

# Treibersoftware enträtselt

***Gerade bei modernen Betriebssystemen wie Windows 95 und Windows NT läuft ohne passenden Treiber überhaupt nichts. Ein Blick hinter die Treiber-Kulissen zeigt ihre Funktion als Mittler zwischen Hard- und Software.***

**S**ie begegnen dem stolzen PC-Besitzer meist schon in den ersten Stunden seiner Computer-Erfahrung, wirken im verborgenen und erledigen ihren Teil an der Computerei nebulös und geheimnisvoll. Treiber, das sind Dinge, die man braucht, ohne zu wissen warum. Wer einmal das Prinzip durchschaut hat, geht mit Treibern anders um und schafft es dann auch, manche Probleme bei der PC-Konfiguration selbst zu lösen.

Allgemein gesprochen ist ein Treiber eine Software-Komponente, die als Schnittstelle zwischen Anwendungen, Betriebssystem und Hardware fungiert. In der Frühzeit des PC waren Treiber noch nicht sehr verbreitet, so daß es oft mehrere Versionen eines Programms für die verschiedenen Grafikkarten oder Drucker gab. Das war schon damals ein großer Nachteil, denn unter Umständen

mußten Firmen deshalb ein Dutzend diverser Programmversionen pflegen. Viel sinnvoller ist, die individuellen Routinen für eine spezielle Hardware-Komponente in ein separates Programm zu packen. Dadurch ist nur noch eine Programmversion notwendig, die auf die jeweilige Hardware über vorher vereinbarte Routinen des Treibers zugreift.

Doch auch das war noch nicht die ideale Lösung: Denn jeder Software-Hersteller mußte sein Anwendungsprogramm (zumeist unter DOS) mit den notwendigen Treibern für alle bekannten Geräte ausstatten, was mehr oder weniger gut gelang. Sobald ein neuer Drucker oder eine andere Grafikkarte auf den Markt kam, mußte ein neuer Treiber geschrieben werden. Dies führte zu einer Polarisierung auf die Marktführer in Hard- und Software.

Erst Microsoft Windows hat dieses Problem für die meisten Anwender dauerhaft gelöst. Seit Windows benötigt man nur noch einen einzigen Druckertreiber, den das Betriebssystem verwaltet. Arbeitet der Drucker damit in einem Windows-Programm, dann funktioniert er auch mit allen anderen Anwendungen.

## Es muß der richtige Treiber sein

Was viele übersehen: Gute, fehlerfreie und schnelle Treibersoftware ist heute genauso wichtig wie gute Hardware. Die schnellste Grafikkarte ist wertlos, wenn es für sie keine stabilen Treiber gibt. Ein SCSI-Controller, für den kein Windows-95-Treiber erhältlich ist, ist für Windows-95-Programme nicht erreichbar.

Schlechte Treiber können Ihr gesamtes System abbremsen oder destabilisieren. Und es gibt mehr davon, als Sie glauben. Gerade manche Noname-Produkte kommen mit Treibern auf den Markt, die das Beta-Stadium nie verlassen haben.

Und denken Sie nicht, jeder Hersteller würde irgendwann passende Treiber produzieren und nachliefern. Einen Treiber für Windows 95, Windows NT oder OS/2 zu entwickeln ist erheblich komplizierter als ein einfacher MS-DOS- oder Windows-Treiber für die Versionen 3.x. Unternehmen in den USA kassieren für das Entwickeln eines Windows-NT-Treibers 100 000 US-Dollar und mehr. Viele Hardware-Hersteller stellen daher bei alten Produkten lieber die Fertigung ein, als diese Summen für ein nicht mehr gängiges Produkt zu investieren.



## Basics

### Wichtige Begriffe zur Kategorie Treiber

**API:** Abkürzung für „Application Programming Interface“; eine einheitliche Software-Schnittstelle, die Zugriffe auf bestimmte Hardware-Komponenten (Laufwerke etc.) oder Betriebssystemfunktionen unabhängig von der verwendeten Hardware ausführt.

**DMA:** Abkürzung für „Direct Memory Access“; ein Verfahren, große Datenmengen ohne Zutun des Prozessors zwischen Peripheriegerät und Arbeitsspeicher zu übertragen.

**IRQ:** Abkürzung für „Interrupt Request“. Über eine Signalleitung, den Interrupt Request Channel auf dem ISA- oder PCI-Bus, erregt eine Steckkarte die Aufmerksamkeit des Prozessors. Wird ein solcher Channel aktiviert, dann verzweigt der Prozessor zu einer speziellen Behandlungsroutine für dieses Signal.

**Kernel:** der zentrale Teil eines Betriebssystems.

**Multitasking-Betriebssystem:** ein Betriebssystem wie Windows 95, NT oder OS/2, bei dem mehrere Prozesse zur gleichen Zeit ablaufen.

### So arbeiten Treiber mit der Hardware zusammen

Es gibt verschiedene Hardware-Zugriffe, die ein Treiber ausführen kann. Der erste und einfachste Zugriff sind direkte Schreib- oder Leseoperationen auf Input/Output-(I/O-)Ports. I/O-Ports sind Adressen, über die der Prozessor direkt mit der Hardware kommunizieren kann. So verwenden viele ältere Adaptec-SCSI-Controller in der Regel Port 330 zur Kommunikation mit der Hardware.

Auf die gleiche Weise kann ein Treiber über den Arbeitsspeicher oder andere Speicherbereiche mit der Hardware kommunizieren. Der Treiber einer Grafikkarte greift zum Beispiel über ein Spei-

cherfenster auf die Karte zu. Um einen Bildpunkt zu setzen oder zu löschen, muß er je nach Auflösung und Farbtiefe einige Bits oder sogar Bytes im Speicher auf der Grafikkarte ändern.

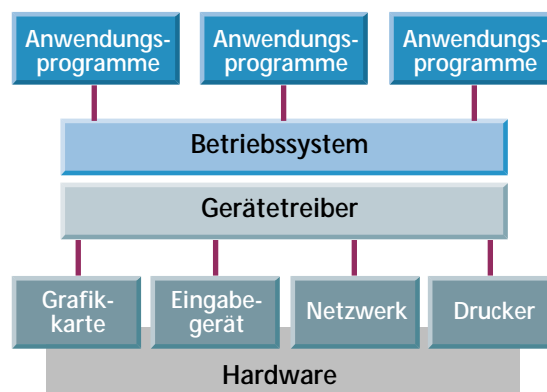
Wenn eine Hardware-Komponente die Aufmerksamkeit der CPU benötigt, dann kann sie einen Interrupt Request (IRQ), kurz Interrupt, auslösen. Ein Interrupt zwingt den Prozessor, seine derzeitige Aufgabe zu unterbrechen und zu einer bestimmten Routine zu verzweigen. Für Interrupts, die von Erweiterungskarten oder den Schnittstellen des PC belegt werden, kann ein Treiber eine eigene Routine einrichten. Der Treiber für eine Maus etwa legt eine Routine auf den Interrupt der seriellen Schnittstelle. Immer dann, wenn die Maus eine Positionsänderung über die serielle Schnittstelle meldet, negiert die CPU alles andere und verzweigt zu einer Routine des Maustreibers. So bleibt der Maustreiber über die Position der Maus informiert. Danach nimmt der Prozessor die unterbrochene Arbeit wieder auf.

Müssen große Datenmengen von oder zu einer Hardware-Komponente übertragen werden, kommt DMA (Direct Memory Access) zum Einsatz. Dabei überträgt der DMA-Controller auf der Hauptplatine und die Hardware des Gerätes Daten direkt zum oder vom Arbeitsspeicher, ohne die CPU zu beanspruchen. SCSI-Controller und Soundkarten arbeiten oft mit DMA. Eine Variante ist Busmaster-DMA, bei dem die Karte die Kontrolle über den PCI-Bus übernimmt.

### CAPL, ASPI und andere Software-Schnittstellen

Viele Treiber unterstützen sogenannte APIs (Application Program Interface) oder Software-Schnittstellen. Ein API stellt eine definierte und standardisierte Schnittstelle zum Zugriff auf die Funktionen der Software dar. Windows besitzt für jede erdenkliche Aufgabe APIs, über die Programme Funktionen des Betriebssystems und der Treiber erreichen.

### Gerätetreiber zwischen Hard- und Software



Die klassische Form der Treibersoftware ist ein Gerätetreiber für das Betriebssystem. Er nimmt alle Zugriffe des Betriebssystems entgegen und übersetzt sie in Zugriffe auf die Hardware. Auf diese Weise ist es für das Betriebssystem und die Anwenderprogramme egal, welcher Drucker oder welche Grafikkarte vorhanden ist – solange es einen passenden Gerätetreiber für dieses Gerät gibt.

Die im Zusammenhang mit Treibersoftware wichtigen Software-Schnittstellen kommen fast immer als Zwitter daher. Einerseits sind sie waschechte Gerätetreiber, die Software-Aufrufe in Hardware-Zugriffe übersetzen; andererseits stellen sie eine definierte Software-Schnittstelle zur Verfügung. Wichtige Vertreter dieser Art sind ASPI, CAPI, DirectX, Twain und Winsock.

**ASPI** ist eine universelle Schnittstelle für Zugriffe auf SCSI-Peripherie. Erst dieser von Adaptec eingeführte Standard hat es möglich gemacht, daß ein einziger Treiber für ein spezielles SCSI-Gerät (ZIP-Laufwerk, Scanner) mit allen SCSI-Controllern von Adaptec bis Ultrastor zusammenarbeitet, solange deren Treiber eine ASPI-Schnittstelle bereitstellt.

**CAPI** läßt sich mit ASPI vergleichen. CAPI ermöglicht einen einheitlichen Zugriff auf ISDN-Komponenten und andere Kommunikationsgeräte. Dadurch ist es zum Beispiel dem T-Online-Dekoder egal, von welchem Hersteller die ISDN-Karte stammt. Hauptsache ist, daß es einen passenden CAPI-Treiber gibt.

**DirectX** ist eine Windows-Software-Schnittstelle mit Treibern für Grafik- und Soundkarten. Sie stellt Low-Level-Routinen für hardwarenahe Programmierung bereit und ist hauptsächlich für Spiele und Multimedia-Anwendungen gedacht. DirectX enthält Routinen aus den Bereichen Grafik, Sound, 3D-Funktionen, Eingabe und Vernetzung. ►



## Tech-Talk

### Was man zur Treiberentwicklung braucht

Suchen Sie ein zeitaufwendiges Hobby? Schreiben Sie doch Gerätetreiber für Windows NT!

Wer wirklich in die Treiberentwicklung einsteigen will, benötigt neben guten C- und Assembler-Kenntnissen das Windows NT Device Driver Kit (DDK) von Microsoft und zwei leistungsfähige PC. Das Windows NT SDK brauchen Sie natürlich auch.

Leistungsfähig heißt: jeweils mindestens Pentium-90 mit 32 Megabyte RAM. Eine schnelle Festplatte erspart Ihnen beim Kompilieren viel Zeit. Der eine Rechner dient dabei als Entwicklungsplattform, der andere als Debugging-Terminal. Beide werden mit einem seriellen Nullmodem-Kabel verbunden.

Das Schwierigste bei der Treiberentwicklung ist nicht, den Treiber zu schreiben; die richtige Arbeit fängt erst beim Debugging an. Wie sonst ließe sich feststellen, an welcher der tausend Fehler-

quellen es liegt, wenn Ihr neuer SCSI-Treiber das System einfach lahmlegt?

Die zur Treiberprogrammierung erforderlichen Dokumente befinden sich auf der CD des DDK, allerdings als Postscript-Datei. Zum Debuggen selbst bietet das Windows NT SDK einen „Debugging Kernel“, der anstelle des normalen Win-NT-Kernels eingesetzt wird. Der eine Rechner dient dabei als intelligentes Terminal, auf ihm läuft die Benutzeroberfläche, der zweite führt den zu debuggenden Code aus. Zum Debuggen von User-Mode-Treibern dient das Hilfsprogramm NTSD. Die Aufrufparameter für beide finden Sie in der SDK-Dokumentation. Bei OS/2 funktioniert das Ganze übrigens sehr ähnlich, auch mit einem Debugging Kernel. Überhaupt werden Sie auf dieser Ebene feststellen, daß es zwischen beiden Betriebssystemen noch viele Parallelen gibt.

Für Scanner und ähnliche Eingabemedien ist *Twain* gedacht. Diese Schnittstelle erlaubt es, Scanner unterschiedlichster Bauart anzusprechen. Der Programmierer einer Scan-Anwendung integriert einfach einen Aufruf zur Twain-Schnittstelle in sein Programm. Solange es einen passenden Twain-Treiber für den verwendeten Scanner gibt, muß sich niemand um die Parameter des Geräts kümmern. Nach Abschluß des Scan-Durchlaufs liefert er das Ergebnis mit Angaben zu Auflösung, Farbtiefe und dergleichen an das Anwendungsprogramm zurück.

### Voller Zugriff aufs Internet

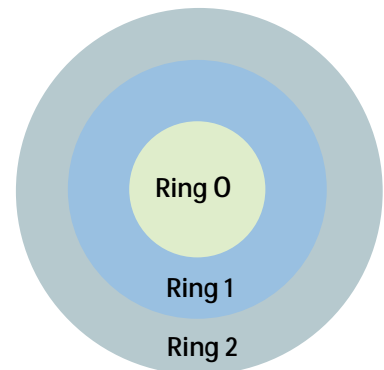
In letzter Zeit hat *Winsock*, die Schnittstelle für das Internet-Protokoll TCP/IP, enorm an Bedeutung gewonnen. Wenn Sie sich über einen Provider in das Inter-

net einwählen, können Sie so beliebige Internet-Programme verwenden. Ob Web-Browser, Mail- oder Newsreader: Mit einer Winsock-Verbindung ist es egal, auf welche Art der Zugang zum Internet erfolgt, ob über Modem, ISDN-Verbindung oder eine Netzwerkkarte.

Die Entwickler von Treibern sollten sich eine Dokumentation der Schnittstelle besorgen. Leider gibt es derzeit keine Stelle, die alle API-Spezifikationen koordiniert, denn im Prinzip kann jeder ein eigenes API definieren. Ein API ist aber nur dann sinnvoll, wenn es einen verbindlichen Standard für Software-Entwickler darstellt.

Ein Aspekt der APIs hat jedoch zur Folge, daß sich API-konforme Treiber schnell stark aufblähen: Selbst ein Treiber, der nur eine einzige Funktion wirklich ausführt, muß alle definierten API-

### Ringstruktur zum Datenschutz



- **Voller Zugriff**  
auf Prozessor und Hardware
- **Beschränkter Zugriff**  
auf Prozessor und Hardware
- **Kein direkter Zugriff**  
auf Prozessor und Hardware

Die Ringstruktur moderner Betriebssysteme schützt vor Systemabstürzen. Nur die Treiber in Ring 0 und das Betriebssystem haben unbeschränkten Zugriff auf den Prozessor und die Hardware. Anwenderprogramme und weniger systemkritische Treiber laufen innerhalb der Ringe 1 und 2. So kann ein Programmfehler in einer Anwendung oder einem Treiber in Ring 1 oder 2 nicht das gesamte System abstürzen lassen.

Funktionen unterstützen. Je nach API-Spezifikationen muß der Treiber dann Zugriffe auf unbelegte API-Funktionen entweder überhaupt nicht ausführen oder mit einem Fehlercode beenden.

### Virtuelle Hardware-Treiber

Bei allen modernen Betriebssystemen, die DOS-Kompatibilitätsmodi bereitstellen, ist ein Problem zu lösen: DOS-Programme greifen gern direkt auf die Hardware zu. In einem innovativen Multitasking-Betriebssystem darf aber keine Anwendung direkt auf Hardware-Komponenten zugreifen. Zu diesem Zweck gibt es Treiber, die der Anwendung im DOS-Fenster Kopien der wirklichen Hardware vorspiegeln. So entstanden die *VxD-Treiber* für Windows 3.x und Windows 95. Jeder direkte Zugriff auf die Hardware wird vom Treiber abgefangen und ausgewertet.

Will ein DOS-DFÜ-Programm unter Windows direkt auf die serielle Schnittstelle zugreifen, fängt der VxD-Treiber



## Vereinfachter Zugriff auf SCSI-Geräte über die ASPI-Schnittstelle



diesen Zugriff ab. Der Treiber überprüft, ob schon andere DOS- oder Windows-Programme die Schnittstelle belegen. Ist sie belegt, verweigert der Treiber den Zugriff. Ansonsten dient der Treiber als Mittler zwischen Schnittstelle und DOS-Software, das DFÜ-Programm „spricht“ also immer nur mit dem Treiber, der wiederum direkt auf die Schnittstelle zugreift. Dazu bedarf es eines sehr großen Aufwands: Der Treiber muß die Hardware komplett per Software nachbilden.

Nur so kann Windows den parallelen Ablauf von DOS- und Windows-Programmen gewährleisten. Darüber hinaus kann ein abgestürztes DOS-Programm nur in Ausnahmefällen das Multitasking-Betriebssystem zum Absturz bringen.

### Software-Treiber, die Hardware emulieren

Es gibt aber noch andere Treiber, die nur so tun, als sei eine bestimmte Hardware vorhanden. Für viele Fälle gibt es Emulatoren in Treiberform, die dem Betriebssystem und den Anwendungsprogrammen bestimmte Hardware vorspiegeln. Für Windows 3.x ist zum Beispiel eine Soundtreiber erhältlich, der eine einfache Soundkarte über den eingebauten PC-Lautsprecher emuliert. Natürlich klingt das Ergebnis ziemlich krächzend, aber wenn eine Software unbedingt auf dem Vorhandensein einer Soundkarte besteht, kann man sich so behelfen. Ähnlich sieht es mit GDI-Druckertreibern aus. Hier übernimmt der Treiber einen Teil der Druckvorbereitung, die sonst ein eigener Prozessor im Drucker erledigt. Dadurch wird der Drucker preiswerter, aber er beansprucht beim Ausdruck auch mehr Rechenleistung und Arbeitsspeicher des PC.

Weitere Beispiele sind Grafiktreiber, die 3D-Funktionen auf Grafikkarten emulieren, oder der Treiber *Wincom*. Er simuliert ein Modem und übersetzt die Modembefehle in Kommandos für eine ISDN-Karte. So lassen sich selbst DOS-Terminalprogramme für eine ISDN-Verbindung nutzen.

### So einfach geht's: Single-Task-Treiber

Die einfachste Form von Treibersoftware ist ein Gerätetreiber für ein Single-Task-Betriebssystem wie DOS. Hier ist er nur ein Assembler-Programm, das eine Anzahl von Funktionen unterstützt und in Hardware-Zugriffe umsetzt. Fast alle Programme, die über DEVICE-Befehle in der Datei CONFIG.SYS geladen werden, fallen in diese Kategorie. Da es in einer solchen Umgebung nur einen Zugriff zur gleichen Zeit geben kann, arbeitet der Treiber ihn einfach von Anfang bis Ende ab und gibt die Ausführung dann wieder an das Betriebssystem zurück.

Nehmen wir als Beispiel den Treiber für ein Wechselpplattenlaufwerk. Beim Boot-Vorgang lädt DOS den Treiber. Danach erhält der Treiber Gelegenheit, eine Initialisierungsroutine zu starten. Nun muß er einige Werte zurückliefern, zum Beispiel, wie viele Laufwerksbuchstaben er reservieren möchte. Wenn ein Zugriff auf diesen Gerätetreiber erfolgt, muß er diesen in entsprechende Hardware-Zugriffe übersetzen und dann das Ergebnis zurückgeben. Dies könnte beispielsweise der Befehl sein, einen Sektor von der Wechselpplatte zu lesen.

### Treiber in Multitasking-Betriebssystemen

Seit Windows 3.x ist das Leben eines Treiber-Entwicklers erheblich schwieriger geworden. Im Gegensatz zu einem einfachen DOS-Treiber müssen Treiber für Multitasking-Umgebungen erheblich vielseitiger und leistungsfähiger sein.

Der wichtigste Punkt ist: Der Treiber muß sich jederzeit mehrfach aufrufen und unterbrechen lassen. In einem Single-Task-Betriebssystem wie DOS kann er sicher sein, kein zweites Mal aufgerufen zu werden. Auch steht fest, daß er seinen Job von Anfang bis Ende durcharbeiten kann, sobald er damit beginnt.

Ein Multitasking-fähiger Treiber kann dies nicht. Er muß darauf vorbereitet

sein, von mehreren Prozessen zur gleichen Zeit aufgerufen zu werden. Er muß auch damit zurechtkommen, jederzeit und beliebig lange bei der Ausführung unterbrochen zu werden, wenn andere Anwendungen zum Zuge kommen.

Zwar hat Microsoft aus Gründen der Kompatibilität Wege geschaffen, wie trotz all dieser Unterschiede Treiber für DOS und Windows 3.x auch unter Windows 95 laufen. Dies führt jedoch zu erheblichen Geschwindigkeitsverlusten und gelegentlich zu Abstürzen, weil diese Treiber wie eine DOS-Anwendung im Emulationsmodus ablaufen. Daher auch hier die Empfehlung: Versuchen Sie, neue Treiber für ihre Hardware zu bekommen, die auf die Multitasking-Betriebssysteme speziell angepaßt sind.

### Treiber für Windows NT in zwei Varianten

Windows NT nutzt eine klare Trennung von Zugriffsrechten, um so das System absturzsicher zu gestalten. Zu diesem Zweck verwendet es, wo immer möglich, voneinander abgetrennte und geschützte Subsysteme. Ein „wildgewordener“ Treiber kann also nicht Datenstrukturen oder Objekte anderer Treiber oder Applikationen zerstören.

Dies und die Anforderungen durch die Portabilität von NT auf verschiedene Hardware-Plattformen führt zu einer Trennung der Treiber in »User-Mode«- und »Kernel-Mode«-Treiber. Stürzt ein Treiber ab, der im User-Mode läuft, kann das System in der Regel stabil weiterlaufen. Ein Fehler in Treibern, die im Kernel-Mode laufen, zieht in der Regel einen Systemabsturz nach sich.

In der Version 4.0 von NT sind zum Beschleunigen der Grafikausgabe Teile der Grafikschnittstelle zu Kernel-Mode-Treibern geworden. Dadurch hat sich die Grafikschnittstelle grundlegend geändert. Aus diesem Grund benötigt NT 4.0 auf jeden Fall neue Grafiktreiber, da solche für Windows NT 3.x unter NT 4.0 nicht mehr laufen. *Martin Hepp (oha)* ☐