

# Konzepte von Betriebssystem-Komponenten

## Windows 2000/XP (3.1,NT,95,98)

Michael Lehmeier  
m\_lehmeier@gmx.de

### 1 Die Geschichte der Windows-Familie

#### 1.1 MS-DOS

Die Entwicklung von preisgünstigen Computern Ende der 70'er, die für jedermann erschwinglich wurden, führte zu einer weitgehenden Abkehr von großen, multi-User-Architekturen hin zu kleinen Desktops. MS-DOS war die Erweiterung des kleinen CP/M-Betriebssystems, das für einen 8-Bit System Rechner konzipiert war. MS-DOS (8KB resident im Arbeitsspeicher) sollte auf dem 8088 basierten IBM-PC laufen, der 1981 eingeführt wurde. Zwei Jahre später wurde MS-DOS richtig brauchbar mit der Version 2.0.

MS-DOS konnte sich auf der IBM-Architektur schnell durchsetzen, da viele Programme des davor weitverbreiteten CP/M-Systems weiterhin benutzt werden konnten. Dies konnte aber nicht ohne Einschränkungen realisiert werden, die noch Jahre später Benutzern Kopfzerbrechen bereiten würden, wie 1 Megabyte maximal adressierbarem Speicher.

Von dem, was man heute unter einem modernen Betriebssystem versteht, war MS-DOS weit entfernt. Es war weder multitaskingfähig, noch enthielt es Mechanismen zur Systemsicherheit, noch konnten mehrere Benutzer verwaltet werden. Benutzerprogramme hatten vollständige Kontrolle über das System.

#### 1.2 Windows 1.0 - 2.0

Inspiziert von der Bedienoberfläche des Macintosh-Vorgängers von Apple schuf Microsoft eine graphische Bedienoberfläche für DOS. Sie war vollkommen von MS-DOS abhängig und auch noch nicht multitaskingfähig. Allerdings konnte man mehrere Programme starten und benutzen, ohne die anderen vorher zu beenden, was in der Command-Shell von MS-DOS unmöglich war.

Dennoch waren die frühen Windows-Versionen ein kommerzieller Fehlschlag.

#### 1.3 Windows 3.1

Der große Durchbruch gelang Microsoft mit Windows 3.0 und dessen Nachfolgern Windows 3.1 und Windows für Workgroups. Obwohl auch hier noch DOS die Kontrolle über Hardware und Dateisystem behält, führte die umfangreichen und komfortablen Bibliotheken der Bedienoberfläche von Windows 3.1 zu einem schnellen Anwachsen der Anzahl von Programmen, die unter Windows zur Verfügung standen. Es gab auch erstmals eine 32-Bit API, auch wenn

diese kaum benutzt wurde. Sie war auch nur mit einer Zusatzbibliothek lauffähig, die die Win32-Aufrufe auf die Win16 API abbildeten.

## 1.4 Windows 95 und Nachfolger

Der zweite große Sprung der Windows-Familie brachte große Verbesserungen im Speichermanagement und dem Dateisystem mit sich. Während anfangs noch das alte DOS-Dateisystem FAT16 verwendet wurde, wurde es bald durch das verbesserte FAT32 ersetzt.

Microsoft begann allmählich, Teile des Systems von DOS in den Windowsteil zu verlagern. Dennoch blieb Windows noch ein DOS-Aufsatz, da Abwärtskompatibilität noch ein Entwicklungsziel war. Obwohl der Anteil an 32-Bit Code immer weiter ansteigt, ist echtes Multitasking immer noch nur beschränkt möglich, da das System weiterhin offenliegt und vor Inkonsistenzen geschützt werden muß.

## 1.5 Windows NT

Mit Windows NT schuf Microsoft erstmals ein reines 32-Bit Betriebssystem. Da es fast nur noch in C geschrieben ist, und nicht wie zuvor in Assembler, war es in den frühen Versionen noch portabel. Mit Multiprozessorunterstützung, dem modernen Dateisystem NTFS und einem guten Sicherheitskonzept war es erstmals vergleichbar mit anderen Betriebssystemen wie UNIX.

Nicht zuletzt wegen des hohen Speicherbedarfs, der mangelnden 32-Bit Software und schlechter Hardwareunterstützung konnte sich NT zunächst nicht durchsetzen. Erst der Nachfolger NT 4.0 fand vor allem in Firmen eine weite Verbreitung, unter anderem wegen einer vergleichbaren Bedienung wie bei älteren Windows-Versionen

## 1.6 Windows 2000

Windows 2000 ist die Fortsetzung von Windows NT. Mit ihr wurde durch bessere Hardwareunterstützung und vergleichbarer Bedienung wie bei älteren Windows-Versionen die endgültige Abkehr von der 16-Bit Architektur durchgeführt.

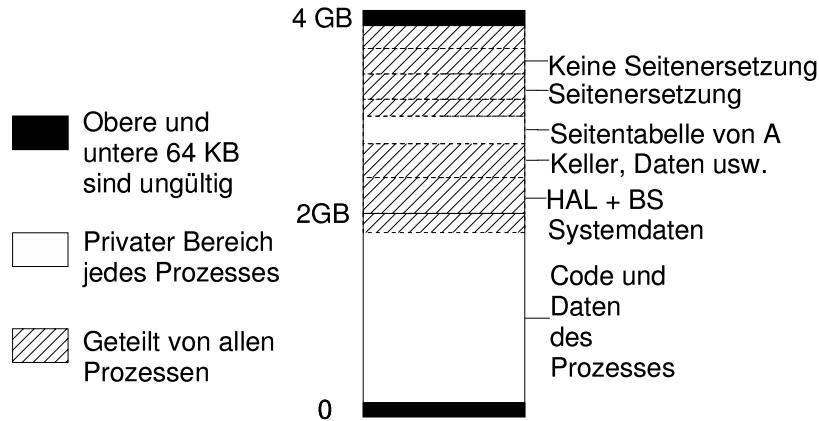
Außerdem gab es noch zahlreiche weitere Verbesserungen im Dateisystem, der Internationalisierung und anderen Komponenten. Allerdings gingen diese Fortschritte auf Kosten der Portabilität. Auch das Geschäftsmodell, unterschiedliche Versionen des selben Programms auf den Markt zu bringen, fand nicht überall Zustimmung.

MS-DOS ist nur noch als Emulation enthalten.

## 2 Der Adressraum von Windows NT / 2000

Der virtuelle Adressraum eines Windows 2000-Prozesses ist 4 Gigabyte (GB) groß. Dieser ist unterteilt in zwei Blöcke zu jeweils 2 GB, von denen 2 GB für Code und Daten des Prozesses vorgesehen sind. Davon sind die untersten 64 KB ungenutzt. Ein Versuch auf sie zuzugreifen würde in einer Fehlermeldung enden (es wird als NULL-Pointer interpretiert). Die obersten 256 MB enthalten z.B. Systemzähler und Zeitgeber und werden mit allen anderen Prozessen geteilt. Der Rest des unteren 2 GB Adressraums steht dem Prozess für Code und Daten voll zur Verfügung.

Oberhalb von 2 GB liegen Bereiche, die geschützt sind und überwiegend von allen Prozessen gemeinsam benutzt werden, sofern sie sich im Kernelmodus befinden. Im Benutzermodus kann auf diese Bereiche nicht schreibend und teilweise auch nicht lesend zugegriffen werden.



Nach: Tennenbaum, Seite 866

Abbildung 1: Virtueller Adressraum eines Prozesses unter Windows 2000

In manchen Windows 2000-Versionen ist es möglich, den Benutzeradressraum auf 3 GB zu erweitern. Sollten auch 4 GB Arbeitsspeicher für einen Prozess nicht ausreichen, kann man auf Bank-Switching ausweichen, einer Möglichkeit, die schon unter DOS-Zeiten verwendet wurde, um die 1 MB Grenze zu sprengen.

### 3 Die Betriebssystem-Architektur

In der ersten Version von Windows NT wurde aus Gründen der Sicherheit und der Portabilität das System in einen kleinen Kernel (lief im Kernelmodus) und verschiedenen Serverprozessen (Benutzermodus) gespalten. Diese Trennung war jedoch mit Performanceproblemen behaftet, weshalb sie wieder aufgegeben wurde. Die Ebenenstruktur existiert jedoch weiterhin, auch wenn wesentlich größere Teile im Kernmodus arbeiten.

#### 3.1 Die HAL

Die sogenannte Hardware Abstraction Layer HAL war der erste Versuch, den Zugriff auf Hardware unabhängig von der dem Aufbau derselben zu machen. In ihr wurden Systemaufrufe abstrahiert, und von denen man im allgemeinen erwartet, daß sie auf jedem modernen Rechner realisiert werden können.

Allerdings hat diese Verallgemeinerung Grenzen: Unterschiede im Bau eines Motherboards können kaschiert werden, nicht jedoch die von komplett anderen Architekturen. Das BIOS ist z.B. über sie ansprechbar, Tastatur und Maus jedoch nicht.

Da die HAL in ihrem Umfang sehr beschränkt ist, wurden Maßnahmen getroffen, wie sie bei Bedarf erweitert werden kann. Das Ergebnis ist DirectX.

#### 3.2 Der Kernel

Der Kernel stellt eigentlich nur einen eine zusätzliche Abstraktionsebene zur HAL dar. Wichtiger ist die seit NT 4.0 ebenfalls im Kernmodus laufende sog. Executive. Sie ist im Gegensatz zum Kernel und HAL vollkommen hardwareunabhängig. In ihr liegt unter anderem der

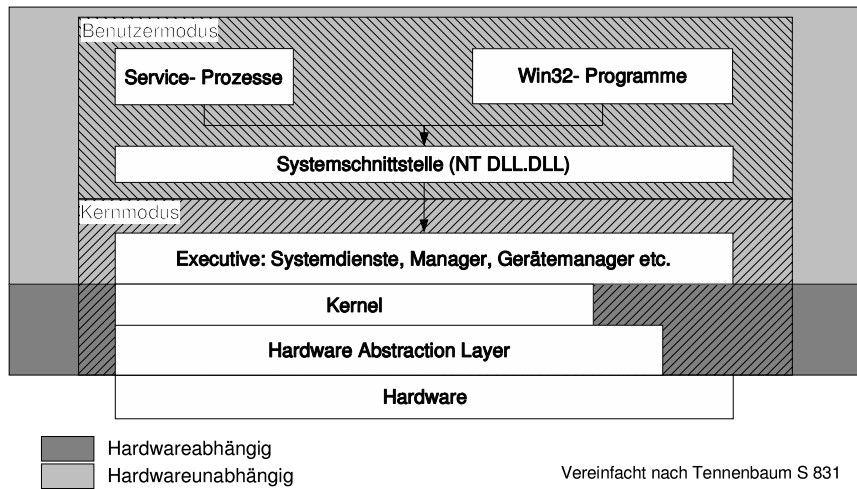


Abbildung 2: Die Windows 2000 Layer

Prozess-Manager, die Speicherverwaltung und der Plug-and-Play-Manager. Die Executive ist also nach außen das, was man langläufig als Betriebssystem bezeichnet. Programmierer greifen auf sie über die Systemschnittstelle auf die Systemdienste zu, die ihrerseits die Win32-API auf die anderen Bereiche der Executive verteilen.

### 3.3 Prozessmanagement

Die Prozesse unter Windows 2000 funktionieren ähnlich wie unter UNIX. Jeder Prozess startet mit einem Thread, dem noch weitere beigefügt werden können. Sie laufen überwiegend im Benutzermodus mit allen Beschränkungen, die für die Sicherheit des Systems wichtig sind. Eine Eigenart von Windows 2000 sind jedoch die sog. Fibers. Sie wurden wegen der ineffizienten Weise eingefügt, mit der in Windows 2000 zwischen Threads gewechselt wird. Es sind gewissermaßen leichtgewichtige Threads, die verwendet werden, um das Problem zu umgehen.

## 4 Einige Besonderheiten von Windows

### 4.1 Die Registry

Eine Besonderheit und Neuerung unter den späteren Windows-Versionen (nach 95) ist die Zentralisierung von Systeminformationen in der sogenannten Registry. Anstatt diese Informationen in dutzenden (wenn nicht hunderten) kleinen .ini-Dateien auf der Festplatte zu speichern, kann man ab Windows 95 auf Systeminformationen über den Registry-Mechanismus zugreifen, der jedoch teilweise schwierig zu handhaben und in früheren Versionen auch noch fehlerhaft war. Das Prinzip ist, daß die Win32-API es dem Programmierer ermöglicht, über Schlüsselwörter, sogenannten Keys, auf eine Art virtuellem Dateisystem zuzugreifen, in dem Daten über Software, Hardware, Systemeinstellungen und vielem mehr abgelegt sind.

## 4.2 Plug-and-Play

Plug-and-Play ist ein Mechanismus, den Microsoft entwickelt hat, um den Benutzern es einfacher zu machen, neue Hardware zu installieren oder neue Geräte zur Laufzeit einzubinden. Dafür ist zunächst eine automatische Geräteerkennung seitens der Hardware notwendig, sowie ein Treiber, der die PnP-Spezifikationen erfüllt. Der PnP-Manager übernimmt dann die Ressourcenverteilung, regelt die Energiesparmodi und ähnliches.

# 5 Die Dateisysteme

## 5.1 Die FAT-Familie

### 5.1.1 Beginn und Eigenschaften

Die FATs wurden geschaffen, um einen einigermaßen kompatiblen Nachfolger des CP/M-Dateisystems zu bekommen. Folglich übernahm man auch die alte Beschränkung der kurzen Dateinamen, die auf 8 + 3 Zeichen basierten. Sie neigt zur Fragmentierung und es können bei Systemabstürzen leicht Inkonsistenzen auftreten.

Die maximale Größe der Dateisysteme hängt von der Größe der Cluster und deren höchsten adressierbaren Anzahl ab. Bei FAT12 waren es anfänglich maximal 2 MB, die später durch höhere Blockgrößen auf 16 MB erweitert wurden. Deshalb wird sie auch heute noch vor allem als Dateisystem für kleine Wechselmedien wie Disketten verwendet.

FAT16 mit seiner maximalen Partitionsgröße von 2 GB blieb bis zur ersten Version von Windows 95. Danach wurde auf FAT32 gewechselt, bei dem über 28 Bit bis zu 2 Terabyte ansprechbar sind mit einer maximalen Dateigröße von 4 GB. Ein weiterer Vorteil der größeren Anzahl an adressierbaren Cluster ist die Tatsache, daß bei kleineren Partitionen die Clustergröße reduziert und somit der Festplattenspeicher effizienter genutzt werden konnte. Mit ihr wurden noch weitere Verbesserungen wie die Kopie des Startsektors und ein Stammverzeichnis ohne Größenbeschränkungen eingeführt.

### 5.1.2 Lange Dateinamen unter FAT

Zur Realisierung von langen Dateinamen unter FAT wurden die ungenutzten Bytes und Bits des ursprünglichen FAT-Dateisystems verwendet. Sollte also ein langer Dateiname vorliegen, wurde vor dem Verzeichniseintrag zusätzlich noch weitere vorangestellt, die diesen Namen vervollständigten. Um herauszufinden, wann das stattfand, wurde ein neues Attribut-Bit verwendet.

## 5.2 NTFS

NTFS ist das Dateisystem für Windows NT und dessen Nachfolger. Mit 64 Bit für die Zuweisung der Cluster können bis zu 128 Terabyte adressiert werden. Es ist ein Journaling-Dateisystem, das für erhöhte Datenkonsistenz sorgt. So werden z.B. unterbrochene Transaktionen wieder rückgängig gemacht. Außerdem ist die MFT, in der alle Dateinamen stehen, noch einmal in der Mitte der Partition gespiegelt, um Redundanz zu schaffen.

Darüber hinaus wurden Erweiterung wie POSIX-Unterstützung, Symbolische Links und Verschlüsselung implementiert. Entgegen der langläufigen Meinung fragmentieren NTFS-Dateisysteme aber sehr wohl noch. Allerdings steht für Defragmentierungsprogramme eine API zur Verfügung.

## Literatur

- [1] Tannenbaum, A. (2002): "Moderne Betriebssysteme", 2. überarbeitete Auflage, 1021 Seiten, Pearson Studium, München
- [2] Solomon, D.; Russinovich, M. (2000): "Inside Microsoft Windows 2000", 3. Auflage, 768 Seiten, Microsoft Press Deutschland, Unterschleißheim
- [3] Switzer, R. (1993): "Operating Systems - A practical approach", 463 Seiten, Prentice Hall, Hemel Hempstead