untrennbar zum Ein-/Ausgabe-Manager gehören die Treiber, die für die Steuerung der spezifischen Hardware verantwortlich sind.

Das Windows NT Betriebssystem unterscheidet drei verschiedene Typen von Treibern, wie etwa

Dateisystemtreiber (NTFS.SYS, FASTFAT.SYS)

Faulttolerance-Treiber (FTDISK.SYS)

und die Einheitentreiber (z.B. AHA154X.SYS)

Ferner gehört der Cache-Manager untrennbar zum I/O Manager, denn er sorgt dafür, daß nicht jede I/O Prozedur einzeln durch den Treiber abgearbeitet werden muß.

Der Sicherheitsmonitor

Dieses Element des NT Executive ist für die Gewährleistung der Sicherheit auf dem lokalen Server oder der lokalen Workstation verantwortlich. Er überprüft Anmeldungen von Benutzern und überwacht die Ressourcen des Betriebssystems. So steuert er beispielsweise die Zugriffssicherheit auf bestimmte Verzeichnisse, deren Benutzung explizit vom Administrator eingeschränkt wurden. So etwas ist beispielsweise sinnvoll, um eine unbefugte Veränderung an den Systemdateien auf jedem Fall zu vermeiden.

Eine weitere Aufgabe des Sicherheitsmonitors ist der Schutz der Objekte, die vom Objekt Manager gesteuert werden, während der Laufzeit des Systems.

Der Security Reference Monitor ist eng mit dem Objekt-Manager verknüpft. Der Objekt-Manager ruft den Security Reference Monitor zur Prüfung des Zugriffs auf, bevor er Anwendungen weitere Operationen wie Lese- oder Schreibzugriffe auf das Objekt genehmigt.

Der Security Reference Monitor verwendet ein Sicherheitsmodell das auf Sicherheitsidentifikatoren (SIDs) und wahlfreien Zugriffsteuerungslisten (DACLS) beruht. Jede ACL enthält verschiedene ACE (Access Control Entries). Jeder Prozess in NT ist mit einem Zugriffseintrag verbunden, das die SID enthält, die den Benutzer identifiziert, der den Prozess besitzt, sowie die SIDs der Gruppe, zu der der Benutzer gehört.

Der Speicher Manager

Der Speicher Manager von Windows NT benutzt ein virtuelles Speicher Management und wird deshalb auch Virtual Memory Manager (VMM) genannt. Da Windows NT ein 32 Bit Betriebssystem ist, kann es 2³² Byte Speicher Adressen verwalten. Auf diese Weise ermöglicht es eine Verwendung von 4 GB, wobei 2 GB von Anwendungen genutzt werden, können und der Rest nur durch das System, sprich für den Executive, verfügbar ist.

Der Bereich für das System ist in drei große Blöcke aufgeteilt. Zwischen 2 GB und 3 GB befindet sich der Adreßraum für den Kernel, Ausführungsschicht, HAL und Boot-Treiber. Daran schließt sich ein Bereich von 8 MB für Prozesseitentabellen und andere systeminterne Tabellen an, die als Hyperspace bezeichnet werden.

Um das System zu verstehen, müssen Sie die Struktur einer virtuellen 32-Bit-Adresse betrachten:

- ?? Der erste Teil der Adresse, der bei Intel-Systemen aus 10 Bit und bei Alpha-Systemen aus 8 Bit besteht, gibt die Position im Seitenverzeichnisindex an.
- ?? Der zweite Teil der Adresse, der 10 bzw. 11 Bit hat, gibt die Position im Seitentabellenindex an.
- ?? Der dritte Teil gibt schließlich den Byte-Index an und besteht aus 12 bzw. 13 Bit.

Windows NT kann grundsätzlich mehr Speicher verwalten und nutzen, als tatsächlich an RAM im System installiert ist. Dies geschieht mit Hilfe der Auslagerungstechnik des Virtueller Memory Managers (VMM), wobei sämtliche Anwendungs- und Benutzerspeicherbereiche ausgelagert werden können. Jener Bereich, in dem sich der Kernel befindet, kann allerdings vom System nicht ausgelagert werden (Stabilitäts- und Performance-Gründe)

Für die Applikationen auf dem Server oder der Workstation steht damit wesentlich mehr Hauptspeicher zur Verfügung, als tatsächlich an physischem RAM im Rechner installiert ist. Damit können speicherintensive Anwendungen wie etwa Datenbankmanagementsysteme befriedigt werden.

Der Virtual Memory Manager (VMM) setzt virtuelle Adressen für Anwendungen in physisches Seiten des Computerspeichers um. Auf diese Weise verbirgt es die Struktur und Verwaltung des realen Hauptspeichers vor den Applikationen. Das stellt wiederum sicher, daß bei Speicheraufrufen durch die Anwendungen keine Konflikte entstehen können.

Bei Intel-Systemen sind die Speicherseiten 4096 Byte groß. Bei Alpha-Systemen hingegen 8192 Byte groß.

Pro Prozess gibt es ein Seitenverzeichnis. Dieses kann bis zu 1024 Einträge bei Intel- bzw. 256 Einträge bei Alpha-Systemen enthalten. Es erfaßt die Adressen aller Seitentabellen eines Prozesses. Die Adresse des richtigen Seitenverzeichnisses für die verschiedenen Prozesse wird von der CPU verwaltet.

Das Seitenverzeichnis verweist über sogenannte PDEs (Page Directory Entries) auf Seitentabellen. Von diesen Seitentabellen gibt es bis zu 512 pro Prozess und weitere 512 systemweite, die wiederum aus bis zu 1024 Einträgen bei Intel, bzw. 2048 Einträgen bei Alpha-Systemen bestehen können.

Die Seitentabellen beschreiben, auf welcher Seite des physischen Speichers sich bestimmte Informationen befinden.

Bei der Benutzung der Auslagerungsdatei verwendet Windows NT das demand paging, eine Methode, bei der Daten in Seitenform aus dem physischen Speicherbereich auf eine temporäre Swap Datei auf der Festplatte ausgelagert werden. Sobald eine Anwendung auf eine ausgelagerte Seite zugreifen möchte, wird diese wieder in den Hauptspeicher zurück gelesen. Unter NT wurde der Paging Algorithmus so verbessert, daß nun die Auslagerung und das Einlesen von Seiten nun nicht mehr auf Gesamtsystembasis geschieht, sondern sich auf den einzelnen Prozeß bezieht.

Windows NT analysiert die Verwendung der sogenannten Prozessarbeitsseiten (Working Sets). Das sind die Seiten, die einem Prozess als physische Seiten im Hauptspeicher zugestanden werden. Der minimale vorhandene Wert variiert je nach vorhandenem physischen Speicher zwischen 20 und 50 Arbeitsseiten. Der maximale Wert wird bei der Initialisierung des Systems auf 45 bis 345 gesetzt. Jede der Seiten hat eine Größe von 4 KB bei Intel-Systemen und 8 KB bei Alpha-Systemen, woraus sich ergibt, daß ein Prozess bis zu 1380 KB RAM belegen darf. Er kann aber, entsprechende Berechtigungen vorausgesetzt, auf bis zu 1984 MB bei Intel-Systemen angehoben werden. Das System überprüft bei knappem Speicher nun, wie viele Seiten ein Prozess tatsächlich nutzt, und lagert die Seiten (LRU, Last recently used) aus, auf die am längsten nicht zugegriffen wurde.

Die Prozeßkommunikation

Der NT Executive muß dafür sorgen, daß die einzelnen Prozesse miteinander kommunizieren können. Zu diesem Zweck verwendet er das Local Procedure Call (LPC) Facility, das eine Variante des Remote Procedure Call darstellt.

Bei der Verwendung der Subsysteme von Windows NT entsteht ein großer Kommunikationsbedarf zwischen den einzelnen Prozessen auf einer Maschine. Das LPC Facility ermöglicht diese Kommunikation, bewirkt aber einen gewissen Overhead.

Das Dateisystem

Eine weitere Komponente des NT Executive ist die Verwaltung des Dateisystems. In einem Dateisystem gibt es feste Definitionen zur Verwaltung von Dateien und Verzeichnissen. Je nach Dateisystemen gibt es sehr unterschiedliche Modelle, mit denen die physische Festplatte genutzt wird. Windows NT unterstützt neben dem eigenen NTFS Dateisystem weitere Dateisysteme wie etwa das von DOS bekannte FAT. In der Windows NT Architektur werden die Dateisysteme mit Hilfe von speziellen Treibern zur Verfügung gestellt (FASTFAT.SYS).

Mit der 4.0 Version von NT wird das OS/2 eigene HPFS nicht mehr unterstützt.

Der Cache-Manager

Der Cache-Manager arbeitet eng mit der virtuellen Speicherverwaltung und den Dateisystemtreibern zusammen. Er verwaltet den globalen, also den allen Dateisystemen zur Verfügung gestellten Dateisystem-Cache. Der Cache in Windows NT ist eher dateiorientiert und nicht, wie in Windows 95, blockorientiert.

Die Subsysteme

Windows NT kann ganz im Gegensatz zu anderen Betriebssystemen als Plattform für verschiedene Applikationen dienen. Unter Windows NT ist es möglich, sowohl MS-DOS 16Bit und 32 Bit Windows als auch zeichenorientierte OS/2 und POSIX Anwendungen zu nutzen. Erreicht wird die Kompatibilität durch die gezielte Unterstützung von mehreren APIs (Application Programming Interface). Ein API dient als Schnittstelle zum Betriebssystemkern, ermöglicht eine einfache Emulation alter Betriebssysteme und bietet die Möglichkeit einer Kompatibilität von zukünftigen Betriebssystemen. Dieser Ansatz ist nicht vollkommen neu und wurde maßgeblich von den Erkenntnissen aus der Entwicklung des Mach Mikrokernels beeinflusst.

Funktionsweise der Subsysteme

Applikationen, die auf einem Windows NT Computer eingesetzt werden, arbeiten nicht direkt mit dem NT Executive, sondern setzen auf den Subsystemen auf. Ein großer Teil der Systemaufrufe einer auf Windows NT ablaufenden Applikation geht direkt an eines der Subsysteme, das dann diese Anforderungen an den entsprechenden Service des Executive weitergibt.

Windows NT stellt heute folgende Subsysteme zur Verfügung:

- ?? Win32
- ?? OS/2
- ?? POSIX

Die Subsysteme sind der Schlüssel zur Kompatibilität von Windows NT. Derzeit können eine Vielzahl von MS-DOS-, 16 Bit- und 32 Bit- Windows- sowie zeichenorientierte OS/2- und POSIX Applikationen unter Windows NT eingesetzt werden.

Die Aufgabe der Subsysteme läßt sich an einem Beispiel sehr gut darstellen. Wenn eine OS/2 Anwendung eine Operation auf einem Datenträger vornehmen will, dann gibt sie den Systemaufruf direkt an das entsprechende Subsystem weiter. Das OS/2

Subsystem setzt den OS/2 spezifischen Systemaufruf in einen NT spezifischen um und gibt ihn an die NT Executive zurück, welches wiederum mit der OS/2 Applikation kommuniziert. Man kann also von einer echten Client/Server Beziehung zwischen Applikationen und Subsystemen sprechen. Durch diesen Mechanismus ist sichergestellt, daß die einzelnen Applikationen der unterstützten Betriebssystemplattformen die von Ihnen erwartete Umgebung vorfinden.

Windows NT bietet ein hohes Maß an Kompatibilität, wie die Anzahl der unterstützten Betriebssysteme zumindest auf dem Blatt verspricht.

Das Win32 Subsystem

Die einzigen Plattformen, die im nahezu uneingeschränkten Umfang auf die volle Leistungsfähigkeit von NT zurückgreifen können, sind Win32 und mit Einschränkungen auch 16 Bit Windows und DOS Applikationen. Diese Tatsache hängt damit zusammen, daß sämtliche Hauptfunktionen im Win32 Subsystem enthalten sind. Es übernimmt:

- ?? die Abwicklung von 32 Bit Windows Applikationen
- ?? die Abwicklung von 16 Bit Windows Applikationen
- ?? die Abwicklung von MS-DOS Applikationen
- ?? die Eingabe- und Ausgabe und
- ?? die grafische Darstellung.

Es existiert kein getrenntes MS-DOS oder Win16 Subsystem. Wenn eine MS-DOS Anwendung gestartet wird, dann läuft diese als Win32 Prozeß. Das hängt zusammen, daß das Programm, mit dem die virtuelle DOS-Umgebung (Virtual DOS Machine VDM) geschaffen wird, eine Win32 Applikation ist. Dieses Programm heißt NTVDM.EXE und setzt 16 Bit Windows Systemaufrufe in 32 Bit Aufrufe um und wickelt diese im Win32 Subsystem ab. Für jedes DOS Programm öffnet NT eine neue VDM-Umgebung, so daß die einzelnen Anwendungen voneinander geschützt sind und das preemptive Multitasking nutzen.

16 Bit Windows Anwendungen werden nach demselben Schema abgearbeitet, mit dem Unterschied, daß standardmäßig alle Applikationen in einer VDM ausgeführt werden und damit auch nur das Nonpreemptive (kooperative) Multitasking zurückgreifen können. Auf Wunsch können Sie aber einer 16 Bit Windows Anwendung in einer eigenen VDM laufen lassen, ohne daß auf OLE- oder DDE Funktionalität verzichtet werden muß.

Das Win32 Subsystem ist allein verantwortlich für die Abwicklung von I/O Prozeduren und für die grafische Steuerung. Wenn eine OS/2 oder POSIX Applikation diese Funktionen nutzen möchte, so muß es zwangsläufig das Win32 Subsystem nutzen. Für die Steuerung der Ein- Ausgabeprozeduren verwenden das OS/2 und das POSIX Subsystem sogenannte Local Procedure Calls, die auf die Funktionalität des Win32 Subsystems zurückgreifen können. Die Steuerung dieser Kommunikation zwischen den Subsystemen übernimmt das LPC Facility der Executive.

Das OS/2 Subsystem

Um ein Mindestmaß an Kompatibilität zu bestehenden OS/2 Applikationen zu erreichen und um sicherlich Signale hinsichtlich der universellen Verwendbarkeit von Windows NT zu setzen, wurde auch das OS/2 Subsystem realisiert.

Es erlaubt standardmäßig eine Nutzung von zeichenorientierten OS/2 1.x Applikationen. Presentations-Manager (Vorgänger der OS/2 Warp Oberfläche (Desktop)) Anwendungen können nur mit Hilfe eines von Microsoft zur Verfügung gestellten Add Ons genutzt werden.

Das OS/2 Subsystem wird genauso wie das POSIX Subsystem automatisch beim Starten von Windows NT geladen. Wenn das System feststellt, daß sie nicht benötigt werden, lagert es die beiden sofort aus dem physischen Hauptspeicher aus und verwendet sie erst dann wieder, wenn ein entsprechendes Programm aufgerufen wird.

Das POSIX Subsystem

Das POSIX Subsystem unterstützt derzeit nur die POSIX 1003.1 Systemdienste. Das ist gleichbedeutend mit der Tatsache, daß auf die Netzwerkfunktionalität, die Sicherheitsmechanismen und das Multithreading nicht zugegriffen werden kann. Auch bezüglich der grafischen Darstellung ist POSIX problematisch. Es kann die Grafikfunktionalität des Win32 Subsystems nicht nutzen, und es gibt derzeit kein Add On zur Behebung dieses Problems.

Weitere Subsysteme

Windows NT verfügt neben den schon erwähnten Subsystemen, die in erster Linie der Kompatibilität dienen, über weitere Subsysteme wie etwa das Sicherheitssystem. Sie stellen oberhalb des NT Executive gewisse Funktionen für die Anwendungen und andere Subsysteme zur Verfügung.

Wertung der Subsysteme

Die Subsysteme von Windows NT bieten eine grundsätzliche Kompatibilität zu anderen Betriebssystemen. Microsoft hat aber verständlicherweise das Hauptengagement auf die Entwicklung des Win32 Subsystems gelegt und dafür gesorgt, daß dieses möglichst performant die volle Funktionsfähigkeit ausnutzen kann. Aus diesem Grund werden auch Elemente wie die I/O Steuerung oder die Grafiksteuerung in das Win32 Subsystem integriert, um den Kommunikationsaufwand so klein wie möglich zu halten.

Die beiden OS/2 und POSIX Subsysteme sind in Ihrer Funktionalität deutlich eingeschränkt und können sicherlich nicht als vollständig kompatible Plattformen gelten. Microsoft weist darauf hin, daß es sich bei dem OS/2 und POSIX Subsystemen eben nur um Emulationen handelt, die nichts anderes tun, als Schnittstellen zu bieten, die auf anderen Betriebssystemen schon existieren.

Architektonische Unterschiede Windows NT Workstation / Server

Der Datei-Vergleich Windows NT Server / Windows NT Workstation ergibt interessante Unterschiede.

Ca. 2.500 Dateien unterscheiden sich zwischen beiden NT Versionen nicht. Ein Binär-Vergleich zwischen den Dateien NT Server und NT Workstation ergibt, daß sich ca. 200 Dateien unterscheiden.

Ca. 100 Dateien werden nur mit NT Server ausgeliefert (überwiegend die zusätzlichen Komponenten wie DHCP, WINS, DNS, GSNW, Macintosh usw.). Ca. 10 Dateien werden mit der Workstation ausgeliefert, welche kein digitales Equivalent bei Windows NT Server haben (unterschiedliche HLP-Dateien usw.).

Die anderen ca. 90 Dateien haben unterschiedlichen plattformspezifischen Inhalt und können nicht verglichen werden.

Wie erkennt das Betriebssystem welches System es ist

Bis zur Version NT 3.51 gab es nur einen Registry-Schlüssel, der die Versions-ID von NT aufwies. NT 4.0 hat jetzt zwei Schlüssel in der Registry. HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\ProductOption – ProductType (WINNT, SERVERNT, LANMANNT) und einen Eintrag, ob es sich um NT 3.51 oder 4.0 handelt.

Operating System Tuning

Der NT Initialisations-Prozess greift auf den Eintrag MmProductType und MmIsThisAnNtAsSystem() zu. Diese beinhalten ca. 25 interne Variablen. Die NT Executive, der Memory-Manager, die File-System-Treiber, der Prozess-Manager, der I/O Manager, der Cache- und der Objekt-Manager werden optimiert, wenn der Produkttyp NT Server erkannt wurde.

In Abhängigkeit von dem Produkt-Type wird auch die Systemspeichergröße angepaßt. Folgende Tabelle gibt Aufschluß über die Unterschiede:

System Size	Workstation (MB)	Server (MB)
Klein	0-19	0-19
Mittel	20-32	20-64
Groß	>32	>64

Arbeits-Threads

Critical Worker Threads

System Size	Verzögerte Worker Threads	WS	SRV
Klein	3	3	3
Mittel	3	3	6
Groß	3	5	10

Windows NT Quantum-Längen

Die Quantum-Länge in Windows NT Server und Workstation bestimmt die zeitliche Länge eines Threads unter Windows NT, bevor der Scheduler von NT den Thread zugunsten eines anderen Threads gleicher oder höherer Priorität verdrängt.

Die Quantum-Längen in Windows NT Workstation und Server 4.0 sind fest (Hardcoded) und können nicht so ohne weiteres geändert werden.

Die Quantum-Length von NT Workstation und NT Server unterscheiden sich. Windows NT Workstation verwendet kürzere Quantum-Length als Windows NT Server mit dem Hintergrund, daß auf einem Server die Priorität auf Netzwerkanwendungen und Prioritäten liegt.

Die Workstation Quantum-Length beträgt 6 – 12 – 18
Die Server Quantum Length beträgt 36 – 36 – 36

Eine Änderung der Quantum-Length unter Windows NT 4.0 ist nur mit Third-Party Software wie NTFROB von Mark Russinovich ([HTTP://WWW.SYSINTERNALS.COM](http://www.sysinternals.com)) möglich.

Windows NT 5.0 Server und Workstation (2000) wird die Möglichkeit bieten über die GUI die Quantum-Length zu verändern.

Device-Treiber Tuning

Nur vier Device-Treiber werden bei den unterschiedlichen Windows NT Versionen anders vom Systemverhalten gehandhabt:

AFD.SYS	verwaltet die MS Winsock Connectivity
SRV.SYS	Der Server Prozess RDR
NWLKNKB.SYS	NWLINK IPX/SPX Implementierung von MS
NTFS.SYS	NTFS Dateisystem-Treiber

AFD.SYS setzt den Frame bei der Datenübertragung auf 4 KB bei der WS und auf 64 KB beim Server. Die Abhängigkeiten werden aus der Registry gelesen.

SRV.SYS legt die maximale Anzahl von Clients am System fest. SRV.SYS bei der WS ist auf 10 Verbindungen limitiert. SRV.SYS beim SRV auf eine unbegrenzte Anzahl Verbindungen.

NWLKNB.SYS hat einen internen Cache für Netzwerkrouen, welcher bei der WS und SRV unterschiedlich gehandhabt wird.

NTFS.SYS beim SRV setzt 11 Look-Aside Listen für Datei-Caching. Windows NT WS setzt im NTFS keine Look-Aside Listen.

Der letzte Unterschied zwischen SRV und WS ist das Programm Festplattenmanager (WINDISK). Auch hier wird der Produkt-Type aus der Registry gelesen und der unterschiedliche WINDISK gestartet. WINDISK für SRV beinhaltet noch RAID Level 1 und 5. RAID 0 und Datenträgersätze werden von WS und SRV unterstützt.

Detailunterscheidung Windows NT Workstation 4.0 - Windows NT Server 4.0

Eigenschaften und Dienste	Windows NT Workstation	Windows NT Server
Architektur	gleich	gleich
Subsysteme	gleich	gleich
Protokolle	gleich	gleich
Benutzerverwaltung	Lokale Gruppen	lokale und globale Gruppen
Oberfläche	gleich	gleich
PPTP	Nein	Ja
NTFS	gleich	gleich
Lizenz Manager	Nein	Ja
RAS Multilink	Nein	Ja
IIS	Nein	Ja
Netzwerkmonitor	Nein	Ja
Remote Boot	Nein	Ja
RIP für IP /IPX	Nein	Ja

WINS / DNS / DHCP Server	Nein	Ja
GSNW	Nein	Ja
Server Manager	Nur lokal	Ja (Domäne)
Verwaltungsassistent	Nein	Ja
Netzwerk Client Manager	Nein	Ja
Macintosh Services	Nein	Ja

Windows NT 4.0 Server ist intern für den Netzwerkeinsatz auf die optimalste Leistung und schnellstes Betriebsverhalten hin konzipiert worden.

Unterschiede der NT Workstation zum NT Server

Eigenschaften und Dienste	Windows NT Workstation	Windows NT Server
Anbindung anderer Clients	10	unbegrenzt
Verbindung zu anderen Netzwerkressourcen	unbegrenzt	unbegrenzt
Symmetric Multiprocessing	2 Prozessoren	bis zu 4 Prozessoren (durch OEM's bis zu 32 Prozessoren)
RAS	beschränkt auf eine Verbindung	bis zu 256 Verbindungen
Directory Replication	nur Import	Import und Export
Logon Validation	nein	ja
Macintosh Services	nein	ja
Raid 0 (Stripe Set)	ja	ja
Raid 1 (Disk Mirroring)	nein	ja
Raid 5 (Stripe Set mit Parität)	nein	ja
Netzwerkfunktionalität	Peer-to-Peer	Server