Windows XP: Architektur und Startvorgang

Christian Zahler

1. Versionen

Windows XP Professional

Professionelles Client-Betriebssystem für Unternehmen. Einsatz in Windows-Domänen, das sind Netzwerke, deren Sicherheitseinstellungen in einer Active Directory-Datenbank zentral verwaltet werden. Active Directory kann nur auf einem Windows Server-Produkt installiert werden (Windows 2000 Server, Windows Server 2003).

Windows XP Home Edition

Einsatz im SOHO-Bereich (*Small Office/Home Office*). Keine Domänenmitgliedschaft möglich; Einschränkungen bei der Sicherheit.

Windows XP Media Center Edition

Speziell angepasste Version für den Einsatz als Medienserver und Unterhaltungszentrale im Wohnzimmer. Diese Version ist als Kern einer AV- und HiFi-Anlage konzipiert.

Windows XP Tablet PC Edition

Speziell angepasste Version für Tablet PCs.

Windows XP 64-bit Edition

Einsatz auf leistungsfähigen PCs mit 64 bit-Architektur. (Diese Angabe bezieht sich auf die Datenbus-Breite.)

2. Architektur von Windows 2000/XP/2003

Quelle: `www.tecchannel.de`

Windows XP basiert komplett auf der Struktur von Windows 2000. Damit hat auch das monolithische Design von Windows 9x/Me endgültig ausgedient, denn XP ist modular aufgebaut. Jede Systemfunktion und jedes Subsystem wird von einem Modul oder einer kleinen Gruppe von Modulen bedient.

Die Vorteile dieser Struktur: Fehlerhafte Module lassen sich leicht austauschen und neue Funktionen leicht implementieren. Zentrale Funktionen wie GUI, Kommunikation und die Benutzerschnittstelle sind in Komponenten gefasst. So können Anwendungen und andere Module auf standardisierte Funktionen zurückgreifen – etwa um Eingaben von der Tastatur zu holen oder Daten auf dem Monitor auszugeben.

Alle hardwarespezifischen Funktionen sind im so genannten *Hardware Abstraction Layer* (HAL) zusammengefasst. Um Windows XP also an andere Plattformen anzupassen, muss lediglich für den HAL neuer Code geschrieben werden. Die restlichen Komponenten werden einfach neu kompiliert.

Wie die Vorgänger Windows NT und 2000 unterscheidet auch Windows XP zwischen dem so genannten *User-Mode* und dem *Kernel-Mode*. Module im *Kernel-Mode* haben beispielsweise direkten Zugriff auf die Hardware oder den Speicher. Das ermöglicht eine höhere Performance, hat aber auch deutliche Nachteile: Ein fehlerhafter Speicherzugriff kann zum Beispiel das ganze System zum Absturz bringen. Deshalb laufen die meisten Module nur im *User-Mode*. Diese Module sind komplett von der Hardware abgeschottet und können Systemfunktionen nur über die so genannten *Executive Services* ausführen, die entsprechende Programmierschnittstellen zur Verfügung stellen.

Executive Services

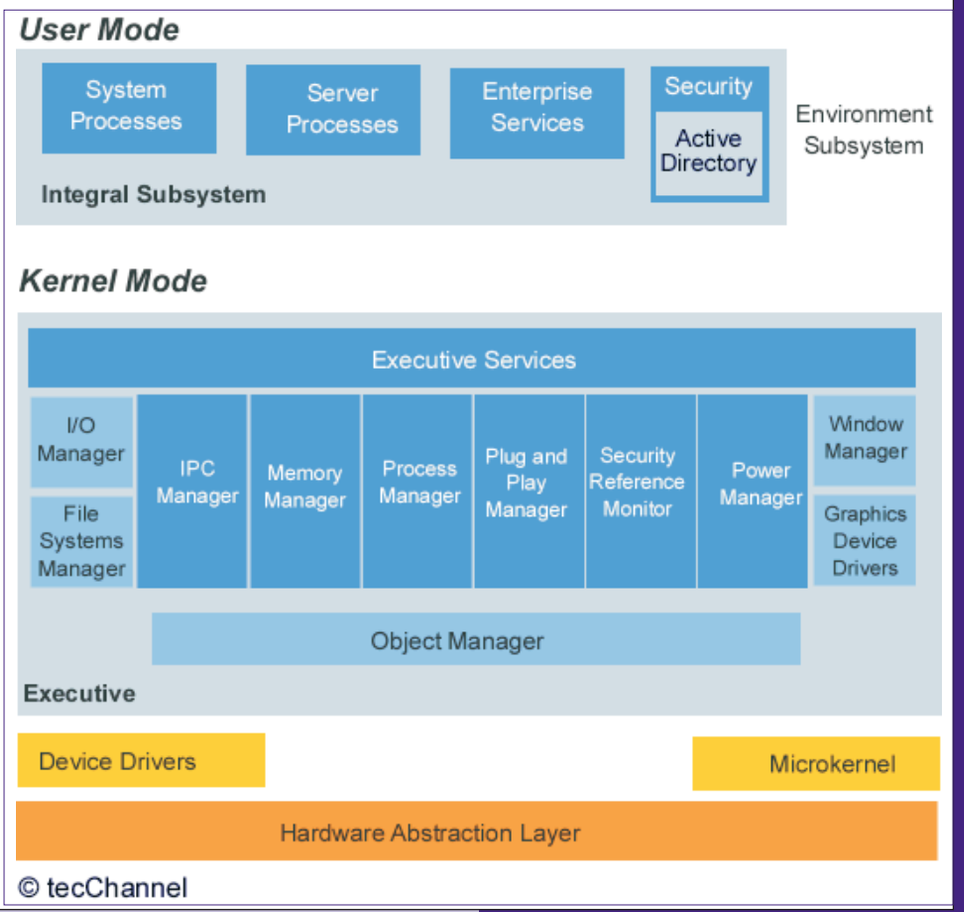
Die *Executive Services* von Windows XP sind eine Sammlung von Komponenten, die den Zugriff auf Hardware und Ressourcen verwalten. Dabei gibt es zwei verschiedene Arten von Funktionen: solche für Programme im User-Mode und interne, auf die nur die anderen Module in den *Executive Services* zugreifen können.

Die Hauptkomponenten der *Executive Services* sind

- *I/O Manager*: Ist zuständig für die Organisation von Ein- und Ausgabe auf verschiedene Geräte. Eine Unterfunktion des I/O-Managers ist der Filesystem-Manager, der Zugriffe auf Speichermedien wie Festplatten, Bandlaufwerke oder Netzwerk-Freigaben verwaltet.
- *IPC Manager*: Verarbeitet die gesamte Kommunikation zwischen verschiedenen Prozessen. Diese Kommunikation kann lokal über den LPC (*Lokal Procedure Call*) erfolgen oder mit Prozessen auf anderen Rechnern via RPC (*Remote Procedure Call*).
- *Memory Manager*: Für die wichtigste Ressource im Rechner, den Speicher, ist eine eigene Komponente verantwortlich. Der Speichermanager stellt jedem Prozess seinen eigenen virtuellen Adressraum zur Verfügung und sichert die verschiedenen Adressräume voneinander ab.
- *Process Manager*: Verwaltet und überwacht alle im System ablaufenden Prozesse.
- *Plug and Play Manager*: Ist für die Erkennung und Überwachung von installierten PnP-Geräten zuständig und handhabt die Installation von Treibern sowie das Starten notwendiger Dienste.
- *Security Reference Monitor*: Überwacht alle Sicherheitsmechanismen wie Authentifizierung, Zugriffe oder Besitzrechte.
- *Power Manager*: Zuständig für alle Funktionen des Power-Managements in Windows XP, wie Batterieüberwachung oder Stromsparfunktionen.
- *Window Manager*: Verwaltet die Benutzerschnittstelle wie etwa Dialogboxen, Fenster oder Benutzereingaben.
- *Graphics Device Drivers*: Sind zuständig für die eigentliche Ausgabe der Informationen auf dem Monitor.
- *Object Manager*: Alles in Windows XP wird als Objekt verwaltet. Dementsprechend ist der Object Manager eine zentrale Instanz von Windows XP.

Der Microkernel von Windows ist die zentrale Schaltstelle des Betriebssystems. Er verwaltet die Ausführung auf dem Prozessor und die Hardware-Interrupts. Zudem koordiniert er alle Aktivitäten der *Executive Services*.

Im User Mode laufen:



- **Systemprozesse** (etwa Sitzungs-Manager, WinLogon)
- **Dienste** (über DLLs = *Dynamic Link Libraries*; etwa Replikationsdienst, Nachrichtendienst, Ereignisanzeige)
- **Anwendungen** (über Subsystem-DLLs)

Die bis Windows 2000 integrierten Subsystem POSIX (für Unix-Anwendungen, die unter Windows laufen sollen) und OS/2 (für OS/2-Anwendungen, die unter Windows laufen sollen), sind in Windows XP und Windows Server 2003 nicht mehr enthalten. Es gibt jetzt nur mehr das Windows-eigene Win32-Subsystem und ein Subsystem für DOS/Win9x-kompatible Anwendungen.

Windows XP hat folgende Eigenschaften:

- Präemptives Multitasking
- Symmetrisches Multiprocessing (SMP)
- (eingeschränkte) Multiuser-Unterstützung

3. Startvorgang von Windows NT 4.0/2000/XP/2003

3.1 Für alle Betriebssysteme gleich

Nach dem Einschalten des Computers wird zunächst der POST *(Power-On Self Test)* durchgeführt. Dabei wird die Größe des physikalischen Arbeitsspeichers ermittelt und die Verfügbarkeit bestimmter Hardware-Komponenten wie etwa der Tastatur überprüft. Der POST ist ein Teil des BIOS *(Basic Input Output System)*, einer Softwaresammlung, die auf einem EEPROM-Chip am Motherboard enthalten ist.

Anschließend versucht das BIOS, ein Betriebssystem zu starten. Dazu untersucht es verschiedene Medien je nach der im BIOS-Setup definierte Boot-Reihenfolge:

- Diskettenlaufwerk
- bootfähige CD-ROM
- bootfähige Netzwerkkarte
- Masterdisk (= Festplatte, die den *Master Boot Record* [MBR] enthält)

Befindet sich im Diskettenlaufwerk eine nicht bootfähige Diskette, so erscheint eine Fehlermeldung.

Nehmen wir an, dass das zu startende Betriebssystem auf einer Festplattenpartition enthalten ist. Zunächst wird der *Master Boot Record* (MBR) geladen, aus dem das BIOS die Partitionstabelle liest. Der *Master-Boot-Record* (MBR) hat nur im Extremfall Einfluss auf den Windows XP-Start: Wenn ein altes System, etwa ein DOS-System oder ein älterer Boot-Manager, sich dort verewigt hat, scheitert unter Umständen der Start von einer größeren Festplatte, wenn die Windows XP-Partition jenseits der einschlägigen Grenzen liegt (8 oder 128 GByte).

Aus dieser Tabelle wird die aktive primäre Partition ermittelt und auf Grund dieser Information zum Boot-Sektor gesprungen.

3.2 Weiterer Startvorgang (gilt nur für Windows NT 4.0/2000/XP/2003)

Erst im Boot-Sektor steckt Windows-eigener Code. Er bringt den für den Start des Betriebssystems verantwortlichen Loader in den Speicher und führt ihn aus. Der Loader steckt in der Datei `„ntldr"`. Fehlt sie, kann Windows XP auf einem normalen x86-PC nicht starten.

Bei korrektem MBR startet normalerweise der NT-Loader (`ntldr`) und erledigt eine ganze Reihe von Aufgaben. Er schaltet das System in den *Protected Mode* und lädt einen enthaltenen minimalen Dateisystemtreiber für den Zugriff auf FAT- oder NTFS-Partitionen. Anschließend wertet er die `boot.ini`-Datei aus und findet so heraus, welche Systeme überhaupt installiert und gegebenenfalls in einer Auswahl anzuzeigen sind, etwa eine zweite Installation von Windows XP oder eine andere Windows-Version.

Ein Malheur mit den Boot-Sektoren lässt sich leicht beheben. Die Wiederherstellungskonsole, die Sie von der Original-Windows XP-CD starten können, ist das Mittel der Wahl: Sie kann den MBR (`fixmbr`) und den Boot-Sektor (`fixboot`) reparieren. Außerdem kann sie nach vorhandenen Windows-Installationen suchen (`bootcfg /rebuild`) und so eine verloren gegangene oder durch Umpartitionieren wertlos gewordene `boot.ini`-Datei restaurieren - der Befehl fragt für jede gefundene Windows-Installation nach, ob sie in die `boot.ini`-Datei aufzunehmen ist.

Lediglich die Boot-Dateien (`ntldr` und `ntdetect.com`) kann die Wiederherstellungskonsole nicht direkt restaurieren. Hier müssen Sie selbst ran und die Dateien aus dem i386-Verzeichnis der CD auf die Festplatte kopieren. Das ist zum Beispiel dann nötig, wenn Sie nach Windows XP noch 2000 installieren; die Installation ersetzt die Datei `ntldr` durch eine Version, die Windows XP noch nicht zu starten vermag.

Die Startdateien (`boot.ini`, `ntldr` und `ntdetect.com`) liegen stets auf der ersten primären Partition im Hauptverzeichnis, gemeinhin also Laufwerk `C:`. Sie sind durch Dateiattribute vor neugierigen Blicken geschützt. Welche tatsächlich verloren sind, bemerkt man also erst, wenn man bei einem Blick auf `C:` Befehle zur Anzeige verborgener Dateien verwendet, etwa `dir /ah`.

In seltenen Fällen spielt eine weitere Datei eine Rolle. Sie heißt `ntbootdd.sys` und enthält den Treiber, den das System braucht, um auf die Festplatte(n) ohne BIOS-Hilfe überhaupt zugreifen zu können. Ob das tatsächlich der Fall ist, verrät ein Blick in die `boot.ini`-Datei - beginnen die Zeilen im Abschnitt `[operating systems]` nicht mit `„multi(..."`, sondern mit `„scsi(..."`, so ist die Datei nötig; in der Regel handelt es sich um die `.sys`-Datei des jeweiligen Host-Adapters.

Wenn Sie über den NT-Loader weitere Betriebssysteme starten, etwa Windows 9x/ME oder Linux, so kommen weitere Dateien hinzu: Je Betriebssystem eine 512 Byte große Datei, die den Boot-Sektor des jeweiligen Systems beziehungsweise seiner Partition enthält. Taucht in der boot.ini-Datei ein Eintrag der Art `c:\="Windows 9x"` auf, so verweist er auf die Datei `bootsect.dos`; im Fall anderer Systeme ist auch der Dateiname im Eintrag enthalten, etwa `c:\lilo.bin="Linux"`.

Das Restaurieren der `bootsect.dos`-Datei ist kompliziert, weil es sich um den Boot-Sektor handelt, den der NT-Loader überschreibt. Im Fall anderer Betriebssysteme ist es in der Regel einfacher. Es genügt, wenn Sie mit einem Diskeditor oder einem vergleichbaren Programm erneut den Boot-Sektor der jeweiligen Partition in eine Datei sichern.

Generell ist es eine gute Idee, alle Startdateien zu sichern - Sie können das auf einer Diskette erledigen: Formatieren Sie unter Windows XP einfach eine Floppy und spielen Sie alle Dateien aus dem Wurzelverzeichnis von Laufwerk C: dort drauf, also `ntldr`, `ntdetect.com`, `boot.ini` und gegebenenfalls `ntbootdd.sys` sowie etwaige Dateien mit Boot-Sektoren - weil die Dateien versteckt sind, geht das am einfachsten über die Kommandozeile und den `xcopy`-Befehl.

Steckt die Diskette beim nächsten Booten im Laufwerk und taucht es in der Boot-Reihenfolge an früher Stelle auf, läuft der erste Teil des XP-Starts (und der etwaiger anderer Betriebssysteme) von der Floppy. Wenn Sie sich wunschgemäßer Funktion versichert haben, sollten Sie diese für Notfälle in den Schrank zu den Backups packen.

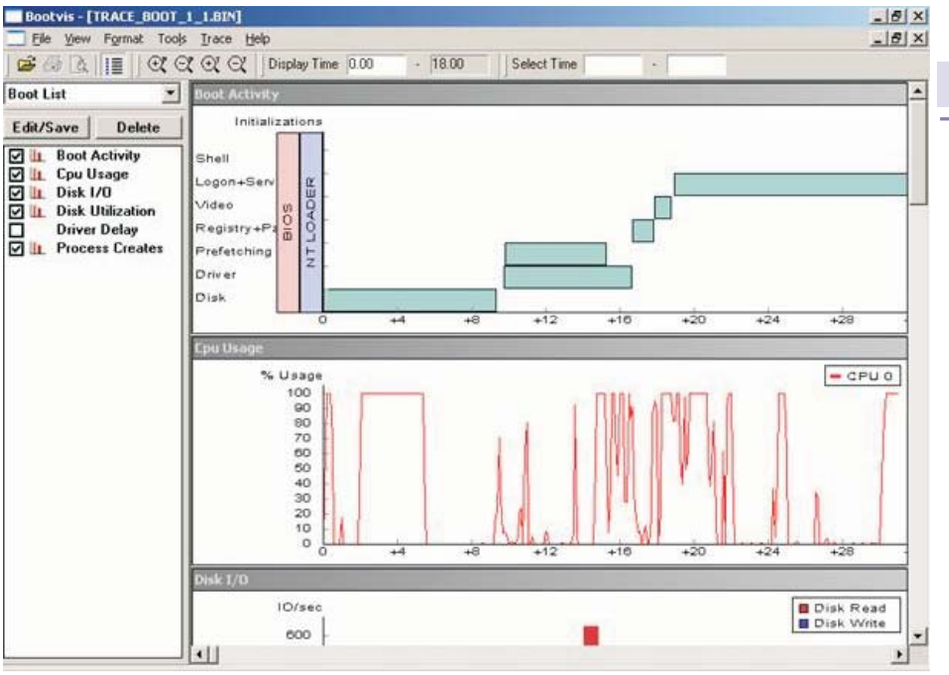
Ist nur ein einziges Windows-Betriebssystem installiert, bekommen Sie die Auswahl des NT-Loaders nicht zu sehen, sondern er beginnt sein Werk direkt: Er lädt eine noch eher rudimentäre Hardware-Erkennung (aus der bereits erwähnten Datei `ntdetect.com`) und startet sie. Deren Erkenntnisse, in der boot.ini vorgegebene Parameter sowie Daten, die der Loader aus der Registry liest, zum Beispiel zu etwaigen Hardwareprofilen, gibt er an `ntoskrnl.exe` weiter, also den eigentlichen Betriebssystemkern von XP, 2000 und NT.

Der Loader lädt nicht nur `ntoskrnl.exe`, sondern auch hal.dll und einige für den Systemstart unverzichtbare Treiber. Welche das sind, steht in der Registry unter `HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet`. Dieser Teil der Registrierung entspricht wohl am ehesten dem, was unter MS-DOS die `config.sys` und `autoexec.bat` oder Unix-Nutzer unter init-Skripten erledigt haben.

Jeder Eintrag dort im Ast Services beschreibt einen Treiber oder Dienst, den das System kennt. Je Eintrag beschreibt der Wert namens „Start", wann ein Treiber zu laden ist. In dieser frühen Startphase gilt das für alle, bei denen 0 eingetragen ist, sowie für Dateisystemtreiber, für die der Wert „Type" in der Registry auf 2 gesetzt ist.

Weitere Werte je Treiber oder Dienst regeln, für wie wichtig das System sie erachtet. Ist „ErrorControl" auf 1 gesetzt, dann gibt es im Fehlerfall eine Warnung, bei 0 macht das System stillschweigend weiter. Der Wert 2 zwingt das System zu einem Neustart mit der zuletzt erfolgreich gestarteten Konfiguration (last known good); schlägt der Ladeversuch auch hier fehl, macht das System weiter. Steht der Wert auf 3, hält das System im Fehlerfall an. Weitere Einträge regeln Abhängigkeiten der Treiber und Dienste untereinander.

Registry-Schäden wirken sich schon beim Booten aus: Gern trifft es die Datei namens „System" in `system32\config`, die den `HKEY_LOCAL_MACHINE`-Ast der Registry enthält und für den Systemstart wegen der dort abgelegten Daten über die Treiber und nachfolgend zu ladende Dinge von tragender Bedeutung ist. Beschädigt wird die Datei mitunter, wenn man einen PC unbedacht abschaltet oder Probleme mit Festplatte oder Hauptspeicher bestehen - solche Probleme können sich sogar durch das „Verschwinden" der Datei äußern.



Optimieren des Startvorgangs mit BootVis

Microsoft hat ein Tool entwickelt, welches den Bootvorgang noch weiter optimieren kann. Es heißt BootVis und ist im Internet kostenlos erhältlich. Dieses Tool analysiert den Bootvorgang und optimiert das Laden der Treiber und Systemdateien, so dass eine Einsparung der Zeit für den Startvorgang von Windows XP von bis zu 30 Prozent erreicht werden kann.

4. Aufbau der Datei BOOT.INI

Aussehen einer üblichen `BOOT.INI`-Datei:

```
[boot loader]
timeout=30
default=multi(0)disk(0)rdisk(0)partition(1)
  \WINDOWS
[operating systems]
multi(0)disk(0)rdisk(0)partition(1)\
  WINDOWS="Windows Server 2003, Enterprise"
  /fastdetect
c:\="Microsoft Windows 98"
```

Im Abschnitt `[boot loader]` finden sich zwei Angaben:

`timeout` gibt an, wie viele Sekunden das Startmenü angezeigt werden soll (`0` ... Startmenü wird nicht angezeigt; `-1` ... Startmenü wird so lange angezeigt, bis der Benutzer einen Eintrag ausgewählt hat)

`default` gibt an, von welcher Partition standardmäßig gestartet werden soll

Im Abschnitt `[operating systems]` sind alle vorhandenen Betriebssysteminstallationen aufgelistet, die über `boot.ini` gestartet werden sollen.

Die Partitionsangabe erfolgt für alle Windows NT-Betriebssysteme nach den so genannten *"Advanced RISC Computing"*-Spezifikationen (ARC) und kann drei verschiedene Formate haben:

```
multi(X)disk(Y)rdisk(Z)partition(W)\
scsi(X)disk(Y)rdisk(Z)partition(W)\
signature(X)disk(Y)rdisk(Z)partition(W)\
```

Für den Typ des Bootmediums steht das erste Wort des ARC-Pfades (`multi`, `scsi` oder `signature`).

Das Schlüsselwort `multi` gibt an, dass der Bootloader sich auf das BIOS des Rechners verlassen kann, um die Systemdateien zu laden. Windows verwendet dazu den Interrupt 13. Es kann sich dabei um IDE-, ESDI-, aber auch um SCSI-Platten handeln. Wichtig ist nur, dass die angeschlossene Platte vom Interrupt 13 angesprochen werden kann. Der Wert X steht dabei für die Nummer des Controllers. Y ist normalerweise 0, und Z steht für die Nummer des Laufwerks: Nummer 0 ist Master und 1 Slave am ersten IDE-Channel, 2 ist Master und 3 Slave am zweiten IDE-Channel.

Mit der Angabe `scsi` als erstes Schlüsselwort veranlassen Sie Windows, den Treiber `NTBOOTDD.SYS` aus dem Root-Verzeichnis des Bootmediums zu laden. Das muss übrigens nicht unbedingt ein SCSI-Gerät sein. Es kann sich dabei beispielsweise auch um einen speziellen IDE-Controller handeln, der nicht durch den Interrupt 13 des BIOS behandelt wird. X steht dabei für die Nummer des Adapters, wie sie von NTBOOTDD.SYS zurückgeliefert wird. Y ist die SCSI-ID des Laufwerks und Z die logical unit number (LUN). In den allermeisten Fällen ist die LUN 0.

Das Schlüsselwort `signature` wird nur eingesetzt, wenn

- das BIOS die INT13-Erweiterungen nicht unterstützt und die Systempartition größer als 7,8 GByte ist oder das Partitionsende nach dem Zylinder 1024 liegt.
- das BIOS des SCSI-Hostadapters ausgeschaltet ist, so dass die Systempartition nicht per INT-13-Aufruf angesprochen werden kann.
- bei signature die Signatur des Bootlaufwerks X ist, wie sie im MBR steht. Ntldr durchsucht dann alle Platten unabhängig vom Controller, an den sie angeschlossen sind.

Für alle drei Verfahren gleich ist die Partitionszählung: Windows nummeriert die Partitionen auf einer Festplatte nicht sequenziell, sondern nach Typ - erst die primären Partitionen und dann die logischen. Wenn Sie also beispielsweise zunächst eine primäre und zwei logische Partitionen haben, bekommt die primäre Partition die Nummer eins, die beiden anderen Nummer zwei und drei. Befindet sich nun auf der ersten logischen Partition eine Windows-Installation, sieht der Eintrag in der `boot.ini` so aus:

```
multi(0)disk(0)rdisk(0)partition(2)\
  WINNT="Windows" /fastdetect
```

multi(x)-Syntax

Die Parameter X, Y, Z und W haben folgende Bedeutung:

- X ist die Ordnungszahl des Adapters und sollte immer 0 sein (der Grund hierfür wird nachfolgend beschrieben).
- Y ist immer 0 (null), wenn der ARC-Pfad mit multi() beginnt, weil multi() den oben beschriebenen INT-13-Aufruf auslöst und daher die Parameterinformation disk() nicht benötigt.
- Z ist die Ordnungszahl für den Datenträger auf dem Adapter und ist gewöhnlich eine Zahl zwischen 0 und 3.
- W ist die Partitionsnummer. Alle Partitionen außer Typ 5 (erweiterte MS-DOS-Partition) und Typ 0 (ungenutzt) erhalten eine Nummer, wobei primäre Partitionen vor logischen Laufwerken aufgezählt werden. Hinweis: Die erste gültige Zahl für W ist 1, während X, Y und Z mit 0 (null) beginnen.

scsi(x)-Syntax

- Die Parameter X, Y, Z und W haben bei Verwendung der SCSI()-Syntax folgende Bedeutung:
- X ist die Ordnungszahl des vom Treiber "NTBOOTDD.SYS" identifizierten Adapters.
- Y ist die SCSI-Kennung des Zieldatenträgers.
- Z ist die SCSI-LUN (Logical Unit Number) des Zieldatenträgers. Diese Nummer ist fast immer 0 (null).
- W ist die Partitionsnummer. Alle Partitionen außer Typ 5 (erweiterte MS-DOS-Partition) und Typ 0 (ungenutzt) erhalten eine Nummer, wobei primäre Partitionen vor logischen Laufwerken aufgezählt werden. Hinweis: Die erste gültige Zahl für W ist 1, während X, Y und Z mit 0 (null) beginnen.

Bei Verwendung der SCSI()-Syntax ist der Wert von X von `"NTBOOTDD.SYS"` abhängig. Jeder SCSI-Treiber hat seine eigene Methode zum Bestimmen der Controller-Reihenfolge, obwohl im allgemeinen die Reihenfolge eingehalten wird, in der das BIOS auf den Controllern geladen wird (vorausgesetzt, das BIOS wird geladen).

Optionen ab Windows NT 4.0

/basevideo

Mit dem Basevideo-Switch wird erzwungen, das das System mit einer Standard-VGA Auflösung bootet - also mit 640x480 und 16 Farben. Diese Auflösung kann mit praktisch jeder Graphikkarte verwendet werden. Hat man zum Beispiel eine falsche Auflösung eingestellt und das System startet nicht mehr richtig - oder man kann nichts mehr sehen - dann kann man diesen Schalter verwenden um wieder ein Bild zu bekommen. Damit kann man dann den fehlerhaften Treiber deinstallieren und zu einer vorherigen Konfiguration zurückkehren.

/baudrate=Zahl

Mit diesem Schalter kan man die Baud-Rate für den Debug-Port einstellen. Beim Debuggen über ein Modem verwendet man z.b

meist 9600 Baud, bei einem Null-Modem Kabel verwendet man 115.200.

/crashdebug

Mit diesem Schalter wird der Kernel-Debugger direkt beim Start des Betriebssystems geladen. Ist der Schalter gesetzt und tritt ein Kernel Fehler auf, dann wird das Remote-Debugging aktiviert.

/debug

Der Schalter tut das gleiche wie `/crashdebug` - allerdings wird der COM-Port der fürs Debuggen verwendet wird sofort als Debug-Port aktiviert. Beim Crashdebug-Schalter passiert das erst, wenn ein Kernel-Fehler eingetreten ist.

/debugport=comXX

Mit diesem Schalter wird festgelegt welche Port als Debug-Port verwendet werden soll. Als Parameter können die gültigen Com-Ports angegeben werden, also zum Beispiel `com1`, `com2`, etc.

/maxmem=Zahl

Mit diesem Parameter kann man die Speichermenge festlegen die Windows zur Verfügung stehen soll. Damit kann man zum Beispiel herausfinden ob bestimmte RAM Module defekt sind. Hat man beispielsweise 128 MB RAM im System kann man mit `/maxmem` die zu verwendende Menge auf 64MB beschränken.

/nodebug

Mit diesem Schalter wird der Debug-Support ausgeschaltet.

/numproc=Zahl

Mit diesem Schalter kann man die Anzahl an CPUs einstellen, die Windows verwenden soll. Auf einem Multi-CPU System kann man damit einzelne CPUs deaktivieren.

/pcilock

Ist der Schalter gesetzt, so weist Windows I/O Ports und IRQs nicht dynamisch an PCI Geräte zu. Statt dessen werden die Einstellungen aus dem BIOS verwendet.

/sos

Ist der Schalter gesetzt, dann werden die Namen der Gerätetreiber angezeigt, während die Treiber geladen werden.

/HAL=DateiName

Mit diesem Schalter kann man festlegen welche DLL mit der Hardware-Abstraktionsschicht geladen werden soll. Im Normalfall ist das die `HAL.DLL`.

/kernel=DateiName

Mit diesem Schalter kann man festlegen, welcher Kernel beim laden verwendet werden soll. Man kann damit zum Beispiel zwischen dem Multi-CPU und dem normalen Kernel wechseln, oder man kann auch eine Debug-Variante des Kernels aktivieren.

/burnmemory=Zahl

Mit diesem Schalter kann man einen Wert in MB angeben. Diese Menge an Speicher wird Windows im Betrieb dann vorenthalten.

/3GB

Mit diesem Schalter wird die Aufteilung des virtuellen Speichers verändert. Normalerweise teilt Windows den virtuellen Adressraum so auf, das das Betriebssystem 2GB und alle Anwendungsprogramme ebenfalls 2GB erhalten. Mit diesem Schalter wird die Aufteilung so verändert das Windows selbst nur noch 1 GB virtuellen Adressraum, Anwendungsprogramme aber 3 GB Adressraum haben.

Optionen ab Windows 2000

/bootlog

Mit diesem Schalter schaltet man das Boot-Logging ein. Die Log-Datei befindet sich im System-Root und hat den Namen `Ntbtlog.txt`.

/channel=Zahl

Mit diesem Schalter - zusammen mit `/debug` - werden die Debug-Informationen nicht seriell sondern über einen IEEE 1394 Port versendet.

/cmdcons

Bootet in die Wiederherstellungskonsole

/fastdetect:comXX

Mit diesem Schalter sucht Windows nicht nach seriellen Mäusen an den angegebenen COM-Ports. Multiple COM-Ports können durch Komma getrennt angegeben werden. Wird kein COM-Port angegeben, sucht Windows an keiner Schnittstelle nach seriellen Mäusen.

/noguiboot

Mit diesem Schalter schaltet man die Anzeige des Bitmaps beim Bootvorgang aus.

/pae

Der Windows Kernel mit *Physical Address Extension* (PAE) wird geladen (`ntkrlnpa.exe`), mit dem 64 bit-RAM-Adressen unterstützt werden.

/nopae

Der Windows Kernel ohne *Physical Address Extension* (PAE) wird geladen, selbst wenn das System PAE unterstützen würde und mehr als 4 GB physikalisches RAM installiert sind.

/nolowmem

Voraussetzung: Schalter `/pae`; die ersten 4 GB physikalisches RAM werden nicht benutzt (etwa zum Testen von Treibern bei großen Arbeitsspeichern)

/safeboot:Parameter

Mit diesem Schalter startet man im Safe Mode. Der Parameter legt dabei die Art und Weise fest. Folgende Parameter sind möglich:

- `minimal`
- `network`
- `minimal(alternateshell)`
- `dsrepair` (Verzeichnisdienstwiederherstellung für Domänencontroller)

/year=2001

Ignoriert die Jahresangabe der Systemzeit und verwendet stattdessen das angegebene Jahr.

Optionen ab Windows XP/2003

/bootlogo

Diesen Schalter kann man benutzen, um einen benutzerdefinierten Startbildschirm anzuzeigen. Dazu erstellen Sie ein 640 x 480 px-Bitmap mit 16 Farben und speichern Sie diese Grafik unter dem Namen boot.bmp ins Windows-Verzeichnis. Mit den Parametern "`/bootlogo /noguiboot`" wird dann der benutzerdefinierte Startbildschirm angezeigt.

/execute

Deaktiviert *Data Execution Prevention* (DEP); siehe `/noexecute`

/minint

Wird von Windows PE verwendet. Lädt den System-Ast der Registry so in den Arbeitsspeicher, dass beim Herunterfahren die Änderungen nicht in die Registry gespeichert werden.

/noexecute

Eingeführt mit Windows XP Service Pack 2. Diese Option aktiviert auf 32-bit-Versionen von Windows die DEP (*Data Execution Protection - DEP*), die bewirkt, dass der Memory Manager bestimmte Seiten als Daten markiert. Daher kann bösartiger Code, der in diesen Speicherseite abgelegt wird, nicht mehr ausgeführt werden. Viele Viren, die Buffer-Overfloaw-Fehler ausnutzen, haben sich dieser Technologie bedient. In 64-bit Versionen von Windows ist diese Option immer aktiviert.

`/NOEXECUTE=OPTIN`: DEP wird standardmäßig aktiviert, es sei denn, eine Installation wird manuell ausgenommen

`/NOEXECUTE=OPTOUT`: DEP wird standardmäßig deaktiviert, es sei denn, eine Installation wird manuell eingeschlossen

`/NOEXECUTE=ALWAYSON`: Aktiviert DEP für alle Partitionen

`/NOEXECUTE=ALWAYSOFF`: Deaktiviert DEP für alle Partitionen

/redirect

Wurde mit Windows XP eingeführt. Damit werden die *"Emergency Management Services"* (EMS) aktiviert, mit denen es möglich ist, Kommandos über eine serielle Schnittstelle abzusetzen, selbst wenn die Maschine selbst nicht mehr gestartet werden kann. Die serielle Schnittstelle und Baud-Rate muss in der `boot.ini`-Datei angegeben werden.

Eine vollständige Beschreibung aller Parameter findet sich auf

http://www.sysinternals.com/ntw2k/info/bootini.shtml.

5. Windows NT 4.0/2000/XP/2003-Bootdiskette

Um die Bootdiskette zu erstellen, sind nur wenige Schritte notwendig:

- Formatieren Sie die Diskette unter Windows NT/2000/XP. Damit ist auf der Diskette automatisch der richtige Bootsektor.
- Kopieren Sie die Dateien `Ntldr`, `Ntdetect.com` und `boot.ini` auf die Diskette.
- Wenn Sie SCSI-Treiber benötigen, kopieren Sie den Treiber (etwa `AIC78XX.SYS` für den Adaptec 2940) unter dem Namen `ntbootdd.sys` auf die Diskette.