

100

Free!

KnowWare

Basics!

Nutze Deinen PC optimal

CONFIG.SYS und AUTOEXEC.BAT

DOS 5, 6.0 und 6.2

Windows 3.1 und 3.11

KnowWare Basics!

Genug Speicher für Spiele
Optimiere Deine Festplatte
DoubleSpace
Memmaker

Verwalte Deine Dateien mit
Norton Commander 3 und 4

Optimale Anwendung von Memory
und viele andere Tips

Michael Maardt

www.KnowWare.de

Acrobat Reader: Wie ...

F5/F6 öffnet/schließt die Ansicht **Lesezeichen**

Strg+F sucht

Im Menü Ansicht stellst du ein, wie die Datei gezeigt wird

STRG+0 = Ganze Seite **STRG+1** = Originalgrösse **STRG+2** = Fensterbreite

Im selben Menü kannst du folgendes einstellen:: **Einzelne Seite**, **Fortlaufend** oder **Fortlaufend - Doppelseiten** .. Probiere es aus, um die Unterschiede zu sehen.

Navigation

Pfeil Links/Rechts: eine Seite vor/zurück

Alt+ Pfeil Links/Rechts: Wie im Browser: Vorwärts/Zurück

Strg++ vergrößert und **Strg+-** verkleinert

Bestellung und Vertrieb für den Buchhandel

Bonner Pressevertrieb, Postfach 3920, D-49029 Osnabrück

Tel.: +49 (0)541 33145-20

Fax: +49 (0)541 33145-33

bestellung@knowware.de

www.knowware.de/bestellen

Autoren gesucht

Der KnowWare-Verlag sucht ständig neue Autoren. Hast du ein Thema, daß dir unter den Fingern brennt? - ein Thema, das du anderen Leuten leicht verständlich erklären kannst?

Schicke uns einfach ein paar Beispielseiten und ein vorläufiges Inhaltsverzeichnis an folgende Adresse:

lektorat@knowware.de

Wir werden uns deinen Vorschlag ansehen und dir so schnell wie möglich eine Antwort senden.

EDIT und Bat(ch)-Dateien	7	Windows 3.1.....	39
ASCII und Zahlensysteme	8	Auslagerungsdatei/Swapfile	39
System mit sechzehn Ziffern	8	32-Bit Zugriff	40
Speicher (RAM, ROM).....	9	Tips zu Windows.....	40
Speicherarten.....	10	DOS 6	41
Konventionelle Speicher.....	10	EMM386.....	41
Upper Memory / Oberer Speicher	10	Memmaker.....	41
Expanded und Extended Memory	11	Custom.....	42
High Memory Bereich	11	More Boots	43
Das Verhältnis von Upper zu Extended Memory	11	F5.....	43
Device/Gerät.....	11	F8 und ?	43
Startdateien	12	Booten mit Optionen	43
Die wichtigen Dateien	12	Numlock	44
Lesehinweis	15	MS-DOS 6.2	45
Der Start- oder Bootvorgang	15	Doublespace	45
CONFIG.SYS	15	DBLSPACE benutzen?	45
286.....	16	Custom	46
386.....	16	Express.....	47
Verlangt EMS-Speicher	17	Custom und Express	47
Weiteres in der CONFIG.SYS.....	21	Rückgängig machen.....	47
AUTOEXEC.BAT	22	Windows f. Workgroups 3.11.....	48
Zeichensatztabellen/Codepages-Diskussion	24	Tips zur Optimierung	49
SMARTDrive Festplatten-Cache.....	25	Die Nutzung des Upper Memory.....	49
Allgemeines.....	25	Schau in den Speicher – MSD.EXE	50
SMARTDRV.EXE.....	25	DOS	51
DoubleBuffering/Doppelte Pufferung	26	CHKDSK/F	51
Die Festplatte.....	27	Boot-Disketten.....	51
Festplattenoptimierung	27	Diskette 1 – clean boot.....	51
Norton Commander (NC)	31	Diskette 2	52
Einleitung	31	CMOS und Setup	53
Einstellungen.....	32	Blind- & Schnellschrift.....	53
Jetzt legen wir los!.....	33	Deine Gesundheit.....	54
Kopieren mit XCOPY	34		
Versteckte Dateien	34		
Dateiangaben wählen	34		
Umbenennen/bewegen	36		
Textdateien ansehen oder editieren	36		
Nach Dateien suchen.....	36		
Anlegen einer neuen Datei	36		
Die zuletzt benutzten DOS-Befehle	36		
Fenster an- und ausschalten.....	37		
Fenster wechseln	37		
Verzeichnisse wechseln.....	37		
Verzeichnisse durchblättern	37		
Verzeichnisse vergleichen.....	37		
Menü	38		
Version 4.0	38		

Ein Heft für dich?

Dies ist *kein* Heft für Einsteiger, sondern eher eine Ergänzung zu vielen Einsteigerbüchern. Hier und da können Einsteiger aber bestimmt auch etwas davon haben.

Die Abschnitte haben absichtlich *verschiedene* Schwierigkeitsgrade. Du hast am meisten von dem Text, wenn du mindestens DOS 5 in deinem PC installiert hast, während DOS 6 selbständig beschrieben wird.

Bist Du Anfänger, empfehle ich erst das Heft *Start mit DOS*

Die Idee von KnowWare

April 1993 kam die erste Ausgabe dieses Heftes in dänisch heraus. Es hatte 48 Seiten, und ich forderte die Leser auf, Verbesserungsvorschläge zu machen. Zum Glück erhielt ich hunderte von Briefen und Anrufen von Leuten, die fanden, daß ein Heft zum Preis von 5,00 DM eine verdammt gute Idee sei. Grob gesagt benutzte ich meine Leser als Korrekturleser und forderte sie auf, das Heft noch einmal zu kaufen, sobald die zweite Auflage vorlag.

Das wichtigste Motiv war, meine PC-Kenntnisse so billig wie möglich an so viele Menschen wie möglich weiterzuvermitteln. Computerbücher sind zu teuer. Was mich und viele andere interessiert, sind Informationen und Grundkenntnisse, die die Arbeit wesentlich erleichtern. Ob ich mir dieses Wissen durch ein schönes Buch mit 4-farbigem Umschlag oder durch ein simples Heft aneigne, ist mir völlig egal – der Preis aber ist nicht gleichgültig.

Die Hefte werden, wie auch viele Tageszeitungen, im Rotationsdruck gedruckt. Das heißt, daß unheimlich viele Seiten auf endlosen Bahnen Papier auf beiden Seiten gleichzeitig gedruckt werden. Ein Heft kann auf diese Weise im Laufe weniger Stunden zu 10.000 Exemplaren werden; der Buchbinder schneidet dann die Seiten aus, setzt Heftklammern ein und packt die fertigen Hefte in Bunde. Es gefällt mir, daß das Heft im Rücken ganz geknickt werden kann, ohne daß es gleich aufspringt und die Seiten herausfallen. Man kann es auch falten und in die Innentasche stecken, eine Briefmarke hinten draufkleben und losschicken – und morgen landet es bei einem guten Freund.

Idee und Hintergrund

Im Herbst 1992 hatte ich die Zeit und die Idee für dieses Heft. Über die notwendigen Kenntnisse für die "schwierigen" Teile des Inhalts verfüge ich erst seit März 1993, das heißt nachdem ich mein erstes echtes Computerfachbuch gelesen hatte: John M. Goodman: *Memory management for all of us*.

Kurz über mich selbst

Ich bin 1952 geboren und von 1970-80 in Dänemark und ein paar Jahre in Deutschland Soziologie und Psychologie studiert. Seit 1986 beschäftige ich mich mit PCs. 1988-91 arbeitete ich jedoch überwiegend mit anderen Computersystemen.

Vielen Dank

an alle, die mich unterstützt und mir viele verschiedene Dinge beigebracht haben. Mein Dank geht auch an alle, die mir helfen, den Inhalt dieses Heftes zu verbessern und zu verbreiten. Danke für all die positiven Zuschriften und Vorschläge. Besonders herzlichen Dank an meine übrigen Lehrer und Berater.

Viel Vergnügen

beim Lesen. Ich hoffe, daß du mehr Spaß an der Arbeit mit deinem PC bekommst, und daß dieser Text dir größeres Verständnis und eine größere Einsicht in die Geheimnisse eines PCs geben wird.

Michael Maardt

Wichtig

Was immer du mit den Vorschlägen und Ideen dieses Textes anfängst: du tust es auf *eigene Gefahr*. Meine Erfahrung habe ich in erster Linie mit PC-Clones gemacht, die in Taiwan gefertigt wurden.

Bist du schon ungeduldig und möchtest sofort deine **CONFIG.SYS** ändern, dann halte um Himmelswillen eine Boot-Diskette bereit, die *funktioniert*. Mit einer Boot-Diskette kannst du deinen PC starten. Lege eine leere, formatierte Diskette in das Laufwerk und gib folgendes ein:

```
C:\>SYS A:
```

Siehe auch den Abschnitt *Bootdisketten* auf Seite 43. Ich würde dir raten, den Text von Anfang bis Ende zu lesen – einschließlich der Verweise, die dich dazu zwingen, ein bißchen hin- und herzuspringen. Dies betrifft insbesondere den Abschnitt **CONFIG.SYS**. Denn wenn du mit dem Betriebssystem DOS 5 arbeitest, *kann* es passieren, daß du eine Eingabe oder Änderung vornimmst, die dazu führt, daß du deinen PC nicht von der Festplatte aus starten kannst, weil diese Eingabe dein Gerät blockiert. Nun bist du gewarnt. Bei DOS 6 tritt dieses Problem zum Glück nicht auf.

Hast du die Startdateien **CONFIG.SYS** und **AUTOEXEC.BAT** schon öfters bearbeitet und eine Bootdiskette, die funktioniert, kannst du mit *CPU und Memory* S. 9 anfangen.

Verstehst du von diesen Abschnitten wenig, ist das nicht so schlimm. Du kannst dich zunächst einmal damit begnügen, die Startdateien nach meinen Vorschlägen zu ändern – oder es von jemand anderem machen zu lassen!

Kommen unterwegs technisch komplizierte Abschnitte vor oder Abschnitte, die sich an fortgeschrittene Benutzer wenden, mache ich darauf aufmerksam. Ich gehe davon aus, daß du DOS im Verzeichnis C:\DOS installiert hast, und beschreibe nur die Windows-Version 3.1x.

Englische Ausdrücke

Ich benutze mit Absicht vor allem die englischen Bezeichnungen wie Directory, path, upper Memory, high Memory, extended Memory u.s.w. Diese Begriffe werden später erklärt. Du wirst immer wieder auf englische Ausdrücke stoßen – also kannst du sie genauso gut lernen.

Jargon

Die EDV-Welt – und damit auch die PC-Welt – hat ihren eigenen Jargon. Ich werde im Verlauf des Heftes versuchen, einige der Begriffe zu erläutern, die andere Benutzer und ich oft gebrauchen. DOS-Befehlszeilen werden wie folgt

mit Courier

hervorgehoben. Wenn ich schreibe: 'schreibe in DOS', bedeutet das, daß du dich "im DOS" oder an der DOS-Eingabeaufforderung (sieht so aus: C:\>) befindest. Auf dieser Ebene kannst du DOS-Befehle eingeben. Einige Programme können vorübergehend auf diese Ebene (DOS) springen, um über die Eingabe **EXIT** wieder zurückzuspringen.

Wenn du in einem Directory stehst oder bist – wenn du dich zum Beispiel in C:\EXTRA befindest – bedeutet das, daß DOS-Befehle ohne weitere Angaben (englisch *Default*) in diesem Verzeichnis ausgeführt werden.

Default ist ein guter Begriff, den es im Deutschen leider nicht gibt und der auch in Englisch schwer verständlich sein kann, wenn man ihn noch nicht kennt.

Die beste deutsche Beschreibung für *Default* wäre "*mangels eines Besseren*" oder "*wenn nichts anderes festgelegt ist, nehmen wir das, was wir haben*". Wenn du dich in einem Verzeichnis befindest und **DIR** eingibst, würde DOS dir mitteilen: "Wenn du mir nicht sagst, für welches Verzeichnis ich dir die Datensätze anzeigen soll, dann nehme ich die Default-Einstellung" – und Default ist dann das aktuelle Verzeichnis, also das, in dem man sich gerade befindet. Und befindet man sich in DOS, dann befindet man sich auch immer in einem Verzeichnis auf einem Laufwerk.

In Programmen wird oft der Begriff Default verwendet, zum Beispiel "using Default Settings". Wenn du ein Dokument in deiner Textverarbeitung speichern oder laden möchtest, wird dazu ein Default-Verzeichnis verwendet, das jederzeit geändert werden kann. Wenn du deine Textverarbeitung lädst und mit einem leeren Dokument beginnst, sind viele Vorgaben wie zum Beispiel linker und rechter Rand bereits mit bestimmten Werten als Standard – also *Default*-Werten, eingestellt. In der Regel können die meisten bzw. alle dieser voreingestellten Werte vom Anwender geändert werden. Ich hoffe, du hast jetzt eine Vorstellung davon, was Default bedeutet.

Dateinamen werden normalerweise folgendermaßen geschrieben: **HIMEM.SYS**. *Festplatten- und Verzeichnis- (Directory-)namen* werden immer mit Großbuchstaben geschrieben – C:\DOS.

(A) bezeichnet die Kapitel, die ich für Fortgeschrittene geschrieben habe – sie werden durch diesen Schrifttyp hervorgehoben.

Diesen Abschnitt bitte lesen

Dieser Abschnitt handelt nicht von technischen Dingen. Bevor ich anfang, diesen Text zu schreiben, half ich vielen Freunden; dabei stellte ich fest, wie zeitaufwendig das Kennenlernen eines PCs und der dazugehörigen Programme ist. Stößt man auf ein schwieriges Problem, kann die Problembeseitigung viele Stunden in Anspruch nehmen – und nicht alle können es sich leisten, andere dafür zu bezahlen, die Probleme für sie zu lösen. Ich hoffe, ich kann dazu beitragen, diese Zeit für dich zu verkürzen – was jedoch bedeutet, daß du zunächst ein wenig Zeit investieren mußt. Du wirst feststellen, daß es sich lohnt.

Falls du Anfänger bist, werde jetzt nur nicht nervös. Millionen vor dir haben es gelernt – aber alles hat seinen Preis. Anfangs wirst du vielleicht nur Freude an ein paar Kapitel finden; später kannst du dann zu den anderen zurückkehren. Noch vor wenigen Jahren verstand ich fast kein Wort von dem, was ich heute beschreibe. Aber Bekanntlich lernt man aber selbst am meisten, wenn man anderen etwas beibringt.

Je später du dich in den PC-Dschungel hineingewagt hast, desto schneller wirst du deinen PC kennenlernen – und benutzen. Die Programme und ihre Online-Hilfen werden immer besser, und Computer-Fachbücher gibt es auch immer mehr.

Es gibt grundlegende Dinge, die einfach in Ordnung sein müssen, bevor dein PC optimal funktioniert. Mit der Zeit mußt du sicherlich auch eine bestimmte Ordnung und Disziplin lernen, wenn du wirklich möglichst viel Nutzen aus deinem PC ziehen willst.

Der traurige Widerspruch dieses Computerzeitalters ist, daß es in Büchern, Zeitschriften und den Köpfen der Menschen eine Menge Wissen gibt – aber niemand genau weiß, wo es ist oder wie schnell es gefunden werden kann. Niemand in dieser Branche weiß alles – denn alles geht viel zu schnell, der Markt ist sehr groß, und dauernd kommen neue Produkte auf den Markt.

Einer der Gründe für das Erscheinen dieses Heftes ist das schlechte Handbuch von Microsoft zu DOS 5. Auf 600 Seiten findet sich nicht einmal Platz für ein einziges Beispiel der wichtigen Startdateien **AUTOEXEC.BAT** und **CONFIG.SYS!**

Je mehr ich mich während des Schreibens mit technischen Details befaßte, desto mehr handelte das Heft von Optimierung. Viele seiner Teile wurden wesentlich anspruchsvoller als ich mir ursprünglich vorgestellt hatte; aber diejenigen, die meine Tips verstehen und sie umsetzen können, freuen sich hoffentlich darüber und helfen sicher auch ihren

Freunden, die das alles nicht so gut verstehen, nicht wahr?

Die Startdateien sind zwar nicht der aufregendste Teil eines PCs; für uns Anwender haben sie aber große Bedeutung, was von den meisten Fachleuten in der Branche “übersehen” wird. Ich wollte mit diesem Heft unter anderem *mein eigenes Wunschbuch* schreiben, also das Buch, das ich selbst gern gehabt hätte, als ich Schwierigkeiten mit meinem PC hatte.

Weil das Heft auf verschiedene Schwierigkeitsstufen ausgerichtet ist, war es etwas schwierig, die einzelnen Abschnitte so anzuordnen, daß ein einmaliges Durchlesen ausreicht. Möglicherweise vermißt du auch einen roten Faden, dem du folgen kannst. Bist du ein erfahrener PC-Anwender, wirst du dich wohl auch mit manchem Anfängerstoff abfinden müssen.

Einige Leser werden das Heft rasch durchblättern, um sich zu orientieren – um dann wieder von vorne anzufangen. Wenn ein Begriff das erste Mal auftaucht, wird er in angemessenem Umfang beschrieben.

Obwohl ich schon früher über andere Themen geschrieben habe, muß ich gestehen, daß es schwer gewesen ist, diesen Text mit seinen zeitweise schwer zugänglichen Themen zu strukturieren. Es ist nicht immer leicht, das Gleichgewicht zwischen Theorie und Praxis zu wahren.

Ich setze voraus, daß du darauf eingestellt bist, deinen Kopf zu benutzen, das heißt *Lust zum selbständigen Denken* hast! Jeder PC birgt tausende spezifischer Probleme in sich. Ich habe versucht, einen theoretischen Überblick mit anwendbaren, konkreten Beispielen zu vermitteln, so daß du diese Hilfestellung auch auf andere als die in diesem Heft genannten Probleme übertragen kannst.

EDIT und Bat(ch)-Dateien

Kennst du dich noch nicht so gut mit dem Bearbeiten von Textdateien oder dem Erstellen von Bat(ch)-Dateien aus, folgt jetzt ein Minikurs – und eine Warnung: ändere oder bearbeite Startdateien erst dann, wenn du wirklich weißt, was du tust. Der Editor **EDIT**, der mit DOS 5 und 6 mitgeliefert wird, wird in *Fang an mit DOS* (erscheint Juli 95) beschrieben.

Wir wollen uns einmal ansehen, wie man auf schnelle Weise eine Textdatei erstellt; dazu stellen wir, um uns das Prinzip klarzumachen, eine Datei namens **EASY.BAT** her.

`Copy con` bedeutet “kopiere von der Konsole”. Konsole ist Bezeichnung für *Bildschirm+Tastatur*.

```
C:\BAT>copy con easy.bat
```

Der Cursor blinkt, und du befindest dich in einer Textdatei. Gib folgende Zeilen ein:

```
c:
cd \dos
dir/w
cd \bat
```

Drücke jetzt **F6** und **Eingabe**. DOS meldet dir:

```
1 File(s) copied.
```

Die Datei **EASY.BAT** ist erstellt. Jetzt kannst du diese Batch-Datei starten, indem du einfach **EASY** schreibst. Versuch’s einfach. Du kannst den Prozeß stoppen und wieder starten, indem du **Strg+S** drückst, wenn du ihn genauer ansehen möchtest. **Strg+S** bedeutet, daß du **Strg** hältst, ohne sie loszulassen, und dann **S** drückst, so daß beide gleichzeitig aktiviert sind.

Als nächstes kannst du eine Batch-Datei namens **PRI.BAT** erstellen, die automatisch **EDIT** aufruft und **EASY.BAT** in den Editor lädt.

```
C:\BAT>copy con pri.bat
edit c:\bat\easy.bat
```

Drücke **F6** und **Eingabe**.

Wenn du jetzt **PRI** schreibst, wird **EDIT** gestartet und lädt (load) die Datei **C:\BAT\EASY.BAT**.

Nun kannst du die folgenden zwei Batch-Dateien erstellen: **EC.BAT**, die das Editieren von der Datei **CONFIG.SYS** startet, und **EA.BAT**, die das Editieren der Datei **AUTOEXEC.BAT** startet. Die zwei Verzeichnisse **DOS** und **BAT** müssen in deinem **PATH** vorhanden sein.

```
C:\BAT>copy con ec.bat
edit c:\config.sys
```

Drücke **F6** und **Eingabe**.

```
C:\BAT>copy con ea.bat
edit c:\autoexec.bat
```

Drücke **F6** und **Eingabe**.

Das Besondere an **EDIT** ist, daß dieser Editor nicht, wie gewöhnliche Textverarbeitungen, ein ‘word-wrap’ macht, was bedeutet, daß auf dem Bildschirm eine neue Zeile angefangen wird, obwohl der Text das eigentlich nicht tut.

Texteditoren arbeiten auf verschiedene Weise; die meisten funktionieren aber folgendermaßen: ein Abschnitt, das heißt der Text zwischen zwei Zeilensprüngen (ein Zeilensprung entsteht durch das Drücken der **Eingabe**) wird als eine einzige Zeile angezeigt, die über den rechten Bildschirmrand hinausgeht (maximal 256 Zeichen pro Zeile). Daran muß man sich zunächst gewöhnen. Daß der Editor nicht mit “word-wrap” arbeitet, ist im übrigen sehr praktisch, wenn man an Dateien mit langen Zeilen arbeitet. Jedes Programm hat seine eigenen Vorteile.

ASCII und Zahlensysteme

Einen meiner ersten Durchbrüche hatte ich vor vielen Jahren, als ich begann, ASCII (American Standard Code for Information Interchange) zu verstehen. Es wird "Aski" ausgesprochen und ist eine Definition für das Verhältnis zwischen einem Zeichen/Symbol und einer Zahl.

Wenn eine Taste gedrückt wird, und auf dem Bildschirm ein Buchstabe erscheint, dann könnte man glauben, der Buchstabe säße sozusagen in der Taste und warte nur darauf, aktiviert zu werden. Aber so einfach ist das nicht.

Ich will auf den *Scan Code* der Tastatur hier nicht näher eingehen – dieser Code ist es aber, der beim Drücken einer beliebigen Taste abgeschickt wird. Der Code wird an einen "Übersetzer" gesendet, der herausfindet, welcher ASCII-Wert dieser Taste zugeordnet ist, und diesen dann in ein Zeichen umsetzt, das auf dem Bildschirm angezeigt wird.

In der elektronischen Datenverarbeitung gibt es nur zwei mögliche Zustände: an oder aus, 0 oder 1 – sonst nichts. 1 Bit ist der Raum, den dieser Zustand beansprucht. Ein Computer kann nichts anderes als entweder eine 0 oder eine 1 abzuschicken! 8 Stück davon machen einen sogenannten *Byte* aus. Buchstaben und andere Zeichen und Symbole werden im Grunde als eine Serie von Nullen und Einsen in Kästchen mit jeweils 8 Plätzen gelagert. Ein Platz kann nur mit entweder einer 0 oder einer 1 ausgefüllt werden (ein binäres oder Zweizahlen-System).

Bei 8 Plätzen hat man "2 hoch 8" oder $2^8 = 256$ Möglichkeiten, einen Zahlenwert zu gestalten. Hier gilt Null auch als ein Wert, weshalb oft die Rede von 0–255 Werten ist.

Mit der Zeit hat man sich zum Glück darauf geeinigt, welche Werte den verschiedenen Zeichen und Buchstaben zuzuordnen sind in diesem System, ASCII, das sich auf die 256 Möglichkeiten begrenzt. Als Beispiel wollen wir uns das große A ansehen. Halte in DOS oder in einem Programm die **Alt** nieder, während du auf der numerischen Tastatur 65 drückst – dann läßt du die **Alt** los.

So einfache Dinge sind also die Grundlagen dessen, was man *Datenkommunikation* nennt. Im Vergleich zur Entwicklung von Gehirn und Geist sind wir Menschen mit Computern also noch nicht besonders weit gekommen, obwohl wir Fortschritte machen. Der einzige Grund für die Verbreitung des Computers ist seine Geschwindigkeit. Glaube nur nicht, ein Computer könne intelligent sein, auch wenn es Leute gibt, die uns das weismachen wollen und von künstlicher Intelligenz, neuronalen Netzwerken und so weiter reden.

System mit sechzehn Ziffern

(A) In der Computerwelt ist das System mit sechzehn Ziffern sehr verbreitet. Unser normales Zahlensystem ist das Zehner-Zahlensystem, mit dem wir uns in der Schule herumplagen mußten.

0 1 2 3 4 5 6 7 8 9 und wieder von vorne – denn mehr Zeichen, Finger, Zehen haben wir halt nicht – nur setzen wir dann eine 1 an die erste Stelle (links) und haben die 10.

Im Hexadezimalsystem, also im System mit sechzehn Ziffern, sieht das anders aus, nämlich:

0 1 2 3 4 5 6 7 8 9 A B C D E F – dann erst (wenn wir in unserem 10er-System bei 16 angelangt sind) kommen wir zu einer Zahl, die der Zehn entspricht.

Im Zehner-System sagen wir: Die erste Ziffer von hinten gelesen sind die Einer, die nächste sind die Zehner, die nächste sind die Hunderter und so weiter. Im Sechzehner-System fangen wir auch hinten an und sagen (weil wir auch weiterhin im Zehner-System denken): Das erste Zeichen bis F (=15) sind Einer, das nächste Zeichen sind Sechzehner, das dritte sind 256er und so weiter. Gar nicht so schwierig – oder wie?

Das Dezimalsystem (Zehner-Zahlensystem) ist in Verbindung mit Computern gar nicht anwendbar. Es ist ein Überbleibsel des Menschen, eines Geschöpfes, das mit 10 Fingern denkt.

Zahlensysteme		
2er	10er	16er
00010000	16	10
00100010	34	22
11111111	255	FF
	256	100
	65536	10000
	1048576	100000

In der EDV-Welt wird mit riesigen Zahlenwerten gearbeitet. Deshalb ist das Hexadezimalsystem auch so praktisch. Ab und zu werden Zahlen mit einem h am Ende angegeben, zum Beispiel A0000h. Wie wir später sehen werden, wird die letzte Null oft weggelassen, so daß aus obenstehender Zahl A000h wird, was im 10er-System zu den bekannten 640KB = $640 \times 1024 \text{ Bytes} = 655360 \text{ Bytes}$ wird.

Mit dem Windows-Taschenrechner (wähle VIEW, Scientific), kannst du mit diesen Zahlen ein wenig herumspielen und sie leicht umrechnen.

Speicher (RAM, ROM)

Ein Teil des nachstehenden Textes ist ziemlich technisch und kann gegebenenfalls übersprungen werden. Der Speicher ist der Ort, wo Informationen aufbewahrt werden. Ein PC verfügt über zwei verschiedene Speicher: den ROM- und den RAM-Speicher.

Der erstere, das **Read Only Memory**, wird in diesem Heft nicht ausführlich beschrieben. Um es einfach zu machen: Man hat sich dazu entschlossen, einen Teil des Betriebssystems in Chips zu speichern. Andere ROM-Speicher findet man in den physischen Komponenten des Computers, wie zum Beispiel der Grafikkarte und dem Festplattencontroller.

Der zweite Speicher, das **Random Access Memory**, ist der flüchtige Speicher, der verfügbar ist, solange der PC angeschaltet ist. RAM ist wichtig. Die CPU benutzt und bearbeitet die Daten im RAM, dessen Inhalt sich folglich laufend ändern kann.

Zunächst etwas über die Zuteilung des RAM in einem PC – die ziemlich kompliziert ist, wenn das Betriebssystem DOS heißt. Computer arbeiten mit dem binären Zahlensystem (2er-System). Man trifft immer wieder auf Zahlen, die zum Beispiel Potenz-erhöhungen der Zahl zwei sind, das heißt sie sind auf der Zweier-Basis berechnet: 2, 4, 8, 16, 32, 64, 128, 256, 512 Bytes – hast du 1.024 Bytes, nennt man das 1 **KiloByte (KB oder K)**; danach folgst du dem gleichen Muster 2, 4... und so weiter bis 1.024 KB, was einem **MegaByte** entspricht (**MB**); dann weiter bis zu 1.024 MB=1 **GigaByte**.

Als IBM den ersten PC baute, konnten sich die Hersteller nichtvorstellen, daß er ein großer Erfolg würde. Man rechnete damit, daß 1 MB (1024 KB) RAM mit 640 KB Speicherplatz für Programme voll und ganz ausreichen würde.

Sehen wir uns nun das erste MB RAM, also die ersten 1024 KB RAM, genauer an. Stell dir ein riesiges Bücherregal mit viel Platz für identische Bücher vor. Jeder freie Buchplatz kann ein Byte enthalten. Ein solcher Platz hat eine sogenannte Adresse. Nachstehend wird beschrieben, wie diese im Grunde logischen Adressenfelder definiert und behandelt werden.

Die Arbeit der CPU besteht hauptsächlich darin, alle Bytes zwischen der Festplatte, der CPU selbst, dem RAM-Speicher, dem Monitor und so weiter. hin- und herzubewegen, und das unheimlich schnell. Fast die gesamte Kommunikation in einem Computer – das heißt der Transport von Daten zwischen den einzelnen Einheiten (*Devices*) – muß über die CPU vor sich gehen. Monitor, Tastatur,

Laufwerk und so weiter sind physische Einheiten beziehungsweise Geräte (*Devices*); aber auch andere eher logische oder abstrakte Teile eines PCs können diese Bezeichnung tragen.

Wir werden später sehen, daß ein jedes *Device* gesteuert werden muß. Für diese Steuerung sorgt ein spezielles Programm, das *Device Driver* bzw. *Gerätetreiber* oder schlicht *Treiber (Driver)* genannt wird. *Device* ist meines Erachtens ein merkwürdiger Begriff; es hat Jahre gedauert, bis ich ihn richtig verstand.

DOS kan nur den ersten MB “sehen”. Wenn der Prozessor in seinem primitivsten Modus läuft, dem *Real Mode*, kann er ausschließlich diesen Adressenbereich im RAM “sehen”. DOS ist ein sogenanntes *Real Mode* Programm. Wenn du deinen PC bootest, “erwacht” der Prozessor in diesem *Real Mode*. Wenn man mehr von seinem PC haben möchte als *Real Mode* und DOS, dann muß dieses Fundament auf- und ausgebaut werden.

Um 8086er und 80286er sowie ältere DOS-Versionen weiterhin benutzbar zu halten, sind DOS 5 und DOS 6 noch immer *Real Mode* Programme und *rückwärtskompatibel*, was bedeutet, daß alle Programme, die mit älteren DOS-Versionen erstellt wurden, auch mit neueren Versionen arbeiten.

Will man mehr als 1 MB RAM-Speicher nutzen, benötigt man ein Programm, das den Prozessor im sogenannten *Protected Mode*, im geschützten Modus, laufen läßt, was bedeutet, daß zwei Programme, die sich gleichzeitig im RAM-Speicher befinden, nicht denselben Adressenbereich im Speicher benutzen. Der 80286er war der erste Prozessor, der sowohl im *Real Mode* als auch im *Protected Mode* laufen konnte.

Der Prozessor kann sich immer nur in jeweils *einem* Modus befinden; und es dauert einige Zeit, vom *Real Mode* in den *Protected Mode* umzuschalten. Eine CPU arbeitet mit einer sogenannten Taktfrequenz, das heißt Schwingungen pro Sekunde (Hertz) – zum Beispiel 25, 33, 40 oder höher. Diese Frequenz wird in MHz gemessen (1 Mio. Hertz).

Speicherarten

Dieses Kapitel ist hauptsächlich technisch, kann aber vielleicht dazu beitragen, daß das Kapitel über die Startdateien leichter zu verstehen ist.

1024–	extended high = die ersten 64K
640–1024	upper
0–640	konventionell

Konventionelle Speicher

Der Speicherbereich von 0 – 640 KB heißt *konventioneller Speicher (Lower Memory)*. DOS ist der Speicher-Manager für den konventionellen Speicher, das heißt DOS steuert diesen Speicherbereich.

Upper Memory / Oberer Speicher

Ich habe diesen Speicherbereich mit eingeschlossen, da er für die Betriebssysteme DOS 5 und höher wichtig ist. Dort besteht die Möglichkeit, bestimmte Programme in diesen Adressbereich zu laden und damit mehr Platz im konventionellen Speicher zu schaffen, um DOS-Programme laufen zu lassen.

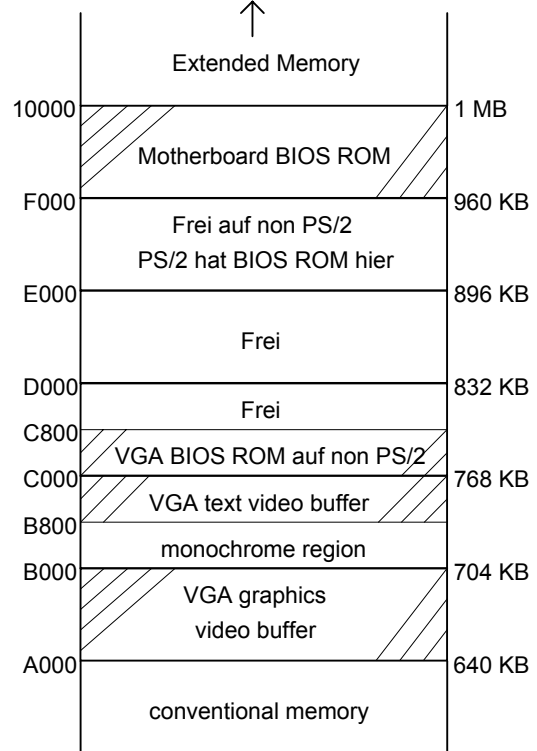
Ein großer Teil dieses Abschnitts handelt von diesem Thema. Falls du noch kein Wort davon verstehst, mach dir nichts draus – ich selbst verstehe auch noch nicht alles. Aber jedesmal, wenn ich darüber schreibe, verstehe ich etwas mehr – hoffe ich jedenfalls.

Manchmal frage ich mich, wie vernünftige Menschen einen PC mit so komplizierten Bestandteilen wie einem Betriebssystem, Speicher und so weiter konstruieren konnten – aber sie haben es getan!

IBM hat klugerweise den Adressbereich von 640 1024 (384KB) für Systemzwecke, technische Zwecke und zukünftige Zwecke reserviert. Diesen Speicherbereich nennt man im allgemeinen *Upper Memory* (Oberer Speicher) und ist die Bezeichnung für einen *Adressbereich*. Es gibt Vorschriften dafür, an welcher Stelle im Upper Memory die verschiedenen Geräte (*Devices*) ihre *Daten* positionieren sollen, damit die CPU im direkten Zugriff mit ihnen kommunizieren kann. Die markierten Bereiche der Grafik zeigen, wo diese Stellen sind.

Aber es wird nicht der gesamte Bereich ausgefüllt; also kann der verfügbare Platz unter diesen Adressen für Treiber und speicherresidente Programme genutzt werden. (*Resident* bedeutet, daß das Programm permanent im Speicher geladen ist und deshalb nicht jedesmal von der Festplatte geladen werden muß, wenn es gebraucht wird. Es liegt sozusagen dauernd auf Lauer. Speicherresidente

Programme nennt man auch TSR(Terminate and Stay Resident)-Programme).



Das interessante dabei ist, daß dieser Bereich in Reichweite von DOS und Real Mode liegt. DOS kann ihn also steuern bzw. sich mit ihm unterhalten; *deshalb* können DOS-Treiber und DOS-residente Programme in diesen Adressbereichen laufen.

In den letzten Jahren wurden Methoden entwickelt, um mit diesen Adressbereichen "sprechen" zu können. Über EMM386.EXE, das den Kontakt herstellt, kann DOS mit diesem Bereich in Verbindung treten. In diesem Heft werden Programme wie QEMM und 386MAX nicht behandelt.

Der Begriff Upper Memory ist *nicht* das gleiche wie die sogenannten Blocks, bzw. Upper **Memory** **Blocks** (UMB), obwohl diese Blocks – wenn sie zum Beispiel mit EMM386.EXE erstellt sind – im oberen Adressspeicherbereich platziert sind. Upper Memory ist die Bezeichnung für ein *Adressfeld*. UMB ist der *tatsächlich nutzbare RAM* (Speicherplatz) in diesem Bereich.

Der Upper Memory wird entweder von der Datei HIMEM.SYS oder von DOS gesteuert. Beim Booten steuert zunächst die HIMEM.SYS diesen Bereich. Gewöhnlich übernimmt jedoch dann DOS durch den Befehl DOS=UMB, der in die CONFIG.SYS eingefügt wird, die Steuerung.

Die Abbildung zeigt beispielhaft, wie der Upper Memory eines PCs aussehen kann. Sie kann dir vielleicht behilflich sein, wenn du anfängst zu

untersuchen, an welchen Stellen in diesem Bereich sich verschiedene Programme selbst laden.

Das **IBM PS/2-System** stellt das BIOS-ROM in die beiden "Top"-Bereiche, zum Beispiel die Seiten E und F. Das VGA-BIOS-ROM wird zwischen E000 und E7FF eingerichtet. Somit sind die Bereiche von C000 bis E000 frei für die UMBs.

Expanded und Extended Memory

Für die folgenden Ausführungen ist es wichtig, einen Unterschied zu machen zwischen dem physischen RAM bzw. Speicher und den abstrakten Bezeichnungen (Upper, Extended etc. Memory) für die Art und Weise, wie der physische RAM genutzt wird. Nehmen wir an, du hast 4 MB physischen RAM. Dieser Abschnitt beschreibt nun, wie du ihn auf verschiedene Arten benutzen kannst. Um festzulegen, wie der RAM-Speicher genutzt wird, gebraucht man den Ausdruck "Speicher konfigurieren".

Die Verwendung von mehr als 1MB RAM begann vor Jahren vor allem mit der Tabellenkalkulation Lotus 1-2-3, bei der man rasch an die Decke des RAM stieß, die 640KB, die die obere Grenze der DOS-Programme waren. Die drei Firmen Lotus-Intel-Microsoft entwickelten Regeln, nach denen das sogenannte *Expanded Memory*, der erweiterte Speicher, benutzt werden sollte. Diese Regeln nannte man **Expanded Memory Specification (EMS)** – manchmal auch LIM.EMS

Damals wurden die ersten Speichererweiterungskarten für den PC hergestellt. Es sind in erster Linie ältere Programme, die diese Speicherart benutzen können, aber auch viele Spiele benutzen das EMS Memory (siehe *Mehrere Boots*, S. 43). Ein Speichermanager für das Expanded Memory heißt **Expanded Memory Manager (EMM)**. Im DOS trägt er den Namen **EMM386.EXE**.

Später wurde ein weiterer Standard für die Nutzung des Speichers oberhalb von 1 MB eingeführt, **eXtended Memory Specification (XMS)** (Erweiterungsspeicher). Für die meisten Anwender ist heute der XMS-Standard von größerer Bedeutung, da die neueren Programme für das Extended Memory geschrieben werden. Der dafür erforderliche Speicher-Manager im DOS heißt **HIMEM.SYS**.

Diese Begriffe bezeichnen die Benutzung des physischen RAM. Man könnte sie auch als Logische Bezeichnung bzw. Standardisierte Vorschriften dafür bezeichnen, wie der physische RAM genutzt werden kann. Der RAM-Speicher oberhalb von 1 MB kann also als Kombination verschiedener Speicherarten genutzt werden; allerdings muß jeweils der betreffende Speicher-Manager aktiviert und imstande sein zu steuern, wieviel Speicher

wofür genutzt werden kann. Expanded Memory wird auch EMS-Speicher genannt, wenn er verfügbar und gemäß den hier beschriebenen Standardangaben konfiguriert ist.

High Memory Bereich

oder High Memory Area, abgekürzt HMA, sind die ersten 64 KB des Extended Memory, also von 1.024 bis 1.088 KB. Mit einem kleinen Trick wird dieser Bereich so verfügbar, als gehöre er zu dem ersten MB, und kann von DOS genutzt werden. Viele Anwender lagern hier Teile des DOS aus, indem sie **DOS=HIGH** eingeben. Der Speicher-Manager für diesen Bereich ist die Datei **HIMEM.SYS**, da sie den gesamten Speicher oberhalb von 1 MB steuert. Die **HIMEM.SYS** steuert also sowohl den High Memory- als auch den Extended Memory-Bereich.

MEM/A ist meines Wissens das einzige Kommando in DOS 6.x, das Informationen über die Nutzung dieses Bereichs gibt.

(A) Die HMA ist scheinbar ein Teil des Extended Memory – aber auch nur scheinbar: Der Prozessor kann im Real Mode auf die HMA zugreifen, wohingegen er auf den Extended Memory-Bereich nur im Geschützten Modus (*Protected Mode*) Zugriff hat.

Das Verhältnis von Upper zu Extended Memory

Der Upper Memory-Bereich leiht sich den eigentlichen RAM vom Extended Memory.

Umgangssprachlich sagt man, daß das Extended Memory in den Adressbereich des Upper Memory *eingetragen* (engl. *map*) wird. Der CPU wird vorgegaukelt, daß eine bestimmte Adresse im Speicher an einer bestimmten Stelle steht. In Wirklichkeit stimmt das nicht, wenn man vergleicht, wo der physische RAM steht. Die Summe Upper Memory + XMS=Konstant.

Device/Gerät

Alle diese Speicherarten, außer dem konventionellen Speicher, sind genauso wie Monitor, Tastatur, Drucker und so weiter Geräte (Device). Deshalb benötigen diese Speicherarten, ebenso wie alle anderen Geräte, einen Geräte-Treiber (Device-Driver).

Es gibt rein physische, und, wie ich das nenne, logische bzw. abstrakte Devices. Monitor und Tastatur als solche sind keine Devices, aber der logische Ausdruck Con oder Konsole bezeichnet ein Gerät, das sich auf die beiden physischen Objekte gemeinsam bezieht: ein Ein- und Ausgabegerät.

Startdateien

Der Computer wird entsprechend den in den Startdateien `CONFIG.SYS` und `AUTOEXEC.BAT` enthaltenen Informationen konfiguriert. In Bezug auf die Startdateien sind DOS 5 und 6 grundlegend identisch. Ich konzentriere mich auf DOS 5, gebe aber einige Hinweise auf DOS 6, zu dem ein eigener Abschnitt am Ende des Heftes folgt. Meine Beispiele wollen nur Vorschläge sein. Die Dateien müssen im Hauptverzeichnis auf dem Boot-Laufwerk stehen, typischerweise ist das `C:\`

Wenn du Programme installierst, schlagen diese oft vor, die Startdateien zu ändern. Daher ist es gut zu wissen, was die verschiedenen Zeilen dieser Dateien bedeuten.

Die Datei `CONFIG.SYS` enthält normalerweise Calls (Aufrufe) an Geräte-Treiber (*Device-Drivers*) – also Programme, die Geräte steuern.

Die Datei `AUTOEXEC.BAT` enthält DOS Befehle, die automatisch ablaufen, sobald der PC gestartet (gebootet) wird.

Für beide Startdateien gilt: Wenn am Zeilenbeginn `REM` bzw. `rem` steht, und es folgt ein Leerzeichen, wird diese Zeile als Bemerkung (*remark*) und nicht als Befehl erkannt.

Ob mit großen oder kleinen Buchstaben geschrieben wird, spielt keine Rolle. Vergiß nicht, eine funktionierende BOOT-Diskette zur Hand zu haben, bevor du Änderungen an der `CONFIG.SYS` vornimmst. Ich habe Beispiele mit einer Zeichensatztablette (Codepage) ausgewählt, aber im Kommentar ist beschrieben, was zu tun ist, wenn du zwei Zeichensatztabellen brauchst.

Zeichensatztablette (*Codepage*) ist ein etwas ungewöhnlicher Begriff, der eine Sammlung von 256 verschiedenen Zeichen beschreibt, die du auf dem Bildschirm siehst. Jede Zeichensatztablette hat eine Nummer.

Die wichtigen Dateien

`HIMEM`, `EMM386`, `SMARTDRV`, `RAMDRIVE` und `MEM` sind wichtige Dateien (hier nur mit deren Vornamen angegeben) in DOS 5, Win 3.1, DOS 6 und Win 3.11. Die genannten Versionen von DOS und Windows sind absichtlich in dieser Reihenfolge aufgeführt und werden im Text in *chronologischer* Reihenfolge nach Erscheinungsdatum beschrieben. `SMARTDRV` hat eine Sonderstellung. Dieses Festplatten-Cache-Programm hat viele Änderungen durchgemacht, weshalb es ein Kapitel für sich bekommen hat.

DOS 5 (22-03-91)

`HIMEM.SYS`, `EMM386.EXE`, `SMARTDRV.SYS` und `MEM.EXE`. Diese Dateien befinden/befanden sich gewöhnlich im Verzeichnis `C:\DOS`. Es ist dabei wichtig zu wissen, daß `SMARTDRV.SYS` (Ver. 3) nur als Geräte-Treiber funktioniert und nur mit `CONFIG.SYS` installiert werden kann.

`MEM.EXE`, das den Gebrauch der verschiedenen Speicherformen anzeigen kann, ist zusammen mit dieser DOS-Version erschienen.

Windows 3.1 (10-03-92)

enthält eine neue und bessere Version (4.0) des `SMARTDRV` und war nun eine ausführbare Exe-Datei – `SMARTDRV.EXE`. Sie konnte jetzt von der `AUTOEXEC.BAT` ausgeführt werden. Win 3.1 führte darüberhinaus ein vorzügliches Programm ein – `MSD.EXE` – das das erste MB des Speichers erkennt. Alle diese Dateien stehen normalerweise in `C:\WINDOWS`.

DOS 6.0 (10-03-93)

enthält die gleichen, wesentlichen Dateien wie Windows 3.1. Die `EMM386.EXE` und `SMARTDRV.EXE` (Ver. 4.1) waren verbessert. Alle diese Dateien zur Abwechslung wieder in `C:\DOS`. Darauf solltest du achten, wenn du Windows 3.1 nach DOS 6 installierst und gleichzeitig die Windows 3.1 Setup-Option akzeptierst, die automatisch deine Startdateien ändert. In diesem Falle wirst du nicht die neueste DOS 6-Version benutzen können. Deshalb mußt du deine Startdateien ändern, damit die `EMM386.EXE` und so weiter von `C:\DOS` gelesen werden und nicht von `C:\WINDOWS`.

DOS 6.2 (30-09-93)

einzigste Änderung war `SMARTDRV.EXE` Ver. 5.0

Windows 3.11 (01-11-93)

(Windows for Workgroups), lieferte dieselben Versionen der wichtigen Dateien wie DOS 6.2.

Lege in deinen Startdateien die neuesten Versionen der wichtigen Dateien fest. Sobald du das gemacht hast, löschst du die alten.

Das spart nicht nur Platz auf der Festplatte; es trägt auch zur Vermeidung von Durcheinander bei. In den folgenden Beispielen habe ich für die betroffenen Verzeichnisse zwei Punkte `..` und nicht das normalerweise benutzte `[path]` verwendet.

Ich empfehle dir, die entsprechenden Erklärungen für jeden Teil der `CONFIG.SYS` zu lesen, bevor du mit dem Bearbeiten anfängst.

DOS 5 – SMARTDRV.SYS**CONFIG.SYS**

```
Device=\DOS\himem.sys
dos=high
Device=\DOS\emm386.exe noems
dos=umb
Devicehigh=\DOS\smartdrv.sys 512 256
Devicehigh=\DOS\display.sys con=(, ,2)
rem Devicehigh=\UTI\gmouse.sys
Devicehigh=\DOS\ansi.sys
rem Devicehigh=\DOS\setver.exe
country=049,850,\DOS\country.sys
files=30
rem fcbs=1
buffers=5
rem stacks=9,256
break=on
rem shell=\DOS\command.com \DOS /p /f /e:1024
```

AUTOEXEC.BAT

```
@echo off
path C:\BAT;C:\DOS;....
mode con cp prep=((850 437) \DOS\ega.cpi)
mode con cp sel=850
LH keyb.com gr,,\DOS\keyboard.sys
rem LH \DOS\doskey.com
LH \DOS\share.exe
rem LH \UTI\gmouse.com
set temp=\temp
set tmp=\temp
prompt $p$g
rem \DOS\emm386.exe auto
```

Windows 3.1x oder DOS 6 – SMARTDRV.EXE**CONFIG.SYS**

```

Device=..\himem.sys
dos=high
Device=..\emm386.exe noems
dos=umb
rem Devicehigh=..\cdmke.sys /d:mscd01
rem Devicehigh=..\ramdrive.sys 2048 /e
Devicehigh=\DOS\display.sys con=(,2)
rem Devicehigh=\UTI\gmouse.sys
Devicehigh=\DOS\ansi.sys
rem Devicehigh=\DOS\setver.exe
rem Device=..\smartdrv.exe /double_buffer
country=049,850,\DOS\country.sys
files=40
rem fcbs=1
buffers=5
rem stacks=9,256
break=on
rem shell=\DOS\command.com \DOS /p /f /e:1024

```

AUTOEXEC.BAT

```

@echo off
path C:\BAT;C:\DOS;C:\WINDOWS;C:\UTI;....
rem LH MSCDEX /D:MSCD01 /M:20
..\smartdrv.exe
mode con cp prep=((850 437) \DOS\ega.cpi)
mode con cp sel=850
LH \DOS\keyb.com gr,,\DOS\keyboard.sys
rem LH \DOS\doskey.com
rem LH \UTI\gmouse.com
LH \DOS\share.exe
set temp=\temp
set tmp=\temp
prompt $p$g
rem win :

```

Lesehinweis

Diese Startdateien sind in erster Linie für Einzel-PCs (*standalones*) gedacht. Ich habe keine Erfahrung mit Netzwerken. Aus typographischen Gründen sind lange Zeilen in den folgenden Erklärungen abgekürzt. Zeilen, die mit REM anfangen, sind optional – möglicherweise benötigst du sie gar nicht. Ich habe sie einbezogen, um ihre meiner Meinung nach korrekte Platzierung anzugeben.

Siehe die Kommentare, wo ich auch mögliche Zeilen angebe, die *nicht* in den Beispielen auftreten.

Der Start- oder Bootvorgang

(A) Während des Bootvorgangs laufen unter anderem folgende Aktionen ab: zunächst wird die Hardware (Speicher, Ausgänge, Grafikkarten etc.) von einem ROM-Chip überprüft und die Informationen aus dem ROM BIOS geladen; dann startet ein Programm. Zuerst wird nach den Systemdateien IO.SYS und MSDOS.SYS gesucht.

An dieser Stelle des Boot-Vorganges weiß der Computer nicht, welche Geräte (Devices) installiert sind. Jedes Gerät benötigt ein Programm (Software), von dem es gesteuert wird. Nur wenn dieses Programm läuft, kann das Gerät sozusagen ein "Teil der Familie" werden das heißt aller Teile deines Computers, die in der Lage sind, miteinander zu kommunizieren. Wenn diese Prozesse erfolgreich abgeschlossen sind, ist das DOS Betriebssystem bereit.

DOS 5: Wenn es bis zu dieser Stelle keine Probleme gab, kann der PC während des Bootens nicht blockieren, egal welche Zeilen die AUTOEXEC. BAT enthält. Deshalb ist es wichtig, zumindest eine CONFIG.SYS zu haben, die keine fatalen Fehler verursacht. Sobald die CONFIG.SYS vollständig geladen ist, ist der Bootvorgang gelungen.

DOS 6: hier kann man (beinahe) die CONFIG. SYS überspringen, indem man zu Beginn des Bootens F5 drückt – genaueres dazu im Abschnitt DOS 6 (F5).

Zu Beginn des Bootvorganges ist der konventionelle Speicher leer. Zuerst werden systemtechnische Angaben vom ROM geladen. Danach läuft CONFIG.SYS zweimal durch. Beim ersten Durchlauf wird die Zeile DOS=HIGH und anscheinend auch die Zeile DOS=UMB gesucht – also ist es gleichgültig, wo diese beiden Zeilen stehen. Wird beim zweiten Durchlauf die HIMEM.SYS gefunden, wird zuerst sie in den konventionellen Speicher gelesen; anschließend wird eine eventuelle EMM386 und danach DOS in das High Memory geladen. Danach kommt je nachdem DOS=UMB, und zuletzt lädt der RAM-Speicher die Treiber und andere Programme.

CONFIG.SYS

Abgesehen von den Zeilen Device= werden die Zeilen *ohne Rücksicht auf ihre Platzierung* in der folgenden Reihenfolge gelesen:

Zunächst wird die ganze Gruppe der Device (high) = Zeilen in der Reihenfolge ihres Erscheinens geladen. Diese Reihenfolge ist also wichtig für eine sinnvolle Nutzung des Upper Memory-Bereiches! Dann werden FILES= FCBS= BUFFERS= LASTDRIVE= und STACKS= eingelesen – auch wenn diese Zeilen nicht existieren! Zuletzt wird COMMAND.COM geladen, das die DOS-Befehle übersetzt, evt. durch die Zeile SHELL= .

Zum Glück können Treiber auf viele verschiedene Weisen arbeiten, das heißt ein Gerät (*Device*) in unterschiedliche Funktionsweisen versetzen. Wieviel von deinem RAM-Speicher du für verschiedene Zwecke benutzen möchtest, das entscheidest allein du. In der jeweiligen Zeile bestimmst du einen oder mehrere Parameter, eine Art Variabel, und außerdem je nach Lage sogenannte "Schalter" (*switches*) . Ein Schalter kann entweder ON oder OFF sein. Ein Parameter bezieht sich in der Regel auf das Objekt der Handlung – in diesem Fall ein Gerät (*Device*). Ein Parameter kann nur einen von verschiedenen Werten bestimmen, wobei es sich nicht unbedingt um eine Zahl handeln muß. Der Parameter *Wochentag* beispielsweise kann nur auf einen von sieben möglichen Werten eingestellt sein.

Das hat seine Vor- und Nachteile. Wenn du mit allen verschiedenen Kombinationen und ihren Beziehungen zueinander umgehen kannst, ist das problemlos. Aber wie viele sicher wissen, ist das oft die reine Hölle. Dieser Abschnitt hilft dir hoffentlich, dieses Thema zu verstehen.

DEVICE=\. . \HIMEM. SYS

HIMEM.SYS muß vor EMM386.EXE stehen.

HIMEM.SYS ist der Speicher-Manager sowohl für den High Memory als auch für den Extended Memory Bereich.

(A) High und Extended Memory haben *erst jetzt* einen Zustand erreicht, der sie verwendbar macht. Sowohl High als auch Extended Memory haben nun einen Speicher-Manager. Das bedeutet, daß du nun Zugriff auf den High und den Extended Memory-Bereich hast. Bei dieser Festlegung wird der gesamte Speicher oberhalb von 1088 KB so konfiguriert, daß er als XMS Speicher verfügbar (*available XMS Memory*) ist, falls DOS=HIGH benutzt wird.

DOS=HIGH

Voraussetzung hierfür ist, daß die [HIMEM.SYS](#) aktiv ist. DOS=HIGH bedeutet, daß DOS sich selbst soweit nur möglich im High Memory positioniert. Dies ist eine der wichtigsten Eigenschaften von DOS 5 – durch diesen Vorgang schafft es mehr konventionellen Speicherplatz, um andere Programme laufen zu lassen. Der High Memory *kann* auch für andere Dinge verwendet werden.

286

Hast du einen 286er Rechner mit mindestens 1 MB RAM, kannst du vielleicht das High und/oder das Upper Memory benutzen. Um den Upper Memory-Bereich nutzen zu können, braucht man zunächst eine spezielle Hardware – die nur einige 286er haben – und dann einen Upper Memory-Manager. Leider habe ich keine Erfahrung mit dem Upper Memory-Manager eines 286er. Du kannst aber auch selbst herausfinden, ob dein PC den High Memory-Bereich nutzen kann. Füge die zwei folgenden Zeilen am Anfang deiner [CONFIG.SYS](#) ein:

```
Device=...\Himem.sys
DOS=HIGH
```

und schaue, welche Mitteilungen er beim Booten gibt. Geht es dir zu schnell, sogibst du unmittelbar nach dem Startvorgang MEM ein. Die letzte Zeile sollte so lauten:

```
MS-DOS resident in high Memory Area
```

Ist dies nicht der Fall, kommt die Nachricht:

```
unable to control A20 line
```

Versuche dich in diesem Fall mit ...Himem.sys /machine:11

Die Zahl ist eine Maschinenidentifikation. Zugelassene Werte sind 1-14 (schaue für deinen PC im DOS Handbuch), oder probier es aus, indem du z.B. 11, 12 oder 13 eingibst. Auf manchen Geräten habe ich es ans Laufen gebracht.

Wenn es klappt, hast du DOS im High Memory Bereich untergebracht, worauf dein PC schneller wird, da er jetzt mehr konventioneller Speicher zur Verfügung hat. Benutze MEM oder MEM/C/P zur Überprüfung. Du solltest jetzt mehr Speicherplatz für Programme zur Verfügung haben. Kannst du den Upper Memory Bereich nicht nutzen, mußt du Device= statt Devicehigh= schreiben und alle Verweise auf LH in der [AUTOEXEC.BAT](#) entfernen.

386

```
.. EMM386.EXE ..
```

[EMM386.EXE](#) kann *nur* auf einem PC benutzt werden, der mindestens einen 386er Prozessor hat, und ist eine Wissenschaft für sich mit zahllosen Möglichkeiten. Daß wir uns diese Datei etwas genauer ansehen, liegt daran, daß sie, was die sinnvolle Verwendung des Memory angeht, eine große Rolle spielt. Ich werde allerdings nicht genau auf alle möglichen Parameter eingehen.

Dieses Programm hat zwei Grundfunktionen. Es kann der Speicher-Manager für einen eventuellen EMS-Speicher sein, und es kann Zugang zum Upper Memory-Bereich *schaffen*.

[EMM386.EXE](#) kann sowohl als *Treiber* als auch als *DOS Program* benutzt werden, woran wir leider noch nicht gewöhnt sind ... Es ist ungewöhnlich, daß ein Geräte-Treiber (*DeviceDriver*) gleichzeitig auch als Programm von der DOS Eingabeebene aus laufen kann.

Bevor diese Zeile beim Booten gelesen wird, ist der gesamte RAM-Speicher oberhalb von 1.088 KB als Extended Memory konfiguriert. [HIMEM.SYS](#) ist erst einmal Speicher-Manager für den Upper Memory. Wenn aber DOS=UMB angegeben ist, übergibt [HIMEM.SYS](#) die Kontrolle über das Upper Memory an DOS.

Erst *nachdem* HIMEM.SYS und EMM386.EXE geladen sind, können Programme über Devicehigh und loadhigh in den Upper Memory geladen werden.

LH ist eine Abkürzung für LoadHigh, das heißt "Lade in das Upper Memory." LoadHigh und Devicehigh hätten eigentlich LoadUpper bzw. DeviceUpper heißen sollen – die Begriffsverwirrung scheint vollkommen! Versuchst du ein Programm oder einen Treiber mit Devicehigh oder LH in den Upper Memory zu laden, wenn dort *kein* Platz mehr ist, so wird es statt dessen in den konventionellen Speicher geladen – ohne daß du eine Fehlermeldung bekommst.

```
Device= .. EMM386.EXE noems
```

Ich fange mit dem Setup an, das für die meisten Anwender wichtig ist. Dies ist der Parameter NOEMS. Er bedeutet, daß ein Teil des Extended Memory *nicht* in einen EMS Speicher konvertiert werden soll. NOEMS bedeutet auch "Schaffe Zugang zum Upper Memory". Du benutzt dies, wenn du für kein Programm einen EMS Speicher benötigst, den Upper Memory benutzen möchtest und deinen gesamten freien RAM-Speicher oberhalb von 1.088 KB als XMS Speicher nutzen willst.

... I=E000-EFFF

(A). Betrifft nicht das PS/2 System. Diese Zeile ist ein *Parameter*, der unabhängig von anderen Parametern ist und in der gleichen Zeile wie [EMM386.EXE](#) stehen kann. Er schließt einen Adressbereich im Upper Memory ein und ist von dem Bereich abhängig, der nicht bereits von etwas anderem genutzt wird (von etwas anderem heißt in diesem Zusammenhang: einer Systemkomponente, wie zum Beispiel dem BIOS der Mutterplatine.)

Ein Speicherbereich wird von einer Anfangs- und einer Endadresse definiert. In der Regel wird das Sechzehner-Zahlensystem benutzt, und in diesem Zusammenhang jeweils 4 Ziffern (die letzte 0 fällt weg). Beispielsweise werden die 64 KB in dem Bereich von 896-960 KB zu E000-EFFF. EFFF ist die Adresse genau vor F000. Ein Bereich von 64 KB wird jeweils als A-Seite, B-Seite und so weiter bezeichnet. 640 KB = A000 etc.

[EMM386](#) im DOS 5 schließt im Gegensatz zu [EMM386](#) in DOS 6 diesen Bereich *nicht* standardgemäß ein, da er vom PS/2 für das BIOS-ROM benutzt wird.

Wenn du – um sicher zu gehen, daß er von keinem Programm benutzt wird – einen Bereich aus dem Upper Memory *ausschließen* möchtest, kannst du das auf die gleiche Art festlegen, indem du ein X statt einem I eingibst; zum Beispiel

... X=B000-B7FF

Dadurch wird dieser Bereich, der monochromer Bereich (*Monochrome Region*) genannt wird, ausgeschlossen. Im monochromen Bereich kommuniziert die CPU mit einer monochromen (schwarz/weiß) Grafikkarte. Lies dazu die Kommentare unter [MEMMAKER S. 41](#). Es ist sinnvoller, diesen Bereich *einzuschließen* als ihn *auszuschließen*.

Verlangt EMS-Speicher

Einige Programme benötigen EMS-Speicher. Um EMS und Upper Memory besser verstehen zu können, kann man sagen, daß die meisten von EMS abhängigen Programme verlangen, daß 64 KB *zusammenhängende* UMB benutzt werden, um einen sogenannten Seitenrahmen (*Page Frame*) zu erstellen.

Zum Thema EMS-Memory wollen wir zwei Parameter detailliert diskutieren, nämlich RAM und AUTO, da das Handbuch sie nur kurz erwähnt. Und an der Stelle, wo es um den AUTO -"Schalter" geht, sind die Angaben obendrein verkehrt. HELP DOS 6 ist auch falsch. Schau unter AUTOEXEC.BAT nach, wohin AUTO gehört!

Nehmen wir an, du hast insgesamt 4 MB RAM. Nachdem du [HIMEM.SYS](#) installiert hast, verfügst du über 3 MB Extended Memory. Ich lasse die 64 KB im High Memory für einen Moment außer

Betracht, um die Sache ein bißchen einfacher zu gestalten. [EMM386.EXE](#) ist in der Lage, Extended Memory in Expanded Memory umzuwandeln. Wenn du also 1MB=1.024 KB deiner 3 MB XMS als EMS benutzt (und die restlichen 2 MB als XMS behältst) und Zugang zum Upper Memory haben möchtest, schreibst du

... RAM 1024

RAM bedeutet 'Verschaffe Zugang zum Upper Memory'. 1024 bedeutet 'Benutze max. 1.024 KB für den EMS Speicher'. Wenn du nur eine Zahl eingibst, bekommst du *keinen* Zugang zum Upper Memory.

... I=E000-EFFF FRAME=E000

(A). Siehe auch das vorgenannte I=E000-F000. Betrifft nicht PS/2. Gemäß der LIM EMS Spezifikation Version 3.2 oder 4.0 muß der EMS Speicher mit sogenannten Seitenrahmen (*page frame*) arbeiten, die hier nicht ausführlicher beschrieben werden. Aber die meisten Programme, die EMS benötigen, verlangen einen Seitenrahmen – ein "Fenster" im Upper Memory – der auf einen Teil des EMS Speichers hinweist. Es ist sehr wichtig, daß dieses 64KB-Fenster im Upper Memory verfügbar ist – andernfalls verschwindet ein ordentlicher Happen aus dem konventionellen Speicher.

Leider wählt [EMM386.EXE](#) im DOS 5 standardgemäß die Startadresse D000 für einen Seitenrahmen (das IBM System PS/2 startet sein Motherboard BIOS ROM bei dieser Adresse). Wenn du beispielsweise EMS mit den Parametern RAM 1024 installierst, siehst du wahrscheinlich beim Booten den Hinweis auf die Startadresse nach *starting at address.. (beginnt bei Adresse ..)*.

Der Parameter FRAME=E000 bestimmt die Startadresse für den Seitenrahmen (*page frame*); das bedeutet, daß es nicht der [EMM386](#) überlassen wird, die Startadresse zu finden. Das obengenannte Beispiel nutzt den Upper Memory besser.

[EMM386](#) im DOS 6 ist verbessert worden. Der [MEMMAKER](#) versucht anscheinend zu sehen, ob E000-EFFF (die sogenannte Epage) frei ist. Ist dies der Fall, wählt er E000 als Startadresse aus – was nicht schlecht ist.

... FRAME=NONE

(A) Vielleicht benötigt dasjenige deiner Programme, das den EMS-Speicher benutzen könnte, keinen Seitenrahmen (*page frame*). Das ist aber nur bei sehr wenigen Programmen der Fall.

Verhält es sich aber so, und du möchtest die 64 KB des Upper Memory lieber anderen Programmen zuweisen, kannst du diesen Parameter festlegen. Falls du erweiterte (*extended*) DOS-Programme benutzt, das heißt Programme, die den Speicher

über die normalen DOS Grenzen hinaus selbst erweitern, wie zum Beispiel AutoCAD386 oder Lotus 1-2-3 Version 3.x, solltest du anhand deines Handbuches überprüfen, ob es sinnvoll ist, diesen Parameter festzulegen.

FRAME=NONE wird in DOS 5 nicht beschrieben, während DOS 6 es mit dem Hinweis erwähnt, daß einige Programme dadurch gegebenenfalls nicht einwandfrei arbeiten: Im Rahmen zeige ich einige Beispiele für Zeilen mit EMM386.

DOS 6: Siehe EMM386, S. 41.
Genug über EMS Memory und EMM386.EXE

DOS=UMB

bedeutet, daß DOS die Handhabung des gesamten Upper Memory übernimmt. DOS verlangt von der HIMEM.SYS die Übergabe der Kontrolle des gesamten Upper Memory. Du *kannst* die beiden Zeilen auch als eine schreiben:

```
DOS=HIGH und DOS=UMB als DOS=HIGH, UMB
```

Da es sich um zwei verschiedene Kommandos dreht, habe ich sie im Beispiel in jeweils eine Zeile für sich geschrieben.

REM DE ..IGH=\..\CDMKE.SYS /D:MSCD01

Dies ist ein Treiber für ein Panasonic CD-ROM-Laufwerk, den ich nur aktiviere, wenn ich ihn brauche – er beansprucht 11 KB. Der Name nach dem /D: macht es möglich, daß MSCDEX – das von AUTOEXEC.BAT aufgerufen wird – das CD-ROM Laufwerk erkennt. Siehe MSCDEX, S. 22.

..SMARTDRV.SYS 512 256

Dies betrifft nur diejenigen, die ausschließlich das nicht so gute Cacheprogramm SMARTDRV.SYS des DOS 5 haben. Falls du nicht weißt, was ein Platten-cache ist, lies zuerst die *allgemeine* Beschreibung in "Der SMARTDRV Platten Cache" auf Seite 25 und kehre danach hierher zurück.

Die Zahlen stehen für die Menge an KB für die Anfangs- und Mindestgröße des Cache. Ob du nur die erste Zahl, beide Zahlen oder überhaupt keine angeben möchtest, liegt an dir. Die optimale Zahl hängt von der verfügbaren Größe des Extended Memory ab. Der Standardwert (Default) für die erste Zahl ist 256. Die letzte Zahl ist für Windows relevant. Bestimmte Programme haben die Fähigkeit, die Mindestgröße zu verändern und ihr sogar den Wert 0 zu geben, damit sie diesen

Speicherplatz selbst nutzen können.

Um dies zu vermeiden, mußt du das Minimum angeben, indem du die zweite Zahl einsetzt. Ich nehme an, daß du keinen Expanded Memory installiert hast. Vorstehend findest du ein Beispiel mit 2 MB RAM und NOEMS bei EMM386.EXE. Anders ausgedrückt kannst du über 960 KB XMS-Speicher verfügen. Wenn du für SMARTDRV 512 KB verbraucht hast, bleiben dir 448 KB für Programme, die Extended Memory benutzen können. Weißt du, daß das bei deinen Programmen nicht der Fall ist, dann erhöhst du die erste Zahl und schaust, ob dein System jetzt schneller wird.

Hast du 4 MB RAM, schreibst du 1024 512. Es ist zwar nicht notwendig, einen Platten-cache zu verwenden – es empfiehlt sich aber.

SMARTDRV wurde erst durch die Version 4.0 in Windows 3.1 annehmbar und durch Version 4.1 in DOS 6 wirklich gut. Hast du SMARTDRV.EXE, bist du besser dran. Es sollte nicht aus der CONFIG.SYS, sondern aus der AUTOEXEC.BAT geladen. Auch wenn du nur DOS 5 hast, kannst du SMARTDRV.EXE 4.0 benutzen; vgl. Seite 25.

..RAMDRIVE.SYS 2048 /E

ist wichtig, wenn du mindestens 6 MB RAM hast und Programme benutzt, die sehr intensiv mit Temporärdateien arbeiten, was viele Windows-Programme tun.

Die Datei erstellt eine sogenannte RAM-Disk – die IBM allerdings ein *virtuelles Laufwerk* nennt. Beim Booten siehst du am Ende einer virtuellen Plattenzeile (virtual disk) den nächsten verfügbaren Laufwerk-Buchstaben. Die Zahl gibt an, wieviel RAM beiseite gestellt wird, um sich wie eine Festplatte zu verhalten. Hier sind es 2048 KB Extended Memory – gekennzeichnet durch das /e.

Wenn du eventuell eine Karte mit Expanded Memory hast und diesen Speicher nutzen möchtest, gibst du /a ein.

Bei einem Stromausfall oder einem System-"Hänger" riskierst du natürlich, daß alle in der RAM-Disk enthaltenen Daten verloren sind. Der Vorteil ist, daß man, wenn man viel mit Temporärdateien zu tun hat, im RAM schnelleren Zugriff auf seine Daten hat, da man nicht auf die Festplatte zugreifen muß. Bei meinen Programmen habe ich von der RAM-Disk keinen Nutzen; das soll dich aber nicht davon abhalten, es zu versuchen.

```
Device=\..\emm386.exe ram 1024
Device=\..\emm386.exe i=e000-efff frame=e000 ram 2048
Device=\..\emm386.exe i=e000-efff x=b000-b7ff noems
Device=\..\emm386.exe i=e000-efff noems
```

..DISPLAY.SYS CON=(, ,2)

ist der Treiber für die Konsole (so wird die Kombination von Bildschirm und Tastatur genannt). Der *erste* Parameter, der sich auf den Typ der Grafikkarte bezieht, *muß* nicht angegeben werden, da **DISPLAY.SYS** die Grafikkarte automatisch überprüft.

Der *zweite* Parameter bezieht sich auf die Nummer der Zeichensatztablelle (Codepage) für die Konsole, die normalerweise die Standardeinstellung für die USA hat. Nach meiner Erfahrung *muß* dies hier nicht angegeben werden, wenn du `mode con cp prep` und `mode con cp select` in der **AUTOEXEC.BAT** benutzt, aus der diese Zeichensatztablelle ausgewählt wird. Unterstützt deine Hardware, das heißt deine Grafikkarte, nicht 850, dann *muß* du eine Nummer angeben. Um es dir genauer anzuschauen, gibst du im DOS folgendes ein:

```
c:\>mode con
```

Der *dritte* Parameter ist die Anzahl der Zeichensatztablellen, für die Platz im Speicher reserviert werden *muß* – es geht noch immer nur um CON. Der Standard ist 1. Wenn du immer nur eine Zeichensatztablelle benutzen möchtest, zum Beispiel 850, dann schreibst du eine 1 am Ende.

Dies ist alles notwendig, um `MODE CON CP prep` und `select` in der **AUTOEXEC.BAT** benutzen zu können.

Hast du zwei Zeichensatztablellen, ermöglicht die Zahl 2 am Ende dir den Wechsel zwischen etwa Zeichensatztablelle 850 und 437. Bedingung hierfür ist, daß du zwei Zeichensatztablellen vorbereitest (siehe `mode con...`) und daß **NLSFUNC** in der **AUTOEXEC.BAT** eingegeben wird; dann kannst du mit dem Befehl `CHCP 437` oder `CHCP 850` umschalten.

Im Kommentar unter **BUFFERS** auf Seite 20 erfährst du, wieviele Buffer in den High Memory geladen werden können.

..SETVER.EXE

Einige Programme, meist ältere, kommunizieren mit der DOS-Versionsnummer und erwarten diese Nummer im Speicher. Dafür sorgt **SETVER**, indem es dem Programm fiktive Versionsnummern angibt. Schreibe **SETVER**, um nachzuschauen. Benötigt 432 Bytes.

..GMOUSE.SYS

lädt einen Maustreiber für DOS Programme. Windows hat seinen eigenen Treiber, der aber nur für Windows-Programme benutzt werden kann. Unser Beispiel gilt für eine Genius-Maus. Dein

Maustreiber kann eine andere Bezeichnung haben und in einem anderen Verzeichnis stehen.

..ANSI.SYS

ist der Treiber für Bildschirmzeichen, Cursorbewegungen und die Tastaturbelegung. Notwendig, wenn du im DOS eine andere als die Standardeinstellung von 25 Zeilen mit 80 Zeichen haben möchtest, die Farben des DOS Bildschirms ändern willst u.a.m. Benötigt ca. 4 KB.

```
rem ..smartdrv.exe /double_buffer
```

Siehe Seite 26

COUNTRY=049,850,..

In diesem Beispiel habe ich einen Anwender ausgewählt, der die Zeichensatztablelle 850 als die aktive Zeichensatztablelle benutzen will.

Die erste Zahl, in diesem Falle 049, stellt das deutsche Format für Zeit, Datum, Währungssymbol, Sortierkriterien der Dateien und Verzeichnis- und Dateinamen-Symbole ein. Die zweite Zahl gibt die aktive Zeichensatztablelle an – die allgemeine, aktive Zeichensatztablelle für alle Geräte, die von ihr unterstützt werden.

Gibst du den zweiten Parameter nicht an, wird bei einem deutschen Computer während des Bootens automatisch die Zeichensatztablelle 850 gewählt – Standardeinstellung für Deutschland ist nämlich die 850, während die Alternative 437 heißt. Falls du später die 437 benutzen möchtest, müssen `MODE CON CP prep (850 437)` und **NLSFUNC** in der **AUTOEXEC.BAT** geladen werden. Anschließend wählst du die Zeichensatztablelle mit `CHCP 850` aus.

FILES=40

Die Standardeinstellung (falls die Zeile nicht vorhanden ist) ist 8. Gültig sind 8–255. Legt fest, wieviele Dateien gleichzeitig geöffnet sein können. Eine Datei ist "geöffnet", wenn sie benutzt oder gelesen wird. Viele neueren Programme haben mehrere Dateien gleichzeitig geöffnet. Falls du Windows benutzt und mehrere Programme nebeneinander laufen läßt, *muß* du eine höhere Zahl eingeben.

Dafür wird etwas Platz im Speicher benötigt – aber nicht allzuviel. Probier's mit 30, 40 oder 50 aus und schau dir an, wie es funktioniert. Größere Datenbankprogramme mit vielen Referenzdatenbanken arbeiten oft an vielen Dateien gleichzeitig. Ich frage mich, warum Microsoft kein Programm zur Verfügung stellt, das die Anzahl offener Dateien angibt.

REM FCBS=X

Dateikontrollblocks (*File Control Blocks*) ist eine veraltete Art, Zugang zu Dateien zu bekommen. Heute wird `FILES=` benutzt. Bestimmte ältere Programme wie zum Beispiel SideKick wenden die ältere Methode an. `x` gibt an, wieviele dieser Blocks in DOS gleichzeitig offen sein können. Die Standardeinstellung ist 4.

Weißt du nicht genau, ob du Programme hast, die FCBS verlangen, kannst du versuchen eine 1 anzugeben. Erfolgt darauf keine Fehlermeldung, dann hast du ein wenig Speicherplatz gespart.

BUFFERS=5

ist ein Mini-Festplatten-Cache-Programm (vgl. S. 25 über Festplatten-Cache). Wenn du mit dem Programm `SMARTDRV` arbeitest, empfiehlt es sich, die Anzahl der Puffer (*buffer*) niedrig einzustellen, zum Beispiel auf 6. Jeder Puffer benötigt ungefähr 532 Bytes. Benutzt du kein Festplatten-Cache-Programm, kannst du ein wenig experimentieren.

Versuche es mit Werten zwischen 6 und 20 und schau, bei welchem Wert dein PC am besten läuft. Du erhältst keine Fehlermeldung, sondern einen schnelleren oder langsameren PC.

Ist dort Platz, sollte laut Microsoft der größte Teil der Puffer in das High Memory gelesen werden. Ich stellte allerdings über die Befehle `MEM/C/P` und `MEM/D/P` fest, daß dies nicht der Fall war. Später fand ich den Grund: wird `DISPLAY.SYS` eingelesen – was bei allen nicht-englischsprachigen PCs der Fall ist, da sonst die landespezifischen Buchstaben wie β nicht benutzt werden könnten – läuft irgend etwas im High Memory ab. Ich vermute, daß im Speicher Platz für die Zeichensatztabellen von `DISPLAY.SYS` reserviert wird, die einen Teil des High Memory in Anspruch nehmen.

Mein Test zeigte: wird `DISPLAY.SYS` eingelesen und der letzte Parameter auf 1 gestellt (Anzahl der Zeichensatztabellen), ergibt sich folgendes: bis zu 27 Puffer gehen gut in den High Memory, und nur 512 Bytes werden im konventionellen Speicher besetzt. Bei 28 Puffern werden alle Puffer in den konventionellen Speicher geladen. Eine weitere Drolligkeit: Bei `buffers=24-27` belegt

`COMMAND.COM` ganze 5K, ansonsten nur 3K. Das optimale bei einer Zeichensatztabelle ist `buffers=23`.

Wird der letzte Parameter in `DISPLAY.SYS` auf 2 gesetzt, werden bis zu 8 Puffer im High Memory-Bereich plaziert. Bei 9 buffers geht es schief. Bei `buffers=6-8` belegt `COMMAND.COM` 5K, ansonsten nur 3K. Das optimale bei zwei Zeichensatztabellen ist `buffers=5`. Zum Glück brauchen nur die wenigsten viele Puffer; die meisten benutzen sicher ein Festplatten-Cache wie zum Beispiel `SMARTDRV`. Es ist aber ein Fehler, daß das Handbuch und Help

darauf nicht aufmerksam machen. Den Amerikanern fällt das bestimmt nicht auf, da sie keine Anwendung für `DISPLAY.SYS` haben. (IBM's PC DOS 6.1 und 6.3 haben das gleiche Problem)

Willst du die Sache nachprüfen? Nach jedem Booten führst du `MEM/D/P` aus und siehst nach `BUFFERS`. Du kannst auch den undokumentierten Befehl `MEM/A` eingeben, der dir Informationen über das High Memory gibt.

REM STACKS=9,256

Die Zahlen sind nur ein Beispiel. Im Default-Falle, also wenn es die Zeile nicht gibt, ist die Zahl 0,0 für den Original-IBM-PC und 9,128 für alle anderen. Dieser Parameter legt fest, wieviel Speicherplatz beiseite gelegt wird, um sogenannte Hardware-Unterbrechungen (*hardware interrupts*) zu handhaben.

Eine Unterbrechung (*Interrupt*) tritt ein, wenn sozusagen ein Anruf für Herrn DOS kommt, während "er" mit irgendetwas beschäftigt ist. Wenn du zum Beispiel eine Taste drückst, während DOS – und damit der Prozessor – mitten in einem Arbeitsgang ist, wird dieser Tastendruck als "Anruf"/Unterbrechung interpretiert, der nicht ignoriert werden kann. Also wird DOS das, was es gerade ausführt, vorübergehend in ein kleines Fach in einem sogenannten Stack (einem kleinen Puffer) auslagern, während es sich um die Unterbrechung kümmert. Nach Bewältigung der Unterbrechung kehrt DOS zum ersten Arbeitsgang zurück.

Die erste Zahl gibt die Anzahl der Stacks an. Gültige Werte sind hier 0 bzw. 8 bis 64. Die zweite Zahl gibt an, wieviele Bytes für jeden Stack reserviert werden. Gültige Zahlen sind in diesem Fall 0, 32, 64, 128, 256 oder 512. Hast du zusätzliche Karten für Scanner, Faxmodem, Modem oder ähnliches und bekommst die Fehlermeldung *stack overflow*, kannst du versuchsweise die Werte erhöhen. Vielleicht hilft es! Versuch's mit 9,256 – 9,512 – 10,128 und so weiter.

Hast du weder einen Original-IBM-PC noch diese Zeile, ist der Speicher an dieser Stelle erschöpft. Willst du das optimalste Setup haben, versuche es mit der Zeile 0,0 – es könnte ja sein, daß keines deiner Programme diese Stacks benötigt. Bekommst du Probleme, dann schreibe 9,128 – dies ist zwar das gleiche, als wenn du die Zeile weglassen würdest, falls du aber später einmal eine höhere Zahl brauchst, ist die Zeile schon vorhanden und du kannst sie einfacher ändern.

Dieser Parameter benötigt jedoch einigen konventionellen Speicherplatz. Wir reden dabei zwar von Peanuts, aber was sein muß, muß sein.

BREAK=ON

erhöht die Anzahl der Überprüfungen, die DOS durchführt, um zu sehen ob der Anwender die Tastenkombination **Strg+C** oder **Strg+Pause** gedrückt hat, um ein Programm zu verlassen. Viele Programme sind jedoch so geschrieben, daß diese Tasten keinen Effekt haben.

BREAK=ON bedeutet hier, daß du ein Programm mit **Strg+C** oder **Strg+Pause** verlassen kannst. Hat keine Auswirkung auf die RAM-Benutzung, kann aber die Geschwindigkeit herabsetzen, weil öfter auf *interrupts* überprüft wird.

```
shell=.. /p /f /e:1024
```

```
shell=..\dos\.. /p /f /e:1024
```

Einige Anwender plazieren die **COMMAND.COM** im Hauptverzeichnis und benötigen die Zeile mit **SHELL** nicht, was völlig in Ordnung ist. Andere wiederum plazieren die **COMMAND.COM** in C:\DOS zusammen mit der obenerwähnten Zeile, wo auch angegeben ist, wo sie zu finden ist.

Die erste Vorgehensweise ist die einfachste, da beim Booten standardmäßig dort nach der **COMMAND.COM** gesucht wird. Es ist sinnvoll, die **COMMAND.COM** in C:\DOS zu plazieren, da man selten im DOS-Verzeichnis arbeitet und somit nicht in die Gefahr gerät, **COMMAND.COM** aus Versehen zu löschen. Wird sie gelöscht, verlierst du deinen Befehlesinterpreter. Die Lösung: boote von einer Diskette und kopiere **COMMAND.COM** von hier auf die Festplatte. Jetzt kannst du wieder von der Festplatte aus booten.

/p bedeutet, daß du den permanenten Befehlsinterpreter festgelegt hast.

/f bedeutet, daß die Option **Fail** automatisch ausgewählt wird, wenn du die Fehlermeldung **Abort, Retry or Fail** erhältst. Diese Meldung erscheint, wenn ein Programm versucht, ein Diskettenlaufwerk zu lesen und keine Diskette findet.

/e:1024 – ein weiterer wichtiger Zweck dieses Befehls ist, das sogenannte DOS-Environment auf die am Ende der Zeile genannte Anzahl an Bytes zu erweitern; dieses Environment ist eine Art Anschlagtafel im Speicher, mit deren Hilfe die Programme und DOS Informationen austauschen oder lagern können.

Einige Programme verlangen, daß in dieser Umgebung eine Variable definiert ist. Während der Installation eines solchen Programmes, wird in der **AUTOEXEC.BAT** eine Zeile eingefügt bzw. empfohlen: zum Beispiel **SET LIB=C:\QB**. Wenn du **SET** und **HELP SET** auf der DOS-Eingabeebene eingibst, erhältst du eine Vorstellung davon, worum es sich handelt.

Wenn *keine* **SHELL=** -Zeile gefunden wird, oder wenn **/e** nicht angegeben wird, werden für DOS 5: 160 Bytes und für DOS 6: 256 Bytes reserviert.

Wenn du die Fehlermeldung *Out of environment space* bekommst, kannst du es mit dieser Zeile versuchen. Ich mußte die Bytes-Anzahl bisher noch nie erhöhen, aber Microsoft empfiehlt für Windows 3.1 mindestens 1024. Es dreht sich um wenige Bytes, also benutzt du ein kleines Stück vom konventionellen Speicher.

Hast du einen langen *path* (Pfad), bis zu 127 Zeichen, werden also 127 Bytes für den Pfad in Anspruch genommen. In diesem Fall lohnt es sich, diesen Wert zu definieren – die meisten sollten mit 512 oder 1024 auskommen.

DOS 6: **MEMMAKER** setzt diese Zeile ein, falls sie nicht vorhanden ist, jedoch ohne **/e:1024**

Weiteres in der CONFIG.SYS

Alles hat einmal ein Ende – sogar dieser Text!

REM LASTDRIVE=X

gibt an, zu wieviel Laufwerken du maximal Zugang hast. Die Standardeinstellung ist das nächste freie Laufwerk. Schreibst du **Z**, verschwendest du Speicherplatz.

Netz: Bestimmte Netzwerkprogramme möchten Laufwerke über das Maximum hinaus bezeichnen – in diesem Fall geht **Z** nicht.

FASTOPEN

Wenn in deiner **CONFIG.SYS** etwas mit **FASTOPEN** steht, empfehle ich dir, diese Zeile zu löschen. Die Idee, einen schnelleren Zugriff auf die Dateien zu ermöglichen, ist gut; aber etliche Anwender haben nach der Benutzung von zerstörten Dateien berichtet. Auch DOS 6 enthält diese Datei!

AUTOEXEC.BAT

ist eine Datei, die DOS-Befehle enthält. Das Besondere an dieser Datei ist, daß sie gelesen wird, wenn der Computer bootet – daher ihr Name: AUTOMATIC EXECUTION (automatische Ausführung). Allerdings muß sie dafür im Hauptverzeichnis des Laufwerkes stehen, von dem aus gebootet wird. Normalerweise ist das C:\. Die Datei ist optional – sie *muß* nicht vorhanden sein.

@ECHO OFF

Das erste Zeichen @ (auch Schnabel-A genannt) bewirkt, daß die Zeile (die Buchstaben) `echo off` nicht auf dem Bildschirm erscheint, aber der *eigentliche Befehl* ausgeführt wird.

Der Befehl `echo off` bewirkt, daß nachfolgende Befehle im `AUTOEXEC.BAT` während der Ausführung nicht auf dem Bildschirm erscheinen – *wohl aber ihr Ergebnis*. Du kannst das @, das du eventuell durch Eingabe von **Alt+64** auf dem Nummernblock deiner Tastatur schreibst, auch weglassen. Kommt dir diese Erklärung etwas kompliziert vor, dann schreibe an den Zeilenanfang `REM`, führe `AUTOEXEC.BAT` nochmals aus, und du wirst den Unterschied merken.

PATH C:\BAT;C:\DOS;....

Gib hier höchstens 127 Zeichen ein! Willst du die enthaltenen Verzeichnisse sehen, so gebe `PATH` ein. Viele Programme möchten während der Installation ein oder mehrere Verzeichnisse zur `AUTOEXEC.BAT` hinzufügen. Akzeptierst du das, wirst du möglicherweise eine zusätzliche Zeile mit % sehen. Sie bedeutet "Füge dieses Verzeichnis in den PATH ein". Berichtige das selber und lösche anschließend die Zeile mit %.

Wird `C:` ausgelassen, funktioniert der Pfad nur dann, wenn man sich auf dem `C:` Laufwerk befindet – was ziemlich unpraktisch ist ... In diesem Falle könntest du zum Beispiel nicht den Befehl `xcopy` ausführen, wenn du dich auf dem `A:` Laufwerk befändest.

rem LH MSCDEX /D:MSCD01 /M:20

Das Programm `MSCDEX` gibt Zugang zum CD-ROM-Laufwerk. Der Name nach `/D:` muß mit dem vom Treiber in der `CONFIG.SYS` angegebenen Namen identisch sein. Es ist sehr wichtig, daß diese Zeile *vor* der Zeile mit der `SMARTDRV` steht, so daß diese den Lesecache (*Read Cache*) auf dem CD-ROM-Laufwerk ausführen kann. Vgl. `SMARTDRV` s. 25. Während des Startes untersucht `SMARTDRV`, ob `MSCDEX` vorhanden ist; ist das so, stellt sie per Default den Lesecache für das CD-ROM-Laufwerk ein.

Wenn du sehen möchtest, wie der Speicher genutzt wird, kannst du `/V` in die Zeile eingeben; nun kannst du seine Nutzung während des Bootens verfolgen. `/M:20` gibt die Anzahl Sektor Puffer an.

\. . \SMARTDRV . EXE

enthält in diesem Beispiel keine Parameter für die Anfangs- und Mindestgröße. `SMARTDRV` überprüft den verfügbaren XMS Speicherplatz und wählt die Größe selbst aus. Für die meisten Anwender ist das in Ordnung. Um sicher zu gehen, daß `SMARTDRV` in das Upper Memory geladen wird, sollte sie zuoberst in der `AUTOEXEC.BAT` stehen, da sie bis zu 26 KB Speicherplatz belegt. Vgl. S. 25, *Laufwerk Cache Smartdrv*.

MODE CON CP PREP=(865) ..

MODE CON CP SEL=865

`MODE CON` ist eine Einstellung für die Konsole. `CP` ist die Abkürzung für *Codepage*, Zeichensatztafel. `Prep(are)` heißt *Vorbereiten*. Die Datei `EGA.CPI` enthält die Grafikzeichen, die auf dem Bildschirm erscheinen.

Ich habe Beispiele ausgesucht, die zwei Zeichensatztabellen zulassen. Dies ist nur möglich, wenn als erstes im Speicher Platz frei gelassen wurde; lies dazu `display.sys`. Die erste Zeile bereitet die Benutzung von zwei verschiedenen Zeichensatztabellen durch die Konsole vor. Die zweite Zeile wählt die Zeichensatztafel aus, die die Konsole benutzen soll. Das bedeutet in der Praxis: wie ASCII Werte auf dem Bildschirm dargestellt werden.

`C:\>MODE CON` gibt Informationen über:

- 1 die (optional auszuwählende) hardware-unterstützte Zeichensatztafel für `CON` – dies hängt vom zweiten Parameter der Zeile mit `display.sys` ab.
- 2 die vorbereitete(n) Zeichensatztafel(n) – wird kontrolliert von `MODE CON CP prep=..`
- 3 die ausgewählte Zeichensatztafel – wird kontrolliert von `MODE CON CP sel=..`

Hast du nur Platz für eine Zeichensatztafel in der Zeile mit `DISPLAY.SYS` reserviert, wählst du nur eine Zeichensatztafel aus beziehungsweise bereitest nur eine vor – 850 oder 437. Hast du zwei gewählt, schreibst du in der ersten Zeile `.. prep=(850 437) ..`

.KEYB.COM. .KEYBOARD.SYS

muß nach `mode con cp select` folgen. Sie installiert oder konfiguriert die Tastatur für das deutsche Layout. Da der zweite Parameter nicht angegeben ist, wird die Zeichensatztafel aktiviert,

die mit `mode con cp select=` ausgewählt wurde.

Ist der zweite Parameter explizit angegeben, muß er mit dem durch `select` ausgewählten Parameter identisch sein.

REM LH . . NLSFUNC.EXE

lädt ein Programm, das es ermöglicht, über den CHCP-Befehl zwischen den Zeichensatztabellen zu wechseln, z.B. `chcp 437` oder `chcp 850`. Nur relevant, wenn du die zwischen zwei Zeichensatztabellen wechseln möchtest. Vgl. `display.sys` unter [CONFIG.SYS](#) und `mode con...` unter [AUTOEXEC.BAT](#).

rem LH . . DOSKEY.COM

`DOSKEY` ist ein kluges kleines Programm, das sich, während du in Dos arbeitest, die letzten DOS-Befehle merkt. Über die Pfeiltasten – aufwärts oder abwärts – kannst du sie durchstöbern.

LH \DOS\SHARE

`SHARE.EXE` steuert, welche Dateien "offen" sind, so daß zwei Programme nicht *dieselbe* Datei öffnen, unterschiedliche Änderungen vornehmen und diese später lagern. Zwei Programme können also nicht gleichzeitig eine Datei teilen (*share*). Ich habe es mit verschiedenen Programmen versucht, aber nicht alle geben eine Warnung von sich. Du kannst dich also nicht bei allen Programmen darauf verlassen, daß sie `SHARE` benutzen können.

Es empfiehlt sich, `SHARE` zu benutzen, besonders wenn du mit Windows arbeitest. Diese Datei benötigt nur 6 KB. Sie eignet sich vor allem für WinWord 2.0. Für WinWord 6.0 schlägt Microsoft folgende Zeile vor:

```
LH \DOS\SHARE /L:500 /F:5100
```

`/L: 500` gibt die Anzahl von "Sperrern" (*locks*) an und definiert, wieviele gesperrte Dateien gleichzeitig gesteuert werden können. `/F: 5100` gibt in Bytes an, wieviel Speicherplatz benutzt werden soll, um sicherzustellen, daß zwei Programme nicht die gleiche Datei gleichzeitig benutzen.

Wenn du Windows 3.11 benutzt und DOS-Programme *immer* von Windows aus ausführst, brauchst du `SHARE` nicht.

```
set temp= \temp
```

```
set tmp = \temp
```

zwingt Programme wie Windows, das festgelegte Verzeichnis für Temporärdateien (TEMPorary) zu benutzen. Bestimmte Programme benutzen den

variablen Wert `temp`, andere `tmp`, deshalb die zwei Zeilen.

Es kann gelegentlich vorkommen, daß dein Computer "hängt", das heißt er läßt sich nicht mehr ansprechen, und du mußt ihn neu booten, um weiterzukommen. Wenn das geschieht und ein Programm nicht ordentlich verlassen wurde, können Temporärdateien übrigbleiben. Sie sind leicht zu finden, besonders wenn sie in diesem Verzeichnis stehen. Der Dateiname fängt gewöhnlich mit einem ~ Symbol an und führt oft die Dateiendung `TMP`. Lösche diese Dateien regelmäßig von deiner Festplatte – aber nur, wenn du dich in DOS befindest und keine anderen Programme laufen.

```
rem SET TEMP=D:\
```

```
rem SET TMP=D:\
```

Wenn du eine RAM-Disk (*RAMdisk*) in deiner `CONFIG.SYS` installiert hast und sie auch als eine Laufwerk (*disk*) für deine Temporärdateien benutzen möchtest, kannst du die RAM-Disk (in diesem Beispiel Laufwerk D) für diesen Zweck verwenden. Hat dein nächstes freies Laufwerk eine andere Bezeichnung, mußt du natürlich diese angeben. Beim Booten zeigt `RAMDRIVE.SYS` das ausgewählte Laufwerk an. Vergiß nicht, beide Zeilen einzugeben.

prompt \$p\$g

DOS 6 setzt diesen Prompt automatisch ein. Du brauchst ihn also nicht in deine `AUTOEXEC.BAT` zu schreiben. Fast jeder benutzt diesen Prompt, der dir Informationen darüber gibt, in welcher Platte und welchem Verzeichnis du dich befindest. Dieses `C:\>` wird Prompt genannt, weil der Computer damit anzeigt, daß er für eine (unverzögliche, prompte) Befehlseingabe bereit ist. Der `PROMPT` Befehl kann auch zur Neudefinierung von Tasten benutzt werden bzw. andere Bildschirmfarben einstellen. Für den nachfolgenden Prompt-Befehl mußt du, wie bereits beschrieben, `ANSY.SYS` aus deiner `CONFIG.SYS` installiert haben. Ist das der Fall, gib probenhalber folgendes ein:

```
C:\>prompt $e[0;1;37;44m$p$g
```

Falls du die eckige Klammer auf deiner Tastatur nicht hast oder findest, gibst du `Alt+91` ein. Dann gibst du `CLS` ein. Damit wird dein Bildschirm blau und die Schrift weiß. Wenn dir das besser gefällt, kannst du statt `PROMPT pg` die obenstehende Zeile in `AUTOEXEC.BAT` einfügen. Wenn du in DOS arbeitest, ist dann der ganze Bildschirm blau. Gefällt dir das nicht, kannst du entweder den PC neu booten oder `PROMPT $e[m$p$g` eingeben –

womit du zur 'normalen' prompt \$p\$g -
Einstellung zurückkehrst.

rem ..gmouse.com

Ich hatte eine Genius Maus mit dem Treiber
C:\UTIGMOUSE.COM. Wenn ich `rem` entfernte,
wurde `GMOUSE.COM` in das Upper Memory gela-
den. Im allgemeinen benutze ich die Maus nicht bei
DOS-Programmen, da ich finde, daß es hier mit der
Tastatur schneller geht.

Windows hat seine eigenen Maus-Treiber für
verschiedene Mäuse, die dann in *Windows*-Program-
men funktionieren. Wenn du eine Maus in DOS-
Programmen unter *Windows* benutzen möchtest,
muß der Maus-Treiber installiert werden, bevor du
Windows ausführst. Soll die Maus auch in der
sogenannten Fenster-Anzeige (nicht Vollbild-An-
zeige) funktionieren, muß der Treiber mindestens
die Version 8.20 von Microsoft oder kompatibel
sein. Außerdem muß du folgendes in die
`SYSTEM.INI`-Datei einfügen:

```
[NonWindowsApp]
MouseInDosBox=1
```

rem ..Emm386.exe auto

Dies betrifft DOS 5 nur, falls du Expanded Memory
installiert hast. Wenn `EMM386.EXE` als Programm
geladen ist, kann dieser Parameter durch einen
DOS-Befehl festgelegt werden. In der `CONFIG.SYS`
funktioniert er nicht. `AUTO` bedeutet, daß die in der
`CONFIG.SYS` als EMS-Speicher beiseite gelegte KB-
Menge nur als Speicher benutzt wird, wenn ein
Programm danach fragt. Andernfalls wird sie als
Extended Memory benutzt. Wurde sie allerdings
einmal als XMS-Speicher benutzt, dann kann sie
nicht automatisch zu EMS zurückwechseln.

REM WIN :

Viele Anwender benutzen die letzte Zeile, um ein
Programm, meist ein Menü oder *Windows*, zu
starten. Mit der Eingabe `WIN :`
überspringt man den Werbebildschirm, der
erscheint, wenn *Windows* startet.

set winpmt=Type EXIT and press

ENTER to return to Windows\$_\$_\$p\$g
ist eine gute Idee – muß aber in *einer* Zeile geschrie-
ben werden. `winpmt` ist die Abkürzung für 'Win-
dows Prompt'. Diese Zeile bedeutet, daß du, wenn
du dich in einem DOS Fenster befindest, daran
erinnert wirst, daß du ein DOS Fenster unter
Windows laufen hast. Wenn du vergißt, daß du dich
in *Windows* befindest und von deiner DOS Box aus
neu bootest, kann das schlimme Folgen haben.
\$_\$ entspricht dem Drücken der **Eingabe**.

Zeichensatztabellen/Codepages-Diskussion

Dies betrifft nur DOS Zeichensatztabellen.

Windows hat seine eigene Zeichensatztabelle, die im
Windows Setup definiert wird.

Ich habe viel Zeit darauf verwendet, dies und das
auszuprobieren, Handbücher zu lesen und während
des Bootens an verschiedenen Stellen in der
`AUTOEXEC.BAT` die Punkte `chcp`, `keyb` und
`mode con` einzugeben. Aber im Grunde kann ich
für viele Dinge, die die Zeichensatztabellen
betreffen, noch immer keine kristallklare Definition
geben. Ich habe auf verschiedenen PCs viele
verschiedene Versionen der relevanten Zeilen der
Startdateien gesehen. Die Mehrheit der Anwender
hat selten Probleme, selbst wenn es einen
Unterschied zwischen der aktiven `cp` (`CHCP`) und
der Zeichensatztabelle für die Konsole (`MODE CON`)
gibt – einige aber haben nun einmal Schwie-
rigkeiten. Daß die Informationen zu den
Zeichensatztabellen nicht jeweils in der
Landessprache gegeben werden, ist nicht gerade
hilfreich – und schlechter Kundendienst von
Microsoft. Das DOS 5 Handbuch und die
Bildschirm-Meldungen der oben genannten Befehle
– die irgendetwas mit `CP` zu tun haben – sind das
Widersprüchlichste, sowohl vom Wortlaut her als
auch in ihrer Bedeutung, das ich seit langem
gesehen habe. Das DOS 6 Handbuch ist etwas
besser, allerdings nicht an den Stellen, an denen es
wichtig ist.

Anscheinend hat Microsoft `CHCP` zum "König
aller Befehle" erkoren – dieser Befehl kann auf
einen Streich alle `CP`-Definitionen für *alle* Geräte
ändern. Nach meiner Erfahrung genügt es, den
zweiten Parameter im `country`-Befehl zu
benutzen.

Daß man mehrere recht komplizierte Zeilen in
bestimmter Reihenfolge in die Startdateien einfügen
muß, nur um eine Zeichensatztabelle zu definieren
bzw. auszutauschen, ist lächerlich. Ich habe vergeb-
lich versucht, von Microsoft eine Antwort auf die
Frage zu *codepages* zu bekommen.

Wie ich bereits erwähnt habe, besteht eventuell
ein Problem, wenn du den zweiten Parameter im
`country`-Befehl in der `CONFIG.SYS` nicht angibst.
Ich weiß, daß dies für die meisten Leser nicht
relevant ist; je mehr internationalen Datenaustausch
es gibt, desto wichtiger wird die Sache aber.

SMARTDrive Festplatten-Cache

Allgemeines

Es gibt verschiedene Laufwerkscache-Programme; wir wollen uns nur mit **SMARTDRV** befassen, die als Ergänzung zum DOS mitgeliefert wird. Ein Cache ist ein Puffer, eine Verbindung bzw. ein Zwischenspeicher zwischen der CPU und der Festplatte. `buffer=` in der `CONFIG.SYS` ist ein primitiver Cache.

SMARTDRV benutzt einen Teil des Extended Memory für seine Puffer. Wenn die CPU Daten erwartet, überprüft sie zunächst, ob die Daten im Cache stehen. In diesem Falle (bei einem Treffer oder *Hit*) werden die Daten von dort gelesen. Das kann bis zu 100mal schneller sein als bei einem Fehlschlag oder *Miss*, der bedeutet, daß die CPU sie sich von der Festplatte holen muß. Man nennt das "den Cache lesen"

Erhält die CPU zum Beispiel die Anweisung, ein Dokument zu speichern, speichert sie es im Cache. Das heißt "in den Cache schreiben". Wenn du nicht weißt, ob dieses Programm installiert ist, schreibe:

```
C:\>SMARTDRV/S
```

So kann nichts passieren. Wenn das Programm *nicht* installiert ist, zeigt es dir zuerst die **SMARTDRV** Versionsnummer sowie die verschiedenen Möglichkeiten, Parameter, an.

Ist es installiert, gibt es in der ersten Zeile an, wieviele Treffer/*Hits* und Fehlschläge/*Misses* **SMARTDRV** hatte. Das Wichtigste ist natürlich, so viele Treffer wie möglich zu haben. Als nächstes werden Anfangs-Größe des Cache und Größe des Cache während der Ausführung von Windows angegeben. Du kannst diesen Befehl ausprobieren, um zu sehen, wie das ganze funktioniert – auch während Windows läuft. Zum Schluß wird angezeigt, für welche Laufwerke der Schreib- und Lese-cache aktiviert ist.

SMARTDRV.EXE

Version 4.0 und 4.1: Der Rest dieses Abschnitts behandelt die Versionen 4.0 (Windows 3.1) und 4.1 (DOS 6.0). Soweit möglich wird **SMARTDRV** im Upper Memory-Bereich installiert.

SMARTDRV findet selbst heraus, wieviel *XMS Memory* es benötigt, ohne daß genauere Angaben zu Startgröße und Windows-Mindestgröße gemacht werden brauchen, was bei den meisten Anwendern ausgezeichnet funktioniert. Laufwerk C wird standardmäßig als Schreib- und Lese Cache eingestellt. Der Anfangswert oder die Startgröße wird manchmal auch als Laufwerk-Cache-Größe bezeichnet, während DOS-Programme laufen.

Bei 4 MB RAM wird 1 MB unter DOS benutzt und ein Minimum von 512 KB unter Windows. Wenn Windows läuft, bestimmt es selbst den Platz für Extended Memory und verfügbares Upper Memory für unterschiedliche Zwecke – Windows hat einen eigenen Speicher-Manager! Und Windows und **SMARTDRV** kooperieren, indem sie den Speicher effizient ausnutzen. Deshalb kannst du unter Windows eine Mindestgröße angeben.

SMARTDRV 1024 1024

Dies ist ein Beispiel einer Start- und Mindest-Windows-Größe, die in KB angegeben ist. Du mußt hier allmählich selbst herauszufinden, was für deine Zwecke am besten ist. Ich schlage vor, daß du einige Zeit *ein* Setup laufen läßt, in *ein anderes* Setup wechselst, wieder zum ersten Setup zurückkehrst und schaut, ob du einen Unterschied merkst. Objektiv mag es einen Unterschied geben – wenn man ihn aber nicht *merkt*, kann er nicht so groß sein. Es geht darum, ein gutes Gleichgewicht zu finden.

Du mußt neu booten, um die neuen Größen einzustellen! Die folgenden Befehle sind in allen Versionen zugänglich:

```
C:\>SMARTDRV/?
```

listet die verschiedenen Parameter auf

```
C:\>SMARTDRV C+
```

aktiviert sowohl den Schreib- als auch den Lese-Cache für das angegebene Laufwerk (C)

```
C:\>SMARTDRV C-
```

deaktiviert sowohl den Schreib- als auch den Lese-Cache für das angegebene Laufwerk (C)

```
C:\>SMARTDRV C
```

schaltet den Schreib-Cache auf Laufwerk C aus, behält aber den Lese-Cache. Wenn du mehrere Laufwerke auf deiner Festplatte hast, möchtest du möglicherweise den Schreib-Cache auf einem der Laufwerke ausschalten. Das kann im DOS eingegeben werden; ich empfehle dir aber, zu warten, bis du sicher bist, daß alle Aktivitäten auf der Festplatte beendet sind, damit keine Daten mehr im Cache "rumhängen". Dies führt uns ganz natürlich zu:

```
C:\>SMARTDRV/C
```

Sie zwingt **SMARTDRV**, alles im Cache enthaltene auf der Platte zu archivieren, das dort noch nicht gespeichert wurde. Es empfiehlt sich immer, diesen Befehl einzugeben (zum Beispiel in BAT-Dateien), besonders, wenn du oder dein PC schnell von einem in das andere Programm wechselst.

In Bezug auf SMARTDRV 4:1 warnt das DOS 6.0 Handbuch: "Schreiben Sie SMARTDRV/C im DOS, bevor Sie den PC ausschalten, um sicher zu gehen, daß SMARTDRV alle Ihre Daten auf die Festplatte gespeichert hat. Nachdem sämtliche Aktivitäten auf der Platte beendet sind, können Sie Ihren Computer ohne Risiko ausschalten."

Erster Tip: Verlasse Windows jedesmal, wenn du deinen PC ausschaltest. Zweiter Tip: bevor du deinen PC ausschaltest, kehrst du zur DOS-Prompt zurück und wartest, bis alle Aktivitäten der Festplatte beendet sind – dann erst schaltest du den Computer aus.

SMARTDRV 4.2: Etliche Benutzer haben sich über verlorengangene Daten beschwert, weil sie ihren PC in gutem Glauben zu schnell ausschalteten. Aus diesem Grund veröffentlichte Microsoft – ohne jegliche Ankündigung und vor Erscheinen von DOS 6.2 – ungefähr im September 1993 die Version 4.2. Sie hat zwar, genauso wie ältere Versionen, noch immer einen Schreib-Cache als Standardeinstellung, schreibt jedoch auf die Platte, *bevor* sie nach Beendigung eines Programmes zum DOS Prompt zurückkehrt. Das ist eine neue Einrichtung! Nun ist man zwar sicher, keine Daten mehr zu verlieren, muß dies aber mit leicht reduzierter Geschwindigkeit bezahlen.

SMARTDRV 5.0: Wird mit DOS 6.2 ausgeliefert und schaltet den Schreibe-Cache aus, schreibt also direkt auf die Platte. Wenn bei der DOS 6.2-Installation festgestellt wird, daß SMARTDRV einen Schreib-Cache benutzt, dann übernimmt es diesen.

Die Versionen 4.2 und 5.0 enthalten zwei neue Einstellmöglichkeiten:

`/X` schaltet den Schreib-Cache auf allen Laufwerken aus – anders gesagt: schreibt direkt auf die Platte.

`/N` läßt SMARTDRV darauf warten (wie in den Versionen 4.0 und 4.1), daß die CPU einen freien Moment hat, um auf die Platte zu schreiben.

Die Version 5 unterstützt das Caching für ein CD-ROM-Laufwerk. Vorausgesetzt, du hast nur ein Laufwerk auf deiner aktuellen Festplatte, ist es das Sicherste und Einfachste, folgendes in deine AUTOEXEC.BAT einzufügen:

Version 4.0 und 4.1:

```
C:\..\Smartdrv C
```

(Es wird direkt auf den Datenträger/die Platte geschrieben)

Version 4.2 und 5.0:

```
C:\..\Smartdrv
```

(Es wird in den Cache geschrieben, aber vor der Rückkehr zum DOS Prompt auf die Festplatte übertragen). Die Windows 3.11 Installation setzt `/X` ein, also ohne jeglichen Schreib-Cache.

In jedem Fall wird der Lese-Cache benutzt (was das Wichtigste ist, um den PC schneller laufen zu lassen), und in jedem Fall wird auf die Festplatte geschrieben, *bevor du* die Eingabestelle siehst. Gibst du SMARTDRV/S an der Eingabestelle ein, kannst du feststellen, welche SMARTDRV-Version du hast.

DoubleBuffering/Doppelte Pufferung

ist, wie der Name andeutet, ein doppelter Puffer, den bestimmte Festplatten-Controller benötigen. Der AT-Bus braucht ihn nicht, aber SCSI und einige ESDI- und MCA-Controller benötigen ihn. Die Zeile in CONFIG.SYS

```
Device=\..\smartdrv.exe/double_buffer
```

macht diese Controller mit der Art Speicher kompatibel, die von EMM386.EXE oder von Windows erstellt werden und in einem erweiterten Modus laufen. Der Treiber benötigt 2 KB im konventionellen Speicher und kann *nicht* in das Upper Memory gelesen werden. Auf gut deutsch bedeutet das, daß ein PC mit einem solchen Controller diese Zeile in der CONFIG.SYS haben muß. Und wie stellst du das fest? Füge die Zeile in CONFIG.SYS und SMARTDRV in AUTOEXEC.BAT ein, boote und schreibe SMARTDRV /S.

Wenn in der "Pufferung" Spalte **ja** steht, benötigt dein PC diese Zeile in der CONFIG.SYS. Steht dort nicht **ja**, kannst du die Zeile wieder löschen.

Die Festplatte

Viele Faktoren bestimmen, wie schnell Daten von der Festplatte zur CPU befördert werden. Wir schauen uns hier einige an. Die nachstehende Beschreibung ist nicht 100% korrekt, dafür aber lehrreich.

Auf einer Festplatte werden Daten in einer bestimmten Art und Reihenfolge angeordnet. Stellen wir uns die Festplatte als runde Scheibe mit dem Lesekopf in der Mitte vor, können wir sie mit einer Schallplatte vergleichen. Die Dateien beginnen in der Mitte, eine nach der anderen in einer langen Spiralkette, ähnlich wie die Musik in einer langen Rille.

Stell dir vor, der Lesekopf ist der Tonkopf des Plattenspielers – in der Mitte plziert. Wenn der Lesekopf aufgefordert wird, eine Datei zu lesen, schaut er in der *File Allocation Tabelle* nach, wo der physische Standort der Datei auf der Festplatte eingetragen ist. Der Lesekopf bewegt sich dann ein bißchen nach außen und dreht die Platte solange, bis der Dateianfang gefunden ist.

Der Unterschied zwischen der Schallplatte und der Festplatte ist, daß die Dateien auf der Festplatte nicht statisch sind – sie kommen und gehen. Von Zeit zu Zeit werden einige gelöscht, andere neu geschrieben. Stellen dir vor, daß du 100 Dateien auf deine Festplatte legst, ohne eine einzige zu löschen. Die 100 Dateien werden schön der Reihe nach angeordnet. Nun löschst du drei hintereinander stehende Dateien mit den folgenden Größen: die erste hat 50 KB, die zweite 100 KB und die dritte 80 KB. Anschließend hast du drei "leere" Stellen auf der Festplatte.

Dann wird eine Datei gespeichert/kopiert. Falls diese Datei größer als 230 KB ist, plziert DOS die ersten 50 KB an der ersten freien Stelle, die zweiten 100 KB an der zweiten und die nächsten 80 KB an der dritten; der letzte Teil der Datei wird hinter allen anderen angeordnet. Diese Datei wurde geteilt oder *fragmentiert*.

Wird diese Datei das nächste Mal gebraucht oder gelesen, muß der Lesekopf einen langen Trip rund um die Festplatte machen – und das kostet Zeit! Ich habe das natürlich sehr vereinfacht dargestellt. In Wirklichkeit dreht sich die Festplatte ständig, um die Geschwindigkeit zu erreichen, in der sie gelesen wird. Diskettenlaufwerke drehen sich nur, wenn man sie dazu auffordert.

Du hast wahrscheinlich schon spitzgekriegt, daß die fortlaufenden Dateien, die der Plattenmitte am nächsten liegen, am schnellsten gelesen werden. Der wichtigste Faktor ist, daß die Dateien fortlaufend bzw. zusammenhängend sind. Weniger wichtig ist,

wo auf der Festplatte sie sich befinden (je größer die Suchzeit, desto wichtiger wird das). Später wollen wir ein Programm beschreiben, das die Daten auf der Festplatte hin- und herbewegen, die verschiedenen Teile einer Datei einsammeln und sie auf der Platte zu einer zusammenhängenden Einheit zusammenfügen kann.

Die Zeit, die der Lesekopf braucht, um auf den Anfang einer Datei zuzugreifen, wird "durchschnittliche Suchzeit" bzw. Zugriffszeit genannt. Zur Zeit haben die verkäuflichsten Festplatten eine durchschnittliche Zugriffszeit von 10-15 Millisekunden. In der Werbung wird meistens nur diese Angabe zu einer Festplatte herausgestellt, als ob sie das einzige Qualitätsmerkmal wäre.

Sie ist wichtig; andere Faktoren, wie z.B. ein Festplatten-Cache (auf Software- oder Hardware-Basis) sind aber wichtiger. In PC Magazine's Tests wird von 'throughput', einem Durchschnittswert aus verschiedener Faktoren, geredet. Windows-Programme arbeiten mit viel größeren Datenmengen, als es noch vor einigen Jahren üblich war.

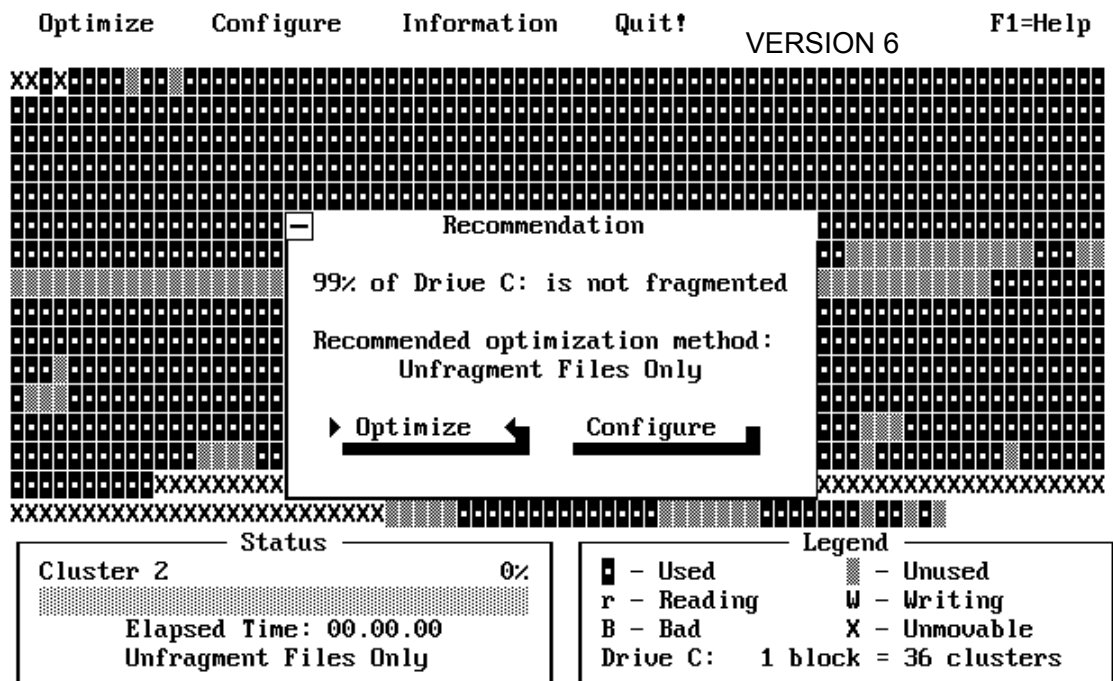
Schauen wir uns nun an, was du tun kannst, um deinen PC schneller zu machen. Du kannst die Dateien aufeinanderfolgend sortieren und diejenigen, die du am häufigsten benutzt, so nah wie möglich beim "physischen" Zentrum der Festplatte, ihrem logischer "Start", plazieren. Dazu nehmen wir uns ein Programm zur Hilfe, das das vermag.

Festplattenoptimierung

Es gibt viele Programme, die die Festplatte optimieren oder ordnen (defragmentieren). DOS 5 enthält kein Festplattenoptimierungsprogramm. DOS 6 enthält eine Mini-Version der Norton *SPEEDISK.EXE*, die *DEFRAG* heißt. Ich beschreibe hier die klassische Form der *SPEEDISK.EXE* Version 6.0/6.01/7.0 der Norton Utilities (eine Sammlung verschiedener "Haushaltungsprogramme", die DOS normalerweise nicht enthält).

Ich schrieb diesen Abschnitt, während ich eine Festplatte mit einer Zugriffszeit von 18 Millisekunden hatte. Später bin ich dann auf Festplatten mit Zugriffszeiten von 10-12 Ms gestoßen. *Je langsamer* die Festplatte, *desto größer* der Effekt einer Festplattenoptimierung.

Egal welches Plattenoptimierungsprogramm du benutzt, mache ein BackUp deiner wichtigen Dateien, bevor du das Programm startest. Ich hatte nie Probleme mit *SPEEDISK*, aber zum Beispiel durch einen Stromausfall kann einiges schiefgehen. Außerdem solltest du keinen Platten-Cache aktiviert haben. Falls du einen in einer deiner Startdateien hast, setze ihn vorübergehend außer Betrieb, indem



Select button and press ENTER to continue

| Speed Disk

du REM am Anfang der Zeile schreibst, die ihn lädt. Dann bootest du neu.

Microsoft hat versäumt, das für [SMARTDRV](#) zu empfehlen. Ich habe einige Male "vergessen", [SMARTDRV](#) außer Betrieb zu setzen – und bis jetzt ist alles gut gegangen; aber sicher ist sicher!

- 1 Du solltest dich im "reinen DOS" befinden; d.h. direkt nach dem Booten, ohne daß ein Programm läuft. Bei DOS 6 kannst du **F5** drücken, während die Meldung `STARTING MS-DOS` angezeigt wird. Dann kannst du sicher sein, daß kein Platten-Cache aktiviert ist.
- 2 Ein anderer, viel wichtigerer Punkt, ist, Laufwerk C (bzw. das Laufwerk, das du optimieren möchtest) mit `CHKDSK /F` zu überprüfen (Siehe S. 51). Damit sollten sämtliche Fehler repariert werden. Wenn du eine komprimierte Festplatte überprüfen möchtest – [DBLSPACE](#), [STACKER](#) oder [SUPERSTOR](#) – haben diese gewöhnlich ein eigenes Prüfprogramm, das das DOS Programm `CHKDSK` ersetzt. Benutzt du DOS 6.2, führst du [SCANDISK](#) aus. Wenn du [DBLSPACE](#) benutzt, mußt du [DEFRAG](#) starten.

Das Wichtigste, das ein Optimierungsprogramm macht, ist, alle Dateien zusammenzubringen; das heißt, daß jede Datei physisch zusammengefügt wird. Wenn du solch ein Programm noch nie benutzt hast, wirst die Verbesserung dich vermutlich positiv überraschen.

Wenn du zunächst nicht gerne mit der Reihenfolge der Verzeichnisse und Dateien und so weiter herumexperimentieren möchtest, dann kannst du mit einer Optimierung beginnen, die ausschließlich Dateien zu einem ganzen Stück zusammenfügt. In dem Programm [SPEEDISK](#) heißt dies [Unfragment files only](#).

Dieses Programm stellt dir viele Möglichkeiten zur Wahl. Du kannst etwa die Reihenfolge deiner Verzeichnisse bestimmen und in jedem Verzeichnis die Art der Dateien nach den folgenden Kriterien auswählen: Name, Dateieindung, Größe, Datum – je nachdem in aufsteigender oder absteigender Reihenfolge. Wie ich schon erwähnt habe, kannst du gegebenenfalls auch "nur" die Dateien optimieren, die fragmentiert sind.

Natürlich empfiehlt es sich, zunächst einen Plan zu machen. Das geht einfacher, wenn du schon ein bißchen Erfahrung mit dem Programm hast. Du wirst auch feststellen, daß du das meiste aus einem Optimierungsprogramm herausholen kannst, wenn du bereits einige Zeit mit deinem PC gearbeitet hast. Dann wirst du imstande sein herauszufinden, welche Programme du am häufigsten benutzt und welche Verzeichnisse diese Programmdateien enthalten.

Als nächstes mußt du wissen, welche Programme am häufigsten auf die Festplatte schreiben bzw. von der Festplatte lesen. Bist du dir nicht sicher, dann höre auf die Geräusche der Festplatte, während du mit verschiedenen Programmen arbeitest. Windows und die meisten

Windows-Programme greifen sehr häufig auf die Festplatte zu.

Nach meiner Erfahrung greifen Textverarbeitungsprogramme, auch solche, die unter Windows laufen, nicht sehr häufig auf die Festplatte zu. Mit Tabellenkalkulationsprogrammen habe ich noch nicht viel gearbeitet, aber die großen sind sicherlich aktive Benutzer der Festplatte. Datenbankprogramme stellen noch größere Ansprüche an die Festplatte. Auch Programme, die viel mit Grafik arbeiten, beanspruchen die Festplatte intensiv.

Du mußt also verschiedene Dinge miteinander abwägen: wie oft benutzt du ein Programm und wie festplattenaktiv ist es? Ich stelle die Verzeichnisse für die festplattenaktiven Programme an die erste Stelle, wenn ich sie regelmäßig und oft benutze. Die restlichen Verzeichnisse ordne ich entsprechend ihrer Aufgabenstellung an.

Wir wollen uns **SPEEDISK** Version 6 anschauen. Mit der eigentlichen Optimierung warten wir noch – zunächst sehen wir uns nur die Möglichkeiten an. Aktiviere deine Maus, wenn du eine hast. Du startest das Programm, indem du auf der DOS-Eingabeebene **SPEEDISK** tippst. Zunächst wirst du gefragt, welches Laufwerk du optimieren möchtest. Normalerweise wird dies C sein. Das Programm überprüft u.a. die Verzeichnisstruktur und wie fragmentiert die Festplatte ist. Wenn relativ wenig Dateien fragmentiert sind, schlägt das Programm dir vor, daß du nur diese Dateien optimierst. Du darfst nicht **Eingabe** drücken – solltest du es dennoch getan haben, drückst du **Esc** und antwortest **Abbrechen**. So kannst du jederzeit eine Optimierung unterbrechen.

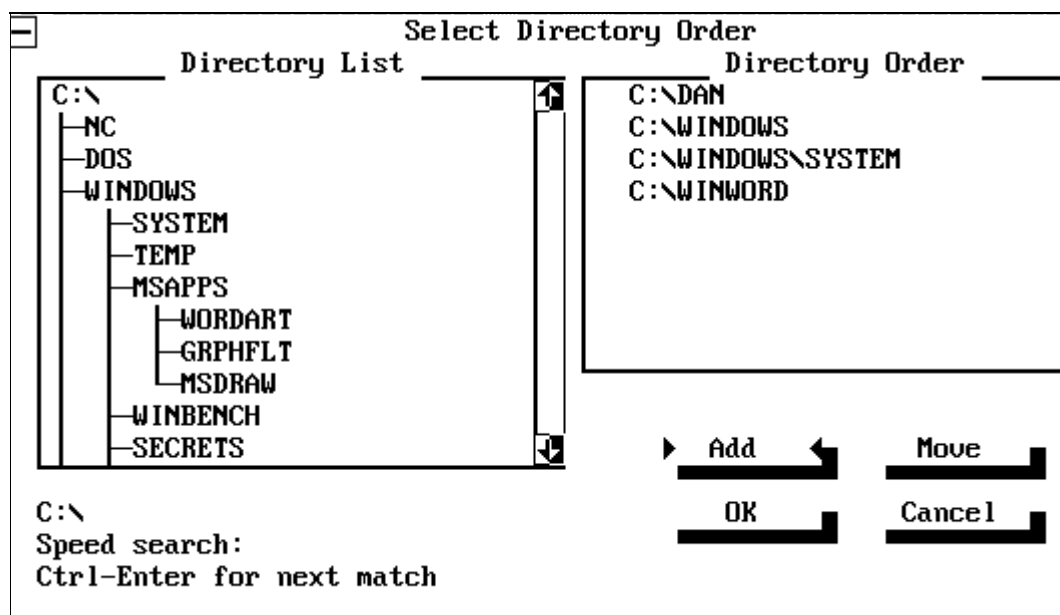
Gehe nun weiter und wähle **configure** mit der → und **Eingabe** aus oder klicke es mit der Maus an. Das Programm zeigt die Festplatte, die in kleine Blöcke aufgeteilt ist.

Anhand der Darstellung kannst du erkennen, welche Blöcke belegt und welche leer sind. Dateien, die nicht bewegt werden können, wie zum Beispiel DOS-Systemdateien und evtl. Windows Auslagerungsdateien (*swapfile*), sind mit Kreuzen markiert. **SPEEDISK** verschiebt diese Dateien *nicht*.

Ich werde nicht alle, sondern nur die wichtigsten Optionen erklären. Hier kommt es vor allem auf die *Methode* an, die Speedisk benutzen soll. Unter **Optimize** wählst du die Option **Optim. Method**. Wenn du mit dem Keyboard arbeitest, bewegst du den Cursor mit den Pfeiltasten und markierst, bzw. entfernst eine Markierung, mit der Leertaste.

Full optimiert alle Dateien, beachtet dabei aber nicht ihre Reihenfolge der Dateien und die der Verzeichnisse. **Full with File reorder** ist die Option, die ich selbst benutze. Diese Option ordnet Dateien in der Reihenfolge an, die mit **Directory Order** unter **Configure** ausgewählt wurde. Das beschreibe ich gleich. **Unfragment Files Only** ist die schnellste Art und Weise, um zusammengefügte Dateien zu erhalten.

Gehe zu **Configure** und wähle **Directory Order** aus. An dieser Stelle ist es ein bißchen schwierig, mit der Tastatur zu arbeiten. Auf der linken Seite hast du deine Verzeichnisstruktur. Rechts hast du die Reihenfolge, die **SPEEDISK** benutzt, wenn du eine Methode wählst, die auf die Reihenfolge der Verzeichnisse Rücksicht nehmen soll. Wenn du als **Optim. Method** die **Full with File reorder** wählst, werden die unter **Directory Order** ausgewählten



Verzeichnisse an den "Anfang" der Festplatte bewegt.

Mit **Tab** bewegst du dich zwischen den Fenstern, und mit den Pfeiltasten bewegst du den Cursor. Wie du sehen wirst, kannst du rechts ein Verzeichnis markieren und dann seine Priorität verändern. **Eingabe** wirkt unterschiedlich, abhängig davon, was du gerade tust. Das ist zunächst etwas verwirrend; darum ist es einfacher, wenn du die Maus benutzt. Experimentiere ein bißchen, denn wirst du schon herausfinden, wie die Verzeichnisse, die du ausgewählt hast, aufgelistet werden. Wähle **Save Options**, um deine Auswahl zu speichern.

Durch die Auswahl **File Sort** kannst du wählen, wie die Dateien in jedem Verzeichnis sortiert werden sollen. Wenn du viele Dateien bzw. große Dateien in einem oder mehreren Verzeichnissen hast, kann dies wichtig sei; ansonsten ist es weniger wichtig als die Reihenfolge der Verzeichnisse. Wichtig sind andererseits die Möglichkeiten unter **Other Options**.

Ich empfehle dir sehr, **Read after Write** auszuwählen. Dies bedeutet, daß das Programm überprüft, ob die Daten vor und nach dem Bewegen noch die gleichen sind. Wenn du diese Option nicht wählst, geht die Optimierung zwar viel schneller, aber du riskierst, fehlerhafte Dateien zu erhalten. Das kann bedeuten, daß du einige Programme neu installieren mußst. Speichere diese Optionen, indem du **Save Options** wählst.

Die komplette Optimierung einer stark fragmentierten 210 MB Festplatte kann mehrere Stunden dauern. Wenn du die Optimierung abgeschlossen hast, ist es interessant nachzuschauen, wo sich die Dateien auf der Festplatte befinden. Dies kann man machen, indem man **Walk Map** unter **Info** wählt. Auch hier ist es viel bequemer, die Maus zu benutzen.

Wenn du einige neue Dateien auf eine komplett optimierte Festplatte speichern möchtest, werden diese ganz hinten auf der Festplatte angeordnet. Da das nicht unbedingt das ist, was du möchtest, folgt hier ein kleiner Tip. Das Prinzip ist, eine Datei in eine oder mehrere Verzeichnisse zu kopieren, in denen du "Löcher" – leere Stellen – haben möchtest, nachdem **SPEEDISK** ausgeführt wurde. Diese Stellen müssen am Anfang der Festplatte stehen.

Nehmen wir einmal an, du speicherst deine Daten in C:\DATA\DOC. Lege C:\EXTRA an, wenn du es nicht bereits getan hast. Kopiere eine große Datei, 0,5 oder 1 MB in dieses Verzeichnis. Gib der Datei einen anderen Namen, zum Beispiel A.A. Bevor du eine komplette Optimierung mit **SPEEDISK** ausführst, kopierst du diese Datei nach C:\DATA\DOC. Dann aktivierst du **SPEEDISK** und

löscht anschließend C:\DATA\DOC\A.A. Du hast jetzt eine freie Stelle in C:\DATA\DOC. Dort wird die nächste Datei plaziert, die du speicherst; außerdem befindet sich diese Stelle am Anfang der Festplatte. Das kannst du natürlich mit mehreren Verzeichnissen machen, wo du deine Dateien archivierst. Eine Batchdatei, die dies ausführt, sieht folgendermaßen aus:

```
XCOPY C:\EXTRA\A.A C:\DOC
SPEEDISK
DEL C:\DATA\DOC\A.A
```

Das Programm **DEFRAG** im DOS 6 kann nur den konventionellen Speicher benutzen und enthält die Optionen: **Directory Order**, **Full with File reorder**, **Other Options nicht**; somit auch nicht **Read after Write**. Microsoft hat offensichtlich Vertrauen in die Sicherheit. Für Fortgeschrittene ist **DEFRAG** unbefriedigend. Aber selbst in der vorliegenden Form ist es besser als überhaupt keine Optimierung. Es enthält eine nicht dokumentierte Einstellmöglichkeit:

```
C:\>DEFRAG/Q
```

Hier steht Q für Schnell (Quick) – und es geht tatsächlich schnell. Mit dieser Methode sammelt man alle Dateien in einer langen Reihe, ohne jede einzelne Datei zu optimieren, und macht so einen Block von angrenzendem freiem Platz verfügbar, der beispielsweise von einer Windows-Auslagerungsdatei genutzt werden kann. Siehe Seite 39. Die Einstellmöglichkeit **/H** bedeutet, daß **DEFRAG** versteckte Dateien bewegen kann.

Norton Commander (NC)

Einleitung

Dieses Kapitel ist ein wenig anders als die übrigen, da ich hier die Funktionen beschreibe, die ich selber viel benutze. Es erinnert also an Kursusmaterial.

Viele Leute benutzen NC (Norton Commander); darum will ich etwas zu diesem ausgezeichneten Programm sagen, das ursprünglich von John Socha geschrieben wurde.

Mit der Zeit sammeln sich auf einer Festplatte viele Dateien, die in etlichen verschiedenen Verzeichnissen liegen. Um sie alle zu ordnen, benutze ich seit jeher Norton Commander. Solltest du dich wundern, warum ich in einem Text wie diesem so viel zu einem Program schreibe, das nicht mit DOS geliefert wird, so kann ich das verstehen. Wenn du dieses Programm nicht hast, wirst du leider von diesem Kapitel nichts haben; du kannst aber hier und da ein bißchen lesen – vielleicht regt es dich dazu an, das Programm zu kaufen.

Es gibt viele ähnliche Programme, u.a. Dosshell, PCTools und Datei-Manager in Windows. Ich halte den Norton Commander für das einfachste, schnellste und intelligenteste Programm.

Es wurde vor Jahren entwickelt, um die Verwaltung von Daten und Verzeichnissen zu erleichtern. Du kannst im Norton Commander die Maus benutzen; ich beschreibe aber nur den Gebrauch der Tastatur.

NC startest du, indem du auf der DOS-Ebene NC eingibst; du verläßt ihn mit **F10**. Mit **Tab** springst du zwischen den zwei Fenstern hin und her. **Esc** benutzt du, wenn etwas rückgängig machen möchtest – öffnest also du ein Fenster oder wählst eine Funktion, die du doch nicht ausführen willst, so drückst

du **Esc**.

Der Bildschirm ist in zwei Fenster unterteilt; du bekommst also einen Überblick über zwei Verzeichnisse oder sogar, falls vorhanden, zwei verschiedene Laufwerke gleichzeitig. Die Idee ist genial! Am oberen Ende der beiden Fenster siehst du die jeweiligen Verzeichnisse. Ich bezeichne die horizontale Markierung als "Balken". Der Balken markiert ein Laufwerk, ein Verzeichnis oder möglicherweise eine Datei. Ist das aktuelle Verzeichnis leer, befindet sich der Balken am oberen Ende des Bildschirms.

Beide Fenster stellen Dateien und Verzeichnisse in der Form dar, wie sie durch das DropDown-Menü ausgewählt werden kann. Probiere es aus, indem du **F9** und **Eingabe** oder den Pfeil **↓** drückst.

Dies aktiviert die DropDown-Menüs. Du kannst dich mit den Tasten **↓** und **↑** durch ein Menü bewegen und mit den Tasten **→** und **←** zu einem anderen Menü wandern. Drücke zweimal auf **Esc**, und du kehrst zurück.

Am unteren Ende des Bildschirms werden die Aktionen der verschiedenen Funktionstasten angezeigt. Drücke einmal die **Alt** – du wirst sehen, daß die Funktionstasten nun andere Funktionen erhalten. Die beiden Fenster zeigen ihre Dateien und Verzeichnisse so, wie du das durch deine Wahl im DropDown-Menü bestimmst.

C:\NU8				C:\			
Name	Größe	Datum	Zeit	Name	Größe	Datum	Zeit
disktool exe	55578	28-02-94	8.00	EXCEL	▶SUB-DIR◀	11-04-95	14.40
disktool hlp	10131	28-02-94	8.00	MERE	▶SUB-DIR◀	11-04-95	14.41
drvmap3 dll	24272	28-02-94	8.00	MOUSE	▶SUB-DIR◀	11-04-95	14.41
filefix exe	114775	28-02-94	8.00	NC	▶SUB-DIR◀	11-04-95	14.41
filefix hlp	17475	28-02-94	8.00	NU8	▶SUB-DIR◀	11-04-95	14.41
filefix ico	766	28-02-94	8.00	PICTPUB4	▶SUB-DIR◀	11-04-95	14.41
image exe	13769	28-02-94	8.00	SCANNER	▶SUB-DIR◀	12-04-95	14.32
kstack com	604	28-02-94	8.00	SHARDISK	▶SUB-DIR◀	11-04-95	14.42
ncd exe	59404	28-02-94	8.00	TEMP	▶SUB-DIR◀	11-04-95	14.42
ncd hlp	22976	28-02-94	8.00	TERMINAT	▶SUB-DIR◀	11-04-95	14.42
ncd ico	766	28-02-94	8.00	UTI	▶SUB-DIR◀	11-04-95	14.42
ndd exe	222991	28-02-94	8.00	WINDOWS	▶SUB-DIR◀	11-04-95	14.43
ndd hlp	26390	28-02-94	8.00	WINFAX	▶SUB-DIR◀	11-04-95	14.46
ndd msg	31067	28-02-94	8.00	WININST	▶SUB-DIR◀	11-04-95	20.42
ndiags exe	291120	28-02-94	8.00	WINTST	▶SUB-DIR◀	11-04-95	21.31
ndiags hlp	104978	28-02-94	8.00	WINWORD6	▶SUB-DIR◀	11-04-95	14.48
ndiags snd	56992	28-02-94	8.00	WOW	▶SUB-DIR◀	11-04-95	14.49
nlib200 rtl	200650	28-02-94	8.00	WP51	▶SUB-DIR◀	11-04-95	14.49
nlib200.rtl	200650	28-02-94	8.00	WP51	▶SUB-DIR◀	11-04-95	14.49

C:\NU8>
 1Links 2Rechts 3Beta 4Bearb 5Kompr 6DeKomp 7Suchen 8Stapel 9EGA-M. 10Baum

Konfiguration	
Farbpalette (>) S & W (●) Laptop (>) Farbe1 (>) Farbe2	Fenster-Optionen <input checked="" type="checkbox"/> Versteckte Dateien zeigen <input checked="" type="checkbox"/> Einfg bewegt nach unten <input checked="" type="checkbox"/> Verzeichnisse wählen <input checked="" type="checkbox"/> Auto-Verzeichniswechsel
Bildschirm löschen <input checked="" type="checkbox"/> An <input checked="" type="checkbox"/> 3 Minuten	Weitere Optionen <input type="checkbox"/> Menüzeile immer sichtbar <input checked="" type="checkbox"/> Setup automat. speichern <input type="checkbox"/> Linkshändermaus <input type="checkbox"/> Schneller Maus-Reset <input type="checkbox"/> Kopiere nur neuere Dat.
Druckeranschluß (>) COM1 (●) LPT1 (>) COM2 (>) LPT2 (>) COM3 (>) LPT3	
Drücken Sie LEER, um eine Option zu ändern, ↑ und ↓ zum Wechsel zwischen den Optionen.	

Einstellungen

Wir wollen zunächst einmal deinen Bildschirm und dein Programm so konfigurieren, daß sie, während du diese Zeilen liest, genauso funktionieren wie meine. Später kannst du das wieder ändern. Drücke **F9**; im Fenster *Optionen* wählst du *Konfiguration* aus. In jedem Dialogfeld wählst du mit der Leertaste einen Punkt, um ihn zu aktivieren; du bewegst dich mit den Pfeil-, Tab- oder **Eingabe**. Einige dieser Konfigurationen kannst du später ändern – ich würde dir aber folgendes vorschlagen:

Bildschirm löschen bedeutet, daß der Bildschirm (fast) leer wird, wenn du innerhalb eines bestimmten Zeitintervalls die Tastatur nicht berührst. Dadurch wird dein Bildschirm geschützt. Meine Einstellung ist 3 Minuten.

Versteckte Dateien zeigen wird nicht angewählt, das heißt versteckte Dateien werden nicht angezeigt. Du solltest sie dir erst dann anzeigen lassen, wenn du mit dem NC vertraut bist.

Einfg bewegt nach unten heißt, daß der Balken durch Drücken der **Insert** (**Einfg**) eine Stufe nach unten wandert. *Auto-Verzeichniswechsel* bedeutet: wenn du einen Verzeichnisbaum im linken Fenster und die entsprechenden Dateien eines markierten Verzeichnisses im rechten Fenster hast, werden jedesmal, wenn du das Verzeichnis im linken Fenster wechselst, die Dateien des neuen Verzeichnisses angezeigt.

Menüzeile immer sichtbar bedeutet, daß die Menüzeile am oberen Bildschirmrand sichtbar ist.

Setup autom. speichern Diese Option bedeutet, daß die Konfiguration gespeichert wird, wenn du NC beendest. Beende mit **Ok** – du drückst auf **Eingabe**, bis du **Ok** erreichst.

Wir sind wieder unter Optionen, also **F9** und entweder **↓** oder **Eingabe**. Wenn du einen Punkt wählst, wird er je nach Zustand an- oder ausgeschaltet. Leider kann man nicht mehrere gleichzeitig umschalten. Die folgenden Optionen solltest du aktivieren (was durch einen kleinen Haken neben dem jeweiligen Begriff angezeigt wird), indem du **Eingabe** drückst bzw. den hervorgehobenen Buchstaben tippst:

Optionen	Rechts
<div style="background-color: black; color: white; padding: 2px;">Konfiguration...</div> Editor... Bestätigungen... Komprimiereinstellungen...	
Automat. Menüs » Pfadprompt » Funktionstasten » Volle Bildschirmgröße » Ministatus Uhr	
Setup speichern	

Pfadprompt – Der DOS Verzeichnis-Pfad: Der DOS-Prompt am unteren Bildschirmrand zeigt das aktuelle Verzeichnis an.

Funktionstasten – Die Funktionstasten werden am unteren Bildschirmrand angezeigt.

F9 und **Eingabe**. Das Menü für das linke und das rechte Fenster ist in drei Abschnitte unterteilt; innerhalb eines jeden Rahmens kannst du eine der aufgelisteten Optionen wählen.

Links	Dateien	Befehle
Kurz		
» Voll		
Status		
Baumdiagramm		
Quick View		
Komprimierte Datei		
Verbinden		
Fenster An/Aus		Strg-F1
» Name		Strg-F3
Erweiterung		Strg-F4
Zeit		Strg-F5
Größe		Strg-F6
Unsortiert		Strg-F7
Neu lesen		
Filter...		
Laufwerk...		Alt-F1

OBERER RAHMEN – was willst du im Fenster sehen?

Kurz Zusammenfassung: Nur Dateinamen

Voll Alle Einzelheiten: Dateiname, Größe, Datum

Status Information über das aktuelle Laufwerk, belegter und freier Speicherplatz

Baumdiagramm Verzeichnis-Diagramm: Verzeichnisbaum in diesem Fenster und Dateien im anderen

MITTLERER RAHMEN – Sortierreihenfolge der Dateien. Die verschiedenen Möglichkeiten verstehen sich von selbst. Normalerweise habe ich *Name* aktiviert; änderst du aber etwas in einem aus mehreren Dateien bestehenden Programm und möchtest sehen, in welcher Datei das Programm bestimmte Dinge speichert, empfiehlt sich das *Zeit*-Kriterium. Wählst du die Sortierung nach *Zeit*, so wird diese Datei ganz oben stehen. Wenn du mehrere Dateien mit der gleichen Datei-Endung betrachten möchtest, wählst du *Erweiterung*.

UNTERER RAHMEN – benutze ich nie.

Du kannst entweder mit den Pfeiltasten wählen und dann **Eingabe** drücken oder den entsprechenden großgeschriebenen Buchstaben eingeben. Wähle **Kurz** aus; das Fenster zeigt das aktuelle Verzeichnis mit ausschließlich den Dateinamen an. Tu nun das gleiche im anderen Fenster, indem du **Tab**, **F9** drückst und dann zu **Kurz** gehst. Hast du versehentlich **F9** gedrückt, drückst du **Esc**. Bewege den Balken mit **↑** und **↓**. Du wirst sehen, daß am unteren Ende des kleinen Rahmens alle Informationen über das Verzeichnis bzw. die Datei angezeigt werden, die der Balken gerade markiert. Verzeichnisse werden in Großbuchstaben und Dateien in Kleinbuchstaben angegeben. Wähle jetzt **Full** für beide Fenster aus (**F9**, **Eingabe**, **Full**). Damit werden die Dateien mit Name, Größe, Datum und Uhrzeit angezeigt. Ich selbst bevorzuge diese Einstellung; aber das ist Geschmackssache.

Jetzt legen wir los!

Du mußt dich in beiden Fenstern im Hauptverzeichnis befinden; überprüfe deshalb bitte, ob oben in beiden Fenstern C:\ steht. Ist dies nicht der Fall, ist eines oder beide Fenster in einem Unterverzeichnis. Drücke die **Home**, und der Balken bewegt sich auf zwei Punkte (UP-DIR). Drücke **Eingabe**, und du kommst einen Schritt näher an das Hauptverzeichnis. Mit **Tab** springst du zwischen den Fenstern hin- und her. Bewege dich in den Verzeichnissen und Unterverzeichnissen hin und her, damit du mit dem Layout und der Steuerung vertraut wirst. Probiere die Tasten Home(Pos1), End(Ende), PgUp (Bild), PgDn (Bild), die Pfeiltasten und **Tab** aus.

Verschiebe den Balken in das rechte Fenster.

Wir werden ein neues Verzeichnis im Hauptverzeichnis anlegen. Drücke **F7** und nenne es EXTRA – vorausgesetzt du hast noch kein Verzeichnis mit diesem Namen. Beachte, daß der Balken automatisch auf das neue Verzeichnis EXTRA springt. Drücke **Eingabe**. Damit wählst du dieses Verzeichnis an.

Am oberen Fensterrand steht nun C:\EXTRA. Dieses Verzeichnis ist leer. Die beiden Punkte markieren, wo du einen Schritt näher an das Hauptverzeichnis kommst, wenn du **Eingabe** drückst. Der DOS-Prompt am unteren Bildschirmrand zeigt das aktuelle Laufwerk und das Verzeichnis an, das vom Balken markiert ist. Der NC hat den großen Vorteil, daß du dich in ihm bewegen und *gleichzeitig* normale DOS-Befehle auf der normalen DOS-Ebene eingeben kannst. Benutzst du **Tab**, um von einem Fenster in das andere zu wechseln, kannst du beobachten, wie sich der DOS-Prompt ändert. Setze nun den Balken in das rechte Fenster, so daß du im Unterverzeichnis EXTRA bist.

Nimm irgendeine Diskette mit Dateien. Auf der Rückseite einer Diskette oben links befindet sich ein Schieber. Verschiebe ihn mit dem Fingernagel oder einem Kugelschreiber so, daß die Öffnung sichtbar ist. Nun kann der Inhalt der Diskette nicht mehr geändert werden – sie ist "schreibgeschützt". Du kannst die Dateien immer noch lesen (man spricht von *lesen*, wenn man Informationen von der Diskette erhält, und von *schreiben*, wenn Informationen auf eine Diskette gespeichert werden). Lege jetzt die Diskette in das Laufwerk.

Wir wollen uns den Inhalt der Diskette, Laufwerk A, im linken Fenster anschauen. Drücke **Alt+F1**, dann **A** auf der Tastatur. Der Bildschirm wird einen Moment lang leer, dann wird der Inhalt des Laufwerkes A angezeigt. Mit **Alt+F2** wählst du entsprechend das rechte Fenster. Plaziere den Balken nun hier. Alle Dateien des Hauptverzeich-

nisses der Diskette werden in das EXTRA-Verzeichnis kopiert.

Kopieren mit XCOPY

Schreibe bei diesem Prompt

```
C:\EXTRA>XCOPY A:
```

Die Dateien werden kopiert und das linke Fenster aktualisiert. Mit XCOPY gibst du an, *woher* du was *wohin* kopieren möchtest. Hier nutzen wir die Tatsache, daß dieser Befehl für alles, was nicht spezifiziert wird, Standardwerte einfügt. In voller Länge hätten wir schreiben müssen:

```
C:\EXTRA>XCOPY A: . C:
```

Der Punkt bedeutet "alle Dateien". Da der Prompt sich im Verzeichnis C:\EXTRA befindet, wird dies automatisch für das *wohin* eingefügt. Die Standard-einstellung lautet: für alle Dateien, also konnten wir den Punkt hinter dem **A:** weglassen.

Wollten wir andersherum kopieren, das heißt von EXTRA nach A, müßten wir eingeben:

```
C:\EXTRA>XCOPY . A:
```

"Kopiere alle Dateien des aktuellen Verzeichnisses nach A:" Man kopiert meistens zwischen zwei Laufwerken, typisch zwischen Festplatte und Diskette.

Der Befehl XCOPY kann ausgeführt werden, ohne daß man zunächst das Inhaltsverzeichnis der Diskette mit **Alt+F1** anschaut; aber ich finde es besser, es sich erst einmal anzusehen. Man muß sich noch nicht einmal im NC befinden, um XCOPY benutzen zu können. Der Vorteil vom NC ist, daß du das Verzeichnis problemlos wechseln kannst, und das sogar an zwei verschiedenen Stellen gleichzeitig.

Sichere dich aber, daß du das richtige schreibst, also die korrekten Angaben zu *nach* und *von* machst. Hast du zum Beispiel alle deine Textverarbeitungsdateien in einem Verzeichnis deiner Platte, von dem du regelmäßig einen Backup machst, ist es wichtig, daß du von C nach A kopierst – und nicht andersherum. Bis DOS 6.2 warnt XCOPY dich *nicht*, daß die Dateien bereits dort existieren, wo du deine Kopie erstellen willst. Übrigens tut COPY das auch nicht.

Bist du zunächst noch etwas unsicher im Gebrauch von XCOPY bist, empfiehlt es sich, NC allein zu benutzen, ohne DOS-Befehle im Prompt.

Versteckte Dateien kann XCOPY leider nicht kopieren.

Versteckte Dateien

Versteckte Dateien (*hidden files*) sind versteckt, damit der Benutzer nicht mit ihnen spielt. **DIR** und **DELete** (Löschen) können solche Dateien nicht

sehen, also können sie im DOS nicht mit **DELETE** gelöscht oder über den **DIR**-Befehl besehen werden. Ziemlich vernünftig – normalerweise brauchst du eine versteckte Datei nämlich nicht zu löschen.

Willst du eine solche Datei löschen, kopieren, umbenennen oder verschieben, mußt du **F9** drücken, Optionen und Konfiguration wählen und *Versteckte Dateien zeigen* ankreuzen. Anschließend kannst du sie mit NC bearbeiten.

Dateiangaben wählen

Man wählt oft bestimmte Dateien, um sie zu bearbeiten: sie zu kopieren, zu löschen oder umzutau-fen/zu bewegen. Platziere den Balken im rechten Fenster. Wir wollen einige der Dateien löschen, die wir eben in das Verzeichnis C:\EXTRA kopiert haben. Sichere dich, daß du in C:\EXTRA stehst (am oberen Fensterrand muß C:\EXTRA stehen).

Schiebe den Balken über die Datei und drücke **Ins(ert)** (**Einfg**). Du kannst entweder die kleine graue oder die große weiße **Einfg** verwenden; bei der letzteren muß der **NumLock** ausgeschaltet sein. Nun wechselt die Zeile der Datei Zeile die Farbe und wird vermutlich gelb. Die Datei ist jetzt gewählt oder markiert.

Beachte, daß der Balken automatisch zur nächsten Datei springt. Dies wurde durch die Konfiguration (**F9+Optionen+Konfiguration**) *Einfg bewegt nach unten* bestimmt. In der Version 3.0 kann man keine Verzeichnisse, sondern nur Dateien markieren – anders als in der Version 4.0. Markiere einige Dateien, taste dann **↑** zu den gelb markierten Dateien und drücke **Ins (Einfg)**; bei der weiteren Bearbeitung werden diese Dateien nicht mehr berücksichtigt – so kannst du also Dateien auswählen. Wähle einige wenige aus. Drücke **F8**, um sie zu löschen. Zunächst wirst du gewarnt, daß du dabei bist, einige Dateien zu löschen. Drücke **Eingabe**. Du wirst noch einmal gewarnt und kannst noch mit **Esc** oder **Abbrechen** abbrechen. Drückst du **Eingabe**, werden die Dateien jetzt gelöscht.

F5, **F6** und **F8** bearbeiten die ausgewählten Dateien bzw. die Dateien, auf denen sich der Balken befindet. Wenn du also Dateien ausgewählt hast, wird die entsprechende Aktion nur für diese ausgewählten Dateien durchgeführt, egal wo sich der Balken im Verzeichnis befindet – nur muß er in dem Fenster stehen, wo der Befehl ausgeführt werden soll. Sind keine Dateien markiert, wird der Befehl in der Datei durchgeführt, auf der sich der Balken befindet.

Wir verschieben jetzt einige Dateien von einem Verzeichnis in das andere. Das rechte Fenster befindet sich noch immer in C:\EXTRA; ich hoffe, du hast hier noch einige Dateien. Springe mit **Tab** in

das linke Fenster. Wenn du dich nicht auf C: befindest, drückst du **Alt+F1** und dann **C**. Gehe in das Hauptverzeichnis, indem du **Home(Posl)** und **Eingabe** drückst, bis C:\ in der obersten Zeile auf dem Bildschirm steht.

Drücke **F7** und nenne das neue Verzeichnis DELETE. Gehe in dieses Verzeichnis. Das linke Fenster ist nun in C:\DELETE und das rechte in C:\EXTRA. Springe mit **Tab** in das rechte Fenster. Wähle mit **Ins(Einfg)** einige Dateien aus. Nun drückst du **F6**. Das Programm meldet

```
Umbenennen oder Bewegen von x Dateien nach
C:\SLET
```

F5 und **F6** nehmen standardmäßig an, daß du in das Verzeichnis, das im anderen Fenster steht, verschieben, umbenennen oder kopieren möchtest – ziemlich clever. Drücke **Eingabe**. Die ausgewählten Dateien werden in das andere Verzeichnis verschoben. Du befindest dich nun nicht mehr im Verzeichnis EXTRA, sondern in DELETE.

Fragst du dich, wie eine Datei so schnell von einem Verzeichnis ins andere verschoben werden kann, ist das gar nicht so dumm gefragt. Obwohl man die Datei "*verschiebt*", bleibt sie in Wirklichkeit auf der gleichen physischen Stelle auf der Platte. DOS verwaltet alle Dateien in der File Allocation Tabelle (FAT). Hier ist aufgezeichnet, an welcher physischen Stelle sie sich befinden und in welchem Verzeichnis sie stehen. Wird eine Datei verschoben, wird lediglich der Verzeichnisname in der FAT geändert – was nicht lange dauert.

Nun kopieren wir alle Dateien von DELETE nach EXTRA. Springe mit **Tab** in das linke Fenster. Drücke die große graue Plus-Taste ganz rechts auf deiner Tastatur. Ein Fenster auf dem Bildschirm meldet

```
Auswahl der Dateien
*.*
```

. ist ein Vorschlag; du kannst selbst etwas anderes eingeben. Wenn du **Eingabe** drückst, werden alle Dateien im aktuellen Verzeichnis markiert. Achte auf die unterste Zeile im Fenster, die dir die Anzahl und Größe der Dateien anzeigt. Die graue Taste mit dem Minuszeichen arbeitet umgekehrt – sie macht die Auswahl rückgängig (*unselect*).

Drücke **F5** und **Eingabe**. Die Dateien werden jetzt nach EXTRA *kopiert*. Ich benutze die Plus-Taste hauptsächlich, um zu sehen, wieviele Dateien sich in dem Verzeichnis befinden und wieviel Platz sie benötigen. Will ich fast alle Dateien eines Verzeichnisses behandeln, wähle ich zunächst alle und mache die Auswahl dann über **Ins** für die Dateien rückgängig, die ich nicht haben will.

Eine weitere nützliche Funktion der Plus-Taste: Nehmen wir an, du hast ein Verzeichnis mit Dateien, die mehrere MB belegen, und möchtest sie auf Diskette kopieren. Dein anderes Fenster befindet sich in A. Wähle alles mit der Plus-Taste aus, drücke **F5** und **Eingabe**. Der Kopiervorgang beginnt, bis kein Platz mehr auf der Diskette ist. Das Gute ist nun, daß NC alle Dateien schon als kopiert markiert hat. Du brauchst jetzt nur noch eine neue Diskette einzulegen und weiterzumachen, bis alle Dateien kopiert sind.

Zurück zum Beispiel. Das rechte Fenster ist noch immer in EXTRA und das linke in DELETE. Gehe in das rechte Fenster. Wähle wieder alle Dateien aus, drücke **F5** und **Eingabe**.

NC stellt fest, daß die Datei, die kopiert werden soll, bereits dort existiert, wo du sie hinkopieren möchtest, und gibt dir die Möglichkeiten **Überschreiben**, **Jede**, **Nächste Datei** und **Abbruch**. **Überschreiben** überschreibt die aktuelle Datei, **Jede** überschreibt alle ausgewählten Dateien, und mit **Nächste Datei** vermeidest du das Überschreiben der aktuellen Datei (Esc bewirkt das gleiche wie **Nächste Datei** und gehst stattdessen zur nächsten. Probier es aus, bis du das System verstanden hast. Achte darauf, daß **Esc** nur für eine Datei gilt, während **J** sich auf alle bezieht. Benutzt du nicht DOS 6.2, verfügen die DOS-Befehle COPY und XCOPY nicht über diese Optionen.

Umbenennen/bewegen

F6 kann drei Aufgaben ausführen – 1) eine Datei in ein anderes Verzeichnis verschieben, 2) einen neuen Namen innerhalb des gleichen Verzeichnisses vergeben, oder 3) ein Verzeichnis umbenennen (ein Verzeichnis ist eigentlich auch eine Datei, nur eben besonderer Art).

Wenn eine Datei von einem Verzeichnis in das andere verschoben wird, sieht es so aus, als würde sie den Standort wechseln; tatsächlich aber wird ihr nur ein neuer Name gegeben (Laufwerk + Verzeichnis + Dateiname + Dateiendung).

Schiebe den Balken auf irgendeine Datei in C:\EXTRA oder C:\DELETE. Wir werden die Datei in MARIA.DOC umbenennen. Drücke **F6** und gib MARIA.DOC ein. Drücke **Eingabe**. Wie du siehst, löscht NC den Text im Eingabefeld. Die Datei hat einen neuen Namen. Nun nennen wir die Datei um in KRISTINA.DOC. Drücke **F6** und du mußt jetzt nur noch KRISTINA.* eingeben.

Der Stern am Ende gibt an, daß du die Dateiendung behalten möchtest. Die Datei heißt nun KRISTINA.DOC. Entsprechend gibst du nur *.BRE ein, wenn du die Datei KRISTINA.BRE nennen möchtest. Tip: Willst du vermeiden, daß **F5** oder **F6** ein Verschieben in das andere Fenster vorschlagen, drückst du **Strg+F1** oder **Strg+F2**, um es kurz zu schließen – du öffnest es wieder mit der gleichen Kombination.

Jetzt weiß NC, daß du die Datei nicht in ein anderes Verzeichnis verschieben willst. Das ist vor allem praktisch, wenn du nur einen Namen ein bißchen korrigieren möchtest oder eine Datei mit leicht verändertem Namen kopieren willst. Der originale Name wird vorgeschlagen; nun kannst du den Pfeil → tasten drücken und den Namen ändern. Beginnst du mit einem Buchstaben statt des Pfeils, wird der ganze Name gelöscht.

Manchmal möchte man einigen Dateien mit gleicher Endung eine neue Datei-Endung geben. Nehmen wir an, daß du allen deinen Textverarbeitungsdateien die Endung DOC gegeben hast, bzw. daß das Programm dies automatisch so macht. Statt dessen möchtest du die Endung DOK haben. Vergewissere dich, daß du nicht schon eine Datei mit dem gleichen Namen und der gleichen Endung hast. Du kannst mit Ins alle Dateien einzeln wählen oder sie mit der grauen Plus-Taste markieren; dann tippst du *.DOC. Nun hast du alle Dateien mit der Endung DOC markiert. Drücke **F6** und gebe *.DOK ein. Drücke **Eingabe**. Alle Dateien haben anschließend die Endung DOK.

Möchtest du Kopien von allen Dateien machen, die die Endung DOC haben, benutzst **F5** statt **F6** und schreibst genauso wie oben *.DOK. Anschließend

hast du diese Dateien mit zwei verschiedenen Endungen im gleichen Verzeichnis. Nun löschst du alle Dateien in den beiden Verzeichnissen C:\EXTRA und C:\DELETE. Lösche auch die beiden Verzeichnisse.

Textdateien ansehen oder editieren

Mit **F3** und **F4** öffnest du Textdateien genauso wie mit einer Textverarbeitung. Mit **F3** kannst du die Datei nicht bearbeiten, sondern nur ansehen. **F4** kann eine Textdatei, die max. 26464 Bytes groß ist, bearbeiten. NC verfügt über einen kleinen Editor (kleine Textverarbeitung). Möchtest du über **F4** einen anderen Editor verwenden, drücke **F9**, wähle **Optionen, Editor, Extern** und gib den Programmnamen und den Pfad für das Programm ein, das aufgerufen werden soll.

Ich benutze den in NC eingebauten Editor, um kleinere Änderungen in Textdateien auszuführen, zum Beispiel in der AUOTEXEC.BAT, CONFIG.SYS, INI- und BAT-Dateien. Er ist leicht zu handhaben und schnell. **F7** ist die Suchfunktion.

Strg+Y löscht eine komplette Zeile; die Tasten **Home** (Posl), **End** (Ende), **PgUp** (Bild↑) und **PgDn** (Bild↓) funktionieren so wie in anderen Programmen. Hast du etwas geändert und willst die Datei verlassen, drückst du einfach **Esc** – und wirst gefragt, ob du die Datei speichern oder ohne Speicherung verlassen willst.

Nach Dateien suchen

Manchmal weiß man, daß man eine Datei hat, hat aber vergessen, wo auf der Festplatte sie sich befindet. Drücke **Alt+F7**, schreibe den Dateinamen, zum Beispiel MUSIK.DOC, und drücke **Eingabe**. Erinnerst du dich nicht an den ganzen Namen, gibst du einfach nur einen Teil des Names ein. MU* lokalisiert alle Dateien, die mit MU anfangen. Das Programm sucht das ganze Laufwerk ab und listet alle entsprechenden Dateien auf, die es findet. Wähle die gewünschte Datei, drücke **Eingabe** – und schon bist du da.

Anlegen einer neuen Datei

Shift+F4. Gib den Namen ein und drücke zweimal **Eingabe**.

Die zuletzt benutzten DOS-Befehle

Mit **Strg+E** kann man die eben benutzten DOS-Befehle noch einmal aufrufen. **Strg+X** blättert wieder nach vorne – genau so wie ↑ und ↓ im DOSKEY.

Fenster an- und ausschalten

Strg+O schaltet NC auf dem Bildschirm an und aus. Probier es aus. Man benutzt diese Funktion normalerweise, wenn man einen DOS-Befehl schreibt, bei dem NC das Ergebnis verdeckt. Wenn du weißt, daß deine nächsten Eingaben DOS-Befehle sind, die du am ganzen Bildschirm verfolgen willst, kannst du die NC-Anzeige vorübergehend ausschalten, indem du **Strg+O** eingibst. Du befindest dich jetzt im DOS und kannst DOSKEY benutzen, wenn es aktiviert ist. Gibst du wieder **Strg+O** ein, kannst du **Strg+E** benutzen, um zurückzublättern.

Fenster wechseln

Mit **Strg+U** vertauschst du die beiden Fenster. Probier es aus. Diese Funktion wird normalerweise benutzt, wenn man in den beiden Fenstern Verzeichnisse ausgewählt hat und plötzlich feststellt, daß sie eigentlich anders herum stehen müßten. Es empfiehlt sich, von vornherein zu entscheiden, an welcher Stelle A und C stehen sollen, wenn du mit Disketten arbeitest. Es scheint logisch, die Ange-wohnheit, von links nach rechts zu lesen, und die alphabetische Reihenfolge beizubehalten, also A nach links zu stellen. Hast du zwei Diskettenlaufwerke, A und B, machst du es genauso. Dadurch verringert sich das Risiko, beim Kopieren Fehler zu machen.

Verzeichnisse wechseln.

Mit **Alt+F10** wechselst du schnell Verzeichnisse. Probiere es aus. Gibst du einen Buchstaben ein, findet NC das erste Verzeichnis, das mit diesem Buchstaben beginnt. Schreibe genug Buchstaben, bis du an der richtigen Stelle landest. Falls zwei Verzeichnisse den gleichen Anfangsbuchstaben haben, kannst du mit **Strg+Eingabe** zum nächsten springen. Du kannst dich auch mit den Pfeil-Tasten hin- und herbewegen. Anschließend drückst du **Eingabe**. Das Fenster, in dem sich der Balken befand, zeigt nun das ausgewählte Verzeichnis an. C:\TREEINFO.NCD enthält diesen Verzeichnisbaum. Wenn sie gelöscht wird, legt NC wieder eine neue Baumstruktur an und erstellt erneut TREEINFO.NCD, wenn du das nächste mal **Alt+F10** eingibst.

Wenn du dich mitten in der Baumstruktur befindest und schnell zum Hauptverzeichnis zurückkehren möchtest, drückst du **Alt+F10** und ←, bis du dort bist.

Verzeichnisse durchblättern

F9, linkes oder rechtes Fenster, **Enter**, **Baumdiagramm**. Wanderst du mit den Pfeilen nach oben und nach unten, zeigt das andere Fenster automatisch die Dateien des ausgewählten Verzeichnisses an. Die Funktion ist für ein schnelles Durchblättern geeignet, etwa um festzustellen, wieviele Dateien in den verschiedenen Verzeichnissen enthalten sind. Und hast du Namen und Existenz einer Datei völlig vergessen, kannst du sie per Zufall beim Durchblättern findest.

Verzeichnisse vergleichen

F9, Befehle, Verzeichnisse vergleichen ist eine praktische Funktion, wenn du eine Sicherungskopie eines Verzeichnisses von der Festplatte in ein entsprechendes Verzeichnis auf Diskette anlegen möchtest. Du erhältst umgehend die Information, ob es einen Unterschied zwischen den beiden Verzeichnissen gibt. Dateien, die sich unterscheiden, werden markiert, und du kannst sie durchsehen, um festzustellen, ob du einiges löschen kannst oder ob du eine Sicherungskopie von der Festplatte auf die Diskette machen solltest.

Hast du dies alles gelesen und eine zeitlang geübt, solltest du eigentlich Weltmeister im Verwalten von Dateien sein, sowohl auf der Festplatte als auch auf Diskette. Also folgt jetzt mein Rat, wie du Verzeichnisse auf der Festplatte anlegen solltest. Wenn du Programme installierst, folge deren Empfehlungen. Legst du eigene Verzeichnisse an, dann erstelle möglichst wenige Unterverzeichnisse. Es ist besser, im Hauptverzeichnis mehr Verzeichnisse zu haben.

Im Laufe der Zeit wirst du immer mehr mit NC arbeiten. Du wirst mehr Dateien erhalten, die du verwalten muß. Plaziere deine Datendateien in separaten Verzeichnissen

Deine Dateien solltest du in einem speziellen Verzeichnis anbringen – zum Beispiel C:\DATA; hierunter erstellst du Unterverzeichnisse für jedes von dir benutzte Programm. Vermutlich benutzt du etwa eine bestimmte Textverarbeitung, deren Dateien du in C:\DATA\DOK anbringst; eine Datenbank, deren Dateien du in C:\DATA\DB setzt; und eine Kalkulationstabelle, deren Dateien in C:\DATA\RECHN gut aufgehoben wären.

Ich schlage vor, daß du alle Dateien hier speicherst; das hat viele Vorteile. Im NC erhältst du einen schnellen, generellen Überblick über deine Dokumente und kannst so einige oder alle auf Diskette kopieren.

Menü

Die Datei C:\NC\NC.MNU enthält das Menü, das erscheint, wenn du **F2** drückst. Mit dieser Textdatei kannst du etliche clevere Dinge anstellen.

Sehen wir uns einige Beispiele dafür an, was in NC.MNU stehen könnte. Die **Leertaste** und **Tab** können zum Einrücken benutzt werden.

```
D:   Gehe zu DOC
'   hier kommt ein Kommentar
    cd c:\data\dok
```

```
Editiere CONFIG.SYS
    C:\BAT\ec.bat
```

Die erste Zeile jeder Gruppe, die links starten muß, erscheint im Menü auf dem Bildschirm. **D:** am Anfang bedeutet, daß du den Befehl durch Drücken von **D** aktivierst. Das ist klug, wenn du viele Punkte hast.

Die nächste Zeile besteht aus einem oder mehreren DOS Befehlen. Du kannst mehrere NC.MNU in verschiedenen Verzeichnissen anlegen, die zu aktiven Menüs werden, wenn du zu diesen Verzeichnissen wanderst. So kannst du schnell zwischen vielen verschiedenen Verzeichnissen und somit verschiedenen Menüs wechseln

Version 4.0

ist sehr viel besser geworden. Hier nur einige wenige der Neuheiten: Wie beim Windows-Datei-Manager kannst du jetzt ein Verzeichnis gemeinsam mit seinen Unterverzeichnissen bearbeiten. Unter **Konfiguration (Optionen)** kannst du *Verzeichnisse* wählen, worauf die graue Plus-Taste neben den Dateien auch Unterverzeichnisse markiert. Der Editor (**F4**) kann viele verschiedene Funktionen ausführen, wie Block Anfang mit **F3**, Block Ende mit **F3** und Block Abbrechen mit **Shift+F3**, Suchen/Ersetzen usw.

Quick View gibt Informationen zum gewählten Verzeichnis: Anzahl der Unterverzeichnisse sowie Anzahl und Größe sämtlicher Dateien.

Weiterhin wurde ein elegantes Programm eingebaut, mit dem man ZIP-Dateien und andere komprimierte Dateien verwalten kann. NC hat sein eigenes Packprogramm im ZIP-Format. Drückst du **Eingabe**, während du auf einer Zip-Datei stehst, erscheinen die Namen aller enthaltenen Dateien, genauso wie im normalen NC. Du kannst jetzt jede einzelne Datei verwalten, zum Beispiel kopieren, löschen und so weiter – allerdings nicht einsehen! Markierst du eine Reihe von Dateien, kannst du sie mit **Alt + F5** ins ZIP-Format komprimieren. Liegen in deinem Pfad andere Packprogramme wie zum Beispiel LHA, kannst du mit **Alt+F6** Dateien in den Formaten dieser Packer auspacken (dekomprimieren).

Strg+F9 schickt eine Datei zum Drucker. **Strg+F3, F4, F5** und **F6** sortieren Dateien nach Dateiname, Dateierdung, Datum bzw. Dateigröße. Eine weitere Neuheit ist ein eingebautes Kommunikationsprogramm. Nach einem Vergleich zweier Verzeichnisse werden einige Dateien in einem Verzeichnis markiert. Jetzt kann man mit ** den Markierungszustand* der Dateien umschalten. Diese Funktion ist sehr hilfreich, wenn man etwas mit den nicht-markierten Dateien machen möchte.

Windows 3.1

Hier werden nur Themen behandelt, die zum übrigen Text eine Beziehung haben.

Windows 3.1 ist *nicht* ein Betriebssystem – es ist ein *operating environment* (Betriebsumgebung), ein Zusatz zum DOS, ein grafisches Interface, eine praktischerere und simplere Methode, Programme, Verzeichnisse und Dateien zu verwalten, als die traditionelle DOS Befehlszeile.

Windows spielt seine Vorteile aus, wenn du mit Grafik arbeiten möchtest, zum Beispiel mit Layout, Zeichenprogrammen etc., oder wenn du laufend sehen möchtest, wie bestimmte Sachen *ungefähr* aussehen, wenn sie ausgedruckt werden. In allen Windowsprogrammen kannst du die Maus benutzen und dir deinen Weg durch fast alle Ebenen klicken. Eine weiterer Vorteil ist, daß alle Programme ein ähnliches Layout und ähnliche Menü-Befehle etc. haben, genau wie beim Macintosh. Außerdem brauchst du nicht so viel von DOS-Befehlen zu wissen.

Du kannst deine normalen DOS Programme wie bisher auf dem ganzen Bildschirm laufen lassen – oder aber in einem kleineren Fenster. In manchen Situationen kannst du Daten von einem Programm ins andere verschieben. Du kannst mehrere Programme gleichzeitig laufen lassen und einfach von einem zum anderen wechseln.

Hat man sich an die Geschwindigkeit unter DOS gewöhnt, so ist das größte Problem von Windows die Tatsache, daß es eine schnellere CPU und gerne viel RAM braucht. Aus Geschwindigkeitsgründen halte ich einen 386DX/33 MHz mit 4 MB RAM für das Minimum.

Windows benötigt einen XMS Speicher und möglichst eine schnelle Festplatte und Grafikkarte. Standardmäßig (ausreichende RAM-Kapazität vorausgesetzt) startet Windows im 386 Erweiterten Modus (*386 Enhanced Mode*). Windows' sogenannter Standard Modus (*Standard Mode*) –

`C:\>WIN/S` – ist ungefähr 10-20% schneller.

Wenn du Windowsprogramme laufen läßt, die nur einen geringen Speicherplatz in Anspruch nehmen, solltest du diesen Modus benutzen. Im Erweiterten Modus für 386-PCs kannst du eine sogenannte Auslagerungsdatei (*Swapfile*) erstellen.

Auslagerungsdatei/Swapfile

Ist kein ausreichender Speicher verfügbar, weil du mehr Programme geöffnet hast, als dein RAM faßt, dann kann die Festplatte als Zusatzspeicher, also als Virtueller Speicher (*Virtual Memory*) benutzt werden. Wenn du Programme mit **Alt+Tab** verschiebst, wird alles, was der RAM-Speicher nicht faßt, aus der Swapfile aufgerufen. Das ist eine wesentliche Verbesserung der Windows-Geschwindigkeit; wenn du deine Swapfile groß genug anlegst, kannst du dich von deinen "out of Memory"-Meldungen verabschieden.

Du kannst deine Swapfile sowohl *temporär* (vorübergehend) als auch *permanent* (dauerhaft) anlegen. Die temporäre muß jedesmal erstellt werden, wenn du Windows startest. Das kostet Zeit und ist nur empfehlenswert, wenn dein Speicherplatz auf der Festplatte knapp ist. In diesem Falle wärest du besser beraten, deine Festplatte aufzuräumen, so daß du Platz für eine permanente Swapfile findest.

Die permanente Swapfile bleibt auf deiner Festplatte. Bevor du diese Datei erstellst, solltest ein Plattenoptimierungsprogramm benutzen oder

`C:\>DEFRAG/Q`

Wähle **Kontrollfeld**, **386 Erweitert**, **Virtueller Speicher**, **Verändern**. Wie groß sollte er sein? Im allgemeinen sagt man, daß dein freier XMS Speicher inclusive der Swapfile nach dem Booten 12 MB entsprechen sollte. 6-10 MB dürfte für die meisten Benutzer reichen.

Du kannst überprüfen, ob die CPU auf die Swapfile zugreifen muß, um Daten einzulesen, indem du das Lämpchen der Festplatte beobachtest, wenn du **Alt+Tab** eingibst. Hörst du beim Austausch von Informationen zwischen offenen Programmen jede Menge Plattenaktivität, solltest du versuchsweise die Swapfile vergrößern. Sie kann allerdings nur bis zu einem gewissen Bruchteil des freien Speicherplatzes auf der Festplatte erweitert werden. (Swapfile zusammen mit DoubleSpace von DOS 6: Siehe S. 45). Du solltest auch die folgende Option aktivieren:

32-Bit Zugriff

Sie ermöglicht eine schnellere Kommunikation mit deiner Festplatte, indem das langsame BIOS und DOS umgangen werden. Überprüft Windows deinen Festplattenkontroller und stellt fest, daß er mit einem bestimmten Standard (**Western Digital 1003**) kompatibel ist, kannst du 32Bit-Zugriff aktivieren, indem du das Kästchen ankreuzst. Aus technischen Gründen, die mit tragbaren PCs und deren Batterieverbrauch zu tun haben, hat Microsoft dies nicht als Standardeinstellung vorgesehen.

(A) Unglücklicherweise hat Microsoft die Bezeichnung "32Bit" für diese Kommunikationsmethode gewählt. Eine andere Bezeichnung ist FastDisk, was nichts mit dem I/O Bus oder der CPU-Adressbus-Größe zu tun hat – es geht um einen technischen Zusammenhang mit dem 386er Adressregister. Windows benutzt einen Geräte-Treiber, der im geschützten Modus direkt mit dem Festplattencontroller kommuniziert und die Kommunikation (*Throughput*) um ca. 20% erhöht.

Kannst du Windows nicht starten, nachdem du den 32Bit-Zugriff aktiviert hast, dann starte mit `C:\>WIN/D:F` und schalte den 32Bit-Zugriff aus.

Windows 3.1 kann im Geschützten oder im Erweiterten Modus (*Protected* und *Enhanced Mode*.) laufen. Während Windows im Erweiterten Modus läuft, wird jedem DOS Programm soviel Speicher zur Verfügung gestellt, als liefe es auf einem 8086'er PC. Hast du 4 DOS Programme aktiviert, simulierst du vier zusätzliche PCs. Dieser Modus wird *Virtual 8086* genannt, abgekürzt V86 Modus. Hier simuliert der 386'er Prozessor einen 8086'er Prozessor, *während* er im Geschützten Modus läuft.

Das hat den Vorteil, daß du *Real Mode* DOS Programme mit den Vorteilen von Geschützten Programmen laufen lassen kannst, was vor Speicherkonflikten schützt. Außerdem sieht es so aus, als ob du mehrere Programme gleichzeitig laufen lassen könntest. In Wirklichkeit wechselt der 386er Prozessor nur schnell zwischen den Programmen, von denen jedes die CPU für kurze, präzise Zeit kontrolliert. Das nennt man Multitasking (*Task: Aufgabe, Prozeß*). Jedes DOS-Programm hat also Zugang zum gesamten verfügbaren, konventionellen Speicher – deshalb ist die Speicheroptimierung so wichtig.

Ein 486'er Prozessor enthält in diesem Zusammenhang grundsätzlich dieselben Möglichkeiten wie ein 386'er.

Tips zu Windows

die mir sehr genutzt haben. Wenn du Einstellungen des Programm Managers speichern möchtest, ohne Windows zu beenden, halte **Umschalt** gedrückt, während du Windows mit zum Beispiel **Alt+F4** beendest. Nun sind deine Einstellungen gespeichert, ohne daß Windows beendet wurde. In diesem Fall kannst du *Einstellungen nach Beenden speichern* im Menü Optionen ausschalten.

Die grünen Buchstaben unter Windows' Hilfe sind schwer lesbar. Du kannst in `WIN.INI` unter `[Windows Help]` folgendes schreiben:

```
Jumpcolor=0 0 128
Popupcolor=128 0 0
```

Die Zahlen geben an, wie viel Rot, Grün und Blau du gibst. Spiele ein bißchen damit,, bis du weißt, was für dich am besten ist. Dank an Brian Livingston, der außerdem folgenden Tip gab: Gib in `SYSTEM.INI` folgendes ein:

```
[386enh]
MaxBPs=768
```

Diese Eingabe bedeutet *Max Breakpoints*, das heißt Maximale Anzahl von Bruchstellen. Ein Breakpoint entspricht 10 Bytes, die Windows für die Kontrolle von DOS-Arbeitvorgängen benutzt, das heißt DOS-Programmen, die wie ein *virtueller* PC laufen, also als hätten sie einen PC für sich selbst. Genauer gesagt wird eine Bruchstelle jedesmal benutzt, wenn Windows im *Real Mode* laufen soll. Kurz gesagt dreht es sich darum, daß Windows standardmäßig nur mit einer bestimmten Anzahl Bruchstellen läuft, die die Entwickler für ausreichend hielten. Viele Benutzer brauchen aber in einem einzigen Windows-Session mehr als das, und dann passieren allmählich unangenehme Fehler. Seit ich diese Zeile bei mir einführte, habe ich weniger Abstürze.

Wie wir alle wissen, sind weder Windows 3.1 noch 3.11 perfekt und stürzen oft ab. Ich habe mir angewöhnt, Windows ungefähr einmal pro Stunde abzurechnen und ganz von vorne anzufangen (ggf. auch neu zu booten), um den Speicher zu säubern. Windows frißt still und leise ein Stück Speicher, wann immer du ein Programm öffnest oder es beendest. Ich ziehe es vor, eine Minute pro Stunde zu opfern und üble Überraschungen zu vermeiden.

Wie Windows auch gestartet werden kann, erfährst du, indem du `WIN/?` eingibst.

DOS 6

Im folgenden dreht es sich um Microsofts MS-DOS, nicht um PC-DOS von IBM. Von DOS 6 wird beschrieben, was im Verhältnis zum Rest des Heftes relevant erscheint.

EMM386

Durch den Parameter `RAM` nimmt **EMM386** den gesamten Extended Memorybereich unter seine Fittiche und gibt jedem Programm, was es benötigt, ob das nun EMS- oder XMS-Speicher ist. Das setzt die Benutzung eines *PageFrame* voraus, der 64KB des Upper Memory belegt.

Unter DOS 5 konnte nur eine bestimmte Menge XMS im EMS-Speicher konvertiert werden; danach war diese Speicher-"Portion" vollständig an diese Funktion gebunden – und wenn man das wieder ändern wollte, mußte die **CONFIG.SYS** geändert und der PC neu gebootet werden.

Schreibst du zum Beispiel in deine **CONFIG.SYS**

```
..emm386.exe ram min=0 .. /v
```

kannst du während des Boot-Vorgangs in der Bildschirmmeldung sehen, daß **EMM386** jeden EMS- oder XMS-Speicher simulieren kann, den ein Programm benötigt. Benötigst du gelegentlich EMS-Speicher, empfiehlt sich die obengenannte Zeile. `min=0` bedeutet, daß anfänglich nicht 256K für EMS reserviert werden – was einträte, wenn nicht `min=0` festgelegt würde.

Du wirst zwar im Upper Memory 64K benutzen (die vom Extended Memory weggenommen wurden), mußst dich aber nicht darum kümmern, ob dein Programm EMS oder Extended Memory braucht.

Vgl. Seite 43 *More Boots*, wo du mehr über die Steuerung der Speichereinstellung lesen kannst.

Memmaker

MEMMAKER trägt einiges dazu bei, deine Startdateien zu optimieren.

Schreib zunächst einmal `MEM/?`, um die neuen Möglichkeiten dieses Programms anzuschauen. Die Informationen auf dem Bildschirm sind in dieser Version leichter zu lesen und zu verstehen als in der Version unter Win 3.1. Gib `MEM/C` ein und notiere dir die Informationen über den freien Speicherplatz – so kannst du den Prozeß leichter verfolgen, wenn du später **MEMMAKER** ausführst.

Du brauchst vor diesem Programm keine Angst zu haben. Du kannst jederzeit zur früheren Konfiguration deiner Startdateien zurückzukehren; allerdings immer nur eine Stufe zurück. Willst du weiter zurückgehen, kannst du nach jeder Ausführung die Startdateien mit der Endung `UMB` umbenennen und ihnen die Endung 1, 2 und so weitergeben (vergiß aber nicht, daß sie sich in `C:\DOS` befinden – etwas unpraktisch). Starte das Programm mit

```
C:\>Memmaker
```

Du kannst entweder die Konfigurationsoptionen *Express* oder *Custom* laufen lassen. Memmaker kopiert zunächst deine bestehenden Startdateien in `C:\DOS` und hängt ihnen die Endung `UMB` an. Anschließend ändert er nur deine Startdateien – und sonst nichts.

Willst du dorthin zurückkehren, wo du dich befandest, bevor du die letzten Änderungen gemacht hast – also nur einen Schritt zurück – schreibst du

```
C:\>Memmaker/Undo
```

MEMMAKER fügt selbständig einige Zeilen ein und ändert andere mit allen möglichen Parametern und Schaltern – und es sind etliche hinzugekommen!

Ich habe zuerst die Option *Express* ausprobiert, um zu sehen, was passiert. Es wird gebootet und ein paar Mal überprüft, was eine Weile dauert. Laufend kommen Bildschirmtexte, die mitteilen, wie weit **MEMMAKER** gerade ist. Hinterher kannst du in deinen Startdateien feststellen, was geändert wurde.

MEMMAKER hat den Vorteil, daß man ihn immer wieder laufen lassen kann, bis die optimale Konfiguration gefunden ist. Behalte den Bildschirm immer im Auge – falls Probleme auftreten. Bei mir sind bisher noch keine entstanden; sollte bei dir etwas passieren, mußst du im Handbuch nachsehen. Gerät alles völlig durcheinander, kannst du immer noch `Memmaker/Undo` ausführen.

Mit der Leertaste änderst du eine Auswahl; mit den Pfeiltasten bewegst du dich zwischen den Optionen nach oben und nach unten.

Brauchst du Expanded Memory?

Hier geht es um die Zeile mit `EMM386.EXE` und seinen Parametern.

Sowohl die Express- als auch die Custom – Installation fragen ab, ob eines deiner Programme EMS Speicher benötigt. Weißt du das nicht, rät dir `MEMMAKER`, nein zu antworten. Antwortest du ja, wird in diese Zeile `RAM` eingefügt – antwortest du Nein, kommt `NOEMS`.

Dies sind eben die zwei Möglichkeiten, um Zugang zum Upper Memory zu bekommen. Mit dem Parameter `RAM` werden 64 KB für einen PageFrame benötigt. Deshalb ist es wichtig, diese 64 KB als "freien Platz" im Upper Memory zu haben, damit nicht stattdessen 64 KB des konventionellen Speichers belegt werden.

Custom

Die fortgeschrittenere Auswahl heißt sowohl *Custom* als auch *Advanced Options*. Das Handbuch enthält nur eine sehr kurze Beschreibung der Advanced Options; auch "Help" bietet hier nicht viel mehr. Ich gebe nun eine sehr kurz gefasste Übersetzung der Hilfe. Achte darauf, **Eingabe** erst dann zu drücken, wenn du alle Entscheidungen getroffen hast. `MEMMAKER` fügt der Zeile mit `EMM386.EXE` verschiedene Parameter hinzu.

Specify ..

Hättest du keine Probleme, während `MEMMAKER` lief, antwortest du mit **Nein**. Tauchten Probleme auf, kann das an einem der Programme liegen, die mit einer Startdatei eingelesen werden. Antwortest du nun **Ja**, kannst du für jedes Programm **Ja/Nein** eingeben und so herausfinden, welches Programm Schwierigkeiten macht.

Hast du das Problem-Programm gefunden, fügst du am Anfang der Zeile, die es lädt, `REM` ein. Dann führst du `MEMMAKER` aus und löschst das `REM`, so daß das Programm geladen wird. Microsoft schlägt vor, den Namen des Programms (mit oder ohne Stern am Anfang) in `C:\DOS\MEMMAKER.INF` zu schreiben. Dies hat die gleiche Funktion.

Scan the upper ..

`MEMMAKER (DOS 6.0)` versucht standardmäßig – durch `EMM386` – den gesamten freien Platz im Upper Memory zu belegen. Wenn du an dieser Stelle **Ja** antwortest – und du solltest mit **Ja** antworten, wenn du keine Probleme hattest – wird `HIGHSCAN` in die `EMM386` Zeile eingefügt. Andernfalls antwortest du mit **Nein**; das heißt, daß ein sicherer Bereich im Upper Memory gescannt wird. Versuche es zunächst mit **Ja**; falls es nicht funktioniert, mit **Nein**. DOS 6.2 benutzt standard-

mäßig nicht `HIGHSCAN`, scannt also das Upper Memory vorsichtiger.

Move Extended BIOS ..

Antwortest du mit **Ja**, erhältst du zusätzlich 1 KB freien konventionellen Speicher. Falls später Probleme auftauchen, führst du `MEMMAKER` nochmal aus, antwortest hier mit **Nein** und schaut, ob sich dadurch das Problem beheben läßt. Das sind aber schon Feinheiten.

monochrome region ..

Schau dir die Zeichnung über das Upper Memory auf Seite 10 an. Die meisten Leute haben sicherlich einen "Super VGA"-Monitor, der eine 800 x 600 – Auflösung hat. Wenn du weißt, wie die `SYSTEM.INI` in Windows bearbeitet wird, solltest du mit **Ja** antworten. `MEMMAKER` fügt den Parameter

```
I=B000-B7FF
```

in die Zeile der `EMM386.EXE` ein. *Bevor du Windows startest, bearbeitest du Windows' `SYSTEM.INI` und schreibst in die Sektion [386Enh]:*

```
Device=C:\DOS\monoumb.386
```

Hast du einen EGA oder VGA Monitor- das heißt mit der Maximalauflösung 640x480 - probierst du es mit der Antwort **Ja** und schaut, wie es funktioniert. Klappt es, hast du 32KB im Upper Memory "gewonnen", die du verwenden kannst, um je nach Wunsch noch mehr Programme in das Upper Memory zu laden. Ob es klappt, zeigt sich, wenn du Windows startest!

Keep current EMM386 ..

Hier geht es um die Parameter `I=..` und `X=..`, die bestimmte Bereiche im Upper Memory ein- bzw. ausschließen. Verstehst du wirklich, was diese Parameter bedeuten und was sie ein- und ausschließen, antwortest du mit **Ja**. In diesem Falle werden deine `I=..` und `X=..` übernommen. Andernfalls antwortest du mit **Nein**, das heißt du überläßt die Aufgabe dem `MEMMAKER`, der sie vielleicht besser lösen kann als du selbst.

Optimize .. for Windows

Dies betrifft nur die Übersetzungs-Puffer, die ich schon erwähnt habe (*Translation Buffers*). Sie enthalten Daten im Speicher, die Windows braucht, um DOS-Programme auszuführen.

Wird **Ja** eingegeben, werden die Parameter `WIN=EA00-ECFF` und `WIN=ED00-EFFF` eingefügt. `MEMMAKER` fügt all diese Parameter der `EMM386.EXE`-Zeile hinzu, um diese Adressen im Upper Memory für die Nutzung als Übersetzungs-Puffer zu reservieren. Deshalb greift nichts anderes mehr auf diese UMBs zu.

Mein Vorschlag lautet, mit **Nein** zu antworten und mit `MEM` nachzuschauen und sicherzustellen, daß du 8KB – oder 24KB, wenn du im Netzwerk arbeitest – im Upper Memory frei hast, bevor Windows startet.

Erstellst du eine `CONFIG.SYS` mit Menüs und führst danach `MEMMAKER` aus, gerät er in Schwierigkeiten. Du mußt die Startdatei in mehrere Teile bringen und dann `MEMMAKER` auf jeden dieser Teile anwenden, um anschließend die optimierten Startdateien zu einer zu verschmelzen. Ziemlich umständlich.

Wichtig: `MEMMAKER` kann auch die Reihenfolge der Devicehigh-Zeilen *nicht* ändern, was der wichtigste Aspekt beim Memory Management ist. Das mußt du selbst tun.

Meine Bewertung: `MEMMAKER` ist besser als nichts, aber überwältigt bin ich nicht ...

More Boots

F5

Versuche zu booten und **F5** zu drücken, sobald `Starting MS-DOS ..` auf dem Bildschirm erscheint.

Dies führt einen sogenannten "reinen Boot" (*clean boot*) aus; das heißt, daß deine Startdateien nicht gelesen werden. Schreibst du `SET`, siehst du, daß dein PC einen fast reinen Bootvorgang durchlaufen hat. **F5** ist angenehm, wenn du mit der `CONFIG.SYS` experimentierst (ist das Problem nur ein Fehler in der `CONFIG.SYS`, benötigst du keine Boot-Diskette), oder bevor du ein Festplattenoptimierungsprogramm u.ä. ausführst.

DOS 6.2: falls du DoubleSpace installiert hast, kannst du es durch **Strg+F5** überspringen. Das dient meistens dazu, Probleme zu beheben, die von `DBLSPACE` verursacht werden.

F8 und ?

Boote nochmals und drücke jetzt **F8**. Du erhältst die Möglichkeit, zu jeder Zeile der `CONFIG.SYS` **J(a)** oder **N(ein)** zu sagen. Anschließend kannst du entscheiden, ob du die ganze `AUTOEXEC.BAT` laufen lassen willst oder nicht. In DOS 6.2 kannst du **J/N** in jeder einzelnen Zeile der `AUTOEXEC.BAT` benutzen.

Wird unmittelbar nach einem Befehl in der `CONFIG.SYS` ein Fragezeichen eingefügt (etwa `NUMLOCK?=ON`), wirst du gefragt, ob der Befehl ausgeführt werden soll. DOS 6.2: **Strg+F8** übergeht `DBLSPACE` auf die gleiche Art wie **Strg+F5**.

Booten mit Optionen

Anwender, die mehrere unterschiedliche Konfigurationen brauchen oder einen PC mit anderen teilen, haben es jetzt einfacher. Du kannst Konfigurieren und Booten viel fortschrittlicher gestalten durch Menüs in der `CONFIG.SYS`, mit denen du zwischen verschiedenen Optionen wählen kannst.

Text, der in der Zeile mit `menuitem` nach dem Komma geschrieben wird, wird beim Booten auf dem Bildschirm angezeigt. Die ersten Zeilen sind die Gruppenüberschriften, die sich auf die Namen der nachfolgenden Gruppen beziehen. Diese Namen müssen miteinander identisch sein!

Du kannst schreiben, was du willst, darfst aber nur ein einzelnes Wort als Gruppennamen verwenden. Ich schlage vor, daß du als letzte Zeile in der `CONFIG.SYS` eine `[COMMON]` Zeile einfügst, um mit den Programmen klar zu kommen, die der `CONFIG.SYS` während der Installation eine oder mehrere Zeilen hinzufügen. Das macht etwa DoubleSpace.

Eine leere Gruppe mit [COMMON] kann nicht schaden. Auch vor [menu] können Zeilen stehen – die in diesem Falle allgemein (*common*) sind. Ich habe versucht herauszufinden, wie man den maximalen konventionellen Speicherplatz bekommen kann für Spiele, die das fordern. Manche Spiele benötigen konventionellen Speicher, andere können Expanded Memory nutzen.

Die Variable %config% nimmt je nach Wahl einen Wert an, und es wird zu einem sogenannten Label gegangen, zum Beispiel zu :SPIELE1.

Bei mir werden 625 KB im konventionellen Speicher frei – und 622, wenn ich auch ein Expanded Memory erstelle. In der **AUTOEXEC.BAT** habe ich unter SPIELE1 die **SMARTDRV** angegeben. Du hast Zugang zum Upper Memory geschaffen und kannst jetzt ausprobieren, ob das Spiel mit **SMARTDRV** schneller läuft.

Hast du nur DOS 5 und liest dennoch dieses Kapitel über DOS 6, brauchst du nicht zu verzweifeln. Bilde aus deinen Startdateien drei Gruppen, die den vorgenannten Beispielen entsprechen. Du platzierst diese drei Gruppe in deinem DOS-Verzeichnis, gibst den Dateien jeder Gruppe (**CONFIG.SYS** und **AUTOEXEC.BAT**) die Endungen NOR, GM1 und GM2, und erstellst folgende BAT-Dateien:

```
C:\BAT>COPY CON NORMAL.BAT
COPY C:\DOS\CONFIG.NOR C:\*.SYS
COPY C:\DOS\AUTOEXEC.NOR C:\*.BAT
```

Dann drückst du **F6** und **Eingabe**, und die Datei **NORMAL.BAT** ist erstellt. Die beiden anderen Dateien, **SPIELE1.BAT** und **SPIELE2.BAT**, stellst du mit der gleichen Methode her. Möchtest du normal booten, schreibst du

```
C:\>NORMAL
```

Dies kopiert deine 'normalen' Startdateien nach C:\, und danach bootest du. Entsprechend geht es vor sich, wenn du *mit* oder *ohne* Expanded Memory spielen willst. Du erstellst eine BAT-Datei für jedes Spiel, und diese BAT-Dateien kopierst du zum Beispiel in das C:\BAT oder ein anderes Verzeichnis deines PATH. Ich hoffe, diese Beschreibung war nicht zu komprimiert!

Numlock

NUMLOCK=ON oder OFF in der **CONFIG.SYS** bedeutet...na, rate mal!

```
DEVICE=C:\DOS\HIMEM.SYS
DOS=HIGH

[menu]
menuitem=SPIELE1, mit Expanded
Memory
menuitem=SPIELE2, nur konventionell
menuitem=NORMAL

[SPIELE1]
DEVICE=C:\DOS\EMM386.EXE ram min=0
DOS=UMB
STACKS=0,0
FCBS=1
BUFFERS=10
SHELL=C:\DOS\COMMAND.COM C:\DOS\ /P

[SPIELE2]
STACKS=0,0
FCBS=1
BUFFERS=30
SHELL=C:\DOS\COMMAND.COM C:\DOS\ /P

[NORMAL]
DEVICE=C:\DOS\EMM386.EXE NOEMS /V
.. und so weiter
[COMMON]
    Vielleicht brauchst du FILES=15 oder eine
    größere Zahl. In SPIELE2 brauchst du 30 Puffer.
    du kannst es mit verschiedenen Werten versuchen.
    Wie schon erwähnt kann MEMMAKER eine solche
    CONFIG.SYS nicht optimieren, genau wie es auch
    keinen [include] Satz bearbeiten kann.

PATH C:\DOS;C:\BAT;C:\;C:\WINDOWS
goto %config%
:NORMAL
@echo off
.. und so weiter mit deiner
'normalen' autoexec.bat
goto Ende
:SPIELE1
LH smartdrv
goto Ende
:SPIELE2
:Ende
```

MS-DOS 6.2

Der wichtigste Grund für die DOS 6.2-Version, war laut Microsoft die Verbesserung der **DBLSPACE** gegenüber der 6.0-Version; darum wollen wir in diesem Abschnitt auch DoubleSpace beschreiben. Zunächst aber etwas zu den übrigen Verbesserungen der Version 6.2:

SCANDISK ist ein Programm, das Festplatten untersuchen und reparieren kann. Seitens Microsoft ist es als Ersatz für **CHKDSK** gedacht. Das Programm kann auch Fehler auf einem Laufwerk prüfen und reparieren, das mit **DBLSPACE** komprimiert wurde.

VERIFY=ON in der **AUTOEXEC.BAT** kann Probleme bei bestimmten Festplatten verursachen. Lösche deshalb die Zeile, falls sie vorhanden ist.

DISKCOPY, der eine identische Kopie einer Diskette erstellt, kann diesen Vorgang endlich simpel ausführen: du legst die Originaldiskette (Quelldatenträger) in das Laufwerk ein, dann die Kopiediskette (Zieldatenträger), und schon ist der Prozess erledigt.

COPY, **XCOPY** und **MOVE** warnen dich nun, bevor du eine Datei überschreibst.

DEFRAG kann nun Extended Memory benutzen und somit eine größere Anzahl Dateien steuern. **HIMEM.SYS** unterstützt jetzt 64 MB RAM und prüft standardmäßig die RAM-Chips im Extended Memory beim Booten.

Doublespace

In den letzten Jahren ist der Bedarf an Festplattenplatz enorm gestiegen. Mehrere Firmen haben Programme entwickelt, die die Daten auf der Festplatte einpacken (komprimieren), so daß sie weniger Platz in Anspruch nehmen. Microsoft liefert in DOS 6.0 und 6.2 das Komprimierungsprogramm DoubleSpace mit (Dateiname **DBLSPACE**).

Das funktioniert so: speichert ein Programm Daten, werden sie komprimiert; wenn sie wieder gebraucht werden, werden sie ausgepackt (dekomprimiert) – und das möglichst so, daß der Benutzer es nicht in Form von herabgesetzter Geschwindigkeit bemerkt.

DBLSPACE benutzen?

Ich habe DoubleSpace des DOS 6.2 nur kurz getestet, um etwas dazu sagen zu können; mir sind keine Mängel aufgefallen. Wird mein PC langsamer? Diese Frage beantwortet Microsoft in "HELP" frei übersetzt so:

"Hast du eine schnelle CPU und schnelle Festplatte, wirst du sicher keinen Unterschied merken. Hast du eine schnelle CPU und langsame Festplatte,

wird DoubleSpace vielleicht die Leistung erhöhen. Hast du eine langsame CPU, wird die Leistung möglicherweise reduziert."

Die Frage ist: was ist eine schnelle CPU und so weiter? Nach meiner Meinung war damals eine schnelle CPU mindestens eine 386DX/40 Mhz. Eine langsame Festplatte ist eine Standardplatte von vor 1992 mit einer sogenannten *transfer rate* von 5-700 KB/Sek; eine langsame CPU ist eine 386SX. Willst du es wirklich wissen, mußt du es ausprobieren!

Fehlt dir Platz auf der Festplatte, kannst du damit anfangen, die von mir empfohlene Installation durchzuführen und zu schauen, wie es funktioniert. Vergiß nicht, daß du 38 KB freien Upper Memory benötigst, wenn du den konventionellen Speicher nicht benutzen möchtest.

Willst du das Programm anwenden, empfehle ich vor der Installation:

- 1 Upgrade zu DOS 6.2, weil **DBLSPACE** in dieser Version verbessert und sicherer ist.
- 2 Fertige eine Sicherheitskopie deiner wichtige Dateien an.
- 3 Optimierte deine Festplatte (*Festplattenoptimierung*). Wähle die Option **FULL**, um das Maximum an freiem Festplattenplatz zu schaffen.
- 4 Bist du an ein Netzwerk angeschlossen, dann Sorge dafür, daß du verbunden bist, *bevor* du **DBLSPACE** ausführst.

Ich gehe im folgenden davon aus, daß deine Festplatte nur aus einem Laufwerk besteht, nämlich C:. Spreche ich also von einem Laufwerk, dann ist die Rede von der Festplatte. Hast du DOS 6.2, wird **SCANDISK** automatisch gestartet, bevor die Komprimierung durchgeführt wird.

Möchtest du mehr wissen, kannst du im Handbuch nachlesen und `Help Dblspace` eingeben, um weitere Informationen zu erhalten, bevor du anfängst. **DBLSPACE** Installation kann auf zweierlei Weise installiert werden.

```
C:\>DBLSPACE
```

Die erste Bildschirmmitteilung ist, daß Setup die Datei **DBLSPACE.BIN** (eine Systemdatei) lädt, die ca. 40 KB Speicher benötigt. Dann werden dir die die zwei Installationsmöglichkeiten vorgestellt: *Express* und *Custom*. *Express* wird als die einfache Methode empfohlen, *custom* für den erfahrenen Benutzer.

Das stimmt schon, nur erzählt Microsoft nicht, daß es am schwierigsten ist, eine *Express-Installation*, das heißt Komprimierung des gesamten Laufwerkes C, zu widerrufen.

Über *Express* kannst du ein ganzes Laufwerk komprimieren – wenn du nur über C: verfügst, heißt das die gesamte Festplatte.

Über *Custom* kannst du ein neues Laufwerk auf dem schon bestehendem Laufwerk erstellen – in diesem Falle wird der freie Platz auf der Festplatte benutzt.

Die meisten Benutzer werden vermutlich die Festplatte mit *Express* komprimieren – was am einfachsten durchzuführen ist, aber am schwierigsten zu *widerrufen / rückgängig zu machen*. Um Erfahrungen zu sammeln, oder falls du anfänglich unsicher bist, empfehle ich dir, die Custom Installation zu wählen, die schnell und problemlos rückgängig gemacht werden kann – dazu später mehr.

Ob du nun die eine oder die andere Methode wählst – in jedem Falle muß das neue Laufwerk einen Buchstaben als Laufwerkbezeichnung haben. **DBLSPACE** überspringt 4 Buchstaben von dem letzten reellen Laufwerk. Ist dein letztes logisches Laufwerk C:, wird der nächste Buchstabe H sein; dadurch besteht Platz für andere Programme, die zur Zeit Laufwerke neben den physisch installierten benutzen oder benutzen möchten, wie zum Beispiel Ramdisk oder ein Netzwerk.

Je nach Installationsmethode wird das neue H: Laufwerk unterschiedlich konfiguriert. Weiteres darüber später.

Custom

Ich empfehle dir, zunächst einmal ein kleines **DBLSPACE** Laufwerk zu erstellen, um ein wenig Erfahrung zu sammeln – vor allem um festzustellen, ob dein PC langsamer wird. Entscheidest du dich später, **DBLSPACE** wieder zu entfernen, ist das leicht getan, und du kannst wieder zur "normalen" Einstellung zurückkehren. Willst du stattdessen die ganze Festplatte komprimieren, kannst du eine *Custom-* (Benutzer-) Installation ziemlich einfach und schnell ändern, und zu diesem Zeitpunkt weißt du bestimmt einiges mehr über **DBLSPACE**. Du wählst also die Custom-Installation.

Die Erklärung der nächsten Bildschirmmeldung ist ein wenig verwirrend. Das Programm gibt zwei Optionen vor:

Komprimiere ein bestehendes Laufwerk
Erstelle ein neues leeres komprimiertes Laufwerk

(Falls du nur ein Laufwerk hast, entspricht die erste Option einer Express-Installation, obwohl wir dem Programm mitgeteilt haben, daß wir eine Benutzer-Installation durchführen möchten. Das Programm sollte überprüfen, ob der PC nur ein Laufwerk hat).

Ich schlage vor, du wählst mit **Eingabe** die Option **Erstelle ein neues...**, wonach die nächste

Bildschirmmeldung den freien Festplattenplatz in MB (current free space) und die zu erwartende Größe des neuen Laufwerks (projected size of new drive) anzeigt. Schreibe die erste Zahl auf, du wirst sie später gebrauchen.

Drücke **Eingabe**; nun folgt die nächste Bildschirmmeldung, daß der freie Platz auf Laufwerk C für das neue, komprimierte Laufwerk H benutzt wird. Jetzt wird's wichtig. Die oberste Linie im Rahmen zeigt an, daß lächerliche 2 MB (des freien Platzes, es passiert nichts mit deinen Dateien) auf Laufwerk C: hinterlassen werden. Hast du zum Beispiel 50 MB freien Platz, werden 2 MB in unkomprimierter Form hinterlassen. Die restlichen 48 MB wird **DBLSPACE** komprimieren, sodaß sie nachher ca. 96 MB fassen können.

Ich empfehle dir, daß du, statt die vorgegebenen 2 MB zu akzeptieren, eine größere Zahl eingibst, sodaß du nur ein neues komprimiertes Laufwerk machst, das 20 MB fassen kann. Hast du 50 MB freien Platz, bewegst du dich aufwärts auf die 2 MB, drückst **Eingabe**, schreibst im nächsten Schirmbild 40 und tastest wieder **Eingabe**. **DBLSPACE** hat nun $50-40=10$ MB vom freien Platz für ein neues Laufwerk H zur Verfügung bekommen.

Daß nicht jede Bildschirmmeldung den freien Platz auf der Festplatte anzeigt, finde ich schlecht. **DBLSPACE** ist das große Zieh Pferd im DOS 6.0. Wie wär's mit etwas Benutzerfreundlichkeit?

Du kannst übrigens ein wenig schummeln, indem du eine viel zu große Zahl, zum Beispiel 999, eingibst. Das Programm zeigt dann an, wie groß die Zahl maximal sein kann, nämlich der freie Festplattenplatz. Die Entwickler *haben* es also ausprobiert, nur keine Lust gehabt, den Anwender daran zu erinnern.

Der letzte Bildschirmtext, bevor es ernst wird, teilt mit, daß das Programm bereit ist und du auf C (continue) drücken sollst, um fortzusetzen. Bei meinem Test meinte das Programm, der Vorgang werde 53 Minuten dauern würde – er dauerte 10 ... Nun startet **SCANDISK**; werden Fehler gefunden (*lost clusters* oder *crosslinked files*), mußt du das Programm wieder verlassen, um diese Fehler zu beheben (siehe S. xxx **CHKDSK**). Tust du das mit **SCANDISK**, brauchst du keinen *Surface Scan* durchzuführen, da **DBLSPACE** dies später ohnehin tut. Danach mußt du noch einmal mit **DBLSPACE** anfangen.

(A) Jetzt macht **SCANDISK** einen *surface scan*; ist der okay, wird gebootet und die Datei C:**DBLSPACE.INI** erstellt. **DBLSPACE.BIN** wird von C:\DOS nach C:\ kopiert und beide Dateien erhalten die Dateiattribut *System*, *Versteckt* und *Nur lesen*.

Der letzte Bildschirm teilt die Größe des neuen komprimierten Laufwerkes und die Größe des übriggebliebenen freien Platzes auf C mit.

PS: Laufwerk C ist noch immer Laufwerk C, aber *HOST* für das neue komprimierte Laufwerk H.

Express

Wenn du die leichte Methode benutzt, passiert erstmal nichts weiter, als daß deine Festplatte größer und dein PC vielleicht langsamer wird. Du kannst deinen PC benutzen wie eh und je; geht irgendetwas schief oder möchtest du alles wieder rückgängig machen, solltest du wissen, was eigentlich passiert.

(A) **DBLSPACE** erstellt ein neues Laufwerk, normalerweise H, *Host Drive* genannt (das H bezieht sich nicht auf das Wort *Host!*). Dieses Laufwerk H ist nicht komprimiert; es enthält die eigentlichen Daten von C: in einer großen komprimierten Datei namens Compressed Volume File. Achtung: Laufwerk H ist Host für Laufwerk C, das auf Laufwerk H **DBLSPACE.000** genannt wird, das beim Booten (vor dem Einlesen der **CONFIG.SYS**) auf Laufwerk C ausgelagert (swapped) wird, weil **DBLSPACE.BIN** ein Teil des Betriebssystems in DOS 6.0 ist. Das heißt **DBLSPACE.BIN** ist eine Systemdatei wie die zwei anderen Systemdateien. Erstellst du mehrere **DBLSPACE**-Laufwerke, erhalten sie die Endung **DBLSPACE.001** und so weiter.

Auf Laufwerk H werden auch die anderen Systemdateien plazierte, das heißt Boot-Dateien, versteckte Dateien anderer Programmen sowie evt. permanente Auslagerungsdateien von Windows. Diese Dateien können nämlich nicht in komprimierter Form existieren. Laß um Himmelswillen die Finger von den zu **DBLSPACE** gehörigen Dateien. Du riskierst den Verlust aller deiner Dateien.

Windows' permanente Auslagerungsdatei kann ausschließlich auf einem unkomprimierten Laufwerk existieren. Wenn du also vor der **DBLSPACE**-Installation eine permanente Auslagerungsdatei hattest, sollte sie während der Installation nach H bewegt worden sein.

Hattest du vor der **DBLSPACE**-Installation *keine* Auslagerungsdatei (vielleicht hast du Windows gar nicht) und willst später eine erstellen, mußt du vielleicht (ich habe es nicht selbst ausprobiert) mehr freien Platz auf Laufwerk H schaffen. Dies tust du, indem du **DBLSPACE** startest und im Menü **Drive**, **Resize** (Laufwerk, Größe ändern) den freien Platz erhöhst, sodaß genügend Platz für die Auslagerungsdatei geschaffen wird. Der gleiche Vorgang soll über den folgenden Befehl leichter zu aktivieren sein:

```
C:\>DBLSPACE/SIZE/REServe=8
```

reserve=8 bedeutet: reserviere 8 MB auf dem unkomprimierten Host-Laufwerk.

Custom und Express

Für beide Installationen gilt folgendes:

In der **CONFIG.SYS** wird die Zeile

```
Devicehigh=c:\dos\dblSPACE.sys/move
```

als letzte hinzugefügt. Verschiebe sie, sodaß sie zur ersten **Devicehigh**-Zeile wird. Dann bist du sicher, daß **DBLSPACE.BIN** in den Upper Memory Bereich gelesen wird.

(A) **DBLSPACE.SYS** liest **DBLSPACE.BIN** ein; ist die Zeile in der **CONFIG.SYS** **buffers=8** oder weniger, kannst du ziemlich sicher im High Memory Platz für einen Teil von **DBLSPACE.BIN** schaffen (Siehe **buffers=** S. 20).

Allgemeines

Das Komprimieren von Daten ist schnell erklärt: Stell dir ein Dokument mit vielen gleichen Wörtern vor (daß, der, und, die und so weiter.). Anstatt die Wörter jedesmal, wenn sie auftreten, zu lagern, erhält jedes Wort einen Code, der einmal gelagert wird. Bei jedem neuen Auftreten des Wortes wird hinter dem Code schlicht auf die Plazierung im Dokument hingewiesen. Ein erhöhtes Risiko, daß Fehler entstehen, wenn irgendetwas mit der komprimierten Datei passiert, ist schon dabei. Es entspricht ungefähr der Situation, wenn man für hundert verschiedene denselben Code Kreditkarten hat und ihn vergißt.

DBLSPACE.BIN ist sowohl 'disk space manager' als auch ein Programm, das komprimieren und dekomprimieren kann. Da **DBLSPACE** ein integrierter Teil des Betriebssystems ist, wird **DBLSPACE.BIN** vor der **CONFIG.SYS** eingelesen. **DBLSPACE.BIN** wird zuerst in den konventionellen Speicher gelesen, später aber (durch die hinzugefügte Zeile in der **CONFIG.SYS**) in den Upper Memory Bereich überführt.

SMARTDRV kann Daten 'hinter' **DBLSPACE** auffangen, was bedeutet, daß der Cache relativ größere Datenmengen fassen kann.

Rückgängig machen

Es ist einfach, ein komprimiertes Laufwerk zu entfernen; das setzt aber voraus, daß du an anderer Stelle auf deiner Festplatte genügend freien Platz für die dekomprimierten Dateien hast. Vielleicht mußt du einige Dateien auf eine Diskette oder Kassette überführen.

Du startest **DBLSPACE**, wählst im Menü **Tools** und dann **Uncompress** (dekomprimieren). Gibt es keine komprimierten Laufwerke mehr, schlägt das Programm dir vor, **DBLSPACE** völlig zu entfernen, was du mit Ja beantwortest. (**C:\DBLSPACE.INI**, **C:\DBLSPACE.BIN** und das komprimierte Laufwerk **DBLSPACE.000** werden gelöscht. In der **CONFIG.SYS** wird die Zeile mit **DBLSPACE.SYS** gelöscht).

Windows f. Workgroups 3.11

Dieses Kapitel steht hier, weil das Programm nach DOS 6.2 auf den Markt kam. Windows für Workgroups 3.11 ist ein kleines Update der Version 3.1 und kann von einem PC angewendet werden, ohne daß dieser an ein Netzwerk angeschlossen sein muß. Es dreht sich hauptsächlich um eine schnellere Kommunikation mit der Festplatte durch ein 32-Bit-Datenzugriffssystem (ein Festplatten-Cache); außerdem dürfte diese Windows-Version stabiler sein. Windows 3.11 enthält u.a. ein kleines Faxprogramm und einen schnelleren Druck-Manager.

Die Installation fügt folgende Zeile in die CONFIG.SYS ein:

```
Device=C:\WINDOWS\ifshlp.sys
```

Das solltest du in Devicehigh ändern. Win 3.11 liefert die gleichen Versionen der wichtigen Dateien HIMEM.SYS, EMM386.EXE und SMARTDRV.EXE wie DOS 6.0/6.2, ändert aber in den Startdateien den Hinweis auf das Verzeichnis, aus dem die Dateien abgerufen werden, zum Windows-Verzeichnis, also C:\WINDOWS. Das ist okay, nur solltest du wissen, daß MEMMAKER das nächste Mal, wenn du ihn anwendest, den Hinweis wieder in C:\DOS ändert. Warum tut MEMMAKER das??

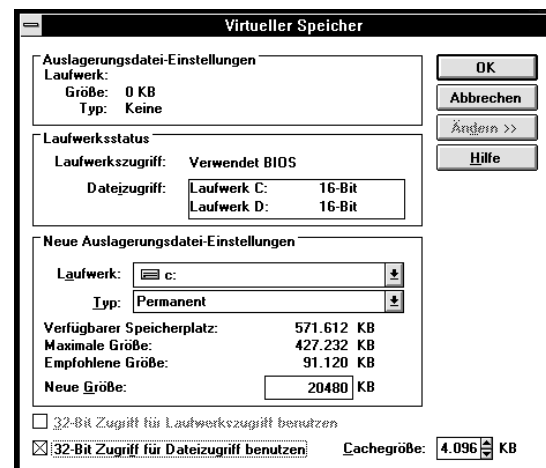
Ich vermute folgendes: Als DOS 6 eingeführt wurde, wollte Microsoft den Windows 3.1-Anwendern bei der Installation von DOS 6.0 helfen, die richtigen Einstellungen zu wählen – das heißt die neueren und besseren Versionen im DOS 6.0. Wenn du nicht damit rechnet, deine DOS Installation zu ändern, kannst du die Dateien von C:\WINDOWS nach C:\DOS kopieren. Andernfalls muß du deine Startdateien ändern, nachdem du MEMMAKER ausgeführt hast.

Das neue 32-Bit-System ist ein Cache, der SMARTDRV ersetzt, während Windows läuft. In dem Dialogfeld, wo du die Auslagerungsdatei wählst (Kontrollfeld, Erweiterter Modus für 386, Virtueller Speicher, Ändern), kannst du die Laufwerke sehen, die den neuen Cache benutzen, wenn du die 32-Bit – Option ankreuzst. Außerdem kannst du sehen, wie groß der Cache ist. Das Programm wählt einen Wert im Verhältnis zum freien Extended Memory, ähnlich wie SMARTDRV das tut. In der AUTOEXEC.BAT ändert die Installation den SMARTDRV-Parameter (den zweiten), der für SMARTDRV's Cachegröße unter Windows zuständig ist, in 128. Bei 8 MB RAM wird folgende Änderung vorgenommen:

```
\..\smartdrv 2048 128 /X
```

SMARTDRV unterscheidet sich von dem neuen 32-Bit-Zugriff (VCACHE.386) unter anderem darin, daß es einige folgende Daten der zuletzt gelesenen Sektoren in sein Lager liest, in der Hoffnung, daß die demnächst abgerufenen Daten sich in diesen Sektoren befinden.

Der neue Cache dagegen speichert die nächsten Daten der zuletzt gelesenen Datei, in der Hoffnung, daß die bald benötigten Daten aus der selben Datei stammen. Da dies sehr wahrscheinlich ist, erreicht man eine wesentlich höhere Trefferquote/hits.



Dieser neue Cache arbeitet am besten, wenn man fast nie eine Festplattenoptimierung durchführt! Tust du das häufig, ist sein Vorteil sehr gering.

Ein anderer Unterschied ist, daß der neue Cache bestimmte Vorgänge von der Real Mode in die Protected Mode bringt, was dazu führt, daß alles jetzt viel schneller geht.

Meine Erfahrung: Win 3.11 arbeitet ein bißchen schneller mit der Festplatte. Dagegen sind andere Funktionen, beispielsweise Öffnen und Schließen von Fenstern, Gänge durch Dialogfelder und so weiter, merkbar schneller geworden.

Tips zur Optimierung

Du hast sicherlich schon festgestellt, daß es keine allgemeingültigen Regeln zur Optimierung eines PC gibt. Wir alle nutzen unseren PC, wie es uns am besten paßt. Der beste Ratschlag ist: *Ausprobieren und studieren*. Unterwegs wirst du einiges lernen und möglicherweise zum Nachdenken über Dinge gebracht, die dir vorher vielleicht nicht auffielen was gar nicht so schlecht ist. Zumindest hat mich diese Erfahrung veranlaßt, dieses Buch zu schreiben.

Im allgemeinen kann man sagen: je weniger RAM, desto wichtiger wird deine Speichernutzung. Mir ist kalr, daß ich wichtige wie auch weniger wichtige Tips gegeben habe – also kommen hier nochmal die wichtigsten:

- 1) DOS=HIGH
- 2) Plattencache
- 3) Festplattenoptimierung
- 4) Permanent Swapfile / Auslagerungsdatei

2 MB RAM

Bei einem 286er mit installiertem Upper Memory Manager oder einem 386SX/SL, auf dem **EMM386** UMBs anlegen und DOS sie steuern kann, ist es wichtig, so viel wie möglich in UMBs zu installieren. Hast du eine Add-On-EMS-Karte mit RAM, brauchst du nur diesen EMS-RAM mit dem mitgelieferten Treiber zu benutzen. Lies das beigefügte Handbuch und/oder die Software für weitere Anweisungen. Du mußt sicher **DOS=HIGH** laden und, falls möglich, auch **DOS=UMB**. Das hängt davon ab, wieviel XMS-Speicher du für die Nutzung als UMB "ausleihst". Wenn du beispielsweise ca. 800KB RAM übrig hast, mußt du entscheiden, wie er genutzt wird. Falls du nur DOS-Programme laufen läßt, verwende ein Gutteil – evt. 512KB – für einen Plattencache, zum Beispiel **SMARTDRV**.

Der Cachebereich – 512 KB – wird vom Extended (oder Expanded) Memory genommen und berührt deinen konventionellen Speicher nicht. Nutzst du keine Programme wie Windows, solltest du das richtige Verhältnis zwischen der Größe (Minimum und Maximum-Werte) des **SMARTDRV** und dem freien Extended Memory finden, das Windows nutzen kann. Ich habe es zwar noch nicht ausprobiert, aber mit jeweils 400 KB dürfte die Sache ungefähr stimmen – du mußt es selbst ausprobieren.

4 MB: Du hast vermutlich mindestens einen 386er PC. Der Text wird dir sicher helfen. **SMARTDRV** sollte wie gesagt als Standard unter DOS 1024KB zum Cache wählen und unter Windows 512KB. Das

wird den meisten reichen. Wendest du Windows an, empfehle ich dir, versuchsweise einen größeren Cache zu verwenden – du kannst zum Beispiel folgendes in deine **AUTOEXEC.BAT** schreiben:

```
smartdrv 1024 768           oder
smartdrv 1024 1024
```

Benutzst du im allgemeinen nur ein bis zwei nicht gerade speicherintensive Programme, kann es sein, daß sie schneller laufen, wenn du den Cache größer einrichtest als 512 KB.

6-8 MB Generelle Ratschläge zur Größe des Plattencache oder der möglichen Nutzung eines Radisk lassen sich nicht so ohne weiteres geben. Es hängt sehr davon ab, *wieviele* Programme du normalerweise unter Windows verwendest – aber auch davon, *welche* Programme du nutzt. Ich könnte natürlich schlicht sagen, daß du es ausprobieren solltest; ich hatte 8MB und benutzte gewöhnlich Winword, ein Datenbankprogramm und evt. NC und/oder Winfax. Das lief am schnellsten, wenn ich **SMARTDRV** auf 2048 2048 stellte – das ist der Standardwert für 8MB RAM. Das heißt, daß ich für **SMARTDRV** keine Parameter einzugeben brauchte.

Die Nutzung des Upper Memory

(A) Möchtest du viele Programme im Upper Memory haben, solltest du ein großes Feld mit aufeinanderfolgenden **Upper Memory Blocks** – UMB – anlegen; allerdings nicht mehr (über ein eventuelles **I= . . .**) als nötig.

Nun solltest du zunächst die Programme laden, die den meisten Platz belegen, dann erst die kleineren. Das erste Programm, das aus der **CONFIG.SYS** geladen wird, wird in den größten, freien UMB plaziert, das nächste in den größten verbleibenden UMB, und so weiter, also in die 'Löcher'.

Mit **MEM/C/P** kannst du einigermaßen genau sehen, wie viel jedes Programm belegt. Versuche es auch mit **MEM/D/P**. Eventuell mußt du die Reihenfolge deiner **DeviceHigh-** und **Loadhigh-Befehle** ändern.

Problem: du hast anscheinend im Upper Memory Platz für ein Programm, von dem du genau weißt, wieviele KB es in Anspruch nimmt. Nur läßt sich das Programm einfach nicht in das Upper Memory laden. Grund: Es wird nicht unmittelbar ins Upper Memory gelesen. Zuerst kommt es in den Konventionellen Speicher; dann wird untersucht, ob im Upper Memory Platz vorhanden ist – wenn ja, kommt es dorthin, aber ...

Während des Einlesens ins Upper Memory beanspruchen die Programme mehr Platz, als sie benötigen, wenn sie eingelesen *sind*. Dann erst geben sie das nicht mehr gebrauchte Upper Memory wieder ab.

In **MEMMAKER.STS** *MaxSize* wird die Anzahl Bytes angegeben, die ein Programm benötigt, um "hoch" zu kommen. Kriegt du ein Programm nicht ins Upper Memory, laß ein paar Programme weg, führe **MEMMAKER** aus und sieh anschließend in **MEMMAKER.STS** nach.

Um so viele Programme wie möglich in den Upper Memory zu laden, geht es vor allem um die Reihenfolge der Devicehigh-Befehle, dann um die der LH-Befehle. Bei der Reihenfolge des Ladens kann **MEMMAKER** nicht helfen – das mußt du selbst entscheiden.

Benutzt du Windows und hast keine Netzwerktreiber installiert, solltest du nach Laden aller Programme mindestens 8 KB im Upper Memory übrig haben. Hast du Netzwerktreiber installiert, müssen 24 KB übrig sein.

Im Enhanced Mode muß Windows sogenannte *Translation Buffers* (Übersetzungspuffer) benutzen, die 8 oder 24 KB belegen. "Translation Buffers" werden gebraucht, um Daten zwischen dem Real und dem Protected Mode zu bewegen, und wenn ein DOS-Programm in einem DOS-Fenster im Enhanced Mode läuft, erfolgt ein Wechsel vom Real zum Enhanced Mode, wozu diese Puffer, vorübergehende Speicherplätze für bestimmte Informationen oder Daten, benötigt werden.

Windows: Deine Startdateien nur sollten nur die Programme (Gerätetreiber, residente Programme und so weiter) laden, die *alle* deine Programme benötigen. Brauchst du ein residentes Programm nur, während du ein bestimmtes DOS Programm benutzt, ist es besser, eine Batch Datei für dieses Programm zu schreiben, in der du das TSR *vor* dem DOS Programm lädst. Sobald das Programm beendet ist, wird das TSR aus dem Speicher entfernt.

Schauen in den Speicher – MSD.EXE

(A) Falls du das Programm MSD.EXE (aus Windows oder DOS 6) hast, kannst du es unter DOS laufen lassen und M eingeben, um zu sehen, wie das Upper Memory genutzt wird. Der graue Bereich ist für das System reserviert. **F=Free** (frei); **U=Used** (benutzt). Da das PS/2 sein BIOS-ROM-Motherboard in dem Bereich E000-EFFF hat, ist dieser Bereich *nicht* in der Standardeinstellung des **EMM386.EXE** unter DOS 5 enthalten.

Bei den meisten PC-Clones ist dieser Bereich jedoch nicht für das System reserviert und wird also verschwendet, falls er nicht in die Nutzung des *Upper Memory* eingeschlossen wird. Das ist nur von Bedeutung, wenn du einen Teil dieser 64KB im Upper Memory benutzen mußt. Der Parameter **I=E000-EFFF** schließt diesen Bereich ein. Auf diese Art werden 64 KB im Upper Memory "gewonnen"; überprüfe jedoch zuerst mit MSD, ob der Bereich frei ist (wird mit F angezeigt).

Wählst du **Utilities – Alt+U** – und **Memory Block Display** kannst du dir das erste MB in der Speicherblockanzeige anschauen, möglicherweise außer den Seiten E und F, also ab E000. Befindest du dich in Windows oder einem anderen Programm und kannst auf MSD oder ein anderes Programm umschalten, das den Speicher anzeigt, kannst du feststellen, wo im Speicher die verschiedenen Programme und Dateien plaziert sind. Allerdings ist die Information (wie auch die Bildschirmmeldung besagt) eventuell nicht ganz korrekt, da gleichzeitig Windows läuft. Abgesehen davon ist das eine gute Sache, wenn man sehen will, ob eine Änderung funktioniert hat.

Du kannst eine Menge über deinen PC herausfinden, wenn du **Alt, Datei, Drucken, Datei** und **Eingabe** wählst. Fast die gleichen Ergebnisse erzielst du, wenn du

```
C:\>MSD/P sysinfo.doc
```

eingibst (dieser Dateiname ist mein Vorschlag), aber wie bei allen DOS Befehlen kannst du die unterschiedlichen Möglichkeiten durch die Eingabe

```
C:\>MSD/?
```

herausfinden.

DOS

CHKDSK/F

wird an mehreren Stellen dieses Heftes erwähnt. Vor DOS 6.2, in dem dieses Programm durch **SCANDISK** ersetzt wurde, benutzte man es, um Plattenfehler zu reparieren. Hast du DOS 6.2, kannst du statt **CHKDSK** **SCANDISK** verwenden.

Ein typischer Fehler (von **CHKDSK**) auf einer Platte sind *lost clusters* und *crosslinked files*. Lost clusters sind Datenteile, die nicht mit einem Namen versehen sind, während crosslinked files sich auf einen Zustand beziehen, in dem zwei Dateien den gleichen Platz auf der Platte teilen. bzw. dem gleichen Platz zugeordnet/verbunden sind.

Lost clusters: werden Fehler gefunden, antworte **Ja** auf die Frage nach einer Reparatur, dann kannst du entsprechenden Dateien namens **FILE0000.CHK** und so weiter, die im Hauptverzeichnis liegen, untersuchen und sie je nachdem unter einem verständlichen Namen speichern oder löschen.

Crosslinked files: Falls **CHKDSK** crosslinked files berichtet, notierst du ihre Namen, kopierst sie unter anderem Namen irgendwo auf deine Platte und löschst das Original. Dadurch löschst du auch die Verbindung zu der gleichen Stelle auf der Platte.

Boot-Disketten

Nachdem du jetzt einiges über die Startdateien weißt, mache ich dir ein paar Vorschläge zu Disketten, die du anwenden kannst, wenn du später auf Probleme stoßen solltest. Eine *Boot-Diskette* nennt man auch eine Systemdiskette. Eine Setup-Diskette ist etwas anderes – sie installiert (kopiert) ein Programm (in diesem Fall DOS) auf die Festplatte. Du hast möglicherweise eine Setup-Diskette, damit aber nicht unbedingt eine Bootdiskette.

Erfahrungsgemäß ist es eine gute Idee, sicher zu gehen. Besonders wenn man in *DOS 5* mit der **CONFIG.SYS** herumexperimentiert, sollte man eine Boot-Diskette zur Hand haben. Ich habe öfter nicht von meiner Festplatte booten können, während ich Versuche anstellte. Der Inhalt in der **CONFIG.SYS**-Datei kann deinen PC blockieren, das heißt der Bootvorgang wird plötzlich gestoppt und nicht abgeschlossen. DOS 6 hat dieses Problem zum Glück gelöst.

Auch eine nicht vorhandene oder kaputte **COMMAND.COM** kann den PC blockieren. Ich habe mir einmal die **COMMAND.COM** bloß mit dem Norton Commander Editor angeschaut (man will ja mal sehen was drin ist), ohne an ihr rumzufummeln, und bin dann wieder rausgegangen. Mein PC

strikte – Booten von der Festplatte war unmöglich. Faß also *nie* den **COMMAND.COM** an.

Ein PC kann leider nicht einfach angeschaltet werden, und dann funktioniert er. Das Betriebssystem muß zuerst in den Speicher des PC geladen sein.

Daß du DOS hast und eine der Setup-Disketten deinen PC starten kann, bedeutet nicht, daß du damit eine Boot-Diskette hast. Ich empfehle dir deshalb, die folgenden Disketten zu erstellen. Ein PC kann unmittelbar nur von Laufwerk C und A booten. Viele Disketten sind im Voraus schon formatiert; hast du eine *unformatierte* Diskette, kannst du sie mit folgendem Befehl formatieren:

```
C:\>FORMAT A:
```

Diskette 1 – clean boot

Du legst ein *leere, formatierte* Diskette in das Laufwerk und schreibst

```
C:\>SYS A:
```

Es erscheint die Meldung `system transfered` was bedeutet, daß zwei versteckte System- oder Bootdateien und **COMMAND.COM** auf die Diskette kopiert wurden (wenn du DOS 6.0 benutzt wird auch **DBLSPACE.BIN** kopiert). Diese Dateien braucht DOS, um zu starten. Benutzst du DoubleSpace, dann ist **DBLSPACE.BIN** notwendig, wenn du Zugang zu einem von DoubleSpace komprimierten Laufwerk haben möchtest.

Kopiere die Datei **SYS.COM** auf die Diskette:

```
C:\DOS>COPY SYS.COM A:
```

Schreibe **CLEAN BOOT** oder **REINES BOOTEN** auf die Diskette. Jetzt hast du eine Boot-Diskette, mit der du jederzeit deinen PC starten kannst, wenn er nicht von der Festplatte booten kann. Diese Diskette enthält nur die drei oder ggf. vier wichtigen Dateien des Betriebssystems; bootest du mit dieser Diskette, machst du einen sogenannten *reinen (clean) Boot*. PS: manche Buchstaben und Zeichen befinden sich unter einer anderen Taste, da der deutsche Tastaturtreiber nicht eingelesen ist.

Willst du die Dateien auf der Diskette sehen, kannst du folgenden undokumentierten **DIR** mit einem Komma eingeben:

```
A:\>DIR,
```

Bestimmte Spiele verlangen einen reinen Bootvorgang von einer Diskette, wozu diese Diskette gebraucht werden kann. Die Diskette kann zwei Probleme lösen:

Problem 1

Du kannst nicht von der Festplatte booten und bekommst die Fehlermeldung `Non-System disk`. Wenn das Problem nur darin besteht, daß etwas an einer der Systemdateien oder `COMMAND` nicht funktioniert, kannst du nach dem Booten mit der Diskette

```
A:\>SYS C:
```

eingeben, womit du die drei oder vier Dateien auf C:\ kopierst. Erscheint darauf die Meldung `system transferred`, solltest du wieder von der Festplatte booten können.

Problem 2

Bekommst du die Fehlermeldung `missing or bad Command Interpreter`, ist ein Fehler in der `COMMAND.COM` aufgetreten. Ist das der Fall, bootest du mit der erstellten Bootdiskette und gibst ein:

```
A:\>copy command.com C:\
A:\>copy command.com C:\DOS
```

Diese Eingaben kopieren eine intakte Kopie der `COMMAND.COM` von der Diskette in die zwei relevanten Verzeichnisse. Vielleicht brauchst du die `COMMAND.COM` nur in einem Verzeichnis. Entferne die Diskette aus dem Laufwerk und boote wieder von der Festplatte.

Diskette 2

Nimm eine weiterreformatierte Diskette und mach nochmal das selbe:

```
C:\>SYS A:
```

Diese Diskette soll genauso booten, als würde von der Festplatte gebootet. Also müssen alle notwendigen Dateien auf der Diskette vorhanden sein – daher die Idee von den Startdateien ohne Angabe des C:\s vor den Befehlen. Die Startdateien sind gleich, abgesehen vom PATH.

Die folgenden vielen Kopien sind natürlich einfacher mit einem Programm wie NC durchzuführen; damit aber alle mitmachen können, kommen hier die DOS-Befehle. Vergewissere dich, daß du vorher von der Festplatte booten kannst, und kopiere die Startdateien auf die Diskette, indem du folgendes eingibst:

```
C:\>copy config.sys a:
C:\>copy autoexec.bat a:
```

Ändere in A:AUTOEXEC.BAT die Pathangabe zu `path=;\DOS`

(Die folgenden Befehle sind vielleicht ein wenig ungewöhnlich für einige Anwender, u.a. weil ich davon rede, ein Verzeichnis mit dem CD Befehl als Standardverzeichnis einzustellen. Aber genau das

tut CD. Für einige sieht es vielleicht ein bißchen merkwürdig aus, daß ich ein Verzeichnis auf A: erstelle/wechsle ohne dort zu stehen).

Erstelle das Verzeichnis A:\DOS

```
C:\>MD A:DOS
```

Stelle dieses Verzeichnis als Standardverzeichnis auf A: ein

```
C:\>CD A:DOS
```

Stelle C:\DOS als Standardverzeichnis auf C: ein

```
C:\>CD DOS
```

Kopiere die Dateien, die in deinen Startdateien auftreten, nach A:\DOS. Benutze evt. F3 nach jedem Befehl und editiere ein wenig im zuletzt eingegebenen Befehl. Die ersten Befehle könnten so aussehen (A:\DOS ist als Standard eingestellt!):

```
C:\DOS>copy himem.sys a:
C:\DOS>copy emm386.exe a:
C:\DOS>copy display.sys a:
```

und so weiter, bis alle Dateien, die in deinen Startdateien auftreten, kopiert sind.

Diese Diskette kann deinen PC mit der gleichen Einstellung booten wie deine Festplatte, ohne Dateien von der Festplatte zu gebrauchen. Kopiere entsprechend folgende Dateien von C:\DOS zu A:\DOS: `UNFORMAT.COM`, `FORMAT.COM`, `CHKDSK.EXE (SCANDISK.EXE)`, `UNDELETE.EXE`, `FDISK.EXE` und `SYS.COM`. Kopiere evt. auch andere Hilfsprogramme auf die Diskette. Vergewissere dich, das sie funktioniert. Schreibbesetze sie und lege sie an eine sichere Stelle.

Die Diskette kannst du gebrauchen, wenn deine Festplatte so kaputt ist, daß du überhaupt keinen Zugriff auf ihre Dateien hast, das heißt sie nicht mehr lesen kannst (`cannot read drive C:` oder `error reading drive C:`).

Solltest du das Pech haben, daß nur eine neue Partitionierung und/oder Formatierung oder ein `UNFORMAT` deiner Festplatte hilft, kannst du diese Diskette benutzen. Das kommt zum Glück selten vor, aber wen es trifft, dem ist es egal, wie selten das ist ... Kann dir jemand helfen, benutze das ruhig in diesem speziellen Fall.

Ein Experte wird dir in dieser kritischen Situation wesentlich mehr helfen. Bist du selbst ein Experte, der anderen hilft, empfiehlst sich diese Diskette. Lege sie ins Laufwerk und boote, um zu sehen, ob dein PC ohne Probleme bootet. Fehlender Kontakt zur Festplatte kann auch an einem Problem liegen, das im folgenden Kapitel beschrieben wird.

CMOS und Setup

(A) Eine Sicherheitsmaßnahme, die das Handbuch selten beschreibt, ist, die Informationen in der CMOS aufzuschreiben (sprich: siehmos). Die CMOS ist der Teil des RAM-Speichers, der die Informationen zu PC-Konfiguration die Festplatte speichert. Diese Information wird durch eine – gewöhnlich aufladbare – Batterie "am Leben gehalten". Wenn ein "Unfall" passiert und die Festplatte aus dem einen oder anderen Grund nicht mehr funktioniert – vielleicht auch, weil die CMOS geändert oder gelöscht wurde – solltest du die Werte der CMOS kennen, um die Festplatte wieder ans Arbeiten zu bringen.

Glücklicherweise passiert dies selten. Je neuer deine Festplatte ist, desto geringer ist das Risiko. Meine Festplatte ist einmal abgestürzt, als ich die Resettaste in Windows drücken mußte. Mir blieb nichts anderes übrig – und schon knirschte meine Festplatte. Keines meiner Ersthilfe-Programme konnte den Fehler beheben. Die FAT, Boot Record – alles war verschwunden. Das einzige, was ich tun konnte, war, ganz neu zu starten, die Festplatte zu formatieren, die Programme neu zu installieren und meine Daten (die ich auf Diskette hatte) wieder auf die Festplatte zu kopieren. Damals lernte ich, Sicherungskopien zu machen.

Deinem PC sollte ein Prospekt des Herstellers deiner Festplatte beigelegt sein. Wenn du die Typenbeschreibung deiner Festplatte kennst, findest du in diesem Prospekt die entsprechenden Angaben. Hast du den Prospekt nicht, fragst du deinen Händler nach den Werten.

Gibt deine Festplatte Samstagabend wegen der CMOS-Werte den Geist, macht es keinen Spaß, bis Montagmorgen warten zu müssen, bis der Händler seinen Laden öffnet.

Du kannst auch in deinem Hardware-Setup nachschauen. Das solltest du aber nur tun, wenn du damit vertraut bist! Boote mit der Resettaste. Gewöhnlich erscheint auf deinem Bildschirm die Meldung, daß du eine oder mehrere Tasten drücken mußt, um in das Setup zu kommen, wo die CMOS Informationen gespeichert sind. Meist werden die Esc, Del oder evt. Strg+Alt+Esc dazu benutzt. Sollte nichts passieren, wenn du das Setup starten möchtest, schlage die Einzelheiten in deinem Handbuch nach.

Du mußt das CMOS-Setup finden, oder – falls du ein Inhaltsverzeichnis hast – etwas über Festplatten. Notiere dir die Werte deiner Festplatte: Typ (gewöhnlich benutzerdefiniert Nr. 46 oder 47), die Anzahl der Zylinder, Zylinderköpfe und Sektoren. Schreibe diese Angaben auf einen Aufkleber und hefte ihn an das PC-Gehäuse. Beispiel: Typ 47, 1024 Zylinder, 10 Köpfe und 17 Sektoren. Um SETUP zu verlassen, mußt du normalerweise F10, Esc oder ähnliches drücken. Gelegentlich wirst du gefragt, ob

du die Änderungen speichern möchtest. Antworte mit NEIN. Mußt du JA zu antworten, dann tu das – sei aber sicher, daß du keinen der Werte veränderst und auch sonst nichts getan hast, ohne die Konsequenzen zu kennen. Dann drücke nochmals die Resettaste. Falls du weißt, daß nicht gespeichert ist, ist das sicherste Weg, das Programm zu verlassen und wieder in den "Urzustand" zurückzukehren: Drücke Reset!

Einige Setups haben sowohl einen Standard-Setup als auch einen Fortgeschrittenen. Du mußt wissen, was du tust. Ändere niemals Werte einfach aus Spaß, um zu sehen, was eventuell passiert. Dein Händler wird nicht gerade glücklich sein, wenn du etwas tust, wovon PC-Anwender bedie Finger lassen sollten. Ich weiß das, weil es mir passiert ist! Ganz allgemein: Sei vorsichtig mit dem Setup, vor allem dem Fortgeschrittenen. Dieses Thema behandle ich nicht in diesem allgemeinen Text.

Blind- & Schnellschrift

Vielleicht hat der beste Ratschlag, den ich dir geben kann, gar nichts direkt mit Computern zu tun. Ich habe jetzt viele Jahre in der Computerwelt gearbeitet und bin überrascht, wie wenig Leute die Brailleschrift beherrschen.

Ich schreibe dies absichtlich, um die Leute zu provozieren, die ganz von ihren schnellen PCs angetan sind, den schlaun Programmen und so weiter. Aber kann dir dein Turbo PC helfen, wenn er darauf wartet, daß du die richtige Taste findest? – und ich rede nicht nur von den Buchstabentasten sondern auch von Strg, Shift, Alt und all den anderen Funktionstasten und ihren Kombinationen.

Ich habe 30 Stunden an einer alten Schreibmaschine verbracht, um Brailleschreiben zu lernen. Ich wurde fast verrückt, als mein kleiner Finger das "p" und das "?" drücken mußte, aber ich war beharrlich.

Ich habe meinen kleinen Finger so hart trainiert, daß er mir heute auf der Tastatur keine Probleme mehr bereitet. Du kannst sowohl Bücher als auch Programme finden, die dich zwingen, deine Finger-muskeln zu trainieren.

Ich habe sehr viele Stunden gespart und konnte somit Dinge schreiben, die ich sonst nie hätte schreiben können. Ich mache selten Fehler. Eine gute Tastatur zu günstigem Preis kenne ich nicht – IBM hat immer noch die beste, die ist aber sehr teuer).

Deine Gesundheit

Ich habe meinen Computer auf einer kleinen Kiste fast auf Augenhöhe stehen, sodaß ich leicht nach unten sehen muß. Beherrscht du die Blindschrift, brauchst du kaum auf die Tastatur zu schauen; das schont deine Nacken- und Schultermuskeln. Lernst du sie, wirst du wahrscheinlich deiner Tastatur einiges abverlangen.

Stelle den Kontrast und die Helligkeit deines Bildschirms so ein, daß du ausreichend gut und klar sehen kannst. Windows-Anwender: Probiere Systemsteuerung, Farben, LCD Standard. Schiebe den Monitor weit genug weg – meiner steht ca. 75 cm entfernt.

Deine Handgelenke werden überanstrengt, wenn du sie nicht schützt. Ich habe ein Handtuch mehrfach gefaltet und es direkt vor meine Tastatur gelegt – das ist eine gute Handstütze.

Der Ventilator und die Festplatte sind laut, auch bei Geräten mit niedrigem Geräuschpegel (Low Noise). Ich bin besonders empfindlich für die Hochfrequenzöne der Festplatte. Stelle den Rechner selbst so weit wie möglich weg, zum Beispiel unter den Schreibtisch – du kannst Verlängerungskabel kaufen, um den Monitor an den Computer anzuschließen (sowohl für Strom, als auch für Daten). Nehme zur Geräuschisolierung Decken oder Handtücher o.ä., natürlich ohne die Luftzufuhr zu blockieren. Es sollte zwar eine Kühlung geben, aber hier im Norden braucht sie nicht so stark zu sein wie im Süden.

Schaff dir einen guten Bürostuhl an, wenn du längere Zeit vor dem Computer sitzt. In wirklichen Druckzeiten habe ich bis zu 12 Stunden hintereinander dort gesessen. Laß du das lieber sein! Zum Schluß wurde ich krank. Bewege dich gelegentlich, mache eine Pause, einen Spaziergang, spüle, öffne ein Fenster (Nein, nicht in dem gleichnamigen Programm, sondern das im Haus, mit Glas drin!!!), mache Liegestützen und Kniebeugen, mache einfach etwas ganz anderes.

Höre auf deinen Körper. Egal wie aufregend es ist. Wenn dein Körper sagt, er braucht eine Pause, gib ihm eine Pause. Wenn du das nicht machst, wirst du früher oder später dazu gezwungen sein.

Wenn du viel auf den Bildschirm schaut, mach gelegentlich ein paar Augenübungen. Schaue ganz nach rechts, dann nach links, dann hoch und runter, dann links nach unten, links nach oben, nach rechts unten, nach rechts oben, sowohl mit offenen als auch mit geschlossenen Augen – gerade so viel, daß du deine Muskeln spürst. Und atme gleichzeitig tief durch.

Nun, nachdem du etwas trainiert hast, solltest du rundherum gucken, sowohl mit geschlossenen als

auch mit geöffneten Augen – Kinder sehen dabei gern zu. Diese Übungen sind simpel, aber effektiv. Zwei Minuten pro Tag wirken Wunder.

Glaube ja nicht, daß die meisten Bildschirmhersteller hauptsächlich an deine Gesundheit denken, auch wenn die "Grüne Vermarktung" in der Industrie sehr populär geworden ist.

286, 8; 15
286'er, 48

32-Bit Zugriff, 39
386 enhanced mode, 38

ANSI.SYS, 18
AT-Bus, 25
Auslagerungsdatei, 38

Bildschirm, blau, 22
BREAK=, 20

CD-ROM, 17; 21
CHKDSK/F, 27; 50
CMOS, 52
codepage, 11
COMMAND.COM, 20
consol, 10
COUNTRY=, 18

Default, 4
DEFRAG, 26; 29; 44
Device, 9; 10; 11; 14
Devicehigh, 15
DISPLAY.SYS, 18
DOS=HIGH, 15
DOS=UMB, 17
DoubleBuffering, 25

editor, 35
EMM386.EXE, 15
EMS, 16
enhanced mode, 39

FastDisk, 39
FASTOPEN, 20
FAT, 34
FCBS=, 18
FILES=, 18
fragmentieren, 26

HIMEM.SYS, 14; 44

interrupt, 19

Konsole, 10

LASTDRIVE=, 20
LH, 15
LoadHigh, 15

monochrome region, 16

Multitasking, 39

NOEMS, 15

page frame, 16; 41
PageFrame, 40
Parameter, 14
PATH, 21
Protected Mode, 10; 39
PS/2, 10

RAMdisk, 22
RAMDRIVE.SYS, 17
RAM-Platte, 17
Real Mode, 8
Real Mode, 10
REM, 11
resident, 49
resident, 9

SCANDISK, 44
SETVER, 18
shell=, 20
SMARTDRV.SYS, 17
SPEEDISK, 26
Spiele, 43
stack overflow, 19
STACKS=, 19
standalone, 14
standard mode, 38
swapfile, 28; 38
switch, 14
Systemdateien, 14

translation buffers, 42; 49
TSR, 9

UMB, 9

V86 Modus, 39
virtual disk, 17

Windows, 17; 22

Zeichensatztafel, 11
Zugriffszeit, 26

Nutze Deinen PC optimal