

DOS EXTRA

18,- DM 145 öS 18,- sfr 26,50 Fm

Sonderausgabe DOS EXTRA

Nr. 9 1989/90

ISSN 0934 - 2842

DMV-Verlag

Eine Publikation der
DOS INTERNATIONAL

MS-DOS-Spezialitäten

- **Filter, Pipes, Umleitungen**
- **Debug**
- **Edlin**

Hardware im Detail

**Prozessor, Grafikkarte,
Festplatte, Monitor, Drucker**

**Einsteiger-Tips
für die Praxis**

Auf einen Blick

- **MS-DOS-Befehle**
- **Batch-Kommandos**
- **Fehlermeldungen**

**Nützliche Listings
schnell abgetippt**

**Alles für
Einsteiger**





CPS Computertechnik GmbH
Hamburger Str. 283 • 3300 Braunschweig
Fax (05 31) 33 06 61 • BTX *20088 1490#

Ladenöffnungszeiten: Mo-Fr 9-18 Uhr
Sa 9-13 Uhr
Studentenrabatte geg. Nachweis a. Anfrage !

Tel. (05 31) 33 06 63

CPS AT



CPS AT I (siehe Test DOS 1/89)

80286-12 CPU 8/12 MHz. 1 MB RAM,
bis 4MB on board. 0-Wait-State,
Sockel für Co-Prozessor, EMS, Phoenix Bios
64 K ROM mit Setup-Funktion, Uhr u. Kalender,
1 ser./2 par./Game-Port, Herc. komp. Grafikkarte
20 MB Festplatte 38 ms Zugriffszeit,
1 Lw 5,25" 1,2 MB und 1 Lw 3,5" 720KB/1,44MB,
Cherry Tastatur mit sep. Cursorblock,
14" Flat Screen Monitor s/w oder amber,
MS-DOS 4.01 dt. und GW-Basic 3540,-

CPS AT I Mini-Tower

80286-12 CPU 8/12 MHz. 1 MB RAM,
bis 4 MB on board. EMS, 0-Wait-State,
sonst wie AT I 3660,-

CPS AT I Tower

80286-12 CPU 8/12 MHz. 1 MB RAM,
bis 4 MB on board, 0-Wait-State, EMS,
Sockel für Co-Prozessor 80287,
Bios 64 K ROM mit Setup-Funktion,
Uhr u. Kalender, 1 ser./ 2 par./ Game-Port,
Herc. komp. Grafikkarte, 20 MB Festpl. 38 ms,
1 Lw 5,25" 1,2 MB und 1 Lw 3,5" 720KB/1,44 MB
Cherry Tastatur mit sep. Cursorblock,
14" Flat Screen Monitor s/w oder amber,
MS-Dos 4.01dt. und GW-Basic 4175,-

CPS 386 SX

80386 SX CPU 16 MHz, 0-Wait-State,
2 MB RAM, bis 4 MB on board, Sockel für 80387,
Herc. komp. Grafikkarte mit Printerport,
1 ser./ 1 par./ Game-Port,
1Lw 5,25" 1,2 MB und 1 Lw 3,5" 720KB/1,44 MB,
40 MB Festplatte 28 ms Zugriffszeit,
Cherry Tastatur mit sep. Cursorblock,
14" Flat Screen Monitor s/w oder amber,
MS-DOS 4.01 dt. und GW-Basic 4175,-

CPS AT III/ Cache

80386-25 CPU 25 MHz, 2 MB RAM,
bis 16 MB on board, 0-Wait, Sockel für 80387,
2 par./2 ser./1 Gameport, Memory Expansion
Card, EMS, 32 KB Cache, 1Lw 5,25" 1,2 MB,
1Lw 3,5" 1,44 MB, 40 MB Festplatte 28 ms,
Cherry Tastatur, Tower-Gehäuse,
14" Flat Screen Monitor, Mouse,
MS-DOS 4.01 dt. und GW-Basic 7990,-

Technische Änderungen vorbehalten.
Gerätekonfigurationen nach Ihren Wünschen
veränderbar.

DISKETTEN



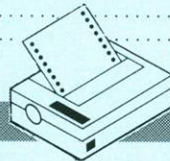
NN 2DD 3,5" 10 Stck. 15,-
NN 2DD 5,25" 10 Stck. 6,90
Magix MF 2DD 3,5" 24,50
Select MF 2DD 3,5" 33,50
Select MD 2DD 5,25" 19,-
Fuji MD 2DD 5,25" 27,-

FARB BÄNDER



STAR NL/NG/ND/NR-10, Stück 12,-
EPSON LX-800/LQ-500, Stück 8,-
PANASONIC KX-P, Stück 13,-
OKI ML 320 14,50
OKI ML 390 16,85
NEC 2200 13,50
NEC P6+/P7+ 15,30
Star LC 10 10,-

DRUCKER



OKI Microline 390 Centr. Sonderpreis a. Anfr.
OKI Microline 393 Centr. Sonderpreis a. Anfr.
OKI Microline 320 Sonderpreis a. Anfr.
STAR LC 10 Centr. 499,-
Epson LX 400 499,-
NEC P 6 Plus Centr. 1499,-
NEC P 7 Plus Centr. 1985,-
Centronics Drucker kabel 18,80

Weitere Drucker auf Anfrage

Wir liefern nur mit dt. Handbuch, Seriennummer und
Herstellergarantie !!! Drucker-Grauimporte mit engl.
Handbuch, ohne Seriennummer, ohne Herstellergarantie
sind bei uns ausgeschlossen.

COMMODORE

AMIGA 500 795,-
AMIGA 2000 1985,-
AMIGA 500 + Mon. 1084 1410,-
AMIGA 2000 + 20 MB Harddisk SCSI 2990,-
LW ext. AMIGA 500 3,5" 290,-
LW ext. AMIGA 2000 5,25" 419,-
LW int. AMIGA 2000 3,5" 220,-
RGB Monitor Commodore 1084 630,-
RGB Monitor Stereo Philips CM 8833 630,-
PC 20/III, 20 MB Festpl., MS-DOS 2300,-
PC/AT-Karte 2400,-
PC/XT-Karte 899,-
A 2058 2 MB RAM-Erweiterung 1180,-
A 501 512 KB RAM-Erweiterung 330,-
68020 Prozessorkarte Preis a. Anfr.
68030 Prozessorkarte Preis a. Anfr.

Nutzen Sie unseren Bequem-Kauf-Kredit!

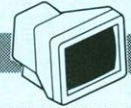


**weil Preis &
Leistung stimmen !**

COMMODORE

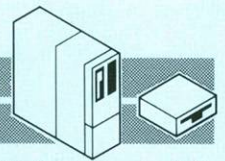
PC 10 III, 1 Laufwerk 360 K, Festplatte 20 MB,
EGA Grafikkarte, EGA Colormonitor 14" Quadram,
MS-DOS 3.3 / GW-Basic,
Drucker Panasonic KX-P 1081 2999,-

MONITORE



EGA Standard 815,-
EIZO 9060s 1855,-
NEC MULTISYNC 2 A 1550,-
TTL 14" Flat Screen m. Fuß sw/amber 235,-
TTL 12" Flat Screen m. Fuß sw/amber 180,-
Mitsubishi 1481 1398,-
weitere Monitore auf Anfrage

Festplatten und Laufwerke



SEAGATE ST 125-0, 20 MB, 38ms 549,-
SEAGATE ST 125-1, 20 MB, 28ms 558,-
SEAGATE ST 4096, 80 MB, 28ms 1610,-
SEAGATE ST 251-1, 40 MB, 28ms 999,-
Controller NCL 5525 HDD/FDD 290,-
Laufwerk 360 KB 5 1/4" TEAC 190,-
Laufwerk 1,2 MB 5 1/4" TEAC 215,-
Laufwerk 1,44 MB 3,5" TEAC 235,-

Sämtliche Angebote freibleibend, Zwischenverkauf
vorbehalten. Wir liefern an Nicht-Kaufleute nur per
UPS-Nachnahme. Ins Ausland nur per Vorkasse.
Fordern Sie unseren Gesamtkatalog gegen 3,-DM
in Briefmarken. Es gelten unsere allgemeinen
Geschäftsbedingungen.

Versand nur über Braunschweig

"Wer pffiff rechnet und jetzt
eine EDV-Anschaffung plant,
kann bei CPS eine Menge
Geld sparen und hat
3 weitere dicke Vorteile."



✗ 24 Monate Garantie auf CPS-Rechner
✗ 10 Tage Rückgaberecht auf Hardware
✗ Eigener Reparatur-Service

CPS Computertechnik GmbH
Braunhirschstr. 29 • 3100 Celle
Tel. 0 51 41/3 20 04 • Fax 0 51 41/ 38 14 22

CPS Computertechnik GmbH
Großbeerenstr. 5 • D-1000 Berlin 42
Telefon 0 30/ 7 05 30 55

Keine Angst vor dem bösen Wolf

Harmlos und unbedarft tritt Rotkäppchen dem bösen Wolf entgegen, ebenso harmlos und unbedarft ist wohl so mancher Einsteiger bei seinem ersten Kontakt mit dem neuen Computer. Da beruhigt auch keine Binsenweisheit wie »Aller Anfang ist schwer« oder »Ohne Fleiß kein Preis«, denn allzuoft stellt sich heraus, daß erstens nicht nur der Anfang, sondern erst recht das Weitere alles andere als einfach ist, und daß zweitens ohne kompetente Anleitung keine Preise zu gewinnen sind.

Doch schon bei den Gebrüdern Grimm gab's glücklicherweise den wackeren Jägersmann, der als Retter in der Not dem armen Rotkäppchen aus der Patsche half. Ebenso möchte dieses DOS Extra, das speziell auf Einsteigerfragen und -sorgen zugeschnitten ist, allen in die Bredouille geratenen PC-Newcomern hilfreich zur Seite stehen. Gemeinsam mit diesem DOS Extra sollte es keine Schwierigkeiten mehr bereiten, den »bösen Wolf« Computer zu zähmen und dazu zu bringen, daß er Ihnen aufs Wort gehorcht.

Nach der Lektüre dieses DOS Extra werden auch Schlagworte wie Controller, Interleave-Faktor, RAM oder ROM, die Sie aus den Werbesätzen der Hard- und Software-Anbieter kennen, keine Fremdworte mehr für Sie sein. Sie erhalten fundierte Grundlageninformationen zum Betriebssystem MS-DOS, Hilfestellungen bei den ersten Programmierschritten und einen Einblick in Ihre Hardware-Konfiguration.

Vor den Erfolg hat der PC – und das soll nicht verschwiegen werden – eine Portion Fleiß gesetzt. Fachliche Begleitung anstelle eines entmutigenden Alleingangs sorgt dafür, daß sich mit dem aufzuwendenden Engagement auch baldmöglichst der wohlverdiente Erfolg und damit nicht zuletzt der Spaß einstellt.

Unter dieser Anleitung erweisen sich selbst die allerersten Schritte als gar nicht so schwierig und »gefährlich« wie allgemein befürchtet, und zu Recht dürfen Sie auf Ihre eigenhändig formatierte und erfolgreich eingerichtete Festplatte stolz sein.

Dieses DOS Extra gibt Ihnen auch eine Anleitung zum Programmieren Ihrer ersten Autoexec-Datei und anderer nützlicher Stapeldateien. Um die Arbeit mit dem Betriebssystem MS-DOS – um die kein privater PC-Anwender herumkommt – übersichtlich zu gestalten, finden Sie in einer ausführlichen Referenztafel alle MS-DOS-Befehle mit einer kurzen Beschreibung aufgelistet. Welche Befehle brauchen Sie, welche nicht? Auch darüber gibt ein Beitrag Auskunft.

Um auch dem Informationsbedürfnis hardwareinteressierter PC-Neulinge gerecht zu werden, weist je ein Beitrag in den Aufbau und die Funktion von Prozessor, Grafikkarte, Speichermedium, Tastatur und Drucker ein.

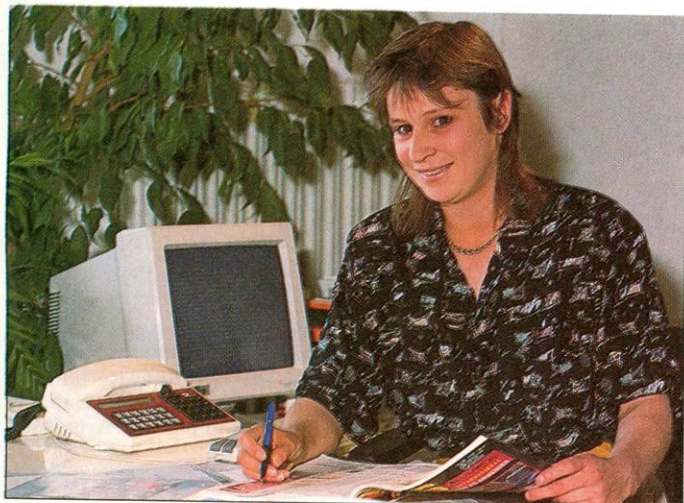
Nicht zuletzt bietet Ihnen dieses DOS Extra kleine, hilfreiche Programme zum Abtippen an, und schon verfügen Sie zum Nulltarif über eine Uhr am PC, über eine Minidatenbank oder einige nützliche Utilities.

Also – keine Angst mehr vor dem »bösen PC«, denn bei näherem Hinsehen entpuppt er sich als »Schaf im Wolfspelz«.

Ihre

Eva Hierlmeier

Eva Hierlmeier



MS-DOS

Die Entstehung von MS-DOS	6
Dateiverwaltung unter MS-DOS	10
Referenz: Alle MS-DOS-Befehle bis Version 4.0	14
Referenz: Batch-Befehle	25
Referenz: config.sys	28
Vorsicht, Umleitung!	30
– Umleitungen, Filter und Pipes unter MS-DOS	
Was Ihnen MS-DOS sagen will	35
– Fehlermeldungen unter MS-DOS	
Die wichtigsten MS-DOS-Befehle	38
Der ganz persönliche Computer	40
– Config- und Autoexec-Datei	
Der Debug-Quicky	44
– Tips zum schnellen Abtippen von Listings	
Am Anfang und für zwischendurch: Edlin	46

Hardware

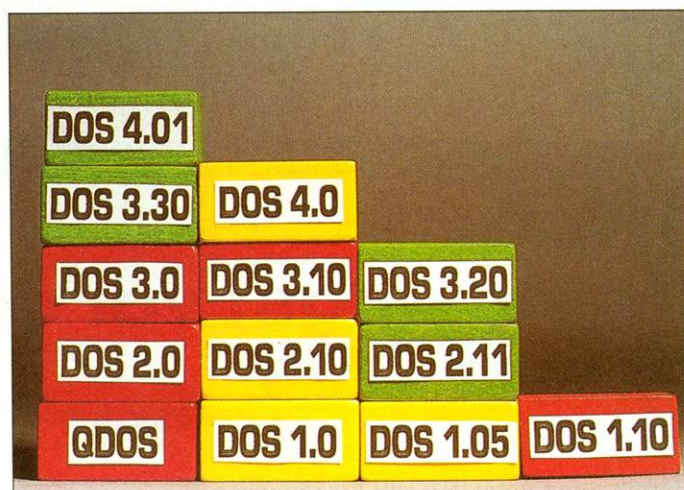
Die Mikroprozessoren der PC-Welt	52
Die Grafikstandards der PC-Welt	60
Das Geheimnis der Datenspeicherung	67
Alles über Ihre Festplatte	70
Wie sag' ich's meinem Rechner?	74
– Aufbau und Funktion der Tastatur	
Punkt für Punkt zum Erfolg	80
– So funktionieren Drucker	
Hardware für Spieler	83

Programmierung

Welche Sprache für welchen Zweck?	86
Das Programmier-Lexikon	90

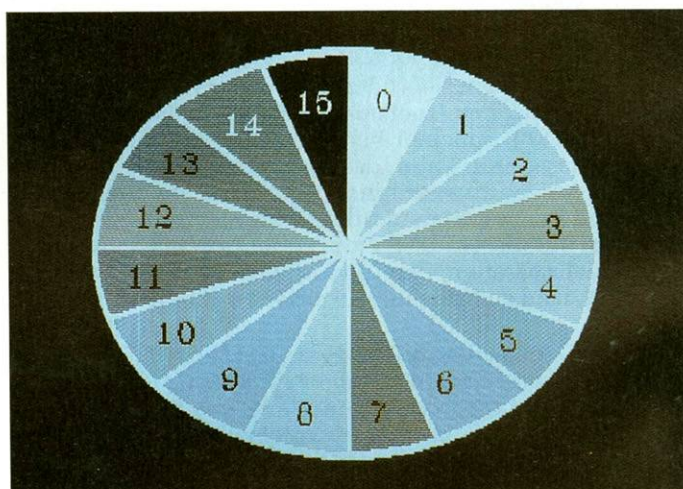
Grundlagen

Das Labyrinth der technischen Daten	92
Kleiner Einsatz, große Wirkung	94
– Shareware und Public Domain	
Power on!	98
– Wie funktioniert ein PC	
Einsteiger-Lexikon	102



Ohne Betriebssystem geht bei Ihrem PC gar nichts, und das Betriebssystem für Ihren Personalcomputer heißt MS-DOS. Dieses MS-DOS hat bereits eine lange Vorgeschichte. Interessieren Sie sich für den Werdegang Ihres PCs?

Seite 6



Es muß nicht immer Farbe sein, denn mit unserem Listing »ega2hgc.pas« werden Bilder, die im EGA-Modus erzeugt wurden, einfach auf die Hercules-Karte übertragen: Die Farben Ihrer EGA-Bilder erscheinen als verschiedene Schraffuren

Seite 153



Computer, Prozessor, Festplatte, Drucker, Grafikkarte und Tastatur, das sind die Komponenten, aus denen sich eine komplette PC-Ausstattung zusammensetzt. Was für Sie bei der Arbeit mit diesen Geräten wissenswert ist, lesen Sie auf

Seite 102

Werkstatt

Die MS-DOS-Werkstatt	108
Die Befehlsstapel-Werkstatt	118

Tips & Tricks

Die Tips-und-Tricks-Fundgrube	126
Tips und Tricks zum Prompt-Befehl	130

Listings

Wie sag' ich's meiner Batchdatei?	134
Datenstapelei	136
– Eine Minidatenbank als Batchdatei	
Stapeldatei mit Pfiff	141
– MS-DOS-Befehlseingabe mit mehr Komfort	
»delq.com« macht Datenlöschen sicher	143
– Löschen ohne Risiko	
Pulldown-Komfort unter GW-Basic	145
Briefbogen-Druckerei für daheim	146
Analoguhr für Selfmade-Designer	147
Grafikkonvertierung	153
– Schraffur statt Farbe	

Rubriken

Editorial	3
– Keine Angst vor dem bösen Wolf	
ASCII-Zeichentabelle	48
Ansi-Steuersequenzen	50
– Tastatur und Bildschirm im Griff	
Bücher	65
SCheck 2.02	158
– Checksummer für sichere Listing-Eingabe	
Impressum	162
Inserenten	162

POWER ON!

DAS PASSIERT ALLES
BEIM EINSCHALTEN
IHRES PC

Eine Einführung von Werner Küstenmacher, m.a.

und MS-DOS

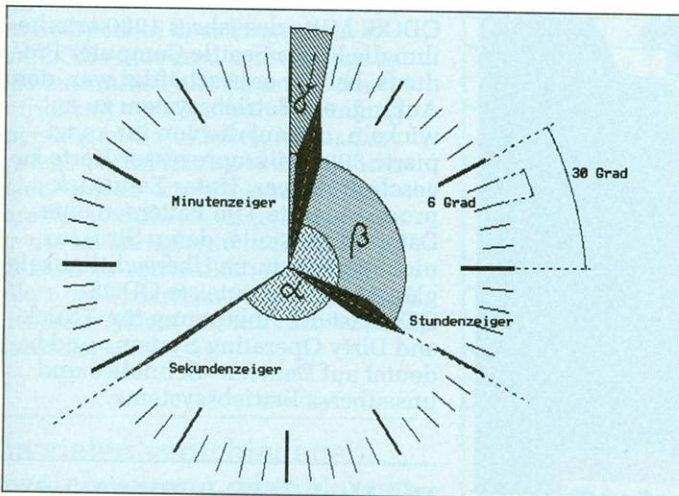
Wenn Sie mit dem berühmten »Big red switch« Ihren PC an das Weltenergieversorgungsnetz anhängen, geschieht danach weit mehr als nur beim Anknipsen einer Glühbirne.

ES IST DER START
EINES VERZICK-
TEN PROZESSES...

NACH DER LEKTURE DIESER SEITEN
WERDEN SIE ES DOPELT ZU SCHÄTZEN
WISSEN, WENN SICH IHR PEE-CEE BEIM
NÄCHSTEN EINSCHALTEN ZUR STELLE
MELDET

»Power on« und los geht's! Auf spaßige Weise vermittelt Ihnen eine amüsante Cartoonserie den Einstieg in die Welt der Personalcomputer. Wer auf diese Weise keine Lust aufs Computern bekommt, ist selbst schuld

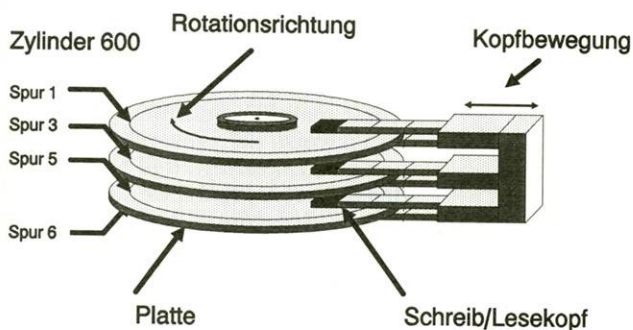
Seite 98



Für wen spielt Zeit schon keine Rolle? Verpacken Sie doch Ihre Uhrzeit in ein individuelles Design. Malen Sie mit einem Grafikprogramm eine Uhr, und unser Listing »clock.pas« erweckt sie zu Leben

Seite 147

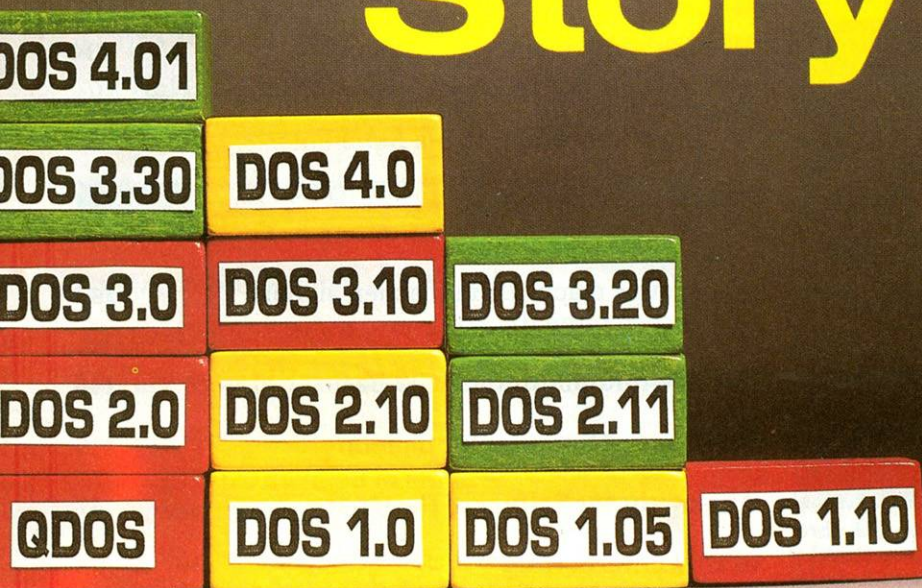
Rund um die Festplatte



Eine Festplatte gehört schon fast zur Grundausstattung eines Personalcomputers, doch wer weiß schon, wie solch ein Massenspeicher funktioniert. Über die Interna eines Festplattenlaufwerkes, wie Sie es in Betrieb nehmen und Tips zum Umgang vermittelt ein Beitrag auf

Seite 70

Die MS-DOS Story



Das Betriebssystem MS-DOS hat sich als Standard im PC-Bereich etabliert. Doch seit seiner Einführung im Jahre 1981 durchlief es viele Entwicklungsstadien. Es wurde immer wieder an die Weiterentwicklung der Hardware und an die gesteigerten Anwendersprüche angepaßt. Der Weg von der ersten Betriebssystemfassung über MS-DOS 1.0 bis zur aktuellen Version 4.0 ist eine interessante und bemerkenswerte Geschichte.

Die Entwicklungsgeschichte des MS-DOS

MS-DOS-Version	Veröffentlichung	Funktionsumfang
1.00	1981	Ursprungs-MS-DOS mit zeilenorientierter Kommandoebene und Basic
1.05	1981	Beseitigung der Fehler im Basic-Interpreter
1.10	1982	Unterstützung doppelseitiger Disketten
2.00	1983	Unterstützung von Festplatten
2.10	1983	MS-DOS-2.0-Version ohne Fehler
3.00	1984	Unterstützung von 1,2-MByte-Laufwerken und erweiterter Zeichensatz
3.10	1984	Einsatzfähig in lokalen Netzwerken
3.20	1986	Unterstützung von 720-KByte-Mikrodisketten (3,5 Zoll)
3.30	1987	Anpassung an PS/2 und Unterstützung von 1,44-MByte-Mikrodisketten
4.00	1988	Unterstützung von Festplatten größer 32 MByte und neue Oberfläche

Tabelle 1. Die Version 4.0 steht am vorläufigen Ende der Reihe von MS-DOS-Fassungen für PCs

MS-DOS (Microsoft Disk Operating System) ist das grundlegende Betriebssystem für die PC-Familie von IBM und alle kompatiblen Computer. Es wird Diskettenbetriebssystem genannt, weil der größte Teil seiner Arbeit darin besteht, durch Umorganisieren von Daten Disketten und Diskettendateien zu verwalten. Doch MS-DOS ist nicht allein auf Aufgaben, die sich mit Disketten und deren Dateien befassen, beschränkt. So organisieren seine Programme, aus denen es besteht, den Datenfluß zwischen den Bausteinen des Computers und den peripheren Geräten wie Diskettenlaufwerk, Bildschirm und Tastatur und kommunizieren mit den eingesetzten Anwendungen, um einen störungsfreien und schnellen Ablauf zu garantieren.

Als Urahn des heute auf PCs eingesetzten MS-DOS gilt das vor neun Jahren von Tim Patterson entwickelte QDOS. Mitte des Jahres 1980 erteilte ihm die Firma Seattle Computer Products, bei der er beschäftigt war, den Auftrag, ein Betriebssystem zu entwickeln, das auf die von ihr konzipierte 8086-Mikroprozessor-Karte zugeschnitten war. Unter Zeitdruck programmierte Tim Patterson eine Datenschnittstelle, deren Struktur nicht gerade durch Übersichtlichkeit glänzte, und nannte sie QDOS. QDOS ist die Abkürzung für »Quick and Dirty Operating System« und bedeutet auf Deutsch »Schnelles und unsauberes Betriebssystem«.

Die Verbindung IBM-Intel-Microsoft sorgte für den Aufstieg von MS-DOS

Auf der Basis dieses Programms baute Tim Patterson den Nachfolger namens 86-DOS auf, den man Ende 1980 veröffentlichte. Die neue Version enthielt im Gegensatz zu QDOS kaum noch Fehler und gefiel durch gesteigerte Effizienz. Ein entscheidender Vorteil, über den 86-DOS verfügte, förderte den Verkauf: Der Autor hatte die Schnittstellen beziehungsweise Betriebssystemfunktionen so strukturiert, daß sie mit denen des Betriebssystems CP/M identisch waren. Aufgrund dieser Eigenschaften konnten die Software-Firmen ihre für CP/M gedachten Produkte problemlos an das 86-DOS anpassen. Darüber hinaus verschwanden die Einschränkungen, denen alle Programme, die unter CP/M liefen, unterworfen waren.

Die 1981 mit der Veröffentlichung des PCs getroffene Entscheidung des Elektronik-Riesen IBM, in seinen Computern den Intel-Chip 8086 zu

verwenden, hat wesentlich zum Erfolg von MS-DOS beigetragen. IBM suchte nach einem leistungsfähigen und vielseitigen Betriebssystem für seine Computer. Neben einigen anderen Bewerbern wählte IBM auch Microsoft als einen möglichen Kandidaten für die Position des Software-Lieferanten aus.

Nachdem Microsoft schon vor der Veröffentlichung des 86-DOS die Rechte für dieses Software-Produkt erworben hatte, verfügte man über eine solide Grundlage für die Entwicklung eines IBM-Betriebssystems. Unter der Federführung des inzwischen »übergelaufenen« Tim Patterson und mit der Hilfe eines ersten PC-Prototyps machten sich die Microsoft-Programmierer an die Arbeit und stellten den IBM-Managern nach kurzer Zeit die erste Version des »neuen« Betriebssystems vor: MS-DOS Version 1.0. Die IBM-Leute fanden Gefallen an dem Programm, erwarben die Lizenzen und stellten es 1981 zusammen mit ihrem PC der Öffentlichkeit vor. Das war der Anfang einer langen Geschichte, in der das Betriebssystem immer wieder modernisiert und an die verbesserte Hardware der Peripherie angepaßt wurde. Um die Leistungsentwicklung von MS-DOS zu verdeutlichen, faßt Tabelle 1 die Produktmerkmale und den zeitlichen Ablauf der Microsoft-Reihe zusammen.

Die vielen verschiedenen Bezeichnungen oder Namen für MS-DOS müssen Sie nicht verwirren

Lassen Sie sich von der Vielzahl der auf dem Markt befindlichen MS-DOS-Bezeichnungen von AT&T, Epson oder Tandon nicht verwirren. Selbst IBM hat den Namen des Betriebssystems nicht von Microsoft übernommen, sondern einen eigenen geschaffen: PC-DOS. Aber den einzigen Unterschied zwischen MS-DOS und PC-DOS finden Sie in der unterschiedlichen Bezeichnung einiger Dateien; funktionell unterscheiden sich die Programme nicht. Das einzige Kriterium, das über den Zeitpunkt der Entwicklung einer Betriebssystemfassung und somit über den Leistungsumfang Auskunft gibt, ist die Versionsnummer.

Die erste Version von MS-DOS genügte den Ansprüchen der Anwender schon bald nicht mehr, ein Nachfolger mußte her. Die Programmierer waren sich darüber einig, daß es zu den wichtigsten Kriterien bei einer Weiterentwicklung des Betriebssy-

Die Funktionen der MS-DOS-Versionen

Zugrundeliegende MS-DOS-Versionen												Bedeutung
Funktion	Unterf.	1.0	1.1	2.0	2.1	2.25	3.0	3.1	3.2	3.3	4.0	
0hex		+	+									Beenden eines Programms
Fhex		+	+									Öffnen einer Datei
10hex		+	+									Schließen einer Datei
11hex		+	+									Ersten FCB-Eintrag suchen
12hex		+	+									Nächsten FCB-Eintrag suchen
13hex		+	+									Löschen einer Datei
14hex		+	+									Sequentiell Lesen
15hex		+	+									Sequentiell Schreiben
16hex		+	+									Erzeugen einer Datei
17hex		+	+									Umbenennen einer Datei
19hex		+	+									Laufwerk ermitteln
18hex		+	+									Aktuelle Laufwerksdaten lesen
1Chex		+	+									Laufwerksdaten lesen
21hex		+	+									Wahlfrei lesen
22hex		+	+									Wahlfrei schreiben
23hex		+	+									Größe einer Datei ermitteln
24hex		+	+									Datensatznummer setzen
27hex		+	+									Wahlfrei Block lesen
28hex		+	+									Wahlfrei Block schreiben
29hex		+	+									Dateinamen übersetzen
2Fhex		-	-	+	+	+	+	+	+	+	+	Disk-Transfer-Adresse lesen
30hex		-	-	+	+	+	+	+	+	+	+	MS-DOS-Versionsnummer ermitteln
31hex		-	-	+	+	+	+	+	+	+	+	Programm ohne Löschen beenden
33hex		-	-	+	+	+	+	+	+	+	+	Control-C-Check
35hex		-	-	+	+	+	+	+	+	+	+	Lesen eines Interrupt-Vektors
36hex		-	-	+	+	+	+	+	+	+	+	Platz auf Laufwerk ermitteln
38hex	0	-	-	+	+	+	+	+	+	+	+	Landes-Code lesen
38hex	x	-	-	-	-	-	+	+	+	+	+	Landes-Code setzen (x=Land)
39hex		-	-	+	+	+	+	+	+	+	+	Verzeichnis einrichten
3Ahex		-	-	+	+	+	+	+	+	+	+	Verzeichnis entfernen
3Bhex		-	-	+	+	+	+	+	+	+	+	Unterverzeichnis wechseln
3Chex		-	-	+	+	+	+	+	+	+	+	Datei über Handle erzeugen
3Dhex		-	-	+	+	+	+	+	+	+	+	Datei über Handle öffnen
3Ehex		-	-	+	+	+	+	+	+	+	+	Datei über Handle schließen
3Fhex		-	-	+	+	+	+	+	+	+	+	Gerät über Handle lesen
40hex		-	-	+	+	+	+	+	+	+	+	Gerät über Handle beschreiben
41hex		-	-	+	+	+	+	+	+	+	+	Datei über Handle löschen
42hex		-	-	+	+	+	+	+	+	+	+	Position des Dateizeigers ändern
43hex		-	-	+	+	+	+	+	+	+	+	Dateiattribut ermitteln/festlegen
44hex	0	-	-	+	+	+	+	+	+	+	+	IOCTL: Daten lesen
44hex	1	-	-	+	+	+	+	+	+	+	+	IOCTL: Daten setzen
44hex	2	-	-	+	+	+	+	+	+	+	+	IOCTL: Zeichen lesen
44hex	3	-	-	+	+	+	+	+	+	+	+	IOCTL: Zeichen schreiben
44hex	4	-	-	+	+	+	+	+	+	+	+	IOCTL: Block lesen
44hex	5	-	-	+	+	+	+	+	+	+	+	IOCTL: Block schreiben
44hex	6	-	-	+	+	+	+	+	+	+	+	IOCTL: Eingabe-Status
44hex	7	-	-	+	+	+	+	+	+	+	+	IOCTL: Ausgabe-Status
44hex	8	-	-	-	-	-	+	+	+	+	+	IOCTL: auswechselbar ?
44hex	9	-	-	-	-	-	-	+	+	+	+	IOCTL: Block-Remote
44hex	10	-	-	-	-	-	-	+	+	+	+	IOCTL: Handle-Remote
44hex	11	-	-	-	-	-	-	+	+	+	+	IOCTL: Wiederholungsrate ändern
44hex	12	-	-	-	-	-	-	-	+	+	+	Generic IOCTL for handles
44hex	13	-	-	-	-	-	-	-	-	+	+	Generic IOCTL for devices
44hex	14	-	-	-	-	-	-	-	-	+	+	Laufwerk ermitteln
44hex	15	-	-	-	-	-	-	-	-	+	+	Laufwerk festlegen
45hex		-	-	+	+	+	+	+	+	+	+	Datei-Handle verdoppeln

Tabelle 2. Während der Entwicklung wurden viele neue Funktionen hinzugefügt und veraltete Betriebssystemaufrufe durch modernere ersetzt (Teil 1)

Funktion	Unterl.	1.0	1.1	2.0	2.1	2.25	3.0	3.1	3.2	3.3	4.0	Bedeutung
46hex		-	-	+	+	+	+	+	+	+	+	Doppelten Handle erzwingen
47hex		-	-	+	+	+	+	+	+	+	+	Aktuelles Unterverzeichnis lesen
48hex		-	-	+	+	+	+	+	+	+	+	Arbeitsspeicher reservieren
49hex		-	-	+	+	+	+	+	+	+	+	Reservierten Speicher freigeben
4Ahex		-	-	+	+	+	+	+	+	+	+	Speicherblock festlegen
4Bhex	0	-	-	+	+	+	+	+	+	+	+	Programm laden und starten
4Bhex	3	-	-	+	+	+	+	+	+	+	+	Overlay laden
4Chex		-	-	+	+	+	+	+	+	+	+	Prozeß beenden
4Dhex		-	-	+	+	+	+	+	+	+	+	Return-Code (Prozeß) ermitteln
4Ehex		-	-	+	+	+	+	+	+	+	+	Erste Datei finden
4Fhex		-	-	+	+	+	+	+	+	+	+	Nächste Datei finden
54hex		-	-	+	+	+	+	+	+	+	+	Verify-Status ermitteln
56hex		-	-	+	+	+	+	+	+	+	+	Datei umbenennen
57hex		-	-	+	+	+	+	+	+	+	+	Datei-Uhrzeit/Datum setzen/lesen
58hex		-	-	-	-	-	+	+	+	+	+	Reservierungsstrategie setzen
59hex		-	-	-	-	-	+	+	+	+	+	Erweiterte Fehlerinformation holen
5Ahex		-	-	-	-	-	+	+	+	+	+	Datei vorübergehend erzeugen
5Bhex		-	-	-	-	-	+	+	+	+	+	Neue Datei erzeugen
5Chex	0	-	-	-	-	-	+	+	+	+	+	Sperren
5Chex	1	-	-	-	-	-	+	+	+	+	+	Freigeben
5Ehex	0	-	-	-	-	-	-	+	+	+	+	Name der Maschine lesen
5Ehex	2	-	-	-	-	-	-	+	+	+	+	Drucker-Setup
5Fhex	2	-	-	-	-	-	-	+	+	+	+	Eintrag aus Assign-Liste lesen
5Fhex	3	-	-	-	-	-	-	+	+	+	+	Eintrag in Assign-Liste vornehmen
5Fhex	4	-	-	-	-	-	-	+	+	+	+	Eintrag in Assign-Liste löschen
62hex		-	-	-	-	-	+	+	+	+	+	PSP-Adresse lesen
63hex		-	-	-	-	-	+	-	-	-	-	Byte-Tabelle lesen
65hex		-	-	-	-	-	-	-	-	+	+	Erweiterten Landes-Code holen
66hex		-	-	-	-	-	-	-	-	+	+	Code-Page setzen/lesen
67hex		-	-	-	-	-	-	-	-	+	+	Handle-Anzahl bestimmen
68hex		-	-	-	-	-	-	-	-	+	+	Datei übergeben

(+) = Funktion wird unterstützt; (-) = Funktion wird nicht unterstützt;
(leeres Feld) = Funktion veraltet; sollte durch Aufruf der moderneren Version umgangen werden

Tabelle 2. Während der Entwicklung wurden viele neue Funktionen hinzugefügt und veraltete Betriebssystemaufrufe durch modernere ersetzt (Teil 2)

stems gehört, die Kompatibilität zu den Vorgängerversionen zu wahren. Was bedeutet nun kompatibel, ein Wort, das Sie in der Welt der Computer ständig verfolgen wird. Die Hauptbedeutung von Kompatibilität

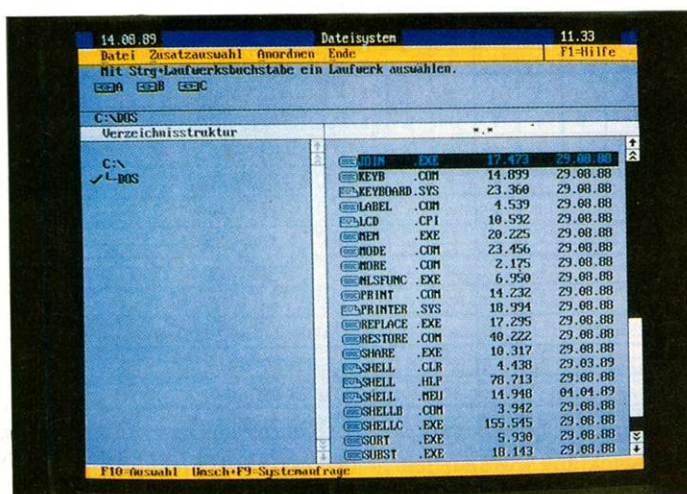
besteht hier abgesehen von einigen technischen Faktoren darin, in welchem Ausmaß Disketten und Daten von seinem System in ein anderes übernommen werden können. Sind zwei Systeme voll kompatibel, kön-

nen Sie deren Daten beliebig austauschen. Sind hingegen zwei Systeme inkompatibel, wie beispielsweise ein Heimcomputer wie der Commodore 64 und ein IBM PC, so ist dies nicht möglich, da ein System die Dateien des anderen nicht lesen, also nichts damit anfangen kann. Die große Verbreitung des Betriebssystems MS-DOS gewährleistet somit diese begehrte Kompatibilität, da es auf so vielen verschiedenen Personalcomputern läuft.

Diese Leitlinie der Programmierer dieses Betriebssystems wurde dank ausgiebiger Planung bei der Entwicklung der späteren Versionen eingehalten. Der Nachteil dieses Konzepts liegt allerdings darin, daß das Betriebssystem aufgrund der Kompromisse, die aus Rücksicht auf die Kompatibilität geschlossen werden müssen, Beschränkungen unterliegt. So konnte man die interessanten Fähigkeiten der 80386-Prozessoren bei der Entwicklung von MS-DOS nicht berücksichtigen, da sonst die Kompatibilität gefährdet gewesen wäre. Dieser negative Aspekt fällt aber nicht besonders ins Gewicht. Es ist wesentlich wichtiger, daß die Programme, die unter Berücksichtigung der vom Hersteller gegebenen Konventionen auf den Vorläufern ausgearbeitet wurden, ohne Änderungen unter dem aktuellen Betriebssystem lauffähig sind. Wie konsequent dieses Verfahren weitergeführt wurde, sehen Sie in Tabelle 2, die die Funktionen des Interrupt 21hex auflistet. Falls Sie Programme schreiben wollen, die unter mehreren MS-DOS-Versionen lauffähig sind, beachten Sie bitte die Aufstellung. Alle Funktionen, die nicht in der Tabelle enthalten sind, haben sich nicht verändert.

MS-DOS ist abwärts kompatibel, neuere Versionen unterstützen die älteren

MS-DOS hat sich mit seinen unterschiedlichen Versionen immer an der Weiterentwicklung der Hardware orientiert. Die Version 1.00 beinhaltet einen Basic-Interpreter und arbeitet nur mit einseitigen Diskettenlaufwerken zusammen, die eine Kapazität von 160 KByte besitzen. Das Betriebssystem kennzeichnet alle angelegten Dateien ausschließlich mit dem aktuellen Datum. Da die Nachfrage nach größerer Speicherkapazität stieg, veröffentlichte Microsoft nach der Version 1.05, die gravierende Fehler im Basic-Interpreter beseitigte, MS-DOS 1.10. Diese Version unterstützt doppelseitige Disketten mit einer Kapazität von 320



MS-DOS hat sich im Laufe der Zeit an die gestiegenen Ansprüche angepaßt

KByte und markiert Dateien neben dem Datum mit der aktuellen Zeit. Die Einführung des IBM-XT mit einer 10-MByte-Festplatte erforderte umfassendere Funktionen. Mit der Fassung 2.00 wurde diesem Umstand Rechnung getragen: Jedes Speichermedium konnte ab diesem Zeitpunkt mit Hilfe von Unterverzeichnissen übersichtlich eingeteilt werden, und der Dir-Befehl wurde um seiten- und spaltenweises Auflisten des Inhaltsverzeichnis erweitert.

Die bald folgende Version 2.10 beseitigte einige Fehler ihres Vorgängers. Ansonsten unterscheidet sie sich nicht von ihrem Vorläufer. Im Jahre 1984 erschien mit der Veröffentlichung des IBM-AT die MS-DOS-Version 3.0. Sie unterstützt 1,2-MByte-Disketten und macht es möglich, fremdsprachige Zeichen mit ASCII-Codierungen größer als 127 darzustellen. Die Fähigkeit, Dateien zeitweilig zu sperren, deutet erste Netzwerkmerkmale an, die jedoch erst mit der Version 3.10 vollständig umgesetzt wurden.

Mit der Version 3.20 führte man 1986 eine MS-DOS-Fassung ein, die 3 1/2-Zoll-Disketten mit einer Kapazität

von 720 KByte bearbeiten kann. Bei der Publikation der PS/2-Modelle 1987 paßten die Entwickler das Betriebssystem an und erweiterten die Diskettenformate um das neue 1,44-MByte-Laufwerk. Die im Jahre 1988 vorgestellte Version 4.0 unterstützt Festplatten mit einer Kapazität von mehr als 32 MByte und stellt eine neue Benutzeroberfläche zur Verfügung, die nicht mehr zeilenorientiert arbeitet (Bild).

Um größere Festplatten verwenden zu können, muß man sich bei älteren Versionen mit Partitionen behelfen. Dabei wird die Festplatte in mehrere logische Laufwerke unterteilt, die dann mit unterschiedlichen Laufwerksbuchstaben angesprochen werden.

MS-DOS ist ein vielseitig einsetzbares Betriebssystem mit interessanten Anwendungsmöglichkeiten. Es hat sich im Laufe der Jahre als zuverlässige Grundlage für sinnvollen Computereinsatz erwiesen und wird auch in naher Zukunft eine tragende Rolle im PC-Bereich spielen, wenn die Entwickler seine Fähigkeiten an die steigenden Ansprüche anpassen, wie dies beispielsweise durch die neue

Benutzeroberfläche von MS-DOS 4.0 geschehen ist.

Darüber hinaus spricht für MS-DOS, daß die meisten der bisher auf dem Markt befindlichen Programme nur einen Teil seiner Fähigkeiten nutzen. Allerdings liegen auch potentielle Nachfolger beziehungsweise Konkurrenten schon »auf der Lauer«.

Betriebssystemen wie OS/2, Windows oder anderen liegt im Vergleich zu MS-DOS ein modernes Konzept zugrunde, das beispielsweise Multitasking- und Multiuser-Fähigkeit beinhaltet. Sie versuchen, die Vorteile der aktuellen Mikroprozessoren wie zum Beispiel die des 80386 oder 80486 auszunutzen und zusätzlich eine bedienungsfreundliche Benutzeroberfläche zur Verfügung zu stellen. Doch bis zur flächendeckenden Verbreitung der moderneren Betriebssysteme wird noch einige Zeit vergehen. MS-DOS ist für den Verlauf der nächsten Jahre marktbestimmend und verspricht bei entsprechender Unterstützung durch den Hersteller noch interessante Erweiterungen.

(Markus Zietlow/wn)

MS-DOS NEUEINSTEIGER ???

Damit der Erfolg nicht zu lange auf sich warten läßt, das:

EINSTEIGERPAKET VON PD-SERVICE-LAGE

Für alle IBM/kompatiblen PCs: PC-TEXT 2.0/PC-TUTOR / MATHEASS / HALLO-PC / FINGERPAINT / MAU-MAU / TEACHER / BOBO / BALKENDIAGRAMME / HELP-DOS / ZINSEN PLUS KATALOGDISKETTE M. DEN STARTHINWEISEN

Der Paketpreis (13 Disketten)

5 1/4"

Scheck: 50,-/NN: 52,- DM

3 1/2"

Scheck: 70,-/NN: 72,- DM

Infos für 1.80 DM R.porto b.

PD-SERVICE-LAGE

Bernd Schulz
Postfach 1473
4937 Lage/Lippe

**Public-Domain
Shareware
und
Low-Cost-Software**

Software und Service aus einer Hand!

Werner Stähle
Softwarelösungen

Rahm 27 • 8500 Nürnberg 20
Telefon 0911/55 46 51

Unsere Software Spitzenprodukte aus eigener Herstellung:

Finanz-Manager PC Vers. 4.0	DM 79.50
OMEGA-ALPHA-HOROSKOP	DM 99.00
VIDEO-Datenbank PC	DM 49.90
CD-Verwaltung PC	DM 49.90

und jetzt brandneu.....

ATLANTIK PC

DOS-Benutzeroberfläche
mit Mausunterstützung
und DBF-Editor
DM 99.00



Star-Division-Produkte zum Knüllerpreis:

Star-Manager PC 2.0	DM 189.00
Star-Writer PC 3.02	DM 289.00
Star-Writer PC 5.0	DM 489.00
Laser Star	DM 489.00
David	DM 489.00
Star Planer	DM 389.00

Fordern Sie unser kostenloses Informationsmaterial mit unserem umfangreichen Hard- und Software-Gesamtangebot mit vielen Knüllerpreisen an! Jetzt auch Leasing von Hard- und Software möglich!

Händleranfragen willkommen!

Hotline: 0911/ 55 46 51 Sofort anrufen!



ABACOMP GmbH

Schriftliche Bestellungen:

Abt. DT12, Kronsberger Weg 24,
6000 Frankfurt/M. 50

Ladenverkauf: Mo-Fr 10-12 + 14-18h
Heerstr. 149, 6000 Frankfurt 90

Technische Auskunft:

Mo-Sa 8-9.30h
(069) 76 30 39

AT's

ab 1311,-!

ABACO-PC's "made in Germany"

Alle Geräte mit individuellen Ausstattungen und Erweiterungen lieferbar (siehe auch "c't" 11/89 S. 279)!

Zum Beispiel:

ABACO 16-286, AT-kompatibel, 8/12MHz ohne wait-states (Norton SI=13.7, Landmark 16MHz), 512KB RAM (ausbaubar bis 4MB on board, EMS-fähig), 1 Diskettenlaufwerk 1.2MB, hercules-kompatible Grafikkarte, Druckschnittstelle, Profi-Tastatur, Stromverbrauch Niederspannung ca. 40W, Netzteil >150W (>110W Reserveleistung), 200 Tage ABACOMP-Garantie

1311,-

ABACO 16-286, wie vor, jedoch 8/15MHz o. wait-states (Norton SI=16.7, Landmark 20MHz) 1653,-
Aufpreis für 20MB-Festplatte für ABACO 16-286/12 und ABACO 16-286/15 570,-

Auch XT's supergünstig:

ABACO 16HS, XT-kompatibel, 256KB RAM (bis 1MB on board mögl.), 1 Diskettenlaufwerk 360KB, 4.77/12MHz Taktfrequenz, V-20-Prozessor (Norton SI: 4.8 - Landmark: 5.0), Color-Grafik-Karte, Druckschnittstelle, Profi-Tastatur, Turbo und Reset, Stromverbrauch Niederspannung ca. 15W, Netzteil 135W (120W Reserveleistung), 200 Tage ABACOMP-Garantie 798,-
ABACO 16HS, wie vor, jedoch mit 512KB RAM bestückt 883,50 dilo, 1MB RAM 969,-

Zubehör für Ihren PC

NEU

Super Hi-Tech für den AT I

Der superschnelle Koprozessor für den 80286: IIT 2C87, voll kompatibel zum 80287, jedoch bei gleicher Taktfrequenz doppelt so schnell und mit komplettem 80387-Befehlssatz, dafür nur mit einem Bruchteil des Stromverbrauchs! Außerdem für 20MHz Taktfrequenz erhältlich, das Beste aber ist der Preis:
10MHz-Version 570,- 20MHz-Version 798,-
Turbo-20MHz für volle Taktfrequenz 85,50
Turbo-20MHz 85,50

Wir bieten auch die langsamere Alternative:

Original Intel 80287 (10MHz-Version) 570,-
Original Intel 80387 (20MHz-Version) 912,-

Hinweis: Einzelne Artikel aus unserem Sortiment sind aufgrund herstellender Fertigungsengpässe nicht sofort lieferbar. Viele weitere Produkte auf Anfrage. Händleranfragen erwünscht. Fordern Sie unsere aktuelle Sonderliste an. Wir suchen ständig Mitarbeiter für Verwaltung und Verkauf.

Grafikkomponenten

VGA-Grafikk., 256KB RAM, 800x600 Pkt 342,-
dilo, jedoch 16bit, 512KB, 1024x768 Pkt. 570,-
12" TTL-Monitor grün 114,-
14" VGA-s/w-Monitor nur 342,-
14" VGA-Farbmonitor, 640x480 Pkt., 0,31mm Pixelabstand nur 798,-

Gehäuse

Mini-AT-Gehäuse für 4 Laufwerke, davon 1x super-slim 3 1/2", komplett mit Netzteil und Montagematerial 228,-

Sonstiges

24-Nadel-Drucker ab 570,-
100er-Pack Disketten 3 1/2" 2DD 142,50

Dateiverwaltung unter MS-DOS

Das Kürzel »DOS« steht für den englischen Ausdruck »Disk Operating System«, deutsch etwa »Plattenspeicher-Betriebssystem«. Was sich hinter diesem Begriff verbirgt, ist dem Einsteiger meist unklar. Wie kann man Dateien verwalten und praktisch handhaben? Dieser Beitrag erklärt die wichtigen Funktionen zur Dateiverwaltung unter MS-DOS.

Wie der Begriff »Disk Operating System« bereits aussagt, orientiert sich das Betriebssystem MS-DOS sehr stark an sogenannten »Platten«. Damit sind alle externen Speichermedien wie Disketten oder Festplatten gemeint, die alle mehr oder minder »Plattenform« besitzen. Daher wird das Betriebssystem auch erst richtig komfortabel, wenn in dieser Hinsicht optimale Bedingungen herrschen, also eine Festplatte an den Computer angeschlossen ist. Diese erlaubt die Benutzung vieler Programme ohne lästigen Diskettenwechsel und ist zudem um ein Vielfaches schneller als der »kleine Bruder«, das Diskettenlaufwerk. Trotz der immensen Unterschiede bezüglich der Speicherkapazität von Festplatten und Disketten ist die Benutzung beider Speichermedien unter MS-DOS beinahe identisch. Auch technisch finden sich viele Gemeinsamkeiten.

Da die Art der Datenspeicherung unter MS-DOS in den Grundzügen einfach zu verstehen ist und dieses Wissen viel zum Verständnis des Betriebssystems beiträgt, sollte jeder PC-Anwender hierüber zumindest einmal etwas gehört haben. Jeder »Plattenspeicher« wird von MS-DOS auf die gleiche Art angesprochen. Die »Platte« wird beim Formatieren in sogenannte »Spuren«, »Sektoren« und »Cluster« eingeteilt. Die »Spuren« weisen die Form konzentrischer Kreise auf der Platte auf, deren gemeinsamer Mittelpunkt auch der physikalische Mittelpunkt der Platte ist, also die Drehachse. Je nach Speichermedium hat eine solche Platte 40 bis 80 (bei Disketten), bei Festplatten sogar bis zu mehr als 1000 Spuren. Jede Spur ist wiederum in »Sektoren« unterteilt. Dies können Sie sich so vorstellen, als ob die ganze Platte wie ein Kuchen aufgeschnitten würde. Disketten haben 8 bis 18 solcher »Kuchenstücke«, während Festplatten mehr als 30 haben können. Jede Spur einer Platte ist also in maximal diese Anzahl »Stückchen« aufgeteilt. Da hierdurch vor allem bei Festplatten eine sehr große Menge an »Sektoren« entsteht, faßt man diese wiederum zu

sogenannten »Clustern« zusammen, die dann je nach Speichermedium bis zu acht Sektoren umfassen. Jeder Cluster erhält – vom äußeren Rand der Platte an gezählt – eine Nummer, über die sich der Cluster und die darin enthaltenen Daten ansprechen lassen.

Die Datenspeicherung auf der Festplatte ähnelt der auf Disketten in hohem Maße

Die Speicherkapazität eines Sektors beträgt üblicherweise 512 Byte. Demnach enthält ein Cluster zwischen 1 KByte und 4 KByte Daten, je nachdem, wie viele Sektoren zusammengefaßt werden. Hier offenbart sich bereits ein Nachteil von MS-DOS: Da nicht die Sektoren, sondern Cluster vom Betriebssystem verwaltet werden, kann in einem Cluster immer nur eine Datei (beziehungsweise ein Teil einer Datei) stehen. Zwei Dateien können sich keinen Cluster teilen, auch wenn sie beide nur einen Umfang von wenigen Bytes aufweisen. Daher belegt jede Datei auf einem Plattenspeicher immer ein Vielfaches der Cluster-Größe und nie genau die tatsächliche Anzahl an Bytes, die die Datei umfaßt. Diese Informationen

genügen bereits, um sich die Arbeitsweise des Betriebssystems vorstellen zu können.

Auf Plattenspeichern werden äußerst verschiedenartige Daten gespeichert. Es kann sich bei einer Datei um ein Programm handeln, einen Treiber, eine Text- oder eine Stapeldatei. Um schon von Anfang an eine Unterscheidung dieser Daten zu gewährleisten, vergeben Sie unter MS-DOS in vielen Fällen bestimmte Dateinamen, die eine sichere Erkennung zulassen. Jeder Dateiname besteht aus bis zu acht Zeichen, gefolgt von einem Punkt und drei weiteren Zeichen. Hierbei sind die ersten acht Zeichen frei wählbar und werden vom Benutzer beziehungsweise vom Programmierer festgelegt, während den letzten drei Zeichen, der sogenannten Dateinamenserweiterung, eine Sonderfunktion zukommt. Ausführbare Programme besitzen typischerweise die Endung »exe« oder »com«, während Stapeldateien mit der Endung »bat« versehen sind. Mit einer Textverarbeitung geschriebene Texte erhalten zum Beispiel das Kürzel »txt« (für »Text«) oder »doc« (für englisch »document«), und Treiberprogramme haben meist ein »sys« als Erweiterung. Sie sehen, eine sinnvolle Namensgebung erleichtert den Umgang mit mehreren Dateien.

Da auf einem großen Plattenspeicher oft mehr als 1000 verschiedene Dateien erfasst werden können (die zudem manchmal noch ähnliche oder sogar gleiche Namen besitzen) gestattet MS-DOS eine weitere Unterteilung des Datenträgers in sogenannte »Unterverzeichnisse«.

Stellen Sie sich eine Festplatte als großen Schrank vor, in dem die Daten gesammelt werden. Bei Bedarf können Sie nun eine Schublade in den Schrank einsetzen, in der bestimmte Dateien getrennt von den anderen aufbewahrt werden können. Dieser Schublade können Sie – wie

Bild 1. So könnte das Inhaltsverzeichnis einer Festplatte aussehen, wie es sich dem Benutzer nach dem Befehl »dir« präsentiert

```
D:\DATABOX\DOS_03.88>dir
```

Datenträger in Laufwerk D ist DISK3			
Inhaltsverzeichnis von D:\DATABOX\DOS_03.88			
.	<DIR>	16.07.88	15:45
..	<DIR>	16.07.88	15:45
ASSEMBL	<DIR>	16.07.88	15:46
C	<DIR>	16.07.88	15:46
DBASE	<DIR>	16.07.88	15:46
GW BASIC	<DIR>	16.07.88	15:46
MAUS	<DIR>	16.07.88	15:46
MP	<DIR>	16.07.88	15:46
PR	<DIR>	16.07.88	15:46
PRAXIS	<DIR>	16.07.88	15:46
SPESEN	<DIR>	16.07.88	15:46
TRICK	<DIR>	16.07.88	15:46
TSR	<DIR>	16.07.88	15:46
TT	<DIR>	16.07.88	15:46
START	BAT	12	26.01.88 1:06
INHALT	DOC	3656	25.01.88 19:31
READ	ME	1203	25.01.88 0:25
17 Datei(en) 23199744 Bytes frei			
D:\DATABOX\DOS_03.88>			

jeder Datei – einen Namen geben. Es lassen sich nach Belieben weitere Schubladen einsetzen, um den »Schränkinhalt« zu ordnen, und es können sogar Schubladen in Schubladen eingebaut werden und so fort. Sinnvollerweise benutzen Sie eine Schublade für das Speichern eines Programms und für Daten, die zu diesem Programm gehören.

Schalten Sie Ihren Computer ein, und starten Sie MS-DOS. Wenn Sie eine Festplatte besitzen, so geben Sie den Befehl »c:«, gefolgt von [Enter], ein, um die Festplatte zum aktuellen Laufwerk zu machen, das heißt, um alle folgenden Zugriffe auf die Festplatte auszuführen. Mit der Anweisung »dir« erhalten Sie ein Inhaltsverzeichnis (englisch »directory«) Ihres Speichermediums (Bild 1). Alle enthaltenen Dateien werden so aufgelistet, wobei außer dem Dateinamen und der Erweiterung auch noch Datum und Uhrzeit der letzten Änderung der Datei gespeichert sind. Geben Sie nun »md dos« ein, und lassen Sie sich neuerlich ein Inhaltsverzeichnis auf dem Bildschirm anzeigen. Jetzt ist ein weiterer Eintrag »DOS <DIR>« sichtbar. »<DIR>« sagt aus, daß es sich hierbei um ein Unterverzeichnis handelt. Sie haben Ihre erste »Schublade« eingebaut. Der Befehl »md« steht für »make directory« (»erzeuge Unterverzeichnis«), wobei Sie nach »md« einen beliebigen Namen angeben können, der, wie bei Dateien, aus maximal acht Zeichen, gefolgt von einem Punkt und der dreistelligen Erweiterung, bestehen darf.

Auch auf der Festplatte herrscht Ordnung – dank der Unterverzeichnisse

Um das neue Unterverzeichnis wieder zu löschen, geben Sie »rd dos« ein (»rd« bedeutet »remove directory«). Wie Sie sehen, werden MS-DOS-Befehle einfach mit ihren Namen aufgerufen. Dies gilt sowohl für »interne« Befehle, die MS-DOS sofort ausführen kann, wie auch für »externe« Anweisungen, die in Form eines Programms mit der Endung »com« oder »exe« auf den Systemdisketten stehen.

Selbstverständlich läßt sich jedes Unterverzeichnis dazu benutzen, um Dateien dort zu speichern. Hierzu können Sie entweder das Unterverzeichnis über seinen Namen ansprechen oder direkt in das Unterverzeichnis überwechseln, um alle folgenden Dateioperationen innerhalb dieses Unterverzeichnisses vorzunehmen. Möchten Sie in das Unterverzeichnis wechseln, so geben Sie den

Befehl »cd dos« ein, um mit diesem »cd« für »change directory« (»wechseln Unterverzeichnis«) die Verzeichnisebene zu wechseln. Die Eingabeaufforderung (Prompt) sieht nun folgendermaßen aus: »C: \DOS>«. (Ist dies nicht der Fall, so programmieren Sie den Prompt mit dem MS-DOS-Befehl »prompt \$p\$g« entsprechend.)

Anhand der Eingabeaufforderung können Sie also feststellen, in welchem Unterverzeichnis Sie gerade arbeiten. Das Hauptverzeichnis, oft auch »root directory« oder »Stammverzeichnis« genannt, besitzt die Bezeichnung »\« (Backslash), die Sie nicht ändern können.

Lassen Sie sich ein Inhaltsverzeichnis von »C: \DOS« ausgeben, indem Sie einfach wieder »dir« eingeben. Durch den Wechsel in das Unterverzeichnis werden alle Befehle innerhalb dieses Verzeichnisses ausgeführt. Sie erhalten eine Auflistung mit den zwei Benennungen ».« und »...«. Der einzelne Punkt symbolisiert das aktuelle Unterverzeichnis, wohingegen die beiden Punkte für das eine Ebene übergeordnete Verzeichnis stehen. Sie können daher mit »cd ..« wieder in das Stammverzeichnis zurückkehren. Die Anweisung »cd \« bewirkt das gleiche. Innerhalb von »C: \DOS« stellen Sie mit »md <name>« beliebige weitere Unterverzeichnisse her, für die die gleichen Regeln wie für »C: \DOS« gelten.

Im Unterverzeichnis »C: \DOS« sollten Sie Ihre MS-DOS-Systemprogramme ablegen, die Sie auf den Betriebssystemdisketten finden. Um Dateien zu kopieren, verwenden Sie den Befehl »copy«. Wollen Sie von der Betriebssystemdiskette im Laufwerk A alle Programme in das MS-DOS-Unterverzeichnis auf Ihrer Festplatte kopieren, so geben Sie den Befehl

```
copy a:.*.* c: \dos
```

ein. Bei »copy« muß grundsätzlich angegeben werden, welche Datei kopiert und wo die Kopie abgelegt werden soll. »a:« bedeutet, daß vom Laufwerk A kopiert wird. Um alle Dateien gleichzeitig zu kopieren, verwenden Sie den Befehl »*.*«, eine Art »Joker«. Ein Sternchen »*« ist ein Stellvertreter für jeden erdenklichen Dateinamen. Somit hat »*.*« zur Folge, daß alle Dateien unabhängig von ihrem Namen und ihrer Namensweiterung, angesprochen werden. Natürlich können Sie bei »copy« auch einen einzelnen Dateinamen angeben.

Das »Ziel« wird durch »c: \dos« gekennzeichnet, gemeint ist das MS-DOS-Unterverzeichnis auf dem Laufwerk C. Sofern innerhalb dieses Unterverzeichnisses weitere Unterver-

zeichnisse bestehen, lassen sich Dateien durch

```
copy <dateiname> c: \dos \ <unterverzeichnis>
```

auch dorthin kopieren. Dies gilt für beliebig tief verschachtelte Verzeichnisse, Sie müssen die Unterverzeichnisnamen nur durch ein »\« trennen.

»Wildcards« erlauben den gleichzeitigen Zugriff auf viele verschiedene Dateien

Die »Joker«, auch »Wildcards« genannt, sind eine sehr praktische Einrichtung unter MS-DOS. Sie erlauben auch eine differenzierte Auswahl der Dateien. Beispielsweise werden durch den Joker »*.txt« alle Dateien mit der Endung »txt« angesprochen. Mit »pc*.*« erreichen Sie alle Dateien, die mit der Zeichenfolge »pc« im Dateinamen beginnen, und mit »pc*.txt« erhalten Sie eine Kombination dieser beiden Möglichkeiten. Natürlich lassen sich unter MS-DOS nicht nur Dateien in Unterverzeichnisse kopieren. Eine wichtige und sehr einfache Funktion ist der Löschbefehl »del«. Sie wird mit der Syntax

```
del <dateiname>
```

aufgerufen, wobei »<dateiname>« hier ebenfalls Joker enthalten darf oder auch ein eindeutiger Dateiname sein kann. Mit »del *.*« löschen Sie alle Daten in dem Unterverzeichnis, in dem Sie derzeit arbeiten (im »aktuellen« Unterverzeichnis), wobei MS-DOS aber sicherheitshalber nochmals nachfragt, ob dies auch wirklich geschehen soll.

Textdateien im sogenannten ASCII-Format können Sie direkt vom Betriebssystem auf den Bildschirm ausgeben lassen. ASCII-Dateien dürfen nur Buchstaben, Zahlen und Satzzeichen enthalten. Durch

```
type <dateiname>
```

wird der Inhalt der Datei »<dateiname>« auf dem Bildschirm ausgegeben. Da der Text auf dem Bildschirm »durchläuft«, drücken Sie die Taste [Pause], um den Text anzuhalten. Ein Druck auf eine beliebige andere Taste läßt den Text wieder weiterlaufen. Durch die Anweisung »type <dateiname> | more« wird der Bilddurchlauf nach einer Bildschirmseite gestoppt, was eine komfortablere Variante des Lesens einer Datei darstellt. Das Zeichen »|« erreichen Sie durch die Tastenkombination [Alt-124].

Eine ebenfalls häufig benötigte Funktion im Umgang mit Dateien ist »rename«, womit Dateinamen geändert werden können. Die Syntax dieses Befehles lautet


```

D:\DATABOX\DOS_03.08\TT>dir

Dateiträger in Laufwerk D ist DISK3
Inhaltsverzeichnis von D:\DATABOX\DOS_03.08\TT

      <DIR>          16.07.88  15:46
      <DIR>          16.07.88  15:46
CTRLBR  BAS        398  30.12.87  2:09
CTRLBRK SUB       485  30.11.87  20:49
TASTPLOE BAS        86  30.12.87  2:11
AUTOMSC  PAS       431  30.12.87  18:44
GSEITEM  INC      1715  1.01.88  0:01
COMZX    BAS       131  30.12.87  2:28
INTERR   C        184  30.12.87  2:29
DEMO     PAS       368  1.01.88  0:02
DEMO     COM     17272  1.01.88  0:53
BILD1    BLD     16256  1.01.88  0:42
BILD2    BLD     16256  1.01.88  0:42
BILD3    BLD     16256  1.01.88  0:43
14 Datei(en) 23199744 Bytes frei

D:\DATABOX\DOS_03.08\TT>

```

Bild 2. So werden Unterverzeichnisse sinnvoll angelegt

Eine ebenfalls häufig benötigte Funktion im Umgang mit Dateien ist »rename«, womit Dateinamen geändert werden können. Die Syntax dieses Befehles lautet

```
ren <alter dateiname> <neuer dateiname>
```

Dabei wird der ehemalige Dateiname einfach überschrieben. Falls der neue Dateiname bereits vergeben ist, erhalten Sie eine Fehlermeldung, und Sie müssen einen anderen Namen verwenden.

MS-DOS findet Ihre Daten auf Wunsch automatisch – durch »path«

Um unter MS-DOS die tägliche Arbeit erst richtig komfortabel zu gestalten, existieren neben den bisher genannten, eher essentiellen Befehlen auch noch einige weitere »Luxusanweisungen«, ohne die die Arbeit eigentlich kaum mehr denkbar ist. Stellen Sie sich vor, Sie haben Ihre Festplatte ordentlich in Unterverzeichnisse aufgeteilt und jedes Programm in ein eigenes Verzeichnis kopiert (Bild 2). Womöglich besitzen Sie sogar ein Verzeichnis »\tools«, in dem sich Ihre »gesammelten Unterverzeichnisse« für Hilfsprogramme befinden, also beispielsweise »\tools\pctools« und »\tools\norton«, falls Sie die Programmpakete »PC-Tools« und »Norton Utilities« besitzen. Unter Umständen sind Ihre Verzeichnisse sogar noch tiefer verschachtelt. Um nun ein Programm in einem Unterverzeichnis zu starten, müssen Sie den kompletten Unterverzeichnisnamen mit angeben, also zum Beispiel

```
c:\tools\pctools\pctools
```

um das Programm »pctools« in seinem Unterverzeichnis zu starten. Das bedeutet viel Tipparbeit, die zudem

fehlerträchtig ist. Unter MS-DOS lassen sich daher alle Unterverzeichnisse mit häufig benötigten Programmen auf spezielle Weise ansprechen, wobei der lange »Pfadname« zum Unterverzeichnis entfällt. Diese Vereinfachung bewirkt die Anweisung »path«, zu deutsch »Pfad«. Eine Beispielanwendung für »path« wäre

```
set path=c:\;c:\tools\pctools;c:\tools\norton;
```

»path« ist eine sogenannte Systemvariable, das heißt, daß das Betriebssystem diese Variable für Sonderzwecke verwendet. In diesem Fall legt der Inhalt von »path« fest, in welchen Unterverzeichnissen MS-DOS nach Programmen »suchen« soll, falls sich ein Programm, das Sie starten möchten, nicht im aktuellen Verzeichnis befindet. Mit »set« können alle Systemvariablen – wie auch beliebige andere Variablen – mit Werten belegt werden. Befindet sich ein Programm bei Aufruf nicht im aktuellen Unterverzeichnis, so werden alle Unterverzeichnisse in der Variablen »path« in der dort angegebenen Reihenfolge durchsucht. Beachten Sie bitte, daß Sie den Befehl »path« nur einmal verwenden sollten, wobei sie gleich alle gewünschten Unterverzeichnisse in einer Zeile angeben. Ein zweites Setzen von »path« löscht nämlich alle zuvor angegebenen Unterverzeichnisse aus dem »Suchpfad« und ersetzt sie durch die neu angegebenen.

Bisher wurden nur MS-DOS-Befehle angesprochen, die innerhalb des Systems Funktionen ausführen. An einen Computer sind aber häufig diverse Peripheriegeräte angeschlossen. Um beispielsweise Dateien auf dem Drucker auszugeben, gibt es verschiedene Alternativen. Die einfachste bietet der bereits bekannte Befehl »copy«. Um eine Datei zum Drucker zu »kopieren«, genügt

```
copy <dateiname> prn
```

Die Zeichenfolge »prn« ist wieder eine Systemvariable, und zwar stellvertretend für den Drucker. Ebenso denkbar wären die Stellvertreter »lpt1« bis »lpt3«, wobei hier die parallelen Schnittstellen Nummer 1 bis 3 direkt angesprochen werden.

Systemvariablen haben Sonderfunktionen, die die Arbeit vereinfachen

Normalerweise hat »prn« den Wert »lpt1«, beide Bezeichner sind also gleichwertig. Mit einem einfachen

```
set prn=lpt2
```

läßt sich dies aber beliebig verändern. Sollte Ihr Drucker an einer seriellen Schnittstelle betrieben werden, so lassen sich für »prn« auch die Werte »com1« bis »com4« verwenden, die stellvertretend für die vier erlaubten seriellen Schnittstellen stehen. Etwas anspruchsvoller ist die Anweisung

```
print <dateiname>
```

Sie bewirkt das Laden des MS-DOS-Programms »print.com«, das in Ihrem MS-DOS-Unterverzeichnis »c:\dos« abgelegt sein sollte. Dieses Programm ist ein sogenannter »Drucker-Spoober« und verwaltet den angeschlossenen Drucker komfortabel. Das Programm organisiert eine »Warteschlange«, wodurch sich mehrere Dateien hintereinander, ohne weiteres Zutun des Anwenders, ausdrucken lassen. Beim ersten Aufruf von »print« fragt das Programm nach der Schnittstelle, auf der die Druckerdaten ausgegeben werden sollen. Bis zum nächsten Systemstart bleibt »print« im Speicher Ihres Computers resident im Hintergrund. Falls Sie während eines laufenden Ausdrucks eine weitere Datei in die Druckerwarteschlange aufnehmen wollen, so genügt hierzu die Anweisung »print <dateiname>«. Sobald der Drucker den Ausdruck der ersten Datei beendet hat, beginnt er sofort mit der zweiten Datei. Sie können mehrere Dateien auf »Warteposition« schicken und während des Ausdrucks an Ihrem Computer weiterarbeiten. Eine spezielle Eigenschaft von MS-DOS stellen die sogenannten »Stapeldateien« dar, die das vereinfachte Ausführen immer wiederkehrender Folgen von MS-DOS-Kommandos gestatten. Eine »Stapeldatei« (auch »Batch-Datei« genannt) besteht im Grunde ausschließlich aus MS-DOS-Befehlen, die zusammen wie eine Art kleines »Programm« ausgeführt werden können. Mit jedem ASCII-Editor können Sie Stapeldateien schreiben. Eine sinnvolle Anwendung von Stapeldateien

ist wiederum der vereinfachte Programmaufruf. Meist können Sie nicht alle Programme über den Befehl »path« direkt zugänglich machen. Dies ist auch nicht nötig, wenn Sie in den Suchpfad ein Unterverzeichnis aufnehmen, in dem Stapeldateien abgelegt sind, die ihrerseits Programme aufrufen. Richten Sie ein Unterverzeichnis »C:\BATCH« ein, und schreiben Sie einige Stapeldateien, die Sie dort ablegen.

Wiederkehrende Befehlsfolgen werden in »Stapeldateien« zusammengefaßt

Eine Stapeldatei besitzt immer die Endung ».bat«. Dies und das reine ASCII-Format sind eigentlich die einzigen Richtlinien, um eine funktionierende Stapeldatei zu schreiben. Das Beispiel einer Stapeldatei, die das Programm »word« im Unterverzeichnis »c:\text\word« aufruft, soll die Vorgehensweise verdeutlichen. Hier ist eine der zahlreichen Varianten:

```
cls
c:
cd \
cd \text\word
word
```

Der Befehlsablauf ist hier recht einfach: Zunächst wird mit »cls« (für »clear screen«) der Bildschirm gelöscht. Dann schaltet MS-DOS auf Laufwerk C um (»c:«) und wechselt dort in das Stammverzeichnis (»cd \«). Von dort aus wird das Unterverzeichnis »\text\word« angewählt (»cd \text\word«) und mit »word« schließlich das Programm gestartet. Stapeldateien können sehr komplex sein, da alle MS-DOS-Befehle in einer Stapeldatei verwendet werden können. Um die Arbeiten mit Ihrem Computer zu erleichtern, sollten Sie für jedes Programm eine entsprechende Startstapeldatei schreiben. Auf diese Weise können Sie auch Standardeinstellungen vornehmen: Muß vor einer Anwendung beispielsweise der Maustreiber »mouse.com« geladen werden, so kann dies fest in der Startstapeldatei des Anwendungsprogramms verankert werden. Die wohl wichtigste Anwendung von Stapeldateien ist gewiß die Datei »autoexec.bat«, die jedesmal beim Start von MS-DOS aufgerufen und ausgeführt wird. Zusätzlich zu »normalen« Stapeldateien sind in »autoexec.bat« einige Befehle sinnvoll, die die Voreinstellung von Systemeigenschaften vornehmen. Das Beispiel einer »durchschnittlichen« Autoexec-Datei verdeutlicht dies:

```
echo off
cls
```

```
keyb gr
country=049
set comspec=c:\dos\command.com
set path=c:\;c:\batch;c:\tools\pctools;c:\tools\norton;
prompt $p$g
ver
echo on
```

Mit der Anweisung »echo off« werden alle folgenden Bildschirmmeldungen von MS-DOS-Anweisungen unterdrückt, und der Bildschirm wird via »cls« gelöscht. Der Tastaturreiber »keyb.com« wird mit der Option »gr« aufgerufen, wodurch die deutsche DIN-Tastaturbelegung installiert wird. Durch »country = 049« teilen Sie Ihrem Computer mit, daß die deutschen Formate für Datum und Uhrzeit verwendet werden sollen (049 ist die internationale Telefonvorwahl für die Bundesrepublik Deutschland). Die Systemvariable »comspec« wird durch den Wert »c:\dos\command.com« belegt, der angibt, in welchem Verzeichnis der MS-DOS-Kommandoprozessor »command.com« zu finden ist. Anschließend wird der Prompt festgelegt, die MS-DOS-Versionsnummer ausgegeben (»ver«). Zuletzt werden alle Bildschirmmeldungen von MS-DOS-Befehlen wieder freigegeben (»echo on«).

Die Datei »config.sys« besitzt ebenfalls Sonderfunktionen zum Systemstart

So einfach wie in diesem Beispiel sind in der Regel viele Autoexec-Dateien. Die aufgeführten Anweisungen reichen normalerweise aus, um das System korrekt zu betreiben. Natürlich können auch alle anderen MS-DOS-Befehle zusätzlich eingebaut werden, um beliebige Betriebssystemfunktionen beim Systemstart automatisch auszuführen. Eine weitere Datei mit dem Namen »config.sys« hat ebenfalls große Bedeutung, da auch sie beim Systemstart – sogar noch vor »autoexec.bat« – geladen wird. In dieser Datei kann nur eine begrenzte Anzahl von speziellen Befehlen verwendet werden, die unter MS-DOS nicht direkt eingegeben werden können, wie zum Beispiel die Befehle von »autoexec.bat«. Auch hier dient ein praktisches Beispiel zum besseren Verständnis:

```
buffers=20
files=30
shell=c:\dos\command.com /p /e:1024
device=c:\dos\ansi.sys
device=c:\mouse\mouse.sys
```

Alle hier verwendeten Anweisungen können nur in »config.sys« stehen und zur Ausführung gebracht werden. Um die getroffenen Einstellungen zu ändern, müssen Sie »con-

fig.sys« entsprechend ändern und dann MS-DOS neu starten.

Die Systemvariable »buffers« legt fest, wie viele »Eingabepuffer« für Dateien von externen Speichermedien reserviert werden sollen. Dies ist gerade bei der Verwaltung großer Datenbestände – beispielsweise mit einem Datenbankprogramm – sehr wichtig, da bei entsprechend großer Zahl von »Puffern« der Zugriff auf einzelne Dateien beschleunigt wird. Leider belegt jeder dieser Puffer im Computer Speicherplatz, der dann nicht mehr für ausführbare Programme zur Verfügung steht.

Die Variable »files« (Dateien) erfüllt einen ähnlichen Zweck wie »buffers«, nur wird hier die Zahl der maximal gleichzeitig geöffneten Dateien festgelegt. Bei der Arbeit mit Datenbankprogrammen kann es wichtig sein, daß der Computer gleichzeitig auf mehrere Dateien – zum Beispiel eine Kundendatei und eine Artikelliste – zugreifen kann. Ähnliches gilt für Compiler oder eventuell sogar Spiele. Eine ausreichende Anzahl von Dateien (»files«) sollte reserviert werden, da MS-DOS standardmäßig nur acht Dateien reserviert, von denen fünf immer belegt sind.

Die Shell-Anweisung markiert nochmals das Unterverzeichnis, in dem der »command.com« (Kommandoprozessor) abgelegt ist. Die Option »/p« bewirkt die Ausführung von »autoexec.bat«, die sonst unterdrückt würde. Der Speicherbereich für Systemvariablen wie »path« oder »comspec« wird durch »/e:1024« auf 1024 Byte festgesetzt. Zuletzt werden noch einige Treiberprogramme aktiviert, indem diese als »devices« (Geräte) im System eingebunden werden. Der Bildschirmtreiber »ansi.sys«, der eine genormte Ansteuerung des Bildschirms gewährleistet, wird ebenso wie der Maustreiber »mouse.sys« nach dem Laden sofort »verbannt« und arbeitet von nun an im Hintergrund. Dies bedeutet, daß die Funktionen dieser Programme ständig zur Verfügung stehen, ohne daß Sie etwas von den Treiberprogrammen bemerken. Häufig werden zu Programmen und Hardware-Komponenten, die Sie für Ihren Computer erwerben können, spezielle Treiber geliefert, die zum Beispiel die Fähigkeiten einer Grafikkarte erweitern. Auch diese Treiber können Sie auf die oben gezeigte Weise aktivieren, indem Sie eine entsprechende Zeile in die Datei »config.sys« einfügen und den Computer neu starten. Wie Sie im Laufe dieses Beitrags bemerkt haben, bietet MS-DOS viele auch sehr komfortable Funktionen, um die anfallenden Datenbestände Ihres Computers fest in den Griff zu bekommen.

(Thomas Rueß/as)

Referenz: Alle MS-DOS-Befehle bis Version 4.0

Diese Übersicht enthält alle Befehle der Betriebssysteme MS-DOS und PC-DOS einschließlich der Version 4.0. Dabei sind sowohl die internen Befehle des Kommando-Interpreters »command.com« als auch die auf den Systemdisketten mitgelieferten Dienstprogramme berücksichtigt. Ein Beispiel zu jeder Beschreibung erleichtert es Ihnen, die genaue Funktionsweise der Befehle zu verstehen.

Die Befehle von MS-DOS gliedern sich in zwei Teile: Die internen (oder residenten) und die externen (oder transienten) Kommandos. Die internen Befehle »versteht« der Computer sofort, sie werden unmittelbar nach der Eingabe ausgeführt. Bei den externen Befehlen hingegen handelt es sich strenggenommen nicht um Kommandos des Betriebssystems, sondern um kleine Hilfsprogramme. Alle externen Befehle sind auf dem Datenträger (also der Systemdiskette oder der Festplatte) als ausführbare Programme mit den Kennungen »com« oder »exe« gespeichert.

Damit Sie die externen Befehle benutzen können, müssen Sie daher sicherstellen, daß MS-DOS eine Möglichkeit hat, auf die externen Programme zuzugreifen. Die Systembefehle müssen also im aktiven Verzeichnis oder im Suchpfad zugänglich sein.

APPEND <Pfad>; [<Pfad 2>; [... <Pfad N>;] [<Optionen>]

Mit dem externen Append-Kommando erweitern Sie die Suche nach einer von einem Programm angeforderten Datei über das aktive Verzeichnis hinaus auf andere Verzeichnisse. Dies funktioniert mit allen Programmen, welche die Standardfunktionen von MS-DOS zum Öffnen von Dateien verwenden. Zur Suche von ausführbaren Dateien mit den Endungen »com«, »exe« und »bat« ist hingegen der Path-Befehl zuständig.

Mit dem Parameter »Pfad« geben Sie den kompletten Suchpfad eines Verzeichnisses an, in dem zusätzlich gesucht werden soll.

Sie können dem Append-Befehl mehr als einen Pfad übergeben. Die einzelnen Verzeichnisangaben müssen dann durch einen Strichpunkt voneinander getrennt werden.

Mit der Option »/e« erreichen Sie, daß der definierte Suchpfad in der Umgebungsvariablen »append« gespeichert wird. Anwendungsprogramme können diese Variable dann gezielt abfragen.

Die Option »/x« dient dazu, den erweiterten Suchpfad zusätzlich zur Umgebungsvariablen »path« auch für

ausführbare Dateien zu verwenden. Ab MS-DOS 4.0 schalten Sie diese Erweiterung mit der Option »/x:off« bei einem weiteren Aufruf von »append« jederzeit wieder aus (erneutes Einschalten dann durch »/x:on« oder erneutes »/x«).

Rufen Sie das Append-Kommando nur mit einem Strichpunkt als Pfadname auf, so werden alle zusätzlichen Suchpfade gelöscht.

Beispiel:

```
append c:\ ws;
```

ASSIGN <Laufwerk> = <Neuer Name>

Mit diesem externen Befehl weisen Sie einem logischen Laufwerk einen neuen Namen zu. Dazu geben Sie mit dem Parameter »Laufwerk« die Bezeichnung des Laufwerks an, das Sie ansprechen möchten. Mit dem Parameter »Neuer Name« wird die Bezeichnung gesetzt, unter der Sie das Laufwerk fortan ansprechen möchten.

Rufen Sie »assign« ohne Parameter auf, so werden die alten Bezeichnungen für alle Laufwerke wieder hergestellt.

Beispiel:

```
assign a=c
rem Festplatte nun als
rem Diskettenlaufwerk ansprechbar.
```

ATTRIB [+R | -R] [+A | -A] <Datei> [/S]

Mit dem externen Attrib-Befehl ändern Sie die Attribute einer Datei. Erst ab MS-DOS 3.0 heißt der Befehl »attrib«. Die Versionen vor 3.0 verwenden einen Befehl namens »chmod« für den gleichen Zweck.

Mit dem Parameter »Datei« geben Sie an, welche Datei vom Attrib-Kommando bearbeitet werden soll. Die Joker-Zeichen »*« und »?« sind dabei zulässig. Rufen Sie »attrib« nur mit dem Namen einer Datei ohne Parameter auf, so werden deren Attribute auf dem Bildschirm angezeigt.

Möchten Sie die Attribute der Datei ändern, so gelten hierfür die folgenden Kürzel:

+ R: Die Datei wird schreibgeschützt, sie läßt sich also nur noch lesen.
- R: Der Schreibschutz der Datei wird aufgehoben.

+ A: Das Archiv-Bit der Datei wird gesetzt.

- A: Das Archiv-Bit der Datei wird gelöscht.

Die Bedeutung des Archivbits ist im Zusammenhang mit anderen Befehlen (zum Beispiel »backup« oder »xcopy«) erläutert.

Die Option »/s« im Attrib-Befehl ist dann von Bedeutung, wenn Sie Joker-Zeichen im Dateinamen verwenden. Dann bewirkt die Option »/s«, daß die Veränderung oder Ausgabe des Attributs auch auf alle Unterverzeichnisse ausgedehnt wird, in denen Dateinamen mit der Maske übereinstimmen.

Beispiel:

```
attrib +r kapitel1.txt
```

BACKUP <Quelle> <Ziellaufwerk> <Optionen>

Der externe Backup-Befehl dient dazu, vom Inhalt einer Festplatte eine Kopie auf Diskette herzustellen. Sie können entweder eine komplette Festplatte oder auch nur ein einzelnes Verzeichnis oder eine Gruppe von Verzeichnissen sichern. Der Parameter »Quelle« gibt an, welches Laufwerk oder Verzeichnis gesichert werden soll. Das Ziellaufwerk muß die Bezeichnung eines Diskettenlaufwerks oder einer Festplatte sein. Der Laufwerkbuchstabe muß gefolgt von einem Doppelpunkt angegeben werden.

Das Backup-Kommando sichert die angegebenen Dateien fortlaufend auf Disketten. Dateien, die nicht auf eine einzige Diskette passen, werden automatisch aufgeteilt. Dazu kopiert der Befehl die Quelldateien in fortlaufend nummerierte Dateien des Namens »backup.nnn«. Das Kürzel »nnn« steht für eine fortlaufende Nummer.

Die Informationen über die Gesamtzahl der Dateien eines Backups wird in die Datei »control.nnn« geschrieben. Um die Dateien wieder auf die Festplatte zurückzuschreiben, muß das Restore-Kommando verwendet werden.

Beim Backup-Befehl sind eine Reihe von Optionen verwendbar. Die Option »/s« (»subdirectories«) weist MS-DOS an, alle Dateien des aktuellen Verzeichnisses und der darunterliegenden zu sichern.

Mit der Option »/m« (»modified«) werden nur Dateien gesichert, deren Archivbit gesetzt ist. Dieses Bit ändern Sie durch den Attrib-Befehl. Es wird außerdem von MS-DOS immer dann gesetzt, wenn eine Datei geändert wurde.

Die Option »/a« (»append«) bewirkt, daß die Dateien an einen bestehenden Satz von Backup-Disketten angehängt werden. Bereits bestehende Dateien werden dabei nicht gelöscht.

Bei einem normalen Backup geht der Inhalt der Disketten, auf die gesichert wird, verloren. Der Parameter »/f« (»format«) weist MS-DOS an, die Disketten zu formatieren, bevor sie mit den Daten der Festplatte beschrieben werden. Zu diesem Zweck muß das Dienstprogramm »format« im aktuellen Verzeichnis oder im Suchpfad zu finden sein. Bei neuen Disketten geschieht das Formatieren automatisch. Wenn Sie für den Backup-Prozeß eine genaue Kontrolle benötigen, dann lassen Sie sich mit dem Parameter »/l« eine Logbuchdatei anlegen. Unter dem Namen »backup.log« wird dann in einer Textdatei festgehalten, welche Programme kopiert wurden. Wenn Sie für die Logbuchdatei einen anderen Namen verwenden möchten, geben Sie hinter der Option »/l« einen alternativen Dateinamen an. Sehr nützlich sind die Optionen »/d« und »/t«, mit denen Sie die Auswahl der zu sichernden Dateien auf solche beschränken, die nach einem bestimmten Datum oder einer bestimmten Zeit erstellt oder verändert worden sind. Hinter der Option »/d« (»date«) geben Sie dazu ein Datum im Format »tt.mm.jj« (t = Tag, m = Monat, j = Jahr) an.

Möchten Sie die Backup-Funktion außerdem auf Dateien beschränken, die nach einer bestimmten Zeit erstellt worden sind, verwenden Sie die Option »/t«. Hinter ihr wird im Format »hh:mm:ss« eine Uhrzeit angegeben. Die Datei wird nur kopiert, wenn sie nach der angegebenen Zeit erstellt oder verändert wurde. Die Uhrzeiteoption ist nur in Verbindung mit einer Datumsangabe sinnvoll. Beispiel:

```
backup c: a: /f /d:13.7.89
```

BREAK [ON | OFF]

Mit dem Break-Befehl legen Sie fest, wie der Rechner auf die Tastenkombination [Ctrl-C] beziehungsweise [Ctrl-Break] reagiert. Wenn Sie diese Tasten betätigen, lösen Sie dadurch eine bestimmte Programmroutine im Betriebssystem aus. Die Routine teilt dem gerade laufenden Programm dann mit, daß der Anwender eine Unterbrechung wünscht. Ob das Programm daraufhin wirklich beendet wird, ist jedoch nicht vom Break-Kommando abhängig. Vielmehr hat der jeweilige Programmierer bereits festgelegt, ob sein Programm unterbrochen werden darf.

Der Break-Befehl legt lediglich fest, ob das Betriebssystem auch bei Festplattenzugriffen und bei Bildschirmausgaben die Abbruchtastensequenz erkennen soll (»break on«) oder nur bei Tastaturabfragen des gerade laufenden Programms (»break off«). Bei eingeschaltetem Break können Sie beispielsweise das Formatieren einer

Diskette unterbrechen, bei »break off« hingegen nicht.

Wenn Sie »break« ohne Parameter eingeben, zeigt Ihnen das Betriebssystem den aktuellen Status der Funktion an, ohne diesen zu verändern. Beispiel

```
break=on
```

CALL <Stapeldatei>

[<Optionen>]

Das interne Call-Kommando ist ab MS-DOS 3.3 verfügbar und startet eine zweite Stapeldatei als Unterprogramm einer anderen Datei. Anschließend kehrt MS-DOS wieder in das erste Stapelprogramm zurück und setzt dort die Ausführung in der Zeile nach dem Call-Befehl fort. Die Optionen werden der nachgeladenen Stapeldatei als normale Parameter in den Variablen »%1«, »%2« und so weiter zur Verfügung gestellt. Die aufgerufene Datei wird zuerst im aktuellen Verzeichnis, dann im MS-DOS-Pfad gesucht. Die Namenserverweiterung »bat« ist optional, darf also auch fehlen.

Mit dem Call-Befehl können nur Stapeldateien, jedoch keine ausführbaren Programme gestartet werden.

Beispiel

```
call subdatei
```

CHCP <Zeichensatztable>

Ab MS-DOS 3.3 wird die Verwaltung von mehreren Zeichensatztabellen unterstützt, und zwar unabhängig vom gerade aktiven Tastaturtreiber. Mit dem internen Chcp-Befehl ist es möglich, zwischen diesen Zeichensatztabellen zu wechseln, ohne das System neu starten zu müssen. Der Parameter »Zeichensatztable« gibt die Nummer der Tabelle an, die verwendet werden soll:

```
437 IBM-Standardzeichensatz
850 mehrsprachig
860 Portugal
863 Kanada (französisch)
865 Norwegen
```

Damit der Chcp-Befehl auch tatsächlich zwischen den Zeichensatztabellen umschalten kann, müssen diese zuvor mit dem Befehl »mode prep« vorbereitet werden.

Das Chcp-Kommando muß auf die Datei »country.sys« zugreifen, in der sich landesspezifische Informationen befinden. Beim Aufruf von Chcp muß sich »country.sys« also im aktiven Verzeichnis befinden.

Beispiel:

```
chcp 437
```

CHDIR <Verzeichnis>

CD <Verzeichnis>

Mit dem internen Chdir-Befehl (er wird meist in der verkürzten Form »cd« verwendet) wechseln Sie das aktive Verzeichnis.

Rufen Sie den Cd-Befehl ohne Parameter oder nur mit einer Laufwerks-

angabe auf, wird angezeigt, welches Verzeichnis auf dem betreffenden Laufwerk zur Zeit aktiv ist.

Mit »cd ..« wechseln Sie in das nächsthöhere Verzeichnis.

Chdir wechselt nicht auf ein anderes Laufwerk. Wenn Sie sich auf Laufwerk A befinden und zum Beispiel »cd c: \tmp« eingeben, dann wechselt MS-DOS auf Laufwerk C in das angegebene Verzeichnis, bleibt jedoch weiterhin auf Laufwerk A.

Beispiel:

```
cd c: \qc \lib
```

CHKDSK [<Laufwerk>]

[<Datei>] [<Optionen>]

Mit dem externen Chkdsk-Kommando überprüfen Sie den Inhalt einer Diskette oder Festplatte auf Fehler in der Dateianordnung.

Der optionale Parameter »Laufwerk« gibt das Laufwerk an, das geprüft werden soll. Der zweite Parameter »Datei« begrenzt die Überprüfung auf bestimmte Dateien.

Der Chkdsk-Befehl untersucht die Dateizuordnungstabelle (FAT = File Allocation Table) des Datenträgers. Es wird zum Beispiel festgestellt, ob jeder als belegt gekennzeichnete Bereich (Cluster) eines Datenträgers auch eindeutig einer Datei zugeordnet ist. Auch Kreuzverbindungen (ein Bereich ist fälschlicherweise mehreren Dateien zugeordnet) und defekte Unterverzeichnisse (ein Unterverzeichnis enthält ungültige Dateieinträge) werden von »chkdsk« gefunden.

Rufen Sie den Chkdsk-Befehl mit dem Parameter »/v« (»verbose«, deutsch: »ausführlich«) auf, werden die untersuchten Dateien einzeln aufgelistet.

Normalerweise zeigt »chkdsk« gefundene Fehler nur an, korrigiert sie jedoch nicht. Mit dem Parameter »/f« (»fix«, englisch für »befestigen«, »beheben«) aufgerufen, versucht das Programm, die Fehler so gut wie möglich zu beheben.

Alle Daten, die keiner Datei zugeordnet werden können, schreibt »chkdsk« in Dateien mit der Bezeichnung »filexxxx.chk«. Das Kürzel »xxxx« steht für eine fortlaufende Nummer, die automatisch vom Chkdsk-Kommando vergeben wird.

Beispiel:

```
chkdsk a: /f
```

CLS

Dieser interne Befehl löscht den Bildschirm. Es sind keine Parameter zulässig. Die Bildschirmfarbe wird aktualisiert.

Beispiel:

```
cls
```


**COMMAND [<Pfad>]
[<Optionen>]**

Den Kommandoprozessor »command.com« können Sie jederzeit neu aufrufen, zum Beispiel aus einem Anwendungsprogramm heraus. Mit dem Parameter »Pfad« geben Sie an, wie die Umgebungsvariable »comspec« gesetzt wird. Anhand dieser Variablen suchen viele Programme den Kommandoprozessor, falls er einmal nachgeladen werden muß. Der Parameter »/e:x« gibt die Größe des Umgebungsbereiches in Byte an. Erlaubt sind Werte für »x« zwischen 160 und 32768. Wird kein Wert angegeben, setzt MS-DOS automatisch den Wert 160 ein.

Die Option »/p« bewirkt, daß diese Kopie des Kommandoprozessors ständig im Speicher bleiben soll. Sie kann dann nicht mehr mit »exit« verlassen werden. Zusätzlich führt »command.com« die Autoexec-Datei im Hauptverzeichnis des aktuellen Laufwerks aus.

Als dritte Option läßt sich bei MS-DOS 4.0 »/msg« wählen. Sie gibt an, daß die erweiterten Fehlermeldungen in den Speicher geladen werden sollen.

Sie können den Kommandoprozessor auch so aufrufen, daß nur ein einziges Kommando, meist eine Stapeldatei, von ihm ausgeführt werden soll. Dazu setzen Sie hinter die Option »/c« den gewünschten Dateinamen. Nach Bearbeitung des Programms beendet MS-DOS den Kommando-Interpreter automatisch.

Beispiel:

```
command /c suchen.bat
```

COMP <Datei1> <Datei2>

Der externe Comp-Befehl erlaubt den Vergleich des Inhalts von zwei Dateien. »Comp« liest »Datei1« und vergleicht den Inhalt mit dem von »Datei2«. Dazu wird je ein Zeichen beider Dateien miteinander verglichen.

Stößt das Programm auf zwei Zeichen, die nicht übereinstimmen, gibt es eine entsprechende Meldung aus. Die Meldung gibt darüber Aufschluß, beim wievielten Zeichen der Datei die Unstimmigkeit aufgetreten ist, und zeigt außerdem die Werte an, die an der entsprechenden Stelle in beiden Dateien gefunden wurden.

Falls eine Datei mit einer anderen verglichen werden soll, die den gleichen Namen besitzt, sich jedoch in einem anderen Laufwerk befindet, genügt als Parameter »Datei2« die Angabe des Laufwerkbuchstabens mit einem nachfolgenden Doppelpunkt.

Beispiel:

```
comp c:\texte\kapitel1.txt
a:\backup1.txt
```

COPY <Quelle> <Ziel>

[<Optionen>]

COPY <Quelle 1> + <Quelle 2>

[... + <Quelle N>] <Ziel>

[<Optionen>]

Dieser interne Befehl dient zwei unterschiedlichen Funktionen. Zum einen können Sie damit Dateiinhalte in andere Dateien kopieren. Zum anderen ist es möglich, unterschiedliche Dateien zu verketteten und unter einem neuen Namen abzulegen.

Bei der ersten Anwendung gibt »Quelle« den Namen der Datei an, die kopiert werden soll. »Quelle« kann einen kompletten Pfad und eine Laufwerksbezeichnung enthalten. Das »Ziel« gibt den Namen der Datei an, die den Inhalt von »Quelle« erhalten soll. Auch »Ziel« kann einen Pfad und eine Laufwerksbezeichnung enthalten.

Falls »Ziel« noch nicht existiert, wird die Datei neu angelegt, andernfalls wird sie ohne weitere Meldung durch den Inhalt von »Quelle« überschrieben. Falls dies unerwünscht ist, sollten Sie eine Sicherheitsabfrage der Form

```
if not exist <Ziel> copy <Quelle>
<Ziel>
```

einsetzen.

Die Option »/a« des Copy-Befehls gibt an, daß es sich bei »Quelle« um eine ASCII-Datei (Text) handelt. Ihr Inhalt wird daher nur bis zum ersten Auftreten des Steuerzeichens [Ctrl-Z] kopiert, das als Kennung für das Dateiende gilt. Programmdateien können Sie mit dieser Option nicht kopieren.

Die Option »/b« bildet das Gegenstück zu »/a«. Sie gibt an, daß es sich bei »Quelle« um eine Binärdatei handelt und diese nicht durch ein [Ctrl-Z] abgeschlossen ist. Wenn Sie keine Optionen angeben, geht MS-DOS automatisch davon aus, daß »/b« verwendet werden soll.

Mit dem selten verwendeten Zusatz »/v« schließlich hat der Benutzer die Möglichkeit, nach dem Kopiervorgang überprüfen zu lassen, ob die Daten richtig übertragen wurden. Die Qualität der Massenspeicher ist heutzutage jedoch so gut, daß Sie auf diese Option verzichten können. Sie verlangsamt den Vorgang nur unnötig. Mit der zweiten Variante des Copy-Kommandos verketteten Sie Dateien miteinander. Dazu geben Sie eine Anzahl von Dateien »Quelle 1« bis »Quelle N« an, jeweils durch Pluszeichen miteinander verbunden. Den einzelnen Dateinamen können komplette Pfadangaben vorausgehen. Die Inhalte der Dateien kopiert MS-DOS dann in die angegebene Zielfeile.

Beispiel:

```
copy c:\word4\dosref.txt a:
copy kap1-1.txt+kap1-2.txt kapitel1.txt
```

CTTY <Eingabegerät>

Normalerweise erwartet MS-DOS Eingaben von der Tastatur und schickt Textausgaben an den Bildschirm. Diese Gerätekombination wird Konsole genannt (Abkürzung »con«). Mit dem Ctty-Befehl ändern Sie dieses Gerät. Zum Beispiel bewirkt »ctty com1«, daß die serielle Schnittstelle »com1« ab sofort als standardisiertes Ein- und Ausgabegerät aktiv ist.

Der Ctty-Befehl ist auch dann nützlich, wenn Sie die Fehlermeldungen eines Programms unterdrücken möchten. Dazu schreiben Sie in eine entsprechende Stapeldatei »ctty nul«, rufen dann das Programm auf und setzen anschließend mit »ctty con« wieder Bildschirm und Tastatur ein. Während des Programmlaufs sind keine Tastatureingaben zulässig.

Beispiel:

```
ctty nul
```

DATE [<Datum>]

Das interne Date-Kommando setzt das MS-DOS-Systemdatum neu. Das Format des Datums ist länderspezifisch und richtet sich nach dem Country-Eintrag in der Datei »config.sys«. Für ein deutsches MS-DOS besitzt das Datum die Form »tt.mm.jj« (t = Tag, m = Monat, j = Jahr), also zum Beispiel »25.8.89«. Wenn Sie den Date-Befehl zusammen mit einem gültigen Datum als Parameter aufrufen, wird dieser Wert als neues Systemdatum gesetzt.

Das Kommando »date« ohne Parameter bewirkt zunächst eine Ausgabe der Systemzeit. Danach können Sie entweder [Enter] drücken, um das Datum unverändert zu lassen, oder Sie geben einen neuen Wert ein.

Beispiel:

```
date 20.4.90
```

DEL <Dateiname> [/p]**ERASE <Dateiname> [/p]**

Der interne Del-Befehl (gleichbedeutend mit »erase«) löscht eine oder mehrere Dateien von Diskette oder Festplatte.

Der Dateiname kann ein kompletter Dateiname inklusive Pfadangabe oder eine Dateimaske mit Wildcards und Jokern oder nur ein Verzeichnis sein. So ist »del.« gleichbedeutend mit »del *.*«. Im ersten Fall löschen Sie das aktuelle Verzeichnis und der zweite Befehl löscht alle Dateien. Der belegte Speicherplatz der Dateien wird wieder freigegeben.

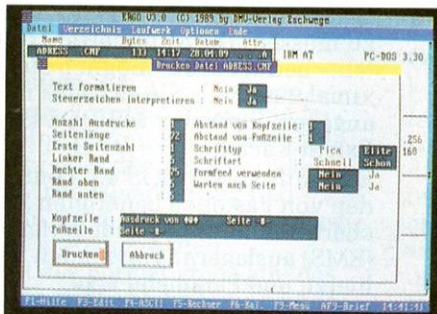
Für den Gebrauch des Del-Kommandos im Zusammenhang mit Dateimasken ist die Option »/p« (ab MS-DOS 4.0) nützlich. Sie bewirkt, daß vor jeder Datei eine Sicherheitsabfrage durchgeführt wird.

Beispiel:

```
del a:*.txt /p
```


ERGO 3.0

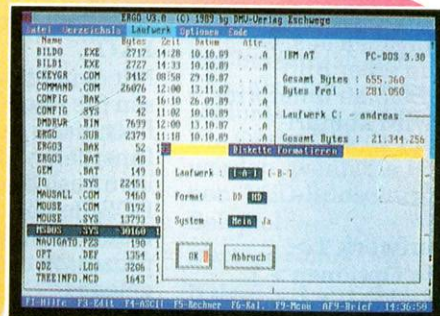
Die DOS-Benutzeroberfläche,
die einfach alles hat



Drucken

Komfortabler als mit
mancher Textver-
arbeitung arbeiten
Sie mit der ERGO-

Hausdruckerei: mit Layout, Kopf- und Fußzeilen,
Seitennummern, in verschiedenen Schrifttypen
und -arten, Punktkommandos und automatisch
formatiert – professionell, wie es sich gehört.



Formatieren

Auch das Formatieren
von Disketten
können Sie getrost
ERGO überlassen.

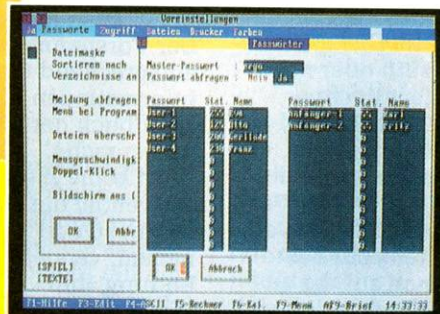
Ob 5 1/4- oder 3 1/2-Zoll, als Datendiskette oder
mit dem DOS-System – mit der komfortablen
menügesteuerten Formatier-Funktion von
ERGO sparen Sie sich Zeit, Ärger und Nerven.



TSR- Verwaltung

Wenn schon Be-
nutzeroberfläche,
dann richtig! Mit
der integrierten

TSR-Verwaltung von ERGO haben Sie alle Ihre
residenten Programme fest im Griff: Löschen,
Nachladen, Ermittlung des freien Speicherplat-
zes und Erinnerung an die Aufruftasten.



Paßwort

Mit dem profes-
sionellen ERGO-
Paßwortsystem
schützen Sie Ihre

wichtigen Daten vor Zugriff. Durch ein "Profil"
von Zugriffsberechtigungen für Dateien, Einstel-
lungen usw. brauchen Einsteiger keine Angst
vor Schäden durch Fehlbedienung zu haben.

Dateien:	Kopieren, Löschen, Umbenennen, Verschieben, Attribute
Verzeichnisse:	Wechseln, Anlegen, Löschen, Umbenennen, Verstecken, Verschieben
Laufwerk:	Wechseln, Kopieren, Formatieren, Optimieren
Sektor-Editor:	Suchen, Weitersuchen, Position, Abspeichern, Wiederherstellen
Optionen:	Voreinstellungen, Paßwort, Zugriffsrechte, Drucker-codes, Farben, Zeit/Datum setzen, TSR-Verwaltung, Systeminfo
OS-Shell:	sofort in DOS - mit Exit zurück in ERGO
Hilfe-Funktion:	ASCII-Tabelle, Kalender, Online-Manual
Taschenrechner:	Fließkomma, exponential, scientific, hexadezimal, binär, Winkel
Anwendermenü:	Programme lassen sich von diesem Menü aus starten
Editor:	Blockkommandos, Merkerfunktionen, Suchen und Ersetzen, Einrücken, Spaltenblock und Tabulator

Für 100% Software-kompatible PCs mit mindestens 384 kByte RAM und PC-/MS-DOS ab Version 3.0 aufwärts. Mit Einschränkungen auf älteren DOS-Versionen. Zur Verwendung der Maussteuerung ist eine Microsoft-kompatible Maus erforderlich.

ERGO DOS-Benutzer- oberfläche der Profiklasse

199,- DM (unverbindliche Preisempfehlung) Wenn Sie über den DMV-Bestellservice bestellen, gilt folgendes:			
Inland:		Ausland:	
Einzelpreis	199,- DM	Einzelpreis	199,- DM
zzgl. Versandkosten	4,- DM	zzgl. Versandkosten	6,- DM
Endpreis	203,- DM	Endpreis	205,- DM

69,- DM (unverbindliche Preisempfehlung) Wenn Sie über den DMV-Bestellservice bestellen, gilt folgendes:			
Inland:		Ausland:	
Einzelpreis	69,- DM	Einzelpreis	69,- DM
zzgl. Versandkosten	4,- DM	zzgl. Versandkosten	6,- DM
Endpreis	73,- DM	Endpreis	75,- DM

Update auf
ERGO 3.0

Bitte benutzen Sie die Bestellkarte.

DMV-Verlag · Postfach 250 · 3440 Eschwege

DIR [<Datei>] [<Optionen>]

Das interne Dir-Kommando zeigt den Inhalt eines Unterverzeichnisses an. Rufen Sie den Befehl ohne Parameter auf, erscheint das Inhaltsverzeichnis des aktiven Laufwerks. »dir« mit einer Dateimaske als Parameter, gibt nur die entsprechenden Dateien aus. »dir a:« ist ebenfalls gültig und bezieht sich dann auf das Laufwerk A. Das Dir-Kommando kennt zwei Optionen: Verwenden Sie die Option »/p« (»page«, »Seite«), dann erfolgt die Ausgabe langer Verzeichnisse zeilenweise.

Die Option »/w« (»wide«, »weit«, »breit«) gibt nur die Dateinamen ohne Größen- und Zeitangaben aus. Dadurch lassen sich in einer Bildschirmzeile mehr Dateien unterbringen, die Anzeige wird komprimierter.

Beispiel:

```
dir c:\dos\*.exe /p
```

DISKCOMP <Laufwerk 1> <Laufwerk 2> [<Optionen>]

Der Diskcomp-Befehl dient dem Vergleich des Inhalts zweier Disketten. Die beiden Parameter »Laufwerk 1« und »Laufwerk 2« geben die Laufwerksbezeichnungen an, die die zu vergleichenden Disketten enthalten. Dazu werden der Reihe nach alle Spuren und Sektoren Byte für Byte verglichen. Am Ende des Vergleichs teilt MS-DOS mit, ob die Disketten tatsächlich identisch sind oder nicht. Die Option »/1« teilt dem Rechner mit, daß nur die erste Seite der beiden Disketten verglichen werden soll. Die zweite Diskettenseite bleibt unbeachtet.

Der Parameter »/8« weist MS-DOS an, ausschließlich die ersten acht Sektoren jeder Spur mit denen auf der anderen Diskette zu vergleichen.

Beispiel:

```
diskcomp a:b
```

DISKCOPY <Quelle> <Ziel> [/1]

Der Diskcopy-Befehl kopiert Disketten. Die Parameter »Quelle« und »Ziel« stellen die Bezeichnungen der Diskettenlaufwerke dar, bestehen also aus einem Laufwerksbuchstaben mit einem nachfolgenden Doppelpunkt.

Beide Disketten müssen das gleiche Format aufweisen, weil »diskcopy« nicht nach Dateien, sondern nach Spuren kopiert. Wenn der Diskcopy-Befehl beim Kopieren feststellt, daß die Formate der Disketten unterschiedlich sind, wird die Zieldiskette entsprechend dem Format der Queldiskette automatisch neu formatiert. Der Diskcopy-Befehl fordert den Benutzer dazu auf, die Disketten zu wechseln, wenn »Quelle« und »Ziel« das gleiche Laufwerk darstellen. Die Option »/1« weist das MS-DOS

an, nur eine Seite der Diskette zu kopieren.

Beispiel:

```
diskcopy a:b
```

DOSSHELL

Dieser externe Befehl ermöglicht ab MS-DOS 4.0 das Starten der grafischen Benutzeroberfläche, die auch als MS-DOS-Shell bezeichnet wird. Bei diesem Befehl handelt es sich um eine Stapeldatei.

Beispiel:

```
dosshell
```

ERASE <Dateiname> [/p]

Siehe Beschreibung zum Del-Kommando

EXIT

Mit dem internen Exit-Befehl verlassen Sie den Kommandoprozessor, falls dieser zusätzlich geladen wurde. Innerhalb von Anwendungsprogrammen besteht oft die Möglichkeit, MS-DOS-Befehle auszuführen. Hierzu lädt das Programm den Kommandoprozessor nach. Mit dem Exit-Befehl gelangen Sie dann ins Anwendungsprogramm zurück.

Beispiel:

```
exit
```

EXE2BIN <Quelle> [<Ziel>]

Der Exe2bin-Befehl konvertiert eine Exe-Datei, die mit dem Linker einer Programmiersprache erzeugt wurde, in ein ausführbares Com-Programm. Der Vorspann der Datei wird dabei entsprechend umgewandelt. Nähere Informationen hierzu finden Sie im Handbuch der jeweiligen Programmiersprache.

Wird keine Zieldatei angegeben, so verwendet das Exe2bin-Kommando den in »Quelle« verwendeten Dateinamen, versieht ihn jedoch mit der Endung »bin«.

Das Exe2bin-Kommando ist nur bis MS-DOS 3.3 verfügbar. Die meisten späteren MS-DOS-Versionen enthalten das Kommando nicht mehr.

Beispiel:

```
exe2bin protect protect.com
```

FASTOPEN <Festplatte>

[= <Anzahl>] [, <Extents>]

Der Fastopen-Befehl wurde mit MS-DOS 3.3 eingeführt, um Dateizugriffe zu beschleunigen.

Der Fastopen-Befehl verhindert, daß MS-DOS bei jedem Zugriff auf eine Datei der Festplatte unter Umständen die gesamte Verzeichnisstruktur durchsuchen muß, bis die Datei gefunden ist. Dazu werden in einer Liste die Positionen der letzten Dateien, auf die zugegriffen wurde, gespeichert. Tritt dann ein Zugriff auf eine solche Datei auf, muß ihre Position nicht mehr in den Inhaltsverzeichnissen gesucht werden, da sie bereits im Puffer des Arbeitsspeichers steht.

Mit dem optionalen Parameter »Anzahl« geben Sie an, wie viele Zugriffswege MS-DOS speichern soll. Erlaubt sind hier Werte zwischen zehn und 999. Fehlt die Angabe, wird ein Wert von 34 benutzt.

Ab MS-DOS 4.0 gibt es einen weiteren Parameter. Über »Extent« bestimmen Sie, wie viele Einträge MS-DOS im sogenannten »Extent-Cache« verwalten soll. Im Extent-Cache merkt sich MS-DOS die Nummern aller Sektoren einer Datei, auf die zugegriffen wird. Dadurch können Sektornummern ermittelt werden, ohne die Dateizuordnungstabelle (FAT) laden zu müssen. Die Minimalzahl für »Extent« beträgt zehn, möglich sind maximal 999. Falls Sie »Extent« nicht angeben, verwaltet MS-DOS keinen Extent-Cache.

Ebenfalls ab MS-DOS 4.0 können Sie den von Fastopen benötigten Speicherbereich ins Expanded Memory (EMS) auslagern. Verwenden Sie hierzu den Parameter »/x«.

Beispiel:

```
fastopen c:100
```

FDISK

Der externe Fdisk-Befehl dient der Partitionierung einer Festplatte. Bevor eine Festplatte benutzt werden kann, muß sie zunächst mit dem Format-Befehl formatiert und dann partitioniert werden. Dazu dient das Fdisk-Kommando.

Der Befehl wird ohne Parameter aufgerufen. Als Resultat wird ein menüorientiertes Programm gestartet, mit dem Sie Partitionen erstellen, löschen, anzeigen und wechseln können.

Beispiel:

```
fdisk
```

FIND [<Optionen>]

« <Zeichenkette> » [< >] <Datei>

Mit dem externen Find-Befehl durchsuchen Sie eine Datei nach einer vorgegebenen Zeichenkette. Die Zeichenkette wird als Parameter übergeben und muß in Anführungszeichen eingeschlossen sein. Der Parameter »Datei« gibt die Datei an, die überprüft werden soll. Jokerzeichen (»*« und »?«) sind nicht zulässig. Dem Dateinamen kann ein Pfad und eine Laufwerksbezeichnung vorausgehen. Es gibt eine Reihe von Optionen, die sich gemeinsam mit dem Find-Befehl verwenden lassen. Diese Optionen müssen vor dem Parameter »Zeichenkette« gesetzt werden, nicht wie üblich am Ende der Befehlszeile.

Die Option »/c« (»count«, »zählen«) weist MS-DOS an, die Zeilen zu zählen, die die Zeichenkette enthalten. Die Option »/n« (»numbers«, »Zahlen«) bewirkt, daß alle Zeilen der Datei, die die gesuchte Zeichenkette

enthalten, mit einer Zeilennummer versehen ausgegeben werden.

Die Option »/v« (»versus«, »kontra«) schließlich sucht die Zeilen, die die angegebene Zeichenkette nicht enthalten, aus der Datei heraus. Im Zusammenhang mit »/c« bewirkt »/v« dann, daß alle Zeilen gezählt werden, die die Zeichenkette nicht enthalten.

»find« kann auch Texte bearbeiten, die aus einer Ausgabeumleitung (Pipe) stammen.

Beispiel:

```
find /c "werden" c:\texte\artikel.txt
dir *.exe | find "EXE" | sort
```

FORMAT < Laufwerk >

[< Optionen >]

Das externe Format-Kommando formatiert eine Diskette oder Festplatte. Dieser Vorgang dient zur Vorbereitung der Datenspeicherung und löscht alle auf der Diskette oder Festplatte befindlichen Daten.

Während des Formatierens gibt das Programm aus, welche Spur gerade bearbeitet wird. Außer bei Verwendung von Compaq DOS 3.31 können Sie das Formatieren einer Festplatte jederzeit mit [Ctrl-Break] ohne Datenverlust abbrechen, solange der Spürzähler noch nach oben zählt. Voraussetzung ist, daß Sie die Break-Funktion eingeschaltet haben (»break on«). Die Option »/b« (»Boot-Sektor«) bewirkt, daß auf der Diskette Platz für einen Boot-Sektor geschaffen wird. Dieser Boot-Sektor wird auf allen Disketten oder Festplatten benötigt, die startfähig sein sollen.

Mit dem Parameter »/s« (»system«) läßt sich die Diskette nach dem Formatieren booten.

Der Parameter »/v« (»volume label«, »Datenträgerbezeichnung«) veranlaßt »format«, Sie nach dem Formatieren nach einem maximal achtstelligen Namen für den Datenträger zu fragen. Dieser wird zum Beispiel beim Dir-Befehl mit ausgegeben.

Die weiteren Optionen beziehen sich auf die Formate des jeweiligen Datenträgers. Mit der Option »/1« wird nur die erste Seite der Diskette formatiert, »/8« bewirkt, daß mit acht anstelle von neun Sektoren pro Spur formatiert wird.

Bei der Verwendung von HD-Laufwerken, die Kapazitäten von 1,2 beziehungsweise 1,44 MByte erlauben, bewirkt die Option »/4«, daß die Diskette mit niedriger Schreiddichte (360 KByte bei 5 1/4-Zoll- und 720 KByte bei 3 1/2-Zoll-Laufwerken) formatiert wird.

Mit der Option »/n:xxx« (»number of sectors«, »Anzahl der Sektoren«) legen Sie fest, wie viele Sektoren pro Spur verwendet werden. Für »xxx« setzen Sie die gewünschte Anzahl ein.

Die Option »/t:xxx« (»tracks«, »Spuren«) schließlich erlaubt ähnlich »/s:xxx« eine Anpassung der Anzahl von Spuren pro Diskettenseite. Für normale Disketten- und Festplattenformate sind »/s:xxx« und »/t:xxx« nicht nötig.

Für Benutzer ab MS-DOS 4.0 hält das Format-Kommando eine weitere Option parat. Mit »/f:xxx« können Sie direkt die Kapazität in KByte des Datenträgers angeben, zum Beispiel »360« oder »720«, sofern das jeweilige Laufwerk diese Kapazität kennt.

Beispiel:

```
format a: /v
```

GRAFTABL [< Zeichensatztabelle >]

[< Optionen >]

Mit dem externen Graftabl-Befehl werden die Darstellungsmuster für die Zeichen 128 bis 255 des erweiterten ASCII-Zeichensatzes in den Speicher geladen. Diese Zeichencodes sind nur von Bedeutung, wenn Sie im Grafikmodus Text darstellen. Damit MS-DOS in der Lage ist, die Zeichen 128 bis 255 auch im Grafikmodus darzustellen, müssen Informationen dazu in den Speicher geladen werden.

Der Parameter »Zeichensatztabelle« gibt die Nummer der Zeichensatztabelle an, deren Zeichen verwendet werden sollen. Die folgenden Zeichensatztabellen sind möglich:

```
437 IBM-Standardzeichensatz
850 mehrsprachig
860 Portugal
863 Kanada (französisch)
865 Norwegen
```

Anstatt eine Zeichensatztabelle in den Speicher zu laden, können Sie mit dem Befehl auch erfahren, welche Zeichensatztabelle gerade aktiv ist. Dazu rufen Sie den Befehl mit der Option »/sta« auf. Verwenden Sie anstelle von »/sta« die Option »/?« (ohne vorangestellten Schrägstrich), so wird die komplette, verfügbare Zeichensatztabelle ausgegeben.

»Graftabl« installiert sich speicherresident, darf also nicht von anderen Anwendungsprogrammen oder Benutzeroberflächen aus aufgerufen werden.

Beispiel:

```
graftabl 437
```

GRAPHICS [< Drucker >]

[< Optionen >]

Mit der Taste [PrtScr] (oder [Druck] auf deutschen Tastaturen) geben Sie eine Hardcopy des Bildschirminhalts aus. Normalerweise funktioniert dieser Befehl nur im Textmodus. Um auch im Grafikmodus eine Hardcopy zu ermöglichen, gibt es das externe Graphics-Kommando. Es installiert eine entsprechende Programmroutine für CGA-, EGA- und VGA-Grafikkarten.

Der Parameter »Drucker« gibt an, welcher Druckertyp unterstützt werden soll. Erlaubt sind »color1«, »color4«, »color8«, »graphics«, »graphicswide« und »thermal«. Die Bezeichnungen beziehen sich auf die unterschiedlichen Typen der IBM-Drucker. Fehlt die Angabe, wird vom Druckertyp »graphics« ausgegangen. Wählen Sie für »Option« den Parameter »/r« (»revers«, »invertiert«) wird das Druckbild invertiert gedruckt, das heißt, alle gesetzten Punkte werden gelöscht und umgekehrt.

Mit der Option »/b« (»background«, »Hintergrund«) wird veranlaßt, daß auch der Hintergrund in der jeweiligen Farbe gedruckt wird. Das ist aber nur bei farbfähigen Druckern möglich, also bei Verwendung von »color4« oder »color8«.

Die Option »/lcd« muß verwendet werden, wenn die Hardcopy von einem LCD-Bildschirm stammt.

Beispiel:

```
graphics color4
```

JOIN [< Laufwerk >]

[< Verzeichnis >] [/d]

Mit dem externen Join-Befehl ordnen Sie ein Unterverzeichnis dem Namen eines Laufwerks zu. Der zweite Parameter »Verzeichnis« kennzeichnet einen gültigen Pfad zu einem Verzeichnis. Nach Ausführung des Join-Kommandos können Sie das entsprechende Laufwerk unter dem neuen Unterverzeichnis ansprechen (siehe auch Subst-Befehl).

Um die Umleitung wieder aufzuheben, rufen Sie »join« mit der Option »/d«, jedoch ohne den Parameter »Verzeichnis« auf.

Wird »join« ohne Parameter aufgerufen, zeigt es die gerade aktiven Umleitungen.

Beispiel:

```
join a: c:\texte\gbuch\word
rem cd c:\texte\gbuch\word wechselt
rem jetzt zum Laufwerk A
```

KEYB < Landes-Code >

[, < Zeichensatztabelle >]

[, < Datei >] KEYBGR

Mit dem externen Keyb-Kommando können Benutzer ab MS-DOS 3.3 einen landesspezifischen Tastaturtreiber installieren. Der Parameter »Landescode« gibt das Kürzel des Landes an, in dem der PC betrieben wird. Dieser Code besteht aus einem zwei Buchstaben langen Kürzel. Für den deutschen Tastaturtreiber geben Sie »gr« (für »german«) an.

Optional setzen Sie im Keyb-Befehl auch die Nummer der Zeichensatztabelle ein, die verwendet werden soll. Die folgenden Nummern sind zulässig:

```
437 IBM-Standardzeichensatz
850 mehrsprachig
860 Portugal
```


863 Kanada (französisch)
865 Norwegen

Falls sich die Datei »keyboard.sys« nicht im Hauptverzeichnis des aktuellen Laufwerks befindet, muß mit dem Parameter »Datei« der Name und der Pfad des Tastaturtreibers angegeben werden.

Benutzer von MS-DOS, älter als Version 3.3, müssen den landesspezifischen Tastaturtreiber mit einem eigenen Programm für jedes Land installieren. Für den deutschen Tastaturtreiber muß das Programm

»keybgr.com« (für »keyboard german«, also »Tastatur deutsch«) gestartet werden.

»keyb« ohne Parameter gibt das Kürzel des gerade aktiven Tastaturtreibers auf dem Bildschirm aus.

Beispiel:

```
keyb gr,437,c:\dos\keyboard.sys
```

LABEL [<Laufwerk> [<Zeichenkette>]]

Der interne Label-Befehl dient dazu, einer Diskette oder Festplatte einen Namen zu geben oder einen bestehenden Namen zu ändern. Der Parameter »Laufwerk« gibt den Namen des Laufwerks an, dessen Name gesetzt werden soll. Fehlt er, nimmt MS-DOS das gerade aktive Laufwerk an.

Der Parameter »Zeichenkette« gibt den Namen an, den der Datenträger erhalten soll. Die Zeichenkette darf maximal elf Zeichen enthalten, unter denen sich keines der folgenden Zeichen befinden darf:

* ? / \ | . , ; : + = < > [] () @ ^

Die einzelnen Zeichen der Zeichenkette werden in Großbuchstaben umgewandelt.

Wird die Zeichenkette nicht angegeben, so wird der Name vom MS-DOS nach dem Aufruf des Befehls abgefragt.

Beispiel:

```
label c: platte-c
```

MEM [<Optionen>]

Mit dem externen Mem-Befehl, der erst ab MS-DOS 4.0 verfügbar ist, kann die Belegung des zur Verfügung stehenden Arbeitsspeichers inklusive Extended und Expanded Memory angezeigt werden.

Die Option »/program« gibt zusätzlich eine Liste aller zur Zeit aktiven Programme aus. Mit der Option »/debug« wird die Ausgabe außerdem um Informationen zu den installierten Gerätetreibern ergänzt.

Die Anzeige der Größe von Extended und Expanded Memory erfolgt nur, wenn zusätzlicher Speicher auch tatsächlich installiert ist. Extended Memory kann nur auf AT-kompatiblen Geräten installiert werden.

Beispiel:

```
mem /program /debug
```

MKDIR <Verzeichnis>

MD <Verzeichnis>

Der interne Mkdir-Befehl (»make directory«), der meist in der verkürzten Version als »md« verwendet wird, legt ein neues Unterverzeichnis an. Das anzulegende Verzeichnis kann optional aus einem kompletten Verzeichnispfad mit Laufwerksangabe bestehen.

Beispiel:

```
md daten
md a:\texte\neu
```

MODE <Bildschirmmodus>

MODE <Gerät> <Angaben>

MODE <Gerät> CODEPAGE <Optionen>

Der externe Mode-Befehl ist einer der leistungsfähigsten MS-DOS-Befehle. Mit ihm setzen Sie Parameter für Geräte, bereiten die Verwendung von Zeichensatztabellen vor und schalten zwischen verschiedenen Darstellungsarten Ihrer Grafikkarte um. Die einfachste Anwendung des Mode-Kommandos liegt im Umschalten zwischen unterschiedlichen Bildschirmmodi. Für den Parameter »Bildschirmmodus« sind folgende Schlüsselwörter erlaubt:

```
mono 80-Zeichen-Textmodus bei MDA oder HGC
bw80 80-Zeichen-Textmodus für CGA/EGA/VGA, einfarbig
bw40 40-Zeichen-Textmodus für CGA/EGA/VGA, einfarbig
co80 80-Zeichen-Textmodus für CGA/EGA/VGA, mehrfarbig
co40 40-Zeichen-Textmodus für CGA/EGA/VGA, mehrfarbig
40 unveränderter Farbmodus, allerdings mit 40 Zeichen
80 unveränderter Farbmodus, allerdings mit 80 Zeichen
```

Sie verwenden den Mode-Befehl aber auch, um eine parallele oder serielle Schnittstelle zu konfigurieren. Dazu muß als Parameter »Gerät« die Bezeichnung des Geräts angegeben werden, das eingestellt werden soll:

```
lptx: parallele Schnittstelle
comx: serielle Schnittstelle
```

Der Buchstabe »x« im Gerätenamen kennzeichnet die Nummer der Schnittstelle, also »com1:« oder »lpt2:«.

Bei einer seriellen Schnittstelle muß hinter dem Gerätenamen zunächst die Baud-Rate festgelegt werden. Erlaubt sind 110, 150, 300, 600, 1200, 2400, 4800 oder 9600. Bei MS-DOS 4.0 sind auch 19200 Baud zulässig, allerdings nur auf ATs.

Danach wird – durch ein Komma getrennt – angegeben, ob die Paritätsprüfung ein- oder ausgeschaltet sein soll, beziehungsweise, welche Parität verwendet werden soll. Zulässig sind »n« (»none«, für keine Parität), »e« (»even«, für gerade Parität) oder »o« (»odd«, für ungerade Parität).

Dahinter erfolgt – wieder durch ein Komma getrennt – die Angabe, wie viele Datenbits die Schnittstelle verarbeiten soll. Dabei kommt eine Menge von sieben oder acht in Frage.

Als letzter Parameter steht – nach einem Komma – die Anzahl der Stopbits. Hier sind die Werte 1, 2 oder 3 erlaubt. Ein Beispiel für die Konfiguration der seriellen Schnittstelle »com2« auf 2400 Baud, keine Parität, acht Datenbits und ein Stopbit:

```
mode com2:2400,n,8,1
```

Für die Konfiguration einer parallelen Schnittstelle muß hinter dem Gerätenamen »lptx:« zunächst die Anzahl der Zeichen pro Zeile eingestellt werden. Hier sind entweder 80 oder 132 als Parameter erlaubt. Dahinter wird – durch ein Komma getrennt – die Anzahl der Zeichen pro Zoll gesetzt. Sie darf sechs oder acht betragen.

Ab MS-DOS 3.3 können Sie das Mode-Kommando auch einsetzen, um Zeichensatztabellen vorzubereiten. Zunächst geben Sie an, für welches Gerät eine Einstellung der Zeichensatztabelle erfolgen soll. Für den Parameter »Gerät« können Sie alle gültigen Gerätenamen (»con:«, »com1:« und so weiter) einsetzen. Hinter dem Gerätenamen folgt das Schlüsselwort »codepage« (auf deutsch »Zeichensatzseite«), das darauf hinweist, daß der Mode-Befehl auf Zeichensatztabellen angewendet werden soll. Statt dessen ist auch die Kurzform »cp« zulässig.

Möchten Sie eine Zeichensatztabelle auswählen, dann setzen Sie das Schlüsselwort »select = « (oder kürzer: »sel = «) und die Nummer der Zeichensatztabelle dahinter. Mögliche Zeichensatztabellen sind:

```
437 IBM-Standardzeichensatz
850 mehrsprachig
860 Portugal
863 Kanada (französisch)
865 Norwegen
```

Soll ein Gerät erstmalig auf die Verwendung mit einer Zeichensatztabelle vorbereitet werden, dann setzen Sie anstelle des Schlüsselwortes »select = « »prepare = « (auf deutsch »vorbereiten«) ein (oder kürzer: »prep = «). Dahinter folgt in Klammern die Nummer der Zeichensatztabelle und der Name (eventuell mit Pfadangabe) der Datei, in der sich die Zeichensatztabelle befindet.

Soll das Gerät auf mehr als eine Zeichensatztabelle vorbereitet werden, können Sie auch mehrere Nummern der Tabellen verwenden.

Beispiele:

```
mode mono
mode com1:9600,e,7,1
mode lpt1:80,6
mode con cp prep=(437 c:\dos\ega.cpi)
```


MORE

Der externe More-Befehl zeigt eine Datei seitenweise an. Die Ausgabe wird dann erst fortgesetzt, wenn der Benutzer eine Taste drückt.

Wenn Sie mit dem More-Kommando einen Dateiinhalt anzeigen möchten, dann leiten Sie diesen mit dem Zeichen »<« an den Filter weiter.

Beispiel:

```
more < text.txt
type text.txt | more
dir | find "TEXT" | sort | more
```

NLSFUNC [<Datei>]

Der externe Nlsfunc-Befehl (»national language support functions«, »Unterstützung von länderspezifischen Funktionen«) ist erst ab MS-DOS 3.3 verfügbar und erlaubt die Anpassung landesspezifischer Funktionen an die Bedürfnisse des Benutzers.

Erst nach der Ausführung des Nlsfunc-Kommandos können Sie Zeichensatztabellen auf bestimmten Geräten nutzen. »nlsfunc« muß ausgeführt werden, bevor zum Beispiel der Chcp-Befehl aktiviert werden kann.

»nlsfunc« lädt die landesspezifischen Informationen in den Speicher des PC. Sie stammen normalerweise aus der Datei »country.sys«. Der Parameter »Datei« kann daher entfallen, wenn »country.sys« sich im aktuellen Verzeichnis oder in einem mit »append« erweiterten Suchpfad befindet. Ist das nicht der Fall, muß der komplette Pfad und Dateiname angegeben werden, über den »country.sys« zu finden ist.

Das Nlsfunc-Kommando muß nur ein einziges Mal ausgeführt werden, um die landesspezifischen Daten zu installieren. Bei einem weiteren Aufruf wird eine Meldung ausgegeben, die besagt, daß »nlsfunc« bereits installiert ist.

Beispiel:

```
nlsfunc c:\dos\country.sys
```

PATH [<Pfad>; [<Pfad 2>; [... <Pfad N>;]]]

Der interne Path-Befehl wird verwendet, um über das aktive Verzeichnis hinaus nach ausführbaren Dateien in anderen Verzeichnissen suchen zu lassen. Der Parameter »Pfad« gibt das Unterverzeichnis an, das zusätzlich zum aktiven Verzeichnis von MS-DOS bei der Suche nach einer ausführbaren Datei (»exe«, »com«, »bat«) verwendet werden soll.

Sie können mehrere Pfade angeben, die alle durch einen Strichpunkt getrennt sein müssen.

Wenn Sie keinen Pfad angeben, gibt MS-DOS den aktuellen Suchpfad aus. Mit »set path = « oder »path ;« löschen Sie den Suchpfad.

Der Path-Befehl bezieht sich nur auf ausführbare Programme. Möchten Sie die Suche auf beliebige Dateien

erweitern, muß das Append-Kommando verwendet werden.

Beispiel:

```
path c:\tools;c:\dos;
```

PRINT [<Optionen>] [<Datei>]

Mit dem Print-Kommando lassen Sie Texte »im Hintergrund« drucken. Das bedeutet, daß der Drucker einen Text ausgibt, ohne daß das laufende Anwendungsprogramm unterbrochen werden muß. Der Druck des Textes und ein beliebiges Programm laufen scheinbar gleichzeitig ab. In Wirklichkeit wird dem Druckprogramm laufend Rechenzeit zugeteilt, allerdings so wenig, daß es im anderen Programm nicht auffällt.

Der Inhalt der angegebenen Textdatei wird nach dem Aufruf des Print-Befehls ausgedruckt. Dazu installiert MS-DOS ein Maschinenprogramm im Speicher, das den Ausdruck steuert. Falls sich die Datei nicht im aktiven Verzeichnis oder im Suchpfad befindet, kann dem Dateinamen eine Laufwerksangabe und ein Pfad vorangestellt werden. Sie können auch mehrere Dateinamen im Print-Kommando angeben oder die Jokerzeichen »*« und »?« verwenden. »print« merkt sich die angegebenen Dateien und druckt sie der Reihe nach aus.

Die Option »/b: <Größe>« (»buffer«, »Puffer«) setzt die Größe des internen Puffers zum Zwischenspeichern der zu druckenden Daten. Normalerweise wird ein Puffer von 512 Byte verwendet.

Die Option »/d: <Gerät>« (»device«, »Gerät«) legt fest, auf welchem Gerät der Ausdruck erfolgen soll. Normalerweise ist dies die erste parallele Schnittstelle, also »lpt1:« oder »prn:«. Fehlt die Angabe, dann fragt »print« beim ersten Aufruf, auf welches Gerät es drucken soll.

Mit der Option »/m: <Anzahl>« (»timeout«, »Aus-Zeit«) legen Sie fest, wie lange der Print-Befehl maximal warten darf, ein Zeichen an den Drucker zu senden. Ein PC löst 18,2mal in der Sekunde eine Unterbrechung aus, um Hintergrundprozesse wie den Print-Befehl zu steuern. Der Wert für »Anzahl« gibt an, wie viele dieser Unterbrechungen der Print-Befehl verstreichen lassen soll, wenn der Drucker nicht empfangsbereit ist. Der Standardwert ist 2. Sie benötigen diese Option normalerweise nur, wenn Sie die Fehlermeldung »device timeout error« erhalten. Sie besagt, daß der Drucker in der festgesetzten Zeit keine Bereit-Meldung geliefert hat. Erhöhen Sie dann den Wert in »/m:xxx« auf 3 oder 4.

Die Anzahl der Dateien, die sich der Print-Befehl merken kann, wird in einer Warteschlange festgehalten. Mit der Option »/q: <Anzahl>« (»queue«, »Schlange«) bestimmen Sie, wie viele

Dateien sich maximal in dieser Schlange befinden können. Der Standardwert für »Anzahl« ist 10.

Mit der Option »/s: <Anzahl>« (»slice«, »Scheibe«) legen Sie den Anteil an Gesamtrechenzeit fest, die dem Druckvorgang zugeordnet wird. Mit dem Parameter »Anzahl« bestimmen Sie, die wievielte Unterbrechung dem Print-Kommando zuteil werden soll. Bei einem sehr schnellen Drucker sollte der Wert höher ausfallen als der voreingestellte Wert von 8.

Die Option »/u: <Anzahl>« (»unacknowledged«, »nicht bestätigt«) gibt an, wie lange der Print-Befehl mindestens wartet, wenn ein Zeichen nicht beim ersten Versuch erfolgreich an den Drucker gesendet werden kann. Die Einheit von »Anzahl« bezieht sich wieder auf die Unterbrechung, die 18,2mal pro Sekunde vom PC ausgelöst wird.

Mit der Option »/c <Datei>« (»clear«, »löschen«) wird erreicht, daß der Print-Befehl alle noch in der Warteschlange stehenden Dateien entfernt, falls sie mit dem als »datei« angegebenen Parameter übereinstimmen.

Das Weiterdrucken wird sofort unterbrochen, falls die gerade im Ausdruck befindliche Datei ebenfalls als Parameter angegeben ist.

Wenn Sie – während das Print-Kommando bereits druckt – weitere Dateien in die Warteschlange einreihen möchten, müssen Sie diese über die Option »/p <Datei>« (»process«, »durchführen«) verwenden. Sie bewirkt, daß die als »Datei« aufgeführte Datei (es können auch mehrere sein) an die Liste von Print angehängt und ebenfalls ausgedruckt wird.

Die Option »/t« (»terminate«, »beenden«) verwenden Sie, wenn das Drucken aller Dateien sofort abgebrochen werden soll. Alle Dateien werden aus der Warteschlange entfernt. Eventuell druckt der Drucker aber noch eine Zeitlang weiter, da er die bereits empfangenen Daten noch in einem internen Puffer gespeichert hat.

Sie können den Print-Befehl mehrmals aufrufen und jeweils neue Dateien in die Schlange aufnehmen oder aus ihr löschen. Nach dem ersten Aufruf installiert das Print-Kommando ein Maschinenprogramm im Speicher, das den Zugriff auf den Drucker steuert.

Die Optionen »/b«, »/d«, »/m«, »/q«, »/s« und »/u« können nur beim ersten Aufruf von »print« verwendet werden.

Der Aufruf von »print« ohne Parameter zeigt Ihnen, welche Dateien noch zu drucken sind.

Beispiel:

```
print c:\texte\buch*.txt
```


PROMPT < Zeichenkette >

Mit dem internen Prompt-Kommando verändern Sie die Eingabe-Aufforderung von MS-DOS, das sogenannte »Prompt«. Nach dem Booten ist das Prompt so definiert, daß es den jeweiligen Laufwerksbuchstaben und ein Größerzeichen enthält, also zum Beispiel »C>«.

Der Parameter »Zeichenkette« gibt an, wie das Prompt fortan erscheinen soll. Sie können normalen Text mit einigen Sonderzeichen mischen, die der Befehl wie folgt interpretiert:

```
$a  (>and<) das Zeichen »&«
$b  das Zeichen »|«
$d  (>date<) das aktuelle Datum
$e  (>escape<) ein Escape-Steuerzeichen
    (Code 27)
$g  (>greater<) das Zeichen »>«
$h  (>Backspace< = [Ctrl-H]) ein Zeichen
    zurück
$l  (>less<) das Zeichen »<«
$n  (>name<) das aktuelle Laufwerk
$p  (>path<) das aktuelle Verzeichnis
$q  (>quotation<) das Zeichen »=«
$t  (>time<) die aktuelle Systemzeit
$v  (>version<) die verwendete MS-
    DOS-Version
$$  das Dollarzeichen
$_  Wagenrücklauf und Zeilenvorschub
    (>carriage return / line feed<),
    entspricht [Enter]
```

Sie können diese Sonderzeichen benutzen, um sich ein persönliches Prompt zu erstellen. Dazu kombinieren Sie sie mit gewöhnlichen Zeichen der Tastatur.

Wichtig ist vor allem das Sonderzeichen »\$«. Dieses Zeichen leitet eine Escape-Sequenz ein. Sie dient meistens dazu, bestimmte Funktionen des Treibers »ansi.sys« auszulösen. Beispiel:

```
prompt $v$_$p$g
```

RECOVER < Datei >

Das externe Recover-Kommando erlaubt die teilweise Rekonstruktion von zerstörten Dateien. Der Parameter »datei« gibt den Namen der Datei an, die fehlerhaft ist.

MS-DOS beginnt nach dem Aufruf des Recover-Kommandos damit, die fehlerhafte Datei Sektor für Sektor zu lesen. Wenn ein zerstörter Sektor auftaucht, werden die bis dahin gelesenen Informationen in eine neue Datei geschrieben. Dann wird der Sektor übersprungen und in der Dateizuordnungstabelle (FAT) als belegt gekennzeichnet. Hinter dem defekten Sektor beginnt MS-DOS nun erneut, die Sektoren auszulesen, bis wieder ein defekter Sektor auftritt. Dieser Vorgang wiederholt sich so lange, bis die Datei zu Ende ist.

Durch den Prozeß wird – je nach Anzahl der defekten Sektoren – die Datei in einzelne Fragmente unterteilt. Das Recover-Kommando legt diese Fragmente auf dem Datenträger unter dem Namen »filexxx.rec« ab.

Anstelle »xxxx« steht eine fortlaufende Nummer. Das erste Fragment der Datei bekommt den Namen »file0001.rec«.

Der Recover-Befehl sorgt also dafür, daß eine Datei bis auf die fehlerhaften Teile wieder lesbar wird. Handelt es sich um eine Textdatei, kann der zerstörte Inhalt wahrscheinlich wiederhergestellt werden. Bei ausführbaren Programmen ist dies mit ziemlicher Sicherheit nicht möglich.

Der Parameter »Datei« kann auch lediglich eine Laufwerksangabe sein. In dieser Form wird der Recover-Befehl benutzt, wenn das Inhaltsverzeichnis einer Diskette zerstört ist und alle Dateien wiederhergestellt werden sollen.

Beispiel:

```
recover a:text.asc
```

RENAME < Alter Name > [=]

< Neuer Name >

REN < Alter Name > [=]

< Neuer Name >

Das interne Rename-Kommando (meist in der verkürzten Form als »ren« verwendet) erlaubt das Umbenennen von Dateien. Die Datei mit dem Namen »Alter Name« erhält einen neuen Namen. »Alter Name« darf eine Laufwerks- und Pfadangabe enthalten. Als neuer Name der Datei wird der »Neue Name« vergeben. Hier darf keine Pfad- oder Laufwerksangabe enthalten sein.

Falls im ersten Namen Wildcards (»*« und »?«) enthalten sind, so müssen sie auch im neuen Dateinamen an der gleichen Stelle stehen.

Beispiel:

```
ren c:\texte\text*.txt*.doc
```

REPLACE < Quelle > < Zielverzeichnis > [< Optionen >]

Der externe Replace-Befehl dient dem Anlegen von Kopien von Dateiinhalten. Durch eine Reihe leistungsfähiger Optionen ist das Replace-Kommando eine interessante Alternative zu »backup« und »copy«.

Der Parameter »Quelle« gibt an, welche Dateien kopiert werden sollen. Er kann eine Laufwerksbezeichnung und eine Pfadangabe enthalten. Wildcards (»*« und »?«) sind erlaubt. Mit dem Parameter »Zielverzeichnis« wird festgelegt, in welches Verzeichnis die ausgewählten Dateien gelangen sollen.

Die Option »/a« (»append«, »anhängen«) weist MS-DOS an, nur die Dateien zu kopieren, die sich noch nicht im Zielverzeichnis befinden.

Mit der Option »/p« (»prompt«, »auffordern«) wird der Name jeder zu kopierenden Datei vor dem Kopiervorgang ausgegeben und der Benutzer zu einer Bestätigung aufgefordert. Die Option »/r« (»read only«, »Nur-Lesen«) bewirkt, daß auch die Datei-

en kopiert werden, die im Zielverzeichnis als schreibgeschützt gekennzeichnet sind.

Mit der Option »/s« (»selected«, »ausgewählt«) erreichen Sie, daß nur die Dateien kopiert werden, die im Zielverzeichnis bereits existieren.

Durch die Option »/u« (»update«, »auf den neuesten Stand bringen«) werden nur die Dateien im Zielverzeichnis ersetzt, die älter als ihr Gegenstück im Quellverzeichnis sind. Diese Option ist erst ab MS-DOS 4.0 verfügbar und erlaubt das effektive Erstellen von Sicherheitskopien.

Mit der Option »/w« (»wait«, »warten«) schließlich wird eine Verzögerung vor der Befehlsausführung erreicht. Der Benutzer kann eine andere Diskette einlegen. Erst danach wird das Replace-Kommando ausgeführt.

Natürlich schließen einige dieser Optionen einander aus. »/a« kann nicht gemeinsam mit »/s« oder »/u« verwendet werden.

Beispiel:

```
replace c:\texte\*.doc a:/u
```

RESTORE < Quelle > < Ziel >

[< Optionen >]

Mit dem externen Restore-Kommando können mit dem Backup-Befehl erstellte Sicherheitskopien auf die Festplatte zurückgespielt werden. Der Parameter »Quelle« gibt den Namen des Laufwerks an, das die Backup-Dateien enthält. Der Parameter »Ziel« spezifiziert das Laufwerk, auf das die Sicherheitskopien überspielt werden sollen.

Wenn »Ziel« ein Unterverzeichnis und einen Dateinamen (eventuell mit den Wildcards »*« und »?«) enthält, dann werden nur die angegebenen Dateien restauriert. Der Verzeichnisname muß in dem Fall der gleiche sein, von dem aus das Backup erfolgte.

Die Option »/s« (»subdirectories«, »Unterverzeichnisse«) weist MS-DOS an, alle Dateien in die entsprechenden Unterverzeichnisse der Festplatte zu kopieren. Dabei wird keine Auswahl vorgenommen. Falls die Unterverzeichnisse nicht existieren, werden sie automatisch angelegt.

Mit der Option »/m« (»modified«, »verändert«) werden nur Dateien gesichert, die seit dem letzten Backup auf der Festplatte verändert wurden. MS-DOS erkennt dies anhand des Archivbits der Dateien.

Die Option »/n« (»new«, »neu«) hat den Effekt, daß nur Dateien kopiert werden, die auf der Festplatte nicht existieren.

Mit der Option »/p« (»prompt«, »auffordern«) kann eine Sicherheitsabfrage vorgenommen werden. Bei allen Dateien auf der Festplatte, die entweder als schreibgeschützt (»read only«)

oder versteckt (»hidden«) markiert sind oder die nach dem letzten Backup erstellt oder verändert wurden, fragt MS-DOS, ob sie tatsächlich kopiert werden sollen.

Sehr nützlich sind die Optionen »/a«, »/b«, »/e« und »/t«, mit denen Sie die Auswahl der Dateien zum Zurückschreiben auf diejenigen beschränken können, die nach oder vor einem bestimmten Datum oder einer bestimmten Uhrzeit erstellt oder verändert worden sind. Hinter den Optionen »/a« und »/b« geben Sie dazu ein Datum im Format »tt.mm.jj« (t = Tag, m = Monat, j = Jahr) an. Bei »/a« werden alle Dateien zurückschrieben, die nach dem angegebenen Datum erstellt oder verändert wurden, bei »/b« alle Dateien, die vor dem Datum erstellt oder verändert wurden. Möchten Sie das Zurückschreiben außerdem auf Dateien beschränken, die nach oder vor einer bestimmten Uhrzeit erstellt worden sind, dann verwenden Sie die Optionen »/e« und »/t«. Dahinter wird im Format »hh:mm:ss« (h = Stunde, m = Minute, s = Sekunde) eine Uhrzeit angegeben. Die Datei wird bei »/e« nur dann kopiert, wenn sie vor der angegebenen Zeit erstellt oder verändert wurde, bei »/t« hingegen, wenn die Veränderung nach der Zeit stattgefunden hat.

Beispiel:

```
restore a: c:\texte\*.*/s/p
```

RMDIR < Verzeichnis >

RD < Verzeichnis >

Mit dem internen Rmdir-Kommando (das meist in der verkürzten Form »rd« verwendet wird) können Sie Verzeichnisse von einer Diskette oder Festplatte entfernen. Der Parameter »Verzeichnis« gibt das Verzeichnis an, das gelöscht werden soll. Falls der Parameter keine Laufwerksbezeichnung oder Pfadangabe enthält, wird das entsprechende Verzeichnis im aktiven Verzeichnis gesucht. Der Befehl funktioniert nur, wenn das zu löschende Verzeichnis keine Dateien mehr enthält, andernfalls wird eine Fehlermeldung ausgegeben.

Beispiel:

```
rd kalkulat
```

SELECT < Quelle > < Ziel >

< Landescode > < Tastaturcode >

Mit dem externen Select-Befehl kann eine Systemdiskette erstellt werden. Das Select-Kommando ist erst ab MS-DOS 3.3 verfügbar. »Select« legt eine neue Systemdiskette an und kopiert landesspezifische Informationen auf diese Diskette. Dazu werden die externen MS-DOS-Befehle »xcopy« und »format« benötigt.

Der Parameter »Quelle« gibt an, in welchem Laufwerk sich diese exter-

nen Befehle und die landesspezifischen Dateien befinden. Als landesspezifische Dateien benötigt der Select-Befehl die Dateien »keyb.com«, »keyboard.sys« und »country.sys«. Der Parameter »Ziel« legt fest, auf welchem Datenträger MS-DOS installiert werden soll. »Ziel« und »Quelle« sind jeweils gültige Laufwerksbezeichnungen. Der Parameter »Landescode« gibt das Land an, auf das MS-DOS angepaßt und installiert werden soll. Für Deutschland und Österreich wird hier 049, für die Schweiz 041 verlangt.

Mit dem Parameter »Tastaturcode« wird die Tastatur an die landesspezifischen Besonderheiten angepaßt. Für Deutschland und Österreich muß hier »gr« eingesetzt werden, der deutschsprachige Teil der Schweiz wird mit »sg«, der französischsprachige mit »sf« aktiviert.

Ab MS-DOS 4.0 ist das Select-Kommando nicht mehr verfügbar. Es wurde durch einen entsprechenden Menüpunkt in der grafischen Benutzeroberfläche ersetzt.

Beispiel:

```
select a: b: 049 gr
```

SET [< Umgebungsvariable > = < Zeichenkette >]

Mit dem internen Set-Kommando können den Variablen im MS-DOS-Umgebungsbereich Werte zugewiesen werden. Die Umgebungsvariablen sind von allen Programmen aus zugänglich. Sie dienen dazu, dem Betriebssystem oder einem Anwendungsprogramm bestimmte Zustände mitzuteilen.

Der Parameter »Umgebungsvariable« gibt den Namen der Variablen an, die verändert oder angelegt werden soll. Die Zeichenkette spezifiziert den Inhalt, den die Umgebungsvariable erhalten soll. Falls die Variable bereits existiert, wird ihr alter Inhalt durch den neuen überschrieben.

Wenn Sie das Set-Kommando ohne Parameter aufrufen, gibt MS-DOS die Namen aller Umgebungsvariablen und deren Inhalt aus.

Die Größe des Umgebungsbereiches ist beschränkt. Mit dem Shell-Befehl in der Datei »config.sys« und dem Parameter »/e« können Sie den Bereich vergrößern.

Beispiel:

```
set comspec=e:\command.com
```

SHARE [/f: < Speicher >] [/l: < Anzahl >]

Der externe Share-Befehl dient der Zusammenarbeit mit Netzwerken. Daher kann er nur funktionieren, wenn ein Netzwerk und die dazugehörige Software installiert sind. Mit dem Share-Kommando schützen Sie Dateien des Host-Rechners vor mehrfachem Schreibzugriff. Dieser Zu-

griffschutz wird über Funktionen des Betriebssystems aktiviert, die nur bei eingeschaltetem »share« zugänglich sind.

Der optionale Parameter »/f: < Speicher > « gibt die Größe in Byte des für das Datei-Sharing zur Verfügung stehenden Pufferbereichs an. Der voreingestellte Wert beträgt 2048. Mit dem Parameter »/l: < Anzahl > « legen Sie die Maximalzahl der Dateien fest, die gleichzeitig über das Netzwerk geöffnet sein dürfen. Der Standard hierfür ist 20.

Beispiel:

```
share /l:50
```

SORT [< Optionen >]

Das externe Sort-Kommando ist ein besonderer Befehl von MS-DOS. Wie bei »more«, so handelt es sich hierbei um einen Filter. Filter werden unter MS-DOS dazu benutzt, die Textein- und -ausgabe von Programmen zu filtern, also durch ein weiteres Programm bearbeiten zu lassen.

Mit dem Sort-Filter kann der Inhalt einer Datei oder die Ausgabe eines Programms zeilenweise alphabetisch sortiert werden. Um die Standardausgabe eines beliebigen Programms mit »sort« zu filtern, verknüpfen Sie das entsprechende Kommando über das Pipe-Symbol »|« mit dem Sort-Befehl (»... | sort«). Die Ausgabe erfolgt dann in sortierter Reihenfolge.

Die Option »/ + < Offset > « sorgt dafür, daß »sort« erst ab der durch »Offset« angegebenen Position jeder Zeile zu sortieren beginnt.

Die Option »/r« bewirkt, daß die Sortierreihenfolge umgedreht wird. Das Sortieren erfolgt dann nicht mehr von A nach Z, sondern von Z nach A. Eine Veränderung des Befehls wurde im Sort-Kommando der MS-DOS-Version 4.0 vorgenommen. Während bei allen MS-DOS-Versionen bis 3.3 Großbuchstaben vor den Kleinbuchstaben einsortiert wurden, sind ab MS-DOS 4.0 Groß- und Kleinbuchstaben gleichberechtigt.

Beispiele:

```
dir | sort | more
sort < text.txt > text.srt
```

SUBST < Laufwerk > < Verzeichnis > [/d]

Mit dem externen Subst-Befehl ordnen Sie einem Laufwerk den Namen eines Verzeichnisses zu. Dazu geben Sie mit dem Parameter »Laufwerk« den Buchstaben eines gültigen Laufwerks an. Der zweite Parameter (»Verzeichnis«) kennzeichnet einen gültigen Pfad zu einem Verzeichnis. Nach der Ausführung des Subst-Kommandos wird ein Zugriff auf das neue Laufwerk automatisch auf das entsprechende Verzeichnis umgeleitet (siehe auch Join-Befehl).

Sie verwenden diesen Befehl vor allem im Zusammenhang mit Anwendungsprogrammen, die keine Unterverzeichnisse verwalten können. Durch den Subst-Befehl kann ein solches Unterverzeichnis wie ein Laufwerk angesprochen werden. Um die Umleitung wieder aufzuheben, rufen Sie »subst« mit der Option »/d«, jedoch ohne den Parameter »Verzeichnis« auf.

Beispiel:

```
subst a: c:\dos
rem dir a: zeigt jetzt den
rem Inhalt von c:\dos
```

SYS <Laufwerk 1>

[<Laufwerk 2>]

Mit dem externen Sys-Befehl können die Systemdateien auf eine Diskette oder Festplatte übertragen werden. Dadurch machen Sie eine Diskette oder Festplatte startfähig.

Wenn beim Befehlsaufruf nur der Parameter »Laufwerk 1« angegeben wird, überträgt MS-DOS die Systemdateien vom aktiven Verzeichnis auf die Diskette in Laufwerk 1. Wird sowohl »Laufwerk 1« als auch »Laufwerk 2« angegeben, so kopiert MS-DOS die Systemdateien vom Datenträger in Laufwerk 1 auf den in Laufwerk 2.

Der Sys-Befehl überträgt die beiden versteckten Systemdateien »ibmbio.com« (oder »io.sys«) und »ibmdos.com« (oder »msdos.sys«) auf den entsprechenden Datenträger. Das Sys-Kommando kann nur aktiviert werden, wenn der Datenträger für die Verwendung als Startdiskette ausgerichtet ist, also entweder noch keine anderen Daten enthält oder mit der Option »/s« oder »/b« beim Formatieren mit dem Format-Kommando vorbereitet wurde. Anschließend müssen Sie mit dem Copy-Befehl noch den Kommando-Interpreter »command.com« der entsprechenden Systemdiskette kopieren.

Beispiel:

```
sys a:
copy c:\command.com a:
```

TIME [<Zeit>]

Das interne Time-Kommando dient dazu, das Systemdatum neu festzulegen oder abzufragen. Möchten Sie die Zeit neu festlegen, dann rufen Sie den Time-Befehl mit dem Parameter »Zeit« auf. Das Format lautet »hh:mm:ss« (h = Stunde, m = Minute, s = Sekunde), also beispielsweise »13:07:06«.

Rufen Sie den Time-Befehl mit dem Parameter »Zeit« auf, wird dieser als neue Systemzeit gesetzt. Anderenfalls fordert Sie MS-DOS auf, eine neue Uhrzeit einzugeben. Wenn Sie nur [Enter] drücken, bleibt die aktuelle Zeit unverändert.

Beispiel:

```
time 12:00
```

TREE [<Laufwerk>] [/f]

Der externe Tree-Befehl dient dazu, die Verzeichnisstruktur eines Datenträgers anzuzeigen. Das Tree-Kommando listet dazu alle Verzeichnisse in ihrer hierarchischen Reihenfolge auf. Wenn Sie die Tree-Anweisung mit einer gültigen Laufwerksangabe als Parameter aufrufen, dann erhalten Sie die Verzeichnisstruktur dieses Laufwerks.

Als Option können Sie »tree /f« eingeben. Dies bewirkt, daß außer den einzelnen Verzeichnisnamen auch die Inhalte dargestellt werden.

Weil die ausgegebene Verzeichnisstruktur sehr unübersichtlich ist, stellt der Tree-Befehl die Struktur ab MS-DOS 4.0 grafisch dar.

Beispiel:

```
tree a:
```

TYPE <Datei>

Der interne Type-Befehl dient der Ausgabe des Inhalts einer Datei. Falls es sich dabei nicht um eine ASCII-Textdatei handelt, sondern beispielsweise um ein Exe-Programm, können die Ergebnisse unvorhersehbar sein. MS-DOS versucht, den Programmcode dann als druckbare Zeichen zu interpretieren.

Beispiel:

```
type c:\texte\dosref.doc
```

VER

Die interne Ver-Anweisung zeigt Ihnen die Versionsnummer des geladenen Betriebssystems. Das Ver-Kommando erwartet keinerlei Parameter. MS-DOS meldet sich mit

```
MS-DOS Version ...
```

und PC-DOS beziehungsweise IBM-DOS gibt den Text

```
IBM Personal Computer DOS-Version ...
```

aus. Eventuell haben Sie auch eine herstellerspezifische Version, zum Beispiel von Compaq oder Epson.

Beispiel:

```
ver
```

VERIFY [ON | OFF]

»Verify« bedeutet zu deutsch »überprüfen«. Wenn Sie über den internen Befehl »verify on« die entsprechende Funktion von MS-DOS einschalten, so überprüft das Betriebssystem bei jedem geschriebenen Sektor durch einen sofort anschließenden Lesezugriff, ob die Daten fehlerfrei lesbar sind. Allerdings wird nicht überprüft, ob die Daten fehlerfrei auf dem Datenträger angekommen sind, sondern nur, ob der jeweilige Sektor ohne Probleme wieder gelesen werden kann.

Beispiel:

```
verify on
```

VOL <Laufwerk>

Das interne Vol-Kommando dient dazu, den Namen einer Diskette oder

Festplatte anzuzeigen. Unter MS-DOS können einem Datenträger Namen vergeben werden. Dazu dient der Label-Befehl oder das Format-Kommando mit der Option »/v«.

Darüber hinaus ermittelt der Vol-Befehl die laufend vergebene Seriennummer des Datenträgers, falls MS-DOS 4.0 verwendet wird.

Beispiel:

```
vol
```

XCOPY <Quelle> <Ziel> [<Optionen>]

Das Xcopy-Kommando stellt eine Erweiterung des Copy-Befehls dar. Über das Kopieren von Dateien hinaus ist die Xcopy-Anweisung in der Lage, komplette Verzeichnisstrukturen zu kopieren.

Der Parameter »Quelle« gibt an, welche Dateien kopiert werden sollen.

»Ziel« gibt darüber Aufschluß, wohin die Dateien zu kopieren sind. Sowohl »Quelle« als auch »Ziel« dürfen Laufwerksbezeichnung und Pfadangaben enthalten.

Die Option »/s« (»subdirectories«, »Unterverzeichnisse«) weist MS-DOS an, auch die Unterverzeichnisse aus »Quelle« zu kopieren. Falls diese in der Zielangabe noch nicht existieren, werden sie automatisch angelegt.

Durch Verwendung der Option »/e« (»empty«, »leer«) in Verbindung mit »/s« legt »xcopy« auch jene Verzeichnisse an, die in der Quellangabe leer sind, also lediglich als Eintrag in einem Verzeichnis vorhanden sind.

Mit der Option »/m« (»modified«, »verändert«) werden nur Dateien gesichert, die seit dem letzten Backup auf der Festplatte verändert wurden. Das Archivbit, anhand dessen die Auswahl erfolgt, wird nach dem Kopiervorgang gelöscht.

Analog zu »/m« arbeitet »/a« (»add«, »addiere«). Der Unterschied besteht darin, daß das Archivbit hierbei nicht verändert wird.

Mit der Option »/p« (»prompt«, »auffordern«) veranlassen Sie eine Sicherheitsabfrage für jede einzelne Datei. Sie können also direkt am Bildschirm mitverfolgen, welche Dateien aus einer Dateimaske kopiert werden sollen.

Durch die Option »/v« (»verify«, »überprüfen«) erreichen Sie, daß alle geschriebenen Daten auf eventuelle Lesefehler hin überprüft werden.

Die Option »/d: <Datum>« schließlich bewirkt, daß »xcopy« nur jene Dateien behandelt, die nach dem angegebenen Datum erstellt oder verändert wurden. Das Datumsformat lautet »tt.mm.jj« (t = Tag, m = Monat, j = Jahr).

Beispiel:

```
xcopy a: c:\subdir /s /e /v
```

(Martin Althaus/tr)

Referenz: Batch-Dateien

Bei der täglichen Arbeit mit dem PC kommt es oft vor, daß Sie eine Reihe von MS-DOS-Kommandos eingeben müssen, um eine immer wiederkehrende Aufgabe zu erledigen. Das Eingeben ist lästig, Tippfehler sind noch lästiger. Unter MS-DOS gibt es aber Mittel und Wege, diese Abläufe zu automatisieren: Batch-Dateien.

Gewöhnlich arbeitet man »im Dialog« mit MS-DOS, das heißt, man gibt ein Kommando ein, MS-DOS führt das Kommando aus, man prüft das Resultat, gibt ein weiteres Kommando ein, und so weiter. Wesentlich einfacher läßt sich über die Stapelverarbeitung mit MS-DOS kommunizieren. Bei dieser Eingabeform geben Sie einen ganzen Stapel Kommandos ein, bevor MS-DOS beginnt, die Kommandos auszuführen. Erst wenn die Eingabe vollständig abgeschlossen ist, beginnt MS-DOS mit der Abarbeitung. Sie können währenddessen getrost etwas anderes tun – MS-DOS wird den ganzen Stapel verarbeiten, ohne daß Sie noch einmal eingreifen müssen. Diese Stapelverarbeitung wird durch Batch-Dateien verwirklicht.

Eine Batch-Datei ist im Prinzip eine Textdatei, die Sie mit jedem Editor (zum Beispiel »edlin«) schreiben können. Der Inhalt einer Batch-Datei besteht aus einer Reihe von MS-DOS-Befehlen, jeweils ein Befehl pro Zeile, die von MS-DOS abgearbeitet werden, als ob sie über die Tastatur eingegeben würden. So läßt sich eine Folge von Befehlen zusammenfassen und anschließend mit wenig Aufwand (und ohne Tippfehler) ausführen.

Das bekannteste Beispiel einer Batch-Datei ist sicherlich »autoexec.bat«. Diese Batch-Datei wird jedesmal beim Start des Computers ausgeführt und enthält daher die Befehle, die für die Arbeit mit dem PC erforderlich sind. In Bild 1 ist der Inhalt einer typischen »autoexec.bat« abgebildet. Die hier enthaltenen Befehle laden den deutschen Tastaturtreiber und den Maustreiber, und sie legen die Form des MS-DOS-Prompts fest.

Wenn Sie die Batch-Datei in Listing 1 starten, sehen Sie, daß jede Zeile der Datei auf dem Bildschirm ausgegeben wird, bevor der entsprechende Befehl ausgeführt wird. Manchmal ist es aber nicht nötig und auch nicht erwünscht, die Befehle auszugeben. Um diese Ausgabe zu unterdrücken, können Sie den Befehl »echo« einsetzen. »echo« hat drei Formen:

```
echo off
echo on
echo »text«
```

»echo off« bewirkt, daß alle nachfolgenden Zeilen in einer Batch-Datei nicht ausgegeben werden, und »echo on« schaltet die Ausgabe wieder ein. Diese Befehle werden gezielt benutzt, um manche Teile einer Batch-Datei zu »verstecken« und andere Teile auf dem Bildschirm ausgeben zu lassen. Die dritte Form des Echo-Befehls ist sehr nützlich, wenn die Ausgabe durch »echo off« unterdrückt wurde. Eine Zeile wie

```
echo Jetzt werden die Dateien kopiert
gibt den Text nach dem Befehl
»echo« aus, das Wort »echo« wird jedoch nicht dargestellt. Die Batch-Datei in Bild 2 zeigt, wie der Echo-Befehl benutzt wird, um Meldungen im Klartext auszugeben.
```

Echo hat eine weitere Form, die allerdings nicht sonderlich brauchbar ist und hier nur der Vollständigkeit halber erwähnt wird. Steht nur »echo« in einer Zeile, so wird eine Meldung wie

```
echo ist aus
ausgegeben, die über den aktuellen Zustand des Echo-Schalters Auskunft gibt. Allerdings steckt ein kleines
```

```
cls
keybgr
mouse
prompt $p$g
```

Bild 1. Die bekannteste Batch-Datei, »autoexec.bat«

```
echo off
cls
echo Der deutsche Tastaturtreiber wird
installiert
keybgr
echo Der Maustreiber wird installiert
mouse
prompt $p$g
```

Bild 2. Der Befehl »echo« gibt Text auf dem Bildschirm aus

```
echo off
pause Bitte eine leere formatierte
Diskette in Laufwerk A: einlegen
copy *.bas a:
```

Bild 3. »pause« wartet auf Bestätigung vom Anwender

```
masm %1;
link %1;
exe2bin %1 %1.com
```

Bild 4. Mit Variablen werden Batch-Dateien flexibel

Problem in dieser Form des Echo-Befehls: Es ist damit nicht ohne weiteres möglich, eine leere Zeile darzustellen.

Ab der MS-DOS-Version 3.3 gibt es eine Methode, die Ausgabe der Batch-Datei am Bildschirm zu verhindern. Jede Zeile, die mit dem Zeichen »@« beginnt, wird ausgeführt, aber nicht ausgegeben. Sie können daher als erste Zeile in eine Batch-Datei die Zeile

```
@echo off
```

schreiben, um auch die Ausgabe des Echo-Befehls zu unterdrücken.

Gelegentlich muß der Anwender in den Ablauf einer Batch-Datei eingreifen, zum Beispiel eine Diskette in ein Laufwerk einlegen. Dabei ist es hilfreich, wenn dem Anwender gesagt wird, was zu tun ist, und wenn die Batch-Datei anschließend auf eine Bestätigung wartet, bevor sie weiterfährt. Zu diesem Zweck gibt es den Befehl »pause«. Das Format dieses Befehls ist

```
pause »Text«
```

wobei der Text auch weggelassen werden kann. »Pause« gibt den Text aus, gefolgt von der Meldung

Wenn bereit, eine beliebige Taste betätigen!

und wartet, bis eine Taste betätigt wird. Anschließend wird die Batch-Datei weiter ausgeführt. In Bild 3 sehen Sie ein typisches Beispiel der Anwendung dieses Befehls.

Wenn Batch-Dateien lang oder kompliziert werden, ist es empfehlenswert, kurze Erklärungen der Befehle einzufügen. Dies können Sie mit dem Befehl »rem« tun. Wie »echo« und »pause« besteht der Rem-Befehl aus dem Schlüsselwort, gefolgt von einem beliebigen Text. Im Unterschied zu den anderen Befehlen bewirkt »rem« keine Ausgabe: Der Text dient als Erklärung der Batch-Datei und hat keine weitere Bedeutung.

Batch-Dateien, wie sie bisher vorgestellt wurden, sind in ihren Anwendungsgebieten sehr eingeschränkt, weil nicht anpassungsfähig. Alle Befehle stehen fest in der Batch-Datei und werden einfach ausgeführt, es gibt keine Möglichkeit, den Ablauf zu beeinflussen. Um Batch-Dateien flexibler zu machen, kann man »Variablen« in einer Batch-Datei benutzen. Über diese Variablen, auch Argumente genannt, läßt sich der Ablauf der Batch-Datei auf ein spezielles Problem abstimmen, zum Beispiel, indem ein Dateiname übergeben wird:

```
makecom neuprog
```

Die Batch-Datei »makecom.bat« sehen Sie in Bild 4. An jeder Stelle, an der der Ausdruck »%1« steht, wird »neuprog« eingefügt. Das Ergebnis ist

eine Reihe von drei Befehlen, die den Quelltext »neuprog.asm« in das lauffähige Programm »neuprog.com« umwandeln. MS-DOS erkennt zehn solcher Variablen in einer Batch-Datei, die mit »%0« bis »%9« benannt werden können. Die Zuordnung zwischen diesen symbolischen Namen und der Eingabe, mit der Sie die Batch-Datei aufrufen, sehen Sie in Bild 5.

Um die Eingabe über die Variablen richtig ausnutzen zu können, ist ein Befehl notwendig, mit dem man die Eingabe abfragen kann. Dieser Batch-Befehl heißt »if« und hat die gleiche Wirkung, die von vielen Programmiersprachen bekannt ist: Eine Bedingung wird geprüft und ein Befehl nur dann ausgeführt, wenn das Resultat der Prüfung »wahr« ist. Der If-Befehl hat die Form

if Bedingung Befehl

wobei »Befehl« jedes MS-DOS-Kommando oder ein Programmname sein kann. Drei Bedingungen werden im If-Befehl erkannt. Zusätzlich kann am Anfang der Bedingung das Wort »not« stehen, um das Resultat zu verneinen.

Die Bedingung »exist« prüft, ob eine angegebene Datei existiert, zum Beispiel

```
if exist %1.asm
```

prüft, ob die an »makecom.bat« übergebene Datei überhaupt existiert. Im allgemeinen akzeptiert »exist« keinen Pfad, das heißt, Sie können nur die Existenz von Dateien im aktuellen Verzeichnis prüfen. Die Bedingung »exist« kann ebenfalls in Verbindung mit »not« benutzt werden, zum Beispiel erzeugt

```
if not exist %1.bak copy %1.txt %1.bak
```

eine Sicherheitskopie mit der Endung »bak«, falls noch keine solche existiert. Auch die Zeichen »*« und »?« dürfen mit »exist« benutzt werden:

```
if exist *.*
```

prüft, ob überhaupt Dateien im aktuellen Verzeichnis stehen.

Die zweite Bedingung des If-Befehls heißt »errorlevel« und erlaubt die Abfrage des (Miß-)Erfolgs eines vorhergehenden Befehls. Der Befehl ist leider nicht so universell einsetzbar, wie er vielleicht klingen mag, weil die meisten MS-DOS-Kommandos (zum Beispiel »copy«, »del« und »ren«) keine Meldung über ihren Ablauf geben. Glücklicherweise arbeiten die meisten Anwendungen etwas konsequenter. So können Sie nach erfolgter Assemblierung abfragen, ob Fehler aufgetreten sind:

```
tasm %1
if not errorlevel 1 tlink/v/t %1
```

Die Bedingung »errorlevel« vergleicht die Meldung vom zuletzt ge-

Bild 5. Die Zuordnung der Variablen in einer Batch-Datei

Name der Batch-Datei	Arg1	Arg2	Arg3	Arg4	Arg5	Arg6	Arg7	Arg8	Arg9
%0	%1	%2	%3	%4	%5	%6	%7	%8	%9

laufenen Programm mit der angegebenen Zahl und ergibt »wahr«, wenn die Meldung größer oder gleich der Zahl ist. In der Regel bedeutet »0« Erfolg, alle anderen Zahlen bedeuten Fehler. Manche Programme geben bestimmte Zahlen zurück, um bestimmte Fehler zu identifizieren. Dies hängt jedoch vom Programm ab; zuverlässig ist nur, daß »0« Erfolg bedeutet. Der If-Befehl im vorstehenden Beispiel bewirkt, daß »tlink« nur dann aufgerufen wird, wenn »tasm« erfolgreich ausgeführt wurde.

Die letzte Bedingung, die in Verbindung mit einem If-Befehl abgefragt werden kann, vergleicht zwei Zeichenketten. Diese Bedingung ist viel flexibler als die anderen beiden, der Bedeutung der Zeichenketten sind keine Grenzen gesetzt. In Bild 6 sehen Sie eine einfache Batch-Datei, die viel Tippen beim Verzeichniswechsel erspart. Von den ersten zwei Zeilen abgesehen, wird in jeder Zeile die Eingabe mit einem Kürzel verglichen und bei Gleichheit zu einem anderen Verzeichnis gewechselt. Eine spezielle Form des Vergleichs ist in der zweiten Zeile enthalten: Hier wird die Eingabe mit einer leeren Zeichenkette verglichen. Die Anführungszeichen sind erforderlich, weil MS-DOS sonst die Eingabe mit dem nachfolgenden Befehl »echo« vergleichen würde.

Mit dem If-Befehl kann man immer nur einen Befehl in einer Batch-Datei bedingt ausführen. Um mehrere Befehle bedingt auszuführen, setzen Sie einen weiteren Befehl, den Befehl »goto«, ein. Die Form des Goto-Befehls ist

```
goto label
```

»label« ist eine Zeichenkette, die in einer anderen Zeile in der Batch-

```
echo off
if "%1"==" " echo on
if %1==w cd \word
if %1==d cd \dbase
if %1==p cd \turbo
```

Bild 6. So werden Zeichenketten in einer Batch-Datei ausgewertet

```
echo off
if not exist %1 goto fehler
type %1
goto ende
:fehler
echo Datei %1 nicht gefunden!
:ende
```

Bild 7. Mit »goto« kann eine Batch-Datei auf Eingaben reagieren

Datei steht, zu der verzweigt werden soll. Diese Zeile muß mit einem Doppelpunkt beginnen wie

```
:label1
```

Ein Label kann beliebig lang sein, jedoch werden nur die ersten acht Zeichen von MS-DOS benutzt, das heißt, die ersten acht Zeichen müssen eindeutig sein.

In Bild 7 sehen Sie ein einfaches Beispiel der Anwendung dieses Befehls. Diese Batch-Datei prüft, ob eine Datei existiert, und gibt den Inhalt auf dem Bildschirm aus. Wenn die Datei nicht gefunden wird, wird eine entsprechende Meldung ausgegeben.

Wie bereits gesagt, können zehn Variablen in einer Batch-Datei benutzt werden. Mit dem Befehl »shift« läßt sich deren Anzahl vergrößern. Die Wirkung von »shift« stellt Bild 8 grafisch dar; wie Sie sehen, werden die Variablen um eine Stelle nach links verschoben. Dies hat den Vorteil, daß ein eventuelles elftes Argument nun über »%9« angesprochen werden kann. Durch wiederholte Benutzung von »shift« können beliebig viele Variablen benutzt werden.

»shift« ist besonderes brauchbar bei der Schleifenprogrammierung, wie zum Beispiel in der Batch-Datei in Bild 9. Hier wird in der Schleife jedesmal »%0« geprüft, da mit dem Shift-Befehl immer ein neues Argument in »%0« geschoben wird. Der If-Befehl prüft, ob »%0« leer ist, und beendet die Batch-Datei in diesem Fall. Sonst wird »del« aufgerufen und anschließend der gesamte Vorgang wiederholt. Hat diese Batch-Datei den Namen »delx.bat«, wird ein Aufruf wie

```
delx *.bak *.sik prog1.* temp.dat
```

alle Dateien in der Liste löschen. Es gibt auch eine andere Art, Schleifen in einer Batch-Datei zu verwirklichen. Der Befehl »for« realisiert eine Schleife, die aus einem Befehl besteht. Die Syntax des For-Befehls lautet

```
for %Variable in (Liste) do Befehl
```

»Variable« ist hier ein beliebiger Buchstabe. »Liste« enthält die Werte, die einer nach dem anderen der Variable zugewiesen werden. »Befehl« ist ein MS-DOS-Kommando oder -Programm, das für jeden Wert der Variable ausgeführt wird. Zum Beispiel führt der Befehl

```
for %a in (text.txt text.doc text.bak) do del %a
```

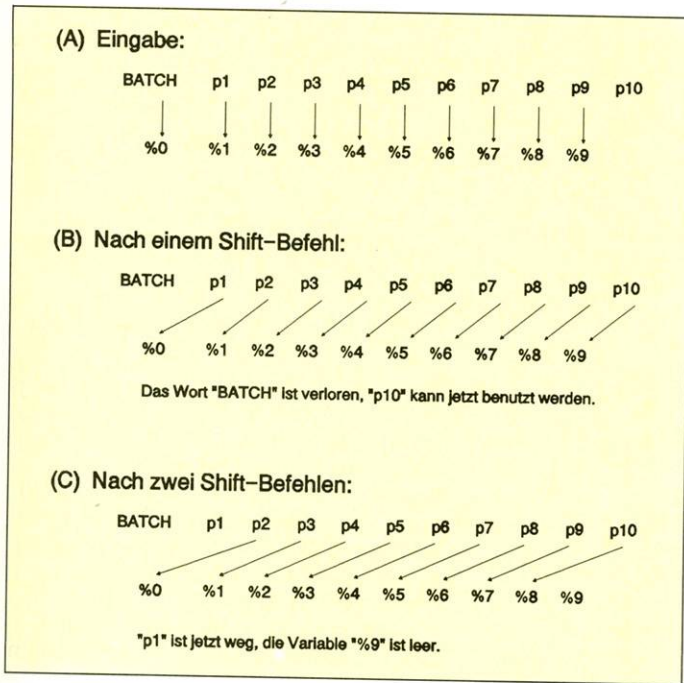



Bild 8. Die Wirkung des Befehls »shift«

das MS-DOS-Kommando »del« insgesamt dreimal wie folgt aus:

```
del text.txt
del text.doc
del text.bak
```

Darüber hinaus läßt sich der Befehl »for« auch für weitere Abfragen verwenden:

```
for %%a in (*.*) type %%a
```

gibt alle Dateien im aktuellen Verzeichnis auf dem Bildschirm aus oder

```
for %%a in (masm link) do %%a %1;
```

wandelt die Datei, deren Hauptname in der Eingabezeile steht, in ein lauffähiges Programm um.

Eine Einschränkung des For-Befehls besteht darin, daß nur ein Befehl ausgeführt wird. Diese Einschränkung können Sie umgehen, wenn der Befehl selbst eine weitere Batch-Datei ist. Wie das gemacht wird, hängt von der MS-DOS-Version ab. Versionen ab 3.3 besitzen dazu den speziellen Befehl »call«. Mit dem Befehl

```
for %%a in (*.asm) do call makecom %%a
```

können Sie zum Beispiel die Batch-Datei »makecom« wiederholt aufrufen, jedesmal mit dem Namen einer weiteren Datei im aktuellen Verzeichnis.

Um das gleiche unter einer älteren MS-DOS-Version zu erreichen, müssen Sie einen kleinen Trick einsetzen. Der entsprechende Befehl sieht so aus:

```
:loesche
shift
if "%0"==" " goto ende
del %0
got loesche
:ende
```

Bild 9. Der Befehl »shift« ermöglicht mehr als zehn Argumente in einer Batch-Datei

```
for %%a in (*.asm) do command /c makecom %%a
```

Die Angabe »command /c« startet »command.com« neu mit der Anweisung, den angegebenen Befehl auszuführen. Wenn der Befehl abgearbeitet ist, beendet sich »command.com« und kehrt zur Batch-Datei zurück. Der Trick ist erforderlich, weil »command.com« in älteren MS-DOS-Versionen nur einen einzigen »Platzhalter« für Batch-Dateien hat. Sobald eine zweite Batch-Datei gestartet wird, wird die erste »vergessen« – eine Rückkehr ist nicht möglich. Nun haben Sie die Regeln zur effektiven Batch-Programmierung kennengelernt. Wenn Sie bei Ihrer eigenen Arbeit am PC entdecken, daß Sie wiederholt eine ähnliche Folge MS-DOS-Kommandos eingeben, fassen Sie diese Kommandos in einer Batch-Datei zusammen. Sie machen sich das Leben einfacher.

Durch geschickte Programmierung kann man mit Batch-Dateien eine simple Benutzeroberfläche gestalten. Die einfachste Version ist das Ausgeben eines Menüs auf dem Bildschirm mit Hilfe des Grafikzeichensatzes und des Echo-Befehls. Im Menü sind die zur Verfügung stehenden Programme und dazu jeweils ein Buchstabe aufgelistet, beispielsweise:

```
d = Datenbank
```

Drückt der Anwender nun die Tasten [d] und [Enter], wird eine weitere Batchdatei, nämlich »d.bat« aufgerufen, in der sich die zum Start der Datenbank nötigen Parameter befinden. Am Ende von »d.bat« befindet sich dann wieder der Aufruf der Menüdatei.

(Thomas Little/ue)



Die moderne und kompetente Fachbuchreihe für die tägliche Anwendung für Einsteiger, Aufsteiger & Profis, leichtverständlich geschrieben.



Martin Böhmer
PC Tools, Deluxe
Der Umgang mit den Tools.
DM 48,—



Klaus Albrecht
Euroscript
Praxisbuch
Vom Einstieg bis zur professionellen Arbeit mit Tips und Tricks für Fortgeschrittene.
DM 48,—



Gerhard Bader
Das neue GEM
Praxisbuch,
Einführung in GEM mit DTP Sonderteil.
DM 50,—



Dr. Peter Haas
Open Access II
Einführung in Kalkulation, Graphik und Datenbank.
DM 58,—

Weitere wichtige Titel aus der Reihe Computer-Bücher

Ingrid Blum, **Windows**, PC-Programmpaket, MS-Windows, 1.0 / 2.0 für Ein-, Um- und Aufsteiger, DM 58,—

Erich Geldreich, **Ventura Publisher**, Version 1.0 / 1.1, mit Tips und Tricks, vom Einsteiger zum Profi, DM 69,—

Friedhelm Simberger, **Witchpen**, Praxisbuch mit vielen Abbildungen, DM 69,—

Hannsper Voltz, **Computer-Fachbegriffe von A - Z**, 2000 Begriffe und mehr, DM 29,80

Martin Böhmer, **Starwriter PC**, Praxisbuch zur Textverarbeitung, 2.0 / 3.0, DM 48,—

Richard Aicher, **Das Midi Praxisbuch**, (3. Auflage), Vom Einstieg bis zur professionellen Arbeit, DM 48,—

Signum MedienVerlag

Elsenheimerstraße 59, 8000 München 21
Telefon: 089-570 58 92, Fax: 089-57 07 03 41

Referenz: config.sys

Die korrekte Zusammensetzung der Steuerdatei »config.sys« hat großen Einfluß auf Funktionalität und Arbeitsgeschwindigkeit von Festplatten und Zusatzgeräten am PC. Beispielsweise verlangsamt eine zu geringe Anzahl an »Buffers« Ihre Plattenzugriffe beträchtlich. Diese Übersicht enthält den Befehlssatz für »config.sys« bis zu MS-DOS-Version 4.0.

BREAK = { ON | OFF }

Mit dem Break-Befehl legen Sie fest, wie der Rechner auf die Tastenkombination [Ctrl-C] beziehungsweise [Ctrl-Break] reagiert.

Der Break-Befehl in der Datei »config.sys« legt lediglich fest, ob das Betriebssystem auch bei Festplattenzugriffen und bei Bildschirmausgaben die Abbruchtastenfolge erkennen soll (»break on«) oder nur bei Tastaturabfragen des gerade laufenden Programms (»break off«). Bei eingeschaltetem »break« können Sie beispielsweise das Formatieren einer Diskette unterbrechen, bei »break off« hingegen nicht.

Nach dem Hochfahren eines PC ist die Break-Funktion ausgeschaltet. Sinnvoll ist in »config.sys« daher nur die Zeile:

```
break=on
```

BUFFERS = <Puffer-Anzahl> [,<Sektoren-Anzahl>][/x]

Der Buffers-Befehl bestimmt die Anzahl der internen Laufwerkspuffer zu je 512 Byte. Bei jedem Lesezugriff auf eine Diskette oder Festplatte kopiert MS-DOS den gerade übertragenen Sektor auch in einen der Puffer. Wird der Sektor von einem Programm dann ein zweites Mal angefordert, ist kein Laufwerkszugriff mehr nötig, denn die Daten befinden sich bereits im Speicher.

Beim Laden von Programmen bringen die Puffer allerdings keine Geschwindigkeitssteigerung. Sie sind hauptsächlich dazu gedacht, um das Suchen eines Programms in einem längeren Verzeichnispfad zu beschleunigen. Dementsprechend richtet sich die maximal sinnvolle Anzahl an Puffern nach der Größe der eingesetzten Festplatte. Bei einer 20-MByte-Platte sind 20 Puffer ein guter Wert. Eine volle 32-MByte-Partition mit zirka 100 Unterverzeichnissen und einer langen Pfaddefinition bringt bei 40 bis 50 Pufferspeichern gute Zugriffszeiten. Mehr ist allerdings nicht sinnvoll, weil der Rechner bei jedem Plattenzugriff zuerst die Pufferspeicher durchsucht.

Jeder der Puffer kann 512 Byte – einen Plattensektor – aufnehmen und belegt im Hauptspeicher

528 Byte (16 Byte zur Verwaltung des Puffers).

Der optionale Parameter »Sektoren-Anzahl« ist erst ab der MS-DOS-Version 4.0 verfügbar. Hiermit bestimmen Sie, wie viele Sektoren bei einem Lesezugriff auf den Datenträger gleichzeitig übertragen werden sollen. Ein Wert von null bis neun ist für diesen Parameter möglich.

Wenn Ihr Rechner über Expanded Memory nach LIM-Spezifikation verfügt, dann können Sie die Puffer bei MS-DOS 4.0 auch dorthin auslagern und haben so mehr Platz im Arbeitsspeicher. Dazu geben Sie beim Buffers-Befehl den Parameter »/x« mit an. Sie müssen dann aber mit dem Gerätetreiber »xmae2ems.sys« die Seite 255 des Expanded Memory auch für die Puffer freihalten.

Beispiel:

```
buffers=20
```

COUNTRY = <Ländercode>

[,<Zeichensatztafel>[,<Datei>]]

Ein PC ist so konzipiert, daß er in einer Vielzahl von Ländern eingesetzt werden kann. Damit Datum- und Zeitformat (zum Beispiel beim Dir-Befehl) korrekt ausgegeben werden, legen Sie mit dem Country-Kommando das Land fest, dessen Datums- und Zeitformat der PC verwenden soll. Mit dem Parameter »Ländercode« geben Sie die Nummer eines der unterstützten Länder an. Diese Nummer ist identisch mit der international üblichen Telefonvorwahl des entsprechenden Landes. Für Deutschland lautet der Länder-Code daher »049«. Die führende Null dürfen Sie nicht weglassen.

Ab der MS-DOS-Version 3.3 werden vom Betriebssystem Zeichensatztabellen unterstützt. Anstelle des üblichen IBM-Zeichensatzes können Sie auch andere Tabellen aktivieren. Wird keine Zeichensatztafel angegeben, so arbeitet MS-DOS mit dem gewöhnlichen IBM-Zeichensatz unter der Nummer 437. Zulässig sind folgende Werte:

```
850 – mehrsprachig
860 – Portugal
863 – Kanada (französisch)
865 – Norwegen
```

Mit dem Parameter »Datei« können Sie ab der MS-DOS-Version 3.3 die

Steuerdatei »country.sys« auch in ein Unterverzeichnis verlegen.

Beispiel:

```
country=049,,c:\dos\country.sys
```

DEVICE = <Gerätetreiber> [<Parameter 1> ... [<Parameter N>]]

Mit dem Device-Befehl binden Sie Gerätetreiber in das Betriebssystem ein. Dazu setzen Sie für »Gerätetreiber« den Pfad und Dateinamen des Treibers ein. Meistens haben diese Programme die Namensweiterung »sys«.

Beispiel:

```
device=ansi.sys
```

FCBS <Anzahl>

[,<Anzahl geschützt>]

Die ersten MS-DOS-Versionen ab 1.0 verwalteten Dateien über die sogenannten FCBs (für »File Control Block«, deutsch: »Datei-Steuerblock«). Ein FCB ist eine Datenstruktur, die beim Zugriff auf eine Datei auf Maschinenebene angelegt werden muß. Sie enthält den Dateinamen, Zugriffsrechte und ähnliche Informationen.

Mit dem FCBS-Kommando bestimmen Sie die Anzahl der FCBs, die maximal verfügbar sind. Diese Anzahl ist meistens gleichbedeutend mit der Anzahl gleichzeitig geöffneter Dateien. Für jeden FCB reserviert MS-DOS 48 Byte im Arbeitsspeicher. Mit dem Parameter »Anzahl geschützt« geben Sie an, wie viele der FCBs vor dem automatischen Schließen durch MS-DOS geschützt werden sollen. Ist keiner der FCBs geschützt, schließt MS-DOS den zuerst geöffneten FCB, wenn ein neuer FCB angefordert wird und alle FCBs schon belegt sind. Dadurch kann es zu Datenverlusten kommen.

Das FCBS-Kommando ist nur für ältere MS-DOS-Versionen oder sehr alte Programme (vor 1982) von Bedeutung. Ab der MS-DOS-Version 2.0 wurden die »File Handles« eingeführt, mit der sich Dateien sehr viel effektiver verwalten lassen als über FCBs. FCBs werden daher nur noch von älteren Programmen aus der Zeit verwendet, als die File-Handle-Methode noch nicht bekannt war.

Beispiel:

```
fcbs=20
```

FILES = <Anzahl>

Das Files-Kommando ist das Gegenstück zum FCBS-Befehl. Es gibt die Anzahl der »File Handles« an, die dem System maximal zur Verfügung stehen. In der Regel entspricht diese Zahl – wenn eine Datei nicht mehrmals geöffnet wird – der maximalen Anzahl der gleichzeitig zu öffnenden Dateien. Der größte Wert für »Anzahl« liegt bei 255, ab MS-DOS 4.0 bei 65536. Für jedes File Handle reserviert das Betriebssystem 38 Byte des Arbeitsspeichers.

Während bei MS-DOS vor Version 3.3 die einmal eingestellte Anzahl der Handles nicht mehr verändert werden kann, besteht ab der Version 3.3 auch die Möglichkeit, diesen Wert nach dem Systemstart zu verändern. Dazu steht Programmierern die Funktion »67hex« von Interrupt 21hex zur Verfügung.

Im allgemeinen sollten Sie den Files-Wert auf 30 festlegen. Dies stellt mit den meisten Anwendungsprogrammen einen ausreichenden Wert dar und muß nur bei sehr umfangreichen Datenbanken noch weiter erhöht werden.

Beispiel:

```
files=20
```

INSTALL = <Programmname>

[<Parameter 1> ...

[<Parameter N>]]

Ab der MS-DOS-Version 4.0 ist der Install-Befehl verfügbar. Er erlaubt das Laden von speicherresidenten Programmen schon während der Phase der Konfiguration. Bis zur Version 4.0 konnten residente Programme (zum Beispiel der deutsche Tastaturreiber »keybgr« beziehungsweise »keyb«) erst in der Datei »autoexec.bat« automatisch geladen werden.

Der Install-Befehl ist für die MS-DOS-eigenen residenten Programme wie »fastopen«, »keyb«, »nlsfunc« und »share« gedacht. Der Vorteil liegt in der besseren Platzierung der residenten Programme im Hauptspeicher und einer damit verbundenen Zugriffsoptimierung. Außerdem können Sie diese Programme auch dann nutzen, wenn Sie beispielsweise einen anderen Kommandointerpreter einsetzen, der – aus welchen Gründen auch immer – das direkte Laden von »keyb« nicht erlaubt.

Beispiel:

```
install=c:\dos\keyb gr,,
c:\dos\keyboard.sys
```

LASTDRIVE = <Laufwerksbuchstabe>

Mit dem Kommando »lastdrive« legen Sie fest, welches die alphabetisch höchste Laufwerksbezeichnung ist, die MS-DOS an ein Disketten- oder Festplattenlaufwerk vergibt. Ohne diesen Befehl sind die Bezeichnungen A bis E verfügbar.

Eine Erhöhung dieses Wertes kann mit dem Lastdrive-Befehl erfolgen. Der Parameter »Laufwerksbezeichnung« gibt den Buchstaben des maximal ansprechbaren Laufwerksbuchstabens an. Eine Erhöhung ist dann sinnvoll, wenn Sie während der Arbeit mit dem PC neue logische Laufwerke (zum Beispiel mit den Dienstprogrammen »subst« oder »join« oder aber in einem Netzwerk) anmelden möchten.

Beispiel:

```
lastdrive=z
```

MULTITRACK = { ON | OFF }

Dieser Befehl ist nur ab MS-DOS 4.0 verfügbar und steht unter PC-DOS (beziehungsweise IBM-DOS) nicht zur Verfügung.

Zur Zugriffsoptimierung wurden ab MS-DOS 3.3 Änderungen in den internen Hardware-Treibern für Disketten und Festplatten vorgenommen, mit denen automatisch mehrere Spuren direkt hintereinander gelesen und beschrieben werden (»Burst-Modus«). Bei nicht hundertprozentig kompatiblen Festplatten-Controllern traten dadurch zum Teil Fehler auf. Um diese Fehler zu beheben, können Sie unter MS-DOS 4.0 das mehrfache Schreiben und Lesen von Spuren (»Multitrack« genannt) unterbinden. Dazu wird der Multitrack-Befehl mit der Option »off« aufgerufen. Da die Probleme nur bei Controllern auftreten, die in IBM-PCs nicht verwendet werden, fehlt der Befehl im PC-DOS (IBM-DOS).

Beispiel:

```
multitrack=off
```

REM <Bemerkung>

Mit dem ab MS-DOS 4.0 verfügbaren Rem-Befehl (»remark«, englisch für »Bemerkung«) fügen Sie Kommentare in die Datei »config.sys« ein. Der Text des Kommentars wird, durch ein Leerzeichen getrennt, direkt hinter den Rem-Befehl gesetzt. MS-DOS ignoriert beim Bearbeiten von »config.sys« alle Zeilen, die mit dem Schlüsselwort »rem« beginnen. Der Kommentar wird nicht auf dem Bildschirm ausgegeben. Rem-Zeilen dienen der Dokumentation einer Konfigurationsdatei.

Beispiel:

```
rem Dies ist ein Kommentar
```

SHELL = <Dateiname> SHELL = COMMAND.COM [/E: <Umgebungsgröße>] [<Comspec-Pfad>] [/P] [/MSG]

Mit diesem Befehl besteht ab MS-DOS 3.3 die Möglichkeit, den Kommandoprozessor zu bestimmen, mit dem die Autoexec-Datei durchlaufen werden soll. Normalerweise wird der Kommandoprozessor (auch »Shell« genannt) »command.com« verwendet. Mit dem Parameter »Dateiname« wird der Name und der Pfad des zu verwendenden Kommandoprozessors angegeben.

Falls Sie »command.com« verwenden möchten, kann die Shell-Angabe in »config.sys« entfallen, es sei denn, »command.com« befindet sich nicht im Hauptverzeichnis. In einem solchen Fall muß der vollständige Pfad und Dateiname von »command.com« angegeben werden. Sie sollten allerdings beachten, daß einige Program-

me (selbst die Dienstprogramme von MS-DOS) den Kommandointerpreter ungeachtet der Comspec-Variablen im Hauptverzeichnis erwarten. Es ist daher nicht empfehlenswert, »command.com« in ein Unterverzeichnis zu verlegen und aus dem Hauptverzeichnis zu löschen.

Mit dem optionalen Parameter »/e« von »command.com« geben Sie die Größe des Umgebungsbereiches (»Environment«) in Byte an. Erlaubt sind Werte zwischen 160 und 32768. Wird kein Wert angegeben, setzt MS-DOS 160 ein.

Die Option »/p« besagt, daß »command.com« resident im Speicher bleibt und die Datei »autoexec.bat« ausgeführt wird.

Der Standard-Interpreter »command.com« kennt noch eine weitere Option, die ohne speziellen Parameter eingeleitet wird. Ein beliebiger Laufwerks- und Verzeichnisname gibt an, auf welchen Pfad die Umgebungsvariable »comspec« gesetzt werden soll.

Als letzte Option läßt sich bei »command.com« ab MS-DOS 4.0 der Parameter »/msg« wählen. Die Option veranlaßt das Betriebssystem, die erweiterten Fehlermeldungen in den Speicher zu laden.

Beispiel:

```
shell=c:\dos\command.com c:\dos /p
/e:512
```

STACKS = <Stapel>, <Größe>

Mit diesem Befehl verändern Sie die Anzahl und die Größe der »Stack-Frames«. Stack-Frames werden vom Betriebssystem beim Aufruf von Unterprogrammen beziehungsweise Interrupts benutzt, um Daten zwischenzuspeichern und Parameter zu übergeben.

Das Stacks-Kommando sollte nur von erfahrenen PC-Benutzern angewendet werden. Eine Änderung der voreingestellten Werte ist normalerweise nicht erforderlich.

Beispiel:

```
stacks=0,0
rem Unterstütze Stack-Frames nicht
```

SWITCHES = /K

Erweiterte Tastaturen senden gegenüber den konventionellen Tastaturen zum Teil andere Tastencodes an den PC. Aus Kompatibilitätsgründen kann es deshalb in manchen Fällen notwendig sein, eine erweiterte Tastatur als konventionelle Tastatur zu konfigurieren. Alle Codes, die dem Tastaturreiber nicht explizit bekannt sind, werden dann ignoriert und gar nicht erst an Anwendungsprogramme weitergeleitet.

Beispiel:

```
switches=/k
```

(Martin Althaus/tr)

Vorsicht, Umleitung!

Umleitungen, Filter und Pipes unter MS-DOS

Aufgrund der vielen Dienstprogramme treten die Eigenschaften, die das Betriebssystem MS-DOS bereits mitbringt, häufig in den Hintergrund. Einige sehr nützliche Funktionen für den Umgang mit Dateien lernen Sie in diesem Beitrag kennen.

Das Betriebssystem MS-DOS arbeitet mit Standardgeräten für die Ein- und Ausgabe. Nach dem Booten des Betriebssystems fungiert die Tastatur als Standardeingabegerät und der Bildschirm als Standardausgabegerät. Weil die Entwickler des Betriebssystems für diese beiden Geräte keine Vorschriften vorgegeben haben, kann jedoch auch jedes andere Gerät als Ein- oder Ausgabegerät einspringen. So dürfen Sie zum Beispiel die serielle Schnittstelle als Standardein- und Standardausgabegerät definieren, so daß Ihr Computer über diese Schnittstelle bedient wird. Dies geschieht mit dem MS-DOS-Befehl

```
ctty com1
```

Achten Sie aber darauf, daß weiterhin eine Verbindung zum Computer besteht, über die Eingaben erlaubt sind. Sie könnten nämlich theoretisch auch den Drucker als Standardeingabegerät definieren, nämlich über

```
ctty lpt1
```

Doch danach müssen Sie einen Reset durchführen, da Sie über den Drucker dem Computer keine Befehle mehr erteilen können.

Wollen Sie eine solche Umdefinition nur für eine bestimmte Aufgabe nutzen, so tragen Sie durch eine Stapeldatei dafür Sorge, daß nach der Ausführung alles wieder »normal« ist:

```
ctty com1
-Anwendung-
ctty con
```

Auch die Entwickler von MS-DOS waren sich bewußt, daß diese Struktur der Ein- und Ausgabesteuerung viel Freiraum bietet, und schufen daher einige Operatoren zur Datenumleitung, die für eine temporäre Ein-/Ausgabumleitung sorgen. Dies sind die folgenden Zeichen:

```
<- woher kommen die Daten
>- wohin gehen die Daten
>>- wo werden die Daten angefügt
```

Um die angeführten Beispiele nachzuvollziehen, legen Sie eine Datei »test.txt« (Bild 1) an. Dies erledigen Sie am besten mit Ihrer Textverarbeitung, mit der Sie den Text unformatiert speichern. Sie können den Text

jedoch auch direkt von der Tastatur auf die Diskette schreiben. Dies erreichen Sie mit der Befehlsfolge

```
copy con test.txt
```

Die Befehlsfolge schließen Sie mit [Enter] ab. Nun geben Sie die Namen ein, die Bild 1 zu entnehmen sind. Jede Zeile wird dabei wiederum mit einem [Enter] abgeschlossen. Nachdem Sie alle Zeilen eingegeben haben, drücken Sie die Taste [F6] oder [Ctrl-Z], was dem Betriebssystem das Dateiende anzeigt. Achten Sie bei der Eingabe bitte darauf, daß Sie die Spaltennamen ab der 20. Spalte und die Ortsname ab der 40. Spalte eingeben.

Sortieren geht über Studieren – und zwar über das Studium des Sort-Befehls

Nun aber zur ersten Anwendung der Datei umleitung. Geben Sie den Befehl

```
sort < test.txt
```

ein. Dieser Befehl besagt, daß Sie das Dienstprogramm »Sort« aufrufen und dieses Programm die Eingabedaten nicht vom Standardeingabegerät, sondern aus der Datei »test.txt« erhält. Die Ausgabe wird zum Standardausgabegerät (hier der Bildschirm) geschickt. Durch die Umleitung der Ein- und Ausgabe werden die Standardgeräte somit für die Dauer der Ausführung des Befehls eventuell geändert. Das Ergebnis sollte eine geordnete Liste der Namen sein.

Waalkes, Otto	Auf der Wiese	3330 Ottihausen
Meier, Karl	Nerzweg 5	4440 Dingsda
Einstein, Albert	Energiestr. 1	1234 Energiestadt
Murphy, Karl	Klar 2	4567 Gesetzhausen
Tell, Wilhelm	Apfelstr. 3	8796 Kernhausen
Müller, Otto	Birnenweg 2	4440 Dingsbums
Meyer, Heinrich	Starweg 3	3330 Egal
Zander, Frank	Miesmacher 2	4440 Dingsbums

Bild 1. Beispieldaten für die Datei »test.txt«, die natürlich beliebig zu erweitern ist

Nun wäre es aber schön, diese Liste in eine neue Datei zu schreiben. Dazu ändern Sie einfach noch die Standardausgabe:

```
sort < test.txt > sort.txt
```

Mit diesem Befehl werden die Daten aus der Datei »test.txt« als Eingabe für das Sortierprogramm verwendet. Die Ausgabe wird in die Datei »sort.txt« geschrieben. Das Beispiel zeigt, daß Sie sich nicht für jede Anwendung sofort eine Sortieroutine schreiben müssen. MS-DOS liefert das Sortierprogramm bereits mit. Genauso einfach ist es, Inhaltsverzeichnisse Ihrer Disketten in eine Datei zu kopieren:

```
dir a: > dir.txt
```

Die Ausgabe des Dir-Befehls (Inhaltsverzeichnis von Laufwerk A) wird in die Datei »dir.txt« umgeleitet. Lassen Sie sich nach der Ausführung des Befehls die Datei »dir.txt« einmal mit

```
type dir.txt
```

anzeigen. Benötigen Sie alle Inhaltsverzeichnisse in einer Datei, so verwenden Sie das dritte Symbol

```
dir a: >> dir.txt
```

So wird die Ausgabe des Dir-Befehls an die Datei »dir.txt« angehängt. Sehen Sie sich auch dieses Ergebnis mit dem Befehl »type« an. Anschließend sortieren Sie diese Datei noch mit »sort«:

```
sort < dir.txt > dirs.txt
```

Wie Sie später sehen werden, läßt sich das Sortieren auch sofort miterledigen. Benötigen Sie die Daten nicht gesondert in einer Datei, so geben Sie sie einfach direkt an den Drucker weiter:

```
sort < test.txt > lpt1
dir > lpt1
```

Neben der Ein-/Ausgabumleitung bietet MS-DOS noch Befehle zum Verketten und zum Filtern:

»find« – sucht vorgegebene Zeichen in der Standardeingabedatei und gibt gefundene Daten zur Standardausgabedatei

»more« – lenkt die Ausgabe von der Standardeingabe seitenweise zur Standardausgabe

»sort« – sortiert auf- oder absteigend die Daten von der Standardeingabe

und schreibt das Ergebnis in die Standardausgabe.

Beachten Sie hierbei, daß das Betriebssystem bei der Ausführung dieser Befehle temporäre Dateien anlegt, Dateien also, die nach Beendigung des Befehls wieder gelöscht werden. Es muß also immer noch genug Platz auf der Diskette für diese vorübergehenden Dateien vorhanden sein. Die allgemeine Syntax der Befehle finden Sie in Bild 2.

Der Befehl »more« ist speziell für die Bildschirmsteuerung vorgesehen. Sind 24 Zeilen ausgegeben, so unterbricht das Programm die Ausgabe und wartet auf einen Tastendruck. Dieser muß immer auf der Tastatur erfolgen, da das Programm für diesen Tastendruck nicht die Standardeingabe benutzt. Hier einige einfache Beispiele:

```
find "Ei" test.txt
```

Sucht nach dem Auftreten der Zeichenfolge »Ei« in der Datei »test.txt«.

```
find /c "Ei" test.txt
```

Zusätzlich zu der Funktion des vorhergehenden Befehls wird die Zeilennummer angezeigt.

```
sort /+20 < test.dat
```

Sortiert die Datei »test.dat« nach den Zeichen ab der 20. Spalte, hier also nach den Straßen.

```
more < test.txt
```

Zeigt die Datei »test.txt« seitenweise auf dem Bildschirm an.

Der Befehl »more« listet das Inhaltsverzeichnis seitenweise auf den Bildschirm

Der Befehl »more < dir« funktioniert nicht, weil »dir« die Ausgabe erst selbst erzeugen muß, die »more« als Eingabe erhalten soll. Die Daten von »dir« müssen sozusagen an »more« weitergeleitet werden – das ist der nächste Schritt, der diese Befehle erst richtig interessant macht, weil sie so untereinander kombinierbar sind. Die Entwickler von MS-DOS haben natürlich auch hieran gedacht. Mit dem sogenannten Pipe-Symbol »|« werden die Daten sozusagen durch eine Pipeline an den nächsten Befehl weitergegeben. Das Pipe-Symbol erhalten Sie durch Drücken der Tastenkombination [Alt-1-2-4]. Die Ziffern 1, 2, und 4 geben Sie dabei auf dem Zehnerblock der Tastatur ein. Nach dem Loslassen der Alt-Taste erscheint das Pipe-Symbol auf dem Bildschirm. Bei AT-Tastaturen ist das Symbol auch auf die Taste »<« gelegt und wird durch Drücken der Tastenkombination [Alt Gr-<] erzeugt. Mit

diesem Operator fängt der Spaß erst richtig an. Der Befehl

```
dir | more
```

veranlaßt, daß ein Inhaltsverzeichnis seitenweise auf dem Bildschirm ausgegeben wird.

Stapeldateien und die Befehle »find« und »sort« – eine hilfreiche Kombination

Noch schöner ist es natürlich sortiert mit

```
dir | sort | more
```

Die Daten des Inhaltsverzeichnisses werden an das Sortierprogramm weitergegeben. Dieses sortiert die Daten und gibt sie an das Programm »more« weiter.

```
dir | sort > Lpt1
```

Mit dieser Befehlsfolge wird das sortierte Verzeichnis auf dem Drucker ausgegeben.

Sie können mit

```
dir | find ".txt" | sort
```

nach bestimmten Dateien suchen. Das Verzeichnis wird nach Dateien mit der Endung »txt« abgesucht und diese Dateien werden sortiert ausgegeben. Solche Befehlsfolgen eignen sich auch für eine Stapeldatei

```
dir | find "%1" | sort
```

Speichern Sie diesen Befehl unter dem Namen »s.bat«, und geben Sie folgenden Befehl ein

```
s .txt
```

Sie erhalten die gleiche Ausgabe wie vorher. Genauso einfach ist das Suchen nach allen Namen einer bestimmten Stadt

```
find "%1" test.txt | sort
```

Speichern Sie diesen Befehl als »f.bat« ab. Die Eingabe von

```
f 4440
```

zeigt die Liste aller Orte, die mit der Postleitzahl 4440 anfangen. Diese Liste wird nach Namen alphabetisch sortiert.

Solch einfache Befehle decken also bereits einige Bereiche ab, die sonst den sogenannten »Utilities« zugeordnet werden. Möchten Sie auch für eigene Programme die Ein- und Ausgabeumleitung nutzen, so ist darauf zu achten, daß die Ein- und Ausgabe-prozesse über die Standardgeräte von MS-DOS abgewickelt werden. Zur Demonstration dieses Sachverhaltes dienen die Programme zum Suchen einer Zeichenfolge (Listing 1) ähnlich dem DOS-Befehl »find« und zum Drucken einer Datei (Listing 2).

Mit nur wenig Aufwand können Sie die nutzbaren Geister von MS-DOS in Ihre Dienste stellen

Turbo Pascal benutzt für die Ein- und Ausgabe eigene Treiber, so daß die Angaben der Dateiumleitung in der Kommandozeile von dem Programm nicht verwendet werden. Sie müssen innerhalb des Programms dafür sorgen, daß Turbo Pascal die Standardgeräte benutzt. Dies geschieht mit der Prozedur »IOUmleitung« (Listing 1). Wichtig ist hierbei, daß das direkte Schreiben in den Bildschirmspeicher ausgeschaltet wird, da sonst die beste Umleitung nichts nutzt. Der ur-

Befehl	Syntax
sort	[/r] [/+n] [< [lw:]eingabedatei] [>[lw:]ausgabedatei]
Parameter	Funktion
/r	Sortierung rückwärts
/+n	Sortierung ab Spalte n
lw	Laufwerksangabe
eingabedatei	Name der zu verwendenden Ein-/Ausgabegeräte
ausgabedatei	Name der Ausgabegeräte bzw. Dateien

Befehl	Syntax
find	[/c] [/n] [/v] "zeichenfolge" [pfad]dateiname
Parameter	Funktion
/c	Anzeige der Anzahl der Übereinstimmungen
/n	Ausgabe der Fundstellen mit Zeilennummern
/v	Anzeige der Zeichen, die die Zeichenfolge nicht enthalten
dateiname	eine oder mehrere Dateien

Bild 2. Die allgemeine Syntax der Befehle »sort« und »find«

sprüngliche Zustand wird durch die Prozedur »IODirekt« wiederhergestellt.

An diesem Programm erkennen Sie, daß mit nur wenig Aufwand solche »dienstbaren Geister« zur Verfügung stehen. Der Aufruf erfolgt wie gehabt, nur wird die Zeichenkette nicht in Anführungszeichen gesetzt:

```
finde Einstein < test.txt
finde Ein < test.txt > gef.txt
```

Das zweite Beispiel ist schon etwas komplexer. Das Druckprogramm selbst ist recht einfach gehalten. Es benötigt lediglich den Dateinamen der zu druckenden Datei, die Anzahl der Zeichen pro Zeile und die Anzahl der Zeilen pro Seite. Der Clou dieses Programms besteht in der Eingabe dieser Größen.

Die Umleitung von Daten läßt sich auch in eigenen Programmen sinnvoll einsetzen

Nehmen Sie einmal an, Sie schreiben ein Listing für die Zeitschrift DOS International, die den Ausdruck auf 43 Zeichen pro Zeile und 65 Zeilen begrenzt haben möchte. Sie selbst möchten jedoch die Listings in gewohnter Weise mit 75 Zeichen pro Zeile und 60 Zeilen pro Seite drucken. Mit Hilfe der Ein- und Ausgabeumleitung ist dies schnell erledigt – und beliebig wiederholbar. Sie erzeugen zwei Dateien mit folgendem Inhalt:

```
drucke.pas
43
65
```

Schließen Sie wieder mit [Ctrl-Z] und [Enter] ab und benennen die Datei »dos.dat«. Die zweite Datei heißt »eigen.dat« und hat den Inhalt

```
drucke.pas
75
60
```

Auch hier schließt wieder ein [Ctrl-Z] und ein [Enter] die Datei. Jetzt erfolgt der Aufruf des Programms mit

```
drucke < dos.dat
drucke < eigen.dat
```

Das Listing sehen Sie nun aber nur auf dem Bildschirm, also geben Sie noch einmal ein:

```
drucke < dos.dat > lpt1
```

Die Ausgabe wird so zum Drucker gesandt.

Nun hätte die Redaktion Ihr Listing auch noch gerne auf Diskette – dank der Umleitung kein Problem:

```
drucke < dos.dat > druckdos.txt
```

Das Druckerprogramm zeigt aber noch mehr. Rufen Sie es ohne Eingabeumleitung auf, so müssen Sie die Parameter (die Anzahl der Zeilen und Zeichen pro Zeile) von Hand eingeben.

Ohne Ausgabeumleitung erfolgt die Listing-Ausgabe auf dem Bildschirm (wie bei MS-DOS-Programmen). Alle Abfragen, die für die Bedienung zuständig sind, werden zum Bildschirm gesandt. Das hat den Vorteil, daß beispielsweise wichtige Fehlermeldungen nicht über den Drucker oder ein anderes, als Standardausgabe definiertes Gerät, sondern auf jeden Fall auf dem Bildschirm ausgegeben wer-

den. Dies wird in der Prozedur »Parameter« erreicht, indem die Ausgabeumleitung nur auf bestimmte, auszugebende Daten angewendet wird. Das Druckerprogramm ist also in der Lage, Ausgaben nicht nur zur Standardausgabe, sondern auch direkt zum Bildschirm zu senden.

Eine definiert ein- und ausschaltbare Umleitung von Ein- und Ausgabedaten schafft Vorteile

Entsprechendes gilt selbstverständlich auch für die Eingabe. Denkbar ist zum Beispiel, daß das Programm Dateien löschen soll. Da dieser Vorgang aber endgültig ist, wünschen Sie vorher eine Sicherheitsabfrage. Erfolgt die Abfrage über die Standardeingabe, so müßten Sie entsprechend obiger Beschreibung eine Datei erzeugen, aus der die Befehle eingelesen werden. Da dieser Vorgang aber sehr wichtig ist, nutzen Sie die Eingabe von Pascal (nicht umgeleitet), so daß auf jeden Fall die Ausführung über die Tastatur bestätigt werden muß. Probieren Sie die genannten Befehle und deren Kommandozeilenparameter ruhig ausgiebig aus, denn dadurch werden Sie am schnellsten mit ihnen vertraut. Die Beispieldatei »test.txt« ist etwas knapp ausgefallen, doch dürfte es für Sie kein Problem darstellen, die Datei dank der Ein-/Ausgabeumleitung einfach zu erweitern.

(Helmut Amshove/zi)

Name	finde.pas
Sprache	Turbo Pascal 5.0
Grafikkarte	jede
Besonderheiten	keine

```

[P84] 1: (* ----- *)
[YA5] 2: (* Programmname: FINDE:PAS *)
[L05] 3: (* (C) 1989 Everts & Hagedorn *)
[705] 4: (* DOS International *)
[P84] 5: (* ----- *)
[L86] 6: (* Demonstrationsprogramm zur Ein- *)
[7X5] 7: (* Ausgabe-Umleitung unter MS-DOS *)
[TS4] 8: (* mit Turbo Pascal 5.0 *)
[P84] 9: (* ----- *)
[KF2] 10: program finde;
[5H1] 11: uses crt;
[000] 12:
[544] 13: var Zeile, Suche :string;
[000] 14:
[P84] 15: (* ----- *)
[176] 16: (* Ein- und Ausgabe auf Standard-DOS *)
[P84] 17: (* ----- *)
[323] 18: procedure IOUmleitung;
[MU0] 19: begin
[C85] 20: (* nicht direkt in den Bildschirm- *)
[2L4] 21: (* speicher schreiben *)
[XE3] 22: DirectVideo:=false;
[DN2] 23: Assign(Input, '');

```

```

[492] 24: Reset(Input);
[4X2] 25: Assign(Output, '');
[DX2] 26: Rewrite(Output);
[ZN0] 27: end;
[000] 28:
[P84] 29: (* ----- *)
[KU5] 30: (* Ein- und Ausgabe auf Unit CRT *)
[P84] 31: (* ----- *)
[WE3] 32: procedure IODirekt;
[MU0] 33: begin
[603] 34: AssignCrt(Input);
[492] 35: Reset(Input);
[U63] 36: AssignCrt(Output);
[DX2] 37: Rewrite(Output);
[ZN0] 38: end;
[000] 39:
[P84] 40: (* ----- *)
[XK4] 41: (* Hauptprogramm *)
[P84] 42: (* ----- *)
[000] 43:
[MU0] 44: begin
[DB1] 45: Clrscr;
[XM6] 46: writeln('Dieses Programm durchsucht ' +
[015] 47: 'die Standardeingabe nach der');
[XC5] 48: writeln('in der Kommandozeile ' +
[DL4] 49: 'angegebenen Zeichenkette');
[L93] 50: if paramcount=0 then
[AP4] 51: writeln('Falsche Parameter')
[WP1] 52: else begin
[T83] 53: Suche:=Paramstr(1);
[A72] 54: IOUmleitung;
[174] 55: write('Gesuchtes Wort: ');
[MP2] 56: writeln(suche);

```



```

[8X3] 57: while not eof(input) do
[MU0] 58: begin
[M82] 59:   readln(zeile);
[R84] 60:   if pos(suche,zeile)<>0 then
[EP2] 61:     writeln(zeile);
[ZN0] 62:   end;
[1K1] 63:   IODirekt;
[ZN0] 64: end;
[KN0] 65: end.

```

Listing 1. Ein Suchprogramm in Pascal, das die Ein- und Ausgabeumleitung unterstützt

Name	drucke.pas
Sprache	Turbo Pascal 5.0
Grafikkarte	jede
Besonderheiten	keine

```

[P84] 1: (* ----- *)
[HK5] 2: (* Programmname: DRUCKE.PAS *)
[L05] 3: (* (C) 1989 Everts & Hagedorn *)
[705] 4: (* DOS International *)
[P84] 5: (* ----- *)
[L86] 6: (* Demonstrationsprogramm zur Ein- *)
[7X5] 7: (* Ausgabe-Umleitung unter MS-DOS *)
[TS4] 8: (* mit Turbo Pascal 5.0 *)
[P84] 9: (* ----- *)
[UT2] 10: program drucke;
[5H1] 11: uses crt;
[000] 12:
[PA3] 13: var zeile :string;
[YM2] 14: zeilenNr : byte;
[M63] 15: dname :string[64];
[W62] 16: ein : text;
[E03] 17: AnzZeichen : byte;
[SU2] 18: AnzZeilen : byte;
[000] 19:
[P84] 20: (* ----- *)
[176] 21: (* Ein- und Ausgabe auf Standard-DOS *)
[P84] 22: (* ----- *)
[323] 23: procedure IOUmleitung;
[MU0] 24: begin
[C85] 25: (* nicht direkt in den Bildschirm- *)
[2L4] 26: (* Speicher schreiben *)
[XE3] 27: DirectVideo:=false;
[DN2] 28: Assign(Input, '');
[492] 29: Reset(Input);
[4X2] 30: Assign(Output, '');
[DX2] 31: Rewrite(Output);
[ZN0] 32: end;
[000] 33:
[P84] 34: (* ----- *)
[KU5] 35: (* Ein- und Ausgabe auf Unit CRT *)
[P84] 36: (* ----- *)
[WE3] 37: procedure IODirekt;
[MU0] 38: begin
[603] 39: AssignCrt(Input);
[492] 40: Reset(Input);
[U63] 41: AssignCrt(Output);
[DX2] 42: Rewrite(Output);
[ZN0] 43: end;
[000] 44:

```

```

[P84] 45: (* ----- *)
[4T4] 46: (* Parametereingabe *)
[P84] 47: (* ----- *)
[PK3] 48: procedure Parameter;
[MU0] 49: begin
[DN2] 50: Assign(input, '');
[492] 51: reset(input);
[063] 52: write('Dateiname: ');
[W72] 53: readln(dname);
[GE5] 54: writeln('Gedruckt wird: ',dname);
[8U5] 55: write('Anzahl Zeichen pro Zeile: ');
[S93] 56: readln(AnzZeichen);
[4H3] 57: writeln(AnzZeichen);
[VY5] 58: writeln('Anzahl Zeilen pro Seite: ');
[G23] 59: readln(AnzZeilen);
[NF3] 60: writeln(AnzZeilen);
[603] 61: AssignCrt(Input);
[492] 62: reset(input);
[ZN0] 63: end;
[000] 64:
[P84] 65: (* ----- *)
[XK4] 66: (* Hauptprogramm *)
[P84] 67: (* ----- *)
[000] 68:
[MU0] 69: begin
[DB1] 70: Clrscr;
[YW5] 71: writeln('Dieses Programm druckt '+'
[8T3] 72: 'die angegebene Datei');
[ZT1] 73: Parameter;
[SX2] 74: assign(ein,dname);
[KT0] 75: (*$I- *)
[4W1] 76: reset(ein);
[ET0] 77: (*$I+ *)
[DI4] 78: if IOresult<>0 then begin
[CD5] 79:   writeln('Datei nicht vorhanden');
[DU0] 80:   halt;
[ZN0] 81: end;
[A72] 82: IOUmleitung;
[R02] 83: zeilenNr:=0;
[C53] 84: while not eof(ein) do
[MU0] 85: begin
[MH4] 86:   if zeilenNr>AnzZeilen then
[MU0] 87:   begin
[8C2] 88:     writeln(#12#13);
[R02] 89:     zeilenNr:=0;
[ZN0] 90:   end;
[5W2] 91:   readln(ein,zeile);
[GE5] 92:   if length(zeile)>AnzZeichen then
[MU0] 93:   begin
[HK3] 94:     writeln(copy(zeile,1,
[Y42] 95:       AnzZeichen));
[8E3] 96:     writeln(copy(zeile,
[3R4] 97:       AnzZeichen+1,length(zeile)));
[6M3] 98:     zeilenNr:=zeilenNr+1;
[4M0] 99:   end;
[BE3] 100:   else writeln(zeile);
[6M3] 101:   zeilenNr:=zeilenNr+1;
[U63] 102:   AssignCrt(Output);
[DX2] 103:   Rewrite(Output);
[UP4] 104:   writeln('Zeile: ',zeilenNr);
[4X2] 105:   Assign(Output, '');
[DX2] 106:   Rewrite(Output);
[ZN0] 107: end;
[8C2] 108: writeln(#12#13);
[1K1] 109: IODirekt;
[DZ3] 110: writeln('Das war''s ...');
[KN0] 111: end.

```

Listing 2. Das Druckprogramm in Pascal, das die Ein- und Ausgabeumleitung zur Parametersteuerung benutzt

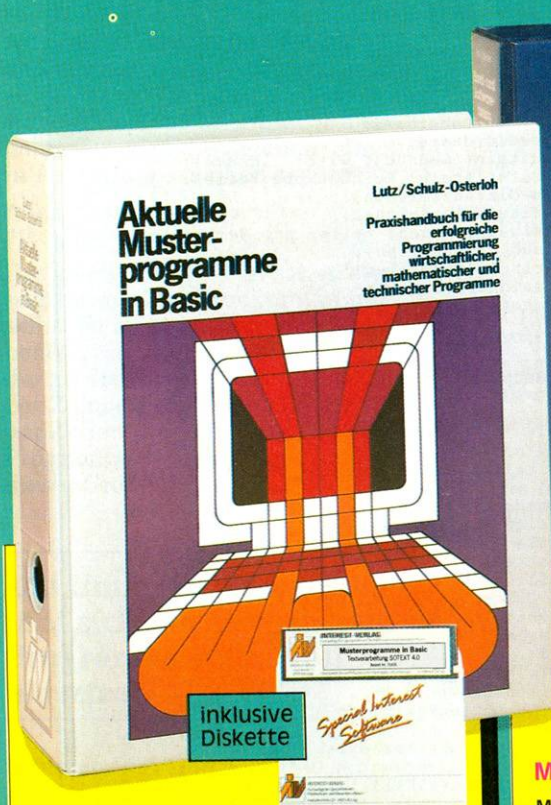
Utilities gesucht

Programme, die den Bildschirminhalt auf den Kopf stellen, suchen unsere Leser sicherlich nicht. Aber einen wissenschaftlichen Taschenrechner, ein Festplatten-Backup-Programm, einen Extended-Memory-Treiber, Druckerspooles oder Cache-Speicher. Zögern Sie deshalb nicht, schicken Sie uns unverbindlich die aktuelle Version Ihres Programms und einen Ausdruck der Dokumentation. Falls Sie sich vorweg informieren möchten: Zwischen 15.00 und 17.00 Uhr stehen wir Ihnen an allen Werktagen unter der Rufnummer (089) 4 61 00 10 zur Verfügung.

Schicken Sie Ihre Unterlagen an folgende Adresse:
 Redaktionsbüro Everts & Hagedorn
 »DOS Software«
 Hans-Stießberger-Str. 3
 8013 Haar/München

„ALLES IN ALLEM EIN MUSS“

(Buchbesprechung c't 2/89)



Basic-Tools, Utilities und Anwenderprogramme für Wirtschaft, Technik, Mathematik

Sie erhalten sofort einsetzbare Programme und Utilities für PC-/MS-DOS-Computer.

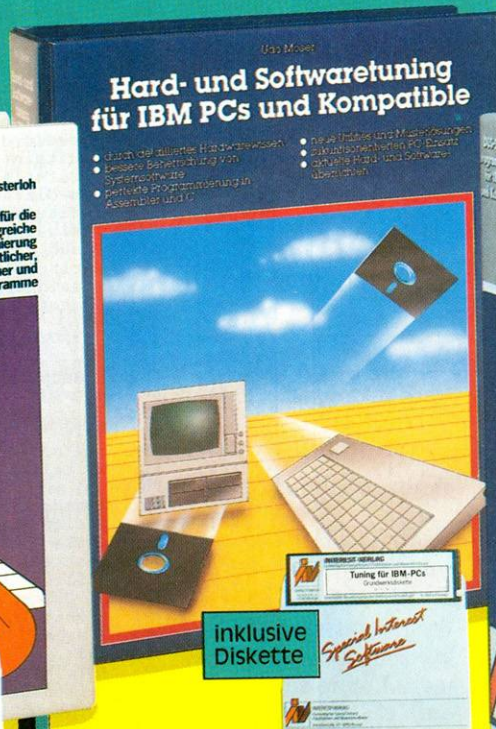
Auszüge aus dem Inhalt:

- **Das leistungsfähige Textverarbeitungsprogramm SOTEXT 4.0**, komplett menügesteuert und mit deutschen Befehlen, zusätzlich im sofort lauffähigen Objekt- und im Maschinencode auf Diskette;
- **Praxiserprobte Musterlösungen** aus den Bereichen Wirtschaft/Verwaltung (Finanzbuchhaltung, Netzplantechnik, Rabattermittlung...), Mathematik (u. a. Analyse von Funktionsgleichungen; lineare Algebra: z. B. Matrizenoperationen, Simplexalgorithmus), Statistik (Berechnung von Mittelwert, Standardabweichung...), Physik und Technik (Berechnen von Querschnittswerten, Wertetabellen von Digitalschaltungen...), Ausbildung (u. a. Vokabeltraining);
- **Vielseitige Tools und Routinen** u. a. für: Dateiverwaltung/-organisation (Dateigenerator, Komprimieren/Expandieren/Konvertieren von Dateien...), Grafik (Grafik-, Zeichengenerator u. a.), Bildschirm, Drucker...
- **systematischer Basic-Programmierkurs**

Fordern Sie noch heute an:

Aktuelle Musterprogramme in Basic
extrabreiter Ringbuchordner, Format DIN A4,
ca. 1100 Seiten, inkl. Diskette,
Bestell-Nr. 1300/71300, Preis: DM 98,-.

Alle 2-3 Monate erhalten Sie Ergänzungsausgaben zum Grundwerk mit je ca. 130 Seiten zum Seitenpreis von 38 Pfennig (Abbestellung jederzeit möglich).



Mehr Erfolg durch PC-Tuning

Mit diesem Nachschlagewerk

- rüsten Sie Ihren PC für zusätzliche Aufgaben
- programmieren Sie hardwarenah
- verfügen Sie über sofort einsetzbare Musterprogramme
- erschließen Sie sich neue Einsatzbereiche

Auszüge aus dem Inhalt:

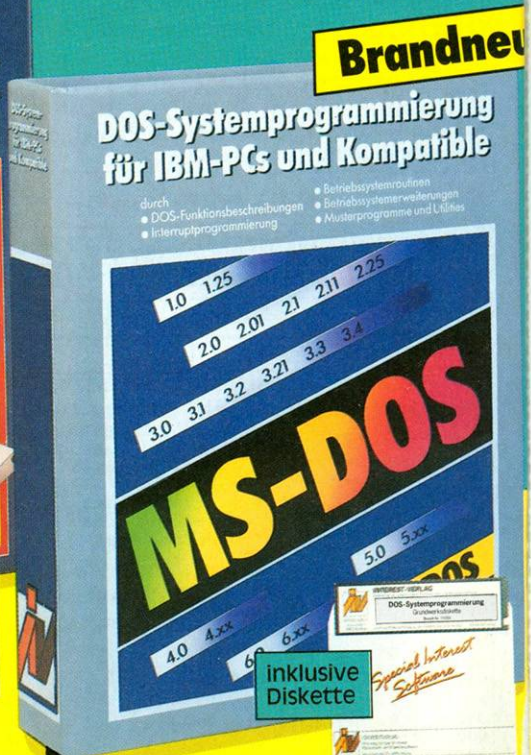
- **Detaillierte Dokumentationen** der IBM PCs, XTs, ATs, PS/2 Systeme und 80386-Systeme
- **80(x)xx-CPUs**: Architekturen, Beschaltungen, interne Abläufe bei Adressierung und Befehlsausführung...
- **Betriebssysteme**: MS-/PC-DOS, OS-2 und Unix
- **Programmiersprachen**: u. a. kompletter SIGMA-Assembler, Turbo-Pascal- und C-Kurs
- **Softwareengineering**: strukturiertes Programmieren mit bewährten Methoden wie z. B. SADT und HIPO-Verfahren
- **Tips, Tricks und Utilities**: Zweifach-Tastaturreiber, Mausanschluß für Turbo-Pascal, Profiler, Arbeiten mit Batchdateien, Turbo-Pascal-Schnittstelle zu MS-DOS/BIOS, V.24-Treiber
- **Komplette Musterlösungen**: z. B. Netzwerkanalyse
- **PC-Erweiterungen für neue Applikationen**: universelle I/O-Karte, Einbau einer Festplatte
- **Know-how zu speziellen Bereichen**: Netzwerke, DFU, CAD

Fordern Sie noch heute an:

Hard- und Softwaretuning für IBM PCs und Kompatible

Stabiler Ringbuchordner, Format DIN A4,
ca. 950 Seiten, inkl. Diskette
Bestell-Nr. 2600/72600, Preis: DM 98,-.

Alle 2-3 Monate erscheinen Erweiterungsausgaben zum Grundwerk mit je ca. 130 Seiten zum Seitenpreis von 38 Pfennig (Abbestellung jederzeit möglich).



Mehr Erfolg mit DOS durch effizienten Einsatz der Systemroutinen

Dieses neue Nachschlagewerk

- erläutert Ihnen sämtliche DOS-Funktionen; auch die sonst nicht dokumentierten für sicheres und gezieltes Programmieren
- reduziert Ihren Entwicklungsaufwand durch effizienten Einsatz der Betriebssystemroutinen;
- zeigt Ihnen, wie Sie durch Interruptprogrammierung die MS-/PC-DOS-Funktionen verändern oder erweitern;

Auszüge aus dem Inhalt:

- **Interruptprogrammierung**: detaillierte Dokumentationen zu BIOS-Interrupts und DOS-Funktionen und deren gezielte Anwendungen.
- **Das DOS-System im Detail**: genaue Beschreibungen der verschiedenen DOS-Versionen einschließlich DOS 4.0; ausführliche Erläuterungen zu den Funktionen der Systemkomponenten
- **DOS-Kommandos**: alphabetisch gegliederte Befehlsbeschreibung mit praktischen Anwendungsbeispielen
- **Tools**: detaillierte Erläuterungen der DOS-Werkzeuge; mit kompletten Tools
- **Multitasking unter DOS**: problemloses Echtzeit-Multitasking mit MULTIDOS
- **Musterlösungen und Utilities**: u. a. zur Ein-/Ausgabe, Dateiverwaltung, Bausteinprogrammierung, Daten- und Parameterübergabe

Fordern Sie noch heute an:

DOS-Systemprogrammierung für IBM-PCs und Kompatible

Stabiler Ringbuchordner, Format DIN A4,
ca. 400 Seiten, inkl. Diskette
Bestell-Nr. 5000/75000, Preis: DM 98,-.

Alle 2-3 Monate erscheinen Erweiterungsausgaben zum Grundwerk mit je ca. 130 Seiten zum Seitenpreis von 38 Pfennig (Abbestellung jederzeit möglich).



Industriest. 21
D-8901 Kissing
Tel. 08233/2101-0

INTEREST-VERLAG
Fachverlag für
Special Interest
Publikationen
und Anwendersoftware

Für Ihre Anforderung verwenden Sie bitte nebenstehende Bestellka

Was Ihnen MS-DOS sagen will

Da steht es nun, Ihr neues Prachtstück: technisch nüchtern und der Dinge harrend, die Sie ihm entlocken wollen. Doch kaum haben Sie die ersten Kommandos an Ihren PC gerichtet, kontert er mit einem »syntax error« – das duldet keinen Widerspruch. Damit Sie in solchen Situationen gelassen bleiben können, sagt Ihnen dieser Beitrag, was Ihnen Ihr PC mit dieser und vielen anderen Meldungen sagen möchte.

Zwischen jedem Betriebssystem und dem Systemnutzer (also Ihnen) gibt es eine Schnittstelle – den Kommando-Interpreter oder Kommandoprozessor. Der Kommando-Interpreter ist ein Programm, das die Eingaben des Anwenders in erster Linie syntaktisch, das heißt nach bestimmten grammatikalischen Regeln, überprüft und dann die entsprechenden Aktionen des Betriebssystems einleitet. Beim Betriebssystem MS-DOS heißt dieser Vermittler »command.com«. Er liegt in Form einer eigenständigen Datei vor und wird direkt im Anschluß an das Betriebssystem in den Arbeitsspeicher Ihres Computers geladen.

»command.com« prüft jede Eingabe daraufhin, ob es sich um ein internes Kommando (beispielsweise »dir«) handelt, das er direkt ausführt, oder ob ein transienter, das heißt externer Befehl (zum Beispiel »format«) vorliegt. In diesem Fall leitet der Kommandoprozessor den Ladevorgang des entsprechenden Programms – von einer Diskette oder der Festplatte – und dessen Start ein.

Es ist ein Faktum, daß Menschen keine Maschinen sind und daher Fehler machen. Der Kommando-Interpreter »command.com« muß deshalb auch auf die Eingabe fehlerhafter Kommandos reagieren. Er tut dies mit einer Anzahl unterschiedlicher Fehlermeldungen. Die wichtigsten dieser Meldungen und ihre Bedeutung finden Sie in der folgenden Aufstellung.

»Codeseite # nicht für alle Geräte vorbereitet!« (»Code page # not prepared for all devices«): Die unter der Nummer # angegebene Codeseite (länderspezifische Tabelle ab MS-DOS 3.3) ist nicht für alle von MS-DOS unterstützten Geräteeinheiten zugelassen.

»Codeseite # nicht für das System vorbereitet!« (»Code page # not prepared for system«): Die angegebene

Codeseite mit der Nummer # ist nicht zugelassen.

»COMMAND.COM kann nicht geladen werden, System beendet!«

(»Cannot load COMMAND, system halted«): Das Betriebssystem kann den Kommando-Interpreter »command.com« nicht laden und hält deshalb die weitere Bearbeitung an.

»COMMAND.COM kann nicht gestartet werden, Abbruch!« (»Cannot start COMMAND, exiting«): Ein Start des Programms »command.com« ist nicht ausführbar. MS-DOS verzweigt daher in die nächsthöhere Kommandoebene.

»COMMAND.COM ungültig!« (»Invalid COMMAND.COM«): Der Kommando-Interpreter »command.com« ist fehlerhaft.

»Datei kann nicht in sich selbst kopiert werden!« (»File cannot be copied onto itself«): Beim Kopieren einer Datei mit dem Copy-Befehl wurde für die Quell- und Zieldatei das gleiche Laufwerk (und Unterverzeichnis) angegeben.

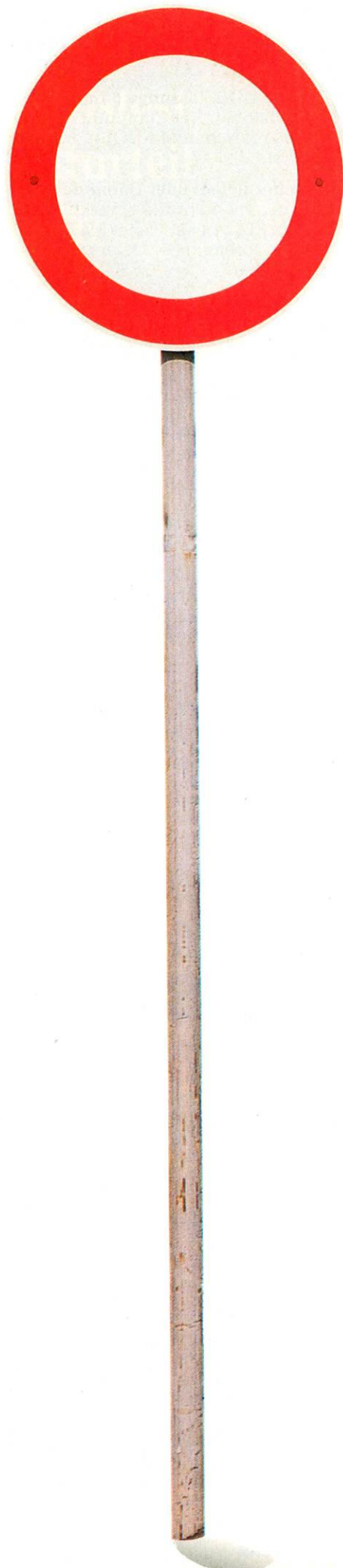
»Datei nicht gefunden!« (»File not found«): Die angegebene Datei ist nicht vorhanden.

»Dateiname bereits vergeben oder nicht vorhanden!« (»Duplicate file name or file not found«): Das Betriebssystem kann die Datei nicht finden, oder es handelt sich um einen schon vorhandenen Dateinamen.

»Dateiname fehlerhaft oder nicht vorhanden!« (»Invalid file name or file not found«): Entweder wurde der Dateiname fehlerhaft eingegeben, oder die Datei konnte nicht gefunden werden.

»Datenfehler« (»Data error«): Bei der Bearbeitung von Daten ist ein Fehler aufgetreten.

»Die Anzahl der Parameter stimmt nicht!« (»Too many Parameters«): Die Parameterangaben für ein Programm



oder einen MS-DOS-Befehl sind fehlerhaft.

»Falsche DOS-Fassung« («Incorrect DOS version»): Der Computer wurde mit einer falschen MS-DOS-Version gestartet.

»Falscher Befehl oder Dateiname!« («Bad command or file name»): Die aktuelle Eingabe ist weder als Befehl noch als vorhandener Dateiname interpretierbar.

»FCB nicht vorhanden« («FCB unavailable»): Der Dateikontrollblock »File Control Block« ist gegenwärtig nicht frei.

»Fehler bei Schreibvorgang im Gerät!« («Error writing to device»): Der Schreibversuch auf eine Geräteeinheit verlief nicht fehlerfrei.

»Fehler beim Anlegen einer Datei!« («File creation error»): Dem Betriebssystem ist es nicht möglich, eine neue Datei anzulegen.

»Fehler in EXE-Datei!« («Error in EXE file»): In der zur Zeit aktiven Programmdatei mit der Kennung »exe« wurde ein Fehler festgestellt.

»Fehler in vorläufiger Datei bei Befehlsverkettung!« («Intermediate file error during pipe»): Die Zwischenspeicherdatei innerhalb eines Pipe-Kommandos (beispielsweise »dir | sort«) ist fehlerhaft.

»Fehlerhafte Dateizuordnungstabelle!« («File allocation table bad»): Die Dateibelegungstabelle (»File Allocation Table« oder »FAT«) für das gewählte Laufwerk ist nicht in Ordnung.

»FOR kann nicht verschachtelt werden!« («FOR cannot be nested»): Es ist nicht zulässig, den For-Befehl (innerhalb von Stapeldateien) zu verschachteln.

»Gesamtfehler!« («General failure»): Es ist ein genereller Fehler aufgetreten.

»Inhalt der Zieldatei beim Kopiervorgang verloren!« («Content of destination lost before copy»): Der ursprüngliche Inhalt der Zieldatei ging vor dem Kopiervorgang verloren.

»Kein Papier« («Printer out of paper error»): Im angeschlossenen Drucker ist kein Papier vorhanden.

»Kein Suchweg!« («No path»): Im Umgebungsbereich von MS-DOS (»Environment area«) ist kein Suchpfad eingetragen.

»Keine freie Dateinummer« («No free file handles»): Gegenwärtig sind keine freien »Handles« (Schlüssel für den Zugriff auf Dateien und Geräteeinheiten) verfügbar.

»Nicht-Bereit-Fehler!« («Not ready error»): Ein Disketten- oder Festplattenlaufwerk ist für die gewünschte Operation nicht betriebsbereit.

»ON oder OFF muß angegeben werden!« («Must specify ON or OFF»): Die Parametereingabe für die Befehle »break«, »echo« oder »verify« war falsch.

»Programm paßt nicht in den Arbeitsspeicher!« («Program too big to fit in memory»): Das zu ladende Programm ist für den Arbeitsspeicher zu groß.

»Prozeß abgebrochen, kann nicht fortgesetzt werden!« («Top level process aborted, cannot continue»): Das Programm auf der obersten Kommandoebene wurde abgebrochen. Das Betriebssystem stoppt die weitere Bearbeitung.

»Schreibschutzfehler« («Write protect error»): Die Diskette beziehungsweise Festplatte ist schreibgeschützt.

»Sektor nicht gefunden« («Sector not found»): Ein Sektor auf der Diskette oder Festplatte wurde nicht gefunden. Das Speichermedium ist vermutlich fehlerhaft.

»Speicherkapazität im Umgebungsbereich erschöpft!« («Out of environment space»): Für weitere Eintragungen in den MS-DOS-Umgebungsbereich (»Environment area«), zum Beispiel mit »set«, ist kein Platz vorhanden.

»Speicherzuordnungsfehler!« («Memory allocation error»): Die Belegung eines Speicherblocks ist fehlerhaft verlaufen.

»Sprungmarke nicht gefunden!« («Label not found»): Das Sprungziel (»Label«) innerhalb einer Stapeldatei wurde nicht gefunden.

»Stapelverarbeitungsdatei fehlt!« («Batch file missing»): Bei der Stapelverarbeitung erfolgte der Zugriff auf eine Datei mit der Kennung »bat«, die jedoch nicht vorhanden ist.

»Suchfehler« («Seek error»): Ein bestimmter Sektor der Diskette oder Festplatte wurde nicht gefunden. Das Speichermedium ist wahrscheinlich fehlerhaft.

»Suchweg für COMMAND.COM ist fehlerhaft!« («Specified COMMAND search directory bad»): Der Suchpfad für den Kommandoprozessor »command.com« ist nicht korrekt.

»Suchweg nicht gefunden!« («Path not found»): Der entsprechende Suchpfad ist nicht vorhanden.

»Syntaxfehler!« («Syntax error»): Der Kommando- beziehungsweise Pro-

grammaufruf in der aktuellen Eingabezeile weist einen syntaktischen Fehler auf.

»Ungültige Datumsangabe!« («Invalid date»): Das Kommando »date« enthielt eine ungültige Datumseingabe.

»Ungültige Gerätenummer« («Invalid handle»): Der verwendete Schlüssel für den Zugriff auf Geräteeinheiten (»Handle«) ist inkorrekt.

»Ungültige Laufwerksangabe im Suchweg!« («Invalid drive in search path»): In einem Suchpfad ist eine unzulässige (nicht verfügbare) Laufwerksangabe enthalten.

»Ungültige Laufwerksangabe!« («Invalid drive specification»): Die angegebene Laufwerkskennung ist fehlerhaft.

»Ungültige Parameter!« («Invalid parameter»): Im angegebenen Kommando wurde (mindestens) ein fehlerhafter Parameter verwendet.

»Ungültige Zeitangabe!« («Invalid time»): Das Kommando »time« enthielt eine ungültige Zeitangabe.

»Ungültiger Suchweg oder Dateiname!« («Invalid path or file name»): Der Suchpfad oder Dateiname eines Programms wurde nicht richtig angegeben.

»Ungültiger Suchweg, kein Verzeichnis oder Verzeichnis nicht leer!« («Invalid path, not directory, or directory not empty»): Entweder liegt ein falscher Suchpfad oder ein fehlerhafter Verzeichniseintrag vor, oder das Verzeichnis ist (beispielsweise für eine Löschoption) nicht leer.

»Ungültiges Gerät!« («Invalid device»): Die verwendete Gerätekennung ist falsch.

»Ungültiges Verzeichnis!« («Invalid path»): Der vorgegebene Suchpfad ist ungültig.

»Unzulässiger Diskettenwechsel« («Invalid disk change»): Es wurde ein unerlaubter Diskettenwechsel durchgeführt.

»Verzeichnis kann nicht angelegt werden!« («Cannot make directory entry»): Das gewünschte Verzeichnis läßt sich nicht einrichten.

»Zu wenig Platz auf der Diskette/Platte!« («Insufficient disk space»): Für die Speicherung weiterer Dateien reicht der Disketten- beziehungsweise Festplattenplatz nicht aus.

»Zugriff verweigert!« («Access denied»): Der Zugriff auf eine Datei wurde verweigert (weil es sich beispielsweise um eine schreibgeschützte (»read only«) Datei handelt).

(zi)

Testen mit System

Checkliste – Labortest – Praxistest
Information – Fazit – Gesamturteil



**Jetzt bei Ihrem
Zeitschriftenhändler**

Drei überzeugende Pluspunkte der DOS-TEST-Konzeption:

Vergleichbare Testberichte mit einheitlichen Kriterien helfen Ihnen bei der Kaufentscheidung.

Knallharte Information über interessante Themen, verständlich und locker mit viel Lesespaß.

Durch Tests im praktischen Einsatz wissen Sie schon heute, welche Hard- und Software morgen aktuell ist.

Die wichtigsten MS-DOS-Befehle

Das Betriebssystem MS-DOS umfaßt etwa 48 kleine Programme, die sogenannten »externen Befehle«. Welche dieser Befehle Sie wirklich benötigen, und welche Sie getrost von Ihrer Festplatte oder Systemdiskette verbannen können, erklärt Ihnen der folgende Artikel.

Obwohl MS-DOS ein relativ kleines und nur durchschnittlich leistungsfähiges Betriebssystem ist, bringt es eine ganze Menge von Befehlen mit. Nur die wichtigsten sind im Kommandozeilen-Interpreter »command.com« integriert, zum Beispiel »copy«, »dir«, »del«, »cd«, »md«, »rd« und einige andere mehr. Etwa 48 weitere Befehle befinden sich jedoch in Form kleiner Programme auf den MS-DOS-Disketten (je nach Rechner-typ hat der Hersteller vielleicht noch den einen oder anderen zusätzlichen Befehl hinzugefügt).

Viele dieser Befehle benutzt der durchschnittliche Anwender sehr selten oder sogar nie, weil sie sehr spezielle Funktionen erfüllen. Daher folgt nun die besonders für Anwender ohne Festplatte sehr wichtige Information, welche Programme sie ohne Gewissensbisse löschen können.

Einen Überblick gibt die Tabelle. Die wichtigste Datei ist »command.com«, ohne die Ihr MS-DOS keine Befehle entgegennehmen und daher nicht bedient werden kann.

Ständig gebraucht werden die folgenden sechs Kommandos: »diskcopy.com« zum schnellen Duplizieren von ganzen Disketten. Dieses Kommando funktioniert entgegen manchen Vorurteilen auch mit einem Diskettenlaufwerk (»diskcopy a: a:«). Ohne »format.com« können Sie neue Disketten nicht verwenden, es sei denn, sie haben sogenannte »vorformatierte Disketten« gekauft. »keyb.com« funktioniert nur zusammen mit »keyboard.sys«. Diese beiden Dateien bilden den Tastaturtreiber, der Ihrem PC die Umlaute beibringt. Einen Überblick über den Disketten-/Plattenzustand gewinnen Sie mit »chkdsk.com«. Es zeigt die Größe und Belegung der Platte an, informiert über eventuell beschädigte Bereiche und findet »verlorengegangene« Dateien, die nicht ordnungsgemäß gesichert wurden (»chkdsk /f«). Oft wesentlich praktischer als »copy« ist der Befehl »xcopy.exe«. Er kopiert unter Ausnutzung allen Speichers sehr effizient und kann mit dem Zusatz »/s« sogar Unterverzeichnisse mitkopieren.

Nun zu den sehr nützlichen Befehlen. Sie sollten sich auf jeden Fall auf der

Festplatte befinden, die MS-DOS-Boot-Diskette muß sie nicht enthalten – zumindest nicht alle davon. So erlaubt »ansi.sys« vielen Programmen eine formatierte, farbige Bildschirmdarstellung. »mode.com« dient der Festlegung gewisser Parameter des PC. Haben Sie eine Hercules-Karte und eine CGA-Karte im Gerät, kann »mode.com« umschalten. Die Befehle »backup.com« und »restore.com« dienen zur Datensicherung für die Festplatte und werden an anderer Stelle ausführlich erläutert. »more.com« ist da schon wieder nützlicher, es dient zur seitenweisen Anzeige von Dateien. Verwenden kann man es auf zwei Arten: »more < liesmich.txt« zur Anzeige von Dateien auf der Platte oder »tree | more« zur Anzeige von Programmausgaben.

Für Besitzer langsamer Drucker ist »print.com« von Interesse. Es »füttert« den Drucker mit Daten, während man selbst weiterarbeiten kann. »recover.com« versucht sich an der Wiederherstellung von beschädigten Dateien. Hat man also einen Lesefehler in einer Datei (zum Beispiel »fehler.dat«), ist es sehr ratsam, sie mit »recover fehler.dat« zu behandeln. »recover« trägt die fehlerhafte Stelle auf der Diskette nämlich als »fehlerhaft« ein, so daß MS-DOS sie nicht mehr verwendet.

Von geringerem Allgemeininteresse, aber oft von großem individuellem Nutzen sind die Befehle »fastopen.exe« und »ramdrive.sys«, die zur Beschleunigung der Arbeit dienen. »append.exe«, »assign.com«, »join.exe«, »subst.exe« erlauben die Erweiterung von Suchpfaden, die Einrichtung neuer Laufwerksbuchstaben und Verzeichnisse, erfüllen also vor allem für fortgeschrittene Anwendungen wichtige Funktionen. Der »gute alte« Zeileneditor »edlin.com« zur einfachen und schnellen Änderung von Dateien ist immer noch nützlich, wenn einmal nichts anderes mehr funktioniert. Nur für Besitzer einer CGA-Karte ist dagegen »graphics.com« von Interesse. Es druckt nach der Installation auf Tastendruck eine Hardcopy des Grafikbildschirms aus, jedoch nur auf Nadeldruckern. »attrib.exe«, »label.com«, »repla-

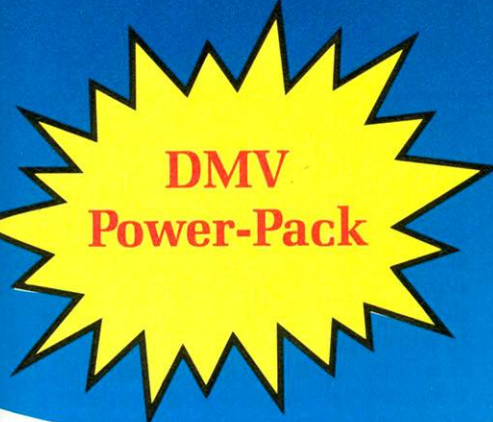
ce.exe« und »share.exe« leisten sehr spezielle Dienste. Wer sie nicht kennt, braucht sie auch nicht.

»sort.exe« dagegen kann jedem nützlich sein, es dient, wie der Name schon sagt, zum Sortieren von Textdateien. »tree.com« zeigt der Reihe nach sämtliche Verzeichnisse und Unterverzeichnisse auf der Festplatte. Mit dem Parameter »/f« werden auch die in den Verzeichnissen befindlichen Dateien aufgezählt.

Für Programmierer lohnt sich der Blick auf folgende Befehle: »debug.com«, das Maschinensprachewerkzeug im PC, wird nicht nur zur Fehlersuche verwendet. Unbedingt haben sollte man jedoch »gwbasic.exe«, denn Basic kann man in jeder Lebenslage brauchen, besonders für die einfachsten Aufgaben. Speziell für Programmierer sind »diskcomp.com«, »exe2bin.exe«, »fc.exe«, »comp.com« und »find.exe« gemacht. Aber selbst diese Zielgruppe benötigt diese Befehle oft nicht. Mit den nächsten drei, und damit auch letzten einigermaßen interessanten Befehlen, hat man eigentlich nur bei der Installation von MS-DOS zu tun. Sie brauchen daher nicht auf der Boot-Diskette oder der Festplatte präsent zu sein: »fdisk.com«, »select.com«, »sys.com«.

Die anderen Dateien werden nur in seltensten Spezialfällen verwendet und sind bis dahin bestens auf den Originaldisketten verwahrt. (sw)

Unverzichtbar	
command.com	
Wichtig	
diskcopy.com	keyboard.sys
format.com	chkdsk.com
keyb.com	xcopy.exe
Oft benötigt	
ansi.sys	print.com
mode.com	recover.com
backup.com	restore.com
more.com	
Nützlich	
fastopen.exe	join.exe
ramdrive.sys	label.com
append.exe	replace.exe
assign.com	share.exe
attrib.exe	sort.exe
edlin.com	subst.exe
graphics.com	tree.com
Für Programmierer	
debug.com	comp.com
diskcomp.com	find.exe
exe2bin.exe	gwbasic.exe
fc.exe	
Zur Installation von MS-DOS	
fdisk.com	sys.com
select.com	
Unwichtig	
nlsfunc.exe	country.sys
ega.cpi	display.sys
lcd.cpi	driver.sys
4201.cpi	printer.sys
5202.cpi	grftabl.com



**DMV
Power-Pack**



DMV

Weihnachts-Präsent

Die Grundausstattung für Einsteiger und Profis

ERGO 3.0

DM 199, - *

Die Benutzeroberfläche, die einfach alles hat

Neben der Arbeit mit Dateien und ganzen Verzeichnissen, dem Formatieren von Disketten, Druckerausgabe usw. bietet Ihnen ERGO noch eine Reihe von wichtigen und nützlichen Funktionen: Verwaltung von TSR-Programmen, je ein Text- und ein Byte-Editor, Paßwortschutz mit Zugriffsberechtigung, Briefing und eigene Anwendermenüs.

Mit ERGO arbeiten Sie unter dem modernsten PC-Benutzerstandard mit kompletter Maussteuerung in verschiebbaren Fenstern mit Scrollbalkenmenüs. Sie werden dabei in jeder Arbeitsphase von einem Online-Hilfe-Manual, einer ASCII-Tabelle, einem Kalender und einem wissenschaftlichen Taschenrechner unterstützt.

CONTEXT ADREVA

DM 49, - *

Universelle Adreßverwaltung

Außer in den ConText-Textverarbeitungen können Sie ConText-ADREVA auch in anderen Anwendungen nutzen, da die Adressen in einem allgemeinen ASCII-Dateiformat ausgegeben werden. Speicherresident installiert können Sie mit ConText-ADREVA bis zu 500 Datensätze verwalten, als normale DOS-Applikation wird die Anzahl der Datensätze nur durch die Hardware begrenzt.

Bei ConText-ADREVA können Sie mit Wildcards in allen Feldern suchen, Datenexport und -import vornehmen und verfügen über eine kontextbezogene Hilfe bei allen Funktionen.

Bitte benutzen Sie die Bestellkarte.

CONTEXT PRO

DM 199, - *

Textverarbeitung mit Konzept

Kompromißlose WYSIWYG-Darstellung:
Sie sehen Fett-, Kursiv- und Breitschrift, Unterstreichen, Hoch- und Tiefstellen original auf dem Bildschirm

Einfache und blitzschnelle Bedienung:

Sie steuern, unterstützt durch Hilfs-Fenster, über Pulldown-Menüs oder Tasten, wahlweise mit Mausbedienung

Vielseitigkeit und hohes Tempo:

Ihnen stehen alle Textverarbeitungs-Standardfunktionen plus zahlreiche durchdachte Extras zur Verfügung

Flexibilität:

Sie verfügen über einen Blockgrafikeditor, block- und spaltenweises Rechnen im Text und Tastenbelegung

DMV-Power-Pack

ERGO 3.0

CONTEXT PRO

CONTEXT ADREVA

DM 349, - *



Sie sparen
DM 98, -
gegenüber den
Einzelpreisen!

* Unabhängig von der Bestellmenge berechnen wir einmalig den Versandkostenanteil von DM 4, - für das Inland bzw. DM 6, - für das Ausland an Porto und Verpackung.

DMV-Verlag · Postfach 250 · 3440 Eschwege

DMV
Daten- und
Medienverlag

Der ganz persönliche Computer

Individuelle Anpassung der Config- und der Autoexec-Datei

Mit dem Begriff »Personal Computer«, dem »Persönlichen Computer« also, machte der Hersteller IBM schon bei der Vorstellung des PCs deutlich, daß dieser Rechner für die individuellen Bedürfnisse des Benutzers gedacht ist. Verwirklicht wird diese persönliche Anpassung über die Dateien »config.sys« und »autoexec.bat«.

Obwohl immer von der Gruppe der »Kompatiblen« die Rede ist, unterscheiden sich die Rechner zum Teil erheblich in ihrer Ausstattung und ihren Fähigkeiten. So gibt es unterschiedliche Speichergrößen und Grafikkarten, Drucker und andere Peripheriegeräte oder eine variable Anzahl von angeschlossenen Diskettenlaufwerken, Festplatten etc. Diese spezielle Ausstattung eines PC bezeichnet man als »Konfiguration«. Eine typische Konfiguration ist beispielsweise ein PC mit 640 KByte Hauptspeicher, 20-MByte-Festplatte, einem Diskettenlaufwerk und einer Hercules-Grafikkarte. Nun verhält es sich so, daß der PC während des Startvorgangs diese Konfiguration in den meisten Fällen selbst erkennt. Durch die geschickte Abfrage bestimmter Bausteine stellt der PC die Anzahl der angeschlossenen Diskettenlaufwerke, die Größe des Hauptspeichers und anderes mehr fest. Die Konfiguration, die der Rechner dabei ermittelt, ist die »Hardware-Konfiguration«, da sie lediglich, die physikalisch gegebenen Besonderheiten berücksichtigt.

Für den Anwender genauso wichtig ist jedoch die »Software-Konfiguration«. Dieser Begriff bezeichnet die gezielte Anpassung des Betriebssystems an die persönlichen Erfordernisse. Ein Beispiel für eine Software-Konfiguration ist die Anpassung der Tastatur. Der PC handhabt die Tastatur nach einem einfachen Prinzip: Jede Taste auf der Tastatur trägt eine bestimmte Nummer. Wird eine Taste gedrückt, so wird die entsprechende Nummer an den Rechner gesendet. Egal, wo der PC betrieben wird: ein Tastendruck in Sydney erzeugt den gleichen Code wie der Druck auf dieselbe Taste in Tokio, Amsterdam oder Frankfurt.

Dennoch erscheint auf den gleichen Tastendruck hin nicht immer der gleiche Buchstabe auf dem Bild-

schirm. Bei einer amerikanischen Tastatur beispielsweise liefert die Taste, auf der bei uns in Deutschland das »Z« liegt, ein »Y«. Der Grund dafür liegt in der Konfiguration der Tastatur. Über ein spezielles Programm, den sogenannten Tastaturtreiber, werden die Codes der Tastatur je nach Erfordernissen in unterschiedliche Buchstaben umgesetzt. Der amerikanische Tastaturtreiber gibt beim Druck auf unsere Z-Taste (die in den USA natürlich mit Y beschriftet ist) die Anweisung »gib ein »Y« auf dem Bildschirm aus«, der deutsche Treiber hingegen »gib ein »Z« auf dem Bildschirm aus«. So lassen sich je nach Land unterschiedliche Buchstaben erzeugen und auf diese Weise die nationalen Zeichensätze unterstützen.

Dazu benötigt der Computer jedoch Informationen darüber, welchen Tastaturtreiber er verwenden soll. Das ist einer der Sachverhalte, die in der Software-Konfiguration festgelegt werden.

Auf einem PC wird die Software-Konfiguration über eine spezielle Datei gesteuert. Diese Datei trägt den Namen »config.sys«. Die Datei »config.sys« enthält bestimmte Schlüsselwörter, die dem Rechner spezielle Konfigurationsvorgaben machen. Diese Schlüsselwörter sind mit den Kommandos einer Programmiersprache oder den Befehlen des Betriebssystems vergleichbar. Im Prinzip handelt es sich bei den Einträgen in

```
country = 049
buffers = 20
fcbs = 10,10
files = 20
device = c: \ dos \ mouse.sys
device = c: \ dos \ vdisk.sys 100
install = c: \ dos \ keyb gr
shell = c: \ dos \ command.com
```

Bild 1. Ein typischer Inhalt der Datei »config.sys«

der Konfigurationsdatei um eine Sammlung von Gleichungen (oder besser Wertzuweisungen). Hinter den speziellen Konfigurationsbefehlen wird nach einem Gleichheitszeichen der Wert angegeben, mit dem das Betriebssystem arbeiten soll.

Die Datei »config.sys« wird vom PC während des Startvorgangs in den Speicher eingelesen. Dabei werden die in ihr enthaltenen Kommandos der Reihe nach als Vorgaben für die Konfiguration verwendet.

Die wichtigste Angabe in »config.sys« ist die Festlegung der nationalen Parameter. Hierzu steht der Befehl »country« zur Verfügung. Zunächst wird hinter das Country-Kommando ein Gleichheitszeichen gesetzt, dann folgt der Code für eines der vom Betriebssystem unterstützten Länder. Dieser Code ist identisch mit der internationalen Telefon-Vorwahlnummer des entsprechenden Landes. In der Bundesrepublik beziehungsweise im deutschen Sprachraum wird daher der Code 049 verwendet.

Daß Ihre Tastatur eine deutsche Tastatur ist, teilen Sie dem PC in der Config-Datei mit

Die führende Null darf dabei nicht weggelassen werden. Neben dem Ländercode kann ab MS-DOS-Version 3.3 hinter einem Komma die Nummer der Zeichensatztafel (Code Page) angegeben werden, mit der der Rechner arbeiten soll. Wird keine Zeichensatztafel ausgewählt, arbeitet MS-DOS mit der standardmäßigen IBM-ASCII-Tabelle, sie trägt die Nummer 437.

Der Country-Befehl ist für die korrekte Funktion des Betriebssystems unbedingt erforderlich! Der Grund dafür liegt in der Tatsache, daß neben der Tastaturbelegung beispielsweise auch die Anzeige des Datums oder der Zeit nationalen Unterschieden unterliegt. In den USA oder Großbritannien werden Datum und Zeit zum Beispiel anders abgekürzt als in Deutschland. Das erste Kommando in der Datei »config.sys« sollte daher immer sein:

```
country=049
```

Mit dem dahinter angebrachten Zusatz »,437« können, wie oben schon angesprochen, Benutzer von MS-DOS-Versionen ab 3.3 noch die richtige Zeichensatztafel auswählen (in diesem Fall ist dies die IBM-ASCII-Tabelle).

Im zweiten Schritt sollten Sie in der Konfigurationsdatei die Parameter für die Datenein- und -ausgabe festle-

gen. Dazu dienen die Kommandos »buffers«, »fcbs« und »files«, die die Verarbeitungsgeschwindigkeit bei Disketten- und Festplattenoperationen zum Teil erheblich erhöhen.

Der Buffers-Befehl legt die Anzahl der Diskettenpuffer fest, die MS-DOS beim Disketten- und Festplattenzugriff verwendet. Bei jedem Lesezugriff auf die Diskette und die Festplatte wird nämlich vom Betriebssystem nicht nur die vom Programm angeforderte Zahl von Bytes gelesen, sondern immer ein ganzer Sektor. Dieser Sektor wird in einem der Diskettenpuffer gespeichert. Bei nachfolgenden Schreib- oder Lesezugriffen überprüft MS-DOS zunächst, ob sich die Daten nicht bereits in einem der Diskettenpuffer befinden. Wenn das der Fall ist, können sie direkt aus dem Puffer übernommen oder im Puffer verändert werden. Da der Zugriff auf den Hauptspeicher natürlich erheblich schneller ist, als der Zugriff auf die Festplatte oder gar die Diskette, steigert sich durch die Verwendung der Puffer die Geschwindigkeit beim Disketten- und Festplattenzugriff erheblich.

Geschickt gewählt, kann der Buffers-Befehl Ihren PC ganz gehörig beschleunigen

Mit dem Buffers-Konfigurationsbefehl wird die Anzahl der von MS-DOS verwendeten Puffer festgelegt. Wenn Sie jetzt annehmen, daß Sie einfach die Anzahl der Puffer auf den Höchstwert setzen müssen, um eine »superschnelle« Festplatte zu besitzen, liegen Sie leider nicht ganz richtig. Denn so einfach ist es nicht.

Es ist zu bedenken, daß jeder von MS-DOS eingerichtete Diskettenpuffer einen gewissen Speicherplatz beansprucht, pro Puffer genau 512 Byte für die Daten und 16 Byte für den Speichersteuerblock zum Puffer. Bei beispielsweise zehn Puffern summiert sich dies also schon auf über 5 KByte weniger Arbeitsspeicher. Dennoch läßt sich in den meisten Fällen die Pufferanzahl höher ansetzen, als MS-DOS dies standardmäßig tut. Wer mit mindestens 512 KByte Arbeitsspeicher arbeitet, ist mit 20 bis 25 Puffern gut bedient. Das entsprechende Kommando für die Konfigurationsdatei lautet:

```
buffers=20
```

Dadurch werden 20 Diskettenpuffer im Arbeitsspeicher angelegt, die Vorgabe eines größeren Wertes steht Ihnen natürlich frei. Es ist auch möglich, die Diskettenpuffer in den erweiterten Speicher zu verbannen

(falls vorhanden). Allerdings sollten Sie bedenken, daß eine hohe Anzahl an Puffern unter Umständen auch dazu führen kann, daß sich der Zugriff verlangsamt, da das MS-DOS bei jedem Zugriff alle Puffer nach eventuell darin vorhandenen Bytes durchsuchen muß.

Das Kommando »fcbs« befaßt sich ebenfalls mit der Ein- und Ausgabe von Datenträgern. Es bezieht sich allerdings auf die Methode der Dateisteuerblöcke (files control blocks), mit denen vor Verwendung von MS-DOS 2.0 Dateien geöffnet, geschlossen und verwaltet werden mußten. Ab MS-DOS 2.0 stehen zusätzlich zu den FCBs sogenannte »File Handles« zur Verfügung, mit denen sich diese Zugriffe erheblich einfacher durchführen lassen. Dennoch gibt es einige Programme (zum Beispiel Wordstar in älteren Versionen), die alle Ein- und Ausgaben über FCBs abwickeln. Zu diesem Zweck legen Sie mit dem Fcbs-Kommando in der Konfigurationsdatei fest, wie viele Dateisteuerblöcke gleichzeitig benutzt werden können. Zehn Dateisteuerblöcke sind bei den heute üblichen Arbeitsspeichergrößen, 512 KByte und mehr, eine angemessene Zahl.

Zusätzlich zu der Anzahl der möglichen FCBs läßt sich mit dem Fcbs-Befehl auch festlegen, wie viele der Dateisteuerblöcke MS-DOS vor dem automatischen Schließen schützen soll. Um Fehlermeldungen und Datenverluste zu vermeiden, empfiehlt es sich, das automatische Schließen für alle FCBs zu verbieten, also erneut die Anzahl der FCBs zu setzen. Der Eintrag in »config.sys« lautet also:

```
fcbs=10,10
```

Eine Parallele zum Fcbs-Befehl stellt das Kommando »files« dar. Wie schon erwähnt, wurde die Methode der Dateisteuerblöcke ab MS-DOS der Version 2.0 durch eine neue Methode erweitert: die Benutzung von »File Handles«. Zwar lassen sich auch die »alten« FCB-orientierten Funktionen ab MS-DOS 2.0 aus Kompatibilitätsgründen weiter verwenden, die Handle-orientierten Funktionen sollen jedoch laut Microsoft und IBM bei der Programmierung bevorzugt werden.

Mit den »File Handles« lassen sich Dateien wesentlich flexibler verwalten als über die schwerfälligen Dateisteuerblöcke. Ähnlich wie der Fcbs-Befehl für die FCB-Anzahl, erlaubt es das Files-Kommando die Anzahl der maximal gleichzeitig geöffneten Dateien festzulegen. Für viele Anwendungsprogramme (Compiler, Datenbanken oder Textverarbeitungen) müssen Sie hier eine relativ hohe Anzahl bereitstellen. Die Maximalzahl

liegt bei 255, ab MS-DOS 4.0 bei 65536. Natürlich wird auch für jeden dieser Handles wieder Speicherplatz benötigt: 38 Byte sind für einen Handle erforderlich. Dennoch ist der Platzbedarf der Handles nicht sonderlich groß. Ein Wert von 20 oder 25 reicht für fast alle Anwendungen aus. Daher gehört die folgende Zeile in die Konfigurationsdatei:

```
files=20
```

Die dritte wichtige Gruppe von Kommandos in der Config-Datei widmet sich der Anpassung von Geräten. Für jedes Gerät, das ideal in das Betriebssystem eingebunden werden soll, benötigt MS-DOS einen Gerätetreiber, den Sie ihm mit dem Device-Befehl in der Konfigurationsdatei mitteilen. »device« ist das englische Wort für »Gerät«. Mit dem Device-Kommando können Sie für bestimmte Geräte besondere Einstellungen setzen oder neue Geräte anpassen. Dazu stehen spezielle Gerätetreiber auf der Systemdiskette zur Verfügung.

Mit den »File Handles« ist die Dateiverwaltung ein gutes Stück flexibler als mit Dateisteuerblöcken

Sie erkennen diese Treiber an der Endung »sys«. Sie heißen also zum Beispiel »ansi.sys«, »vdisk.sys« oder »mouse.sys«. Mit dem Device-Befehl werden solche Gerätetreiber installiert. Dazu läßt sich der Device-Befehl einmal oder mehrmals in der Konfigurationsdatei verwenden. Sind beispielsweise ein Maustreiber und eine RAM-Disk von 100 KByte zu installieren, lauten die Kommando dazu:

```
device=mouse.sys  
device=vdisk.sys 100
```

Wenn sich die Gerätetreiber »mouse.sys« und »vdisk.sys« (oft auch »ramdrive.sys« genannt) nicht im Hauptverzeichnis befinden, muß dem Dateinamen der entsprechende Pfad vorangestellt werden.

Ab MS-DOS der Version 3.3 können Sie Drucker und Bildschirm mit dem Device-Befehl auch auf die Verwendung von Zeichensatztabellen vorbereiten. Dazu stehen die Gerätetreiber »printer.sys« und »display.sys« bereit. Eine Beschreibung dieser Befehle finden Sie in der Config-Befehlsreferenz in dieser Sonderheft.

Für den Benutzer von MS-DOS bis zur Version 4.0 sind das die wichtigsten Kommandos, die er in die Config-Datei einbinden sollte. Es gibt allerdings noch einige spezielle Befehle, die sehr selten verwendet werden. Sie sind ebenfalls in der Config-

Befehlsreferenz in diesem Sonderheft dokumentiert.

AB MS-DOS 4.0 wurde der »Wortschatz« von »config.sys« noch um einige leistungsfähige Befehle erweitert. Ein sehr nützlicher gehört dazu – die Install-Direktive. Damit lassen sich residente Programme, die bis MS-DOS 4.0 erst in der Datei »autoexec.bat« geladen werden konnten, schon in der Datei »config.sys« implementieren. Zu ihnen gehören der Tastaturtreiber und einige Sonderfunktionen. Doch dazu später mehr.

In der MS-DOS-Version 4.0 wurde der Config-Wortschatz um einige Befehle erweitert

Eine weitere Neuerung ab MS-DOS 4.0 ist, daß der verwendete Kommandoprozessor (die sogenannte »Shell«) explizit in der Datei »config.sys« festgelegt werden muß. Der Vorteil dieses Verfahrens liegt einerseits in der Flexibilität (durch Verwendung unterschiedlicher Kommandoprozessoren) und in der Tatsache, daß der Kommandoprozessor nicht mehr im Hauptverzeichnis stehen muß. Möchten Sie mit der gewohnten MS-DOS-Oberfläche arbeiten, müssen Benutzer von MS-DOS ab Version 4.0 also in jedem Fall die folgenden Zeile in »config.sys« unterbringen:

```
shell=command.com
```

Gegebenenfalls müssen Sie vor den Dateinamen noch die Pfadangabe für die Datei »command.com« setzen, sofern sie sich nicht im Hauptverzeichnis befindet.

Wie Sie sehen, es ist gar nicht so einfach, sich eine effektive Konfigurationsdatei zusammenzustellen. Benutzer von MS-DOS ab Version 4.0 haben es da einfacher. Sie können mit dem Befehl »select« automatisch (oder besser: halbautomatisch) eine Datei »config.sys« erzeugen.

Leider ist es mit der individuellen Anpassung der Config-Datei allein noch nicht getan. Zur Konfiguration gehören nämlich auch noch Einträge in der Datei »autoexec.bat«. Führen Sie sich dazu zunächst noch einmal den Prozeß vor Augen, der nach dem Einschalten eines PCs abläuft.

Zunächst wird die Hardware-Konfiguration festgestellt und initialisiert. Im nächsten Schritt wird der Betriebssystemkern von der Diskette oder Festplatte geladen, danach die Software-Konfiguration gemäß »config.sys« ausgeführt. Wenn das geschehen ist, wird der Kommandoprozessor (meist »command.com«) gela-

den. Dieser wiederum lädt das Startprogramm mit dem Namen »autoexec.bat« und führt die darin enthaltenen MS-DOS-Befehle aus. Nun können aber nicht alle Konfigurationsschritte schon in der Datei »config.sys« erfaßt werden, zum Beispiel läßt sich die Anpassung der Tastatur (zumindest bei MS-DOS-Versionen vor 4.0) dort nicht vornehmen, da hier ein zusätzliches Programm gestartet werden muß. Um diese Konfigurationsschritte nicht nach jedem Rechnerstart »per Hand« eingeben zu müssen, ist es ratsam, sie in der Datei »autoexec.bat« zu automatisieren.

Im Prinzip ist die Datei »autoexec.bat« eine »ganz normale« Batch-Datei. Der PC liest sie vom Datenträger und führt die in ihr enthaltenen Befehle der Reihe nach aus. Den gleichen Effekt würden Sie erzielen, wenn Sie die Befehle einzeln eingeben würden. Der Unterschied von »autoexec.bat« zu einer »normalen« Batch-Datei liegt in der Tatsache begründet, daß das Betriebssystem die Datei »autoexec.bat« automatisch nach dem Start ausführt (daher auch der Name, als Abkürzung für den englischen Begriff »automatic execution«, also »automatische Ausführung«).

Als erster Befehl sollte in »autoexec.bat« die folgende Zeile stehen:

```
echo off
```

Sie sorgt dafür, daß die nachfolgenden Befehle zwar ausgeführt, die jeweiligen Befehlsnamen aber nicht auf dem Bildschirm ausgegeben werden.

Die Autoexec-Datei hat einer normalen Batch-Datei ein Quentchen voraus: Sie kommt zuallererst

Benutzer von MS-DOS ab der Version 4.0 sollten in der zweiten Zeile angeben, wo sich der Kommandoprozessor auf dem Datenträger befindet (zur Erinnerung: ab MS-DOS 4.0 muß der Kommandoprozessor nicht mehr im Hauptverzeichnis stehen). Dazu wird der Befehl »set comspec« verwendet. Wenn sich also die Datei »command.com« beispielsweise im Unterverzeichnis »c:\system« befindet, muß die folgende Zeile in »autoexec.bat« eingefügt werden:

```
set comspec=c:\system\command.com
```

Als nächsten Schritt passen Sie die deutsche Tastatur an, dazu gibt es auf der Systemdiskette spezielle Tastaturtreiber. Dieser Treiber heißt »keybgr.com« für MS-DOS bis Version 3.3 und »keyb.com« ab einschließlich Version 3.3. Der Tastaturtreiber ist

ein gewöhnliches Com-Programm, es wird also durch Eingabe des Programmnamens aufgerufen. Bei MS-DOS-Versionen vor 3.3 existiert für jedes Land ein eigener Tastaturtreiber, der deutsche heißt »keybgr.com« (für »keyboard german«, also »Tastatur deutsch«). Benutzer von MS-DOS bis Version 3.3 müssen also die folgenden Zeile in »autoexec.bat« einfügen:

```
keybgr
```

Ab Version 3.3 gibt es nur noch ein einziges Programm für alle Tastaturen, es heißt »keyb.com«. Hier wird der Buchstabencode für den zu verwendenden Tastaturtreiber (also »gr« für Deutschland) als Parameter an das Programm übergeben. Daher muß vor dem »gr« ein Leerzeichen stehen. Benutzer von MS-DOS ab Version 3.3 fügen also die folgenden Zeile ein:

```
keyb gr
```

Falls sich der Tastaturtreiber nicht im Hauptverzeichnis befindet, muß die Pfadangabe natürlich vor dem »keybgr« beziehungsweise vor dem »keyb gr« zu finden sein. Der Tastaturtreiber ist ein residentes Programm, das – einmal geladen – ständig dafür sorgt, daß der PC mit der gewohnten deutschen Tastenbelegung arbeitet.

Für Benutzer ab MS-DOS-Version 4.0 gibt es sogar noch eine einfachere Möglichkeit. Sie können den Tastaturtreiber gleich in der Datei »config.sys« spezifizieren. Mit dem Install-Befehl lassen sich bestimmte residente Programme noch vor der Ausführung von »autoexec.bat« laden. Benutzer von MS-DOS ab Version 4.0 können demnach die obige Zeile »keyb gr« aus »autoexec.bat« entfernen und statt dessen die folgende Zeile in »config.sys« einfügen:

```
install=keyb gr
```

Falls sich der Tastaturtreiber »keyb.com« nicht im Hauptverzeichnis befindet, müssen Sie natürlich wieder einen Pfadnamen voranstellen.

Nach der Festlegung der deutschen Tastatur ist die eigentliche Software-Konfiguration abgeschlossen. Ein wichtiger Befehl sorgt jedoch für zusätzlichen Arbeitskomfort – der Path-Befehl.

Der Path-Befehl legt fest, in welchen Verzeichnissen MS-DOS zusätzlich suchen soll, wenn es einen Befehl oder ein Programm im aktiven Ver-

```
echo off
set comspec=c:\dos\command.com
path=c:\dos
word
```

Bild 2. Ein typischer Inhalt der Datei »autoexec.bat«

zeichnis nicht findet. Damit Sie die MS-DOS-Kommandos auch aus allen Unterverzeichnissen heraus nutzen können, empfiehlt es sich, mit dem Path-Kommando das Verzeichnis festzulegen, in dem sich die Dateien des Betriebssystems befinden. Sind sie im Hauptverzeichnis gespeichert, so sorgt die folgende Zeile dafür:

```
path=c:\
```

Um MS-DOS die Suche nach bestimmten Dateien zu erleichtern, legen Sie einfach »Pfade«

Haben Sie die Dateien des Betriebssystems (das heißt, die externen MS-DOS-Kommandos) in einem Unterverzeichnis untergebracht, geben Sie statt dessen natürlich den Namen des Verzeichnisses an. Finden sich alle Systemdateien beispielsweise im Verzeichnis »c:\dos«, so lautet der entsprechende Path-Befehl:

```
path=c:\dos
```

Bei manchen Rechnern wird die Maus nicht über einen eigenen Gerätetreiber, sondern über ein residentes Programm betrieben. Hier ist ein zusätzliches Programm (meist »mouse.com« oder »mouse.exe« genannt) zur Kontrolle der Maus zu starten. Die erforderliche Zeile lautet dann:

```
mouse
```

Befindet sich der Maustreiber nicht im Hauptverzeichnis, wird natürlich wieder eine Pfadangabe vorangestellt.

Arbeiten Sie häufig mit nur einem speziellen Programm (zum Beispiel einer Textverarbeitung), so können Sie als letzten Befehl ohne weiteres auch das Kommando zum Start des entsprechenden Programms in die Datei »autoexec.bat« einbinden. Mit der Anpassung von »config.sys« und »autoexec.bat« an die persönlichen Erfordernisse läßt sich also eine Menge an der Funktionsweise des Rechners ändern. Die Bilder 1 und 2 zeigen die Inhalte für die Dateien »config.sys« und »autoexec.bat«, wie Sie sie eben kennengelernt haben. Beachten Sie bitte, daß diese Dateien auf MS-DOS 4.0 abgestimmt sind. Für MS-DOS vor Version 4.0 müssen Sie die Install-Anweisung aus »config.sys« herausnehmen und den Befehl »keybgr« oder »keyb gr« (je nach MS-DOS-Version) in »autoexec.bat« einbinden.

Nun werden Sie aber sicher fragen, wie Sie denn diese Dateien auf der Festplatte oder Diskette erzeugen können. Dazu gibt es nun einfache und weniger einfache Methoden. Die unkomplizierteste überhaupt ist

sicher die Copy-Methode. Hierbei geben Sie die Zeilen der Reihe nach ein – die Methode hat jedoch den Nachteil, daß bei einer einmal fertiggestellten Zeile keine Korrekturmöglichkeit mehr gegeben sind. Bei der Copy-Methode kopieren Sie die Zeilen direkt von der Tastatur in die entsprechende Datei. Als erste Zeile geben Sie dazu ein:

```
copy con \config.sys
```

Dieser Befehl weist den PC an, die Tastatur wie eine Datei zu öffnen und den Inhalt dieser Datei in die Datei »config.sys« zu kopieren. Natürlich können Sie anstelle von »config.sys« auch einen anderen Dateinamen wählen. Der Backslash »\« vor dem Dateinamen ist in diesem Fall unbedingt erforderlich, damit die Datei auf jeden Fall in das Stammverzeichnis der Diskette oder Festplatte gelangt.

Die Copy-Methode kopiert ganz normale Tastatureingaben in eine Datei

Es mag zwar ein bißchen verwirrend erscheinen, die Tastatur als Datei zu öffnen, es erfüllt aber in diesem Fall voll seinen Zweck. Der Rechner kopiert nämlich nun alle Zeichen, die Sie auf der Tastatur eingeben, in die gewünschte Datei. Das macht er so lange, bis er das Ende der Datei erkennt. Sie geben also nun jede einzelne Zeile Ihrer Datei ein und beenden eine Zeile jeweils mit der Taste [Return]. Haben Sie eine Zeile auf diese Weise abgeschlossen, läßt sich der Inhalt der entsprechenden Zeile nicht mehr korrigieren. Überprüfen Sie also vor der Betätigung von [Return] genau, ob Sie sich nicht vertippt haben. In einem solchen Fall können Sie nämlich die Buchstaben der Reihe nach, wie gewohnt, löschen und den Fehler korrigieren. Auf diese Weise vervollständigen Sie Ihre Datei und schließen auch die letzte Zeile mit [Return] ab. Nun müssen Sie dem PC noch irgendwie mitteilen, daß die »Tastaturredatei« zu Ende ist. Dazu machen Sie sich die Tatsache zunutze, daß der PC für das Ende einer Datei einen Code verwendet, der sich durch die Tastenkombination [Ctrl-Z] erzeugen läßt. Sie tippen also die beiden Tasten [Ctrl] (auf einigen deutschen Tastaturen auch »Strg« genannt) und [Z] gleichzeitig und schließen danach wieder mit [Return]. Der Computer antwortet mit der Meldung, daß er eine Datei kopiert hat. Er hat also die gewünschten Einträge in die Datei eingefügt.

(Martin Althaus/hi)

Von Einsteigern für Einsteiger

Einsteiger haben ganz spezielle Sorgen und Probleme, die ein »alter Hase« in Sachen Computer manchmal kaum mehr nachvollziehen kann. Deshalb sind gerade die Einsteiger auf ihre Einsteiger-Kollegen angewiesen.

**Sie haben Probleme mit Ihrem PC?
Andere haben dieselben!**

**Sie haben Ihre Probleme gelöst?
Andere noch nicht!**

Aus diesem Grund suchen wir Tips und Tricks von Einsteigern für Einsteiger. Bitte denken Sie nicht, daß Ihre Problemlösung, auf die Sie mit Recht stolz sind, so trivial sei, daß niemand sich dafür interessieren würde. Im Gegenteil, die Erfahrung zeigt, daß gerade in den Anfängen einfache Tips zu grundlegenden Problemen gefragt sind, und die kann niemand besser geben, als jemand, der sich noch vor kurzer Zeit mit eben diesen Problemen auseinandergesetzt.

Seien Sie fair, und lassen Sie auch andere von Ihren Erfahrungen profitieren. Schicken Sie uns Ihre Tips, und zwar an das

**Redaktionsbüro
Everts & Hagedorn
Redaktion
DOS International
Stichwort Einsteiger-Tips
Hans-Stießberger-Straße 3
8013 Haar/München**

Der Debug-Quicky

Tips zum schnellen Abtippen von Listings

Häufig finden Sie in der DOS International oder in der DOS Extra sogenannte »Hexdumps« von kurzen Programmen. Normalerweise brauchen Sie zum Abtippen dieser Listings unsere Eingabehilfe »Scheck«, um aus dem Hexdump ein ausführbares Programm für Ihren PC zu machen. Mit dem Betriebssystem Ihres Computers wird jedoch ein Hilfsprogramm geliefert, das Sie für die schnelle Übersetzung von kurzen Hexdumps einsetzen können.

Zunächst vergewissern Sie sich bitte, daß Sie das Programm »debug.com« entweder in einem über den MS-DOS-Pfad erreichbaren Verzeichnis oder auf einer Diskette »griffbereit« haben. Der Debugger »debug.com« läßt sich zum Übersetzen von kurzen »Hexdumps« verwenden.

Ein Hexdump ist im Prinzip ein ausführbares Programm, das allerdings noch in Textform vorliegt. Der Computer kann mit dieser Textform jedoch nichts anfangen, Sie müssen den Text erst in Binärdaten umwan-

deln. Die Eingabehilfe »Scheck«, wie Sie auch in dieser DOS Extra veröffentlicht ist, führt unter anderem diese Wandlung durch.

Das Abtippen der Eingabehilfe nimmt einige Zeit in Anspruch. Vielleicht haben Sie aber nur ein kurzes Listing in der DOS gefunden, und möchten jetzt nicht erst »Scheck« abtippen. Mit Hilfe des MS-DOS-Debuggers »debug.com« kann Ihnen geholfen werden. Allerdings müssen Sie sich im klaren darüber sein, daß Sie so keinerlei automatisierte Kon-

trolle über eventuelle Tippfehler haben. Hexdump-Listings sind in der DOS immer mit Prüfsummen für jede Zeile versehen, anhand derer Sie in Zusammenarbeit mit »Scheck« eventuelle Tippfehler leicht feststellen können. Bei kurzen Listings bis zirka 50 Zeilen ist »debug.com« aber ausreichend.

Zunächst müssen Sie das Hexdump-Listing abtippen, am besten mit einer Textverarbeitung. Notfalls genügt auch der MS-DOS-Editor »edlin«. Der Name der Textdatei, die Sie gleich eingeben werden, ist unwichtig. Im allgemeinen beläßt man den Dateinamen und ändert nur die Erweiterung auf »deb« oder »inp«. Um den Hexdump des Programms »tauschab.com« einzugeben, speichern Sie ihn also unter »tauschab.deb« oder »tauschab.inp«. Die Textdatei wird später dem MS-DOS-Debugger mitteilen, was er zu tun hat.

Beim Abtippen lassen Sie die in eckigen Klammern stehenden Prüfsummen und die Zeilennummern weg. Wenn im Hexdump-Listing in der DOS also steht

```
[EB5] 1: EB 3A 90 00 ...
```

dann tippen Sie ein:

```
EB 3A 90 00 ...
```

Jede Zeile (mit Ausnahme der letzten) besteht aus 16 zweistelligen Hexadezimalzahlen, die durch Leerzeichen getrennt sind.

Das genügt dem MS-DOS-Debugger, der mit diesen Zeilen ja etwas anfangen soll, jedoch noch nicht. An den Anfang jeder Zeile gehören noch zwei weitere Informationen. Zum einen der Buchstabe »E« und ein Leerzeichen und zum anderen eine hexadezimale Speicheradresse. Die sieht zwar sehr kompliziert aus, ist es aber nicht:

Beginnen Sie in der ersten Zeile mit der Zahl 100 und zählen Sie dann immer 10 hinzu (110, 120, 130...). Bei 190 lautet der nächste Wert jedoch nicht 200, sondern 1A0. Danach folgen 1B0, 1C0, 1D0, 1E0, 1F0 und schließlich 200. Eine Ziffer wird also von 0 bis 9 und dann von A bis F durchgezählt. Genauso folgt auf 2F0 die 300 und auf die 9F0 eine A00. Sie haben übrigens soeben mit dem hexadezimalen Zahlensystem Bekanntheit gemacht.

Bleiben wir aber beim konkreten Beispiel. Die erste Zeile des hier abgedruckten Hexdump-Listings geben Sie also mit Ihrer Textverarbeitung so ein:

```
E 100 EB 3A 90 00 ...
```

Die zehnte Zeile lautet dann:

```
E 1A0 AB B0 C4 8B ...
```

Und die letzte Zeile unseres Beispiel-Listings sieht so aus:

```
E 340 07
```

So sieht eine vollständige Eingabedatei »tauschab.deb« für den MS-DOS-Debugger aus. Der Debugger verwandelt diesen Text in das fertige Programm »tauschab.com«

```
E 100 EB 3A 90 00 00 00 00 2E 88 16 03 01 80 FA 00
E 110 74 08 80 FA 01 74 08 EB 0B 90 B2 01 EB 06 90 B2
E 120 00 EB 01 90 50 55 8B EC 8B 46 08 87 46 02 5D 2E
E 130 FF 1E 04 01 2E 8A 16 03 01 CA 02 00 B8 13 35 CD
E 140 21 2E 89 1E 04 01 2E 8C 06 06 01 B8 13 25 BA 08
E 150 01 8C CB 8E DB CD 21 FC B4 06 B0 00 33 C9 BA 50
E 160 19 B7 07 55 CD 10 5D B4 02 33 D2 B7 00 CD 10 E8
E 170 08 00 B8 00 31 BA 14 00 CD 21 B8 30 00 8B E8 BA
E 180 08 00 B8 00 B8 8E C0 BE DE 03 8B FE B4 1F B0 DA
E 190 AB B0 C4 8B CD F3 AB B0 BF AB 81 C6 A0 00 8B FE
E 1A0 B0 B3 AB B0 20 8B CD F3 AB B0 B3 AB 4A 75 EB 81
E 1B0 C6 A0 00 8B FE B0 C0 AB B0 C4 8B CD F3 AB B0 D9
E 1C0 AB 8D 36 E9 01 BF 86 04 B8 2B 00 8B E8 BA 0B 00
E 1D0 B4 1C 2E 8A 04 AB 46 4D 75 F3 3C 07 74 0A B8 2B
E 1E0 00 8B E8 83 C7 4A EB E5 C3 C9 CD CD CD CD CD
E 1F0 CD CD CD CD CD CD CD CD CD CD CD CD CD CD
E 200 CD CD CD CD CD CD CD CD CD CD CD CD CD CD
E 210 CD CD BB 20 BA 20 20 20 20 20 20 20 20 54 41
E 220 55 53 43 48 41 42 20 20 20 20 20 20 49 6E 73 74 61
E 230 6C 6C 69 65 72 74 20 20 20 20 20 20 20 BA 20 CC
E 240 CD CD CD CD CD CD CD CD CD CD CD CD CD CD
E 250 CD CD CD CD CD CD CD CD CD CD CD CD CD CD
E 260 CD CD CD CD CD CD CD CD CD CD CD CD CD CD
E 270 20 20 20 20 20 20 3C 3C 3C 3C 3C 20 2D 2D 2D 2D
E 280 2D 2D 2D 2D 2D 2D 3E 3E 3E 3E 3E 20 20 20 20 20
E 290 20 20 20 BA 20 BA 20 20 20 20 20 20 20 20 20
E 2A0 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20
E 2B0 20 20 20 20 20 20 20 20 20 20 20 20 20 BA 20
E 2C0 BA 20 20 20 20 20 20 CD CD CD CD CD DB DB CD
E 2D0 CD CD CD 20 20 20 20 20 CD CD CD CD DB DB
E 2E0 CD CD CD CD CD 20 20 20 BA 20 BA 20 20 20
E 2F0 20 20 20 20 20 20 20 20 20 20 20 20 20 20
E 300 20 20 20 20 20 20 20 20 20 20 20 20 20 20
E 310 20 20 20 BA 20 C8 CD CD CD CD CD CD CD 20
E 320 63 2E 20 44 4F 53 20 2D 20 49 6E 74 65 72 6E 61
E 330 74 69 6F 6E 61 6C 20 CD CD CD CD CD CD BC
E 340 07
```

N TAUSCHAB.COM

R CX
241

W

Q

An den Schluß dieses Textes schreiben Sie in jeweils eigene Zeilen noch vier Befehle für den Debugger:

1. Ein »N« und ein Leerzeichen und dahinter den Namen des Programms, also beispielsweise:

N tauschab.com

Den Namen erfahren Sie aus der Bildunterschrift des Listings. Dort steht dann beispielsweise: »Listing 1. Das TSR-Utility »tauschab.com« vertauscht die Gerätenamen der Diskettenlaufwerke A und B«.

Damit haben Sie dem Debugger mitgeteilt, unter welchem Namen er das Programm speichern soll.

2. Die nächste Information, die Sie mit dem Texteditor eingeben, ist die Programmlänge. Dies ist die Gesamtzahl aller zweistelligen Hexadezimalzahlen, die im Listing aufgeführt sind. Zur Berechnung nehmen Sie entweder einen Taschenrechner, der Dezimal- in Hexadezimalzahlen umwandeln kann, oder Sie starten den GW-Basic-Interpreter, der auf den Systemdisketten Ihres PC unter dem Namen »gwbasic.exe« gespeichert ist. Multiplizieren Sie die Anzahl der Zeilen, die volle 16 Werte enthalten (siehe Zeilennummern im Listing), mit 16 und addieren Sie die Anzahl der Werte in der verbleibenden Zeile. Für das Beispiel-Listing wären das 36

volle Zeilen plus ein einzelner Wert, nämlich die »07«. Im GW-Basic-Interpreter tippen Sie ein:

```
PRINT HEX$( 36 * 16 + 1 )
```

Als Ergebnis erhalten Sie die hexadezimale Zahl 241hex. Ans Ende des abgetippten Listings schreiben Sie jetzt in eine eigene Zeile

```
r cx
```

und direkt darunter den eben berechneten Hexadezimalwert:

```
241
```

3. Die letzten beiden Zeilen des Listings sind ganz kurz. Sie bestehen nur aus den beiden Buchstaben »w« und »q«.

4. Das »w« veranlaßt den Debugger, das eingegebene Programm unter dem mit der N-Zeile festgelegten Namen zu speichern (hier also »tauschab.com«) und »q« beendet die Eingabe.

Das Beispiel-Listing auf dieser Seite stammt übrigens aus der Rubrik »Tips & Tricks für Programmierer« (DOS 8'89, Seite 173). Es vertauscht die Bezeichnungen zweier Diskettenlaufwerke. Nachdem es einmal aufgerufen wurde, hat das Laufwerk A also den Namen B und eine Diskette in Laufwerk B ist jetzt unter dem Laufwerksnamen A zu erreichen. Wenn Sie das Beispiel-Listing nach der

eben beschriebenen Methode eingegeben haben, erhalten Sie eine Textdatei, die so aussieht wie hier abgebildet. Die Leerzeichen am Schluß sind nicht nötig, sie dienen lediglich der optischen Gliederung. Danach verlassen Sie Ihre Textverarbeitung und tippen auf der MS-DOS-Kommandoebene ein:

```
debug < tauschab.deb
```

Sie starten dadurch den Debugger und »füttern« ihn mit den Befehlen, die in der Datei »tauschab.deb« enthalten sind. Je nachdem, welchen Namen Sie der Textdatei gegeben haben, kann »tauschab.deb« natürlich auch anders heißen.

Auf dem Bildschirm sehen Sie jetzt blitzschnell alle Befehle vorbeiziehen, die in der Textdatei enthalten sind. Nach einem kurzen Moment befinden Sie sich wieder im Eingabemodus des PCs, und im aktuellen Verzeichnis steht das fertige Programm »tauschab.com« (überprüfen Sie dies mit dem Befehl »dir«). Falls Sie keinen Tippfehler gemacht haben, funktioniert das Programm beim Aufruf mit »tauschab« jetzt einwandfrei. Ansonsten müssen Sie die Textdatei Zeile für Zeile mit dem abgedruckten Listing vergleichen und eventuelle Eingabefehler korrigieren.

(tr)

```
[EB5] 1: EB 3A 90 00 00 00 00 00 2E 88 16 03 01 80 FA 00
[CN5] 2: 74 08 80 FA 01 74 08 EB 0B 90 B2 01 EB 06 90 B2
[XH5] 3: 00 EB 01 90 50 55 8B EC 8B 46 08 87 46 02 5D 2E
[VG5] 4: FF 1E 04 01 2E 8A 16 03 01 CA 02 00 B8 13 35 CD
[7H5] 5: 21 2E 89 1E 04 01 2E 8C 06 06 01 B8 13 25 BA 08
[BT5] 6: 01 8C CB 8E DB CD 21 FC B4 06 B0 00 33 C9 BA 50
[9L5] 7: 19 B7 07 55 CD 10 5D B4 02 33 D2 B7 00 CD 10 E8
[EK5] 8: 08 00 B8 00 31 BA 14 00 CD 21 B8 30 00 8B E8 BA
[6Z5] 9: 08 00 B8 00 B8 8E C0 BE DE 03 8B FE B4 1F B0 DA
[VZ5] 10: AB B0 C4 8B CD F3 AB B0 BF AB 81 C6 A0 00 8B FE
[4W5] 11: B0 B3 AB B0 20 8B CD F3 AB B0 B3 AB 4A 75 EB 81
[TY5] 12: C6 A0 00 8B FE B0 C0 AB B0 C4 8B CD F3 AB B0 D9
[YM5] 13: AB 8D 36 E9 01 BF 86 04 B8 2B 00 8B E8 BA 0B 00
[HN5] 14: B4 1C 2E 8A 04 AB 46 4D 75 F3 C0 07 74 0A B8 2B
[936] 15: 00 8B E8 83 C7 4A EB E5 C3 C9 CD CD CD CD CD
[SF6] 16: CD CD CD CD CD CD CD CD CD CD CD CD CD CD CD
[SF6] 17: CD CD CD CD CD CD CD CD CD CD CD CD CD CD CD
[6C5] 18: CD CD BB 20 BA 20 20 20 20 20 20 20 20 54 41
[G95] 19: 55 53 43 48 41 42 20 20 20 20 20 20 49 6E 73 74 61
[2D5] 20: 6C 6C 69 65 72 74 20 20 20 20 20 20 20 BA 20 CC
[SF6] 21: CD CD CD CD CD CD CD CD CD CD CD CD CD CD CD
```

```
[SF6] 22: CD CD CD CD CD CD CD CD CD CD CD CD CD CD CD
[0X5] 23: CD CD CD CD CD CD CD CD CD B9 20 BA 20 20 20 20
[RE5] 24: 20 20 20 20 20 20 3C 3C 3C 3C 3C 20 2D 2D 2D 2D
[ZF5] 25: 2D 2D 2D 2D 2D 2D 3E 3E 3E 3E 3E 20 20 20 20 20
[P65] 26: 20 20 20 BA 20 BA 20 20 20 20 20 20 20 20 20 20
[G15] 27: 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20
[145] 28: 20 20 20 20 20 20 20 20 20 20 20 20 20 20 BA 20
[325] 29: BA 20 20 20 20 20 20 CD CD CD CD CD DB DB CD
[2X5] 30: CD CD CD 20 20 20 20 20 20 CD CD CD CD CD DB DB
[6N5] 31: CD CD CD CD CD 20 20 20 20 BA 20 BA 20 20 20 20
[G15] 32: 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20
[G15] 33: 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20
[LX5] 34: 20 20 20 20 BA 20 C8 CD CD CD CD CD CD CD 20
[DE5] 35: 63 2E 20 44 4F 53 20 2D 20 49 6E 74 65 72 6E 61
[R36] 36: 74 69 6F 6E 61 6C 20 CD CD CD CD CD CD CD BC
[580] 37: 07
```

Listing. So sieht ein Hexdump-Listing (hier das Utility »tauschab.com«) in der DOS International aus. Um es in ein ablauffähiges Programm umzuwandeln, verwenden Sie am einfachsten den MS-DOS-Debugger

Wer — wo — wann?

Bei Zuschriften bitten wir, folgendes zu beachten: Haben Sie Fragen, Anregungen oder Programme, die Sie der DOS-Redaktion zukommen lassen möchten, adressieren Sie wie folgt:

Redaktionsbüro Everts & Hagedorn · Redaktion DOS International
Hans-Stießberger-Str. 3 · 8013 Haar/München · Tel. (089) 46 10 01-0

Haben Sie Fragen, die zum Beispiel Abonnements, Kleinanzeigen oder Databox-Bestellungen betreffen, wenden Sie sich bitte an den Verlag:

DMV-Verlag · Postfach 250 · 3440 Eschwege

Am Anfang und für zwischendurch

Da steht er nun, der neue PC, ohne Textverarbeitung, ohne Benutzeroberfläche – ohne alles. Doch die Dateien »autoexec.bat« und »config.sys« wollen an die eigenen Anforderungen angepaßt sein. Hilfe bietet der Texteditor »Edlin«, der ein Bestandteil des MS-DOS-Lieferumfangs ist.

Edlin ist ein kurzes Programm, das in Sekundenschnelle geladen ist. Auch der Profi wird für eine kleine Tippfehlerverbesserung in Batch-Dateien nicht extra eine Textverarbeitung laden. Edlin kennt nur wenige Befehle, die meist aus dem Anfangsbuchstaben der gewünschten Bearbeitungsfunktion bestehen und sich daher leicht einprägen. Diesen Texteditor finden Sie auf Ihrer MS-DOS-Diskette oder in dem Festplattenverzeichnis, in dem MS-DOS installiert ist.

Bevor Sie mit Edlin arbeiten, müssen Sie den Suchpfad des Betriebssystems so umleiten, daß Sie den Editor aus jedem beliebigen Verzeichnis starten können. Legen Sie zum ersten Arbeiten den Suchpfad manuell an. Im weiteren Verlauf dieser Einführung finden Sie ein Übungsbeispiel, das zeigt, wie Sie mit Hilfe von Edlin den Suchpfad in Ihre Startdatei »autoexec.bat« schreiben. Befindet sich Edlin im Verzeichnis »dos« auf Ihrer Festplatte C, dann geben Sie den Suchpfad mit

```
path c: \ dos
```

an. Erzeugen Sie eine Sicherheitskopie der Datei unter dem Namen »autoexec.kop«:

```
copy autoexec.bak autoexec.kop
```

Ab sofort arbeiten Sie mit »autoexec.bat« als Übungs- oder Arbeitskopie. Sollten Sie beim Ändern mit Edlin etwas falsch machen, können Sie jederzeit die ursprüngliche Version »autoexec.kop« wieder zu »autoexec.bat« umkopieren.

Geben Sie nun zum Start der Bearbeitung die Kombination »edlin Dateiname« ein:

```
edlin autoexec.bat
```

Übrigens ist jede Befehlszeile, die Sie Edlin mitteilen, mit einem [Enter] abzuschließen. Am Bildschirm erscheint nun die Meldung

Ende der Eingabedatei

Diese Meldung signalisiert, daß die angegebene Datei in den Texteditor geladen wurde. Sollte die Datei mehr Speicherplatz beanspruchen, als Edlin bearbeiten kann, laden Sie bei Bedarf, mit der Anweisung »a« (»append«, englisch für anhängen), weitere Zeilen in den Speicher.

Fordern Sie nun eine Liste der Zeilen an, die in den Texteditor eingelesen wurden. Der Cursor steht bereits in der Befehlseingabeposition (in der zweiten Spalte neben dem Zeichen »*«). Geben Sie hier den Buchstaben »l« (für »list«) ein.

Edlin zeigt die ersten zwölf Zeilen der Datei auf dem Bildschirm an. Hat eine Datei mehr als zwölf Zeilen, dann erhalten Sie mit dem Kommando »l« immer eine Liste der aktuellen Zeile, der elf vorangehenden Zeilen und der elf nachfolgenden Zeilen. Bei den meisten Einsteigern enthält die Datei »autoexec.bat« nicht mehr als zwölf Zeilen – damit haben Sie eine vollständige Liste der Datei auf dem Bildschirm.

Wenn Edlin der Speicher nicht reicht, laden Sie weitere Zeilen mit dem Append-Befehl

Einen Auszug der Liste fordern Sie ebenfalls mit der Anweisung »l« an. Die auszugebenden Zeilen tragen Sie vor dem Befehl ein, im Vergleich zu anderen Programmen ist dies etwas gewöhnungsbedürftig. Zuerst geben Sie die Nummer der ersten Zeile der Liste ein, dann ein Komma, die Nummer der letzten Zeile und schließlich das »l«:

```
1,3 l
```

Nun gehen Sie daran, die Datei zu ändern. Am Dateiende oder zwischen zwei bestehenden Zeilen fügen Sie neue Zeilen ein, indem Sie die Nummer der neuen Zeile und das Edlin-Kommando »i« (»insert«, englisch für »einfügen«) ohne Trennzeichen eingeben. Wollen Sie den neuen Text vor Zeile 3 einfügen, tippen Sie also

```
3 i
```

ein. Die nachfolgenden Zeilen nummeriert Edlin automatisch neu. Um eine Zeile am Ende der Datei einzufügen, müssen Sie als Zeilennummer vor der Einfügeanweisung eine Nummer angeben, die auf die Zeilennummer der letzten Dateizeile folgt. Die Angabe von »999« genügt (sofern Ihre Da-

tei weniger Zeilen hat), Edlin nummeriert anschließend richtig durch. Die zu bearbeitende Zeile steht nun eingerückt am Bildschirm und ist bereit für Eingaben. Geben Sie

```
path Laufwerk: \ Verzeichnisname
```

ein, in diesem Beispiel also »path c: \ dos«. Aber Achtung, die Enter-Taste beendet nicht die Editierfunktion: Edlin ist noch bereit, weitere Zeilen an dieser Stelle in die Datei einzufügen. Es erscheint eine leere Zeile mit der nächstfolgenden Zeilennummer. Um die Editierfunktion zu beenden, geben Sie in dieser neuen Zeile [Ctrl-Z] ein und drücken anschließend [Enter].

Um einen Text zu ändern, geben Sie einfach die Zeilennummer ein und arbeiten mit den von MS-DOS gewohnten Funktionstasten

Edlin erwartet jetzt wieder ein Kommando. Mit einem »l« können Sie sich die Änderung in der Liste ansehen. Wenn Sie neuen Text eingefügt haben, sehen Sie, daß Edlin alle Zeilen neu durchnummeriert hat. Daß dies auch Tücken hat, werden Sie beim Löschen von Zeilen noch sehen. Bestehende Zeilen ändern Sie, indem Sie einfach deren Zeilennummer angeben. Geben Sie die Zeilennummer als Kommando ein, und drücken Sie die Enter-Taste. Auf dem Bildschirm sehen Sie eingerückt die alte Zeile und direkt darunter mit derselben Zeilennummer die neue, noch unbeschriebene Zeile. Sie dient nun zur Eingabe der neuen Version der Zeile. Hier gelten alle Funktionstasten, die Sie auch in der MS-DOS-Kommandozeile verwenden: [F3], um die ganze

Die Edlin-Befehle auf einen Blick

Befehl	Abkürzung für	Funktion
a	append	Teile einer Datei in den Arbeitsspeicher lesen
c	copy	Kopieren einer oder mehrerer Zeilen
d	delete	Zeilen löschen*
e	end	Ende mit Dateispeichern
i	insert	Zeile einfügen
l	list	Anzeigen der Zeilen
m	move	Umstellen eines Textabschnitts
p	page	Seitenweise blättern in Schritten von jeweils 23 Zeilen
q	quit	Ende ohne Dateispeichern
r	replace	Ersetzen von Text
s	search	Suchen von Text
t	transfer	Einfügen von Inhalten einer anderen Datei in die aktuelle Datei
w	write	Datei unter anderem Namen speichern

Zeile zurückzuholen, [F1], um einen einzelnen Buchstaben aus der vorhergehenden Zeile zu übernehmen und so weiter. Soll ans Ende der Zeile etwas angefügt werden, drücken Sie die F3-Taste und fügen dann die Erweiterung an, zum Beispiel »; \data« für einen weiteren Pfad. Die endgültige Zeile hieße dann für dieses Beispiel »path c: \dos; \data«. Falls Sie etwas in der Mitte ergänzen wollen, betätigen Sie die F2-Taste und den Buchstaben, der auf die Einfügeposition folgt. Der Text der oberen Zeile erscheint dann bis zu diesem Punkt in der unteren. Drücken Sie die Einfügetaste [Ins], geben Sie dann den gewünschten Text ein und schließen Sie mit der F3-Taste.

Die Eingabe beendet ein [Enter]. Der Cursor ist jetzt nicht mehr eingerückt, in der jetzt aktiven zweiten Spalte des Bildschirms geben Sie wieder Edlin-Kommandos ein.

Löschbefehl mit Tücken – die zweite, dritte und vierte Zeile bedeutet dreimal die zweite Zeile

Zum Beenden des Editiervorgangs dient das Kommando »e«. Dieser Befehl speichert vor dem Beenden noch alle Änderungen, und anschließend zeigt der Rechner wieder die gewohnte Kommandozeile von MS-DOS. Sehen Sie sich nun mit

```
type autoexec.bat
```

die geänderte Datei an. Sie haben jetzt gesehen, wie man mit Edlin neue Zeilen einfügt und bereits bestehende Zeilen ändert. Die eben eingefügte Zeile dürfen Sie natürlich auch wieder löschen. Das Kommando »d« (»delete«, englisch für »löschen«) erledigt das. Rufen Sie wieder »edlin autoexec.bat« auf, und lassen Sie sich die Datei mit »l« anzeigen. Das ist vor dem Löschen immer ratsam, weil Sie sonst womöglich die falsche Zeile löschen.

Der Löschbefehl »d« hat das gleiche Eingabeformat wie der Einfügebefehl »i«. Geben Sie die Zeilennummer der Zeile ein, die die Pfadanweisung enthält, und fügen Sie den Buchstaben »d« für die Löschfunktion an. Fordern Sie mit »l« wieder die Liste an. Alle nachfolgenden Zeilen werden eine Nummer nach unten gestuft. Hier liegt eine große Tücke des Löschbefehls: Sehen Sie eine Datei mit Zeilennummern auf dem Bildschirm, geben Sie womöglich die Reihenfolge

```
2d
3d
4d
```

Beachten Sie das Ergebnis der Aktion: Hier wurde der häufigste Edlin-Fehler beigegeben

```
C:\edlin autoexec.bat
Ende der Eingabedatei!
```

```
*1
1: *PATH C:\;C:\DOS;c:\batch;c:\tools
2: KEYB GR 437 C:\DOS\KEYBOARD.SYS
3: ECHO OFF
4: mouse4
5: prompt $p$g
6: cls
7: VER
```

```
*3d
*4d
*5d
*1
```

```
1: PATH C:\;C:\DOS;c:\batch;c:\tools
2: KEYB GR 437 C:\DOS\KEYBOARD.SYS
3: mouse4
4: cls
```

ein, um die Zeilen 2, 3 und 4 zu löschen. Das ist natürlich falsch, denn nach jedem Befehl wird neu durchnumeriert. Sie müssen also

```
2d
2d
2d
```

eingeben, um die Zeilen 2 bis 4 zu löschen. Sie können mit Edlin auch mehrere Zeilen gleichzeitig löschen. Dazu trennen Sie die Nummer der ersten und letzten zu löschenden Zeile durch ein Komma und setzen dahinter den Löschbefehl »d«. Für die Zeilen 3 bis 5, sieht das so aus:

```
3,5d
```

Wenn Sie den Editor verlassen wollen, ohne die Änderungen zu speichern, verlassen Sie ihn mit »q« (»quit«, englisch für »aufgeben«). Das Programm versichert sich durch die Abfrage »Editieren abbrechen (J/N)?«, ob Sie Ihre Änderungen wirklich verlieren wollen, oder ob es nur ein Versehen war. Bestätigen Sie mit »j«. Löschen Sie nun diese Übungskopie der Datei »autoexec.bat« mit

```
del autoexec.bat
```

Auch neue Batch-Dateien erzeugen Sie mit Edlin wesentlich komfortabler als mit der Methode »Copy con: Dateiname«. Rufen Sie dazu Edlin auf, und geben Sie einen neuen Dateinamen ein. Da nach dem Löschen »autoexec.bat« nicht mehr existiert, erzeugen Sie so eine neue Datei beispielsweise mit

```
edlin autoexec.bat
```

Der Texteditor meldet sich mit dem Text »Neue Datei«. Geben Sie die Anweisung »i« für »Einfügen« ein, und drücken Sie [Enter]. Sie sehen die Eingabeaufforderung für die erste Textzeile.

Geben Sie die Zeilen einfach wie im folgenden Beispiel nacheinander ein, wobei jede Zeile mit einem [Enter] abgeschlossen wird:

```
ECHO OFF
```

```
KEYB GR
PATH C: \;C: \DOS
DATE
TIME
CLS
VER
```

Danach beenden Sie Ihre Eingaben mit [Ctrl-Z], um den Eingabemodus zu verlassen und in den Befehlsmodus zu gelangen. Dort tippen Sie

```
e
ein und schließen wieder mit [Enter] ab. Wenn Sie Ihre alte Version der Datei wiederverwenden wollen, geben Sie ein: »copy autoexec.kop autoexec.bat«. Jetzt können Sie Ihre eigenen Änderungen an den Dateien vornehmen.
```

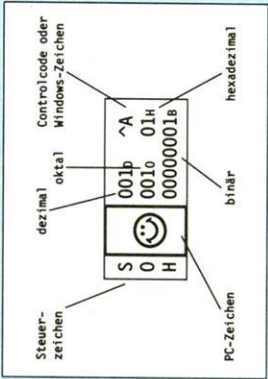
Der MS-DOS-Texteditor ist für manche Zwecke besser geeignet als eine Textverarbeitung

Die wenigen Befehle, die hier nicht erklärt wurden, sind für den Einstieg nicht besonders relevant und auch für den fortgeschrittenen Anwender weniger wichtig. Ihr MS-DOS-Handbuch informiert Sie jedoch bei Bedarf genauer darüber. Doch bereits nach dieser kurzen Einführung verstehen Sie sicher die Originaldokumentation viel besser. Ist Ihnen erst einmal die Arbeitsweise von Edlin vertraut, werden Sie – außer mit Programmen wie dem Norton Commander – kaum etwas finden, das so gut für die schnelle Bearbeitung von Batch-Dateien geeignet ist. Schließlich ist Edlin kurz, flink und gar nicht so schwer zu bedienen, wie es zuerst den Anschein hat. Die Textverarbeitung ist eben doch zu langwierig für manche Zwecke – ein Hoch auf den Purismus von MS-DOS.

(Wolfgang Adis/mk)

ASCII-Zeichentabelle

N	U	L	000b 000b 00H 00000000b	~@ 000b 00H 00000000b		032b 040b 20H 00100000b	@	064b 100b 40H 01000000b	'	096b 140b 60H 01100000b	Ç	128b 200b 80H 10000000b	à	160b 240b A0H 10100000b	L	192b 300b C0H 11000000b	α	224b 340b E0H 11100000b
S	O	H	001b 001b 01H 00000001b	^A 001b 01H 00000001b	!	033b 041b 21H 00100001b	A	065b 101b 41H 01000001b	a	097b 141b 61H 01100001b	ü	129b 201b 81H 10000001b	í	161b 241b A1H 10100001b	T	193b 301b C1H 11000001b	β	225b 341b E1H 11100001b
S	T	X	002b 002b 02H 00000010b	^B 002b 02H 00000010b	"	034b 042b 22H 00100010b	B	066b 102b 42H 01000010b	b	098b 142b 62H 01100010b	ë	130b 202b 82H 10000010b	ô	162b 242b A2H 10100010b	T	194b 302b C2H 11000010b	Γ	226b 342b E2H 11100010b
E	T	X	003b 003b 03H 00000011b	^C 003b 03H 00000011b	#	035b 043b 23H 00100011b	C	067b 103b 43H 01000011b	c	099b 143b 63H 01100011b	â	131b 203b 83H 10000011b	ú	163b 243b A3H 10100011b	F	195b 303b C3H 11000011b	π	227b 343b E3H 11100011b
E	O	T	004b 004b 04H 00000100b	^D 004b 04H 00000100b	\$	036b 044b 24H 00100100b	D	068b 104b 44H 01000100b	d	100b 144b 64H 01100100b	ä	132b 204b 84H 10000100b	ñ	164b 244b A4H 10100100b	—	196b 304b C4H 11000100b	Σ	228b 344b E4H 11100100b
E	N	Q	005b 005b 05H 00000101b	^E 005b 05H 00000101b	%	037b 045b 25H 00100101b	E	069b 105b 45H 01000101b	e	101b 145b 65H 01100101b	ã	133b 205b 85H 10000101b	ã	165b 245b A5H 10100101b	+	197b 305b C5H 11000101b	σ	229b 345b E5H 11100101b
A	C	K	006b 006b 06H 00000110b	^F 006b 06H 00000110b	&	038b 046b 26H 00100110b	F	070b 106b 46H 01000110b	f	102b 146b 66H 01100110b	ä	134b 206b 86H 10000110b	a	166b 246b A6H 10100110b	F	198b 306b C6H 11000110b	μ	230b 346b E6H 11100110b
B	E	L	007b 007b 07H 00000111b	^G 007b 07H 00000111b	'	039b 047b 27H 00100111b	G	071b 107b 47H 01000111b	g	103b 147b 67H 01100111b	ó	135b 207b 87H 10000111b	o	167b 247b A7H 10100111b		199b 307b C7H 11000111b	τ	231b 347b E7H 11100111b
B	S		008b 010b 08H 00001000b	^H 010b 08H 00001000b	(	040b 050b 28H 00101000b	H	072b 110b 48H 01001000b	h	104b 150b 68H 01101000b	é	136b 210b 88H 10001000b	?	168b 250b A8H 10101000b		200b 310b C8H 11001000b	Φ	232b 350b E8H 11101000b
H	T		009b 011b 09H 00001001b	^I 011b 09H 00001001b	)	041b 051b 29H 00101001b	I	073b 111b 49H 01001001b	i	105b 151b 69H 01101001b	ë	137b 211b 89H 10001001b	l	169b 251b A9H 10101001b		201b 311b C9H 11001001b	θ	233b 351b E9H 11101001b
L	F		010b 012b 0Ah 00001010b	^J 012b 0Ah 00001010b	*	042b 052b 2Ah 00101010b	J	074b 112b 4Ah 01001010b	j	106b 152b 6Ah 01101010b	è	138b 212b 8Ah 10001010b	l	170b 252b AAh 10101010b		202b 312b CAH 11001010b	Ω	234b 352b EAH 11101010b
V	T		011b 013b 0BH 00001011b	^K 013b 0BH 00001011b	+	043b 053b 2BH 00101011b	K	075b 113b 4BH 01001011b	k	107b 153b 6BH 01101011b	ï	139b 213b 8BH 10001011b	1/2	171b 253b ABH 10101011b		203b 313b CBH 11001011b	δ	235b 353b EBH 11101011b
F	F		012b 014b 0CH 00001100b	^L 014b 0CH 00001100b	,	044b 054b 2CH 00101100b	L	076b 114b 4CH 01001100b	l	108b 154b 6CH 01101100b	î	140b 214b 8CH 10001100b	1/4	172b 254b ACH 10101100b		204b 314b CCH 11001100b	∞	236b 354b ECH 11101100b
C	R		013b 015b 0DH 00001101b	^M 015b 0DH 00001101b	-	045b 055b 2DH 00101101b	M	077b 115b 4DH 01001101b	m	109b 155b 6DH 01101101b	ï	141b 215b 8DH 10001101b	i	173b 255b ADH 10101101b	=	205b 315b CDH 11001101b	φ	237b 355b EDH 11101101b
S	O		014b 016b 0EH 00001110b	^N 016b 0EH 00001110b	.	046b 056b 2EH 00101110b	N	078b 116b 4EH 01001110b	n	110b 156b 6EH 01101110b	Ä	142b 216b 8EH 10001110b	«	174b 256b AEH 10101110b		206b 316b CEH 11001110b	€	238b 356b EEH 11101110b
S	I		015b 017b 0FH 00001111b	^O 017b 0FH 00001111b	/	047b 057b 2FH 00101111b	O	079b 117b 4FH 01001111b	o	111b 157b 6FH 01101111b	Å	143b 217b 8FH 10001111b	»	175b 257b AFH 10101111b		207b 317b CFH 11001111b	∩	239b 357b EFH 11101111b



196	197	198	199
—		⌋	⌋
218	194	197	217
⌋	⌋	+	⌋
195	192		
⌋	⌋		

205	186	187
=		⌋
201	203	
⌋	⌋	

D	0160	0200	00010000b	▲	0160	0200	00010000b	^P	0	0480	0600	00110000b	0	P	0800	1200	01010000b	p	1120	1600	01110000b	ë	1440	2200	00100000b	■	1440	2200	00100000b	...	1760	2600	10110000b	°	1760	2600	10110000b	°	2400	3600	F0h	ð
L	0170	0210	00010001b	▼	0170	0210	00010001b	^Q	1	0490	0610	00110001b	1	Q	0810	1210	01010001b	q	1130	1610	01110001b	æ	1450	2210	00100001b	◼	1450	2210	00100001b	▣	1770	2610	10110001b	±	1770	2610	10110001b	±	2410	3610	F1h	ñ
E	0180	0220	00010010b	↕	0180	0220	00010010b	^R	2	0500	0620	00110010b	2	R	0820	1220	01010010b	r	1140	1620	01110010b	æ	1460	2220	00100010b	▤	1460	2220	00100010b	▥	1780	2620	10110010b	²	1780	2620	10110010b	²	2420	3620	F2h	ö
	0190	0230	00010011b	!!	0190	0230	00010011b	^S	3	0510	0630	00110011b	3	S	0830	1230	01010011b	s	1150	1630	01110011b	ö	1470	2230	00100011b	▦	1470	2230	00100011b	▧	1790	2630	10110011b	³	1790	2630	10110011b	³	2430	3630	F3h	ó
C	0200	0240	00010100b	⌂	0200	0240	00010100b	^T	4	0520	0640	00110100b	4	T	0840	1240	01010100b	t	1160	1640	01110100b	ö	1480	2240	00100100b	▨	1480	2240	00100100b	▩	1800	2640	10110100b	⁴	1800	2640	10110100b	⁴	2440	3640	F4h	ô
C	0210	0250	00010101b	§	0210	0250	00010101b	^U	5	0530	0650	00110101b	5	U	0850	1250	01010101b	u	1170	1650	01110101b	ö	1490	2250	00100101b	▪	1490	2250	00100101b	▫	1810	2650	10110101b	μ	1810	2650	10110101b	μ	2450	3650	F5h	õ
A	0220	0260	00010110b	■	0220	0260	00010110b	^V	6	0540	0660	00110110b	6	V	0860	1260	01010110b	v	1180	1660	01110110b	ü	1500	2260	00100110b	▬	1500	2260	00100110b	▭	1820	2660	10110110b	¶	1820	2660	10110110b	¶	2460	3660	F6h	ö
Y	0230	0270	00010111b	↕	0230	0270	00010111b	^W	7	0550	0670	00110111b	7	W	0870	1270	01010111b	w	1190	1670	01110111b	ü	1510	2270	00100111b	▮	1510	2270	00100111b	▯	1830	2670	10110111b	⌘	1830	2670	10110111b	⌘	2470	3670	F7h	■
N	0240	0300	00011000b	↑	0240	0300	00011000b	^X	8	0560	0700	00111000b	8	X	0880	1300	01011000b	x	1200	1700	01111000b	ÿ	1520	2300	00100000b	▰	1520	2300	00100000b	▱	1840	2700	10111000b	⌘	1840	2700	10111000b	⌘	2480	3700	F8h	ø
E	0250	0310	00011001b	↓	0250	0310	00011001b	^Y	9	0570	0710	00111001b	9	Y	0890	1310	01011001b	y	1210	1710	01111001b	ÿ	1530	2310	00100001b	▲	1530	2310	00100001b	△	1850	2710	10111001b	⌘	1850	2710	10111001b	⌘	2490	3710	F9h	ù
S	0260	0320	00011010b	→	0260	0320	00011010b	^Z	:	0580	0720	00111010b	:	Z	0900	1320	01011010b	z	1220	1720	01111010b	ÿ	1540	2320	00100100b	▴	1540	2320	00100100b	▵	1860	2720	10111010b	⌘	1860	2720	10111010b	⌘	2500	3720	FAh	ú
E	0270	0330	00011011b	←	0270	0330	00011011b	^[	;	0590	0730	00111011b	;	[	0910	1330	01011011b	{	1230	1730	01111011b	ÿ	1550	2330	00100101b	▶	1550	2330	00100101b	▷	1870	2730	10111011b	⌘	1870	2730	10111011b	⌘	2510	3730	F Bh	û
S	0280	0340	00011100b	L	0280	0340	00011100b	^\	<	0600	0740	00111100b	<	\	0920	1340	01011100b	<	1240	1740	01111100b	£	1560	2340	00100000b	▸	1560	2340	00100000b	▹	1880	2740	10111000b	¼	1880	2740	10111000b	¼	2520	3740	FCh	ü
G	0290	0350	00011101b	↔	0290	0350	00011101b	^]	=	0610	0750	00111101b	=	]	0930	1350	01011101b	}	1250	1750	01111101b	¥	1570	2350	00100101b	▹	1570	2350	00100101b	►	1890	2750	10111001b	½	1890	2750	10111001b	½	2530	3750	F Dh	ý
R	0300	0360	00011110b	◀	0300	0360	00011110b	^^	>	0620	0760	00111110b	>	^	0940	1360	01011110b	~	1260	1760	01111110b	Pl	1580	2360	00100110b	▻	1580	2360	00100110b	▻	1900	2760	10111100b	¾	1900	2760	10111100b	¾	2540	3760	F Eh	þ
U	0310	0370	00011111b	▶	0310	0370	00011111b	^_	?	0630	0770	00111111b	?	_	0950	1370	01011111b	°	1270	1770	01111111b	f	1590	2370	00100111b	▹	1590	2370	00100111b	▹	1910	2770	10111111b	⅞	1910	2770	10111111b	⅞	2550	3770	F Fh	ÿ

204	1F	206	1F	185	1F
200	1F	202	1F	188	1F

214	1F	210	1F	183	1F
199	1F	215	1F	182	1F
211	1F	208	1F	189	1F

213	1F	209	1F	184	1F
198	1F	216	1F	181	1F
212	1F	207	1F	190	1F

Ansi-Steuersequenzen Bildschirm und Tastatur im Griff

Weder Programmiersprachen- noch gar Assembler-Kenntnisse sind notwendig, wenn Sie die Bildschirmdarstellung unter MS-DOS ein wenig ansprechender und farbiger gestalten möchten. Der Gerätetreiber »ansi.sys« stellt Ihnen die entsprechenden Funktionen zur Verfügung.

Zusammen mit dem Betriebssystem MS-DOS erhält der Anwender neben verschiedenen Dienst- und Hilfsprogrammen noch einige zusätzliche Treiber (zum Beispiel »ansi.sys«, »vdisk.sys«, »driver.sys«, »display.sys« und »printer.sys«) geliefert. Dabei handelt es sich um Maschinenprogramme, welche die Kommunikation zwischen MS-DOS und dem entsprechenden Gerät regeln. Die Treiber repräsentieren also die Schnittstelle zwischen dem Gerät und dem Betriebssystem.

Gerade der PC-Neuling hat in den meisten Fällen nicht den »blassesten Schimmer« einer Ahnung, welche Treiber er vorliegen hat, wozu er sie einsetzen kann und welche Treiber nach dem Einschalten seines Computers bereits aktiviert sind. Da allen Gerätetreibern die Dateikennung »sys« gemeinsam ist, können Sie mit dem Dir-Befehl sehr einfach feststellen, welche Treiber Sie besitzen.

```
dir *.sys
```

nacheinander auf alle MS-DOS-Systemdisketten angewandt, schafft hier schnell Klarheit. Auch die Frage, welche Treiber beim Systemstart (Boot-Vorgang) aktiviert wurden, läßt sich leicht klären. Sie müssen sich dazu nur den Inhalt der Konfigurationsdatei »config.sys« ansehen, die sich – sofern vorhanden – im Hauptverzeichnis Ihrer Festplatte oder Boot-Diskette befindet. In ihr ist die Systemkonfiguration Ihres Rechners definiert, vor allem das Arbeitsumfeld des Betriebssystems. Sie ist übrigens die einzige Datei, die trotz der Kennung »sys« kein Maschinenprogramm darstellt, sondern eine reine Textdatei (ohne irgendwelche Steuerzeichen). Die Ausführung der Anweisung

```
type config.sys
```

liefert Ihnen den Inhalt der Datei »config.sys«, der folgendermaßen aussehen könnte:

```
shell=c:\command.com /p
buffers=20
files=30
device=c:\ansi.sys
```

Der letzte Eintrag sagt Ihnen in diesem Fall, daß Ihr Computer die Ansi-Steuersequenzen unterstützt, da der Gerätetreiber »ansi.sys« eingebunden

wurde. »ansi.sys« ist für die Steuerung der Bildschirmausgabe verantwortlich. Der Name »Ansi« leitet sich von »American National Standard Institute« ab. Dies kommt der Ihnen bekannten DIN (Deutsche Industrienorm) gleich.

Die Ansi-Steuersequenzen (Folge von Steuerbefehlen) erlauben Ihnen die Kontrolle über die Cursor-Position, den Darstellungsmodus, die Farbgestaltung und nicht zuletzt die Tastaturzuordnung (Tabelle 1 und 2). Es ist empfehlenswert, »ansi.sys« in jedem Falle zu installieren, selbst wenn Sie die Funktionen nicht nutzen wollen. Sie beugen damit anstrengenden Überlegungen hinsichtlich der Fehlerquelle(n) bei Anwendungsprogrammen vor, die den Ansi-Treiber benötigen.

Mit den Ansi-Steuersequenzen bringen Sie (nicht nur) Farbe in das triste Programmiererleben

Vergessen Sie beim Installieren des Treibers (durch einen Eintrag in die Datei »config.sys«) jedoch nicht, den vollständigen Zugriffspfad für den Ansi-Treiber anzugeben, sofern er sich nicht im Hauptverzeichnis Ihrer Festplatte (oder Boot-Diskette) befindet. Der Eintrag in die Datei »config.sys« muß daher

```
device=c:\drv\ansi.sys
```

lauten, wenn der Treiber im Unterverzeichnis »drv« des Laufwerks C vorliegt.

Wie kommen Sie aber nun in den Genuß der Fähigkeiten des Ansi-Treibers? Wie Sie bemerken, werden die Steuersequenzen jeweils durch die beiden Zeichen »ESC« und »[« eingeleitet. »ESC« bezeichnet hier nicht etwa die Esc-Taste, sondern den ASCII-Wert 27, der sich nicht einfach durch einen Druck auf die gleichnamige Taste darstellen läßt. Gesetzt den Fall, Sie möchten, daß Ihnen die Auflistung Ihrer Dateien nach dem nächsten Dir-Kommando türkis auf blauem Hintergrund entgegenstrahlt: Mit Hilfe der Tabelle 1 stellen Sie die

erforderliche Ansi-Sequenz »ESC [36;44m« zusammen. In Turbo Pascal müssen Sie das in folgende Programmzeilen umformen:

```
begin
  writeln( # 27' [36;44m' )
end.
```

Zu beachten ist hierbei, daß Sie die ab Version 4.0 vorhandene Turbo-Pascal-Unit »Crt« nicht mit der Uses-Anweisung in das Programm einbinden dürfen; infolge direkter Zugriffe auf den Bildschirmspeicher würde die Steuersequenz dann nämlich unwirksam bleiben. Auch in andere Programmiersprachen lassen sich die Ansi-Sequenzen leicht übertragen:

```
print chr$(27); "[30;47m"
```

Mit dieser Basic-Zeile weisen Sie den Treiber an, Ihre Bildschirmausgaben zukünftig »schwarz auf weiß« darzustellen. Wenn Sie das Beispiel allerdings in GW-Basic ausprobieren, werden Sie enttäuscht – denn es tut sich nichts. Der Grund für den mißglückten Versuch liegt in der Tatsache, daß GW-Basic seine Bildschirmausgaben über eigene Programmerroutinen abwickelt und somit der Ansi-Treiber gar nicht zum Einsatz kommt.

Glücklicherweise funktioniert das Spiel mit den Ansi-Sequenzen auch gänzlich ohne Programmiersprache. Das Betriebssystem MS-DOS stellt Ihnen die notwendigen Mittel zur Verfügung: den Prompt-Befehl. Mit dem Wissen, daß die Zeichenfolge »\$e« innerhalb der Prompt-Anweisung den ASCII-Wert 27 (also das Esc-Zeichen) repräsentiert, erfüllen Sie bereits alle Voraussetzungen, um mit den Ansi-Sequenzen arbeiten zu können.

```
prompt $e[30;47m$P$g
```

führt dazu, daß die Bildschirmdarstellung in Zukunft »schwarz auf weiß« geschieht.

Die Ansi-Steuersequenzen lassen sich natürlich auch beliebig kombinieren. Gönnen Sie Ihren Augen daher doch einmal mehr als nur eine Farbkombination.

```
prompt $e[31m$P$g$e[32m
```

sorgt beispielsweise dafür, daß die Eingabeaufforderung in Rot und Ihre Eingaben und anschließende Bildschirmausgaben in Grün dargestellt werden. Eine weitere – nicht nur für Schreibfaule – sehr interessante Funktion des Ansi-Treibers besteht darin, die Tastaturbelegung zu ändern beziehungsweise neu zu definieren (Tabelle 2).

Mit der Anweisung

```
prompt $e[0;68;"dir /p";13p
```

legen Sie fest, daß das Dateiverzeichnis des aktuellen Laufwerks angezeigt wird, sobald Sie die Funktions-

Cursor-Position (CUP) ESC [#1;#2H	CUP bewegt den Cursor an die durch #1 (Zeile) und #2 (Spalte) beschriebene Stelle des Bildschirms.
Cursor positionieren (HVP) ESC [#1;#2f	Der Befehl HVP entspricht CUP.
Cursor nach oben (CUU) ESC [#A	Dieser Befehl bewegt den Cursor # Zeilen nach oben, ohne die Spalte zu verlassen. Steht der Cursor bereits in der ersten Zeile, so wird dieser Befehl ignoriert.
Cursor nach unten (CUD) ESC [#B	CUD bewegt den Cursor # Zeilen nach unten, ohne die Spalte zu verlassen. Wird ignoriert, falls der Cursor bereits in der untersten Zeile steht.
Cursor rechts (CUF) ESC [#C	Der Befehl CUF bewegt den Cursor # Spalten nach rechts, ohne die Zeilenposition zu ändern. Der Befehl wird ignoriert, wenn der Cursor bereits in der äußersten rechten Spalte steht.
Cursor nach links (CUB) ESC [#D	CUB bewegt den Cursor # Spalten nach links, ohne die Zeilenposition zu verändern. Wird ignoriert, falls sich der Cursor bereits in der äußersten linken Spalte befindet.
Cursor-Position abfragen (DSR) ESC [6n	Fragt die momentane Cursor-Position am Bildschirm ab. Die Ausgabe erfolgt automatisch durch die Anweisung CPR. Diese Funktion sollte nicht mit dem Befehl »prompt« verwendet werden!
Cursor-Position zeigen (CPR) ESC [#1;#2R	Dieser Befehl zeigt die aktuelle Cursor-Position an, wobei #1 die Zeile und #2 die Spalte bezeichnet.
Cursor-Position speichern (SCP) ESC [s	Speichert die aktuelle Cursor-Position. Die Werte können mit dem Befehl »RCP« zurückgeholt, und der Cursor kann damit an die ursprüngliche Stelle zurückgeschrieben werden.
Cursor an die gespeicherte Position setzen (RCP) ESC [u	Setzt den Cursor an die Stelle, die mit SCP gespeichert wurde.
Bildschirm löschen (ED) ESC [2J	Entspricht dem MS-DOS-Befehl »cls«
Zeile löschen (EL) ESC [K	EL löscht die restliche Zeile ab der momentanen Cursor-Position.
Farbwahl (SGR) ESC [#;...;#m	Setzt Farb- und/oder Darstellungsform für die Zeichen auf dem Bildschirm auf bestimmte Werte. Die Variablen »#« stehen für die verfügbaren Darstellungsarten.
Wert(#) Farbe/Funktion 	

Tabelle 1. Die Ansi-Steuersequenzen für die Bildschirmansteuerung

taste [F10] betätigen. Die Zahl 13 hinter der Zeichenkette entspricht dem ASCII-Wert der Enter-Taste und sorgt dafür, daß der Dir-Befehl nicht nur auf dem Bildschirm erscheint, sondern auch sofort – ohne eine weitere Tastenbetätigung – ausgeführt wird. Nach diesem Muster können Sie die von Ihnen am meisten genutzten MS-DOS-Befehle auf einzelne Tasten oder Tastenkombinationen legen und sich damit eine Menge Tipparbeit sparen. Der Möglichkeiten gibt es viele. (zi)

Tastaturcode				
Taste	alleine	mit [Shift]	mit [Ctrl]	mit [Alt]
[F1]	0;59	0;84	0;94	0;104
[F2]	0;60	0;85	0;95	0;105
[F3]	0;61	0;86	0;96	0;106
[F4]	0;62	0;87	0;97	0;107
[F5]	0;63	0;88	0;98	0;108
[F6]	0;64	0;89	0;99	0;109
[F7]	0;65	0;90	0;100	0;110
[F8]	0;66	0;91	0;101	0;111
[F9]	0;67	0;92	0;102	0;112
[F10]	0;68	0;93	0;103	0;113
[Home]	0;71	55	119	–
[Cursor aufwärts]	0;72	56	–	–
[PgUp]	0;73	57	132	–
[Cursor links]	0;75	52	115	–
[Cursor rechts]	0;77	54	116	–
[End]	0;79	49	117	–
[Cursor abwärts]	0;80	50	–	–
[PgDn]	0;81	51	118	–
[Ins]	0;82	48	–	–
[Del]	0;83	44	–	–
[A]	97	65	1	0;30
[B]	98	66	2	0;48
[C]	99	67	3	0;46
[D]	100	68	4	0;32
[E]	101	69	5	0;18
[F]	102	70	6	0;33
[G]	103	71	7	0;34
[H]	104	72	8	0;35
[I]	105	73	9	0;23
[J]	106	74	10	0;36
[K]	107	75	11	0;37
[L]	108	76	12	0;38
[M]	109	77	13	0;50
[N]	110	78	14	0;49
[O]	111	79	15	0;24
[P]	112	80	16	0;25
[Q]	113	81	17	0;16
[R]	114	82	18	0;19
[S]	115	83	19	0;31
[T]	116	84	20	0;20
[U]	117	85	21	0;22
[V]	118	86	22	0;47
[W]	119	87	23	0;17
[X]	120	88	24	0;45
[Y]	121	89	25	0;21
[Z]	122	90	26	0;44
[1]	49	33	–	0;120
[2]	50	34	0;3	0;121
[3]	51	35	–	0;122
[4]	52	36	–	0;123
[5]	53	37	–	0;124
[6]	54	38	30	0;125
[7]	55	39	–	0;126
[8]	56	40	–	0;127
[9]	57	41	–	0;128
[0]	48	61	–	0;129

Tabelle 2. Ansi-Steuersequenz zur Tastaturbelegung mit zugehörigen Tastaturcodes

Bild 1 zeigt den Registersatz des 8086/8088. Beide Prozessoren sind mit 14 16-Bit-Registern ausreichend

bestückt, so daß sich auch ohne Speichierzugriffe komplexe Operationen direkt im Prozessor ausführen lassen.

Die Register des 8086/8088 teilen sich in vier Gruppen auf. Jedes Register hat eine spezielle Aufgabe, doch können die meisten Register bei Bedarf auch universell eingesetzt werden. Vier der 16-Bit-Register lassen sich auch in 8 Bit breite Register unterteilen, für den Fall, daß nur mit Zahlen bis 255 ($2^8 - 1$) gearbeitet wird.

Die erste Gruppe der Register ist für alle Arten von Daten zuständig. Das AX-Register (englisch: »Accumulator«) setzt sich aus den beiden 8-Bit-Teilregistern AL und AH zusammen. AX ist der Akkumulator des 8086/8088, der primär für arithmetische und logische Operationen zuständig ist. Das BX-Register (englisch: »Base Register«) besteht aus den beiden 8-Bit-Teilregistern BL und BH. BX ist ein Register für Basisdaten und -adressen, das mit dem HL-Register des legendären Z80 vergleichbar ist.

Das CX-Register (englisch: »Count Register«) setzt sich aus den beiden 8-Bit-Teilregistern CL und CH zusammen. CX ist das Zählerregister, das für Schleifen und sonstige periodische Abläufe verwendet wird. Das DX-Register (englisch: »Data Register«) besteht aus den beiden 8-Bit-Teilregistern DL und DH. DX ist das Datenregister, das entweder Daten für arithmetische und logische Befehle, für Speicher- und Ein-/Ausgabeoperationen oder eine Port-Adresse enthält (Port-Adressen sind das Äquivalent zu den Speicheradressen für den Zugriff des Prozessors auf die Peripherie.)

User Stack: vom Anwender eingerichtet, zusätzlicher Stapel für eigene Programme

Die zweite Gruppe der 8086/8088-Register ist in erster Linie für die Aufnahme von Adressen vorgesehen. Der Stapelzeiger SP (englisch: »Stack Pointer«) enthält die aktuelle Adresse des obersten Stapelwertes. Auf einem Stapel (Stack) werden hauptsächlich die Rücksprungadressen bei Unterprogrammaufrufen gespeichert, damit der Prozessor nach einem Return-Befehl weiß, zu welchem Befehl er zurückkehren muß. Der Basiszeiger BP (englisch: »Base Pointer«) zeigt auf den Anfang oder das Ende einer Datenstruktur, zum Beispiel einer Tabelle, und wird von den Programmie-

ren auch oft als Stapelzeiger für eigene Programme (User Stack) benutzt.

Das Quellregister SI (englisch: »Source Index«) gibt die Quelladresse und das Zielregister DI (englisch: »Destination Index«) die Zieladresse für eine Operation an. Diese beiden Register werden hauptsächlich bei Zeichenkettenbefehlen eingesetzt, in denen das SI-Register auf die alte, noch zu bearbeitende und das DI-Register auf die neue, zu erzeugende Zeichenkette zeigt.

Die dritte Gruppe der 8086/8088-Register sind die Segmentregister. Diese Register geben den jeweils gültigen Speicherbereich für die anderen Register sowie den Befehlszeiger an:

- Das Programmsegmentregister CS (englisch: »Code Segment«) ist für den Befehlszeiger IP zuständig,
- das Datensegmentregister DS (englisch: »Data Segment«) für das Quellregister SI und das Zielregister DI,
- das Stapelsegmentregister SS (englisch: »Stack Segment«) für den Stapelzeiger SP und Basiszeiger BP,
- das Extrasegmentregister ES (englisch: »Extra Segment«) für das Zielregister DI bei Zeichenkettenoperationen.

Die vierte und letzte Gruppe des 8086/8088 umfaßt nur zwei Register. Da ist zum einen der Befehlszeiger IP (englisch: »Instruction Pointer«), der auf den jeweils nächsten aus dem Speicher zu lesenden Prozessorbefehl zeigt und zum anderen das Statusregister PSW (englisch: »Processor Status Word«), das wiederum aus den einzelnen Statusbits besteht. Die Statusbits zeigen durch ihren Zustand (gesetzt oder gelöscht) an, ob das Ergebnis einer Arithmetikoperation 0 lautet, ob das Vorzeichen einer Zahl positiv oder negativ ist oder ob beispielsweise bei einer Multiplikation ein Überlauf aufgetreten ist (das heißt, das Ergebnis war zu groß, um in einem 16-Bit-Register gespeichert zu werden).

Um mit dem 8086/8088 eine bestimmte Speicherzelle zu adressie-

ren, existieren insgesamt sieben verschiedene Methoden:

1. Die unmittelbare Adressierung

Bei der unmittelbaren Adressierung wird der Operand direkt im Befehl angegeben.

Beispiel: ADD AX,256
(Addiere den Wert 256 zum AX-Register)

2. Die direkte Adressierung

Bei der direkten Adressierung ist die Adresse des Operanden im Befehl mit angegeben und wird vom Prozessor unter Berücksichtigung der Segmentadresse (in der Regel in DS enthalten) verwertet.

Beispiel: MOV CX,[256]
(Lade den Wert aus der Speicherzelle 256 in das CX-Register)

Zum Zeichen dafür, daß ein Operand nicht den Wert direkt, sondern nur die Adresse dieses Wertes angibt, ist der Operand in eckige Klammern eingeschlossen.

3. Die indirekte Adressierung

Bei der indirekten Adressierung enthält das Register BX, BP, SI oder DI die Adresse des Operanden.

Beispiel: MOV DX,[BX]
(Lade den Inhalt der durch das BX-Register adressierten Speicherzelle in das DX-Register)

Zum Zeichen dafür, daß ein Register nicht den Wert direkt, sondern nur die Adresse dieses Wertes angibt, ist das Register in eckige Klammern eingeschlossen.

4. Die Basisadressierung

Bei der Basisadressierung enthält das Basisregister BX oder BP die Basisadresse des Operanden, zu der ein Zusatzwert (Offset) addiert wird, der im Befehl angegeben sein muß.

Beispiel: MOV DX,[BX+256]
(Lade den Inhalt der durch das BX-Register zuzüglich des Wertes 256 adressierten Speicherzelle in das DX-Register)

5. Die indizierte Adressierung

Bei der indizierten Adressierung enthält das Indexregister SI oder DI die Basisadresse des Operanden, zu der ein Zusatzwert (Offset), der im Befehl angegeben sein muß, addiert wird.

Beispiel: MOV DX,[SI+256]
(Lade den Inhalt der durch das SI-Register zuzüglich des Wertes 256 adressierten Speicherzelle in das DX-Register)

Die Adressierungsarten des 8086/8088

Adressierungsart	Berechnung der effektiven Adresse (EA)
Unmittelbar	Operand steht im Befehl
Direkt	EA = Offset
Indirekt	EA = [BX] oder [BP] oder [SI] oder [DI]
Basis	EA = ([BX] oder [BP]) + Offset
Indiziert	EA = ([SI] oder [DI]) + Offset
Indiziert Basis	EA = ([BX] oder [BP]) + ([SI] oder [DI])
Indiziert Basis mit Offset	EA = ([BX] oder [BP]) + ([SI] oder [DI]) + Offset

Tabelle 1. Die Adressierungsarten des 8086/8088 im Überblick

6. Die indizierte Basisadressierung

Die indizierte Basisadressierung ist eine Kombination aus der Basisadressierung und der indizierten Adressierung. Hier bildet sich die effektive Adresse des Operanden aus dem Inhalt des Basisregisters BX oder BP, zu dem der Inhalt des Indexregisters SI oder DI addiert wird.

Beispiel: MOV DX, [BX+SI]

(Lade den Inhalt der durch die Summe aus dem BX- und SI-Register adressierten Speicherzelle in das DX-Register)

7. Die indizierte Basisadressierung mit Offset

Diese Adressierung entspricht der indizierten Basisadressierung, nur wird zu der Summe des Basis- und Indexregisterinhalts zusätzlich ein Offset addiert, der im Befehl angegeben sein muß.

Beispiel: MOV DX, [BX+SI+256]

(Lade den Inhalt der durch die Summe aus dem BX- und SI-Register zuzüglich des Wertes 256 adressierten Speicherzelle in das DX-Register)

Tabelle 1 zeigt die sieben Adressierungsarten nochmals im Überblick.

Adreßbus: Verbund von elektrischen Leitungen, die die einzelnen Bits einer Adresse übermitteln

Der 8086/8088-Prozessor besitzt 20 Leitungen zur Adressierung des Speichers, die als Adreßbus bezeichnet werden. Mit den 20 Leitungen des Adreßbusses lassen sich 1048576 (2^{20}) einzelne Speicherzellen adressieren. Diese Anzahl entspricht 1024 KByte (1 KByte = 1024 Byte) oder noch kürzer 1 MByte (1 MByte = 1024 KByte). Aus diesem Grund kann der 8086/8088-Prozessor über seinen Adreßbus 1 MByte Speicher adressieren.

Doch wie bereits erwähnt, sind alle Register des 8086/8088 nur 16 Bit breit und können demzufolge auch nur 65536 Speicherzellen (64 KByte) adressieren. Wie erzeugt der Prozessor nun aus den 16 Bit eines Registers die 20 Bit für den Adreßbus?

Um den vollen Speicherbereich des 8086/8088 adressieren zu können, ist die Mithilfe der Segmentregister erforderlich, die innerhalb des gesamten Speicherbereichs von 1 MByte ein 64 KByte großes Speichersegment definieren, das von den anderen Registern dann komplett angesprochen werden kann.

Eine vollständige 20-Bit-Adresse wird also jeweils aus einem 16-Bit-Wert, der im Prozessorbefehl direkt angegeben oder in einem Nicht-Seg-

Allgemeine Daten- und Adreßregister				
	Bit 31	Bit 16	Bit 15	Bit 0
EAX			AH	AX
EBX			BH	BX
ECX			CH	CX
EDX			DH	DX
ESI			SI	
EDI			DI	
EBP			BP	
ESP			SP	

Bild 2. Die allgemeinen Daten- und Adreßregister wurden beim 80386 auf 32 Bit Breite erweitert

mentregister enthalten ist, und dem 16-Bit-Wert eines Segmentregisters gebildet.

Jedes Segmentregister ist dabei für ein oder mehrere bestimmte Register zuständig. Das Programmsegmentregister CS ist für die Adresse im Befehlszeiger, das Datensegmentregister DS für eine Adresse im Quell- oder Zielregister zuständig, das Stapelsegmentregister SS für die Adressen im Stapel- oder Basiszeiger und das Extrasegmentregister für eine Adresse im Zielregister bei Zeichenkettenoperationen. Diese feste Zuordnung können Sie allerdings durch einen Prozessorbefehl umgehen.

Die effektive Adresse aus einem 16-Bit-Wert in einem Befehl oder Nicht-Segmentregister und dem zugehörigen Segmentregister bildet der Prozessor, indem er zu dem ersten 16-Bit-Wert den mit 16 multiplizierten Wert des Segmentregisters addiert. Dazu zwei Beispiele:

Enthält beispielsweise das SI-Register den Wert 2000 und DS den Wert 1000, so lautet die effektive Adresse

$$2000 + 1000 \times 16 = 18000$$

Enthält der Befehlszeiger IP die Adresse 1000 und CS den Wert 4000, so lautet die effektive Adresse

$$1000 + 4000 \times 16 = 65000$$

Dieses Verfahren ist zwar nicht ganz einfach, bietet aber den Vorteil, daß die Adreßangaben in den Prozessorbefehlen mit einer Länge von 16 Bit überschaubar bleiben und aufgrund der 16-Bit-Struktur des 8086/8088 leichter verarbeitet werden können. Für Verzweigungen, Unterprogrammaufrufe und die Rückkehr aus Unterprogrammen stehen beim 8086/8088 grundsätzlich zwei verschiedene Arten der Adressierung zur Verfügung. Die erste Art der Adressierung wird als Intersegmentadressierung bezeichnet und erlaubt die Adressierung über die Segmentgrenzen hinweg. Deshalb müssen aber auch jeweils eine 16-Bit-Adresse

und eine 16-Bit-Segmentadresse angegeben werden.

Die zweite Art der Adressierung ist die sogenannte Intra-segmentadressierung, die nur eine 16-Bit-Adresse erfordert und sich jeweils auf das aktuelle Speichersegment bezieht.

Der Nachfolger und große Bruder des 8086/8088 ist der 80286-Prozessor, der ebenfalls von Intel stammt und in den Computern der AT-Klasse eingesetzt wird. Aufgrund seiner höheren Leistung gegenüber dem 8086/8088 trägt er in entscheidendem Maße zur größeren Rechengeschwindigkeit dieser Computerklasse bei. Trotzdem ist der 80286 zum 8086/8088 voll kompatibel, so daß Software, die für den 8086/8088 entwickelt wurde, auch mit dem 80286 arbeitet.

Die Neuerungen des 80286 gegenüber dem 8086/8088 sind im wesentlichen, daß der Prozessor neben dem 8086-kompatiblen Modus, der als »Real Mode« bezeichnet wird, eine zweite Betriebsart namens »Protected Mode« kennt, in der sich die Leistung des 80286 voll entfalten kann. Das PC-Betriebssystem MS-DOS verwendet allerdings nur den Real Mode des 80286, so daß in diesem Fall ein Großteil der Fähigkeiten des 80286 ungenutzt bleibt.

Der Protected Mode des 80286 wurde speziell für den Ablauf von Betriebssystemen konzipiert, die mehrere Programme parallel bearbeiten (englisch: »Multitasking«) und mehrere Anwender zugleich bedienen können (englisch: »Multiuser«). Zu diesem Zweck wurde der Adreßbus des 80286 gegenüber dem 8086/8088 um vier Leitungen erweitert, so daß sich damit 16 MByte Speicher adressieren lassen. Zudem wurde – und das ist das Wichtigste – eine Speicherverwaltungseinheit (englisch: »Memory Management Unit«, kurz »MMU«) eingeführt, die eine neue, leistungsfähigere Art der Speicherverwaltung mit verschiedenen Speicherschutzmechanismen erlaubt.

Mit Hilfe der MMU kann der 80286 den gesamten Speicher von 16 MByte adressieren und einzelne Speicherbereiche mit einem Schutz versehen, um Lesezugriffe auf Programmcode und Schreibzugriffe auf Daten zu verhindern. Durch Festlegen einer von vier Privileginstufen läßt sich die Berechtigung noch mehr einschränken, so daß beispielsweise wichtige Programme nicht von jedermann überschrieben und vertrauliche Daten von unberechtigten Anwendern nicht gelesen werden können.

Ein Programm kann jeweils nur auf andere Routinen mit gleicher Privilegstufe und auf Daten mit niedriger oder gleicher Privilegstufe zugreifen. Wird jedem Speicherbereich eine be-

NEU

Wissen ist Macht.

Neu im DMV-Verlag: Computer-Wissen. Ein Magazin, das Ihnen Wissen über Ihren PC vermittelt, das andere nicht haben.

Unsere Themen: Alles, was Ihnen hilft, den PC besser zu nutzen und zu verstehen. Mit DMV-Computer-Wissen steht Ihnen das gesamte Know-how der DMV-Redaktionen in leichtverständlicher Form zur Verfügung.

Aus dem Inhalt:

Programme:

- Deutsche Fehlermeldungen in GW-BASIC
- Datei- und Verzeichniswahl mit Cursortasten
- BASIC-Programme automatisch strukturieren
- Konvertierungsprogramm von GW nach Turbo

Routinen:

- Umfangreiche Berechnungsfunktionen
- Konfiguration feststellen
- Hardcopy programmgesteuert
- Grafikroutinen
- Mausroutinen mit Testprogramm

Specials:

- Kalenderberechnungen
- Wahrscheinlichkeit und Statistik
- Literaturverzeichnis zum Thema BASIC

Und vieles anderes mehr:

Insgesamt über 500 Funktionen!

Damit Sie das Rad nicht ständig neu erfinden müssen, steht Ihnen im ersten Band von DMV-Computer-Wissen eine in Umfang und Vielfalt unvergleichliche Routinensammlung für Ihre eigenen Programme in GW-, Turbo- und Quick BASIC zur Verfügung. Alle Listings und Programme sind auch auf Diskette erhältlich.



AB SOFORT IM HANDEL

DMV-Computer-Wissen Band 1: Basic-Toolbox

18, - DM*

DMV-Computer-Wissen I DATABOX (2 Stck. 5 1/4" und 1 Stck. 3 1/2")

je 35, - DM*

* Unabhängig von der Anzahl der bestellten Artikel berechnen wir für das Inland 4, - DM bzw. für das Ausland 6, - DM Porto und Verpackung. Bitte benutzen Sie die Bestellkarte.

DMV-Verlag · Postfach 250 · 3440 Eschwege



stimmte Privilegstufe zugeordnet, läßt sich dadurch erreichen, daß bestimmte Programme nur auf vorher festgelegte Daten und Routinen zugreifen dürfen.

Protected Mode: neuer Modus des 80286, der den Ablauf von mehreren Programmen gleichzeitig erlaubt

So kann beispielsweise der Systemprogrammierer für andere Anwender ganz einfach den Zugriff auf das Betriebssystem verhindern, indem er den Speichersegmenten mit den Betriebssystemroutinen die höchste Privilegstufe und allen anderen Programmen und Daten eine niedrigere Privilegstufe zuweist. Dieses Verfahren ist auch als Supervisor/User-Konzept bekannt.

Folgende Aufteilung der Privilegstufen bietet sich für ein Multitasking-Betriebssystem an:

- Stufe 0: Betriebssystemkern (Kernel)
- Stufe 1: Betriebssystemumgebung
- Stufe 2: Anwendungsprogramme
- Stufe 3: temporäre Daten

Ein weiterer Vorteil der Speicherverwaltung über die MMU ist, daß Speicherbereiche durch ihre physikalische Unabhängigkeit auch auf sekundäre Speichereinheiten wie Festplatten oder Disketten ausgelagert werden können. Auf diese Weise lassen sich bei Applikationen, die viel Speicher belegen oder große Datenmengen verwalten, Programmteile auslagern, ohne den Speicherzugriffsmechanismus zu ändern.

Das Betriebssystem muß nur noch dafür sorgen, daß die Speicherbereiche zwischen Arbeitsspeicher und externem Datenträger so selten wie möglich ausgetauscht werden, um die Verzögerungen durch die Datenübertragungen in Grenzen zu halten. Betriebssysteme wie OS/2 nutzen bereits diese Möglichkeit der temporären Datenauslagerung, die oft auch als »Swapping« bezeichnet wird.

Der Registersatz des 80286 wurde gegenüber dem 8086/8088-Prozessor lediglich um drei Bit im Statusbitregister und ein vier Bit breites Maschinenstatusregister MSW (englisch: »Machine Status Word«) für prozessorrelevante Informationen und interne Betriebszustände erweitert. Auch die Adressierungsarten des 80286 sind mit denen des 8086/8088-Prozessors identisch.

Während sich Registersatz und Adressierungsarten des 80286 also vom 8086/8088-Prozessor kaum unterscheiden, hat sich beim Befehls-

satz schon mehr geändert. So sind beim 80286 Befehle zur Ein- und Ausgabe von Zeichenketten, zum Anlegen und Löschen von Speicherstrukturen für Hochsprachen sowie zum Laden und Speichern des Maschinenstatusregisters hinzugekommen. Auch lassen sich beim 80286 alle Prozessorregister mit einem einzigen Befehl sichern und zurückholen. Der Großteil der übrigen neuen Befehle zählt zu den sogenannten privilegierten Befehlen, die die besonderen Fähigkeiten des 80286 im Protected Mode unterstützen.

Wichtig ist auch, daß die Entwickler des 80286 bei der Bearbeitungszeit mancher Befehle einige Taktzyklen eingespart haben, indem sie diese Befehle nicht mehr als Mikrocode programmiert, sondern als schnelle, fest verdrahtete Logikschaltungen implementiert haben.

Mußte für die Ausführung von Befehlen des 8086/8088-Prozessors, die als Mikrocode programmiert waren, noch ein kleines, prozessorinternes Programm ablaufen, so lassen sich viele Befehle des 80286, die als Hardware-Rechenwerk realisiert sind, nun in einem einzigen Schritt ausführen.

Mikrocode: Miniprogramm, das die Ausführung komplizierter Prozessorbefehle steuert

Ein internes Konzept, das dem 80286, aber auch allen anderen Intel-Prozessoren zu höherer Geschwindigkeit verhilft, ist die Pipeline-Architektur, die eine bedingte Parallelverarbeitung innerhalb des Prozessors erlaubt. Dazu werden von einer Prefetcher-Einheit in eine Prefetch-Queue (eine Art Datenwarteschlange) laufend die aktuellen Befehle aus dem Speicher geschoben. Gleichzeitig entschlüsselt die Decodereinheit am anderen Ende der Queue die Befehle und schreibt sie (sofern notwen-

dig) in Mikrocode übersetzt in die Instruction-Queue. Die Ausführungseinheit liest die Mikrocode-Befehle wiederum aus der Instruction-Queue und führt sie aus, so daß das Lesen, Decodieren und Ausführen von Befehlen tatsächlich parallel abläuft – sofern nicht eine Programmverzweigung dazwischen kommt. In diesem Fall werden die Queues nämlich komplett gelöscht.

80386: 32-Bit-Prozessor, der zu den 16-Bit-Prozessoren 8086/8088 und 80286 kompatibel ist

Für noch mehr Geschwindigkeit im Personalcomputer sorgt der 80386, der 1986 die Generation des 386-PCs einläutete. Bei dem 80386 handelt es sich um einen echten 32-Bit-Prozessor mit 32 Bit breiten Registern, einem 32-Bit-Datenbus und einem 32-Bit-Adreßbus. Die Tabelle 2 zeigt die wesentlichen technischen Daten des 80386-Prozessors im Vergleich zum 8086/8088 und 80286.

Auch der 80386 ist wieder kompatibel zu seinen Vorgängern, denn er beherrscht sowohl den Real Mode des 8086/8088, als auch den Protected Mode des 80286. Die Abwärtskompatibilität des 80386 zum 8086 und 80286 hat allerdings dazu geführt, daß viele Software-Häuser sich gar nicht mehr die Mühe machen, eine 386-Version ihrer Programme zu entwickeln. So nutzen bisher nur wenige Programme die besonderen Fähigkeiten des 80386. Erst komplett neue Produkte wie OS/2 oder Windows 386 beseitigen dieses Manko und nutzen mit dem Protected Mode (OS/2) beziehungsweise dem 80386-eigenen »Virtual Mode« (Windows 386) die spezifischen Fähigkeiten des 80386 voll aus.

Das Fundament für die Kompatibilität der verschiedenen Prozessoren zueinander bilden Register- und Be-

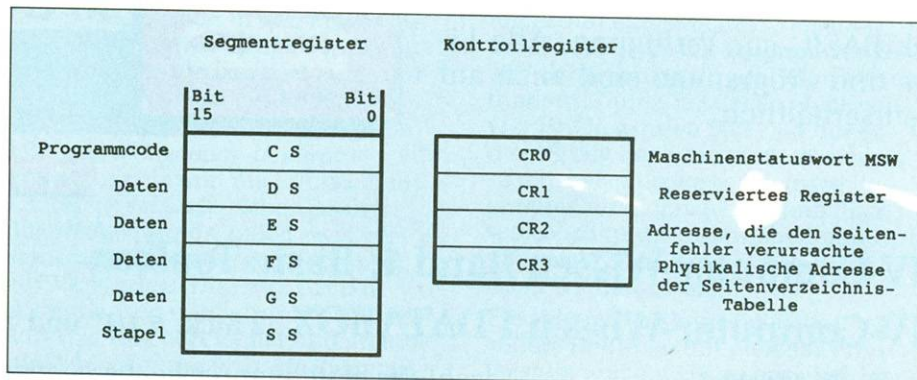


Bild 3. Sechs Segment- und vier Kontrollregister geben dem 80386 die für Multitasking-Betriebssysteme nötige Flexibilität

fehlssatz. Deshalb baut der Registersatz des 80386 auch auf den Registersätzen des 8086 und 80286 auf. Die allgemeinen Daten- und Adreßregister wurden beim 80386 allerdings von 16 auf 32 Bit Breite erweitert (Bild 2), und zusätzlich zu den bekannten Segmentregistern CS, DS, ES und SS sind drei weitere Segmentregister und vier Kontrollregister hinzugekommen (Bild 3).

Der Befehlssatz des 80386 wurde gegenüber dem 80286 nur um wenige Kommandos erweitert. Die Mehrzahl der neuen Befehle ist für die gezielte Manipulation von einzelnen Bits innerhalb eines Registers vorgesehen und schließt damit eine von vielen Programmierern schmerzlich empfundene Lücke im Befehlssatz des 8086/8088 und 80286.

Virtual Mode: neue Betriebsart des 80386 für den parallelen Ablauf mehrerer Real-Mode-Applikationen

Neben der 32-Bit-Architektur ist der Virtual Mode die wichtigste Neuerung beim 80386. Diese dritte Betriebsart, die genaugenommen ein Untermodus des Protected Mode ist,

erlaubt wie der Real Mode die Simulation des 8086-Prozessors. Allerdings kann der 80386 im Virtual Mode nicht nur einen, sondern bis zu 256 voneinander unabhängige 8086-Prozessoren gleichzeitig emulieren, weil er jedem der scheinbar vorhandenen (virtuellen) 8086-Prozessoren einen eigenen 1-MByte-Speicherbereich zuweist. Dadurch ist es dem 80386 möglich, im Virtual Mode mehrere 8086-Applikationen gleichzeitig auszuführen, indem er von einem 1-MByte-Speicherbereich in den nächsten umschaltet und die Register sowie Umgebungsvariablen wie bei einem Task-Wechsel im Protected Mode entsprechend neu initialisiert. Auf diese Weise sind auf einem 80386-Prozessor beispielsweise mehrere Unix- und MS-DOS-Anwendungen gleichzeitig lauffähig. In diesem Fall arbeiten die Unix-Applikationen im Protected Mode und die MS-DOS-Programme parallel dazu im Virtual Mode.

Wie bei dem 80286, so wurden auch bei dem 80386 wieder einige Befehle im Laufzeitverhalten optimiert, indem die Entwickler statt Mikrocode fest verdrahtete Logikschaltungen implementiert haben. Und ein mehrstufiges Pipeline-Konzept zur internen, parallelen Datenverarbeitung

verwendet der 80386 selbstverständlich ebenfalls.

Die vorerst neueste Generation der Personalcomputer begründete der 80486. Erste 486-PCs sind bereits erhältlich, und es steht außer Frage, daß der 80486 im PC-Bereich neue Bestmarken setzen wird.

80486: 32-Bit-Prozessor mit Arithmetikeinheit, Cache-Controller und -Speicher in einem einzigen Baustein

Eigentlich ist beim 80486 gar nicht so viel neu, denn er besteht aus folgenden vier Komponenten (Bild 4):

- eine 80386-kompatible Prozessoreinheit,
- eine 80387-kompatible Arithmetikeinheit,
- ein Cache-Controller,
- ein 8-KByte-Cache-Speicher.

Die Prozessoreinheit des 80486 beherrscht, wie der 80386 den 8086-kompatiblen Real Mode, den 80286-kompatiblen Protected Mode mit MMU und Speicherschutzmechanismen für den Multitasking- und Multiuser-Betrieb sowie den 80386-kompatiblen Virtual Mode.

Neu hinzugekommen sind nur einige Statusbits und Befehle, ansonsten ist alles beim alten geblieben. Register sowie Datenbus sind 32 Bit breit, der Adreßbus adressiert weiterhin 4 GByte direkt und 64 TByte virtuell, und auch bei den Adressierungsarten hat sich nichts geändert. Die im 80486 integrierte Arithmetikeinheit erfährt gegenüber dem Arithmetikprozessor 80387, der von Intel zum 80386 angeboten wird, ebenfalls nur geringfügige Erweiterungen (bei den Befehlen für die vier Grundrechenarten).

Der große Vorteil der Arithmetikeinheit ist, daß der 80486 standardmäßig mathematische Operationen und trigonometrische Funktionen sehr schnell (bis zu 100mal schneller als die Prozessoreinheit) ausführen kann. Bei PCs ist dagegen der Kauf des Arithmetikprozessors 80387, bei ATs der Kauf des 80287 und bei 386-PCs der Kauf des 80387 erforderlich, weil deren Prozessoren nicht serienmäßig mit einer Arithmetikeinheit ausgestattet sind.

Um speziell bei hohen Taktfrequenzen schnelleren Zugriff auf den Speicher zu gewährleisten, ist im 80486 serienmäßig ein Cache-Controller und dazu gleich der passende, 8 KByte große Cache-Speicher integriert. Der Cache-Controller entlastet den Prozessor von zeitaufwendigen Zugriffen auf den Arbeitsspeicher, weil er auf den prozessorinternen



Bild 4. Der 80486 enthält im Prinzip einen 80386-Prozessor, einen 80387-Arithmetikprozessor, einen 82385-Cache-Controller und zweimal 4 KByte Cache-Speicher

Cache-Speicher fast genauso schnell wie die Prozessoreinheit auf die Register zugreifen kann.

Cache-Controller und -Speicher im 80486 sind für den Anwender transparent konstruiert, das heißt, er bemerkt gar nicht (allenfalls an der erhöhten Geschwindigkeit), daß sie vorhanden sind, weil sie automatisch ohne spezielle Programmierung arbeiten.

Der 80486 verfügt wie der 80386 intern über eine 32-Bit-Struktur. Trotzdem arbeitet der 80486 bei gleicher Taktfrequenz mehr als doppelt so schnell wie der 80386. Dazu verhehlen ihm mehrere Tricks:

- die fünfstufige Pipelining-Technik, die es erlaubt, fünf Befehle in

unterschiedlichen Ausführungsstadien parallel zu verarbeiten (Befehl lesen, Befehl decodieren (Stufe 1), Befehl decodieren (Stufe 2), Befehl ausführen, Pipeline-Stufe zurückschreiben),

- die Anwendung von RISC-Techniken beim Design des 80486, so daß sich häufig benötigte Befehle in einem einzigen oder in sehr wenigen Taktzyklen ausführen lassen,
- 128 Bit breite, interne Datenwege zwischen Prozessoreinheit, Arithmetikeinheit und Cache-Speicher,
- ein Burst-Modus für den Cache-Speicher, um ihn besonders schnell mit Daten zu füllen und immer auf dem aktuellen Stand zu bleiben.

Zum Schluß noch ein interessanter Vergleich, der die Leistungssteigerung vom 8086/8088 auf den 80486 verdeutlicht: Der 80486 in der 33-MHz-Version bietet mehr als die 50fache Geschwindigkeit des IBM-PC, der mit einem 8088 und einer Taktfrequenz von 4,77 MHz arbeitet. Übrigens: Der 80486-Nachfolger mit der folgerichtigen Bezeichnung 80586 ist bei Intel bereits in Entwicklung. Bei diesem Prozessor wird es sich dann um einen Baustein mit 64-Bit-Architektur handeln. Mit der Vorstellung des 80586 ist allerdings vor 1992 nicht zu rechnen.

(ma)

Die Entwicklung der Intel-Prozessoren 80x86

Prozessor	Taktfrequenz	Registerbreite	Datenbusbreite	Adreßbereich	maximale Segmentgröße	Besonderheiten
8086	bis 10 MHz	16 Bit	16 Bit	1 MByte	64 KByte	Singletask-Prozessor
8088	bis 10 MHz	16 Bit	8 Bit	1 MByte	64 KByte	Singletask-Prozessor
80286	bis 12,5 MHz	16 Bit	6 Bit	Real Mode: 1 MByte, Protected Mode: 16 MByte (virtuell: 4 GByte)	bis 64 KByte	Multitasking im Protected Mode
80386	bis 33 MHz	32 Bit	32 Bit	Real Mode: 1 MByte, Protected Mode: 4 GByte (virtuell: 64 TByte, entspricht 65536 GByte)	bis 4 GByte	Multitasking im Protected Mode und Virtual Mode (für Real-Mode-Applikationen)
80486	bis 33 MHz	32 Bit	32 Bit	Real Mode: 1 MByte, Protected Mode: 4 GByte (virtuell: 64 TByte, entspricht 65536 GByte)	bis 4 GByte	Multitasking im Protected Mode und Virtual Mode (für Real-Mode-Applikationen), Arithmetikprozessor, Cache-Controller und 8 KByte Cache-Speicher integriert

Tabelle 2 Die Architektur des 80386-Prozessors resultiert aus der logischen Weiterentwicklung seiner Vorgängermodelle

aus der DOS-Software-Reihe: Der Software-Sommerhit

RSM

Resident-Software-Manager

Segment	Program	Dynes	Bekleidung (Vektoren mit Tab)	→ RSM v2.0 ←
0000	MONITOR	14944	Residenter Programmteil des RSM	Laden
1100	KEYBDR	2562	Deutscher Tastaturtreiber	AFB
1220	KEYBDR	2280	Tastaturbeschleuniger	APEND
1280	KEY-PAKE	760	Tastaturmasken in Batch-Dateien	1 ASCII
1300	DRUPRASH	464	(Ctrl)(Alt)(Shift)(CRShift)	2 BREAKOUT
1380	DOSEDIR	2864	Rommodul-Editor	3 CLOCK
1380	COFFCURE	12144	Copy-Cursor: (Alt)->4	4 COMMAND
1670	MOUSE	5504	Maus-treiber	5 COMMAND
1700	PCTOOLS	65080	Disk-Utility: (Ctrl)(Esc)	6 DOW-IT
2770	SPEED	384	DOS-Speedup	7 DOW-IT
2810	GRAPHBL	1312	Grafikzeichensatz	8 DOW-IT
2820	GRAPHICS	3440	Grafik-Hardcopy	9 DOW-IT
2900	OPEN	1776	900-Suchfeld für Dateien	10 DOW-IT
2940	ASCII	21360	Sonderzeichen-Pickup: (Alt)->	11 DOW-IT
2940	CLOCK	880	Bildschirmuhr	12 DOW-IT
2730	HYPERKEY	74496	Tastaturmasken: (Shift)(F1)	13 DOW-IT
4140	LX	79200	Schleisschreiber für 9-Made	14 DOW-IT
5407	PAGES	416	Videoschalttafel: (Alt)->	15 DOW-IT
5404	QUICKDIR	55952	Directory-Utility: (Alt)(F2)	16 DOW-IT

Beendet den Streß mit den bekannt leidigen Problemen: "Not enough memory" Programmabbruch durch Speicherplatzmangel, zumeist im ungünstigsten Moment. "Bad command or filename" In welchem Unterverzeichnis war das jetzt dringend benötigte Programm doch gleich abgelegt? "ALT-F10 oder Shift-F1" Wer kann sich schon die Aufruf- und Löschtafeln all seiner residenten Programme merken?

Komfortables, menügesteuertes Löschen und Nachladen von residenten TSR-Programmen

- * zeigt sofort den noch freien Arbeitsspeicher
- * löscht beliebige, speicherresidente Programme
- * lädt mehrere TSR-Programme in einem Arbeitsgang
- * erinnert an die Tastenkombination zum Aufruf
- * informiert über benutzte Interrupt-Vektoren
- * belegt selbst lediglich 15 KByte Speicherplatz
- * besitzt einen universellen Maustreiber
- * informiert mit einem umfangreichen Handbuch
- * minutenschnelle Installation durch das beigefügte, menügeführte, grafische Installationsprogramm

Für alle PCs, XTs, ATs und Kompatible mit mindestens 256 KByte Arbeitsspeicher

RSM - Ein Programm, das jeder PC-User braucht.

69,- DM (unverbindliche Preisempfehlung)			
Wenn Sie über den DMV-Bestellservice bestellen, gilt folgendes:			
Inland:		Ausland:	
Einzelpreis	69,- DM	Einzelpreis	69,- DM
zzgl. Versandkosten	4,- DM	zzgl. Versandkosten	6,- DM
Endpreis	73,- DM	Endpreis	75,- DM

Bitte benutzen Sie die Bestellkarte.

DMV-Verlag · Postfach 250 · 3440 Eschwege

Noch erhältlich:

DOS EXTRA Nr. 3

- Datenfernübertragungsprogramm in 100% Assemblercode
- Hercules-Grafik-Toolbox in Turbo Pascal
- Grafikfunktionen für Turbo C
- Zeichensatzeditor für Epson-kompatible Drucker
- Treiber für AT-Tastatur
- Alle begleitenden Listings zum Kurs »PC-glasklar«
- Demo für Sprachdigitalisierung

DOS EXTRA Nr. 4

- Desktop-Publishing-Programm »SDTP«
- Kommandoeditor für MS-DOS
- Editor für Disketten und Festplatten
- Speicherresidente Programme auflisten und entfernen
- Paßwort-Schutz für MS-DOS
- Tastatureingabe automatisiert
- Clock-Utility für alle Grafikkarten
- Cursor und Zeichensatz für EGA-Karte
- Fenstertechnik für Turbo C
- Generator für Hilfsbildschirme
- Geprüfte Eingabe mit »Scheck«

DOS EXTRA Nr. 5

- 16 Farben für die CGA-Farbgrafikkarte
- Snapshot: Bildschirmfotos per Software
- Druckerzeichensatz selbstgemacht
- Mausgesteuertes Zeichenprogramm
- Editor in GW-Basic
- Setup-Programm für ATs
- Maschinensprache für GW-Basic
- Erklärtes Listing: das Basic-Spiel »Stardos«
- Neuer Checksummer

DOS EXTRA Nr. 6

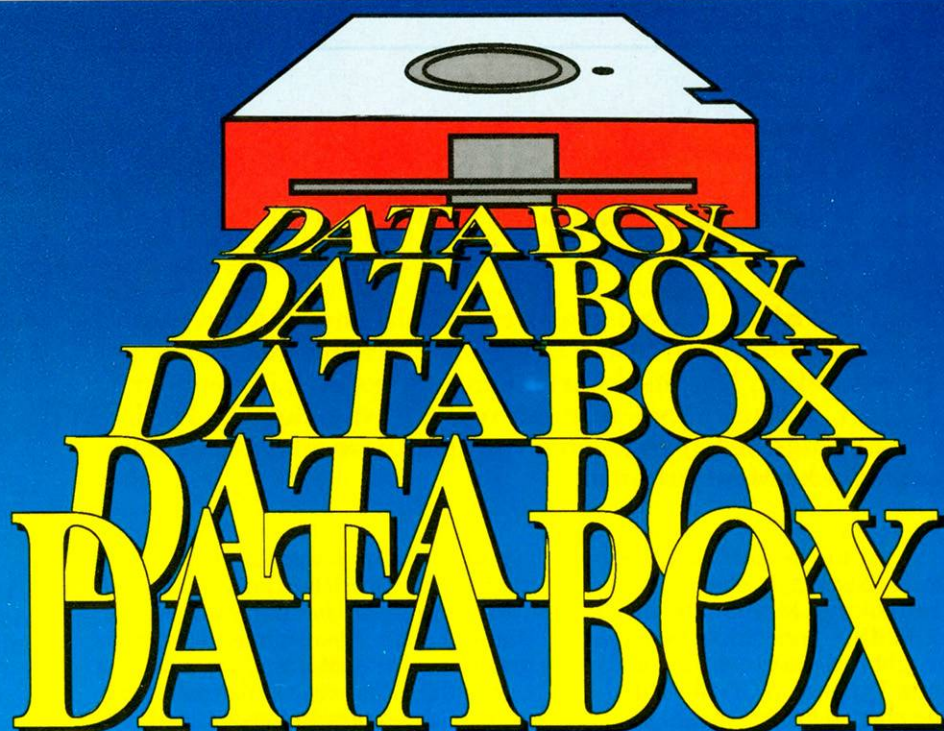
- SBasic-Compiler im Quelltext
- Datenbankprogramm »Parat«
- Fensterorientierte Funktionen für GW-Basic und SBasic
- Turbo-Basic-Bibliothek für 16 Farben
- Analyse von Grafikkarten
- Maskengenerator zum Erzeugen von Binärdateien
- dBase-Datenbanken in Basic
- Superhirn für Quick Basic 4.0

DOS EXTRA Nr. 7

- MS-DOS-Benutzeroberfläche Easydos
- Benutzeroberfläche und Präprozessor für SBasic
- Datengenerator für Basic, C und Pascal
- Sprite-Editor für Farbgrafikkarten
- DOS-Packer
- Virenschutzprogramm
- Weltkarte im Computer
- Schneller Zugriff auf der CGA-Karte
- Kurven einfacher plotten
- Einfacher Prozessor-Emulator
- sowie alle begleitenden Listings zu den Grundlagenbeiträgen und der Tips-&-Tricks-Rubrik

DOS EXTRA Nr. 8

- Makroassembler SAM III mit vielen Beispielen
- Exe2bin-Ersatz durch »exe2com«
- Programm zur Verbesserung der Speicherverwaltung unter MS-DOS
- Routine zum Speichern von Hardcopies auf Datenträger
- Alle Listings zum Einsteigerschwerpunkt
- SCheck 2.01 Checksummer für geprüfte Eingaben
- sowie alle begleitenden Listings zu den Werkstätten



Die Software zu den DOS-EXTRA-Heften

Sagenhaft günstig, extra viel gute Software fürs Geld!

DATABOX:

- mehr als der übliche Softwareservice
- bringt alle Programme der DOS EXTRA auf Datenträger

Für alle PCs unter MS-DOS

Den Aufbau der Databox-Diskette finden Sie in der Datei »read.me« erklärt. Geben Sie deshalb vor dem Start »type read.me« ein. Sämtliche Pascal- und C-Programme liegen auch in einer compilierten Form für Hercules- und Farbgrafikkarten vor. Es ist allerdings nicht gewährleistet, daß diese Versionen auf Hardware, die nicht vollständig kompatibel zum IBM PC ist, problemlos laufen. In diesem Fall müssen Sie den Quellcode nach der Anpassung mit dem entsprechenden Compiler neu übersetzen. Die Basic-Programme arbeiten mit dem GW-Basic-Interpreter von Microsoft einwandfrei. Bei Interpretern und Compilern anderer Hersteller kann es zu Problemen bei der Programmausführung kommen, sofern diese nicht voll kompatibel sind.

**jeweils 2 Disketten 5 1/4"
1 Diskette 3 1/2"**

35,- DM (unverbindliche Preisempfehlung) Wenn Sie über den DMV-Bestellservice bestellen, gilt folgendes:			
Inland:		Ausland:	
Einzelpreis	35,- DM	Einzelpreis	35,- DM
zzgl. Versandkosten	4,- DM	zzgl. Versandkosten	6,- DM
Endpreis	39,- DM	Endpreis	41,- DM

Unabhängig von der Anzahl der bestellten Produkte berechnen wir für das Inland 4,- DM bzw. für das Ausland 6,- DM Porto und Verpackung.

- Bitte benutzen Sie die Bestellkarte. -



DOS EXTRA Nr. 9

- SCheck 2.01: Checksummer für geprüfte Eingaben
- Definitive Eingabe für Batchdateien
- Minidatenbank per Batchdatei
- Utility für komfortable MS-DOS-Befehlseingabe
- Utility für sicheres Dateilöschen
- Pulldown-Menüs unter GW-Basic programmieren
- Briefbogen selbstgemacht
- Analoguhr für Selfmade-Designer
- Konvertierung vom EGA- in den Hercules-Modus: Schraffur statt Farbe
- sowie die begleitenden Listings aller anderen Beiträge

Die Grafikstandards der PC-Welt

Für PCs gibt es nicht, wie beispielsweise für Heimcomputer, nur einen einzigen, fest installierten Grafikstandard mit dem dazugehörigen Monitor. Das Steckkartenkonzept des PC hat vielmehr dazu geführt, daß mittlerweile mehrere Grafikstandards in Form von Erweiterungskarten angeboten werden. Diese erfordern auch unterschiedliche Monitortypen.

Nicht nur der Einsteiger hat Mühe, bei dem großen Angebot an Grafikkarten- und Monitorstandards für den PC die Übersicht zu behalten, denn es gibt inzwischen fünf verschiedene Typen von Grafikkarten und eine ebenso große Anzahl an Monitortypen.

Bei den Grafikstandards handelt es sich im einzelnen um:

- MDA (»Monochrome Display Adapter«)
- Hercules (MDA mit der Fähigkeit, auch Grafiken darzustellen)
- CGA (»Color Graphics Adapter«)
- EGA (»Enhanced Graphics Adapter«)
- VGA (»Video Graphics Array«)

Die fünf Monitortypen sind

- Monochrom-Monitor (für MDA und Hercules)
- Farbmonitor (für CGA)
- EGA-Monitor
- analoger VGA-Monitor
- Multiscan-Monitor (für alle Grafikkarten geeignet)

Die erste Grafikkarte für den PC, die 1981 mit Einführung des IBM-PC vorgestellt wurde, ist keine Grafikkarte im eigentlichen Sinn, sondern vielmehr eine »Textkarte«, denn sie kann nur Textzeichen in 25 Zeilen zu jeweils 80 Zeichen in einer Auflösung von 720x350 Bildpunkten einfarbig darstellen. Offiziell wird diese Karte als Monochromadapter bezeichnet, abgekürzt MDA. Für den Anschluß an den Monochromadapter reicht ein preiswerter Monochrom-Monitor, der im Fachhandel inzwischen für weniger als 300 Mark angeboten wird. Ebenfalls mit Einführung des IBM-PC stellte IBM die CGA-Karte vor, die, wie der Name »Color Graphics Adapter« schon sagt, farbige Grafiken erzeugen kann. Die CGA-Karte vermag Texte in 16 Farben gleichzeitig und Grafiken mit einer Auflösung von 320x200 Bildpunkten in vier und 640x200 Bildpunkten in zwei Farben darzustellen. Für den Anschluß an die CGA-Karte genügt ein Farbmonitor, der 200 Bildzeilen anzeigt, für etwa 500 bis 600 Mark. Die fehlende Grafikfähigkeit des Monochromadapters störte viele PC-An-

wender, und so entwickelte der Hersteller Hercules 1982 die inzwischen weit verbreitete Hercules-Grafikkarte, die sehr schnell – nicht zuletzt aufgrund der frühzeitigen Unterstützung durch die Tabellenkalkulation Lotus 1-2-3 – zu einem weiteren Grafikstandard avancierte. Damit ist die Hercules-Karte einer der wenigen Standards in der PC-Welt, der nicht von IBM gesetzt wurde.

Die Hercules-Karte bietet gegenüber dem Monochromadapter zusätzlich die Fähigkeit zur Darstellung von Grafiken

Die Hercules-Karte ist voll kompatibel zum Monochromadapter, besitzt allerdings zusätzlich einen Grafikmodus, der es erlaubt, 720x348 Bildpunkte einzeln zu adressieren (gegenüber dem MDA unterschlägt die Hercules-Karte die oberste und unterste Bildzeile).

Wie der Monochromadapter, so stellt auch die Hercules-Karte Texte und

Grafiken nur monochrom dar. Auf diese Weise läßt sich an die Hercules-Karte der gleiche Monochrom-Monitor anschließen, den auch der Monochromadapter benötigt. Viele Hersteller bieten mittlerweile Hercules-kompatible Grafikkarten für unter 150 Mark an.

Weil viele PC-Anwender eine Hercules-Karte besitzen, manche Programme und die meisten PC-Spiele allerdings nur die CGA-Karte unterstützen, gibt es inzwischen eine Vielzahl von sogenannten CGA-Emulatoren auf dem Markt, die die CGA-Modi auch auf der Hercules-Karte darstellen. Die Emulatoren setzen dabei die Farbwerte in Helligkeitswerte um, die sich – aufgrund der höheren Auflösung der Hercules- gegenüber der CGA-Karte – durch Bildpunkt-muster erreichen lassen.

Einen anderen Weg ging Commodore mit der AGA-Karte (»Alternative Graphics Adapter«), die sowohl CGA- als auch Hercules-Grafik darstellen kann, allerdings dazu auch den entsprechenden Dualfrequenzmonitor benötigt, der beide Standards verarbeitet.

1984 stellte IBM mit Einführung des IBM-AT die EGA-Karte vor, die eine Auflösung von 640x350 Bildpunkten in 16 Farben erlaubt und auch CGA-kompatibel ist. Zudem lassen sich mit der EGA-Karte die CGA-Auflösungen auf Wunsch ebenfalls in 16 Farben darstellen. Sie können aus einer 64 Farben umfassenden Palette wählen. Die Preise für die Mehrheit der EGA-Karten liegen zwischen 300 und 600 Mark.

Wenn Sie nicht nur die CGA-kompatiblen Modi, sondern auch die volle Auflösung der EGA-Karte nutzen möchten, benötigen Sie entweder einen hochauflösenden EGA-Monitor

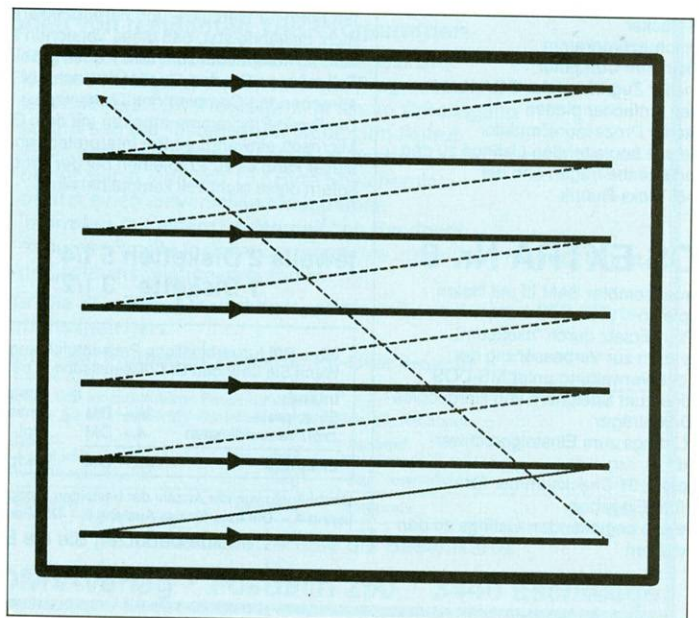


Bild 1. Der Elektronenstrahl der Bildröhre baut das Monitorbild zeilenweise auf

(Kostenpunkt: 800 bis 1000 Mark), oder einen Multiscan-Monitor (1200 bis 2000 Mark), der die Verbindung mit allen gängigen Grafikkarten erlaubt, weil er sich automatisch auf die Signale der angeschlossenen Grafikkarte einstellt.

Inzwischen gibt es auch EGA-Karten auf dem Markt, die noch höhere Auflösungen von 640x480 und 800x600 Bildpunkten erlauben. Lassen Sie sich aber von diesen Angeboten nicht irritieren, denn weil jeder Kartenhersteller bei der Implementierung der neuen Modi sein eigenes Süppchen kocht, können Sie Ihre Programme nur dann mit diesen Auflösungen benutzen, wenn der Hersteller auch die entsprechenden Software-Treiber mitliefert. Und diese liegen dann – sofern überhaupt vorhanden – immer nur für die wichtigsten Standardprogramme vor. Auch können Sie die erweiterten Modi nur nutzen, wenn Sie einen Multiscan-Monitor besitzen, weil ein EGA-Monitor nur maximal 640x350 Bildpunkte darstellen kann.

VGA heißt der gegenwärtig leistungsfähigste PC-Grafikstandard, der inzwischen kaum teurer als EGA ist

1987 schließlich führte IBM mit der Vorstellung der PS/2-Modelle die VGA-Grafik ein, die inzwischen auf dem besten Wege ist, zum wichtigsten Grafikstandard in der PC-Welt zu werden. Die VGA-Grafik beherrscht – neben den CGA- und EGA-Auflösungen – eine neue Auflösung mit 640x480 Bildpunkten in 16 Farben, die den großen Vorteil hat, daß die Bildpunkte quadratisch sind, so daß ein Kreis auch wirklich kreisrund und nicht eiförmig dargestellt wird.

Daneben kann die VGA-Karte im Modus mit 320x200 Bildpunkten auch 256 Farben gleichzeitig darstellen. Der wichtigste Unterschied zwischen

der VGA-Karte und den Vorgängern ist allerdings, daß die VGA-Karte nicht mehr mit digitalen, sondern mit analogen Farbsignalen arbeitet. Dadurch erhöht sich die Palette, aus der Sie die Farben wählen können, ganz erheblich von 16 (CGA) beziehungsweise 64 (EGA) auf 262144! VGA-Karten werden inzwischen für 300 bis 1000 Mark angeboten, so daß die preisgünstigsten Modelle nicht teurer als die preiswertesten EGA-Karten sind. Aufgrund der analogen Farbsignale benötigen Sie für den Anschluß an die VGA-Grafik allerdings einen speziellen, analogen VGA-Monitor, der 200 bis 300 Mark teurer als ein EGA-Monitor ist. Alternativ können Sie auch einen Multiscan-Monitor benutzen, weil dieser Monitortyp ebenfalls einen Eingang für analoge Farbsignale besitzt.

Wie bei den EGA-Karten, so bieten inzwischen die meisten Hersteller auch VGA-Karten an, die über erweiterte Auflösungen mit 800x600 und sogar 1024x768 Bildpunkten verfügen, und – sofern sie mit 512 KByte anstelle der sonst üblichen 256 KByte Bildschirmspeicher ausgestattet sind – die Auflösungen mit 640x480 und 800x600 Bildpunkten auch in 256 statt 16 Farben darstellen.

Für diese erweiterten Modi gilt allerdings das gleiche wie bei den EGA-Karten, denn Software läßt sich nur in diesen Modi nutzen, wenn auch die entsprechenden Treiber vom Kartenhersteller mitgeliefert werden. Allerdings haben sich inzwischen die wichtigsten Grafikkarten- und Monitorhersteller in einer Organisation namens VESA (Video Electronics Standards Association) zusammenschlossen, um die erweiterten VGA-Modi zu standardisieren. So wurde bereits der Modus mit 800x600 Bildpunkten in 16 Farben definiert, und die Modi mit 640x480 und 800x600 Bildpunkten in 256 Farben sollen folgen.

Auf diese Weise lassen sich künftig die erweiterten Modi bei allen VGA-Karten auf die gleiche Art ansprechen, so daß nicht mehr für jede Gra-

fikkarte ein anderer Treiber benötigt und den Software-Herstellern die Anpassung ihrer Programme an diese Modi erheblich erleichtert wird.

Um zu verstehen, warum für jede Grafikkarte ein anderer Monitortyp erforderlich ist, müssen Sie wissen, wie Grafikkarte und Monitor zusammenarbeiten:

Damit der Monitor »weiß«, was für ein Bild er darstellen soll, erhält er von der Grafikkarte ein oder mehrere Videosignale, die den Bildinhalt und die Farbinformationen enthalten, sowie die beiden Synchronisationssignale für den horizontalen und vertikalen Bildaufbau. Das oder die Videosignale geben durch ihren Zustand an, ob ein Bildpunkt hell gesetzt wird oder dunkel bleibt und welche Farbe er erhält.

Die erweiterten Modi, die viele EGA- und VGA-Karten bieten, sind zueinander oft nicht kompatibel

Dazu erzeugt eine Elektronenkanone in der Bildröhre des Monitors einen Elektronenstrahl, der die Videosignalinformationen zeilenweise auf den Bildschirm schreibt. Dazu beginnt der Elektronenstrahl links oben in der Ecke, fährt in die rechte obere Ecke und schaltet sich kurz aus (Austastlücke). In diesem ausgeschalteten Zustand kehrt der Strahl an den linken Rand zurück, ohne eine Spur auf dem Bildschirm zu hinterlassen. Dann beginnt der Elektronenstrahl eine Bildpunktzeile unter der vorherigen Zeile und bewegt sich wieder nach rechts.

Der Monitor wiederholt diesen Vorgang, bis der Elektronenstrahl in der rechten unteren Ecke angelangt ist. Dann tastet er wieder aus, kehrt diesmal jedoch diagonal über den Bildschirm in die linke obere Ecke zurück und beginnt mit dem Aufbau des nächsten Bildes. Bild 1 stellt diesen Vorgang schematisch dar. Die gestri-

Grafikkarte	digital/analog	maximale Auflösung	Angebot an Farben	Zeilenfrequenz (Horizontalfrequenz)	Bildfrequenz (Vertikalfrequenz)
Monochromadapter	digital	720x350 Bildpunkte (nicht grafikfähig)	–	18,43 kHz	50 Hz
Hercules-Karte	digital	720x348 Bildpunkte	–	18,43 kHz	50 Hz
CGA-Farbgrafikkarte	digital	640x200 Bildpunkte	16	15,75 kHz	60 Hz
EGA-Karte	digital	640x350 Bildpunkte	64	21,85 kHz	60 Hz
VGA-Karte	analog	640x480 Bildpunkte	262144	31,47 kHz	70 Hz (60 Hz bei 640x480 Bildpunkten)

Tabelle 1. Die wichtigsten technischen Daten der PC-Grafikkarten im Überblick

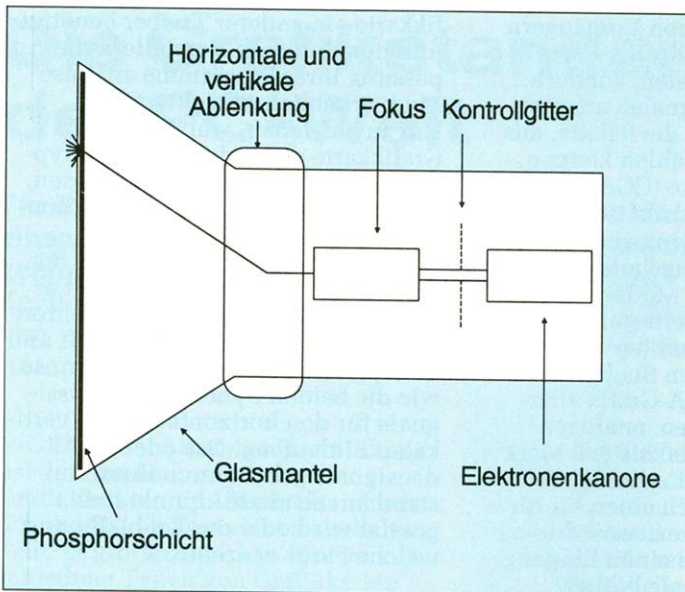


Bild 2. Der schematische Aufbau einer Bildröhre zeigt den Weg des Elektronenstrahls von der Kanone bis zur Mattscheibe

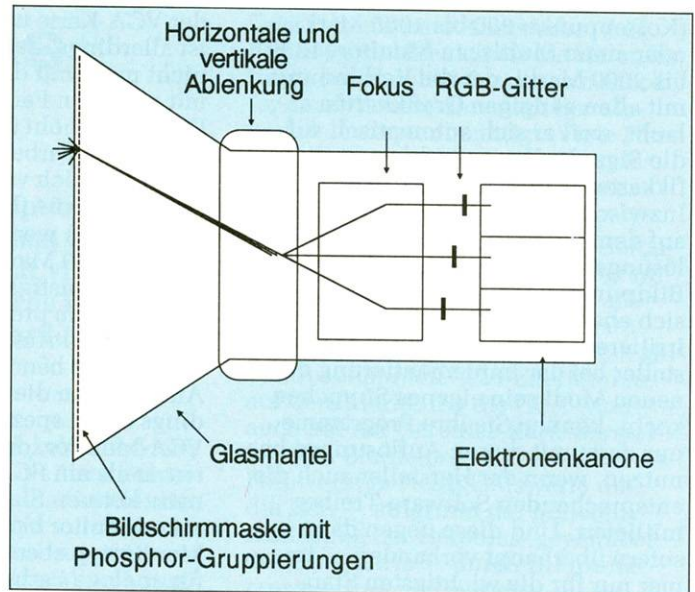


Bild 3. Ein Farbmonitor benötigt für jede der drei Grundfarben eine eigene Elektronenkanone

chelten Linien deuten den Austast- und Rücklaufweg des Elektronenstrahls an.

Bild 2 zeigt die Funktionsweise der Bildröhre im Monitor. Der negativ geladene Elektronenstrahl wird in einer Elektronenkanone am rückwärtigen Ende der Bildröhre erzeugt und durch eine hohe positive Spannung bis zur Mattscheibe beschleunigt. Nach dem Austritt aus der Elektronenkanone bringt eine Ablenkeinheit den Strahl in eine runde Form und lenkt ihn horizontal und vertikal auf den gewünschten Bereich des Bildschirms.

Das Videosignal definiert durch seinen Zustand, ob ein Punkt auf dem Bildschirm gesetzt oder gelöscht wird

Die Auswertung der Videosignalinformationen durch den Elektronenstrahl der Bildröhre ist bei einem Monochrom-Monitor am einfachsten: Hat die Grafikkarte im PC das Videosignal gesetzt, so ist auch der Elektronenstrahl des Monitors aktiv und bringt dadurch die Beschichtung des Bildschirms an der Auftreffposition des Strahls zum Leuchten. Die Leuchtfarbe der Beschichtung bestimmt, ob es sich um einen schwarz-weißen, grünen oder amberfarbenen Monochrom-Monitor handelt.

Ist das Videosignal nur kurzzeitig gesetzt, so sieht der Betrachter lediglich einen Punkt auf dem Bildschirm, ist das Signal länger aktiv, so erzeugt der Elektronenstrahl durch seine seit-

liche Bewegung einen waagerechten Strich.

Wenn das Videosignal nicht gesetzt ist, so streicht der Elektronenstrahl der Bildröhre im inaktiven Zustand über den Bildschirm, und es ist nichts zu sehen. Ein zurückgesetztes Videosignal entspricht also einem schwarzen Punkt beziehungsweise einer schwarzen Linie.

Erzeugt die Grafikkarte beispielsweise ein Videosignal, das periodisch gesetzt und zurückgesetzt wird, so sind auf dem Bildschirm des Monitors abwechselnd helle (das heißt weiße, grüne oder amberfarbene) und schwarze Punkte zu sehen, die den Eindruck einer gestrichelten Linie vermitteln. Je schneller die Grafikkarte das Videosignal setzen und zurücksetzen kann, desto mehr Punkte lassen sich in einer Bildzeile darstellen.

Je nach Grafikkarten- und Monitortyp wird ein Computerbild 50- bis 70mal pro Sekunde aufgebaut. Damit trotz dieser Hektik die Zusammenarbeit zwischen der Elektronik der Grafikkarte und der des Monitors reibungslos klappt, das heißt das Bild nicht verwischt, verzerrt oder unvollständig dargestellt wird, existieren die beiden Signale zur horizontalen und vertikalen Synchronisation.

Das Signal zur horizontalen Synchronisation ist für den Aufbau der Bildzeilen zuständig. Ein Impuls dieses Signals bestimmt den Zeitpunkt, zu dem der Elektronenstrahl den Aufbau einer Bildzeile beendet und an den Anfang der nächsten Zeile zurückkehrt. Das Signal zur vertikalen Synchronisation ist für den kompletten Bildaufbau zuständig und definiert entsprechend, wann der Elektronenstrahl den Aufbau eines Bildes

beendet und zum Startpunkt für das nächste Bild in die linke obere Ecke zurückkehrt.

Damit die Bildzeilen eine konstante Länge aufweisen und alle Bilder gleich hoch sind, werden die beiden Synchronisationssignale mit einer festen Frequenz getaktet. Die Frequenz des Signals zur horizontalen Synchronisation wird mit Zeilenfrequenz oder Horizontalfrequenz und die Frequenz des Signals zur vertikalen Synchronisation mit Bildfrequenz oder Vertikalfrequenz bezeichnet. Tabelle 1 zeigt in einer Gegenüberstellung die wichtigsten technischen Daten der verschiedenen PC-Grafikkarten mit ihren Zeilen- und Bildfrequenzen.

Farbgrafikkarten und -monitore mischen ihre Farbtöne aus den drei Grundfarben Rot, Grün und Blau

Das beschriebene Verfahren ist genau genommen nur für Grafikkarten, die Monochrom-TTL-Signale erzeugen, und für Monitore, die diese Signale verarbeiten, gültig. Grafikkarten mit RGB-TTL-Signalen, also CGA- und EGA-Karten, arbeiten nicht mit einem einzigen, sondern mit mehreren Videosignalen, die die drei Grundfarben Rot, Grün und Blau repräsentieren.

Die Abkürzung TTL steht für Transistor-Transistor-Logik und ist ein Synonym für Schaltungen, die mit binären, also digitalen Signalen arbeiten. TTL-Signale sind demnach digitale Signale. RGB bedeutet Rot-Grün-Blau und gibt an, daß es sich bei RGB-Si-

gnalen um Farbsignale handelt. Monochrom-TTL-Signale sind demnach digitale Signale ohne Farbinformationen und RGB-TTL-Signale digitale Signale mit Farbinformationen.

Bei der CGA-Farbgrafikkarte wird jede der drei Grundfarben durch ein Signal repräsentiert, bei der EGA-Karte sind es zwei Signale pro Farbe. Der Unterschied zwischen einem Monochrom- und einem Farbmonitor besteht im Prinzip darin, daß der Farbmonitor statt mit einer, mit drei Elektronenkanonen arbeitet (Bild 3). Jede Elektronenkanone wird von den Signalen für eine der drei Grundfarben gesteuert, so daß je einer der drei erzeugten Elektronenstrahlen für die Farben Rot, Grün und Blau zuständig ist.

Die Elektronenstrahlen innerhalb der Bildröhre sind bei einem Farbmonitor so justiert, daß sie durch eine loch- oder schlitzförmige Maske nur jeweils auf den Teil der Beschichtung der Bildröhre treffen können, der für die entsprechende Grundfarbe zuständig ist und in der betreffenden Farbe leuchten kann.

Durch Mischen der drei Grundfarben lassen sich alle denkbaren Farbtöne erzeugen. Weil die RGB-TTL-Signale

Pin	Monochrom-TTL-Signale des Monochromadapters und der Hercules-Karte	RGB-TTL-Signale der CGA-Karte	RGB-TTL-Signale der EGA- und VGA-Karte
1	Masse	Masse	Masse
2	Unbenutzt	Unbenutzt	Rot (LSB)
3	Unbenutzt	Rot	Rot (MSB)
4	Unbenutzt	Grün	Grün (MSB)
5	Unbenutzt	Blau	Blau (MSB)
6	Intensität	Intensität	Grün (LSB)
7	Videosignal	Luminanz	Blau (LSB)
8	Horizontale Synchronisation (positiv)	Horizontale Synchronisation (positiv)	Horizontale Synchronisation (positiv)
9	Vertikale Synchronisation (negativ)	Vertikale Synchronisation (positiv)	Vertikale Synchronisation (negativ)

Tabelle 2. Die Pin-Belegung des 9poligen Monitorsteckers für Monochrom-TTL- und RGB-TTL-Signale

jedoch nur zwei Zustände kennen (gesetzt und zurückgesetzt), ist die Farbvielfalt bei der CGA-Farbgrafikkarte auf 16 und bei der EGA-Karte auf 64 Farben beschränkt.

Die CGA-Karte kann mit zwei möglichen Zuständen pro Farbsignal eigentlich sogar nur $2 \times 2 \times 2 = 8$ Farben darstellen, doch erzeugt sie zusätzlich ein Intensitätssignal, das bei Be-

darf die Helligkeit aller acht Farben erhöht, so daß insgesamt acht dunkle und acht helle Farben zur Verfügung stehen. Bei der EGA-Karte ist die Rechnung einfacher. Zwei digitale Signale pro Grundfarbe ergeben vier Zustände pro Farbe, und $4 \times 4 \times 4$ ist gleich 64.

Die VGA-Grafik erlaubt viel mehr Farben, weil sie mit RGB-Analogsi-

Ordnung und Übersicht schaffen die beliebten DMV Sammelmappen



Bitte Bestellkarte benutzen
DMV Verlag · Postfach 250 · 3440 Eschwege

CASIO SHARP

„Die perfekte Kopplung.“

TRANSFILE koppelt Ihren Pocketcomputer mit Ihrem PC/XT/AT.

- sichere Übertragung in beide Richtungen
- schnelles Speichern/Laden auf Disk und Festplatte
- integrierter Editor
- Pull-Down-Menüs
- vergoldete Stecker
- Standard-Datenformate

Für weitere Produktinfos rufen Sie uns einfach an.

ab **DM 129,-**

Händleranfragen erwünscht.
Telefax 0 71 36/2 25 13

yellow
C-O-M-P-U-T-I-N-G
Postfach 1136/6
D-7107 Bad Friedrichshall
Telefon 0 71 36/2 00 16

MÜNZENLOHER GmbH
Tölzer Straße 5 • 8150 Holzkirchen
Tel. (08024) 1814 • FAX: (08024) 4879

ATARI-SCHNEIDER-NEC PANASONIC-PHILIPS Star-OKI-LUCKY GOLDSTAR

Hard- und Software in großer Auswahl

Beratung und Service sollten bei Ihrem Kauf entscheidend sein!

Nicht möglich !!

Der kleinste PC der Welt ist da
ATARI PORTFOLIO Textverar., Adressdat., Termin-kalender-, Kalkulations-Programme im ROM **DM 798,-**

Parallel Interface	DM 98,-	32 kB RAM-Card	DM 128,-
Seriell Interface	DM 158,-	64 kB RAM-Card	DM 248,-
MemoryExpansion	DM 428,-	128 kB RAM-Card	DM 378,-
256 kB		Externes Ramkartenlaufwerk	DM 198,-
Netzteil	DM 19,-		

ATARI PC 3 640 kB RAM, 4,77/8 MHz; Grafikmodi: EGA, 5 Slots, 12" Monitor (bernst.). 1*360 kB 5 1/4" Laufwerk und 30 MB Festplatte/Echtzeituhr, inkl. Maus, Centronics, seriell: Externes Laufwerk anschl. MS-DOS, GW-Basic, GEM Desktop, GEM-Write, GEM-Print, dt. Handbüchern **DM 2498,-**

ATARI AT ABC286-30 640 kB RAM erw auf 4 MB, 6/8 MHz, Grafikmodi: EGA, 3 Slots (AT) frei, 1*144 MB 3 1/2" Laufwerk, 30 MB Festplatte, Echtzeituhr, Centronics, 2x seriell, Platz für 5 1/4" Laufwerk vorhanden. Software: MS-DOS, GW-Basic, deutsche Handbücher **DM 2798,-**

Neue Monitore von ATARI

PCM145 14" VGA-Mono-Monitor/ Weiß/ 640x480/30 MHz, Reflexionfreies Bild **DM 398,-**

PCC1414/15 14" EGA-Farb-Monitor/EGA, 640x350/ 64 Farben/0.39mm Pitch/TTL RGB / umschaltbar grün,bernst., entspiegelt **DM 848,-**

PCC1412/25 14" VGA-Farb-Monitor/640x480 256 Farben/0.29mm Pitch/entspiegelt **DM 1098,-**

Alle Monitore mit Schwenkfuß

Weitere Informationen ? Rufen Sie uns an

gnalen arbeitet. Hier sind für jede Grundfarbe 64 verschiedene Abstufungen möglich, weil die VGA-Grafik drei 6-Bit-Digital/Analog-Wandler zum Erzeugen der analogen Farbsignale benutzt. Durch Mischen kann die VGA-Grafik bis zu $64 \times 64 \times 64 = 262144$ verschiedene Farbtöne auf einem Monitor, der RGB-Analogsignale verarbeitet, darstellen.

Beim RGB-Analogeingang eines VGA- oder Multiscan-Monitors ist ebenfalls für die Farben Rot, Grün und Blau je ein Farbsignal vorhanden. Diesmal bestimmt jedoch die Signalstärke direkt die Helligkeit des entsprechenden Bildpunktes auf der Bildröhre. Liegen beispielsweise die Signalpegel aller drei Farbsignale auf ihrem maximalen Wert, so sind der rote, grüne und blaue Farbpunkt auf volle Helligkeit geschaltet, und der Betrachter sieht die Farbe Weiß.

Die VGA-Grafik kann sehr viele verschiedene Farben darstellen, weil sie analoge Farbsignale verwendet

Umgekehrt sind bei einem minimalen Signalpegel der drei Farbsignale alle Farbpunkte dunkel, so daß das Bild schwarz erscheint. Zwischen den Farben Schwarz und Weiß sind jedoch alle denkbaren Farbkombinationen erlaubt, weil die drei analogen Farbsignale alle Stufen (genau genommen jeweils 64) zwischen minimaler und maximaler Helligkeit annehmen können.

Ein Monitor greift die Signale der Grafikkarte über ein Datenkabel mit einem 9poligen oder 15poligen Stecker ab. Monitore, die mit digitalen Signalen arbeiten, besitzen gewöhnlich einen 9poligen Stecker und Analogmonitore entweder ebenfalls einen 9poligen oder einen 15poligen, PS/2-kompatiblen Stecker. Tabelle 2

Pin (15polig)	Pin (9polig)	Ein-/Ausgang	RGB-Analogsignale der VGA-Grafik
1	1	Eingang	Rot
2	2	Eingang	Grün
3	3	Eingang	Blau
4	—	Ausgang	Monitoridentifizierung Bit 2 (optional)
5	9	—	Masse
6	6	—	Masse für Rot
7	7	—	Masse für Grün
8	8	—	Masse für Blau
9	—	—	Nicht belegt
10	9	—	Masse für Synchronisation
11	—	Ausgang	Monitoridentifizierung Bit 0 (optional)
12	—	Ausgang	Monitoridentifizierung Bit 1 (optional)
13	4	Eingang	Horizontale Synchronisation
14	5	Eingang	Vertikale Synchronisation
15	—	—	Reserviert

Monochrome Analogmonitore benutzen nur das Grün-Signal für die Helligkeit und ignorieren die Farbanteile für Rot und Blau

Tabelle 3. Die Pin-Belegung des 15poligen und 9poligen Monitorsteckers für RGB-Analogsignale

zeigt die Pin-Belegung des 9poligen Monitorsteckers für Monochrom-TTL- und RGB-TTL-Signale und Tabelle 3 die Pin-Belegung des 9poligen und 15poligen Monitorsteckers für RGB-Analogsignale aus der Sicht des Monitors.

Die zentralen Steuerungsfunktionen zum Erzeugen des Videosignals und der Synchronisationssignale nimmt auf der PC-Grafikkarte der Video-Controller wahr. Dieser hochintegrierte Baustein setzt die Daten aus dem Bildschirmspeicher in elektrische Signale für den Monitor um. Um die im Bildschirmspeicher abgelegten Daten in Punkte, Zeichen oder Grafiken für den Bildschirm umzuwandeln, liest der Video-Controller diese Daten ein und formt daraus die elektrischen Signale für den Monitor, der dann den gewünschten Text oder die Grafik anzeigt.

Im Textmodus interpretiert der Video-Controller die Bytes im Bild-

schirmspeicher als ASCII-Codes und ordnet ihnen eine Bildpunktmatrix aus dem Zeichengenerator zu, die den Aufbau des entsprechenden Zeichens auf dem Bildschirm definiert. Im Grafikmodus kann der Video-Controller dagegen die einzelnen Bits des Bildschirmspeichers auslesen und sie unter Berücksichtigung der eventuell vorhandenen Farbinformationen direkt als Bildpunkte für den Bildschirm umsetzen.

Bild 4 zeigt die Arbeitsweise des Video-Controllers auf einer Grafikkarte. Der Video-Controller kommuniziert über den Daten- und Adressbus mit dem Prozessor und dem Bildschirmspeicher. Der Controller liest in periodischen Zeitabständen (50-, 60- oder 70mal in der Sekunde) den Inhalt einer Bildschirmseite und formt ihn in Signale für den Monitor um. Wenn der PC im Textmodus arbeitet, übergibt der Controller die eingelesenen Bytes aus dem Bildschirmspeicher als ASCII-Werte an den Zeichengenerator. Jeder ASCII-Wert definiert dabei eine Adresse, unter der im Zeichengenerator die Bildpunktmatrix zu dem entsprechenden Zeichen abgelegt ist. Diese Bildpunktmatrix liest der Video-Controller und erzeugt daraus über ein sehr schnelles Schieberegister das serielle Videosignal.

Befindet sich der PC dagegen im Grafikmodus, so wandelt der Video-Controller die aus dem Bildschirmspeicher eingelesenen Daten direkt über das Schieberegister in das Videosignal um. Eine Konvertierung ist hier nicht nötig, weil die Bits des Bildschirmspeichers im Grafikmodus direkt den Bildinhalt angeben.

(ma)

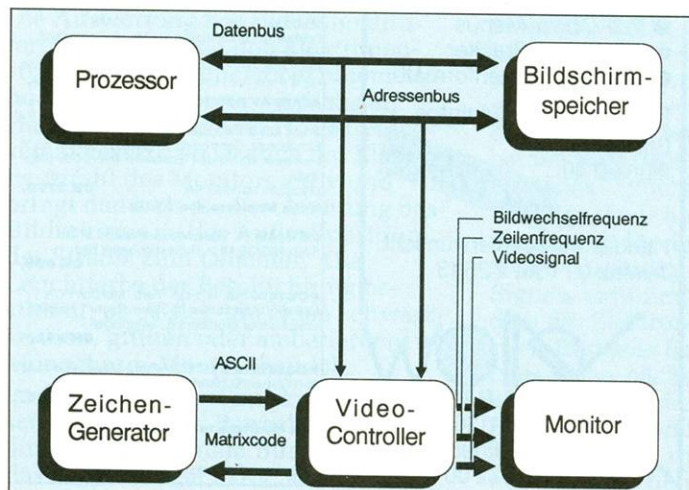


Bild 4. Der Video-Controller dient als Vermittler zwischen Prozessor, Bildschirmspeicher, Zeichengenerator und Monitor

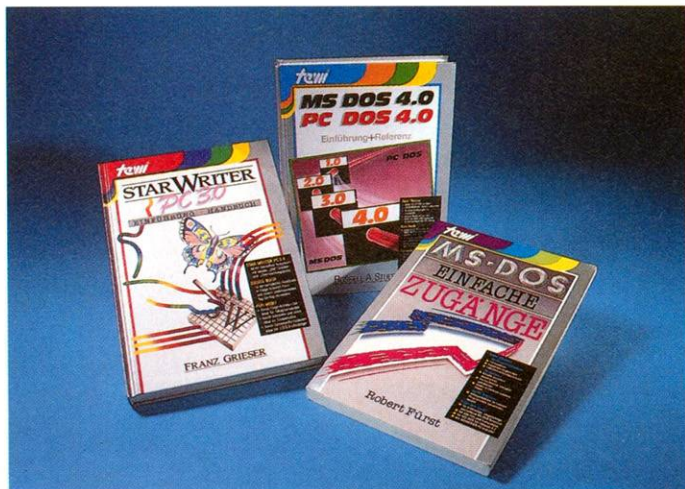


Bild 1. Der Verlag Tewi bietet ein für Einsteiger, aber auch für Fortgeschrittene hilfreiches Konzept

Eine Einsteiger-Computer-Bibliothek

Berührungsängste gerade im Bereich EDV und Computer sind bei Neulingen an der Tagesordnung. Ein gutes Buch, das schnell Erfolgserlebnisse vermittelt, ist hier sehr hilfreich. Doch gerade der Einsteiger fühlt sich bei der Wahl seiner Literatur oft sehr verunsichert und alleingelassen ...

Der Begriff »Computer« wird häufig verbunden mit dem Begriff »Unverständlichkeit«. Der Grund für diese Assoziation ist zum Teil bei anwenderunfreundlichen Programmen zu suchen, verantwortlich dafür sind jedoch auch diverse Autoren von Fachliteratur, die oft schlichtweg nicht in der Lage sind, ihr »Fachchinesisch« einem Laien verständlich nahezubringen. Da wimmelt es von Fremdwörtern und unklaren Satzkonstruktionen – kommt dazu dann noch ein fades Layout, ist die Schlacht schon halb verloren. Eigenartigerweise wird jedoch die Schuld, wenn etwas unverständlich bleibt, immer noch der Thematik zugeschoben – beileibe nicht den Fachbüchern. »Nicht das Buch ist schuld, wenn ich etwas nicht begreife, nein, das liegt an mir selbst und eben an der komplizierten Thematik« so lautet die Argumentation – und man kauft sich ein weiteres Buch ...

Von dem genannten Vorurteil sollten Sie sich freimachen, wenn Sie sich auf die Suche nach Fachliteratur begeben; denn ein schlechtes Buch kann sehr wohl dazu

beitragen, daß Sie Verständnisschwierigkeiten haben. Nun stellt sich die Preisfrage, wie die Konzeption eines Buches, das einem Einsteiger die ihm fremde Thematik leicht verständlich aufbereiten soll, auszusehen hat. Natürlich hängt es auch vom individuellen Geschmack und der Veranlagung der jeweiligen Person ab, ob sie eine Information unterhaltsam verpackt oder ganz nüchtern vorgestellt vorzieht. Einige grundlegende Prämissen gelten jedoch sicherlich allgemein: Grundsätzlich ist zu unterscheiden zwischen reinen »Einsteigerbüchern«, die wirklich nur einen Einstieg in die Materie vermitteln, und Publikationen, die ein Thema komplett abhandeln. Die erstgenannten stellen lediglich einen ersten Kontakt mit einer Software oder einer anderen Thematik her, bei tiefergehenden Problemen bieten sie allerdings keine Hilfestellung mehr. Für den Einsteiger sind jedoch durchaus auch Publikationen geeignet, die zwar eine komplexe Thematik behandeln, sich aber hervortun durch eine klare, auch für den Laien ver-

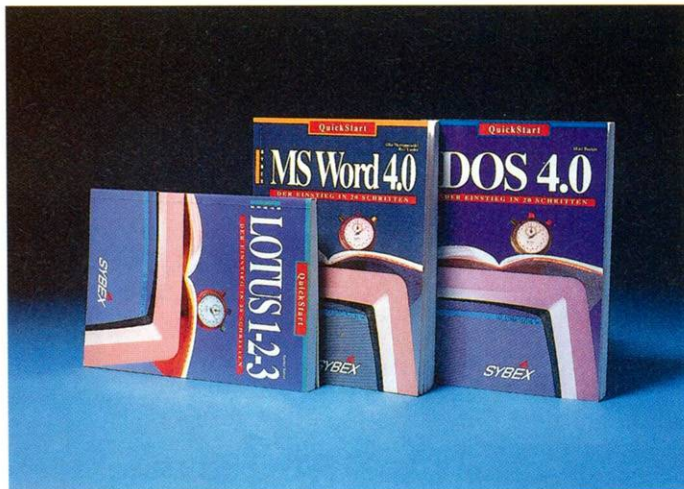


Bild 2. »Quickstart« heißt eine Buchreihe des Sybex-Verlags – sie vermittelt tatsächlich den schnellen Start in ein Programm

ständliche Sprache, einen logischen Aufbau der Thematik und ein übersichtliches Layout mit veranschaulichenden Abbildungen und klar strukturiertem Aufbau. Im Vordergrund steht die Motivation des Lesers, denn er muß die Ausdauer mitbringen, ein für ihn neues Themengebiet zu verinnerlichen. Daher sollte ein Buch nicht nur durch seinen Inhalt, sondern auch durch sein Äußeres, das Layout, fesseln.

Ein pfiffiges Layout erleichtert die Lektüre und motiviert zum Weiterlesen

Publikationen, die außer einem noch dazu sehr klein gedruckten Text keinerlei andere gestaltende Mittel verwenden, verunsichern und deprimieren gerade den Einsteiger. Information wird hier zwar auch geboten, ohne Bilder und sonstige auflockernde Elemente demotivieren solche »Bleiwüsten« jedoch mehr, als sie nützen, denn es entsteht der Eindruck einer schier überwältigenden Informationsflut.

Bilder einer Programmsituation, die einen Wiedererkennungswert aus der Praxis, das heißt vom Bildschirm her, gewährleisten, tragen zur besseren Verständlichkeit bei und bieten Abwechslung fürs Auge.

Ebenso wichtig ist eine großzügige Gliederung sowohl im optischen als auch im thematischen Sinne.

Auffallend hervorgehobene Kapitelüberschriften, Textkästen und Unterstreichungen erleichtern das Auffinden eines Kapitels, sprich Themengebietes, und erlauben ein schrittweises Abarbeiten, nach dem Motto »wieder ein Kapitel geschafft«!

Ebenfalls auflockernd wirkt die Verwendung von Sinnbildern am rechten oder linken Seitenrand, beispielsweise ein Ausrufezeichen für einen besonders wichtigen Abschnitt.

Neben den optischen Merkmalen ist natürlich die Sprache ein wichtiges Moment. Der Autor sollte sich nicht in einem fachspezifischen Fremdwörter-Chaos ergehen, sondern einfache, leicht faßbare Sätze verwenden. Das soll nicht heißen, daß ein Fachbuch keine Fachwörter verwenden darf oder sich auf sprachlichem Grundschulniveau bewegen muß. Wenn ein Sachgebiet aber bereits seiner Natur nach nicht ganz einfach ist, sollte man es sprachlich nicht zusätzlich und unnötig erschweren.

Unverzichtbar für eine schnelle Hilfe bei einer speziellen Frage ist eine ausführliches Stichwortverzeichnis, denn nichts ist ärgerlicher, als vor einem drängenden Problem zu stehen und die notwendige Erklärung erst nach langem Suchen im zugehörigen Buch zu finden.

Am Anfang der Arbeit mit dem PC steht in fast allen Fällen die Konfrontation mit dem Betriebssystem MS-DOS. Wer sich intensiver mit dieser Thematik befassen möchte, benötigt ein konkre-



Bild 3. »Kurz und bündig« erhalten Sie die wichtigsten Infos zu verschiedenen Software-Produkten vom Vogel-Verlag

tes Einsteigerbuch für diese Materie.

MS-DOS stellt den PC-Neuling vor die erste Hürde

Der Tewi-Verlag in München bietet in seinem Programm ein schön und verständlich aufgebautes Buch mit dem Titel »MS-DOS: Einfache Zugänge«. Die Publikation stammt von Robert Fürst und fällt vor allem durch ein übersichtliches, großzügiges Layout auf. Große Überschriften, viele Abbildungen und andere Layout-Mittel erleichtern den Zugang auch zu dieser Thematik. In einfachen Sätzen bringt der Autor dem Leser den Sachverhalt gut verständlich nahe. Das Buch eignet sich gleichwohl für den »absoluten« Einstei-

ger wie auch für weiterführende Zwecke.

Der Tewi-Verlag bietet für den PC-Einsteiger ganz allgemein ein gutes Konzept: Die entsprechenden Titel lauten beispielsweise: »MS-Windows: Einführung + Referenz« oder »MS-DOS 4.0/PC-DOS 4.0: Einführung + Referenz«. Diese Publikationen tragen alle oben beschriebenen, positiv ins Auge fallenden Merkmale, die zu einer guten Vermittlung einer Thematik vonnöten sind. Dazu hält sich auch der Preis, der je nach Umfang und Thematik der Bände zwischen 39 und 79 Mark schwankt, für Computer-Literatur in akzeptablen Grenzen (Bild 1). Eine Buchreihe des Sybex-Verlags gehört ebenfalls eindeutig in die Kategorie der Einsteigerbücher und wendet sich an den interessierten PC-Besitzer, der sich »auf die schnelle« in ein Programm

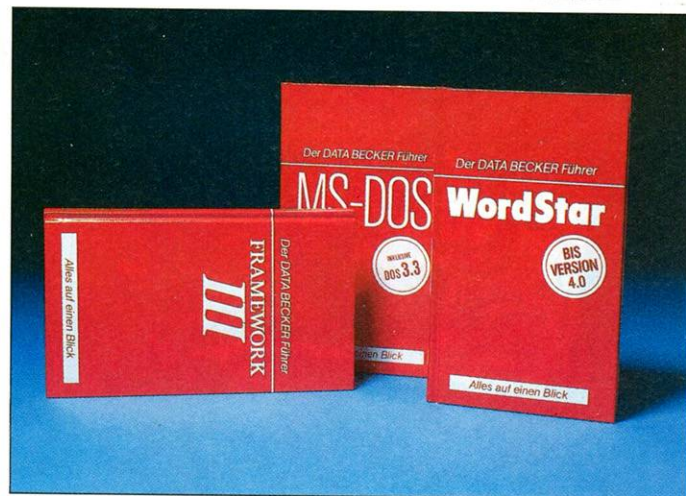


Bild 4. Die roten Führer von Data Becker eignen sich hervorragend als Nachschlagewerke

einarbeiten will. Die Reihe, bezeichnenderweise mit »Quickstart« titulierte, existiert für verschiedene Software-Produkte und vermittelt in jeweils 20 Schritten die grundlegenden Funktionen des betreffenden Programms. Die Informationen sind zwar sehr knapp gehalten, doch die übersichtliche Gestaltung und die klare »Schreibe« der betreffenden Autoren sowie der strukturierte Aufbau der Kapitel an sich gewährleisten einen sicheren Lernerfolg. Die Quickstart-Büchlein, die zirka 130 Seiten umfassen, eignen sich für den Einstieg; für das Vertiefen bestimmter Programmbereiche oder dafür, alle Funktionen bis zum letzten auszuschöpfen, müssen Sie jedoch zu weiterführender Literatur greifen. Für 19,80 Mark ist jedoch die Anschaffung immer rentabel, weil Sie hiermit nicht zuletzt auch das nötige Grundwissen erhalten, um selbst eine weitere Auswahl an vertiefender Literatur zu treffen (Bild 2).

Für PC-Anwender, die knappe Information wünschen, um so bald wie möglich, ohne langes Vorstudium, mit der eigentlichen Arbeit zu beginnen, empfiehlt sich die Reihe »kurz und bündig« des Vogel-Verlags. Auch die Bücher dieser Reihe sind gut strukturiert, vermitteln aber lediglich das Grundwissen zu den entsprechenden Programmen. Lösungen für kompliziertere Problemstellungen werden Sie hier nicht finden, dafür aber eine gute Anleitung für die ersten Schritte. Der Preis bewegt sich hier zwischen 30 und 40 Mark (Bild 3). Die Reihe der

Data-Becker-Führer – rote Büchlein, die die verschiedenen Funktionen einer Software vorstellen – eignen sich dagegen zum direkten Lernen nicht so gut, sind jedoch bereits minimale Vorkenntnisse vorhanden, stellen sie ein gutes informatives Nachschlagewerk dar. Sie werden für die verschiedensten Programme angeboten. Für den Einsteiger besonders lohnenswert ist der Band zu MS-DOS, da es gerade am Anfang Schwierigkeiten bereitet, die genaue Syntax eines Befehls im Kopf zu behalten. Im Nu ist dank eines sehr ausführlichen Stichwortverzeichnisses der gesuchte Begriff gefunden und kann gleich ausprobiert werden. Sie müssen zwischen 20 und 30 Mark pro Exemplar anlegen (Bild 4).

Ähnlich gelagert und auch sehr brauchbar ist eine Buchreihe aus dem Hause Sybex, die sich »Ratgeber« nennt. Auch darin werden die einzelnen Funktionen verschiedener Software kurz beschrieben. Ein umfangreicher Index und ein übersichtlich und strukturiert aufgebautes Inhaltsverzeichnis sorgen für einen schnellen Sucherfolg. Die Ratgeberreihe kostet pro Stück zirka 40 bis 50 Mark (Bild 5). Natürlich basiert jede spezielle »Empfehlung« auf dem subjektiven Urteil des »Empfehlenden«, doch steht es jedem frei, sich in der Buchhandlung ein eigenes Urteil zu bilden. Die genannten Kriterien, die ein für den Einsteiger geeignetes Buch erfüllen soll, machen es jedoch einfacher, die richtige Wahl zu treffen.

(hi)

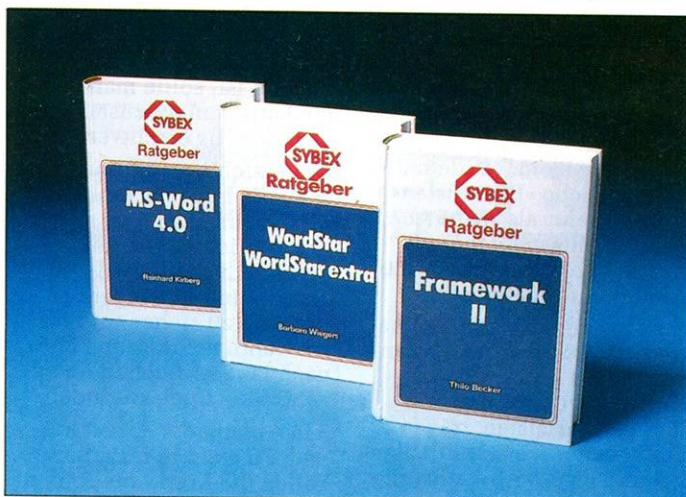


Bild 5. Der gute Rat aus der Sybex-Reihe »Ratgeber« ist zwar nicht ganz billig, doch Sie erhalten dafür sehr ausführliche Informationen

Das Geheimnis der Datenspeicherung

Tagtäglich werden Tausende von Disketten zum Speichern von Daten verwendet, doch kaum einer weiß, auf welche Weise die Daten und Programme auf die runden Scheiben gelangen. Blicken Sie einmal hinter die Geheimnisse der Datenspeicherung.

Diskettenlaufwerke sind für die Arbeit mit einem PC unverzichtbar. Das liegt daran, daß der Computerspeicher, in den Sie bei der Benutzung von Anwendungsprogrammen Texte, Daten, Programme etc. eingeben, »flüchtig« ist. Das heißt, daß alles (oder besser, fast alles), was darin gespeichert ist, verschwindet, sobald Sie den Rechner ausschalten.

Die einzige Möglichkeit, die Daten, die Sie mühsam in den Rechner eingegeben haben, zu sichern, besteht darin, auf ein »nicht flüchtiges« Speichermedium zurückzugreifen. Ein solches Speichermedium ist die Diskette. Auf einer Diskette werden die elektronischen Impulse, die den Speicherinhalt eines Computers repräsentieren, in Form magnetischer Signale gespeichert.

Es gibt verschiedene Diskettengrößen, die auf PCs eingesetzt werden können. Die 5 1/4-Zoll-Disketten, die ursprünglich ausschließlich Verwendung fanden, werden zunehmend durch die neueren 3 1/2-Zoll-Disketten abgelöst. Die Bezeichnung dieser Diskettentypen bezieht sich auf den Durchmesser der Kunststoffscheibe, die in den rechteckigen Schutzhüllen verborgen ist. Die Technik der Informationsspeicherung ist jedoch bei beiden Diskettentypen gleich.

Die Diskette selbst ist eine mit einer magnetischen Oxidschicht versehene, flexible und dünne Plastikscheibe. Sie ist umhüllt von einer rechteckigen Schutzhülle, die mehr oder weniger stabil ist. Bei 5 1/4-Zoll-Disketten besteht diese Schutzhülle aus biegsamen Kunststoff, bei 3 1/2-Zoll-Disketten hingegen ist die Schutzhülle aus festem Plastik.

Die elektronischen Informationen werden mit einem Schreib-/Lesekopf auf die Diskette geschrieben und von ihr gelesen. Dieser Kopf wird über dem Schreib-/Leseschlitz der Diskette positioniert. Bei 5 1/4-Zoll-Disketten ist der Schreib-/Leseschlitz ungeschützt, bei 3 1/2-Zoll-Disketten ist er durch eine seitenverschiebbare Metallfläche vor Staub geschützt. Im Diskettenlaufwerk wird dieser Schutz dann automatisch geöffnet, damit auf die Diskette geschrieben und von der

Diskette gelesen werden kann.

In der Mitte der Diskette befindet sich ein Loch, um das die Diskette rotiert. Bei 3 1/2-Zoll-Disketten ist der Rand des Lochs mit Metall verstärkt. Ferner finden sich zwei weitere Öffnungen auf der Diskette. Oben rechts in der Ecke ist die »Schreibschutzkerbe« zu sehen (Bild 1). Ähnlich wie Musikkassetten lassen sich nämlich auch Disketten davor schützen, beschrieben zu werden. Bei 5 1/4-Zoll-Disketten geschieht das dadurch, daß ein lichtundurchlässiger Klebestreifen über die Kerbe geklebt wird. Bei 3 1/2-Zoll-Disketten ist ein Kunststoffschieber, der sich in vergleichbarer Position befindet wie die Kerbe auf der 5 1/4-Zoll-Diskette, so zu positionieren, daß Sie durch den kleinen Schlitz hindurchsehen können.

Ein Aufkleber schützt Ihre Disketten vor unbeabsichtigtem Überschreiben

Eine Lichtschranke tastet dann diese Kerben der beiden Diskettentypen bei Betrieb des Laufwerks ab. Bei der 5 1/4-Zoll-Diskette wird ein Schreibversuch abgebrochen, wenn diese Lichtschranke unterbrochen wird, bei der 3 1/2-Zoll-Diskette hingegen dann, wenn die Lichtschranke nicht unterbrochen wird.

Ein weiteres kleines Loch findet sich unmittelbar neben der größeren Öffnung. Dieses sogenannte »Indexloch« dient zur Überprüfung der Umdrehungsgeschwindigkeit der Diskette. Um eine Diskette zu lesen oder zu beschreiben, wird sie ins Diskettenlaufwerk geschoben. Das Diskettenlaufwerk ist dafür zuständig, Daten von der Diskette zu lesen oder auf die Diskette zu schreiben.

Der wesentliche Unterschied zwischen 5 1/4- und 3 1/2-Zoll-Laufwerken besteht in der Informationsdichte, mit der sich die Disketten beschreiben lassen. Bei PC-Systemen liegt die maximale Speicherkapazität für 5 1/4-Zoll-Disketten bei 360 KByte (oder 1,2 MByte mit einem speziellen

Laufwerk für den AT), bei 3 1/2-Zoll-Disketten normalerweise aber bei 720 KByte (oder 1,44 MByte mit einem speziellen Laufwerk), also ungefähr der doppelten Speicherdichte.

Vor der Verwendung muß eine Diskette mit Hilfe des Diskettenlaufwerks formatiert werden. Bei diesem Formatieren wird die Diskette in Spuren und Sektoren unterteilt. Normale 5 1/4-Zoll-Disketten arbeiten heutzutage mit zwei Diskettenseiten, das heißt, es werden tatsächlich beide Seiten der Disketten mit Daten beschrieben und nicht nur eine. Auf jeder Seite finden 40 Spuren Platz, von denen jede 9 Sektoren umfaßt und jeder Sektor 512 Bytes. Allerdings gibt es für den AT auch spezielle Laufwerke, auf denen Sie bis zu 1,2 MByte an Speicherraum für Daten unterbringen können. Diese Diskettenlaufwerke verwenden 80 Spuren pro Diskettenseite (anstelle von 40) und 15 Sektoren pro Spur (anstelle von 9).

Bild 2 gibt die Aufteilung der Diskette in Seiten, Spuren und Sektoren am Beispiel einer 5 1/4-Zoll-Diskette wieder. 3 1/2-Zoll-Disketten werden meistens mit 80 Spuren pro Seite formatiert, umfassen also die doppelte Speicherkapazität einer 5 1/4-Zoll-Diskette.

Diskettenlaufwerke, die mehr als eine Diskettenseite (also zwei Seiten) ansprechen können, verfügen immer über zwei Schreib-/Leseköpfe anstelle von einem. Dadurch lassen sich beide Seiten der Diskette verwenden, ohne die Diskette umdrehen zu müssen. Die üblichen 360-KByte- oder 720-KByte-Laufwerke für PCs enthalten ausnahmslos je zwei Schreib-/Leseköpfe.

Es lassen sich auch mehrere Diskettenlaufwerke an den PC anschließen. Damit der PC die verschiedenen Laufwerke unterscheiden kann, tragen sie jeweils einen Kennbuchstaben. Das erste Diskettenlaufwerk heißt »Laufwerk A«, das zweite »Laufwerk B« etc.

Bei PCs, bei denen die Laufwerke nebeneinander eingebaut sind, erhält das linke Laufwerk (vom Anwender aus gesehen) in der Regel die Bezeichnung »Laufwerk A« und das rechte die Bezeichnung »Laufwerk B«. An PCs, bei denen die Laufwerke übereinander eingebaut sind, wird meist das obere Laufwerk »Laufwerk A« und das untere »Laufwerk B« benannt. Festplattenlaufwerke werden meist in Folge mit den Buchstaben »C« und »D« etc. bezeichnet.

Die Daten selbst werden in sogenannten »Dateien« auf der Diskette abgelegt. Eine Datei ist ein Dokument (zum Beispiel ein Text, eine Kalkulation oder ein Programm), das über den »Dateinamen« angesprochen

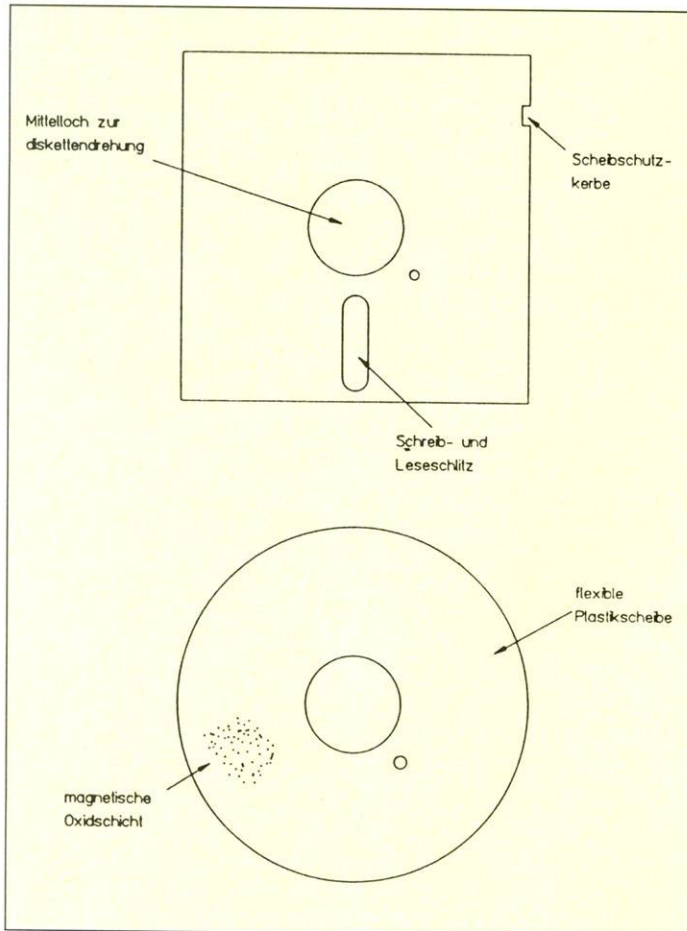


Bild 1. So ist eine Diskette im Detail aufgebaut

wird und eine bestimmte Anzahl von Daten enthält. Wird beispielsweise ein Text mit Hilfe eines Textverarbeitungsprogramms unter dem Namen »Text1« gespeichert, so existiert nach diesem Vorgang eine Datei mit dem Namen »Text1« auf der Diskette, die die einzelnen Buchstaben, Ziffern, Sonderzeichen etc. des Textes enthält. Dateinamen haben unter MS-DOS eine feste Struktur. Sie bestehen aus maximal acht alphanumerischen Zeichen, einem darauffolgenden Punkt und danach wieder aus maximal drei alphanumerischen Zeichen. Die ersten acht Zeichen werden meist als »Dateiname«, die drei Zeichen hinter dem Punkt als »Namenserweiterung« oder »Extension« (das ist der englische Begriff) bezeichnet. Zulässige Dateinamen sind als Beispiele in der Tabelle wiedergegeben.

Das Betriebssystem speichert Dateien in den Sektoren der Diskette. Jede Datei wird dabei auf mehrere einzelne Sektoren verteilt. Nun soll aber der Platz der Diskette optimal ausgenutzt werden, damit sich möglichst viele Dateien auf einer einzigen Diskette unterbringen lassen. Dazu ist eine sogenannte dynamische Verwaltung der Sektoren seitens des Betriebssystems MS-DOS notwendig. Tritt beispielsweise der Fall ein, daß alle Sektoren einer Diskette belegt

sind, dann werden drei einzelne Dateien der Diskette gelöscht, die alle genau einen Sektor (also 512 Zeichen) belegen. Jede dieser Dateien soll sich in unserem Beispiel an völlig anderen Positionen auf der Diskette befinden.

Da die Dateien gelöscht wurden, stehen der Diskette nun wieder drei leere Sektoren zur Verfügung. Wenn Sie nun eine Datei speichern, die genau drei Sektoren lang ist, so kann das Betriebssystem diese Datei nicht in »hintereinanderliegenden« Sektoren der Diskette abspeichern, sondern

muß die drei vorhandenen, weit auseinanderliegenden, freien Sektoren benutzen. Deshalb muß das Betriebssystem genau Buch darüber führen, welcher Platz auf der Diskette von welcher Datei belegt ist. Mit dem Betriebssystem MS-DOS werden die Dateien geschickt organisiert.

MS-DOS kann, wie oben schon beschrieben, mehrere Diskettenlaufwerke verwalten. Es kann jedoch immer nur jeweils ein Laufwerk zu einem Zeitpunkt angesprochen werden. Da das Betriebssystem MS-DOS ein diskettenfixiertes Betriebssystem ist, wirken sich – wie oben erläutert – fast alle Befehle auf die Diskettenlaufwerke aus. Damit MS-DOS weiß, welches Laufwerk es für die Diskettenzugriffe verwenden soll, ist immer ein sogenanntes »Standardlaufwerk« voreingestellt. Lassen Sie bei den Befehlen oder Programmen eine explizite Laufwerksangabe weg, so wird immer das Standardlaufwerk verwendet.

MS-DOS muß in der Lage sein, die unterschiedlichen Sektoren, Spuren, Köpfe oder Platten anzusprechen, um die Datenblöcke der Diskette verwalten zu können. Ein wichtiges Bindeglied dazu ist ein sogenannter »Disketten-treiber«. Das ist ein Teil des Betriebssystems, der dazu dient, dem Betriebssystem bei der Benutzung von Disketten- und Festplattenlaufwerken »unter die Arme zu greifen«.

Normalerweise müßte MS-DOS bei jedem Diskettenzugriff genau wissen, welchen Schreib-/Lesekopf, welche Platte, welche Spur und welchen Sektor es ansprechen möchte. Das Betriebssystem müßte also die Disketten über ihre physikalischen Komponenten adressieren. Das erfordert natürlich sehr großen Verwaltungsaufwand. Deshalb wurde das Konzept des Diskettentreibers entwickelt. Der Diskettentreiber ist die Schnittstelle zwischen dem Betriebssystem und dem Diskettenadapter. MS-DOS muß

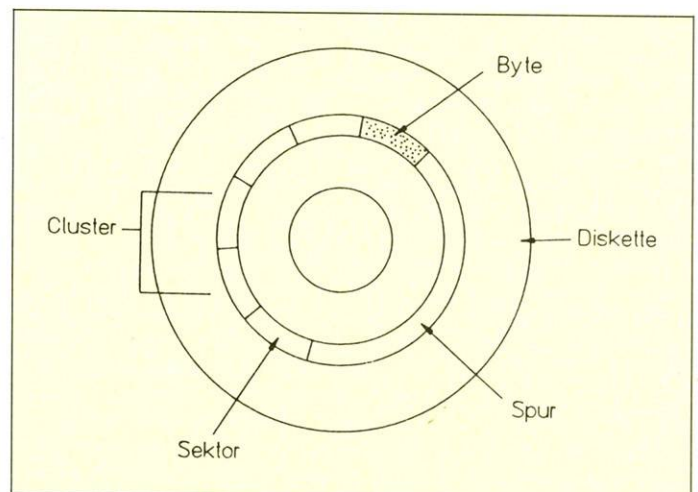


Bild 2. Die Diskette ist intern in Spuren und Sektoren unterteilt, die die Daten aufnehmen

nicht mehr die physikalischen Sektoren adressieren, sondern kann auf sogenannte »logische« Sektoren zurückgreifen. Die Umwandlung eines logischen Sektors in einen adressierbaren physikalischen Sektor der Diskette erledigt der Diskettentreiber. Dazu betrachtet MS-DOS die Diskette also nicht in ihren physikalischen Größen, sondern faßt sie als eine (je nach Diskettengröße) bestimmte Anzahl von logischen Sektoren auf. Wenn ein Sektor gelesen werden soll, schickt MS-DOS dessen Nummer an den Diskettentreiber, der die Nummer dann in die tatsächliche physikalische Position auf der Diskette umwandelt.

Der Diskettentreiber greift MS-DOS hilfreich unter die Arme

Stellen Sie sich als Beispiel einmal ein großes Bucharchiv vor. In diesen Archiven werden die einzelnen Buchpositionen meist nicht durch ihre tatsächliche, physikalische Lage (also zum Beispiel durch »zweite Etage, siebter Gang von rechts, drittes Regal, vierte Reihe, das dritte Buch von links«), sondern durch eine logische Buchnummer identifiziert. Das obige Buch könnte dann vielleicht die logische Nummer 23435 besitzen. Im Verzeichnis des Archivs wird dann hinter einem Titel in der Regel diese logische Buchnummer zu finden sein. Fordert ein Benutzer nun dieses Buch an, so wandelt der Sachbearbeiter die logische Buchnummer mittels einer Liste in die tatsächliche physikalische Position im Archiv um. Nach dem gleichen Prinzip gehen auch Diskettentreiber und Betriebssystem vor. MS-DOS fordert einen logischen Sektor der Diskette an, der Diskettentreiber wandelt dessen Nummer in eine physikalische Sektornummer um und steuert den Diskettenadapter, um den Sektor zu lesen oder zu beschreiben. MS-DOS braucht sich demnach nicht darum zu kümmern, welchen Aufbau eine Diskette hat; es betrachtet einfach jede Diskette als Gesamtheit fortlaufender Sektoren.

Diese Sektoren werden im wesentlichen über zwei »Tabellen« verwaltet. Die wichtigste Tabelle im Zusammenhang mit Disketten ist das »Inhaltsverzeichnis« der Diskette. Im Inhaltsverzeichnis ist gespeichert, wie viele Dateien auf einer Diskette abgelegt sind, welche Namen diese Dateien besitzen, wie groß sie sind und wann sie angelegt wurden. Das Inhaltsverzeichnis wird oft auch als »Directory« bezeichnet. Dieses wird

ebenfalls auf der Diskette gespeichert, damit es auch nach dem Ausschalten des PC nicht verlorengeht. Mit den Festplatten tauchte aber ein Problem auf, das sich mit den bis dahin verfügbaren Versionen von MS-DOS nicht lösen ließ. Eine Festplatte kann aufgrund ihrer enormen Speicherkapazität eine Vielzahl von Dateien aufnehmen. Eine solch große Anzahl von Dateien in einem einzigen Inhaltsverzeichnis unterzubringen, ist sehr aufwendig und für den Benutzer außerdem sehr umständlich und unübersichtlich. Aus diesem Grund weist MS-DOS ab der Version 2.0 eine Organisationsform auf, durch die sogenannte »Unterverzeichnisse« oder »Subdirectories« angelegt werden können.

Anhand einer Kartei läßt sich dieser Sachverhalt gut vor Augen führen. Statt alle Karteikarten in einer einzigen großen Schublade unterzubringen, werden sie auf verschiedene Schubladen verteilt. Das gestattet einen schnelleren und bequemeren Zugriff. Ähnlich haben Sie sich das Verfahren bei MS-DOS vorzustellen.

Auch hier können – mit speziellen Kommandos – Unterverzeichnisse angelegt oder entfernt werden. Diese Unterverzeichnisse können wiederum Unterverzeichnisse enthalten etc. Es lassen sich ganze »Verzeichnisbäume« anlegen, in denen die Dateien untergebracht werden. Das »oberste« Verzeichnis einer Diskette heißt deshalb auch »Stammverzeichnis«.

Die Unterverzeichnisse erhalten symbolische Namen, je nachdem, welche Dateien sich darin befinden. Den Weg zu den Unterverzeichnissen und ihren Dateien zeigen Ihnen sogenannte »Pfade« (englisch »Path«). Der Pfad (auch »Dateipfad« genannt) gibt

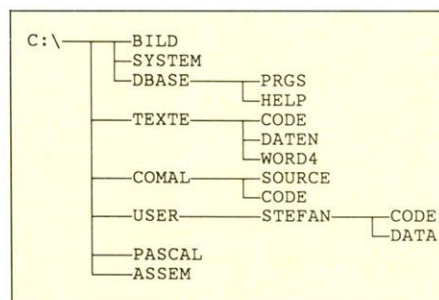


Bild 3. Die Dateien und Unterverzeichnisse sind im Verzeichnisbaum klar strukturiert

Name	DOS speichert sie als
TEXT.TXT	TEXT .TXT
Buch	BUCH .
datei1.DAT	DATEI1 .DAT
TEST.w	TEST .W

Tabelle. MS-DOS gestattet verschiedene Formen von Dateinamen

den »Weg« zu einer Datei auf der Diskette oder Festplatte an. Wenn auf einer Diskette im Laufwerk A zum Beispiel ein Unterverzeichnis »Texte« existiert und dieses Unterverzeichnis wieder ein Unterverzeichnis namens »Daten« enthält, so sieht dies als »Verzeichnisbaum« grafisch dargestellt aus wie in Bild 3.

»Pfade« zeigen MS-DOS den Weg zu einer Datei

Wenn im Unterverzeichnis »Daten« eine Datei »kreis.txt« existiert, so muß MS-DOS über die Verzeichnisse »Texte« und »Daten« zu dieser Datei »finden«. Beim »Weg« dorthin handelt es sich um besagten Pfad. In Computerschreibweise werden die einzelnen »Kreuzungen« eines Pfades durch ein spezielles Zeichen, den Backslash »\«, kenntlich gemacht. Der komplette Dateipfad für die oben aufgeführte Datei »kreis.txt« würde also folgendermaßen aussehen:

a: \texte \daten \kreis.txt

Das bedeutet (gelesen von rechts nach links): »Die Datei, die ich meine, hat den Namen »kreis.txt« und befindet sich im Verzeichnis »Daten«, das im Verzeichnis »Texte« im Laufwerk A untergebracht ist.«

Neben dem Inhaltsverzeichnis existiert noch die sogenannte »Dateizuordnungstabelle«. Diese Tabelle bildet die Schnittstelle zwischen dem Inhaltsverzeichnis und den logischen Sektoren der Diskette. In der Dateizuordnungstabelle (in der Fachsprache »FAT« genannt, was für »File Allocation Table« steht) ist aufgezichnet, welche Sektoren zu einer bestimmten Datei gehören. Intern wird allerdings nicht jeder Sektor einzeln angesprochen, sondern es werden mehrere Sektoren zu einem »Cluster« zusammengefaßt.

Wenn MS-DOS also eine Datei lesen oder schreiben möchte, sucht es zunächst im Inhaltsverzeichnis nach dem Namen der Datei. Dann liest es aus der Dateizuordnungstabelle nacheinander alle Sektoren, die zu der Datei gehören. Jetzt kann die Datei gelesen oder geschrieben werden. Zu jedem Disketten- oder Festplattenlaufwerk gehören also für MS-DOS die logischen Sektoren, das Inhaltsverzeichnis und die Dateizuordnungstabelle. Durch diesen durchdachten, logischen Aufbau ist Ihr Betriebssystem in der Lage, sämtliche Dateien zu jedem Zeitpunkt wiederzufinden.

(Martin Althaus/hi)

Alles über Ihre Festplatte

Einrichten, Formatieren, Partitionieren, Optimieren, Sichern – für den PC-Einsteiger alles Fremdworte in Sachen Festplatte. Da die Erläuterungen in den Benutzerhandbüchern meist bei der Bedienung des einen oder anderen Hilfsprogramms enden und sich über Hintergründe in der Regel ausschweigen, verrät Ihnen dieser Beitrag spezielle Kniffe für den Umgang mit Ihrer Festplatte.

In der Regel erstehen Sie Ihren Computer bereits inklusive Festplatte, so daß der Einbau eines solchen Laufwerks für uns kein Thema ist. Unsicher ist jedoch, ob die Platte auch schon eingerichtet ist.

Im Gegensatz zu Disketten, die schnell und einfach mit dem Aufruf von »format« zur Aufnahme von Daten vorbereitet werden, müssen Sie zur Inbetriebnahme einer Festplatte mehr Zeit investieren. Neben MS-DOS erlaubt Ihnen das ROM-BIOS Ihres Computers, auch andere Betriebssysteme, beispielsweise CP/M-86, auf der Festplatte einzurichten. Da die verschiedenen Systeme und Ihre Versionen weder durch ein und denselben Entwickler, noch zur gleichen Zeit entstanden, ist Kompatibilität bezüglich der Festplattenorganisation ein Fremdwort geblieben. Wie aber kann dieser Gordische Knoten gelöst werden?

Beim ersten Arbeitsgang, der physikalischen Formatierung, erfolgt unabhängig vom späteren Betriebssystem eine Aufteilung der Festplatte in Sektoren. Jeder Sektor erhält zur Identifizierung eine Kennung, auch »Adreßmarke« genannt. Sie beinhaltet Spur-, Kopf- und Sektornummer. Die Computerhersteller liefern in der Regel eingebaute Festplatten bereits physikalisch formatiert aus, trotzdem finden sich unter den mitgelieferten Hilfsprogrammen auch Low-Level-Formatter.

Der zweite Schritt auf dem Weg zur vollen Nutzung des Massenspeichers Festplatte ist die Partitionierung. Mit Hilfe des Programms »fdisk« legen Sie die Bereiche für die verschiedenen Betriebssysteme fest. Zusätzlich benötigt der Computer die Information, von welcher Partition er beim Einschalten das Betriebssystem laden soll. Alle Informationen werden im ersten Sektor der Platte, dem Partitionssektor, untergebracht. Sie können sich vorstellen, was passiert, sollte dieser Sektor einmal nicht mehr gelesen werden können...

Legen Sie also eine Boot-Diskette an (»format a: /4 /s«), auf der Sie alle Befehle unterbringen, die Sie für die normale Arbeit mit Ihrem Rechner

brauchen. Außerdem müssen darauf auch die Befehle »format« und »fdisk« enthalten sein, damit Sie die Laufwerke überhaupt einrichten können. Benutzen Sie eine ältere MS-DOS-Version als die Version 3.3, dann ist der Spaß für Sie hier zu Ende. Das Einrichten einer zweiten Festplattenpartition, die Sie unter MS-DOS ansprechen können, funktioniert erst ab dieser Version oder mit verschiedenen Hilfsprogrammen.

Mit »fdisk« teilen Sie Ihre Festplatte in verschiedene Partitionen auf

Starten Sie »fdisk«, vergegenwärtigen Sie sich aber, daß dieser Befehl sämtliche Daten Ihrer Festplatte quasi löscht. Wählen Sie Menüpunkt 1: »Erstellen einer MS-DOS-Partition«. Sie wollen eine »primäre MS-DOS-Partition« generieren, wählen also wiederum Menüpunkt 1. Jetzt fragt Sie »fdisk«, ob die erste Partition die maximale Größe von 32 MByte haben

soll. Soll die Partition kleiner sein, zeigt Ihnen »fdisk« nun den gesamten Platz der Platte gemessen in »Zylindern« (Bild 2).

Um sich weitschweifende Ausführungen zur Festplattegeometrie an dieser Stelle zu ersparen (die kommen später), verwenden Sie einen einfachen Dreisatz. Errechnen Sie, wieviel die gewünschte Partitionsgröße prozentual zur Kapazität der gesamten Festplatte ausmacht. Der Prozentwert wird nun mit der maximal verfügbaren Zylinderzahl multipliziert. Das Ergebnis ist der Wert, den Sie bei der Partitionsgröße von »fdisk« angeben. Es sollten jedoch dabei nicht mehr als 32 MByte für die zweite Partition übrig bleiben, da Sie sonst eine dritte Partition einrichten müssen. Ist die Festplattenkapazität höher als 64 MByte, können Sie das nicht verhindern, aber wer hat das schon?

Nach dieser Aktion verlangt »fdisk«, den Rechner zu booten. Dazu muß die zuvor generierte Systemdiskette im Diskettenschacht liegen. Starten Sie »fdisk« erneut, und richten Sie eine »erweiterte Partition« ein, der Sie den übrigen Platz überantworten. Diese Partition nennt »fdisk« fortan »logisches Laufwerk D:«. Verlassen Sie das Programm, und lassen Sie den Rechner booten. Alle Partitionen sind nun eingerichtet.

Mit der Eingabe des Befehls »format c:/v /s« unter DOS, den Sie jetzt eingeben, geschieht folgendes auf der Festplatte:

- Eintragung des Boot-Sektors
- Dateizuordnungstabellen (FATs) und das Hauptinhaltsverzeichnis (Root Directory) werden an der richtigen Stelle in der DOS-Partition platziert
- Übertragung der Systemdateien auf die Festplatte.

Die kleinste Zuweisungseinheit, also die kleinste Einheit, in der das Betriebssystem MS-DOS auf der Festplatte Reservierungen für Daten vornehmen kann, heißt »Cluster« oder auch »Block«. Diese Cluster beinhalten je nach Typ eine unterschiedliche Anzahl von Sektoren und weisen verschiedene Blockgrößen auf.

Unter MS-DOS 3.xx hat sich bei der Cluster-Größe für 10-MByte-Platten nichts geändert, die Cluster-Größe bei 20-MByte-Platten reduzierte sich jedoch auf 2 KByte. Dadurch wurde zwar eine effektivere Nutzung des Speicherplatzes erreicht, die Kompatibilität zur früheren Version war jedoch verloren. Auch die Systemdateien sind je nach DOS-Version unterschiedlich und können nicht ausgetauscht werden. Für die Praxis bedeutet dies, daß ein Zugriffsversuch von DOS 3.xx auf eine mit DOS 2.10 installierte Festplatte oder um-

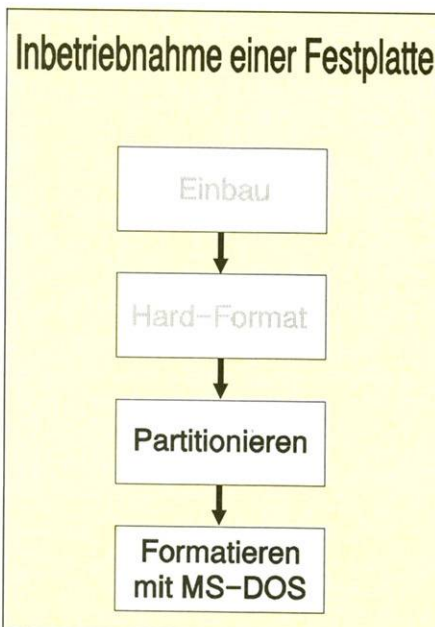


Bild 1. Diese Schritte sind nötig, um eine neue Festplatte einzurichten

gekehrt damit endet, daß gespeicherte Daten auf der Festplatte aufgrund der unterschiedlichen Cluster-Art und -Größe nicht gefunden oder überschrieben werden. Letzteres hat unter Umständen den Verlust aller Daten der Festplatte zur Folge, da kein geordneter Zugriff mehr erfolgen kann.

Mit DOS 3.3 brach ein neues Zeitalter an: Es »sagte« an der 32-MByte-Grenze

Bis hin zu DOS 3.2 war es mit »fdisk« nicht möglich, mehrere DOS-Partitionen auf einer Festplatte einzurichten. Bei einer maximalen Kapazität von 32 MByte ließ sich damit auch leben. In der Zwischenzeit wurden jedoch Festplatten mit erheblich höherer Kapazität erschwinglich, und es mußte Abhilfe geschaffen werden. Während sich bei Microsoft lange Zeit diesbezüglich nichts rührte, tauchten auf dem Software-Markt Hilfsprogramme auf, mit deren Hilfe sich bis zu 16 Partitionen pro Platte einrichten ließen. Erst MS-DOS 3.3 erlaubt nun bis zu 24 logische Laufwerke. Die ersten 32 MByte einer Festplatte (physikalisch gesehen) werden fortan »Primary-DOS-Partition« genannt, alles, was danach kommt, »Extended-DOS-Partition«.

Alle in diesem zweiten Teil eingerichteten logischen Laufwerke dürfen jeweils bis zu 32 MByte Speicherplatz aufweisen. Das Programm »fdisk« der DOS-Version 3.3 wurde völlig umgestrickt, und es versteht sich von selbst, daß eine Kompatibilität zwischen Laufwerken mit dieser und niedrigeren Versionen ebenfalls nicht mehr gegeben ist.

Erst seit dem Erscheinen der neuesten MS-DOS-Version 4.0 ist die Kapazitätsbeschränkung eines Lauf-

Bild 2. Mit »fdisk« von MS-DOS teilen Sie den Platz auf Ihrer Festplatte in einzelne Laufwerke ein

```

Erstellung von DOS-Partitionen
Momentanes Festplattenlaufwerk: 1

1. Erstellen der primären
   DOS-Partition
2. Erstellen einer erweiterten
   DOS-Partition

Wählen Sie eine Option: [1]

ESC, um zurückzukehren!
  
```

werks auf 32 MByte beseitigt. Bis zu einigen Gigabyte darf Ihr Laufwerk C nun groß sein. Der Schritt des Partitionierens jedoch fällt nicht weg, sondern ist lediglich vereinfacht. »fdisk« stellt die Frage, ob »aller verfügbarer Platz für das Laufwerk verwendet werden soll«, man antwortet mit »Ja«, und die Sache ist erledigt.

Mit der MS-DOS-Version 4.01 hat das Laufwerkchaos endlich ein Ende

In der Regel liegen die Cluster einer Datei in einer ununterbrochenen Reihenfolge – falls es der Speicherplatz zuläßt. Werden große Dateien sehr oft erneuert, sind nicht mehr genügend freie Cluster in einer Reihe vorhanden, und ein Teil der Datei wird in einer anderen Cluster-Folge irgendwo auf der Platte abgelegt. Dadurch verlangsamt sich natürlich die Arbeitsgeschwindigkeit der Festplatte, da pro Datei mehr Zugriffe als üblich durchgeführt werden müssen. Geradezu gefährlich wird es, wenn Sie versehentlich eine Datei gelöscht haben und nun versuchen, sie mit einem Hilfsprogramm wiederherzustellen. Daher empfiehlt es sich, einen

Disk-Optimizer zu starten, der die physikalisch richtige Reihenfolge der Dateien wieder in Ordnung bringt. Solche Programme gibt es bei den »PC-Tools«, bei »Norton Computing« und sogar bereits in der Shareware-Szene. Die Shareware-Programme sind jedoch weniger verläßlich und nur Besitzern guter Sicherungskopien zu empfehlen.

Es ist nicht nur deshalb ratsam, von Zeit zu Zeit Ihre gesamte Festplatte (mit allen eventuell vorhandenen Partitionen) zu sichern. Legen Sie dazu eine genügende Anzahl Disketten bereit. Nutzen Ihre Diskettenlaufwerke 1,2-MByte-Disketten, dann verwenden Sie High-Density-Disketten und formatieren Sie mit höchstmöglicher Kapazität. Ist nichts anderes vereinbart, versucht der MS-DOS-Befehl »format« ohnedies mit höchster Kapazität zu formatieren. Ein vorhergehendes Formatieren ist eigentlich nur dann wichtig, wenn Sie mit Disketten einfacher oder doppelter Schreibdichte, aber einem 1,2-MByte-Laufwerk arbeiten.

Sie können nämlich mit dem Programm »backup« (aus dem MS-DOS-Fundus) das Formatieren der Sicherungsdisketten befehlen, doch ist dabei kein Parameter zugelassen, der die Kapazität der Disketten angibt. Daher formatiert »backup« immer mit

Anwendungen gesucht

Haben Sie eine schnelle und komfortable Textverarbeitung geschrieben? Oder eine Datenbank mit frei wählbaren Eingabemasken? Eventuell ein Zeichenprogramm? Wenn Ihr Programm mit professionellen Programmen konkurrieren kann, ist es mit Sicherheit zu umfangreich, um als Listing in der DOS International zu erscheinen. Zögern Sie deshalb nicht, schicken Sie uns unverbindlich die aktuelle Version Ihres Programms und einen Ausdruck der Dokumentation. Eventuell wird Ihr Programm der Verkaufsschlager im Rahmen der neuen DOS-Software-Schiene. Falls Sie sich vorweg informieren möchten: Zwischen 15.00 und 17.00 Uhr stehen wir Ihnen an allen Werktagen unter der Rufnummer (089) 4 61 00 10 zur Verfügung.

Schicken Sie Ihre Unterlagen an folgende Adresse:
 Redaktionsbüro Everts & Hagedorn
 »DOS Software«
 Hans-Stießberger-Str. 3
 8013 Haar/München

Rund um die Festplatte

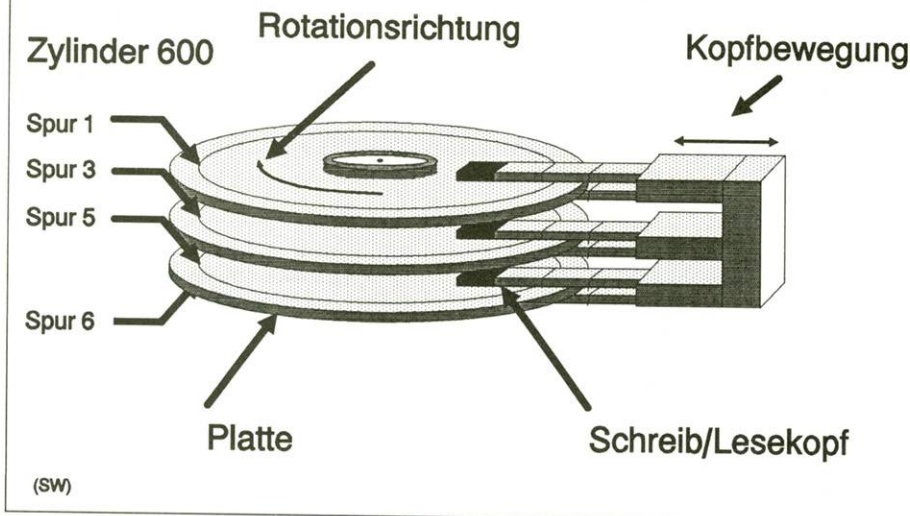


Bild 3. Ein Eindruck vom Innenleben Ihrer Festplatte

der maximal möglichen Kapazität. Spätestens beim ersten zeitraubenden Versuch, eine 360-KByte-Diskette auf 1,2 MByte zu formatieren, gibt »backup« auf.

Verwenden Sie die »richtigen« Disketten, starten Sie »backup« mit der Anweisung

```
backup c:\ a: /s /f
```

Die erste Angabe ist das Laufwerk und der Pfad, dessen Daten auf das Ziellaufwerk (zweite Angabe) gesichert werden sollen. Nach »/s« weiß »backup«, daß es auch alle Verzeichnisse und deren Unterverzeichnisse mitzusichern hat. Der Parameter »/f« gibt Anweisung, die Sicherungsdisketten vor der Übertragung zu formatieren. Dazu muß sich aber der Befehl »format« im Zugriffsbereich befinden. Entweder er ist im aktuellen Verzeichnis, oder das Verzeichnis, in dem er sich befindet, ist in der Umgebungsvariablen »path« erwähnt.

Die Datensicherung mit »backup« ist zwar etwas umständlich, aber dafür ist das Programm in MS-DOS enthalten

»backup« sichert so alle Dateien, doch spätestens beim zweiten Backup ist dies nicht mehr nötig, schließlich haben sich viele Dateien seit der letzten Sicherheitskopie nicht mehr geändert. Daher vermerkt »backup« im sogenannten »Archivbit«, ob die Datei bereits gesichert wurde. MS-

DOS dagegen löscht dieses Bit bei jeder Änderung in der Datei und »merkt« sich diese damit.

Den Wunsch, nur geänderte und neue Dateien zu sichern, können Sie mit dem Parameter »/m« äußern. In diesem Fall prüft »backup«, ob das Bit gesetzt ist, und sichert die Datei nur dann. Genauso funktioniert das mit Dateien, die nach einem bestimmten Datum oder nach einer bestimmten Uhrzeit geändert oder erzeugt worden sind. Diese beiden Angaben »/d <Tag> . <Monat> . <Jahr>« und »/t <Stunde> . <Minute> . <Sekunde>« sind sogar auf eigenartige Weise miteinander verquickt. Benutzen Sie die Zeit ohne die Datums-Option, sichert »backup« nur Dateien, die zum angegebenen Zeitpunkt erzeugt oder modifiziert worden sind. Mit der Option »/t« beziehen Sie aber alle die Dateien in die Sicherung ein, die nach dem vereinbarten Termin entstanden sind, oder eben solche, die danach geändert wurden. Schließlich gibt »/a« an, daß Dateien, die sich auf den Sicherungsdisketten befinden, nicht überschrieben werden dürfen.

Um die Daten zurückzuholen, wird der Befehl »restore« verwendet, der ebenfalls Bestandteil von MS-DOS ist:

```
restore a:*.*/s
```

Die Parameter dieses Kommandos ähneln weitgehend denen des Backup-Befehls. Nur können Sie hier mit »/b <Datum>« oder »/a <Datum>« einzig Dateien zurückholen, die aus Tagen vor oder nach dem angegebenen Datum stammen. Genauso funktioniert das mit »/l <Zeit>« und »/e <

Zeit>«. Die Option »/p« bewirkt eine Rückfrage, ob Dateien, die sich auf der Platte nach dem letzten Backup geändert haben, zurückzuschreiben sind. Spezifizieren Sie diesen Schalter nicht, verlieren Sie eventuell aktuelle Daten, da sie mit alten Daten vom Backup überschrieben werden! Geben Sie den Parameter »/n« an, überträgt »restore« nur Dateien, die auf dem Ziellaufwerk nicht (mehr) existieren.

Nun geht es zum Aufbau der Festplatte, und Sie erfahren, wie der Festplatten-Controller den verfügbaren Magnetspeicherplatz aufteilt (Bild 3).

Das Laufwerk einer 40 MByte großen Festplatte besteht aus drei Magnetplatten, deren einzelne Plattenseiten von je einem Magnetkopf gelesen und beschrieben werden. Insgesamt macht das sechs Schreib-/Leseköpfe. Eine einzelne Platte ist in mehrere konzentrische Kreislinien, die sogenannten »Spuren« (englisch: »Tracks«), unterteilt. Ein handelsüblicher Controller bringt 819 Spuren auf einer Seite der 40-MByte-Festplatte unter. Die jeweils untereinanderliegenden Spuren auf allen Plattenseiten heißen »Zylinder«. Die Gesamtheit aller Spuren präsentiert sich somit als ineinandergeschachtelte, geometrische Zylinder (im Bild 1 sind die Spuren 0 bis 5 zusammen ein Zylinder). Die Festplatte ist also in 819 Zylinder unterteilt, die je sechs Spuren »hoch« sind. Eine Spur besteht üblicherweise aus 17 Sektoren, von denen jeder einzelne 512 Byte aufnimmt.

Nun geht es in die Details der Festplattenarchitektur

Der Controller speichert zwischen den Spuren eigene Kontrolldaten, mit deren Hilfe er die augenblickliche Position der Magnetköpfe bestimmen kann. Bei diversen Controllern liegen diese Daten auch auf einer Plattenunterseite. Dabei speichern diese Platten ihre Daten dann nur noch mit den restlichen Köpfen. Die Anzahl der Sektoren ist auf allen Spuren gleich, also auch auf denen, die der Plattenmitte am nächsten liegen und somit rein physikalisch weniger Platz bieten. An diesen Stellen wird die Spur also noch dichter beschrieben, was sich nachhaltig und unvorteilhaft auf die Datensicherheit auswirken kann.

Um die Anzahl der Köpfe zu bestimmen, könnten Sie natürlich – in einem Anflug von Zerstörungswut – Ihre Festplatte aufschrauben und nachsehen. Dieses Verfahren emp-

fiehl sich aber nicht, weil dadurch Staub ins Innere der Festplatte gelangt, der dem fein abgestimmten System von Platte und Schreib-/Lesekopf schnell den Garaus macht. Die Zahl der Zylinder und Sektoren ist auf diese Weise ohnehin nicht zu ermitteln.

Interessant ist die Verteilung der Sektoren. Die absoluten Sektoren 1 bis 17 befinden sich auf Zylinder 0, Kopf 0. Die nächsten 17, also die mit den Nummern 18 bis 34, liegen einen Kopf darunter, und so weiter. Der Grund für dieses System liegt darin, daß der Zeitraum, in dem der Schreib-/Lesekopf zu dem zu bearbeitenden Sektor gelenkt wird, einen Großteil der Zugriffszeit ausmacht.

Alle sechs Köpfe sind an einem Hebel befestigt und bewegen sich so immer synchron zur gleichen Spur der jeweiligen Plattenseite. Ist eine Spur gelesen, braucht der Controller nur auf den darunterliegenden Kopf umzuschalten, wo er sofort den nächsten zu lesenden Sektor findet.

Damit MS-DOS nicht jeden Sektor einzeln adressieren muß, faßt es mehrere Sektoren zu einem »Cluster« zusammen. Das Betriebssystem spricht immer nur einzelne Cluster an.

Ist auch nur ein Byte eines solchen Clusters beschrieben, gilt er als »belegt« – das wären in diesem Fall 2047 vergeudete Bytes. Verringert man die Sektorgröße, wird Speicherplatz gespart, sofern ein Großteil der auf der Platte abgelegten Dateien die dann vorherrschende Cluster-Größe nicht überschreitet. Kleinere Sektoren führen jedoch zu einer größeren zu verwaltenden Zahl von Clustern. Weil MS-DOS die Cluster-Belegungstabelle (»FAT«) im Arbeitsspeicher behält, bedeutet das dann wiederum einen höheren Bedarf an Arbeitsspeicher und Rechenzeit.

Logisch aufeinanderfolgende Sektoren sind nicht immer auch physikalisch benachbart. Die Reihenfolge der Sektoren einer Spur entspricht bei vielen Platten also nicht einer zusammenhängenden Folge der Sektoren 1 bis 17. Dies rührt daher, daß die Elektronik die Daten eines Sektors erst »verdauen« muß, bevor der Controller den nächsten Sektor in Lesezugriff nehmen kann. Mittlerweile hat die Platte sich aber längst weitergedreht, so daß sich der Schreib-/Lesekopf eventuell schon über dem nächsten oder gar übernächsten Sektor befindet. Physikalisch gesehen werden deshalb die Plattensektoren entlang einer Spur unzusammenhängend nummeriert, damit der Kopf nicht eine ganze Plattenumdrehung abwarten muß, bis er den gewünschten Sektor erreicht.

Aus diesem Grund wird ein »Versatz« der physikalischen Sektoren (auch »Überlappung« oder »Interleave« genannt) eingeführt. Logische Sektornummern sind diejenigen, die aufeinanderfolgend einem geordneten Dateizusammenhang angehören und die der Festplatten-Controller auch in dieser Folge liest. Sektornummern, die sowohl örtlich und logisch direkt nebeneinander liegen, ergeben einen Interleave-Faktor von 1.

Warum macht der Interleave-Faktor Ihre Festplatte langsamer?

Zum besseren Verständnis stellen Sie sich eine Torte, bestehend aus 17 Stücken vor, die von 1 bis 17 nummeriert sind. Dies ergäbe einen Interleave-Faktor von 1. Ein Interleave-Faktor von 3 ergibt folgende physikalische Reihenfolge der Sektoren auf einer Spur:

1-7-13-2-8-14-3-9-15
4-10-16-5-11-17-6-12

Bei der Dreierüberlappung werden folglich erst zwei physikalisch folgende Sektoren übersprungen, um an den nächsten logischen Sektor zu gelangen (Bild 4).

Soll also eine Datei, die fünf Sektoren groß ist, gelesen werden, muß der Lesekopf 13 Sektoren »überfliegen«, um auf alle benötigten Sektoren zugreifen zu können. Oder anders ausgedrückt: Um alle Sektoren einer Spur logisch aufsteigend zu lesen, muß sich die Festplatte in diesem

Beispiel dreimal drehen. Es ist leicht zu erkennen, daß bei steigendem Interleave-Faktor die Zugriffszeit zunimmt. Je langsamer der Computer ist, und je schneller die Platte, um so größer muß der Interleave-Faktor sein.

Der Festplattenhersteller legt in der Regel den Interleave-Faktor mit einem gewissen »Sicherheitszuschlag« fest. Typische Werte des Interleave-Faktors für PC-Laufwerke liegen bei 5, AT-Laufwerke weisen zumeist den Wert 3 auf. So gut wie alle AT-Platten lassen sich jedoch auch auf einen Interleave-Faktor von 2 formatieren. Es gibt allerdings auch schon sehr schnelle Controller, die den Traumwert von 1 erreichen.

Eine Änderung des Interleave-Faktors durch den Anwender ohne geeignete Software ist problematisch. Zwar läßt sich ein neuer Faktor durch erneute Low-Level-Formatierung realisieren, aber erst ein späterer Probeauf bringt bei nicht optimal abgestimmtem Faktor die Probleme an den Tag: Während bei einem zu großen Interleave der Schreib-/Lesekopf stets auf den noch entfernten Sektor warten muß, hat ein zu kleiner Interleave-Faktor zur Folge, daß nach jedem Lesen eines Sektors mehr als eine volle Plattenumdrehung vollzogen wird, ehe der nächste logische Sektor seine Daten abgeben kann. Sicherlich gibt es zum Thema Festplatte noch viel, viel mehr zu sagen, doch einen ersten Eindruck, wie Ihre Festplatte arbeitet und wie Sie mit ihr arbeiten, konnten Sie sicherlich gewinnen.

(sw)

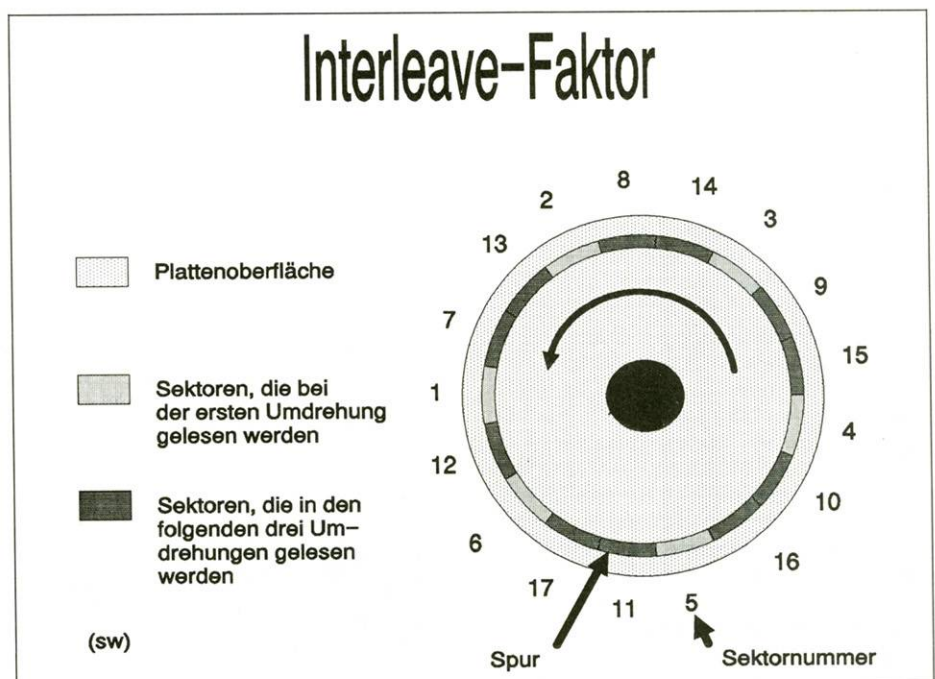


Bild 4. So sind die Sektoren auf einer Platte mit Interleave-Faktor 3 verteilt

Wie sag' ich's meinem Rechner?

Aufbau und Funktion einer PC-Tastatur

Der für Einsteiger komplizierteste Teil eines PC ist mit Sicherheit seine Tastatur. Ist es schon schwierig genug, einen Befehl fehlerfrei Buchstabe für Buchstabe einzutippen, so müssen auch noch die Bedeutungen der vielen Spezialtasten, wie [Escape], [Shift], [Enter] und [Backspace] erlernt werden. Doch dieser Grundlagenbeitrag ist auch für fortgeschrittene Anwender interessant, die zum Beispiel schon immer den Unterschied zwischen [Caps Lock] und [Shift Lock] wissen wollten.

Wie funktioniert eigentlich eine Tastatur? Sicher, Sie drücken eine Taste und sehen einen Buchstaben auf dem Bildschirm. Aber was passiert eigentlich wirklich?

In jeder PC-Tastatur sitzt ein Tastatur-Controller (ab der AT-Version sogar ein Prozessor), der mit nichts anderem beschäftigt ist, als ständig die Tastatur zu überwachen. Sobald Sie eine Taste drücken, sendet er einen bestimmten Zahlenwert über das Kabel an den Rechner. Dieser holt sich den Wert an der Tastaturschnittstelle, sieht nach, welchem Zeichen der Wert entspricht, und stellt es auf dem Bildschirm dar.

Trotz eigenem Controller ist eine PC-Tastatur ein ziemlich »dummes« Gerät

Für die Tastatur selbst hat es also keinerlei Bedeutung, in welcher Reihenfolge bestimmte Tasten gedrückt werden. Sie registriert lediglich die Tastendrücke und sendet sie einzeln an den Computer. Daß man mit bestimmten Tastenkombinationen (Befehlen) den PC veranlassen kann, etwas mehr oder weniger Sinnvolles zu tun, davon »weiß« die Tastatur nichts. Die Tastatur weiß nicht einmal, welchen Buchstaben Sie gerade gedrückt haben, sie weiß nur, daß Sie zum Beispiel die fünfte Taste von links in der dritten Spalte von oben gedrückt haben. Und diese Positionsangabe teilt sie dann dem PC mit. Der Vorteil davon ist, daß der Computer jeder beliebigen Taste jedes beliebige Zeichen zuordnen kann. Von dieser Möglichkeit wird beispielsweise Gebrauch gemacht, um länderspezifische Zeichen möglichst einfach in

einem Betriebssystem zu berücksichtigen.

In Deutschland werden zwei Gruppen von Tastaturen verkauft: Zum einen die »klassische PC-Tastatur« (Bild 1) und zum anderen die »AT-Tastatur«, die offiziell als »MF2-Tastatur« (Bild 2) bezeichnet wird. Die Sonder- und Funktionstasten finden sich bei beiden Typen an unterschiedlichen Positionen wieder. Außerdem ist es bei der Verwendung von MF2-Tastaturen üblich, auch die Sondertasten mit deutschen Namen zu beschriften; bei den PC-Tastaturen findet man häufig eine englische Beschriftung (siehe auch Textkasten). Der wahrscheinlich wichtigste Teil der Tastatur ist der große alphanumerische Bereich. Dieser heißt so, weil in ihm sowohl die Zeichen des Alphabets als auch die Nummern zur Zahlendarstellung (die oberste Reihe) zu finden sind. Daraus hat sich der Begriff »alphanumerisch« entwickelt. Jedem Schreibmaschinenprofi wird sofort auffallen, daß die alphanumerischen Tasten und ihre Verteilung auf der Tastatur im wesentlichen der Anordnung auf einer normalen Schreibmaschine entsprechen. Innerhalb des alphanumerischen Blocks finden sich aber auch einige Tasten, die auf einer Schreibmaschine normalerweise fehlen.

Wahrscheinlich werden Sie eine deutsche Tastatur verwenden. Der Unterschied zu einer amerikanischen

Tastatur liegt darin, daß die Buchstaben »z« und »y« vertauscht sind. Deshalb bezeichnet man die deutsche Tastatur oft auch als QWERTZ-Tastatur und die amerikanische als QWERTY-Tastatur (ausgehend von der Reihenfolge in der ersten Buchstabenzeile). Bei der amerikanischen Tastatur sind außerdem die deutschen Umlaute natürlich nicht berücksichtigt.

Die Sondertasten sind über die gesamte Tastatur verteilt. Sie dienen je nach Programm dazu, besondere Funktionen abzurufen. Leider ist die Verwendung der Sondertasten nicht einheitlich geregelt. So kann es unter Umständen möglich sein, daß das eine oder andere Anwendungsprogramm bestimmte Sondertasten anders interpretiert. Außerdem sind die Tasten bei unterschiedlichen Tastaturen häufig verschieden benannt, und Sie werden englische und deutsche Bezeichnungen vorfinden. Tabelle 1 zeigt die gebräuchlichsten Funktionen der Sondertasten.

Ohne festgelegte Bedeutung, aber dennoch sehr wichtig zum Bedienen eines PC: die Funktionstasten

Rechts vom alphanumerischen Block findet sich bei fast allen PCs der sogenannte »Ziffernblock«. Betrachtet man ihn als geschlossene Einheit, so erinnert er an die Tasten einer Rechenmaschine. Tatsächlich wurde der Ziffernblock auch deshalb in die Tastatur eingebaut, um das schnelle und flüssige Eingeben von großen Zahlenmengen zu vereinfachen. Dazu sind die Zifferntasten und die Tasten für Addition, Subtraktion (und manchmal auch Multiplikation und Division) auf einem eigenen Tastenblock zusätzlich zu ihrer »normalen« Position im alphanumerischen Bereich nochmals berücksichtigt. Die Bedeutung des Ziffernblocks wird über die Taste [Num Lock] geregelt. Ist [Num Lock] aktiv, was meist durch eine Leuchtdiode erkennbar ist, so werden beim Betätigen des Ziffernblocks die Zahlen benutzt. Andernfalls haben die Tasten des Ziffernblocks die auf den Tasten unten stehenden Bedeutungen (die [5] wird

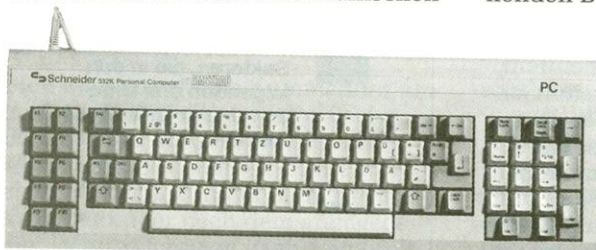


Bild 1. Die »klassische PC-Tastatur« wird heute nur noch mit wenigen PCs verkauft

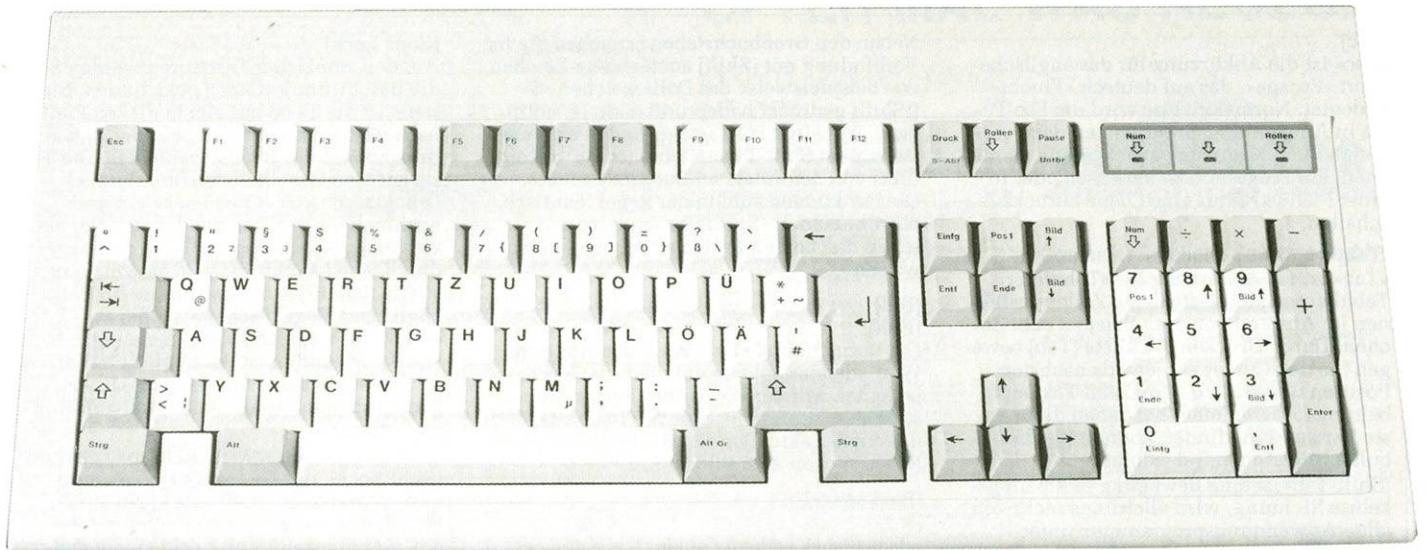


Bild 2. Eine MF2- oder auch AT-Tastatur stellt den aktuellen Stand der Technik dar

nicht verwendet). Tabelle 2 faßt die Bedeutung dieser Tasten zusammen. Die Funktionstasten (zehn auf einer PC- und zwölf auf einer MF2-Tastatur) sind mit »F1« bis »F10« (beziehungsweise »F12«) beschriftet. Sie dienen – wie der Name schon sagt – dazu, bestimmte Funktionen eines Programms abzurufen. Sie sind frei definierbar, das heißt, es gibt keine festen Regeln, wozu sie verwendet werden sollen oder können. Meist dienen die Funktionstasten dazu, für jedes Programm spezifische

Arbeitsschritte abzurufen. In dem einen Textprogramm speichern Sie zum Beispiel mit [F6] einen Text, in einem anderen rufen Sie möglicherweise die Floskel »mit freundlichen Grüßen« ab. Nur für die Funktionstaste [F1] hat sich eine Quasi-Standardfunktion herausgebildet: Diese Taste wird meistens dazu benutzt, um die Hilfsfunktion aufzurufen.

Die Zahlenwerte, die von der Tastatur beim Tippen einer Taste an den PC gesendet werden, müssen einem bestimmten Standard folgen. Denn

wenn unterschiedliche Tastaturen beim Tippen des gleichen Zeichens einen anderen Wert liefern würden, könnte der PC das Zeichen nicht richtig interpretieren.

Daher hat ein amerikanisches Institut, das ANSI (»American National Standards Institute«, vergleichbar mit dem Amt, das in Deutschland die DIN-Normen festlegt), eine Norm vorgegeben, an die sich die Tastaturen halten müssen. Diese Norm zum Austausch von Zeichen und Informationen wurde »ASCII« (für »American Standard Code for Information Interchange«, also »Amerikanischer Standard-Code zum Informationsaustausch«) genannt. Der ASCII-Standard legt genau fest, welchem Zeichen welcher Code entspricht. Dadurch ist gewährleistet, daß alle Tastaturen beim Tippen des Buchstabens »A« auch immer den Code für das »A« (64) an den PC senden – unabhängig vom Hersteller der Tastatur.

Der ASCII-Standard umfaßt 128 Zeichen. Um mehr Flexibilität bei der Bildschirmgestaltung zu erlauben, hat IBM den ASCII-Zeichensatz bei PCs um zusätzliche 128 Zeichen erweitert. Die Erweiterung umfaßt neben vielen Grafiksymbolen auch die Umlaute der meisten Sprachen dieser Erde. Diesen Zeichensatz bezeichnet man seitdem als »IBM-Zeichensatz«. In Bild 3 finden Sie diesen Zeichensatz mit den jeweiligen Codes. Auf einer Tastatur können Sie jedes Zeichen dieses Zeichensatzes erzeugen. Nun werden Sie sich möglicherweise fragen, wie Sie denn 256 Zeichen auf den Bildschirm bringen sollen, wenn viel weniger Tasten zur Verfügung stehen.

Die Zeichen und Funktionen, die der Anwender am häufigsten braucht, sind als eigene Tasten vorhanden. Die anderen Codes, zum Beispiel

Um für Sie als Leser eine Tastenbetätigung klar von sonstigen Eingaben zu trennen, hält sich die Redaktion an bestimmte »Schreibvereinbarungen«. Sie erkennen Tastendrucke an den eckigen Klammern »[« und »]«. Wenn in einer eckigen Klammer ein Bindestrich vorkommt, dann verbindet dieser Strich die Tasten, die gleichzeitig gedrückt werden müssen. Wenn Sie folgendes in der DOS International lesen, dann müssen Sie die aufgeführte Taste drücken.

- [Backspace] Die Taste mit dem dicken Pfeil nach links. Sie befindet sich in der rechten oberen Ecke des großen Tastenfeldes.
- [Ctrl] Je nach Tastaturtyp lautet die Aufschrift dieser Taste »Ctrl«, »Control« oder »Strg«.
- [Ctrl-X] Zuerst [Ctrl] drücken, festhalten, den Buchstaben [X] drücken und beide Tasten wieder loslassen.
- [Shift] Die Taste mit dem dicken Pfeil nach oben. Sie findet sich zweimal auf einer PC-Tastatur, und zwar in der zweiten Reihe von unten einmal ganz links und einmal ganz rechts.
- [Shift-Q] Zuerst [Shift] drücken, festhalten, den Buchstaben [Q] drücken und beide Tasten wieder loslassen.
- [Alt] Die Taste mit der Aufschrift »Alt«. Verwechseln Sie sie nicht mit der Taste [Alt Gr], diese hat eine andere Funktion.
- [Alt-M] Zuerst [Alt] drücken, festhalten, den Buchstaben [M] drücken und beide Tasten wieder loslassen.
- [Alt-1-2-3] Alt-Taste drücken und festhalten. Dann nacheinander die Ziffern [1], [2] und [3] des Ziffernblocks rechts auf der Tastatur betätigen.
- [Space] Die große Leertaste. »Space« bedeutet »Raum« oder »Leerraum«. Weil der Befehl »Bitte [] drücken« in Computerzeitschriften und Büchern ziemlich seltsam aussehen würde, einigte man sich auf die Bezeichnung [Space].
- [F3] Drücken Sie die Funktionstaste mit der Aufschrift »F3« (nicht [F] und [3]).

[Esc]

»Esc« ist die Abkürzung für das englische Wort »Escape«, das auf deutsch »Flucht« bedeutet. Normalerweise wird die Esc-Taste in Anwendungsprogrammen dazu verwendet, ein Kommando rückgängig zu machen, ein Programm zu verlassen oder in einer Menüstruktur eine Ebene zurückzuschalten.

[Tab]

»Tab« ist die Abkürzung für »Tabulator«. Tabulatoren sind festgelegte Zeilenpositionen im Abstand von zum Beispiel acht Zeichen. Durch Drücken der Taste [Tab] bewegen Sie den Cursor zur jeweils nächsten Position weiter. In den meisten Textverarbeitungsprogrammen, bei denen diese Taste Verwendung findet, können Sie die Tabulatorpositionen frei wählen.

[Shift-Tab] ist eine Bewegung in die umgekehrte Richtung, wird allerdings nicht von allen Anwendungsprogrammen unterstützt.

[Alt]

Diese Taste dient in erster Linie dazu, Zeichen des IBM-Zeichensatzes anzusprechen, die als eigene Tasten auf der Tastatur fehlen. Dazu halten Sie [Alt] gedrückt und tippen auf dem numerischen Ziffernblock den Code des Zeichens ein, das Sie erzeugen möchten. Danach lassen Sie die Taste wieder los. Sie können damit aber nicht nur Sonderzeichen, sondern prinzipiell jede Taste simulieren. [Alt-1-3] entspricht zum Beispiel der Enter-Taste.

[Alt Gr]

»Alt Gr« steht für »alternierende Grafikzeichen« und stellt eine deutsche Besonderheit dar. Mit einem geeigneten Tastaturreiber dient diese Taste dazu, die Sonderzeichen

@ { [] } ~ | \

zu erreichen. Bei deutschen Tastaturen wurden diese Zeichen ausgelagert, um den Symbolen

ä å ö ü ß

Platz zu machen. Den Klammeraffen »@« erreichen Sie bei vielen AT-Tastaturen zum Beispiel mit [Alt Gr-Q] und den Backslash »\« mittels [Alt Gr-ß]. Wenn Sie einen PC besitzen, sind diese Zeichen meist über eine Alt-Ctrl-Kombination verfügbar, zum Beispiel [Alt-Ctrl-ß].

[Ctrl]**[Strg]**

Mit Kombinationen in Verbindung mit [Ctrl] ([Strg] auf deutschen Tastaturen) erzeugen Sie besondere Steuerzeichen. So gibt es Programme (zum Beispiel das Textverarbeitungsprogramm »Wordstar«), die vom Benutzer Ctrl-Tastenkombinationen zur Steuerung erwarten. Dazu halten Sie die Ctrl-Taste gedrückt, tippen dann eine andere Taste und lassen beide Tasten wieder los. Ein Beispiel stellt die Tastenkombination [Ctrl-L] dar, durch die Sie bei »Wordstar« eine Suchfunktion auslösen.

[Shift]

Die Shift-Tasten dienen zur Darstellung von Großbuchstaben und ASCII-Zeichen, die in der oberen Hälfte einer Tastenbeschriftung zu finden sind. Auf einer mechanischen Schreibmaschine bewegen diese Tasten die Transportwalze so nach oben, daß beim nächsten Tastenanschlag der Groß- und nicht der Kleinbuchstabe gedruckt wird.

Neben den Großbuchstaben erreichen Sie in Verbindung mit [Shift] auch andere Zeichen, wie beispielsweise das Dollarzeichen »\$« ([Shift] gedrückt halten und dann [4] antippen). Auf einer PC-Tastatur finden sich meistens zwei Shift-Tasten, eine rechts und eine links von der untersten Buchstabenreihe. Die Funktionen sind in der Regel identisch, obwohl manche Spezialprogramme zwischen den beiden Tasten zu unterscheiden wissen.

[Del]**[Entf]**

Die Bezeichnung »Del« kürzt das englische Wort »delete« ab, was auf deutsch »löschen« bedeutet. Mit der Del-Taste ([Entf] auf deutschen Tastaturen) löschen Sie in der Regel das auf der aktuellen Schreibposition befindliche Zeichen vom Bildschirm.

[Backspace]**[< -]**

»Backspace« bedeutet soviel wie »ein Leerzeichen zurück«. Bei PC-Tastaturen ist diese Taste mit einem dicken Pfeil nach links gekennzeichnet und befindet sich rechts oben im Bereich der Buchstabentasten.

Die Schreibmarkierung wird eine Position nach links versetzt und das dort befindliche Zeichen gelöscht.

[Ins]**[Einf]**

»Ins« ist die Abkürzung für das englische Wort »Insert«, das auf deutsch »Einfügen« bedeutet. Bei den meisten Programmen und auch in der MS-DOS-Eingabeebene steuern Sie mit dieser Taste den »Einfügensmodus«. Ist er aktiv, dann werden neu eingegebene Zeichen automatisch eingefügt und alle in der Zeile folgenden Zeichen werden nach rechts verschoben, bei ausgeschaltetem Einfügensmodus überschreiben Sie den nachfolgenden Text.

In anderen Programmen wiederum (zum Beispiel Microsoft Word) fügen Sie mit [Ins] einen bestimmten Text an der aktuellen Schreibposition ein.

[PrtSc]**[Druck]**

»PrtSc« ist eine Abkürzung für den englischen Begriff »print screen«, was zu deutsch »drucke Bildschirm« bedeutet. Diese Taste ([Druck] auf deutschen Tastaturen) veranlaßt den PC, den aktuellen Bildschirminhalt auf einem Drucker auszugeben.

Bei PCs wird diese Funktion durch die Tastenkombination [Shift][*] aktiviert.

[Return]**[Enter]****[Eingabe]**

Diese Taste ist für die Benutzung eines PC von zentraler Bedeutung. Mit [Enter] teilen Sie dem PC mit, daß eine Eingabe nun beendet ist und der Computer diese Eingabe bearbeiten soll.

Eine PC-Tastatur kennt zwei Enter-Tasten, die jedoch die gleiche Bedeutung haben. Die eine mit der Aufschrift [Enter] befindet sich rechts auf dem Ziffernblock und die andere am rechten Rand des Buchstabenfeldes. Sie ist dort mit einem abwärts gerichteten Pfeil, der nach links wegnickt, markiert. Das bedeutet, daß der Cursor zuerst eine Zeile nach unten (»line feed«, »Zeilenvorschub«) und dann ganz nach links gesetzt wird (»carriage return«, »Wagenrücklauf« – eine Bezeichnung, die noch aus den Tagen der mechanischen Schreibmaschinen stammt).

[Caps Lock]

Auf den wenigsten Tastaturen werden Sie die Beschriftung »Caps Lock« finden. Meistens ist die Taste mit einem dicken Pfeil nach unten markiert und befindet sich am linken Rand des Buchstabenfeldes. Die Taste hat eine Umschaltfunktion: Einmal Drücken schaltet »Caps Lock« ein und nochmaliges Drücken wieder aus. Der aktivierte Caps-Lock-Zustand wird meistens durch eine Leuchtdiode gekennzeichnet. »Caps« ist die Kurzbezeichnung für den englischen Begriff »capitals«, der zu deutsch »Großbuchstaben« bedeutet. »Lock« wiederum ist das englische Wort für »verschließen«. Mit der Taste [Caps Lock] schalten Sie also dauerhaft auf Großbuchstaben.

Verwechseln Sie aber nicht [Caps Lock] mit [Shift Lock], das bei vielen Heimcomputern zu finden ist. [Caps Lock] bei einem PC schaltet lediglich die Großbuchstaben ein, wohingegen [Shift Lock] einem dauernden Drücken der Shift-Taste gleichkommt. Trotz eingeschaltetem [Caps Lock] müssen Sie zum Beispiel für das Dollarzeichen weiterhin die [4] zusammen mit [Shift] drücken. Unglücklicherweise haben manche MS-DOS-Versionen diese Definition nicht berücksichtigt und [Caps Lock] wirkt sich wie ein [Shift Lock] aus.

Bei den Original-Tastaturreibern von MS-DOS 3.3 und 4.01 wird die Caps-Lock-Funktion nicht durch erneutes Drücken von [Caps Lock] deaktiviert, sondern durch [Shift]. Der Schreibmaschinenhersteller IBM erinnerte sich an die eigentliche Bedeutung dieser Taste bei einer mechanischen Schreibmaschine. Denn dort wird eine eingerastete Caps-Lock-Taste ebenfalls durch Drücken von [Shift] wieder gelöst.

[Num Lock]

Diese Taste ist genau wie [Caps Lock] eine Umschalttaste. Mit [Num Lock] bestimmen Sie, ob der Ziffernblock für die Eingabe von Zahlen (bei eingeschaltetem »Num Lock«) oder für die Cursor-Steuerung benutzt wird (bei ausgeschaltetem »Num Lock«).

Bei vielen Tastaturen wird eine aktivierte Num-Lock-Funktion durch eine Leuchtdiode angezeigt. Häufig finden Sie einen dicken Pfeil nach unten mit dem Wort »Num« darüber als Beschriftung dieser Taste. Sie befindet sich jedoch immer in der oberen Reihe des Ziffernblocks (von den gedrängten Tastaturen eines tragbaren PC einmal abgesehen).

[Scroll Lock]

»Scroll« läßt sich am einfachsten mit »rollen« übersetzen. Ist die Scroll-Lock-Funktion aktiv, so bewirken Sie damit in den meisten Programmen, die diese Taste abfragen, das »Festhalten« eines bestimmten Objektes an einer Bildschirmposition.

[Break]**[Untrbr]**

»Break« bedeutet »Abbruch« oder auch »Unterbrechung«. Sie bewirken damit einen sofortigen Programmabbruch, sofern der Programmierer dies zugelassen hat. Die Break-Taste müssen Sie zusammen mit [Ctrl] drücken, um die gewünschte Funktion auszulösen. Der MS-DOS-Befehl »break on« bewirkt, daß der Rechner die Break-Taste auch dann abfragt, wenn gerade keine Bildschirmausgaben oder Festplattenzugriffe ausgeführt werden.

Tabelle 1. Die Sondertasten auf der Tastatur

1	2	3	4	5	6	7	8	9	10	11
12	13	14	15	16	17	18	19	20	21	22
23	24	25	26	27	28	29	30	31	32	33
34	35	36	37	38	39	40	41	42	43	44
45	46	47	48	49	50	51	52	53	54	55
56	57	58	59	60	61	62	63	64	65	66
67	68	69	70	71	72	73	74	75	76	77
78	79	80	81	82	83	84	85	86	87	88
89	90	91	92	93	94	95	96	97	98	99
100	101	102	103	104	105	106	107	108	109	110
111	112	113	114	115	116	117	118	119	120	121
122	123	124	125	126	127	128	129	130	131	132
133	134	135	136	137	138	139	140	141	142	143
144	145	146	147	148	149	150	151	152	153	154
155	156	157	158	159	160	161	162	163	164	165
166	167	168	169	170	171	172	173	174	175	176
177	178	179	180	181	182	183	184	185	186	187
188	189	190	191	192	193	194	195	196	197	198
199	200	201	202	203	204	205	206	207	208	209
210	211	212	213	214	215	216	217	218	219	220
221	222	223	224	225	226	227	228	229	230	231
232	233	234	235	236	237	238	239	240	241	242
243	244	245	246	247	248	249	250	251	252	253
254	255									

IBM-Zeichensatz

(DOS International)

Bild 3. Der komplette IBM-Zeichensatz mit den entsprechenden Zahlencodes

Grafikzeichen und Umlaute, werden über die Alt-Taste aktiviert. Um ein beliebiges Zeichen des Zeichensatzes zu erzeugen, halten Sie die Alt-Taste gedrückt und tippen dann nacheinander die einzelnen Ziffern des Tastencodes auf dem Ziffernblock ein.

In Bild 3 finden Sie die Zeichen des kompletten IBM-Zeichensatzes abgebildet samt den dazugehörigen Werten, die Sie in Kombination mit der Alt-Taste eingeben müssen. Der IBM-Zeichensatz enthält übrigens alle Zeichen mit den Nummerncodes zwischen 0 und 255.

Auf den Seiten 48 und 49 finden Sie in einer großen Übersicht zusätzlich dazu den erweiterten ASCII-Zeichensatz, der die Zeichen der Werte von 0 bis 128 sowie die internationalen Sonderzeichen und diverse Grafikzeichen enthält. Sie erzeugen also beispielsweise das Zeichen «1/4», das den Wert 172 besitzt, indem Sie die Tastenfolge [Alt-172] auf dem Ziffernblock Ihrer Tastatur eingeben. Ebenso verfahren Sie mit allen anderen abgebildeten Zeichen. Wenn Ihnen viel daran liegt, möglichst schnell und fehlerfrei zu tippen, dann sollten Sie einen Schreibmaschinenkurs in einer Volkshochschule belegen. Fast überall gibt es Spezialkurse für »Computer-Typisten«, in denen speziell auf die Besonderheiten einer PC-Tastatur eingegangen wird.

(Martin Althaus/tr)

[Home]

Als »Home-Position« bezeichnet man die linke obere Ecke im Bildschirm. Im PC-Bereich bewirken jedoch nur sehr wenige Programme ein Setzen des Cursors an diese Position. Meistens bewegen Sie die Schreibmarkierung an den Anfang der aktuellen Bildschirmzeile. Die eigentliche Home-Funktion ist dann über [Ctrl-Home] erreichbar.

In manchen Programmen schalten Sie mit [Home] an den Anfang einer gerade bearbeiteten Datei.

[End]**[Ende]**

Die »End-Position« ist normalerweise die erste Position in der letzten Bildschirmzeile. In einigen Programmen bewegen Sie den Cursor mit der End-Taste tatsächlich an diese Position des Bildschirms. Bei anderen Programmen hingegen wird [End] dazu verwendet, um an das Ende einer Eingabezeile oder eines Textes zu gelangen.

[< Pfeil nach oben >]**[< Pfeil nach unten >]****[< Pfeil nach links >]****[< Pfeil nach rechts >]**

Diese Tasten dienen dazu, den Cursor auf dem Bildschirm um ein Zeichen zu verschieben. In der MS-DOS-Kommandoebene haben allerdings nur [< Pfeil nach links >] und [< Pfeil nach rechts >] eine Bedeutung: Sie dienen zum Korrigieren einer Eingabe. Die beiden anderen Tasten sind nur in bestimmten Programmen verfügbar.

[PgUp] / [Bild < Pfeil nach oben >]**[PgDn] / [Bild < Pfeil nach unten >]**

»PgUp« ist die Kurzform für den englischen Begriff »Page Up«, zu deutsch »Seite aufwärts«. In den meisten Programmen blättern Sie mit dieser Taste seitenweise (zum Beispiel in einem Text) nach oben. Mit [PgDn], also mit »Seite abwärts«, blättern Sie entsprechend seitenweise nach unten.

Tabelle 2. Die Bedeutung der Cursor-Tasten



Warum woanders mehr bezahlen?

- [066] **FLU-SHOT+** 1.2 Virenschutz-Programm
[070] **PC OUTLINE** 3.34 Textverarbeitung
[230] **Prof. Master Key 1.8** FP-Utilities mit gem. Benutzerobf.
[340] **Anti-Virus** Schutz vor C-Viren
[258] **Wampum** 2.0 ausgezeichnete Datenbank
[452] **RGB Techwriter** wissen.-technische Textverarbeit.
[334] **Lighting Press** Printshop-Clone
[335] **Printshop-Grafik** Zusatzdisk zu #334
[330] **Citydesk** einf. DTP-Programm
[288] **Image 3D 3.0** Erstellung von Drahtmodellen
[117] **HI-RES RAINBOW** variantenreiches Malprogramm
[217] **Imageprint 1.5** erhöht die Druckqualität
[309] **PackDisk 1.4** Festplattenoptimierer
[231] **Powermenu** Menüprogramm
[383] **ASK-IT** Text-Such-System 2 Disk
[371] **PC-Suchteam** Schreibtisch-Organisator
[051] **Expert System** Superprogramm im KI-Bereich
[119] **Graphtime** 3.0 Grafik-Programm 2 Disk
[339] **Flowdraw** 2 Disk Erstellung von Flußdiagrammen
[261] **Video-Monopoly** nur Farbe
[296] **ASDKeys** ITaste=Fuße Folge von Symbolen
[327] **UVESTOR** Überwachung aller Investitionen
[252] **C-Tutor** "C"-Kursus 2 Disk
[019] **VISIBLE-PASCAL** Pascal Compiler für Anfänger
[051] **Express-Calc** 2.1 Kalkulationsprogramm 2 Disk
[426] **RBBS 8.1** 4 Disk eine eigene Mailbox errichten
- [322] **DANCAD 3D** 2.3 Datenbanksystem 4 Disk
[027] **Pianoman** 3.5 Kompositionshilfe
[048] **CH-Writer** wissenschaftl. Textverarbeitung
[293] **CGA** CGA-Herkules-Simulation
[450] **PCC** Personal - C - Compiler
[257] **Shortcut** komfortable Benutzeroberfläche
[408] **File Commando** File Management Spitzell
[259] **Diskcommand** Norton Utility
[356] **LQ-Print 2.5** Druckbildverbesserer
[436] **PKZIP** PKUNZIP & PKSEFX. Daten Ent. und Komprimierung
[306] **Draftsman** Geschäftsgrafiken
[216] **Blackbeard** 7.37 sehr leistungsfähiger Editor
[302] **Fontastic** 2 Disk verschiedene Schriftarten
[021] **Kermit** 2.3 DFU-Programm 2 Disk
[312] **Family History S.** Stammbaum 2 Disk
[384] **Math Pak** 4 Disk
[068] **Spanish for Trav.** 8 Spanisch-Lektionen engl.-span.
[183] **Willy the Worm** Actionspiel
[297] **Lotto-Prophet**
[222] **Spacewar**
[050] **Express-Calc** 2.3 Lotus 1-2-3 Clone 2 Disk
[310] **Komprimierutilities** 3 Disk
[333] **Crossword** Erst.e.eig. Kreuzworträtsel
[401] **Mandelbrot & J.** 2 Disk Fractal-Grafiken in Farbe SUPER!
[394] **Mindreader** Textv. m. künstl. Intelligenz
[451] **FORTH** 2 Disk für den Forth-Programmierer

- [287] **AsEasyAs** 3.1 LOTUS 1-2-3 Clone
[076] **Modula-2 Tutor.** 2 Disk
[326] **Modula-2 Compiler** 2 Disk
[137] **File Express** 3.2 sehr gute Datenbank 2 Disk
[243] **Qubecalc** Tabellenkalkulation
[210] **PC Key Draw** 3.6 Malprogramm in Farbe 4 Disk
[076] **A86/D86 Assemb.** Assembler, Disassembler
[236] **Composer** Kompositionsprogramm
[291] **FingerPaint** 1.5 Super Malprogramm
[233] **Homebase** 2.2 wie Sidekick 3 Disk
[337] **Freec.&Copyprot.** 2 Disk
[162] **Touchtype** 3.2 Schreibmaschinenlehrgang
[359] **BOSS 1.4** 2 Disk C-Window-Manager
[116] **Make My Day** Terminkalender
[045] **DREAM** 2.0 Datenbank 3 Disk
[041] **Sidewriter** seitwärts drucken
[289] **IMP-Shell** sehr gutes Expertensystem
[159] **DOS-Utilities** Multi-Tasking, usw.
[373] **EGA-TREK** EGA-Version von Star-Trek
[319] **3D-Chess**
[191] **SLEUTH** Kriminalspiel
[035] **X-LISP 3.1** Sprache der künstl. Intelligenz
[510] **kl. Spielesammlung** 5 Disk
[391] **DOSMENU deutsch** Festplatte notwendig
[425] **Black Magic** 3 Disk Hypertext Word Processor
[444] **Alchemy Desktop** hilfreiche DTP - Utilities

Deutsche Programme:

- [320] **StockMaster 3** Börsenprogramm 2 Disk
[075] **RW-File 3.2** Datenbank
[065] **Stammbaum**
[350] **PC-Write** Textverarbeitung
[354] **RW-Buch 3.2** Buchhaltungsprogramm
[352] **CASCADE** CAD-Programm
[301] **ADVECO** Adressenverwaltung
[364] **EHO 3.6** 2 Disk Einzelhandelsorganisation
[201] **HeBase** Datenbank
[365] **Vokabeltrainer** Deutsch-Englisch
[360] **Faktura** hervorragendes Fakturierungsprogr.
[280] **Spuk. Goldgräberst.** Adventure
[358] **Pyramiden** Adventure
[422] **RW-Kasse** Kassenverwaltung
[197] **Spielebox** 10 Spitzenspiele
[353] **Königreich** Adventure
[272] **Vereinsverwaltung** mit Paßwort
[380] **Haushaltsbudget** Ein-/Ausg. für 12 Monate
[381] **EIBU** Einnahme-Überschubführung
[382] **PriBu** Privatbuchhaltung
[196] **Astrologie** Charakteranalyse, usw.
[385] **Kinderspiele** Rechen- und Schreibspiele
[392] **Balkengrafik** CGA, EGA, Hercules
[234] **Text + Bild** einfaches DTP-Programm
[455] **Fakturamat** Fakturierungsprogramm

Das PDS-Paket #1

Die besten PD-Programme (Lightning Press, Deskmate, Citydesk und FlowDraw) mit Handbuch in Deutsch! nur DM 39,==

Die kleine Spielesammlung

Die besten Spiele aus der großen Spielesammlung. Insgesamt 5 Disketten.

nur DM 49,==

Kommerzielle Programme:

Stammbaum-PC

Ahnenforschung an Ihrem PC. Erstellen Sie Ihren eigenen Stammbaum! Mit Statistik, Verwandtschaft, Listenausgabe und viel mehr.

nur DM 79,==

Lottomat-PC

Knacken Sie das Lottosystem! Mit allen Zahlen seit 1956. Statistik, Zahlenpaare.

nur DM 59,==

Die große Spielesammlung

Unser Superangebot: Über 250 Spiele auf 26 Disketten. Action-, Abenteuer-, Kinder-, Grusel-, Brett-, Schach-, Karten-, Strategie und Denkspiele. Das ideale Weihnachtsgeschenk!

nur DM 149,==



Mathemat

Soll ich mein neues Auto leasen oder per Bankkredit finanzieren? Was ist besser: lineare oder degressiv-geometrische Abschreibung? Welche Geldanlage bietet die beste Verzinsung? Wie hoch ist der effektive Jahreszins? Wann habe ich mein Darlehen zurückgezahlt? Mathemat ist ein interaktives Programm zur Lösung finanzmathematischer Probleme. Keine finanzmathematische Kenntnisse nötig!

nur DM 39,==

Der Vokabeltrainer-PC

Da macht das Lernen Spaß! Vokabeltrainer mit vielen Funktionen! Mit Statistik und Erfolgskontrolle. Mit englischem Grundwortschatz!

nur DM 39,==

Unsere Katalogdiskette mit vielen tollen Angeboten liegt bei jeder Sendung kostenlos bei!

Bestellung (Nummern ausreichend):

Absender:

Gewünschte Zahlungsweise:

- [] Rechnung (Zahlungsziel 10 Tage)
[] Scheck, bar, Briefm.

[] Nachnahme

Preise:

- DM 9,- pro Disk
ab 10 St DM 8,- pro Disk
ab 25 St DM 7,- pro Disk
ab 40 St DM 6,- pro Disk

[] Lieferung auf 5,25 Zoll

[] Lieferung auf 3,5 Zoll
Aufpreis DM 2.50 pro Disk.

Ab DM 150,- keine Versandkosten, ansonsten DM 4,- pro Lieferung.

PDS-Service, Helga Blankenstein, Haustätter Höhe 10, 8200 Rosenheim

☎ 08031/82488 (24h)

Punkt für Punkt zum Erfolg

So funktionieren Drucker

Drucker gehören zu den wichtigsten Peripheriegeräten eines PC. Denn mit dem Computer entworfene Tabellen, Briefe, Listen und Grafiken sind wenig wert, wenn Sie sie nur auf dem Bildschirm betrachten können. Doch wie bringen Drucker die übertragenen Zeichen in eine lesbare Form, und was ist der Unterschied zwischen Matrix-, Tintenstrahl-, Thermo- und Laserdruckern?

Seit der Erfindung des Buchdrucks mit beweglichen Lettern durch Johannes Gutenberg im Jahre 1455 hat sich die Drucktechnik durch die Jahrhunderte erheblich weiterentwickelt. Schreibmaschinen ermöglichten es später jedem, der Schreiben und Lesen beherrschte, seine Gedanken schnell in standardisierter Schrift zu Papier zu bringen. Die Einführung der Computer mit den zugehörigen Druckern führte das Ende der Vorherrschaft der mechanischen und elektrischen Schreibmaschinen herbei. Selbst die Darstellung von Grafiken ist für die Mehrzahl der modernen Drucker, die ihre Aufgabe in kaum vorstellbarer Geschwindigkeit erledigen, ein leichtes Unterfangen. Aufgrund der hohen Anzahl von unterschiedlichen Anwendungsgebieten und wegen des technischen Fortschritts haben sich allerdings viele verschiedene Druckverfahren behauptet, die um die Gunst der Computerbenutzer buhlen.

Drucker werden in zwei Kategorien eingeteilt: Impact- und Non-Impact-Drucker

Die sieben Druckprinzipien, die in den folgenden Erläuterungen vorgestellt werden, beherrschen fast den gesamten Markt für Computerdrucker. Auf spezielle Abwandlungen der jeweiligen Technik und einige »Exoten« gehen die Ausführungen nicht näher ein. Außerdem beschäftigen sie sich ausschließlich mit den Standard-Funktionen. Es gibt zwar immer wieder Ausnahmen, die Nachteile eines Druckverfahrens abschwächen oder sogar aufheben, aber diese Fälle sind bei weitem nicht die Regel. Für die verwendeten Verfahren gibt es bei fast allen Druckverfahren einen wichtigen Indikator, der die zugrundeliegende Technik repräsentiert: den Druckkopf. Der Druckkopf, der meist auf einen beweglichen Schlitten

montiert ist, hat die Aufgabe, die von einer Steuereinheit übermittelten Informationen in Farbe beziehungsweise Tinte umzusetzen und zu Papier zu bringen.

Grundsätzlich gibt es zwei Kategorien, in die Drucker eingeteilt sind: Impact- und Non-Impact-Geräte (»impact« ist ein englischer Ausdruck für »Anschlag«). Bei den Impact-Druckern, zu denen Kugelkopf-, Typenrad- und Matrixdrucker gehören, werden mechanische Elemente wie kompakte Buchstaben oder dünne Nadeln gegen einen Farbträger beziehungsweise ein Farbband geschlagen und hinterlassen auf diese Weise einen Abdruck auf dem dahinterliegenden Papier.

Non-Impact-Drucker erzeugen hingegen einen Auswurf, indem sie beispielsweise Tinte auf das Papier sprühen, die Konturen elektrischer Thermolemente wiedergeben oder eine lichtempfindliche Trommel mit Laser bestrahlen und die belichteten Stellen mit Farbpulver sichtbar machen. Auf diese Weise spart man bei der Planung und Konstruktion viele bewegliche und schwer zu verarbeitende Bauteile, die man beim mechanischen Anschlag in die Herstellung miteinbeziehen muß.

Die Vorteile der moderneren Non-Impact-Drucker liegen auf der Hand: Erstens ist die Geräuschkentwicklung,

die vor allem am Arbeitsplatz eine große Rolle spielt, gering. Des weiteren liegt der Geräte- beziehungsweise Bauteileverschleiß der oft in Anspruch genommenen Druckkopfkomponenten wesentlich, wodurch eine längere Lebensdauer garantiert wird. Nicht zuletzt wird die Druckgeschwindigkeit wesentlich erhöht, da der Weg der mechanischen Anschlagselemente zum Papier wegfällt.

Der Hauptnachteil der Non-Impact-Drucker liegt darin, daß sie nur ein Blatt auf einmal bearbeiten können – Durchschläge fallen also weg. Darüber hinaus liegt das Preisniveau der Non-Impact-Drucker generell über dem der Impact-Drucker. Zudem erreichen Sie mehrfarbige Ausdrücke fast ausschließlich nur über komplizierte Umwege.

Zu den Impact-Druckern gehören die Drucker mit dem schon von der Schreibmaschine her bekannten Kugelkopf oder Typenrad sowie die Matrix- beziehungsweise Nadel-drucker.

Kugelkopfdruker und Typenraddruker gehören zur Klasse der »Impact-Drucker«

In einem Kugelkopfdruker befindet sich ein halbkugelförmiger Metallblock, aus dem die druckbaren Standard- und Sonderzeichen hervortreten. Soll ein Drucker beispielsweise den Buchstaben »A« ausgeben, erhält er vom Computer den ASCII-Code des auszugebenden Zeichens. Die Steuereinheit des Druckers dreht den Kugelkopf, bis sich der gewünschte Kugelkopfbereich vor dem Blatt befindet und drückt dann dieses Zeichen gegen das Farbband und somit auf das eingespannte Blatt Papier. Die auf dem Farbband befindliche Farbe löst sich und bildet das

Vor- und Nachteile der individuellen Drucktechniken

Druckprinzip	Geschwindigkeit	Druckqualität	Grafik	Durchschläge	Preis	Besonderheiten
Kugelkopf	langsam	sehr gut	nein	ja	niedrig	Schriftarten eingeschränkt
Typenrad	langsam	sehr gut	beschränkt	ja	mittel	Schriftarten eingeschränkt
Matrix	mittel	befriedigend bis gut	ja	ja	mittel	-
Tintenstrahl	mittel	gut bis sehr gut	ja	nein	hoch	-
Thermo	mittel	gut bis sehr gut	ja	nein	niedrig	Spezialpapier notwendig
Thermo-Transfer	mittel	gut bis sehr gut	ja	nein	hoch	-
Laser	schnell	sehr gut und besser	ja	nein	sehr hoch	Ausgabe nur seitenweise

Tabelle 1. Jedes Druckprinzip hat Leistungsmerkmale, die es von den anderen Druckarten abhebt

durch den Kugelkopf gebildete Zeichen ab.

Typenraddrucker basieren auf einem ähnlichen Prinzip. Der Unterschied liegt darin, daß hier die Steuereinheit nicht eine in der Horizontalen befindliche Kugel dreht, sondern eine vertikal angebrachte Kunststoffscheibe in die richtige Position bringt. Ist das gewünschte Zeichen an der richtigen Stelle, dann schlägt ein kleiner Hammer die Type gegen das Farbband. Kugelkopf- und Typenraddrucker, die auch heute ihren Reiz noch nicht verloren haben und für einige Anwendungen gut geeignet sind, gehören zu den ältesten Vertretern der Peripheriegeräte. Wenn Sie ausschließlich Briefe schreiben wollen, kommen Sie an dieser Technik nicht vorbei, denn die Schriftqualität ist hervorragend und auch von den weiter unten vorgestellten Laserdruckern nicht zu überbieten. Ihr größtes Manko ist allerdings die niedrige Geschwindigkeit: ein Typenraddrucker schreibt ungefähr fünfzehn bis dreißig Zeichen in der Sekunde. Außerdem ist die Flexibilität im Bereich der Schrifttypen eingeschränkt, weil Sie bei jedem Schriftartwechsel den Kugelkopf oder das Typenrad austauschen müssen.

Am weitesten verbreitet ist der Matrixdrucker

Der wohl bekannteste Vertreter der Impact-Kategorie ist der Matrix- beziehungsweise Nadeldrucker. Matrixdrucker sind die meistverkauften Drucker der Welt, da sie ein gutes Schriftbild mit einer akzeptablen Geschwindigkeit zu einem günstigen Preis bieten. Sie sind – mechanisch gesehen – mit Typenraddruckern ziemlich nah verwandt. Auch bei ihnen schlagen Hämmer gegen das Farbband, der Unterschied besteht in der wesentlich präziseren Steuerelektronik.

Die grundlegende Funktionsweise eines Matrixdruckers ist leicht zu erklären. Einen Buchstaben kann man in kleine Elemente zerlegen, die die Gestalt des Zeichens ausmachen. Diese lassen sich später wieder zu einem Buchstaben zusammenfügen. Die minimale Anzahl von übereinanderliegenden Punkten, um alle Buchstaben des lateinischen Alphabets darzustellen, beträgt sieben. Die Steuerelektronik eines Matrixdruckers zerlegt das zu druckende Zeichen mittels eines Zeichengenerators in seine Bestandteile, das heißt in eine Matrix aus Zeilen und Spalten aller möglichen Bildpunkte. Danach leitet sie die Information jeder Spalte an den aus

mindestens sieben dünnen und stumpfen Nadeln bestehenden Druckkopf. Diese drücken von hinten gegen das Farbband und erzeugen damit einen Abdruck auf dem Papier. Der Druckkopf besteht aus einem Gehäuse mit einzelnen aus einer Metalllegierung bestehenden Stützebenen, die die dünnen Nadeln führen, und einer Reihe von kleinen Elektromagneten, die die Nadeln im richtigen Moment aktivieren. Zu diesem Zweck fließt durch die Spule eines Elektromagneten ein schwacher Stromstoß, der ein am Rand des Druckkopfes befestigtes Metallplättchen anzieht. Das freie Ende des Metallplättchens schießt vor, hämmert die aus Wolframdraht bestehende Nadel gegen das Farbband und überträgt auf diese Weise einen Punkt auf das Papier.

Nach den ersten 7-Nadel-Druckern führte man schnell die 9-Nadel-Drucker ein, um auch Unterlängen in sinnvollen Proportionen ausgeben zu können.

In letzter Zeit wurde die Matrixtechnik weiter verbessert. So wurde beispielsweise die Zahl der eingebauten Nadeln erhöht. Das Problem liegt dabei allerdings darin, daß die Nadeln, wenn sie zu dünn sind, der Belastung nicht mehr standhalten und abbrechen oder durch das relativ grobe Farbband stechen und somit keinen Farbabdruck hinterlassen. In diesem Fall behilft man sich mit einem Trick: Die große Stunde der Steuerelektronik schlägt. Eine Möglichkeit besteht darin, den Druckkopf jede Zeile mit einem kleinen Versatz von ein paar Bruchteilen eines Millimeters wiederholen zu lassen und somit ein geschlossenes Druckbild zu erzeugen. Ein anderer Trick, der bei den heute weit verbreiteten 18- und 24-Nadel-Druckern Anwendung findet, führt dazu, daß die Nadeln nicht unmittelbar in einer Reihe übereinander angeordnet sind, sondern leicht verschoben in mehreren Reihen nebeneinander. Hat die erste Reihe der Nadeln Punkte auf das Papier geschlagen, bewegt sich der Druckkopf ein winziges Stück weiter, und die zweite Nadelreihe tritt in Aktion. So wird beispielsweise mit einem 48-Nadel-Drucker die Auflösung eines Laserdruckers erreicht. Das stellt dann aber natürlich höchste Anforderungen an Material und Farbband, und damit ist auch die Grenze der Belastbarkeit erreicht.

Die Ansteuerung des Druckkopfes eines Tintenstrahldruckers erfolgt auf ähnliche Weise wie die eines Matrixdruckers. Auch in diesem Fall zerlegt ein Zeichengenerator die zu druckenden Zeichen in ihre Bestandteile. Der gravierende Unterschied besteht darin, daß die Steuerelektronik keine

Toolboxen für Programmierer gesucht

Haben Sie eine Toolbox in C, Pascal, Basic, dBase, Clipper oder Assembler geschrieben? Beispielsweise schnelle, universelle Grafikroutinen, eine Textverarbeitungs- oder eine Datenbank-Toolbox? Dann sollten Sie sich schnellstmöglich mit uns in Verbindung setzen. Schicken Sie uns unverbindlich eine Diskette mit Ihrem Programm und einen Ausdruck der Dokumentation.

Falls Sie sich vorweg informieren möchten: Zwischen 15.00 und 17.00 Uhr stehen wir Ihnen an allen Werktagen unter der Rufnummer (089) 4 61 00 10 zur Verfügung.

Schicken Sie Ihre Unterlagen an folgende Adresse:

**Redaktionsbüro
Everts & Hagedorn
»DOS Software«
Hans-Stießberger-Str. 3
8013 Haar/München**

Sinnvolle Anwendungsgebiete

Druckprinzip	Farbdruck	Korrespondenz	Wissenschaft	Kommerz	Konstruktion	Spezialgebiet
Kugelpunkt	nein	sehr gut	nein	beschränkt	nein	Korrespondenz
Typenrad	nein	sehr gut	nein	beschränkt	nein	Korrespondenz
Matrix	ja	ziemlich gut	beschränkt	ja	bedingt	Einfache Grafik und Schrift
Tintenstrahl	bedingt	gut	ja	ja	bedingt	Einfache Grafik und Schrift
Thermo	ja	gut	beschränkt	ja	bedingt	Einfache Grafik und Schrift
Thermo-Transfer	nein	gut	ja	ja	bedingt	Einfache Grafik und Schrift
Laser	nein	sehr gut	ja	sehr gut	gut	Grafik und Schrift

Tabelle 2. Aufgrund der unterschiedlichen Anwendungsgebiete eignet sich nicht jeder Drucker für jeden Anwendungsbereich

Nadeln sondern Düsen ansteuert. Diese Düsen sprühen die aus einem Tank stammende Tinte auf das gespannte Papier. Bei diesem Druckverfahren haben sich zwei Techniken bewährt.

Tintenstrahl- und Thermo-Drucker gehören zur Gruppe der »Non-Impact-Drucker«

Bei der älteren Unterdrucktechnik herrscht im Druckkopf während des Druckvorgangs Unterdruck, so daß die Tinte nicht aus den Düsen austreten kann und die Düsen nicht verstopfen. Wenn ein Punkt ausgegeben werden soll, wird der Druck in der Düse kurz erhöht, und die Düse spritzt einen Tintentropfen auf das Papier. Die Tintenversorgung erfolgt aus einem leicht austauschbaren Behälter durch Kapillarwirkung in den feinen Zuleitungsröhrchen.

Die modernere Blasenstrahl- oder Dampfblasentechnik schleudert die Tinte über eine kleine Dampfblase heraus. Wenn eine Düse aktiviert werden soll, gibt die Steuerelektronik einen kleinen Stromstoß auf die zugeordnete Leiterbahn. Indem sich ein hinter der Düse befindlicher Widerstand schlagartig auf über 200 Grad Celsius erhitzt, verdampft die in der Zuleitung vor ihm befindliche Tinte. Dabei entsteht eine kleine Gasblase, die sich ausdehnt und dabei die in der Düsen Spitze enthaltene Tinte auf das Papier schleudert.

Thermodrucker, deren Hauptnachteil darin besteht, daß sie teures Spezialpapier brauchen, gehören zu den preiswertesten Druckertypen. Sie erzeugen einen Punkt auf dem Papier, indem Sie einzelne Partikel des

Druckkopfes erhitzen. Bei dem aus einer Farblage und einer darüber liegenden Wachsschicht aufgebauten Papier löst sich die Wachsschicht, und es erscheint ein schwarzer Punkt.

Bei Thermo-Transfer-Drucker erfolgt die Wärmeeinwirkung hingegen nicht auf das Papier, sondern auf das vorgelagerte Farbband. Die Elektroden des Druckkopfes erhitzen sich bei diesem Verfahren ebenfalls nicht. Stattdessen geben sie einen Stromimpuls, der in diesem Fall eine Widerstandsschicht auf dem Farbband aufheizt, damit die Wachsschicht abläßt und das Papier durch die dahinter befindliche Tinte schwärzt.

Laserdrucker – Die Könige unter den Druckern

Das modernste, aufwendigste und gleichzeitig auch beste Druckverfahren ist der Laserdruck. Das Herz eines Laserdruckers ist eine sich drehende lichtempfindliche Trommel. Nachdem das sogenannte Ladecorotron die Trommel statisch aufgeladen hat, zeichnet ein durch ein Ablenkssystem kontrollierter Laserstrahl über einen Polygonspiegel die Bildinformation auf die Trommel. Jede Stelle, auf die der Laserstrahl auftrifft, wird entladen. An diesen Bereich lagert sich das positiv geladene Farbpulver, der Toner, ab und überträgt sich auf das gegen die Trommel gepreßte Papier. Der Toner wird kurzzeitig erhitzt, schmilzt und haftet dann auf dem Papier. Währenddessen säubert eine Reinigungsklinge die Trommel, und der Druckvorgang wiederholt sich. Bei diesem Verfahren erreicht man die höchste Auflösung der vorgestellten Druckprinzipien.

Natürlich hat dieses aufwendige Druckverfahren seinen Preis – und die Laserdrucker kosten denn auch meist erheblich mehr als beispielsweise Matrixdrucker.

Wie Sie sehen, hat jedes Druckverfahren seine Vor- und Nachteile. Um die Unterschiede zwischen den einzelnen Druckprinzipien zu verdeutlichen, faßt Tabelle 1 die wichtigsten Vergleichskriterien in einer Aufstellung zusammen. Aus den Leistungsmerkmalen resultiert natürlich auch ein breit gefächertes Anwendungsspektrum, das den Druckern Einsatzbereiche zuweist, für die sie am besten geeignet sind. Um Anhaltspunkte zu geben, die die Auswahl erleichtern, stellt Tabelle 2 die Informationen übersichtlich dar.

Eine qualitativ hochwertige Druckausgabe hat natürlich ihren Preis

In den letzten Jahren hat sich ein großer und vielfältiger Druckermarkt entwickelt, der fast alle Ansprüche und Wünsche von privaten und kommerziellen Anwendern abdeckt. Der Preisverfall und die fortschreitende Entwicklung von qualitativ hochstehenden und immer effizienteren Druckerverfahren haben dazu geführt, daß fast jeder Benutzer ein Gerät auswählen kann, das für seine Zwecke geeignet ist und vor allem seinen Geldbeutel nicht überfordert.

Drucker sind wichtige und wirksame Hilfsmittel. Eine genaue und die jeweiligen Ansprüche berücksichtigende Auswahl ist deshalb erforderlich. Es hat keinen Zweck, für eine Sekretärin einen teuren Tintenstrahl-drucker zu kaufen, wenn ein Typenraddrucker für weniger Geld bessere Arbeit leistet. Genausowenig ist es sinnvoll, sich einen Thermodrucker zuzulegen, wenn Sie täglich Hunderte von Serienbriefen verschicken.

Im Laufe der Zeit hat sich gezeigt, daß private und semiprofessionelle PC-Besitzer in den meisten Fällen mit einem Matrix- oder Tintenstrahl-drucker gut bedient sind. Im professionellen Bereich kommt es auf das Aufgabengebiet an, in dem der Drucker eingesetzt werden soll; der Laserdruck ist auf jeden Fall das flexibelste der vorgestellten Druckverfahren. Neben der Druckqualität spielt in diesem Bereich zudem die Geschwindigkeit bei der Auswahl des Druckertyps eine entscheidende Rolle.

(Markus Zietlow/wn)

Hardware für Spieler

Während dem PC vor einigen Jahren in Spielerkreisen noch ein eher übler Ruf anhing, etablierte er sich in den USA mittlerweile als Spielmaschine Nummer eins. Durch die immer größere Verbreitung leistungsfähiger Grafikkarten und anderer Zusatzgeräte erreichte der PC auch bei den Spiele-Freaks unserer Breiten große Beliebtheit.

Die Meinung »spielen kann man auf jedem PC« ist weit verbreitet, doch sie stimmt nur in begrenztem Maße. Dabei ist es gerade bei einem PC sehr einfach, ihn für ein individuelles Einsatzgebiet speziell auszurüsten, da sich alle wichtigen Bausteine wie Grafikkarte oder Schnittstellen flexibel auf Erweiterungskarten befinden. Die Hardware-Komponenten des PC spielen für einen uneingeschränkten Spielgenuss eine wichtige Rolle. Nicht nur die Grafikkarte und der Massenspeicher, auch der Computertyp wirkt sich auf das Spielvergnügen aus. Ältere Spiele laufen auf hochgetakteten ATs oft zu schnell ab, weil sie die höhere Arbeitsgeschwindigkeit dieser Rechner nicht ausgleichen; doch ist dies heutzutage kein Problem mehr. Moderne Spiele messen nämlich zu Beginn die Geschwindigkeit des Rechners und fügen gegebenenfalls Warteschleifen ein. So kommt es, daß ein und dasselbe Spiel auf einem einfachen PC genauso schnell läuft wie auf einem hochmodernen 386-Computer.

Besonders bei komplexen Simulationen wie dem Flugsimulator III, die wegen der vielen notwendigen Rechenoperationen auf einfachen PCs recht gemächlich ablaufen, bringt ein schnellerer Computer für den Spieler große Vorteile. Doch was tun, wenn Sie einen AT Ihr eigen nennen und doch obengenannte Probleme mit älteren Spielen auftreten, die einfach zu schnell ablaufen? Auch dann müssen Sie nicht verzweifeln, denn in diesem Fall hilft Ihnen ein »Brems-Utility« weiter, also ein speicherresidentes Programm, das Ihren Rechner verlangsamt. Ein solches Hilfsprogramm ist zum Beispiel »Zeitlupe PC«.

Für den Spielefreund ist die verwendete Grafikkarte allerdings von viel größerer Bedeutung als die Geschwindigkeit des Rechners. Noch vor kurzem war die CGA-Farbgrafikkarte der Standard für Computerspiele, denn sie erfreute sich sehr weiter

Verbreitung und war die preiswerteste von allen – und sie unterstützte als erste PC-Karte die Farbdarstellung.

Ein CGA-ausgestatteter Personalcomputer kann jedoch Amiga-verwöhnte Augen nicht begeistern, denn die Auflösung von 320x200 Punkten mit vier (eingeschränkt wählbaren) Farben schlägt sogar noch der betagte Commodore 64. Dafür unterstützt nahezu jedes Spiel für den PC diesen Grafikmodus und bewegt sich samt Farbmonitor immer noch in preislich erschwinglichen Regionen.

Eine EGA-Karte bietet hohe Auflösung und große Farbvielfalt zu einem annehmbaren Preis

Die derzeit größte Verbreitung hat die Hercules-Karte und deren zahlreiche, kompatible Nachbauten aufzuweisen. Bei einer Auflösung von 720x348 Punkten ist die Hercules-Karte ein guter Kompromiß, wenn Sie Ihren Büro-PC hin und wieder zum Spielen verwenden möchten. Da eine Hercules-Karte aber nur einfarbige Grafiken darstellen kann, befriedigt sie einen überzeugten Spieler ganz und gar nicht, zumal viele ältere und auch einige neuere Spiele mit dieser Karte nicht oder nur sehr unzureichend zusammenarbeiten. Dieses Manko läßt sich zwar mit Hilfe eines CGA-Emulators, der eine CGA-Farbgrafikkarte simuliert, beheben, und so funktionieren die meisten Spiele auch mit der Hercules-Karte. Dabei erscheinen die Farben aber nur als Graustufen, und auch die Emulatoren bereiten bisweilen mit der einen oder anderen Hardware oder Software Schwierigkeiten.

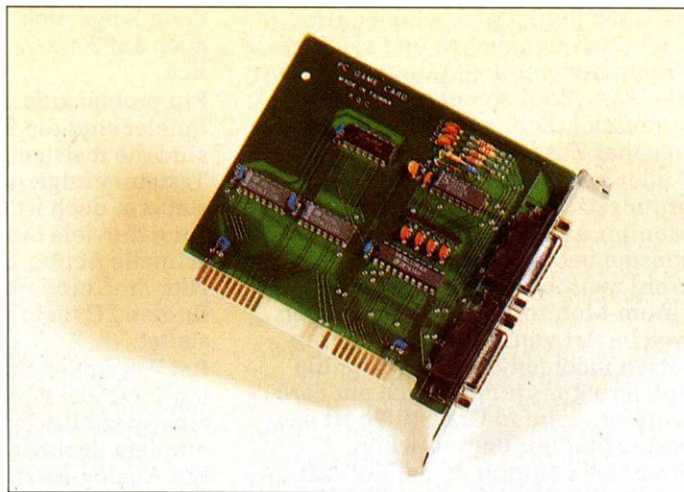


Bild 1. Eine »Gamecard« ist notwendig, um einen analogen Steuerknüppel anzuschließen

Seit dem rapiden Preisverfall der EGA-Karte wurde auch die Farbenvielfalt dieses Grafikmodus für Heim-anwender erschwinglich. Sie kostet inzwischen schon 350 Mark und weniger. Die EGA-Karte eignet sich besonders gut für Computerspiele, denn sie bietet eine große Auswahl an Farben (16 aus einer Palette von 64) bei gleichzeitig hoher Auflösung (bis zu 640x350 Punkte).

Die EGA-Karte zusammen mit einem EGA-Farbmonitor (ab 750 Mark) bildet ein gutes Gespann für Computerspiele, denn sie ist abwärtskompatibel zur CGA-Karte. Das heißt, nahezu alle Spiele, die eine CGA-Karte unterstützen, laufen auch auf einer EGA-Karte. Je nach Modell funktioniert dies auch mit sehr hardwarenahen Spielen, die so direkt auf die Grafikkarte zugreifen, daß selbst die Original-EGA von IBM passen muß. Die Zahl der Spiele, die mit einer EGA-Karte nicht zusammenarbeiten, ist allerdings sehr gering, bei neueren Spielen gibt es damit überhaupt keine Probleme mehr.

Das Manko, daß viele der ausdrücklich für EGA ausgewiesenen Spiele zwar die Farben, nicht aber die hohe Auflösung unterstützen, besteht nach wie vor. Der Grund liegt darin, daß die EGA-Karte mit einer Auflösung von nur 320x200 Punkten mit 16 Farben wegen der Ähnlichkeit zur CGA-Karte programmtechnisch einfacher zu programmieren ist. Doch zumindest einige Neuerscheinungen machen inzwischen auch von der höheren Auflösung und der größeren Farbauswahl Gebrauch.

Das Optimum für Spieler ist derzeit eine VGA-Karte samt passendem analogen Farbmonitor. Diese Grafikkarte ist abwärtskompatibel zur CGA- und EGA-Karte, bietet aber bis zu 256 Farben gleichzeitig (aus mehreren hunderttausend) und eine Auflösung von

640x480 Punkten. Auch hier gilt noch, daß sich derzeit erst wenige Programmierer die Mühe machen, die VGA-Modi zu nutzen, was sich voraussichtlich – und hoffentlich – in naher Zukunft ändern wird.

Außer bei der Hercules-Karte, die grundsätzlich nur einen Monochrom-Monitor unterstützt, haben Sie ansonsten bei jeder Grafikkarte die Wahl zwischen Farb- und Monochrom-Monitor. Unabhängig davon, welche Art von Grafikkarte Sie einsetzen möchten, lohnt sich für die Spielernatur grundsätzlich ein Farbmonitor, denn in Graustufen ist das beste Spiel nur halb so schön.

Ein guter Kompromiß bei der Wahl der Grafikkarte sind die sogenannten erweiterten Karten, die abwärtskompatibel sind. Die erweiterten Modi von bis zu 800x600 Punkten oder sogar mehr werden zwar so gut wie gar nicht von Spielen unterstützt, doch Ausnahmen bestätigen die Regel. Ein erweitertes EGA-Modell von Genoa bietet eine solche hohe Auflösung und zusätzlich einen speziellen CGA-Modus, in dem sie registerkompatibel zur Original-CGA-Karte ist. In diesem Modus laufen dann sogar Uralt-Spiele, die die Register der CGA-Karte direkt programmieren, um spezielle Effekte zu erreichen. So sind Profi-Anwendung und Spiel mit einer einzigen Karte abgedeckt.

Auch für Spiele-Freaks bringt eine Festplatte viele Vorteile

Ein kritisches Kapitel in Sachen Computerspiele sind Laptops, denn eine übliche Flüssigkristallanzeige ist zumindest für Action-Spiele absolut ungeeignet. Ein derartiges LC-Display benötigt nämlich eine längere Zeitspanne, um Änderungen des Bildschirminhaltes darzustellen, als der schnelle Phosphor in modernen Monitoren. Das Ergebnis ist ein mehr oder minder verschwommenes Bild, wobei alle sich rasch bewegenden Objekte einen kometenhaften Schweif hinter sich herziehen. Dieses Problem läßt sich bei den meisten Laptops durch den Einsatz eines externen Monitors beheben.

Zum Thema Massenspeicher gibt es nicht allzuviel zu sagen, inzwischen ist eine Festplatte auch für »Nur-Spieler« sinnvoll, da die Spiele immer umfangreicher werden und das Nachladen dann um ein Vielfaches beschleunigt wird. Außerdem müssen Sie sich dann nicht mehr ständig als »Diskjockey« betätigen. Bei den Diskettenlaufwerken ist das 5 1/4-Zoll-Format immer noch von Vorteil,

denn lange nicht alle Spiele sind auch auf 3 1/2-Zoll-Disketten erhältlich.

Ein problematisches Kapitel für den Spieler sind die Eingabegeräte. Zwar sind die meisten PC-Spiele mit der Tastatur einigermaßen akzeptabel zu steuern, doch ist diese Art der Steuerung für viele Simulationen oder für schnelle Action-Spiele indiskutabel. Hier muß ein Joystick her, was sich für den PC nicht ganz einfach gestaltet.

Bei PCs finden üblicherweise Analog-Joysticks Verwendung, die über eine spezielle Game-Adapter-Karte mit dem Rechner verbunden sind. Ein Analog-Joystick enthält für die x- und die y-Achse je ein Potentiometer (einen regelbaren Widerstand), der die Auslenkung des Joysticks durch unterschiedliche Widerstandswerte an den Computer übermittelt. Der PC bestimmt die Position des Joystick-Hebels, indem er einen Kondensator auflädt und durch den Widerstand des Joysticks entlädt. Dabei mißt das Programm die vergangene Zeit. Je größer der Widerstand, desto langsamer entlädt sich der Kondensator. Im Gegensatz dazu befindet sich in einem Joystick für die gängigen Heimcomputer (Atari oder Commodore) für jede Richtung ein Schalter, über die der Computer feststellt, in welche Richtung der Anwender den Joystick gerade bewegt.

Beide Prinzipien haben ihre Vor- und Nachteile. Analog-Joysticks informieren den Computer nicht nur über die Richtung, sondern auch über die Stärke der Bewegung. Wenn der Anwender den Analog-Joystick sehr weit nach links drückt, kann das entsprechende Programm darauf reagieren, bei einem Spiel also zum Beispiel die Spielfigur entsprechend schneller nach links bewegen.

Ein Digital-Joystick stellt im Gegensatz dazu nur fest, in welche Richtung der Anwender den Joystick aus-

lenkt; wie weit, das ist ihm völlig gleichgültig.

Theoretisch ist ein Analog-Joystick also für den Spieler günstiger als ein Digital-Joystick. Doch die Praxis sieht anders aus. Erstens sind Analog-Joysticks noch immer vergleichsweise teuer, da ihr mechanisches Innenleben weitaus komplizierter ist als das eines Digital-Joysticks. Außerdem benötigen Sie zum Anschluß eines Analog-Joysticks eine spezielle Erweiterungskarte, einen sogenannten »Game Adapter« (Bild 1), der zusätzlich mit zirka 80 Mark zu Buche schlägt. Andererseits ist die Abfrage des Analog-Joysticks sehr ungenau, weil von verschiedenen Faktoren abhängig. Die PC-Joysticks verwenden nämlich keine richtigen A/D-Wandler.

Joysticks sind bei PCs immer noch ein Problem

Aus diesem Grund befinden sich an Analog-Joysticks zwei Drehregler, mit deren Hilfe Sie den Joystick justieren. Diese Justierung ist aber nur selten von Dauer, je nach Taktschwindigkeit, residenten Programmen und der verwendeten Software ist oftmals eine Neueinstellung notwendig.

Bei den Analog-Joysticks gibt es inzwischen eine reichhaltige Auswahl; von etwa 50 Mark an erhalten Sie brauchbare Exemplare. Inzwischen gibt es auch einen Adapter, mit dem Sie einen einfachen Heimcomputer-Joystick an die Druckerschnittstelle Ihres PC anschließen und einfache Heimcomputer-Joysticks an Ihrem PC verwenden können.

Speziell für die Freunde von Flugsimulatoren gibt es »Maxx«, einen Joystick in Form eines Flugzeugsteuerknüppels (Bild 2), für den Sie allerdings einige hundert Mark berappen

Bild 2. Eine spezialisierte Form des Steuerknüppels ist »Maxx«. Das Flugwunder arbeitet wie ein gewöhnlicher Analog-Joystick



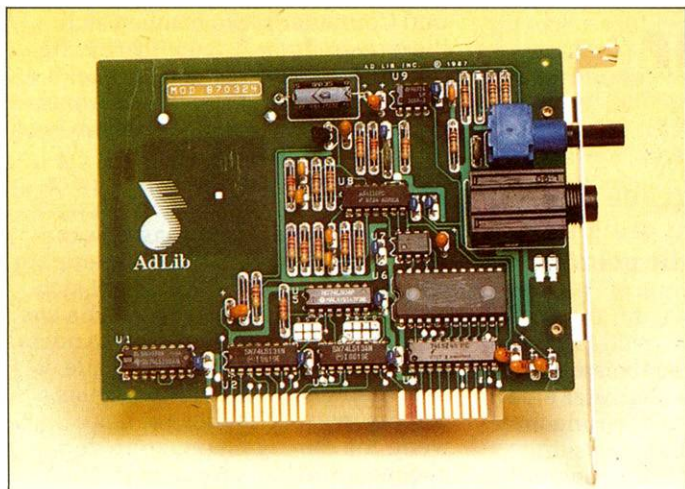


Bild 3. Stereo-Sound, der von immer mehr Spielen unterstützt wird: die Adlib-Karte

müssen. Dafür ist das Fluggefühl dann um einiges realistischer. Ganz neu – und vom Hersteller des Flugsimulators – sind die »Flight Controls«, die das Gefühl, in einem echten Cockpit zu sitzen, fast perfekt machen.

Da die Abfrage eines Analog-Joysticks recht aufwendig und auch zeitkritisch ist, gibt es eine Reihe von Spielen, die sich nur mit der Tastatur bedienen lassen. Insbesondere auf

Spiele aus der Public-Domain- und Shareware-Szene trifft dies sehr häufig zu.

Auch die Maus wird inzwischen von immer mehr Programmen unterstützt. Je nach Art des Spiels ist sie eine ernstzunehmende Konkurrenz für Joystick und Tastatur, jedoch nicht bei jedem Spiel sinnvoll. Ein letzter wichtiger Punkt ist die Tonausgabe bei PCs. Da bei der Entwicklung des PCs niemand an den

Einsatz als Spielecomputer dachte, verfügen PCs nur über einen einzigen Tonkanal, der einstimmige Musikuntermalung in minderer Qualität erlaubt.

Mit einigen Tricks und viel Aufwand läßt sich zwar softwareseitig eine gewisse Zweistimmigkeit oder die Wiedergabe von Geräuschen erreichen, insgesamt sind die Fähigkeiten des PC auf diesem Gebiet aber doch sehr beschränkt.

Aber auch hier ist Abhilfe erhältlich. Schon seit einiger Zeit sind diverse Sound-Karten für den PC auf dem Markt. Die bekannteste stammt von der amerikanischen Firma Adlib (Bild 3) und wird inzwischen von vielen namhaften Spieleherstellern unterstützt. So werden die neueren Sierra-Spiele von hervorragender Musikuntermalung begleitet, wenn eine solche Karte vorhanden ist. Mit etwa 500 Mark ist diese Erweiterung zwar nicht ganz billig, für die Freunde ansprechender Tonuntermalung aber durchaus empfehlenswert, zumal Musikprogramme zur Steigerung der eigenen Kreativität mitgeliefert werden.

(Martin Hepp/mk)

Sesam öffne dich!

BCi DisAsm erschließt jedes Programm!

Wandelt den Programmcode in ein dokumentiertes Assemblerlisting um. Mit normalen Editoren können Sie nun verändern und modifizieren, und mit gängigen Assemblern und Debuggern kann danach wieder assembliert werden.

Es gibt kaum einen, der BCi DisAsm das Wasser reichen kann!

Auch Experten kennen nicht immer die Lösung, aber sie wissen, wo sie suchen müssen.

BCi DisAsm hilft JEDEM, die verwirrenden und frustrierenden Software-Puzzles, welche sich beim Umgang mit dem Computer ergeben, zu lösen. Und zwar schneller als mit jedem anderen Produkt, das Sie kaufen können, egal, um welche Preisklasse es sich handelt.

Mit dem BCi DisAsm erwerben Sie sofort ein Know-how, daß Sie normalerweise Jahre an Erfahrung kosten würde. Versuchen Sie nicht, das Rad neu zu erfinden!

BCi DisAsm – Der intelligente Disassembler

BCi DisAsm ist ein intelligenter, selbsttätig dokumentierender MS-DOS-Disassembler. Die ausgefeilten Routinen im BCi DisAsm analysieren in kürzester Zeit selbst komplexen und umfangreichen Programmcode und zeigen dabei potentielle Problempunkte auf.

BCi
DisAsm₁₈₆
Intelligenter Disassembler Version 1.0



BCi DisAsm – Das Lernmittel

Sie brauchen kein Spezialist in Assembler-Programmierung zu sein. BCi DisAsm produziert aus jedem beliebigen Programm ihres IBM-kompatiblen Computers ein gut dokumentiertes, leicht verständliches Assemblerlisting.

BCi DisAsm – Einfach in der Handhabung

BCi DisAsm kann sowohl über Parameter von der DOS-Kommandozeile aus als auch interaktiv über Menüs gesteuert werden. Die Menüstruktur ist Lotus-kompatibel und selbstverständlich in Deutsch. Es besteht keine Notwendigkeit, neue Befehle zu lernen oder ständig im Handbuch nachzuschlagen. BCi DisAsm ist unmittelbar anwendbar!

Minimale Systemanforderungen

256 kByte RAM, 8088/8086/80186/80286 oder 80386 CPU, MS-DOS oder PC-DOS Version 2.0 oder größer

BCi DisAsm wird Ihnen bei den folgenden Schritten helfen:

1. Lernen Sie Assembler-Programmierung, wenn Sie möchten.
2. Entdecken Sie, warum Programm XYZ auf Ihrem Rechner nicht läuft, und tun Sie etwas dagegen.
3. Ändern oder entfernen Sie Programmteile.
4. Sichern Sie sich gegen destruktive Programme ab. Finden Sie einen "Virus", und entfernen Sie ihn.
5. Integrieren Sie Code aus kompilierten Programmen in eigene Anwendungen.
6. Erhöhen Sie die Kompatibilität Ihrer Software.
7. Passen Sie Software an verschiedene MS-DOS-Versionen an.

© 1987 BCi, West Germany und USA

Lotus ist eingetragenes Warenzeichen der Lotus Corp. MS-DOS ist eingetragenes Warenzeichen der Microsoft Corp.

Bitte benutzen Sie die Bestellkarte.

BCi DisAsm kostet inkl. eines sehr ausführlichen deutschen Handbuchs (mehr als 100 Seiten) nur

199,- DM (unverbindliche Preisempfehlung)
Wenn Sie über den DMV-Bestellservice bestellen, gilt folgendes:

Inland:		Ausland:	
Einzelpreis	199,- DM	Einzelpreis	199,- DM
zzgl. Versandkosten	4,- DM	zzgl. Versandkosten	6,- DM
Endpreis	203,- DM	Endpreis	205,- DM

DMV-Verlag · Postfach 250 · 3440 Eschwege



Welche Sprache für welchen Zweck?

Der Programmiersprachen gibt es gar viele, doch nicht alle sind für ein und dieselbe Problemlösung gleichermaßen gut geeignet. Dieser Beitrag vermittelt Ihnen einen Überblick über die gebräuchlichsten Sprachen und deren Eigenheiten.

Am Anfang einer Programmentwicklung liegt das zu lösende Problem. Sie werden sich üblicherweise Gedanken machen, in welcher Programmiersprache die Codierung erfolgen soll, und dann den entsprechenden Lösungsweg ausarbeiten. Häufig bestimmen dabei eher subjektive Kriterien die Wahl der Sprache, weil viele objektive Entscheidungspunkte oft nicht in ihrer Wichtigkeit erkannt werden. Das führt dann dazu, daß Sie ein Problem eventuell nicht optimal lösen. Die Wahl einer vielleicht im ersten Moment nicht besonders günstig erscheinenden Programmiersprache könnte dagegen für die Lösung viele Vorteile bringen.

Zur Zeit besitzen im PC-Bereich vier Sprachen eine besondere Bedeutung – Basic, Pascal, Fortran und C. Um die Verwendbarkeit einer Sprache für ein Problem überhaupt abschätzen zu können, sollten Sie diese unter verschiedenen Gesichtspunkten betrachten:

Sehr wichtig ist die Lesbarkeit einer Sprache. Soll ein anderer Programmierer Ihr Werk verstehen, wird ihm das bei unübersichtlichen Programmen verständlicherweise schwerfallen. Auch wenn ein Programm öfter geändert werden muß, ist eine gute Lesbarkeit von Vorteil. Erfahrungsgemäß ist es für einen Programmierer nach einer gewissen Zeit problematisch, sein eigenes Programm zu verstehen und nachzuvollziehen. Eine klare Darstellung des Programmtextes ist in solchen Fällen sehr hilfreich.

Obwohl sich in jeder Sprache sehr »sauber« programmieren läßt, gibt es doch einige, welche die Übersichtlichkeit durch ihre Struktur und Syntax besonders unterstützen. Hierzu gehört auch die Dokumentierbarkeit von Programmen. Damit ist nicht nur der Kommentar im Quelltext gemeint, sondern auch die Fähigkeit, Funktionen oder Variablen mit richtigen Namen zu benennen. Eine Variable mit dem Namen »ergebnis« ist sicher leichter zu deuten als »er«.

Bei großen Projekten ist es fast unerlässlich, auf die sogenannte »strukturierte Programmierung« zurückzugreifen. Dies sind zum einen aufwen-

dige Kontrollstrukturen, beispielsweise Mehrfachauswahlen oder eine Vielzahl von Schleifensteuerungen. Andererseits muß es möglich sein, einzelne Programmteile als sogenannte »Black Boxes« zu betrachten. Damit ist gemeint, daß es für den Programmierer unerheblich ist, welche Variablen in einer Prozedur verwendet werden oder wie mit den übergebenen Daten verfahren wird. Bekannt sein muß lediglich die Schnittstelle der jeweiligen Programmroutine.

Die Qualität eines Programms hängt in erster Linie von der Disziplin des Programmierers ab

Etwas Ähnliches sind die sogenannten »Units«, Programmteile, die außerhalb des momentan bearbeiteten Quelltextes stehen. Sie sind bereits übersetzt (compiliert) und werden bei der Programmbearbeitung nur zum fertigen Code hinzugefügt. Der Compiler erledigt, für den Benutzer unsichtbar, diese Aufgabe. So ist es möglich, den Code für ein Programm stark zu reduzieren und seine Übersichtlichkeit noch zu erhöhen.

Bei den Units gibt es mitgelieferte und selbstprogrammierte Teile. Die mitgelieferten sind im Handbuch dokumentiert und stellen oft eine Anzahl sehr nützlicher Befehle dar, die aber im eigentlichen Umfang der Programmiersprache nicht vorgesehen sind. Selbstprogrammierte Units entstehen aus der Forderung, immer wiederkehrende Befehle in Programmen nicht jedesmal neu schreiben zu müssen. Die notwendigen Routinen werden einmal programmiert und als Unit abgelegt. Wer sie verwenden möchte, braucht nur die entsprechenden Units in sein Programm einzufügen.

Die Umsetzung der angesprochenen Punkte hängt natürlich sehr stark vom jeweiligen Programmierer ab. Es ist nämlich einige zusätzliche Arbeit erforderlich, um ein gut dokumentiertes und lesbares Programm zu schreiben. Freizeitprogrammierer

und Computerfreaks machen sich diese – in ihren Augen überflüssige – Mühe meist nicht. Berufliche »Computerspezialisten« sind jedoch auf diese Hilfen angewiesen, um ein Chaos zu vermeiden.

Etwas, das bei der Betrachtung der Problemstellung oft vernachlässigt wird, ist der Entwicklungsaufwand für ein Programm. Wollen Sie nur ein einfaches Problem lösen, bietet sich eine Programmiersprache an, die wenige Formalismen beinhaltet. Anders verhält es sich, wenn schon Teile einer Lösung existieren. Dann gelangen Sie meist schneller ans Ziel, wenn Sie die Ergänzungen in der gleichen Sprache programmieren. Eine Zeitersparnis ist allerdings nur dann gegeben, wenn die fertigen Teile ausreichend dokumentiert sind. Ebenso ist es sinnvoll, bei Systemergänzungen in derselben Programmiersprache weiterzuarbeiten – auch wenn die neuen Programme eigenständig sind. So ist es zum Beispiel in der Industrie heute noch üblich, die Prozeßrechner für Steuerungen in Fortran zu programmieren. Ein neues Programm schreibt man dann am besten auch in Fortran; das erleichtert die Programmpflege erheblich.

Zum Entwicklungsaufwand gehört auch die Fehlersuche. Gerade in diesem Bereich gibt es sehr große Unterschiede zwischen den einzelnen Sprachen. Ein gutes Beispiel stellt Pascal dar, das dem Entwickler durch eine Vielzahl von eingebauten Testmechanismen fast jeden Fehler direkt am Entstehungspunkt meldet. Ähnlich gut ist die Fehlersuche in Basic gelöst. Im Gegensatz dazu steht C. Aufgrund der Tatsache, daß standardmäßig kaum Fehler abgefangen werden, kann die Fehlersuche situationsbedingt sehr langwierig ausfallen.

Nicht nur die Sprache ist entscheidend – spezielle Anwendungen erfordern spezielle Compiler

Für verschiedene Programmiersprachen gibt es Testsysteme, die umfangreiche Fehlerüberprüfungen während der Entwicklungsphase vornehmen. Mit der Größe des zu entwickelnden Programms wird es zunehmend wichtiger (wenn nicht gar zwingend notwendig), auf solche Prüfungssysteme zurückgreifen zu können. Stellen Sie sich nur einmal das Dilemma vor, in 4 MByte Quelltext einen Fehler suchen zu müssen. Da diese Systeme aber sehr teuer sind,

werden sie nur dann eingesetzt, wenn die Anwendung es erfordert. Eine weitaus billigere Methode ist es daher, eine gegen Fehler ausreichend geschützte Sprache zu verwenden. Für einige Anwendungen ist es von Bedeutung, auf die Ausführungszeit der Programme zu achten. Sicher haben Sie schon mehrmals die Aussage gehört oder gelesen, daß in C programmierte Lösungen die schnellsten seien. Für einen bestimmten Compiler mag dieser Sachverhalt zutreffen – aber es gibt eine Vielzahl von Compilern für Personalcomputer. Manchmal finden Sie daher eine besonders schnelle Routine für einen speziellen Zweck nur bei einem bestimmten Compiler einer Hochsprache. Bei ausgabeintensiven Programmen ist dann zum Beispiel ein Pascal-Compiler einer bekannten Firma zu bevorzugen, denn er hat die nachweislich schnellste Bildschirmausgabe.

Ihre Wünsche und Anforderungen bestimmen die Wahl, denn eine Universal-sprache gibt es nicht

Es gibt auch einen Unterschied zwischen Interpreter- und Compiler-Sprachen. Der Interpreter arbeitet das Programm Zeile für Zeile ab. Dabei setzt er die gerade bearbeitete Zeile in für den Computer verständliche Befehle, die Maschinenbefehle, um. Weil dies während des Programmlaufs immer wieder geschehen muß, verbringt der Interpreter alleine mit der Umsetzung viel Zeit. Ein Compiler dagegen übernimmt die Übersetzerarbeit einmal vor Ablauf des Programms und schreibt das fertige Maschinenprogramm als ausführbare Datei auf eine Diskette oder Festplatte. Die Ausführung dieser Programme ist daher viel schneller.

Allerdings erweist sich der Interpreter dann als vorteilhaft, wenn ein Programm nicht sehr lang ist und häufig geändert werden muß. Nach einer Änderung ist das Programm sofort wieder lauffähig, und Sie müssen nicht erst den Compiler- und Link-Vorgang abwarten.

Überhaupt ist es wichtig, auf die Qualität des verwendeten Compilers zu achten. So ist der Fall denkbar, daß ein Compiler bei besonderen Datenstrukturen Schwierigkeiten hat (beispielsweise umständliche und aufwendige Programmierung der Datenstrukturen). Erfordert die Aufgabenstellung die Verwendung dieser Strukturen, wird dies die Verwendung des ansonsten sehr guten Com-

piler ausschließen. Auch wenn Ihre Anwendung beispielsweise spezielle Grafikroutinen benötigt, kann es vorkommen, daß Sie auf ein oder zwei Compiler festgelegt werden, die diese Prozeduren zur Verfügung stellen. Beide Beispiele stehen für eine Vielzahl von Gründen, um unabhängig von der Sprache einem bestimmten Compiler den Vorzug zu geben. Besteht die Forderung, daß die entwickelten Programme portierbar – das heißt übertragbar auf andere Computer – sein müssen, schränkt dies die Wahl der Sprache zusätzlich stark ein. Wichtig ist hierbei, daß die Compiler auf den beiden Computersystemen einen sehr ähnlichen (am besten identischen) Sprachumfang besitzen und auch Programmformulierungen in gleicher Weise verarbeiten. Als hervorragendes Beispiel läßt sich hier C aufführen: Fast alle existierenden C-Compiler verarbeiten die gleichen Befehle.

Nach all diesen Punkten wird klar, daß es keine Sprache gibt, welcher – ohne nachzudenken – ein Vorzug eingeräumt werden kann. Einige Aufgabentypen lassen sich besonders elegant in einer bestimmten Programmiersprache lösen, aber universell übertragbar ist diese Aussage nicht. Es hängt immer wieder vom Einzelfall ab, welche Wahl Sie sinnvollerweise treffen. Um Ihnen die Auswahl zu erleichtern, wollen wir Ihnen die vier wichtigsten Sprachen im Hinblick auf die besprochenen Kriterien vorstellen.

Als erste und wohl noch am weitesten verbreitete Sprache ist Basic zu nennen. Entstanden ist Basic eigentlich als Sprache zum Lösen von mathematischen und technischen Aufgaben. Aufgrund der bewußt einfachen Konzeption und der leichten Erlernbarkeit erkannte man schnell, daß sie sich gut als Anfängersprache eignet. Ebenso schnell wurde klar, daß der ursprüngliche Befehlsumfang für viele Anwendungen zu knapp oder zu umständlich zu verwenden war. Jeder Hersteller erweiterte daher den von ihm verkauften Interpreter oder Compiler um mehr oder weniger sinnvolle Befehle. Das ist der Grund, warum heute ein Basic-Programm nur sehr selten auf anderen Computern läuft, als auf dem, auf dem es entwickelt wurde.

Die meisten Programmierer beherrschen heute Basic, da zumal bei fast jedem Computer ein Basic-Interpreter mitgeliefert wird. Die strukturierte Programmierung wird nur sehr selten unterstützt. Dadurch fallen große Programme meistens recht unübersichtlich aus. Weil jede Programmzeile einer Zeilennummer zugeordnet werden muß, ist es nicht einfach, Teile herauszulösen und in anderen

Programmen zu verwenden. Erst moderne Compiler unterstützen diese Art der Programmierung und machen Zeilennummern überflüssig.

Weil Basic eine Interpreter-Sprache ist (zumindest war sie es ursprünglich), ist es für kurze, schnell zu schreibende Programme prädestiniert. In letzter Zeit sind Basic-Compiler auf den Markt gekommen, die einige Vorzüge in sich vereinigen: Durch die geringe Übersetzungszeit bleibt der Vorteil der direkten Programmkontrolle bestehen, vereint mit der schnellen Ausführungszeit. Zusätzlich werden Prozeduren und Units eingeführt. Auf diese Weise ist es den Herstellern gelungen, die Gründe für den schlechten Ruf von Basic fast ganz zu beseitigen.

Pascal ist eine Programmiersprache, die von ihrem Schöpfer zunächst nicht für die Öffentlichkeit bestimmt war. Professor Niklaus Wirth an der ETH Zürich entwickelte eine Lehrsprache für seine Studenten, um ihnen anhand eines existierenden Beispiels die Grundzüge strukturierter Programmierung beizubringen. Tatsächlich wird dieses Stichwort immer in Zusammenhang mit Pascal verwendet. Daher war diese Sprache sehr auf Übersichtlichkeit zugeschnitten. In der Tat liest sich ein Pascal-Programm fast wie eine Dienstanweisung im Telegrammstil. Man erkannte sehr schnell, daß durch die klare Konzeption und flexible Gestaltung ein Entwicklungswerkzeug vorlag, nach dem man lange gesucht hatte.

Wer »strukturiertes Programmieren« sagt, muß auch »Pascal« sagen, und umgekehrt

Die Vorteile liegen in der sogenannten Syntax der Sprache. Der Programmierer wird förmlich dazu gezwungen, sein Programm in sehr übersichtlicher Form darzustellen. Alle Mittel der strukturierten Programmierung sind in Pascal enthalten. Durch den Verzicht auf Zeilennummern ist es kein Problem, Teile des Quelltextes zu verschieben.

Pascal ist eine Compilersprache. Die Ausführungsgeschwindigkeit der Programme ist entsprechend schnell. Leider war der Sprachumfang zu Anfang nicht so komplett, wie sich viele Anwender dies wünschten. Das führte – wie bei Basic – dazu, daß viele Compiler-Hersteller eigene Erweiterungen anboten. Obwohl es keinen wirklichen Pascal-Standard gibt, orientieren sich die verfügbaren

Compiler an dem von Wirth definierten Sprachstandard.

Ein Buchstabe ist in aller Munde – gemeint ist C. Auch diese Sprache wurde für einen bestimmten Zweck entwickelt. Bei der Erweiterung des Betriebssystems Unix stieß man an eine Grenze. Es war einfach unmöglich geworden, bei ungefähr 20 MByte Maschinencode die Übersicht zu behalten. Einige Entwickler entwarfen aus diesem Grund eine Programmiersprache, nämlich C, und schrieben einen Compiler dazu. Als das Betriebssystem auf den Markt kam, wurde einfach der Compiler mitgeliefert, damit die Kunden eigene Programme schreiben konnten. Aufgrund der Problemstellung bei der Entwicklung von C wurde darauf geachtet, den Programmierer so wenig wie möglich durch Formalismen einzuschränken. Andererseits steht aber eine Sprache zur Verfügung, die es ihm erlaubt, sehr klare Programme zu schreiben. C stellt eine interessante Mischung aus strukturierter Programmierung und unübersichtlicher Kurzschrift dar. Es hängt daher in besonderem Maße vom Programmierer ab, wie gut seine Programme für andere noch zu lesen sind. Beide Extreme sind realisierbar: Sie können Programme schreiben, die einem Pascal-Quelltext täuschend ähnlich sehen; es ist aber ohne weiteres machbar, vier bis fünf Befehle, ja sogar eine ganze Schleife in eine einzige Anweisung zu packen.

Weil C eine verhältnismäßig junge Sprache ist, sind die meisten heute benötigten Sprachelemente bereits in der Grunddefinition enthalten. Es existiert daher ein anerkannter Sprachstandard für C, an den sich alle ernstzunehmenden Hersteller halten. Somit läßt sich die Übertragung von Programmen auf andere Computersysteme in der Regel ohne Probleme durchführen.

Ob auf Lochkarten oder modernen Massenspeichern: In Industrie und Forschung wird Fortran nach wie vor eingesetzt

Als letzte zu erwähnende Sprache ist Fortran zu nennen. Obwohl sie in Heimanwenderkreisen nicht allzu-sehr verbreitet ist, besitzt sie doch eine recht große Bedeutung. Ursprünglich als mathematische Sprache gedacht, hielt sie ihren Einzug in die Computerwelt, als noch Lochkarten als Eingabemedium verwendet wurden. Darauf ist die Tatsache zurückzuführen, daß in der grundle-

Basic	Fortran
<pre> 10 defint eingabe, i 20 rem Dies ist ein Kommentar 30 input eingabe 40 if eingabe > 10 then eingabe=10 50 for i=1 to eingabe 60 print using "####";i 70 next i 80 end </pre>	<pre> PROGRAM SIMPLETEST INTEGER EINGABE,I DIES IST EIN KOMMENTAR C READ (*,*) EINGABE IF (EINGABE .GT. 10) EINGABE=10 DO 10,I=1,EINGABE WRITE(*,FMT='I4') I CONTINUE END 10 </pre>
Pascal	C
<pre> program SimpleTest; var eingabe, i : integer; { Dies ist ein Kommentar } begin readln(eingabe); if eingabe > 10 then eingabe:=10; for i:=1 to eingabe do writeln(i:4); end. </pre>	<pre> #include <stdio.h> int eingabe, i /* Dies ist ein Kommentar */ main() { scanf("%i",&eingabe); if (eingabe > 10) eingabe=10; for (i=1; i<=eingabe; i++) printf("%4i\n",i); } </pre>

Eingabe-, Ausgabe-, Bedienungs- und Schleifenanweisungen in Basic, C, Fortran und Pascal

genden Sprachdefinition keinerlei textdarstellende Hilfsmittel, wie zum Beispiel »Strings«, vorhanden waren. Eine besondere Stärke von Fortran war und ist die Behandlung von mathematischen Problemen. Inzwischen gibt es allerdings auch Fortran-Versionen, mit denen die Verarbeitung von Zeichenketten keine Schwierigkeiten mehr bereitet.

Weil viele Wissenschaftler und Techniker ihren ersten Computerkontakt über Fortran pflegten, ist der Gebrauch der Sprache in diesen Kreisen noch sehr weit verbreitet. Damit hielt die Sprache aber auch Einzug in die Industrie und ist dort in vielen Anwendungen sehr etabliert. Da Personalcomputer sich immer mehr in professionellen Anwendungsgebieten ausbreiten, wurde Fortran auf sie übertragen.

Vom Umfang der Programmstrukturierung ist Fortran in der Mitte zwischen Basic und Pascal einzuordnen. Die meisten Anwendungen in dieser Sprache werden von professionellen Programmierern geschrieben.

Oft werden die Sprachen Pascal und C ziemlich polemisch miteinander verglichen, da sie – bezogen auf den Befehlsumfang – annähernd gleiche Möglichkeiten bieten. Dennoch ist ein Urteil, welche Sprache die bessere ist, aus einigen Gründen nicht angebracht: Die Anhänger von Pascal schimpfen oft mit Recht über das furchtbare »Kauderwelsch« der Sprache C. C-Fans hingegen beschwerten sich dagegen häufig über die aufgeblähte Darstellungsweise von Pascal und darüber, daß einige hardwarenahe Funktionen nur sehr umständlich zu programmieren sind.

Beide Seiten haben nicht unrecht,

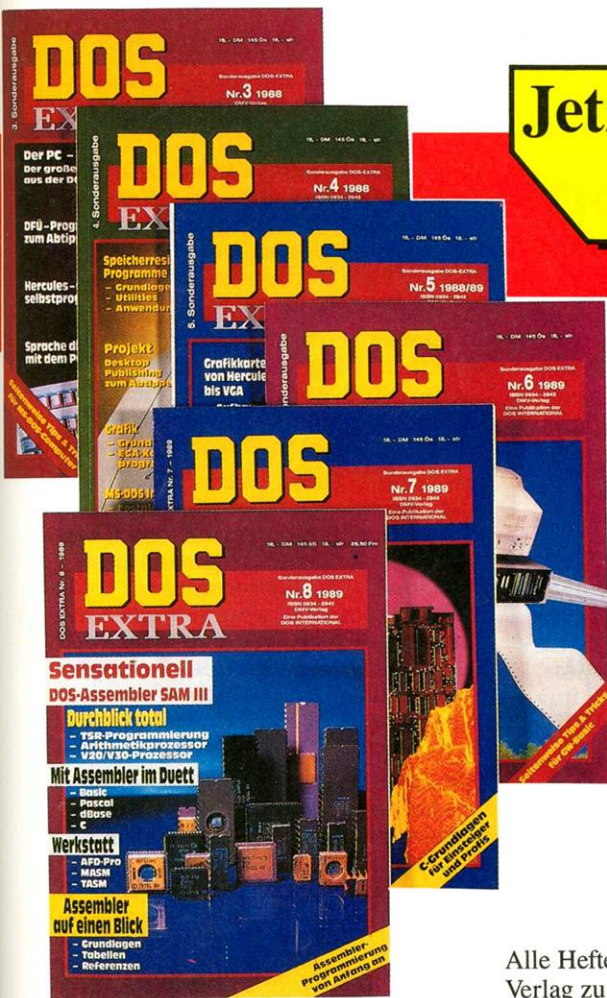
und doch sind die angeführten Punkte fast ausschließlich vom jeweiligen Programmierer abhängig. Die endgültige Entscheidung für eine bestimmte Programmiersprache treffen Sie am besten nach einer Abwägung, welche Vorteile beziehungsweise Nachteile im besonderen Problemfall stärker zählen.

Soll nur ein einziges Programm für eine bestimmte Aufgabe geschrieben werden, so ist die Wahl der Programmiersprache also ziemlich gleichgültig. Die Sprache, in der Sie sich am besten auskennen, stellt die zur Realisierung günstigste Wahl dar. Damit ist sichergestellt, daß Sie das gesteckte Ziel mit einem Minimum an Aufwand und Zeit erreichen.

Es wurden schon Textverarbeitungssysteme in Fortran geschrieben, obwohl diese Sprache auf den ersten Blick ungeeignet für eine solche Programmieraufgabe erscheint. Der entsprechende Programmierer hat fast alle ihm gestellten Aufgaben in dieser Sprache gelöst; so auch das Textverarbeitungssystem, weil er diese Sprache eben am besten beherrscht. Ganz anders sieht die Sache natürlich aus, wenn Programmteile für ein System geschrieben werden müssen. Sehr viele Rahmenbedingungen sind dabei zu berücksichtigen.

Bei der Diskussion um die Wahl der Programmiersprache sollten Sie sich immer vor Augen halten, daß es sich dabei nicht um eine Prestigefrage handelt – ganz nach dem Motto: »Alle programmieren in C, ich auch«. Es geht eigentlich immer nur um eines, nämlich darum, mit dem geringsten Aufwand das gestellte Problem zu lösen.

(Guido Weckwerth/zi)



**Jetzt auch im
Abo**

DOS EXTRA

**Immer wieder
aktuell**

- 3 Der PC - glasklar
- 4 MS-DOS intern
- 5 Von Basic bis BIOS
- 6 Alles über Basic
- 7 Power für Programmierer
- 8 Assembler von Anfang an



Alle Hefte und Databoxen sind direkt über den
Verlag zu beziehen. DOS Extra 3 bis 9

18,- DM (unverbindliche Preisempfehlung) Wenn Sie über den DMV-Bestellservice bestellen, gilt folgendes:			
Inland:		Ausland:	
Einzelpreis	18,- DM	Einzelpreis	18,- DM
zzgl. Versandkosten	4,- DM	zzgl. Versandkosten	6,- DM
Endpreis	22,- DM	Endpreis	24,- DM

Bitte benutzen Sie die Bestellkarte.

Databox-Inhalte der DOS Extra Nr. 9

- Scheck 2.01: Checksummer für geprüfte Eingaben
- Definitive Eingabe für Batchdateien
- Minidatenbank per Batchdatei
- Utility für komfortable MS-DOS-Befehlseingabe
- Utility für sicheres Dateilöschen
- Pulldown-Menüs unter GW-Basic programmieren
- Briefbogen selbstgemacht
- Analoguhr für Selfmade-Designer
- Konvertierung vom EGA- in den Hercules-Modus: Schraffur statt Farbe
- Sowie die begleitenden Listings aller anderen Beiträge

Alle Listings auf
Datenträger erhältlich

DOS

TASCHENBÜCHER

**Das "goldene" DOS Extra 1
ist wieder erhältlich!**
- als Taschenbuch immer griffbereit

Anstelle des ausverkauften,
legendären Sonderheftes DOS
Extra 1 bringt Ihnen das erste
DOS-Taschenbuch alle wichtigen
Beiträge, Illustrationen, Tabellen
und Listings in neuer ergonomi-
scher Form - platzsparend und
immer griffbereit.

DOS-Taschenbuch Band 1

Unter dem Titel "Von Assembler
bis MS-DOS" finden Sie in dem
neuen DOS-Taschenbuch Band 1
alles Notwendige über die Inter-
na Ihres PCs. Sichern Sie sich
ein Exemplar - noch heute!

29,80 DM (unverbindliche Preisempfehlung) Wenn Sie über den DMV-Bestellservice bestellen, gilt folgendes:			
Inland:		Ausland:	
Einzelpreis	29,80 DM	Einzelpreis	29,80 DM
zzgl. Versandkosten	4,- DM	zzgl. Versandkosten	6,- DM
Endpreis	33,80 DM	Endpreis	35,80 DM

Bitte benutzen Sie die Bestellkarte.



Das Programmierlexikon

Sie haben sich also dazu durchgerungen, Ihren PC selbst zu programmieren. Da Sie bereits über Grundkenntnisse in Basic verfügen, machen Sie sich zuversichtlich ans Werk. Doch bereits den ersten Blicken in die Fachliteratur folgt die Ernüchterung: Die vielen – zudem meist englischen – Begriffe entpuppen sich schnell als Fachchinesisch. Unser Programmierlexikon hilft Ihnen dabei, im Dschungel der Fachbegriffe nicht die Orientierung zu verlieren.

Adresse: Ein numerischer Wert, der die Position einer Speicherstelle innerhalb des Arbeitsspeichers angibt. Bei den Prozessoren der 80x86-Familie wird die Adresse durch zwei 16-Bit-Werte gebildet. Der erste Wert repräsentiert dabei die Segment- und der zweite die Offset-Adresse. Die Position einer Speicherstelle ermittelt man, indem die Segmentadresse mit 16 multipliziert und die Offset-Adresse addiert wird.

Adreßbereich: Die Anzahl an Speicherstellen, die ein Prozessor ansprechen kann.

Arbeitsspeicher: Die Menge an schreib- und lesbarem internen Speicher, der dem Computersystem zur Verfügung steht. Das Betriebssystem MS-DOS verwaltet maximal 640 KByte Arbeitsspeicher.

ASCII: Der »American Standard Code for Information Interchange« ist eine standardisierte Zuordnung von Symbolen (Buchstaben, Zahlen, Satz- und Sonderzeichen) zu numerischen Werten im Bereich von 0 bis 255.

BIOS: Das »Basic Input Output System« ist in einem nur lesbaren Speicherbaustein (ROM) des PC untergebracht. Es stellt elementare Funktionen zur Kommunikation mit den einzelnen Komponenten des Systems (Tastatur, Bildschirm, parallele und serielle Schnittstelle) zur Verfügung.

Cache (Pufferspeicher): Der Cache-Speicher (auch kurz »Cache« genannt) ist ein Zwischen- oder Pufferspeicher mit besonders schnellem Zugriff. Er vermeidet den wiederholten Austausch häufig verwendeter Daten und erhöht somit die Gesamtleistung des Computers.

Call by reference (Referenzaufruf): Aufruf eines Unterprogramms, bei dem nur die Adressen der verwendeten Parameter übergeben werden.

Call by value (Wertaufzuruf): Aufruf eines Unterprogramms, bei dem die tatsächlichen Werte der Parameter übergeben werden.

Chaining (Verkettung):

– Eine Methode zur Ausführung von Programmen, die größer als der Arbeitsspeicher sind. Dabei werden die

einzelnen Programm-Module sequentiell (nacheinander) geladen und ausgeführt.

– Eine Methode zur Vergabe von Prioritäten bei Interrupts. Jede Geräteeinheit, die ein Interrupt-Signal sendet, leitet das Quittungssignal des Prozessors entweder weiter oder blockiert es. Die Einheit, die dem Prozessor am nächsten ist, erhält damit automatisch die höchste Priorität.

Com-Dateien: Ausführbare Programme, die auf einen 64 KByte großen Speicherbereich (Speichersegment) beschränkt sind. Code-, Daten- und Stack-Bereich (Stapelspeicher) sind in diesen 64 KByte untergebracht.

Compiler: Ein Programm, das Anweisungen einer höheren Programmiersprache (Basic, C, Pascal und so weiter) in eine Folge binärer Anweisungen umwandelt (Objektcode), die der Prozessor direkt »versteht«.

Direkte Adressierung: Hierbei wird die Offset-Adresse einer Speicherstelle direkt, zum Beispiel durch einen symbolischen Namen, angegeben. Die Assembleranweisung »mov ax,summe« lädt das AX-Register mit dem Inhalt der Speicherstelle, deren Adresse durch den Namen »summe« bezeichnet ist.

DTA: Die »Disk Transfer Area« wird als Datenübertragungsbereich bei Datei- und Verzeichniszugriffen benötigt. Ihre Größe ist nicht festgelegt, sondern hängt von den ausgeführten Operationen ab. Das Betriebssystem MS-DOS legt nach dem Start eines Programms die DTA in die Speicherstelle 128 des PSP. Dort stehen ihr 128 Byte zur Verfügung.

Editor: Ein Programm zur Eingabe und Bearbeitung (Einfügen, Löschen, Verändern) von Texten.

Effektive Adresse: Die effektive Adresse entspricht dem Offset-Anteil einer physikalischen Adresse. Es handelt sich dabei um eine vorzeichenlose 16-Bit-Zahl.

Emulation: Eine Simulation in Echtzeit. Wenn ein Computer vom Typ A einen Computer vom Typ B emuliert, dann bedeutet dies, daß er sich genauso verhält wie Typ B.

Environment (Umgebungsbereich): Der MS-DOS-Umgebungsbereich ist ein Speicherbereich, der Informationen enthält (beispielsweise Angaben zum Suchpfad), die allen Stapeldateien und Programmen zugänglich sind.

Exe-Dateien: Ausführbare Programme, die beliebig lang sein dürfen. Der Code-, Daten- und Stack-Bereich ist dabei jeweils in einem eigenen Speichersegment untergebracht.

Far-Befehle: Damit werden Befehle in Maschinensprache bezeichnet, die als Adresse für eine Variable oder ein Unterprogramm eine Segment- und eine Offset-Adresse verwenden (32-Bit-Adresse). Dadurch lassen sich Unterprogramme ansprechen, die sich in einem anderen Speichersegment befinden, das heißt, weiter als 64 KByte von der Aufrufadresse entfernt sind.

FAT: Die »File Allocation Table« ist eine Tabelle, die auf jeder Diskette oder Festplatte vorhanden ist und dem Betriebssystem MS-DOS angibt, welche Bereiche des Speichermediums noch frei und welche belegt oder fehlerhaft sind.

Funktion: Ein logisch und funktionell getrennter Programmteil (Unterprogramm), der als Ergebnis seines Aufrufs einen Wert (Funktionsergebnis) an die aufrufende Stelle zurückliefert.

Garbage Collection: Eine Programmroutine, die – als Bestandteil eines jeden Basic-Interpreters – den Variablenspeicher eines Programms nach nicht mehr benötigten Variablen durchsucht und neu organisiert.

Gleitkommadarstellung: Die Darstellung von Dezimalzahlen in einem Format mit fester Länge (beispielsweise 48 Bit). Dabei wird eine Zahl »N« normalisiert und in Form einer Mantisse »M« und eines Exponenten »E« dargestellt:

$$N = M * e^E$$

wobei gilt:

$$0,1 \leq M < 1$$

Globale Variable: Eine Variable, deren Name, und damit auch ihr Wert, im gesamten Programm bekannt ist.

Handle: Ein numerischer Wert, der als Schlüssel für den Zugriff auf Dateien und Geräteeinheiten dient. Benutzt ein Programm eine Funktion zum Öffnen oder Anlegen einer Datei, so übergibt ihm MS-DOS den benötigten Schlüssel.

Hardware-Interrupt: Ein Interrupt, der durch die Hardware des PC (Tastatur, Festplatten- oder Disketten-Controller) ausgelöst wird und damit eine bestimmte Reaktion einleitet.

Hexadezimalsystem: Ein Zahlensystem mit der Basis 16, das heißt, es

wird von 0 bis 15 gezählt, bevor ein Stellenübertrag geschieht. Zur Darstellung finden die Ziffern »0« bis »9« und die Buchstaben »A« bis »F« Verwendung.

Indirekte Adressierung: Hierbei wird die effektive Adresse in Form eines Registereintrags gebildet. Der Assemblerbefehl »mov ax,[bx]« lädt den Inhalt einer Speicherzelle, die wiederum durch den Inhalt des Registers BX adressiert ist, in das AX-Register.

Indizierte Adressierung: Die effektive Adresse wird hier durch die Addition des Inhalts eines Registers mit einer Konstanten gebildet. Die Assembleranweisung »mov al,feld[di]« lädt beispielsweise das AL-Register mit dem Inhalt der Speicherstelle, deren Adresse sich aus der Summe des Inhalts von Register DI mit dem Konstantenwert »feld« errechnet.

Integrierte Entwicklungsumgebung: Eine Benutzeroberfläche, welche die Programmierungskomponenten Editor, Compiler und Linker zusammenfaßt. Ein Programm läßt sich somit schreiben, übersetzen und zu einer lauffähigen Com- oder Exe-Datei zusammenbinden, ohne daß die dazu notwendigen Entwicklungswerkzeuge und Hilfsprogramme getrennt aufgerufen werden müssen.

Interpreter: Ein Übersetzungsprogramm, das die Anweisungen einer höheren Programmiersprache ausführt. Da die Anweisungen vor jeder Ausführung übersetzt werden müssen, sind interpretierte Programme deutlich langsamer als kompilierte. Zudem sind sie nicht selbst lauffähig, sondern benötigen zu ihrer Ausführung den Interpreter.

Interrupt: Ein Interrupt unterbricht ein laufendes Programm. Daraufhin wird die entsprechende Interrupt-Bearbeitungsroutine ausgeführt und anschließend das unterbrochene Programm weitergeführt.

Interrupt-Vektortabelle: Eine Tabelle, in der die Adressen von Programmroutinen verzeichnet sind, die beim Auftreten eines Interrupts ausgeführt werden sollen. Jeweils zwei Worte (vier Byte) bilden die vollständige Adresse (Segment- und Offset-Adresse) von insgesamt 256 Interrupt-Routinen.

KByte: Ein Kilobyte sind 2^{10} oder 1024 Byte.

Konstante: Im Gegensatz zum symbolischen Wert bezeichnet die Konstante einen expliziten Wert in einer Anweisung. Allgemein bezeichnet eine Konstante aber auch jede Größe, deren Wert sich im gesamten Programm nicht ändert.

Linker (Binder): Ein Programm, das getrennt geschriebene Programme

und/oder Unterprogramme zu einer Einheit verbindet, die sich in den Arbeitsspeicher laden und eigenständig ausführen läßt.

Lokale Variable: Eine Variable, deren Gültigkeit auf einen bestimmten Programmteil (Unterprogramm) beschränkt ist.

MByte: Ein Megabyte sind 2^{10} KByte oder $1024 \times 1024 = 1048576$ Byte.

MS-DOS: Das »Disk Operating System« der Firma Microsoft ist ein Betriebssystem für Computer, die auf den Prozessoren der Familie 80x86 von Intel basieren.

Multitasking: Unter dem Begriff Multitasking versteht man die (scheinbar) gleichzeitige Ausführung von Programmen.

Near-Befehle: Befehle in Maschinsprache, die als Adresse einer Variablen oder eines Unterprogramms nur eine Offset-Adresse (16-Bit-Adresse) verwenden. Auf diese Weise können nur Unterprogramme im aktuellen, 64 KByte großen Speichersegment aufgerufen werden.

Nibble: Ein Byte wird in zwei Nibble unterteilt: das Low-Nibble belegt die Bits 0 bis 3 und das High-Nibble die Bits 4 bis 7.

Overlay: Eine Art der Speicherverwaltung, bei der verschiedene Programmteile den gleichen Speicherbereich benutzen. Die Programmteile dürfen allerdings nicht zur gleichen Zeit benötigt werden. Wird ein derartiger Programmteil aufgerufen und befindet er sich gerade nicht im Arbeitsspeicher, so wird er zur Laufzeit des Programms dynamisch nachgeladen.

Paragraph: Ein Block von 16 Byte Länge, der an einer durch 16 teilbaren Speicheradresse (0, 16, 32 und so weiter) beginnt.

Parameter: Eine Variable, die zur Übermittlung von Informationen zu oder von einem Unterprogramm verwendet wird.

Parität: Ein einfaches Verfahren zur Fehlererkennung. Das Paritätsbit, eine 1-Bit lange Einheit gibt an, ob die Anzahl der Einsen in einem Byte gerade oder ungerade ist.

Ports: Ports stellen die Verbindung zwischen dem Prozessor und den verschiedenen Bausteinen (Tastatur, Video-Controller, serielle und parallele Schnittstellen) des Systems dar. Jedem Baustein sind ein oder mehrere Ports zugewiesen, die sich jeweils unter einer spezifischen Adresse ansprechen lassen. Der Prozessor kommuniziert mit den Bausteinen, indem er Werte in die Ports schreibt oder aus ihnen liest.

Prozedur: Ein logisch und funktionell getrennter Teil eines Programms (Unterprogramm). Es ist möglich, der Prozedur beim Aufruf Parameter zu übergeben.

PSP: Das »Program Segment Prefix« ist eine Datenstruktur von 256 Byte Länge, die jeder ausführbaren Com- oder Exe-Datei (Programmdatei) im Arbeitsspeicher vorangestellt wird. Sie ist jedoch nicht zusammen mit der Datei auf einem externen Speichermedium (Diskette, Festplatte) gespeichert.

Real time processing (Echtzeitverarbeitung): Hierbei reagiert ein System (Hardware oder Software) augenblicklich auf die gerade aktuellen Gegebenheiten und ablaufenden Prozesse.

Register: Speicherstellen innerhalb des Prozessors, auf die der Prozessor wesentlich schneller zugreifen kann als auf Speicherstellen des Arbeitsspeichers.

Scan-Code: Der Scan-Code wird dem Prozessor beim Betätigen einer Taste übergeben. Es handelt sich dabei nicht um den der Taste entsprechenden ASCII-Wert, sondern um einen von der Position der Taste innerhalb der Tastatur abhängigen Wert.

Segmentregister: Die Prozessoren der 80x86-Reihe verfügen über vier 16-Bit breite Segmentregister. Jedes dieser Register definiert den Anfang eines 64 KByte großen Speichersegments. Entsprechend ihrer Verwendung erhielten die Segmentregister die Bezeichnungen DS (Daten-), ES (Extra-), CS (Code-) und SS (Stack- oder Stapelspeichersegment).

Software-Interrupt: Ein Interrupt, der sich aus jedem Programm heraus mit Hilfe der Int-Anweisung des Prozessors auslösen läßt.

Speicherresident: Siehe TSR

TSR-Programme: Ein Begriff, der sich von »Terminate but Stay Resident« ableitet. Darunter versteht man Programme, die nach der Ausführung (der Initialisierungsphase) resident im Arbeitsspeicher verbleiben. Sie lassen sich durch eine bestimmte Taste(nkombination) oder einen Interrupt aktivieren und unterbrechen dann das gerade aktive Programm, das nach Ablauf der TSR-Routine fortgesetzt wird. Es entsteht so der Eindruck einer gleichzeitigen Programmausführung (Multitasking).

Variable: Eine symbolisch benannte Einheit, der sich, je nach Art, ein oder mehrere Werte zuweisen lassen.

Word: Ein vorzeichenloser, 16 Bit breiter Datentyp. Mit ihm lassen sich Werte im Bereich von 0 bis 65535 darstellen

(zi)

Das Labyrinth der technischen Daten

Computerwerbung und was dahintersteckt

Die Leistungsfähigkeit eines PC wird mit einer Vielzahl von technischen Daten beschrieben, die für den Einsteiger oft verwirrend ist. In vielen Produktanzeigen werden zudem einige wichtige Leistungsmerkmale eines Rechners gar nicht erwähnt. Um die beschriebenen Leistungsmerkmale eines Computers richtig einschätzen zu können, müssen Sie die »Produktanzeigen-Kryptographie« entschlüsseln können.

Vor dem Weg in den Computerladen mit dem Ziel, sich einen Computer zuzulegen, steht erst die Entscheidungsfindung: Welcher Rechner soll es sein, wie will ich ihn einsetzen? Hierbei sind einige Vorüberlegungen zu treffen. Die wohl wichtigste und erste Entscheidung ist die Wahl des Prozessors, mit dem der Rechner arbeiten soll – Sie müssen, mit anderen Worten, feststellen, ob für Ihre Bedürfnisse ein PC ausreicht oder ein schnellerer AT notwendig ist.

In den »landläufigen« PCs werden zwei Prozessorarten verwendet. Der preiswertere ist der Intel 8088, der lediglich einen 8 Bit breiten Datenbus besitzt. Dadurch können jeweils nur Datenpakete von jeweils 8 Bit verarbeitet werden, und damit ist er – wenn auch preiswerter – nur halb so schnell wie sein »größerer Prozessor-Bruder«, der schnellere und auch teurere Intel 8086. Der 8086-Prozessor verfügt nämlich über einen 16 Bit breiten Datenbus mit 16 Adreßleitungen, wodurch die Daten 16-Bit-weise weitergeleitet werden können. Dies macht sich natürlich in der Arbeitsgeschwindigkeit bemerkbar. Alternativ zu diesen Prozessoren kann auch ein NEC V20 (anstelle des 8088) oder ein NEC V30 (anstatt eines 8086) Verwendung finden, deren Leistungsmerkmale identisch mit den Merkmalen der Intel-Prozessoren sind.

Eine weitere Alternative wäre der etwas exotische 80186, der ebenfalls eine vergleichbare Geschwindigkeit bietet. Er ist jedoch wegen seiner geringen Verbreitung und aus Gründen der Kompatibilität weniger interessant.

Der Nachfolger des 8088/8086-Prozessors ist der 80286 von Intel, der sich mittlerweile zum Standard für ATs entwickelt hat. Er verarbeitet 16 Bit parallel und ist mit 16 Datenbusleitungen ausgestattet, was sich in der Arbeitsgeschwindigkeit niederschlägt. Außerdem trägt er höhere Taktfrequenzen als seine Vorgänger und ist dementsprechend schneller.

Der nächstschnellere Prozessor ist dann der 80386. Er besitzt einen 32-Bit-Bus und kann demnach auch 32 Bit gleichzeitig verarbeiten und adressieren. Daraus resultiert erstens wiederum eine deutlich höhere Geschwindigkeit und gleichzeitig auch, aufgrund des größeren Aufwands, ein deutlich höherer Preis. Leider unterstützt das Betriebssystem MS-DOS diese 32-Bit-Architektur nicht, das heißt, eine deutliche Leistungssteigerung gegenüber dem 80286 ist unter MS-DOS nicht zu erreichen.

Der derzeitige Spitzenkandidat dieser Entwicklungsreihe von Intel ist der neue 80486-Prozessor. Ein mathematischer Coprozessor ist hier bereits integriert.

Aber nicht nur die Architektur des Prozessors ist für die Leistungsfähigkeit eines Rechners wichtig. Auch die Taktfrequenz spielt eine große Rolle. In der Regel gilt: Je höher der Rechner getaktet ist, um so schneller ist er auch. Häufig wird in der Werbung jedoch nicht nur die tatsächliche Taktfrequenz angegeben, sondern auch die Taktfrequenz nach dem sogenannten »Landmark-Test«. Bei diesen Angaben handelt es sich um die Geschwindigkeit relativ zum Original-IBM-PC gesehen, die dann wiederum in MHz umgerechnet wird. Sie ist oft etwas höher als die tatsächliche Taktfrequenz.

Leider bewirkt eine höhere Taktfrequenz nicht immer eine höhere Geschwindigkeit. Verschiedene Speicherbausteine sind nämlich nicht auf den hohen Rechnertakt ausgelegt, und dann werden sogenannte Wartezyklen eingelegt. Das bedeutet, daß der Prozessor ein oder mehrere Takte lang warten muß, bis der langsamere Speicherbaustein »nachgekommen« ist. Diese Wartezyklen werden auch als »Waitstates« bezeichnet. Je mehr Waitstates der Rechner also hat, desto mehr wird die Gesamtgeschwindigkeit des Systems gebremst.

Achten Sie auf die tatsächliche Taktfrequenz, der Landmark-Test täuscht

Ein wichtiger Faktor ist auch, mit wieviel KByte (oder MByte) Arbeitsspeicher der Rechner ausgestattet ist. Bezüglich der Speichergröße finden Sie in Anzeigen oft zwei Angaben. Der eine Wert ist die Größe des tatsächlich vorhandenen Speichers, über den Sie dann direkt nach dem Kauf verfügen. Der andere Wert sagt aus, auf welche Speicherkapazität Sie den Rechner auf der Mutterplatine erweitern können. Diese zusätzlichen Speicherbausteine müssen Sie jedoch erst kaufen; zudem werden solche nachgerüsteten Bausteine über den Systembus angesprochen, der oft mit einer sehr viel niedrigeren Taktfrequenz arbeitet als der Prozessor. Das bedeutet dann wieder Wartezyklen – und damit Zeitverlust.

Diese Angabe über die Erweiterbarkeit ist jedoch keineswegs nur dazu vorhanden, um die Kundschaft zu verwirren. Denn sollten Sie später einmal mehr Speicherplatz benötigen, rüsten Sie Ihren AT einfach auf. Mehr Platz ist immer nützlich und auch im allgemeinen nicht viel teurer. Bestehen Sie also darauf, daß Ihr AT für den Einbau von mindestens zwei MByte auf der Hauptplatine eingerichtet ist.

Grafikkarte	mögliche Auflösung	max. Farbangebot
Monochromadapter	nur Text	zweifarbige
Hercules-Karte	720x348	zweifarbige
CGA-Karte	320x200,640x200	16
EGA-Karte	320x200,640x200,640x350	64
VGA-Karte	320x200,640x200,640x350,640x480	262144

Tabelle 1 zeigt die wichtigsten PC-Grafikkarten im Überblick

Wenn Sie sich für einen Rechner mit einer Festplatte entscheiden, die gegenüber Diskettenlaufwerken einen deutlichen Geschwindigkeitsvorteil bringt, achten Sie nicht nur auf deren Speicherkapazität, sondern auch auf die mittlere Zugriffszeit, die Zeit also, die notwendig ist, um Daten von der Festplatte zu holen. Schließlich soll die Festplatte ja nicht nur mehr Daten als eine Diskette speichern können, sondern auch die Ladezeiten verkürzen.

Achten Sie bei Ihrer Festplatten-Controller-Kombination auch auf die Datentransferrate

Bei Festplatten von 20 MByte sollte die Zugriffszeit höchstens 65 ms betragen, besser wären 40 ms. Ab einer Größe von 60 bis 80 MByte sollten Sie auf eine Zugriffszeit von höchstens 28 ms bestehen. Die Zugriffszeit einer Festplatte kann nie zu klein sein!

Erkundigen Sie sich auch nach der mitgelieferten Grafikkarte, denn hier sind die Preis- und Qualitätsunterschiede recht groß. Auch arbeiten viele Programme, die im Grafikmodus laufen, nicht mit allen Grafikkarten zusammen. Textdarstellung ist hingegen immer mit allen Grafikkarten möglich (Tabelle). Grundsätzlich können sie davon ausgehen: Je höher die Auflösung des Bildes ist, desto besser ist die Qualität.

Es gibt einfache Textkarten oder Monochromadapter, die nicht grafikfähig sind. Sie können damit lediglich Text mit unterschiedlichen Attributen, aber in hervorragender Qualität darstellen (zum Beispiel unterstrichen oder aufgehellt). Diese Karten haben jedoch weder ein gutes Preis-Leistungsverhältnis, noch werden sie oft angeboten.

Wem die reine Textdarstellung nicht ausreicht, für den ist die preiswerte CGA-Karte (Colour Graphics Adapter) interessant. Sie wartet neben einer allerdings wenig augenfreundlichen Textdarstellung auch mit farbigen Grafiken auf. Sie kann bis zu 16 Farben darstellen. Der niedrige Preis macht sich aber leider auch hier durch die geringe Auflösung bemerkbar (bis zu 640x200 Bildpunkte). Dennoch ist diese Karte sehr verbreitet und wird dementsprechend von den meisten Programmen unterstützt. Ein weiterer Standard ist die Hercules-Grafikkarte, die eine sehr hohe Grafikaufklärung (720x348) und eine ausgezeichnete Textdarstellung bietet – leider nur monochrom. Diese

Karte ist für die Spieler unter Ihnen weniger geeignet, da nur die wenigsten Spiele diese Karte unterstützen. Hier hilft allenfalls eine Notlösung: Sogenannte CGA-Emulatoren simulieren den CGA-Modus auf Ihrer Hercules-Karte. Diese Emulatoren rechnen die Farbwerte der CGA-Grafik in Helligkeitsstufen um, die dann – dank der hohen Auflösung der Hercules-Karte – als Raster dargestellt werden können. Ganz problemlos geht dies jedoch nicht, da die verschiedenen Emulatoren bei so mancher Hardware-Konfiguration oder auch bei bestimmten Programmen etwas »heikel« sind. Hier geht Probieren über Studieren.

Eine von IBM entwickelte Farbgrafikkarte namens EGA ist qualitativ um Klassen besser. Sie besitzt ein Farbangebot von 64 Farben gleichzeitig, und das bei einer Grafikaufklärung von bis zu 640x350 Bildpunkten. Damit kommt sie der Hercules-Karte sehr nahe.

Ein besonderer Leckerbissen und in Sachen Grafik der letzte Stand der Dinge ist die – leider etwas teure – VGA-Karte, die bei der sehr hohen Grafikaufklärung von 640x280 Bildpunkten ein Farbangebot von 262144 Farben bietet.

Es gibt auch noch weitere Karten, die zum Teil sogar bessere Auflösungen und ein größeres Farbspektrum aufweisen. Sie sind jedoch nicht sehr verbreitet und auch recht teuer.

Abhängig von der verwendeten Grafikkarte ist der Monitor, für den Sie sich entscheiden. Mit einem Schwarzweißmonitor können Sie logischerweise keine Farbgrafiken produzieren, und ein CGA-Monitor ist nicht die beste Wahl, wenn Sie sich der guten Auflösung wegen für eine Hercules-Karte entschieden haben.

Die Grafikkarte ist ein wichtiger Punkt beim Kauf Ihres Rechners

In der Regel können Sie davon ausgehen, daß der Preis für den Monitor mit der Qualität der Grafikkarte steigt. Oft werden auch Rechner mit Farbgrafikkarten angeboten, bei denen aber nur ein Monochrom-Monitor mitgeliefert wird. Möglicherweise müssen Sie dann das vermeintlich gesparte Geld für einen sehr teuren Monitor wieder anlegen.

Um festzustellen, ob der Preis eines Rechners tatsächlich gerechtfertigt ist, müssen Sie all diese Kriterien ins Auge fassen. Nur dann können Sie sinnvolle Preisvergleiche anstellen, die unter Umständen recht viel Zeit in Anspruch nehmen. Doch dieser Zeitaufwand ist sicher gerechtfertigt, denn wer möchte schon viel Geld für wenig Qualität ausgeben?

(Benjamin Kleinke/wn)

MByte:	Megabyte; bezeichnet den Speicherplatz des Arbeitsspeichers oder der Festplatte beziehungsweise der Diskette
MHz:	Megahertz; beschreibt die Taktzyklen pro Sekunde (1 MHz = 1 Million Taktzyklen pro Sekunde)
Wait:	Waitstates; die Anzahl der Taktzyklen, die der Rechner warten muß, bis ihm eine Information aus dem Arbeitsspeicher zur Verfügung steht. Davon hängt unter anderem die Geschwindigkeit des Rechners ab.
BW:	Black and White; Monochrom-Monitor mit Monochromadapter oder Hercules-Karte
CGA:	Colour Graphics Adapter; Grafikkarte, die den CGA-Modus unterstützt
EGA:	Enhanced Graphics Adapter; Grafikkarte, die den EGA-Modus unterstützt oder kompatibel ist
VGA:	Video Graphics Array; Grafikkarte, die den VGA-Modus unterstützt
Slots:	Steckplätze für Erweiterungskarten; teilweise gibt der Hersteller die gesamte Anzahl aller Steckplätze an, teils die Anzahl der freien Steckplätze. Hier ist also Vorsicht geboten!
Parallel:	Parallele Centronics-Schnittstelle; Schnittstelle für den Drucker
Seriell:	Serielle RS232-Schnittstelle, Schnittstelle für Maus, Modem, Trackball etc.
MF2-Tastatur:	Multifunktionstastatur; deutsche Tastatur mit 102 Tasten und abgehobenem Cursor-Block. Manchmal wird auch die Bezeichnung »Tastatur 102« verwendet.
286er:	Computer (AT) mit 80286-Prozessor
386er:	Computer (AT) mit 80386-Prozessor
3,5“:	Diskettenlaufwerk für 3 1/2-Zoll-Disketten
5 1/4“:	Diskettenlaufwerk für 5 1/4-Zoll-Disketten
ms:	Millisekunden; Zugriffszeit auf die Festplatte
Uhr:	batteriegepufferte Echtzeituhr
CPU:	Central Processing Unit, der Prozessor des PC

Tabelle 2 listet Abkürzungen auf, die Sie oft in Produktanzeigen finden, und erklärt, was sie bedeuten

Kleiner Einsatz, große Wirkung

Von Shareware und Public Domain

Es gibt viele Arten von Programmen. Aber vor allem zwei Arten von Software sind hervorzuheben: Die eine erkennen Sie daran, daß Sie vor deren Gebrauch eine Hürde überwinden müssen, die in der Entfernung eines Klebestreifens besteht, wodurch die Disketten erst freigegeben werden – womit Sie aber auch das Rückgaberecht verwirken. Die andere finden Sie entweder in Mailboxen, bekommen Sie von Freunden oder über eine Diskette, die – für wenig Geld erhältlich – eine Datei »read.me« enthält, die Sie mit dem freundlichen Satz auffordert: »Share it with your friends«.

Software wird nicht nur gekauft. Ihr einfacher Kopiermechanismus läßt sie umstandslos von Hand zu Hand weiterreichen. Im rechtlichen Sinne ist die Weitergabe jedoch keineswegs statthaft, aber wo kein Kläger, da kein Richter. Einige gewitzte Anwender bezeichnen die Weitergabe von nicht kopiergeschützter Software ironisch als Anlage einer »dezentralen Sicherungskopie«. Da die Hersteller zunehmend davon absehen, ihre Software vor unerlaubten Kopien durch die verschiedensten Verfahren zu schützen, verbreiten sich beliebte Programme oft lawinenartig von Diskette zu Diskette. Handbücher bieten sich jedoch nicht so offensichtlich zum Kopieren an, wodurch letztlich der Buchmarkt von dieser Art der Software-Verbreitung profitiert.

Software wird aber auch in großen Stückzahlen gekauft, zum Beispiel bei kommerziellen Anbietern, die ihre Produkte nicht mehr zurücknehmen, wenn Sie den beschrifteten Klebestreifen (»Achtung! Rücknahme bei geöffneter Plastikhülle nicht möglich«) gelöst haben. Sie erwerben damit also ein Produkt, von dem Sie nicht wissen, ob es Ihren Erwartungen entspricht. Oder Sie kaufen bei den sogenannten »PD-« oder »Shareware-Vertreibern«, die es auch hierzulande zur Genüge gibt.

Da Software dem Urheberrecht unterliegt, bestimmt der Urheber in weitem Umfang, an wen und zu welchem Preis er Nutzungsrechte überträgt. Nun gibt es für viele Autoren Gründe, den kommerziellen Verkaufsweg über Händler (Distributoren) zu meiden oder aus Marketing-Gründen zumindest vorübergehend nicht in Anspruch zu nehmen. Ein

Grund hierfür ist, daß das Produkt auf dem kommerziellen Weg weder schnell genug noch in ausreichendem Maße zu den Benutzern gelangt, ein anderer, daß die Monopolstellung von bestimmten Firmen den Autor in ungeliebte Marktzwänge einbinden würde. Zu den »Public-Domain-Programmen« zählen aber auch solche, die im Rahmen eines US-Regierungsauftrags (also beispielsweise an Forschungsinstituten) – sprich mit Steuergeldern – entwickelt wurden: »Allgemeingut« wäre der hierfür treffende deutsche Begriff.

Es gibt nicht nur kommerziell zu erwerbende Software – halten Sie ein paar Leerdisketten bereit

Neben dem kommerziellen Markt – der selbst wieder in Direkt- und Händlervertrieb gespalten ist – haben sich drei rechtlich unterschiedliche Nutzungsarten durchgesetzt, die dem Benutzer nicht abverlangen, die Katze im Sack zu kaufen.

»Public-Domain« (PD), »Freeware« und »Shareware« bilden nicht nur eine preiswerte Alternative für den Anwender, sondern sind in ihrer Leistungsfähigkeit in Teilbereichen zu einer echten Konkurrenz für kommerzielle Programme geworden. Da diese Vertriebswege neu (und bislang erst für Software-Produkte relevant) sind, verwundert es nicht, daß viele Leute sie weder genau zuordnen noch entsprechend würdigen können: Wenn der Hersteller alle Rechte an seinem Programm aufgibt, handelt es

sich um Allgemeingut (»Public Domain«). Der Begriff hat allerdings keine Rechtsverbindlichkeit, so daß Sie sich, selbst wenn ein Autor diesen Begriff für sein Produkt verwendet und Sie gegen seine etwaigen weiteren Auflagen verstoßen, trotzdem strafbar machen können.

Meist werden PD-Programme entweder von einzelnen Programmierern aus Spaß an der Freude in die Welt gesetzt, oder es sind Veröffentlichungen von Forschungsprojekten, die von öffentlichen Geldern profitierten.

»Public-Domain« besagt, daß der Autor alle Rechte an seinem Programm aufgibt

So finden Sie neben einer Vielzahl von kleinen Programmen und Utilities, die oft nur Hilfsfunktionen für andere Programme oder für das Betreiben des PC ausüben, auch moderne Sprachentwicklungen, die im Rahmen von Forschungsprojekten entstanden sind – beispielsweise Lisp-Interpreter oder Compiler für Modula-2 und Pascal oder ganze Expertensysteme (etwa in Prolog), Turbo-Pascal-Tools oder C-Libraries. Oft sind die kleinen PD-Programme Vorreiter für kommerzielle Lösungen, sei es, weil neue Ideen in ihnen stecken, oder sei es, weil sie viel schneller zirkulieren als auf den Händlerpfaden.

Eigens spezialisierte Händler (siehe Textkasten) bringen Public-Domain-Programme in den Verkehr, oder sie werden von Interessenten auf dem elektronischen Postweg (über Zusatzgeräte wie Modems, Akustikkoppler) von »Mailboxen« (elektronischen Briefkästen) abgeholt. Die Händler bieten die Software-Entwicklungen keineswegs kostenlos dem interessierten PC-Anwender an, sondern berechnen zumindest den Grundwert der Diskette, eine Kopiergebühr sowie diverse Auslagen des PD-Anbieters – und natürlich die Versandkosten.

Beim autorisierten Händler der PC-SIG-Reihe beispielsweise kommen noch die Preise für Anzeigen und der Preis für die Originaldiskette, die stets aus den USA bezogen wird, hinzu. Für 16 Mark kaufen Sie dann nicht nur eine Programmdiskette, sondern quasi auch Informationen in PC-Fachzeitschriften.

Eine auf diese Weise in Umlauf gebrachte »Freeware« ist auch als »Guiltyware« (»guilty«, englisch für »schuldig«) bekannt, da sich schließlich jeder, der gegen diese finanziel-

len Abmachungen verstößt, wenigstens Schuldgefühle bekommen sollte. Besser wäre der Begriff »non-commercial domain«, da der Freeware-Begriff inzwischen von einem kommerziellen Programmautor okkupiert wurde.

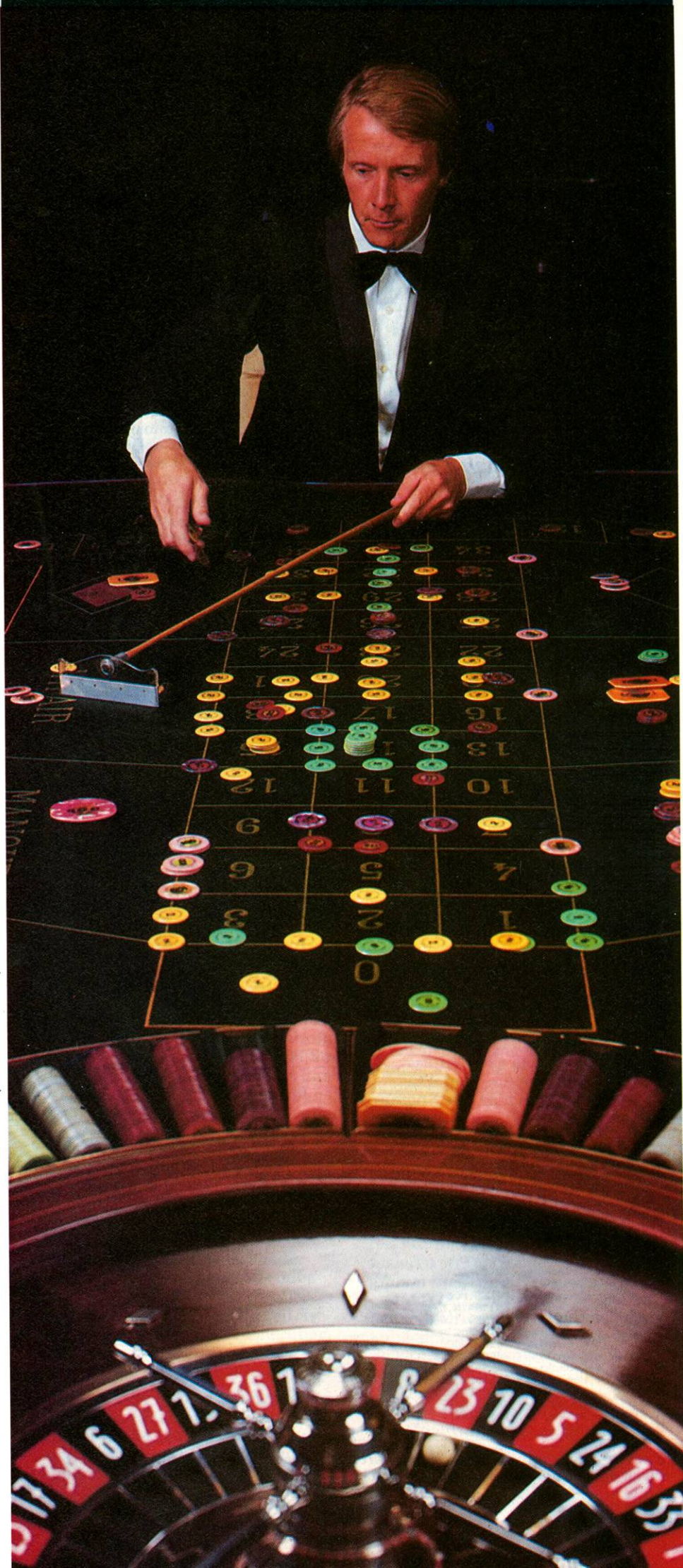
Wenn die Urheber die Nutzung nur unter bestimmten Auflagen freigeben, handelt es sich um »Guiltyware«

Der Anziehungskraft des freien Tauschs oder einseitiger Weitergabe können sich auch andere Anbieter nicht entziehen, die aber stark an der finanziellen Seite interessiert sind: die Shareware-Anbieter. Zwar »betiteln« sie auch, aber nicht um einen Obulus für einen guten Zweck, sondern für ihr Auskommen, für die Verbesserung des Produkts und um eine faire Gegenleistung für ihre geistige Arbeit. Und das tun sie nicht ungewitzt – im kaufmännischen Sinne. Shareware ist zum »Teilen« da – und dies hat mit Redlichkeit zu tun. Das Kopieren von Shareware ist gestattet, aber wenn Sie mit solchen Programmen arbeiten, ist die Zahlung einer nicht allzu hohen, einmaligen Gebühr fällig. Das »Prüfen vor dem Kauf«, etwa in der Autobranche gang und gäbe, wurde erst mit Shareware in die Software-Branche eingeführt. Doch das Prüfen kann zum Dauertest werden, ohne daß der Hersteller eine müde Mark sieht, da es für den Autor einen indiskutablen Aufwand bedeuten würde, seine Geldforderungen einzutreiben.

»Shareware« heißen Programme, die Sie durch Zahlung einer Registrierungsgebühr benutzen dürfen

Deshalb haben viele Shareware-Autoren für das Registrieren zusätzliche Anreize geschaffen – etwa ein ausführliches (gedrucktes) Handbuch anstatt der spärlichen Readme-Datei, Telefonunterstützung, kostenlose Neuversionen und bei Versionen, die noch nicht alle Eigenschaften der Shareware besitzen, natürlich die Vollversion. Zusätzlich wird durch die Registrierung natürlich ein Anreiz geschaffen, daß der Autor sein Produkt noch verbessert, wovon dann der Anwender wiederum den Nutzen hat.

Nun nagen in den USA die Shareware-Autoren keineswegs am Hun-



gertuch. Zum einen ist dort das finanzielle Klima dafür wohl fairer als hierzulande, zum anderen sind die Programme ja auch meist in Englisch gehalten, und zum dritten ist es im Lande bedeutend leichter, dem Autoren Geld zukommen zu lassen, als über den großen Teich. Wenn der Zahlungswillige die Kontonummer und Bankadresse des Autors nicht kennt, empfiehlt sich als Zahlungsweise die Postüberweisung, wobei der Betrag um die Kosten für einen schnellen Luftpostversand.

Zum Schlaraffenland der Shareware-Szene fehlt Ihnen meist die Produktinformation

Zwar werden die Programme über »Interessengruppen« vertrieben – dazu zählt beispielsweise die PC-SIG, die »Personal Computer Software Interest Group«, die auch in Deutschland zu den bekanntesten Kanälen gehört – doch bei der Registrierung geht jeder Hersteller seine eigenen Wege, sowohl was Preis und Zahlungsort als auch Leistung anbelangt. Andere »Usergroups« oder »Libraries« sind die »Public Software Library« und die »Best Bit and Bytes«. Alle drei bieten ihren Kunden mehr als tausend Disketten an – leider handelt es sich hierbei ausschließlich um englische Software.

Über die Nutzungsgebühren versuchen die Autoren, zumindest einen Teil ihrer langwierigen Entwicklungsarbeit entlohnt zu bekommen, denn insbesondere unter diesen Programmen befinden sich häufig sehr hochwertige und leistungsstarke Programme. Namen wie »PC-Write«, »PC-File«, »Automenu« oder »PC-Outline« haben sich hier ihre Sporen verdient.

```
Volume in Laufwerk A ist #1189 v1
Verzeichnis von A:\

ALT      ARC      311287   22.05.88   16.13
ARKE     COM      11402    15.12.86
READ     ME       19047    15.05.88   18.00
INSTALL  BAT      3923    18.05.88   12.30
FILE1189 TXT     1657    24.10.88   16.33
GO       BAT      38      18.01.88   13.38
GO       TXT      617     21.10.88   16.07
7 Datei(en)      11264 Bytes frei
```

Bild 1. So sieht das Verzeichnis der Diskette für das Programm »Alt« aus: nur drei ausführbare Programmdateien (»go.bat«, »install.bat« und »arke.com«) – keine Spur von »alt.exe«

Nun wurde für Shareware-Programme nicht immer diese Verkaufsstrategie beibehalten. Teilweise wurde die Shareware nur noch in Ländern vertrieben, in denen genug Registrierungen ergingen, teilweise wurde der traditionelle Verkaufsweg beschritten. So kursiert beispielsweise PC-Write in Deutschland nur bis zur Version 2.4 (beziehungsweise 2.71 bei ausgesuchten Vertreibern) als Shareware, während die Version 3.0 nur als Vollprodukt (mit Handbuch) für 128 Mark zu haben ist, allerdings auch in Deutsch. Sie müßten in diesem Falle die neuen Shareware-Versionen aus den USA bestellen.

Prüfung vor dem Kauf ist nicht immer möglich: »Crippleware« bietet manchmal nur eine unvollständige Kostprobe

In Deutschland haben sich nicht nur autorisierte Händler des Shareware- und PD-Marktes bemächtigt, die denn auch mit großen Produktanzeigen für »ihre« Programme, für Neuheiten und Highlights werben. Doch

mischen auch viele Kleinhändler mit, die die Disketten gegen eine geringe Gebühr kopieren und zum Teil auch »unleserliche« Kataloge herausgeben. Diese Händler haben aber selbst nur die Disketten der »Großen« kopiert und leisten meist als Briefkastenfirma weder Support, noch greifen sie sonstwie kreativ ein. Unerfreulich ist dabei, daß durch diesen oft rein kommerziell orientierten Zwischenhandel die Shareware-Idee beeinträchtigt werden kann: »Wenn Sie Anwenderunterstützung vom Autor wünschen, bezahlen Sie ihm eine geringe Registrierungsgebühr« heißt es da in einer Anzeige. Hauptsache ist, es werden Kunden für die Gratis-Software »geangelt« – Vermittlungsgebühr 14,40 Mark pro Diskette.

Aus Angst, ihre Produkte könnten zahlungsunwillige Abnehmer finden, haben Shareware-Autoren ihre Produkte oft als »Crippleware« konzipiert. Diese Programmversionen sind in Teilbereichen nicht funktionsfähig oder nicht vollständig leistungsbereit. Selbstverständlich sind diese »Scherze« bei einer Vollversion ausgegült.

Aber worauf Sie bei der Bedienung gefaßt sein sollten: Nicht jede Shareware, besonders wenn Sie direkt von »drüben« bezogen wurde, kennt die deutsche Tastatur und die Umlaute. So scheiden die Programme »Nyword« und »Chewriter« als Textverarbeitungsprogramme hierzulande aus. Tastenbelegung hilft auch nicht immer – besonders bei Datenbanken nicht, wenn die Umlaute im erweiterten Zeichensatz bei den Grafikzeichen landen, wonach bekanntlich nicht sortiert wird.

Nicht nur kommerzielle Produkte haben Fehler, auch in der Shareware-Szene wimmelt es davon. Aber das Konzept der »User-Supported-Software« (anwenderunterstützte Programmierung) ist nicht nur dazu da, dem Autor seinen Lebensunterhalt zu sichern, sondern diesem auch auf direktem Weg Fehler mitzuteilen, so daß sie in einer neuen Version vermieden werden können.

Sogar von »Viren« wird ab und zu in der Shareware-Szene gemunkelt: Das sind Programmteile, die Ihre Original-Software angreifen können – etwa Programmabstürze herbeiführen, auf der Festplatte für Unordnung sorgen oder aber nur ein Spaßchen im Sinn haben. Die Gefahr ist bei autorisierten Händlern und Originaldisketten verschwindend gering, in den Mailboxen der US-Anwender jedoch nicht ganz von der Hand zu weisen. Selbst bei kommerziellen Produkten sollen schon Viren gewütet haben. Und auf noch etwas sollten Sie gefaßt sein, nämlich daß sowohl Programm und Handbücher in Englisch abge-

```
PKXARC FAST! Archive Extract Utility Version 3.4 12-15-86
Copyright (c) 1986 PKWARE, Inc. All Rights Reserved. PKXARC/h for help
```

```
Unarcng: ALT.ARC
unCrunching: ALT.EXE
unCrunching: ALT.HLP
unCrunching: APP.ACC
unCrunching: CALC.ACC
unCrunching: EDIT.ACC
unCrunching: FILER.ACC
unCrunching: INITMAIN.EXE
unCrunching: MAINEDIT.ACC
unCrunching: MEMUPROG.ACC
unCrunching: ORDER.ACC
unCrunching: ROLODEX.ACC
unCrunching: ROLODEX.DAT
unCrunching: ROLODEX.IND
unCrunching: ROLODEX.SET
unCrunching: SETUP.EXE
unSquashing: TUTORIAL
```

Bild 2. Der Befehl »arce alt.arc« expandiert die komprimierte Datei und präsentiert Sie nach dem »Uncrunching« als ordentliche Software

faßt sind. Deutsche Programme gibt es nicht sehr viele, und deutsche Handbücher zu den amerikanischen Programmen sind Mangelware. Sicherlich gibt es Programme, deren Verwendungszweck wohl nur dem Autor selber bekannt sein dürfte, aber die Mehrzahl der Produkte besteht aus qualitativ gehaltvollen Programmzeilen, bei denen es sich lohnt, sie nicht nur zu testen. Die Auswahl von Shareware erfolgt im allgemeinen nicht aus Katalogen, sondern aus Zeitungsinseraten einer der vielen deutschen Händler, oder Sie entnehmen das Angebot dem redaktionellen Teil einer (etwa aus der DOS International). Zwar existieren etwa bei der PC-SIG allein zwei Kataloge, in denen ungefähr 1150 Disketten aufgelistet und beschrieben sind. Wollen Sie einen Überblick über das Angebot einzelner Händler gewinnen, so erhalten Sie (oft gegen Unkosten-erstattung) prall gefüllte Hefte, in denen jedes Programm in einer halben Zeile kurz angerissen ist. Über die Qualität und den effektiven Nutzen der Software ist damit oftmals aber nicht sehr viel ausgesagt. Im Shareware-Magazin der PC-SIG dagegen erfährt der Leser oft mehr als nur ein paar informative Sätze über das Produkt.

Die Software ist zunächst noch nicht einsatzbereit: gepackt und »papierlos«

Zwar erfordert die Kopiergebühr für die Programme mit 10 bis 16 Mark pro Diskette keine unerschwinglichen finanziellen Anstrengungen, doch haben Sie erst den fünften Fehlversuch hinter sich, wissen Sie qualifizierte Beschreibungen und Vergleichstests zu schätzen.

Der deutsche Buchmarkt reagiert auf Shareware bislang relativ hilflos: Lediglich Ralph Möllers mit seinem Buch »Hilfsprogramme, Utilities, Tools« im Rowohlt Taschenbuch Verlag (14,80 Mark) und die expandierende Shareware-Reihe des Systema Verlags, München, sowie wenige Exemplare der Shareware-Bibliothek des Verlags Markt & Technik nehmen die Aneignungsversuche deutscher Shareware-Anwender ernst.

Auf dem nichtkommerziellen Weg fallen bei allen drei angesprochenen, juristisch unterschiedlich behandelten Programmarten eigentlich (neben dem Porto bei Händlerbestellungen) nur die Kopierkosten an, welche die Vertreiber geltend machen. Es ist demzufolge sinnvoll, Diskettenplatz zu sparen, um nicht bei mehreren Disketten sowohl die sich multiplizierenden Kopier- als auch die Dis-

kettenkosten in Rechnung gestellt zu bekommen. Archiv- oder Packprogramme komprimieren zu diesem Zweck alle erforderlichen Dateien, bevor diese in Umlauf gebracht werden (Bild 1). Da diese Komprimierungsroutinen sehr effektiv sind, wird das Archivprogramm mit auf der Diskette untergebracht – schließlich benötigen Sie es zum »Entarchivieren«.

Zum anderen wird die Bedienungsanleitung nicht auf einem Stapel Papier untergebracht, sondern elegant auf Diskette mitkopiert (»on-disk-documentation«), so daß der einzelne Anwender nur seinen Drucker in Betrieb setzen muß, um in seiner Bedienungsanleitung zu blättern. Natürlich steht es Ihnen meist frei, in die Anleitung auch über den Bildschirm Einsicht zu nehmen.

Beide Verfahrensweisen machen es möglich, daß fast jegliche angebotene Software vollständig auf ein oder zwei Disketten zirkulieren kann. Um ein dem kommerziellen Vertriebsweg adäquates Produkt in Händen zu halten, müssen Sie also nur die Dateien auf Ihrer Festplatte »entpacken«, was mit dem mitgelieferten Programm vonstatten geht. Die Bedienung dieses Entpackungsprogramms wird meist in einer Datei namens »go.txt« oder »read.me« beschrieben. Angenommen, Ihr Entpacker heißt »arce«, Ihre Festplatte »C« und das anzulegende Verzeichnis »pd«, dann steht dort etwa folgendes:

```
C:
md \pd
cd \pd
rem Diskette in Laufwerk A einlegen
copy A:*. *
arce *.arc
```

Die Installation auf der Festplatte übernimmt eventuell schon ein bereits so benanntes Programm.

Profi-Software zum Minimaltarif: wider den Kommerz, aber »Zwischenhändler« sind mitunter ganz nützlich

Die Dateien mit der Endung »arc« werden daraufhin »unsquashed«, »uncrunched«, »unsqueezed« oder »unarc« (Bild 2). Bevor Sie allerdings diese Aktion durchführen, ist es empfehlenswert, eine Sicherungskopie anzufertigen mit

```
diskcopy A: B:
```

und mit dieser Kopie dann weiterarbeiten. Als zweites ist Ihr Drucker an der Reihe, der das Handbuch – oft vom Autor schon entsprechend formatiert – zu Papier bringt.

Eine Auswahl an Händleradressen

■ Kirschbaum Software, 8091 Emmering (offizieller deutscher Distributor der PC-SIG; vertreibt neben den amerikanischen Originaldisketten auch eigene oder PC-SIG-Vollprodukte wie Kirschbaum-Link, Multi-Screen, PC-Write, PC-File, PC-Calc Plus etc.)

■ Computer Solutions, 8000 München 22 (mit eigens zusammengestelltem Sortiment an englischer und deutscher PD und Shareware; dient auch mit Vollprodukten wie PC-Vault, Maxi-Disk etc.)

■ Redysoft, 8150 Holzkirchen (verreibt die gepflegte Reihe »The Public Library«; gibt auch einige deutsche Public-Domain- und Shareware-Anwenderhandbücher heraus)

■ PD-Service Lage, Bernd Schulz, 4937 Lage/Lippe (reichhaltiges Sortiment – unter anderem mit deutschen Exklusiv-Programmen und aktuellen Erzeugnissen)

■ Express Service Astrid Günther, 4920 Lemgo

■ Deutsche Software-Bibliothek Nane Jürgensen, 8000 München 19 (unter anderem mit der Reihe »Die Monatsdiskette«, die stets interessante Neuerscheinungen beinhaltet)

■ Softmail AG, 7701 Büsingen (mit eigenem Sortiment, das auch deutsche Programme umfaßt)

In diesem Handbuch ist dann auch vermerkt, wie Sie das Programm Ihrer Systemumgebung anpassen: Typ der Bildschirmkarte, Typ des Druckers, das, was in die Dateien »config.sys« und »autoexec.bat« mit eingetragen werden muß, und so weiter.

Ob Sie nun eine Textverarbeitung, Dateiverwaltung oder Tabellenkalkulation suchen, eine Branchenlösung benötigen oder programmieren wollen, ob Sie sich eine Benutzeroberfläche für MS-DOS wünschen oder Ihren Festplatteninhalt in Menüs organisieren wollen, ob Sie ein Hilfsprogramm für Ihren Drucker brauchen oder für Ihre Unterhaltung gesorgt haben wollen, ob Datenübertragung anliegt oder ein Dienstprogramm seinen Einsatz verlangt – im Shareware- und PD-Bereich finden Sie alles. Und das in einem Ausmaß, das es an dieser Stelle verbietet, auch nur einzelne Andeutungen darüber zu machen.

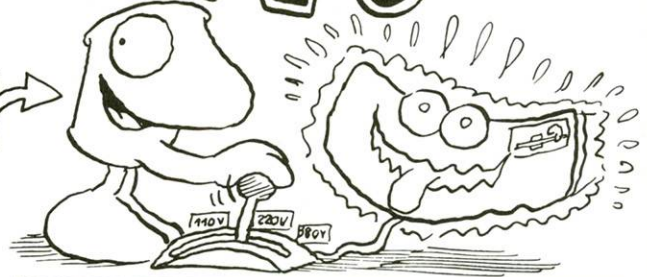
(al)

POWER ON!

**DAS PASSIERT ALLES
BEIM EINSCHALTEN
IHRES PC**

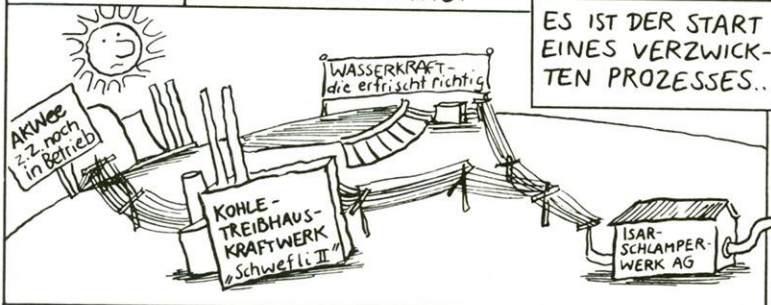
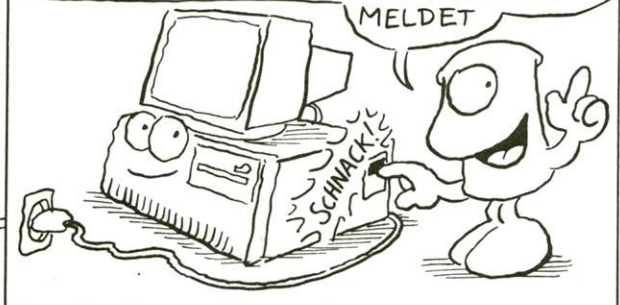
Eine Einführung von Werner Küstenmacher, mal

und
M.S.D.OSe



Wenn Sie mit dem berühmten »Big red switch« Ihren PC an das Weltenergieversorgungsnetz anhängen, geschieht danach weit mehr als nur beim Anknipsen einer Glühbirne.

NACH DER LEKTÜRE DIESER SEITEN WERDEN SIE ES DOPPELT ZU SCHÄTZEN WISSEN, WENN SICH IHR PEE-CEE BEIM NÄCHSTEN EINSCHALTEN ZUR STELLE MELDET



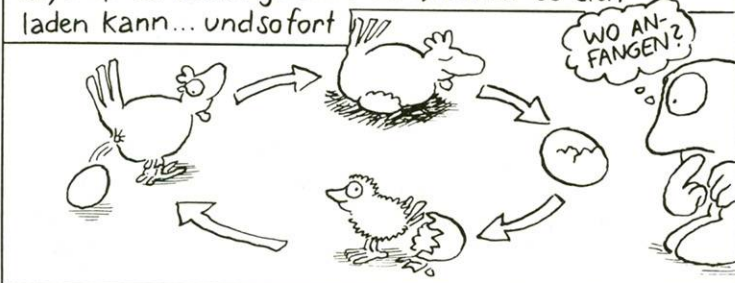
Die erste Wirkung des frischen Elektronenschwarms ist offensichtlich: Der Lüfter in Ihrem eigenen Mini-Umspannwerk (dem Netzteil) läuft an...

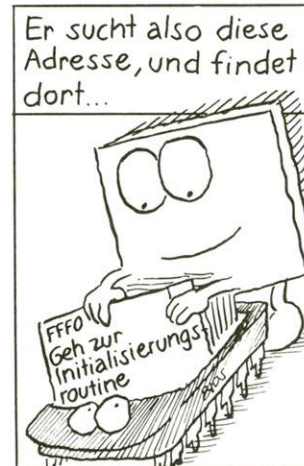
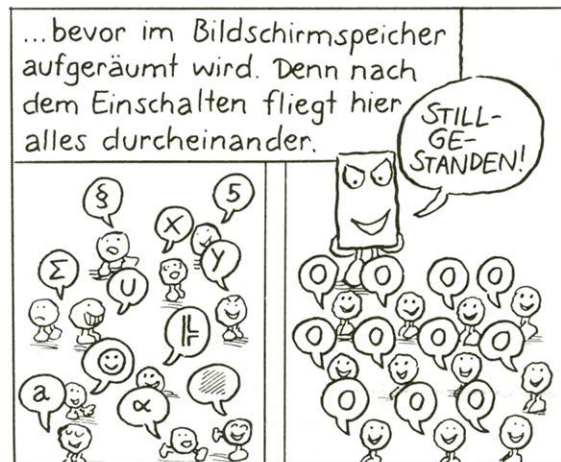
...es wird sogleich kräftig heruntertransformiert, gleichgerichtet und Niederspannung durch die schwarz-roten Kabel gejagt. Wer ohne Wenn und Aber von Anbeginn Saft bekommt, ist der Motor Ihrer Festplatte (so Ihr PC eine hat).



Nun folgt eine philosophische Denksportaufgabe: In den PC muß ein Betriebssystem: Ein Programm, das z.B. dafür zuständig ist, daß Programme geladen werden. Weil es aber selber ein Programm ist, muß es zuerst geladen sein, damit es sich laden kann... und sofort

Die Lösung ist genial: Ein Teil des lebensnotwendigen Programms ist bereits in der Fabrik fest in den PC eingebaut worden.





Was gibt's für Erweiterungs-
teile? Funktionieren sie?



Das BIOS wird gebeten,
seine Copyright-Meldung
auszugeben.



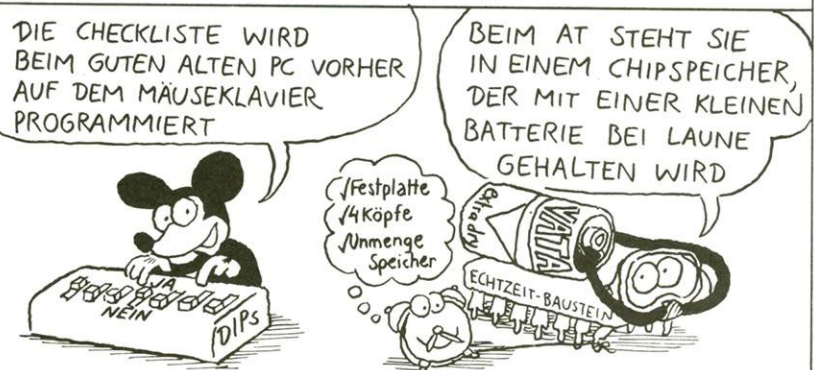
Der Nicht-Maskierbare-
Interrupt wird wieder erlaubt.



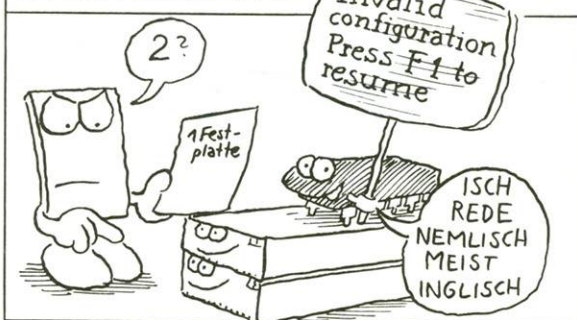
Was folgt, erfordert Zeit:
Appell für alle Speicherzellen.



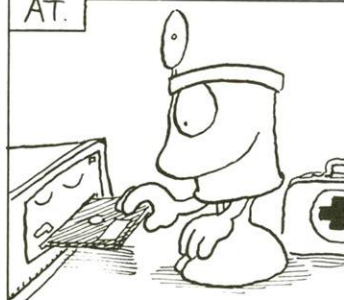
Vor dem Abheben kommt bei jeder komplizierten
Maschine der Hardware-Check.



Wenn der Inhalt der Liste und der
Inhalt Ihres PC nicht übereinstimmen,
gibt das BIOS eine Fehlermeldung
an den Bildschirm.



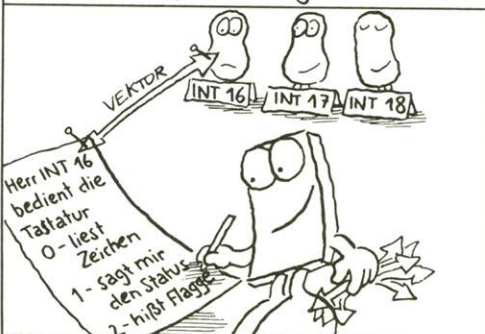
Abhilfe verschafft die
richtige Einstellung der
Dip-Schalter bzw. das
Setup-Programm beim
AT.



Wenn alles stimmt, sucht
das BIOS alle seriellen und
parallelen Schnittstellen
und gibt ihnen Namen.



Nun wird die Adressenliste des
Dienstpersonals ins erste KByte
des Arbeitsspeichers geschrieben.



Bisher ist Ihr Computer noch ein recht unbeschriebenes Blatt:
Er ist noch keine DOS-Maschine. Das wird sich gleich ändern.
Denn jetzt wird Laufwerk A: angeworfen.



Falls das nicht klappt, wird dasselbe auf der Festplatte gesucht.



Damit ist der Dienst des eingebauten Initialisierungsprogramms um.



DOS findet auf der Diskette oder Platte zwei Dateien:



Und, wenn eine da ist, liest DOS die Datei CONFIG.SYS



Dort erfährt es, in welcher Sprache es arbeiten soll...



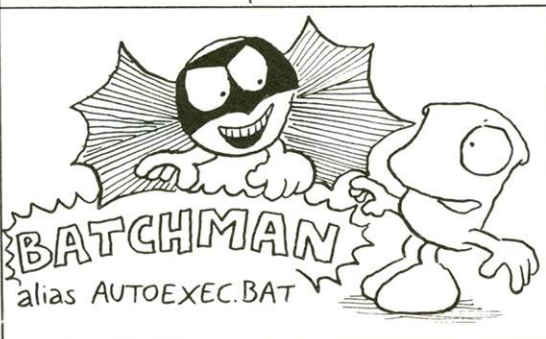
... und ob es sonstwas Besonderes gibt.



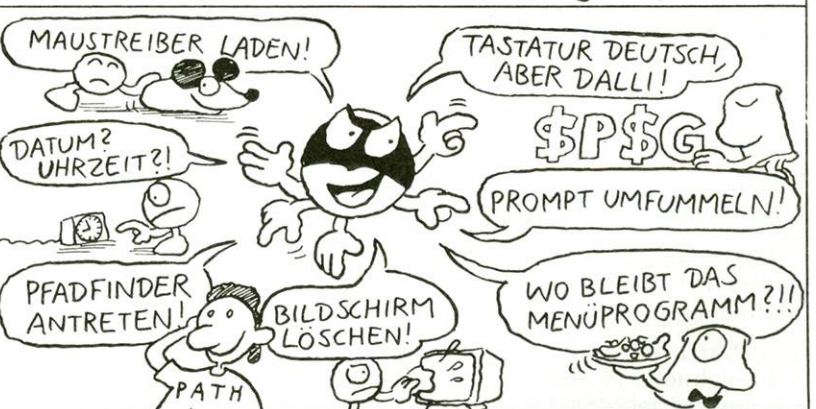
Wenn nicht anderes vermerkt ist, lädt DOS nun sein erstes Programm.



Letzte Station beim fröhlichen Dateisuchen ist die automatische Stapeldatei...



... die meist gleich für allerlei Aufregung sorgt.



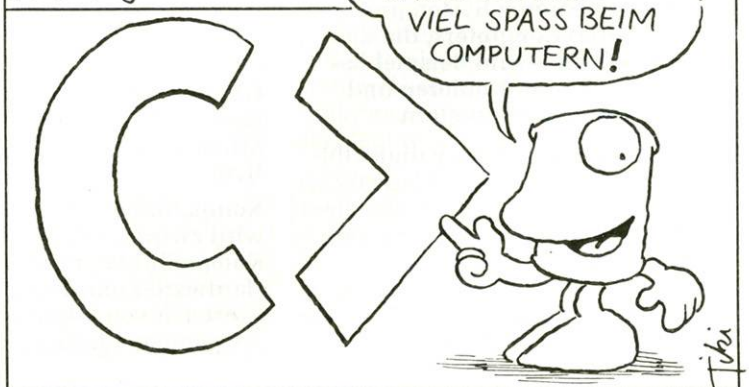
Noch eine Mitteilung in eigener Sache:



Zum Zeichen, daß DOS nun auf Ihre Eingaben wartet, meldet es sich...



... mit dem berühmt-berüchtigten PROMPT:



Einsteiger-Lexikon

Computer:

Legalisierter Geldspielautomat, der zwei Monate nach der Anschaffung entweder um die Hälfte billiger wird oder technisch überholt ist.

Akustikkoppler: Ein Gerät zur Datenübertragung über das Telefonnetz durch einfaches Auflegen eines Telefonhörers, das an die serielle Schnittstelle des Computers angeschlossen wird (Bild 1).

Arbeitsspeicher: Bezeichnung für den internen Speicher eines Computers, siehe auch RAM. Der Zugriff auf die Daten aus dem Arbeitsspeicher erfolgt schneller als von der Diskette oder Festplatte. Die benötigte Speichergröße hängt von der verwendeten Software ab. In der Regel werden mindestens 512 KByte (Kilobyte) benötigt. Wird der Computer ausgeschaltet, geht der Inhalt des Arbeitsspeichers verloren.

ASCII: Abkürzung für »American Standard Code for Information Interchange«. International gebräuchlicher Standardcode für die Zeichendarstellung. Weil Texte im ASCII-Format keine Formatierungsanweisungen und keine Textattribute (beispielsweise fett oder unterstrichen) enthalten, können sie von vielen verschiedenen Programmen gelesen und dort formatiert werden.

AT: Abkürzung für »Advanced Technology«. Mit AT werden die Personalcomputer bezeichnet, die mit dem 80286-Prozessor oder auch dem 80386-Prozessor und einer höheren Taktfrequenz arbeiten.

Batch-Datei: Wird auch als Stapeldatei bezeichnet und stellt eine Sammlung von Betriebssystembefehlen dar, die »am Stück« abgearbeitet werden (Stapelverarbeitung).

Betriebssystem: Das Betriebssystem besteht aus mehreren Systemprogrammen eines Computers, die die Grundfunktionen (zum Beispiel Laden und Speichern, Kopieren und Löschen von Dateien) steuern.

Bit: Abkürzung für »binary digit« (binäre Ziffer). Ein Bit ist die kleinste Darstellungseinheit, die einzige, die der Computer verarbeiten kann. Der Wert eines Bits kann nur 0 oder 1 sein. Diese Zustände werden durch »Strom an« oder »Strom aus« im Computer übertragen.

Byte: 8 Bit ergeben zusammen 1 Byte. Mit einem Byte lassen sich insgesamt 256 verschiedene Bit-Kombinationen darstellen.

Coprozessor: Der sogenannte Arithmetikprozessor unterstützt die Zentraleinheit bei arithmetischen Operationen und steigert dadurch die Arbeitsgeschwindigkeit von Programmen durch Übernahme bestimmter Funktionen, auf die der Coprozessor spezialisiert ist. Der Coprozessor wird nicht von allen Programmen unterstützt.

Cursor: Wird auch Schreibmarke oder Positionsanzeiger genannt und markiert durch einen Pfeil oder ein kleines Rechteck die augenblickliche Position auf dem Bildschirm, an der die nächste Eingabe erfolgt. Der Cursor läßt sich in Anwendungsprogrammen oft mit den Cursor-Tasten (auch Pfeiltasten genannt) nach rechts, links, oben und unten bewegen, um die Stelle zu erreichen, an der weitergeschrieben oder korrigiert werden soll.

Echtzeituhr: In einigen Computern ist eine Uhr integriert, die jederzeit das aktuelle Datum und die Uhrzeit angibt. Die Uhr wird bei abgeschaltetem Computer über Batterien versorgt.

EMS: Abkürzung für »Expanded Memory Specification«. EMS bezeichnet den erweiterten Speicher bei PCs. Auf diesen Speicher, der oberhalb der 640-KByte-Grenze liegt, kann man nur mit Hilfe bestimmter Programme zugreifen.

GByte: Einheit für Speicherkapazität. Ein GByte (Gigabyte) entspricht 1024 MByte beziehungsweise 1073741824 Byte.

Kompatibilität: Im Computerbereich wird zwischen drei verschiedenen Kompatibilitätsstufen unterschieden. Hardware-Kompatibilität ist die Eigenschaft von Computern oder Zusatzgeräten, gleiche Funktionen aus-

führen und die gleichen Produkte nutzen zu können wie das »Vorbild«. Software-Kompatibilität bedeutet, daß Software-Produkte von »Vorbildcomputern« (ohne Einschränkungen) auf dem kompatiblen Gerät arbeiten. Datenkompatibilität bedeutet, daß Daten zwischen zwei unterschiedlichen Produkten ausgetauscht und weiterverwendet werden können.

Laptop: Ein netzunabhängiger transportabler Computer in Form einer Reiseschreibmaschine.

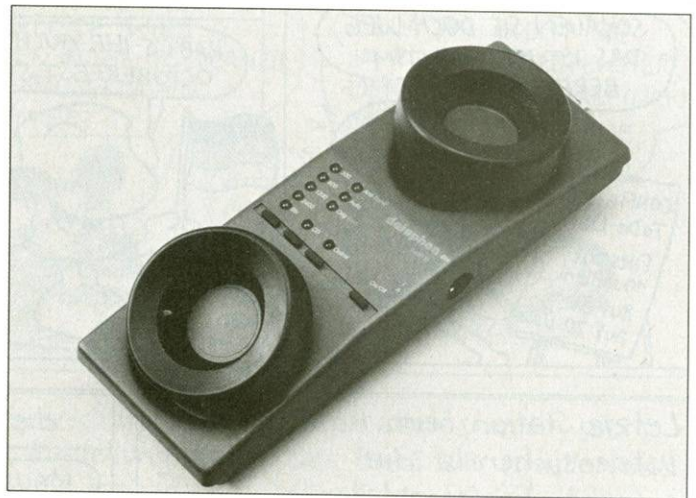


Bild 1. Akustikkoppler: Die Muffen zur Aufnahme des Telefonhörers sind deutlich zu erkennen

Maus: Aufgrund der äußeren Form und der Farbe als Maus (Bild 2) bezeichnetes Eingabegerät zur Steuerung des Cursors (Mauszeiger) auf dem Bildschirm. Die Maus ist mit einer Rollkugel ausgestattet, mit der auf einer glatten Oberfläche (zum Beispiel einer Tischfläche) die Bewegungen des Cursors auf den Bildschirm übertragen werden.

MByte: Einheit für Speicherkapazität. Ein MByte (Megabyte) entspricht 1024 KByte beziehungsweise 1048576 Byte.

MIPS: Abkürzung für »Million Instructions per Second« (Millionen Instruktionen pro Sekunde). Einheit für die Verarbeitungsgeschwindigkeit von Computern.

Modem: Zusammengesetzter Begriff aus **Modulator** und **Demodulator**. Ein Modem wandelt Daten vom Computer in Töne um, die dann beispielsweise über das Telefonnetz übertragen werden können. Die Übertra-

gungsgeschwindigkeit der Daten wird in Baud (Bit pro Sekunde) angegeben.

Netzwerk: Ein System, bestehend aus Netzwerk-Software, -Karten und -Kabeln, das mehrere Computer miteinander verbindet, so daß ein schnellerer Datenaustausch gewährleistet ist.

No name: Bezeichnung für PCs, die meist direkt aus Taiwan oder anderen Ländern mit niedrigen Fertigungskosten importiert werden und in Deutschland ohne bekannte Markennamen auf den Markt kommen.

Parallele Schnittstelle: Computerein- und -ausgang, der zur parallelen Datenübertragung, zum Beispiel bei der Ansteuerung eines Druckers, genutzt wird. Hier werden im Gegensatz zur seriellen Schnittstelle immer jeweils 8 Bit (ein Byte) gleichzeitig übertragen. Die parallele Schnittstelle wird auch als Centronics-Schnittstelle bezeichnet, weil sie vom Druckerhersteller Centronics definiert wurde.

PC: Kurzbezeichnung für einen Personalcomputer. Mittlerweile werden IBM-PC-kompatible Personalcomputer allgemein als PCs bezeichnet. Aber auch die leistungsfähigeren XTs oder ATs werden inzwischen unter diesem Begriff geführt.

Portable: Oberbegriff für alle tragbaren Computer.

Prozessor: Wird auch Zentraleinheit oder CPU (Central Processing Unit) genannt und bezeichnet den Teil des Computers, in dem die Datenverarbeitung erfolgt. In PCs werden 8086- oder 8088-Prozessoren eingesetzt, in ATs die schnelleren 80286- beziehungsweise 80386-Prozessoren.

PS/2-Modell: Das PS/2-Modell (Personal System/2) wurde Mitte 1987 von IBM als Nachfolger der PC-Reihe vorgestellt. Das Besondere an diesem Rechner ist die Mikrokanalarchitektur. Der Mikrokanal (Microchannel) erhöht die Datenübertragungsrate.

RAM: Abkürzung für »Random Access Memory« (Schreib-Lese-Speicher oder Speicher mit wahlfreiem Zugriff), siehe Arbeitsspeicher.

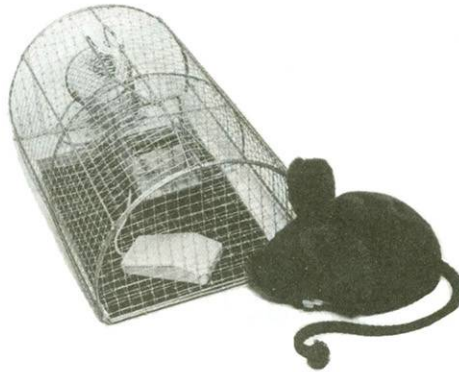


Bild 2. Auf den zweiten Blick haben Maus und Maus nur den Namen gemein

ROM: Abkürzung für »Read-Only Memory« (Nur-Lese-Speicher oder Festwertspeicher). Im Gegensatz zum RAM- ist der Inhalt des ROM-Speichers unveränderbar und bleibt auch nach dem Ausschalten des Computers erhalten. Auf den ROM-Speicher kann nur lesend zugegriffen werden. Beim PC sind im ROM wichtige Daten abgelegt, die für den Start und Betrieb unbedingt notwendig sind und sich nicht verändern, sondern sich nur lesen lassen.

Serielle Schnittstelle: Computerein- und -ausgang, der zur seriellen, das heißt bitweisen Übertragung von Daten, etwa zur Datenfernübertragung oder Mausbedienung, genutzt wird. Genormte serielle Schnittstellen sind beispielsweise RS232 und V.24.

Speicherkarten: Steckkarten, die den Arbeitsspeicher des Computers vergrößern und dadurch das Anwendungsspektrum erweitern.

Steckplatz: Ein freier Einschub, auch »Slot« genannt, der sich auf der Mutterplatte des Computers befindet und in dem eine Erweiterungskarte Platz findet. Ein Computer kann über freie Steckplätze mit verschiedenen Steckkarten (beispielsweise mit einer Grafikkarte) aufgerüstet werden.

Taktfrequenz: Die Taktfrequenz ist ausschlaggebend für die Arbeitsgeschwindigkeit, mit der der Rechner die Grundbefehle ausführt, und gilt als Maß für die interne Rechengeschwindigkeit eines Computers. Die Geschwindigkeit wird in Megahertz (MHz) angegeben. PCs erreichen Werte zwischen 4,77 und 10 MHz, ATs Werte von 10 bis 20 MHz und 80386-Rechner Werte zwischen 16 und 33 MHz.

Tastatur: Die Tastatur eines Computers, auch Keyboard genannt, dient zur Eingabe von Befehlen und Daten in den Computer.

XT: Abkürzung für »Extended Technology«. Die Bezeichnung XT wurde 1983 von IBM für Personalcomputer eingeführt, die im Gegensatz zu den IBM-PCs eine 10-MByte-Festplatte besaßen.

Massenspeicher:

Ein Datenträger dient dazu, wichtige Informationen bis zu dem Zeitpunkt zu speichern, an dem sie dringend benötigt werden, um sie dann dem Anwender durch einen Lesefehler vorzuenthalten.

Bandlaufwerk: Wird auch »Streamer« genannt und bezeichnet ein magnetisches Speichergerät, das im PC-Bereich in erster Linie zum Backup (das heißt, zum Archivieren von Festplatteninhalten) eingesetzt wird.

Diskette: Magnetisches Speichermedium zur Lagerung (Speicherung) von Programmen und Daten. Die Informationen werden auf einer Magnetschicht (etwa wie beim Kassettenrecorder) abgelegt und können dort vom Diskettenlaufwerk jederzeit

wieder gelesen, verändert oder gelöscht werden. Die gebräuchlichen Diskettenformate im PC-Bereich sind 5 1/4 Zoll und 3 1/2 Zoll.

Disketten-Controller: siehe Festplatten-Controller

Diskettenlaufwerk: Schreib-/Lesegerät für magnetische Speicherplatten, von denen zum Beispiel Programme an den Computer übertragen werden.

FAT: Abkürzung für »File Allocation Table« (Dateizuordnungstabelle). Tabelle der Positionen auf dem Speichermedium, an denen der Schreib-/Lesekopf die gesuchten Daten findet.

Festplatte: Magnetisches Speichermedium, ähnlich der Diskette (Bild 3). Die Festplatte zeichnet sich durch größere Sicherheit und eine höhere Speicherdichte aus, die die Speicherung größerer Datenmengen zuläßt.

Die Festplatte ist im Gegensatz zur Diskette fest im Computer eingebaut und erreicht höhere Datenübertragungsgeschwindigkeiten.

Festplatten-Controller: Steuereinheit, die die Datenübertragung vom Computer zur Festplatte und umgekehrt und die Datenspeicherung auf der Festplatte verwaltet.

Formatieren: Das »Einrichten« einer Diskette, also das Aufteilen der Diskette in bestimmte »Spuren« und »Sektoren«, in denen später die Daten abgelegt werden.

Headcrash: Aufsetzen des Schreib-/Lesekopfes auf der Festplattenoberfläche, das in aller Regel schwerste Beschädigungen und Datenverluste zur Folge hat. Kann durch »Parken« des Kopfes weitgehend vermieden werden (siehe Parken).

High-Level-Formatierung: Bei der High-Level-Formatierung werden das Hauptinhaltsverzeichnis und »FATs« angelegt.

Inhaltsverzeichnis: Wird auch »Directory« genannt und bezeichnet eine Liste der sich auf dem Speichermedium befindenden Dateien. Das Inhaltsverzeichnis wird automatisch erzeugt und verwaltet.

Low-Level-Formatierung: Das Aufteilen einer Festplatte in Spuren und Sektoren.

Parken des Schreib-/Lesekopfes: Beim Parken wird der Schreib-/Lesekopf auf einer unbenutzten Spur positioniert und verhindert so beim Aufsetzen eines Schreib-/Lesekopfes auf der Festplatte den Verlust von Daten.

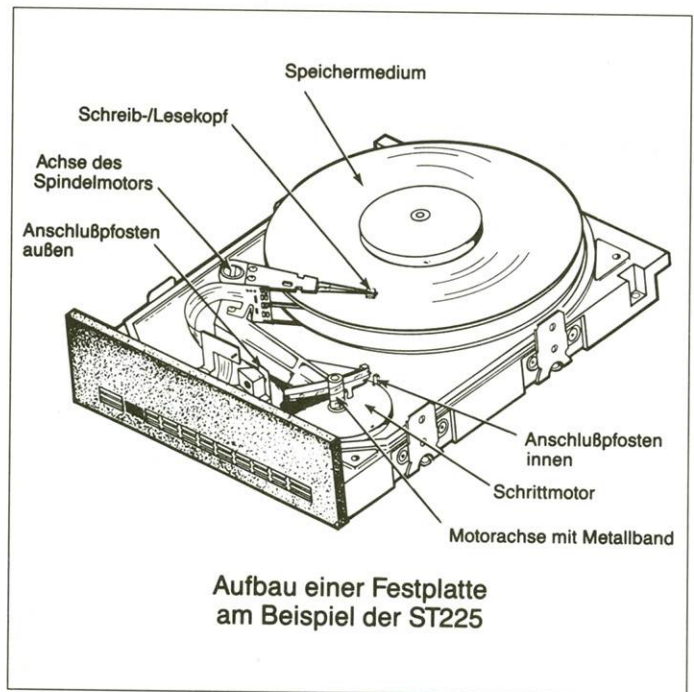
Parken: Den Schreib-/Lesekopf des Festplattenlaufwerks an eine Position fahren, an der die Daten vor Beschädigung geschützt sind.

Partitionieren: Festplatte in logische »Unterlaufwerke« einteilen, die vom Betriebssystem als eigenständige Laufwerke behandelt werden.

RAM-Disk: Virtuelles (scheinbar vorhandenes) Laufwerk im Arbeitsspeicher, das vom Betriebssystem wie ein eigenständiges externes Laufwerk behandelt wird. Die Daten einer RAM-Disk gehen beim Ausschalten des Computers verloren.

Schreib-/Lesekopf: Vergleichbar mit dem Tonkopf eines Kassettenrecorders. Schreibt und liest die Daten auf der Magnetschicht des Speicherme-

Bild 3. Schematische Grafik des Aufbaus einer Festplatte



diums und überträgt sie an den Computer.

Schreibschutzöffnung: Kerbe in der Diskettenhülle, die über eine Lichtschranke abgefragt wird. Ist die Öffnung vorhanden, kann die Diskette normal gelesen und beschrieben werden, ist die Öffnung nicht vorhanden oder lichtundurchlässig verschlossen, ist ein Beschreiben der Diskette nicht möglich.

Sektoren: Durch das Formatieren festgelegte Abschnitte auf der Disket-

tenoberfläche, die vom Schreib-/Lesekopf des Laufwerks angesprungen und gelesen werden können.

Spuren: Konzentrische Kreise auf der Diskettenoberfläche. Die Spuren werden weiter in Sektoren unterteilt.

Zugriffszeit: Ein Maß für die Zeit, die durchschnittlich benötigt wird, um den Schreib-/Lesekopf des Disketten- oder Festplattenlaufwerks zu positionieren, um Daten zu lesen. Die Zugriffszeiten werden in Millisekunden angegeben.

Drucker:

Ein Gerät, das dem Anwender die Arbeit erspart, Texte mit der Hand zu schreiben, so daß er die gewonnene Zeit nutzen kann, um das Druckerpapier einzufädeln.

Buffer: Pufferspeicher, der die zu druckenden Daten zwischenspeichert, so daß der Computer nicht auf das Ende des Druckvorgangs warten muß.

cpi: Abkürzung für »characters per Inch« (Zeichen pro Zoll). Einheit für die Zeichendichte.

cps: Abkürzung für »characters per second« (Zeichen pro Sekunde). Einheit für die Druckgeschwindigkeit.

Dip-Schalter: Kleine Kippschalter, die direkt auf der Platine angebracht sind und über die der Drucker auf bestimmte Druckmodi oder Zeichensätze eingestellt werden kann.

dpi: Abkürzung für »dots per inch« (Punkte pro Zoll). Einheit für die grafische Auflösung.

Draft: Schnellschrift, auch als EDV-Qualität bezeichnet. Bei geringer Schriftqualität wird eine hohe Druckgeschwindigkeit erreicht.

Druckertreiber: Spezialprogramm zur Ansteuerung bestimmter Druckertypen. Ein Anwendungsprogramm arbeitet über den Druckertreiber mit dem vorhandenen Drucker zusammen. Wird der Drucker gewechselt, ist in aller Regel auch ein Wechsel des Druckertreibers erforderlich.

Druckpuffer: Druckerinterner Speicher, der die zu druckenden Daten aufnimmt, zwischenspeichert und dem Druckwerk zuführt, um den Computer zu entlasten.

Einzelblatteinzug: Vorrichtung zur Zuführung von einzelnen Papierseiten, etwa Briefpapier oder Formulare.

Formfeed: Seitenvorschub

Laserdrucker: Das Druckbild wird bei Laserdruckern durch einen Laserstrahl auf einer Bildtrommel erzeugt. Durch die hohe Präzision des Lasers

COURIER, 12 POINT

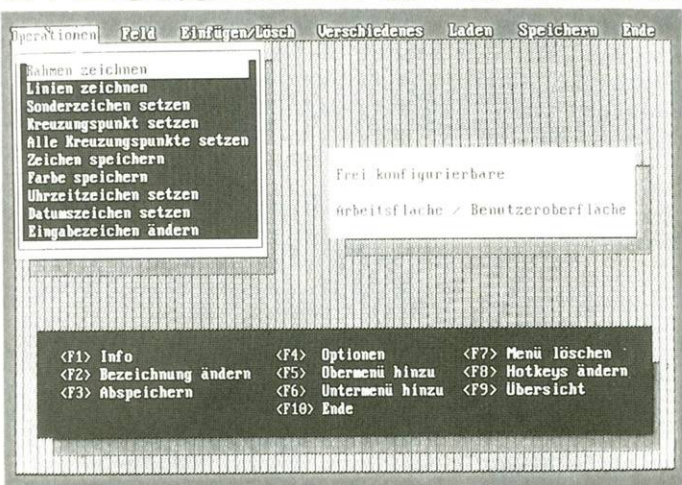
ABCDEFGHIJKLMNOPQRSTUVWXYZ
abcdefghijklmnopqrstuvwxyz
0123456789:;<=>?@{|}~
!"#\$%&'()*+,-./[\]^_`

COURIER ITALIC, 12 POINT

ABCDEFGHIJKLMNOPQRSTUVWXYZ
abcdefghijklmnopqrstuvwxyz
0123456789:;<=>?@{|}~
!"#\$%&'()*+,-./[\]^_`

Bild 4. Die Schriftfamilie Courier, normal und kursiv

MaskEdit PLUS



Version 5.0

Professioneller Maskengenerator für Turbo Pascal 3/4/5, Turbo C ab V.1.5, dBase III + Microsoft C und Quick C, JPI TopSpeed Modula-2

Mit der neuen Version von MaskEdit Plus zaubern Sie komfortable Bildschirmmasken in Sekundenschnelle.

- Jetzt alle Funktionen über Pulldown-Menüs oder Hotkeys
- Jederzeit kontextbezogene Hilfe auf Tastendruck
- Ausnutzung des gesamten Bildschirms zur Maskenerstellung
- Unterstützung von zwei Bildschirmen gleichzeitig
- Leichte Konfiguration von Hotkeys und Pulldown-Menüs
- Verbesserte oder völlig neu entwickelte Maskentreiber

Professionell:

Mit über 50 Funktionen zur Maskenerstellung können auch Teile einer Maske kopiert, gelöscht, verschoben, zentriert, gefärbt und gefüllt werden. Der komplette IBM-Zeichensatz bis 255 wird genauso wie die gleichzeitige Bearbeitung von zwei Masken und die Definition von Ein-/Ausgabefeldern unterstützt.

Mehrere Dateiausgabeformate wie ASCII, Binär, "Gepackt", dBase und ANSI (Farbige Masken auch mit dem DOS-Befehl TYPE!) sind im Lieferumfang enthalten.

Komfort:

Neben acht verschiedenen Rahmentypen sind acht oder 16 Hintergrundfarben, Schattenwurf und Cursorkoordinaten im Pascal- oder dBase-Format darstellbar. Bei Rahmen- und Linienzeichnungen werden die Kreuzungspunkte automatisch angepaßt. Eine integrierte ASCII-Tabelle und ein universeller Maustreiber machen Tempo.

Programmier-Highlights der Maskentreiber

Sämtliche Treiber werden als Sourcecode geliefert. Eigene Routinen können zur Kontrolle von Eingaben, Hilfetexten etc. problemlos in den Steuerungsablauf eingebunden werden. Integration von Datum, aktueller Uhrzeit und automatischer Typüberprüfung bei der Eingabe, auch für den zusätzlichen Datentyp DATUM. Aufruf der generierten Masken durch einen einzigen Befehl und Steuerung der Ein-/Ausgabefelder durch den Maskentreiber. Wahlweise Ausschluß von Teilmengen des Zeichensatzes und automatische Eingabe in Großbuchstaben inklusive der Umlaute. Bei Änderungen von definierten Feldern kann die Kontrolle an das eigene Programm übergeben werden. Jedes Eingabefeld kann einzeln gefärbt werden, ist voll editierbar und läßt sowohl rechts- wie auch linksbündige Ausgabe zu. Der Cursor wird innerhalb der Eingabemaske durch die Pfeil- und TAB-Tasten bewegt, die Eingabemaske kann durch Funktionstasten-Betätigung verlassen werden.

MaskEdit Plus I

Turbo Pascal 3/4/5 und dBase III+

DM 99,- *

MaskEdit Plus II

Microsoft C und Quick C,
Turbo C ab V.1.5 und dBase III+

DM 99,- *

MaskEdit Plus III

JPI TopSpeed Modula-2 und dBase III+

DM 99,- *

Paketpreis mit zwei Treibern

DM 149,- *

Unabhängig von der Anzahl der bestellten Programme berechnen wir für das Inland 4,- DM bzw. für das Ausland 6,- DM Porto und Verpackung.

Bitte benutzen Sie die Bestellkarte.



DMV-Verlag · Postfach 250 · 3440 Eschwege

Für Kenner und anspruchsvolle Anwender!
130 ausgewählte professionelle
Public Domain und Shareware-Programme
für IBM PC/XT/AT mit EGA-, CGA- und Herculeskarte

TRANSLATOR	Übersetzt Englisch-Programm-Dokumentation in Deutsch
LIGHTNING PRESS	DTP-Desktop Publishing, die Nr. 1 im DTP, Fantastic
AS EASY AS	Sehr gutes Tabellenkalkulationsprogramm, Bestseller
PROPHET LOTTO	Statistikprogramm für Lotterieziehungen
3D CHESS	Dreidimensionales Schachprogramm, auch Psion Chess genannt
PC WRITE	Textverarbeitung mit vielen Möglichkeiten, sehr gut
FREE FILE	Datenverwaltung mit vielen Möglichkeiten, sehr gut
COMPOSER	Musik komponieren, Noten schreiben und ausspielen, sehr gut
PC AKT	Erzeugt digitalisierte Aktbilder auf ihrem PC
DANCAD 3D	Dreidimensionales CAD-Zeichenprogramm mit Animation
ELECTRON	Elektronik- und Elektroentwicklung
PC KWIK	Beschleunigt ihr Diskettenlaufwerk
MAHJONGG	Chinesisches Spiel, sehr gut, mit sehr guter Grafik
EDICOMM	DOS Command Processor Manager, sehr gut
MEGAGOPOLY	Wie erreiche ich eine Million? Spiel
DATNOIDS	Ähnlich wie Robot, Spielhallenhit
NUKE NEW YORK	Zeigt einen nuklearen Angriff und die Auswirkung auf New York
DESIGNER	Zeichnungsprogramm, erstellt bewegliche Bilder
JUMP JOE	Ähnlich wie Miner, Arcadespiel
BLACK JACK	Black Jack wie im Casino
CRAPS	Las Vegas, populäres Spiel
HOROSKOP	Erstellt Ihr persönliches Horoskop
KNIFFEL	Kartenspiel
MASTERMIND	Populäres Strategiespiel
PONG PONG	Geschicklichkeitsspiel
SNAKE	Schlängenspiel
STAR TRADERS	Weltraum-Handlerspiel, Aktien kaufen im Weltraum
VORTEIL	Zeigt Vorteile des IBM-Personal-Computers
ETIKETT	Etiketten in verschiedenen Formaten erstellen und drucken
FLUGSIMULATOR	Flugsimulator
TOWERS	Towers of Atlantis, Abenteuer
PC CALC	Populäre Tabellenkalkulation
COPY ALL	Nun können Sie die meisten geschützten Programme kopieren
PC WINDOW	PC-Fensterprogramme
KALENDER	Erstellt einen Kalender für jedes Jahr mit Bildsch./Druck
PRINTUM	Schaltet Drucker in Schriftarten um
BASTOOLS	Basic-Tools
YOUR MENU	Erstellt Ihr eigenes Menü zum Ausführen
FORM MASTER	Umfangreicher Formulargenerator
DOS HELP	Alle DOS-Befehle werden gezeigt und erklärt
JUKEBOX	Eine Musiktruhe voll Platten zum Ausspielen
ALIEN	Abenteuerspiel
3D TIC TAC TOE	Dreidimensionales Tic-Tac-Toe, Strategiespiel
EGGS	Eiergrafiken
MANOR	Abenteuerspiel
CIA	Abenteuerspiel
BIORYTHM	Biorythmen
FINANCE	Finanzen u. Berechnungen aus allen Bereichen
LIFETIME	Die Lebenserwartung wird errechnet
TASKPLAN	Zeitplanungsprogramm
COVER	Diskettenhüllen mit Inhaltsverzeichnis drucken
STOCK CHARTING	Aktienverwaltung, grafische Darstellungen, Kauf-/Verkaufssignal
VCR BASE	Videokassetten verwalten und Aufkleber drucken
ADDRESS	Adressen verwalten und Aufkleber erstellen
BOOKINVENTORY	Inventurprogramm für bis zu 2000 Bücher
PC LOCK	Schützt ihre Festplatte mit Passwort
BEST UTILITIES	Eine Menge Hilfsprogramme, für jeden etwas
CATALOGING	Diskettenkatalog erstellen
FB COPY	Zur Zeit das beste Kopierprogramm, sehr schnell und komfortabel
VIRUSCHECK	Computerviren-Untersuchungsprogramm
BACKGAMMON	Backgammonspiel
SPACE INVADERS	Bekanntes Arcadespiel, sehr schön
PC FOTO	Fotoverwaltungsprogramm und Diaaufkleber drucken
PACKDISK	Macht mehr Platz auf Ihrer Festplatte
BREAKOUT	Spielhallenhit
RECONFIG	Konfiguriert Ihr System für viele Zwecke, z. B. Sidekick
COMPRESS	Datenkomprimer, spart Platz auf der Diskette
CHASM	Super Assembler-Programmiersprache

Alle oben gelisteten Programme sowie weitere 62 Hilfs-, Anwendungs- und Spielprogramme. (Da ist für jeden etwas dabei.)

Insgesamt 130 Programme
KOMPLETTPREIS DM 129,-

Großer Katalog mit weit über 1500 Disketten und zusätzlichen tollen Angeboten liegt jeder Bestellung bei. Schnellversand am Tag des Bestelleingangs.

BATH BEACH Software-Service
Speckstraße 23, 6749 Schweighofen, Telefon (0 63 42) 5 40

Coupon

Die Lieferung erfolgt nur auf 5,25"-Disketten gegen Vorkasse oder Nachnahme! Senden Sie mir bitte umgehend für DM 129,- das 130er Programmpaket Public Domain. Den Gesamtbetrag bezahle ich wie folgt:

- ☐ Scheck/bar + DM 6,- Versandkosten (Endpreis DM 135,-)
☐ per Nachnahme + DM 10,- Versandkosten (Endpreis DM 139,-)
(Ausland nur gegen Vorkasse)

Name:

Straße:

PLZ/Ort:

Telefon:

Unterschrift:

Geburtsdatum

(Bei Minderjährigen der Erziehungsberechtigte)

DOS Extra 9/90

lassen sich äußerst hohe Auflösungen und scharfe Konturen erzeugen.

Linefeed: Zeilenvorschub

LQ: Abkürzung für »Letter Quality« (Briefqualität). Im Gegensatz zur Draftschrift wird hier zugunsten der Darstellungsqualität auf Druckschwindigkeit verzichtet. Die Buchstaben werden gewissermaßen doppelt gedruckt, wobei der zweite Druckvorgang dem ersten gegenüber etwas versetzt wird, wodurch eine geschlossenere Zeichendarstellung erzielt wird, weil die Matrixpunkte ineinander übergehen.

Nadeldrucker: Drucker, der die Buchstaben und Grafiken auf dem Papier aus kleinen Punkten zusammensetzt, die durch den Anschlag von Nadeln auf ein Farbband auf das Papier übertragen werden. Die Anzahl der Nadeln beträgt bei derzeit gebräuchlichen Druckern entweder 9 oder 24. Es gibt allerdings auch schon Drucker, die 48 Nadeln benutzen.

NLQ: Abkürzung für »Near Letter Quality« (Korrespondenzqualität) – siehe LQ. Der Begriff wird immer im Zusammenhang mit Matrixdruckern verwendet.

Online: Betriebsart des Druckers, die angibt, daß der Drucker bereit ist, vom Computer Daten zu empfangen.

Papierparkfunktion: Mit einer Papierparkfunktion kann zum Bedrucken von Einzelblattpapier das Endlospapier auf Knopfdruck zurückgefahren werden, ohne es auszuspannen. Wenn das Bedrucken von Einzelblattpapier beendet ist, wird auf Tastendruck das Endlospapier wieder vorgeschoben.

Plotter: Wird in erster Linie für technische Zeichnungen (CAD) benutzt. Computergesteuerte Stifte zeichnen direkt auf das Papier, wodurch hohe Präzision und exakte Linienführung erreicht werden.

Postscript: Eine sogenannte Seitenbeschreibungssprache oder Programmiersprache zur grafischen Ausgabesteuerung von Druckern, die automatisch mit der höchstmöglichen Auflösung des Ausgabegerätes arbeitet.

Proportionalschrift: Druckschrift, bei der die Abstände zwischen den Buchstaben an die Breite der zu druckenden Zeichen angepaßt sind. Dadurch ergibt sich ein harmonisches Schriftbild.

Thermodrucker: Die Buchstaben und Grafiken werden durch beheizte Na-

deln punktweise von einem Spezialfarbband auf das Papier übertragen oder direkt in Spezialpapier eingebrannt. Die Geräuscentwicklung gegenüber den normalen Nadeldruckern ist deutlich geringer, allerdings können keine Durchschläge erzeugt werden.

Tintenstrahldrucker: Die Buchstaben und Grafiken werden aus kleinen Tintentröpfchen punktweise zusammengesetzt. Der Vorteil gegenüber Nadeldruckern ist die geringe Geräuscentwicklung, der Nachteil die fehlende Möglichkeit, Durchschläge zu erzeugen.

Toner: Rußähnliches, feines, schwarzes Pulver, das aus Farbpigmenten besteht und als Druckfarbe für Laserdrucker benutzt wird.

Traktor: Vorrichtung zur Zuführung von Endlospapier, das heißt zusammenhängende, durch eine Perforation getrennte Seiten.

Zeichensatz: Vollständiges Alphabet mit Sonderzeichen, das komplett aus einem bestimmten Schrifttyp bestehen (Bild 4). Klassische Druckerzeichensätze sind zum Beispiel Times, Courier und Helvetica.

Grafik:

Alibianwendungen, die dem Benutzer die Möglichkeit geben, mangelndes Zeichentalent mit unzureichender Monitorauflösung zu entschuldigen.

Auflösung: Die Zahl der von einem Programm, einer Grafikkarte oder einem Monitor maximal darstellbaren Zeilen und Spalten.

Flatscreen: Flache Bildschirmbauweise, die schärfere Darstellung vor allem in den Ecken des Bildschirms erlaubt.

Grafikkarte: Zusatzkarte, die die Darstellung von Grafiken auf dem Computer erlaubt. Die gebräuchlichsten Grafikkarten sind CGA, EGA, VGA und Hercules. Die Karten unterscheiden sich in der Höhe der Auflösung und der Zahl der darstellbaren Farben voneinander.

Hardcopy: Übertragung des Bildschirminhalts an den Drucker.

Hertz (Hz): Die Einheit Hertz ist ein Maß für die Frequenz. Eine Frequenz von 50 Hz bedeutet, daß 50 Schwingungen in einer Sekunde stattfinden.

Horizontalfrequenz: Auch Zeilenfrequenz genannt. Zahl der pro Sekunde

aufgebauten Bildzeilen. Die Zeilenfrequenz ist von der verwendeten Grafikkarte abhängig und muß mit der des verwendeten Monitors übereinstimmen.

LCD: Abkürzung für »Liquid Crystal Display« (Flüssigkristallanzeige). Bildschirm, der die Punkte durch Ansteuerung und Drehung polarisierter Flüssigkristalle erzeugt.

Monochrom-Monitor: Monitor, der keine Farben darstellen kann.



Bild 5. Ein moderner Hand-Scanner am Beispiel des Scanners Plus von Logi

Multiscan-Monitor: Monitor, der mehrere Zeilenfrequenzen verarbeiten kann und deshalb zum Einsatz mit unterschiedlichen Grafikkarten geeignet ist.

Pixel: Auch Bildpunkt genannt. Kleinste, von einem Programm, einer Grafikkarte oder einem Monitor darstellbare Bildeinheit (siehe auch Auflösung). Auflösung und Darstellungsqualität hängen von der Zahl der darstellbaren Bildpunkte und dem Abstand der Bildpunkte zueinander ab.

RGB: Abkürzung für »Rot-Grün-Blau«. Bezeichnung für das Farbbild eines Computers, das sich aus den drei erwähnten Farbanteilen zusammensetzt und durch verschiedene Intensitäten der drei Anteile auch Mischfarben darstellen kann.

RGBI: Abkürzung für »Rot-Grün-Blau-Intensität«.

Scanner: Lesegerät, das Helligkeitsinformationen einer gedruckten oder fotografierten Vorlage in für den Computer verarbeitbare Informationen umwandelt (Bild 5). Gescannte Bilder können dann im Computer zum Beispiel mit Zeichenprogrammen weiterverarbeitet werden.

Vertikalfrequenz: Auch Bildfrequenz genannt. Zahl der pro Sekunde vom Monitor dargestellten Einzelbilder.

WYSIWYG: Abkürzung für »What You See Is What You Get« (Was Sie sehen ist, was Sie bekommen). Bild-

schirmdarstellung, die die Formatmerkmale der Datei so zeigt, wie sie später im Ausdruck erscheint.

Anwendungen:

Programme, die die Zeit zur Lösung bestimmter Aufgaben von acht auf sieben Stunden verkürzen, nachdem man sich vorher ein Jahr lang in das Programm eingearbeitet hat.

Benutzeroberfläche: Hilfsprogramm für die Erleichterung der täglichen Arbeit mit dem Computer.

CAD: Abkürzung für »Computer Aided Design« (Computerunterstütztes Konstruieren beziehungsweise Entwerfen durch grafische Datenverarbeitung). Konstruktionsprogramm zum Entwurf, zur Darstellung und zur Berechnung von technischen Zeichnungen.

Datenbank: Programm zur Verwaltung und Aufbereitung von Datensätzen, etwa Adressen, Produktinformationen oder Personalkarteien.

DTP: Abkürzung für »Desktop Publishing« (Druckmedienherstellung am Schreibtisch). Programm zum Entwerfen (Layouten) von Dokumenten, zum Beispiel Büchern, Flugblättern oder Zeitungen.

Integriertes Paket: Zusammenfassung verschiedener Anwendungsprogramme unter einer einheitlichen Benutzeroberfläche zum einfachen Aus-

tausch von Daten zwischen den einzelnen Modulen, etwa Textverarbeitung, Datenbank und Tabellenkalkulation.

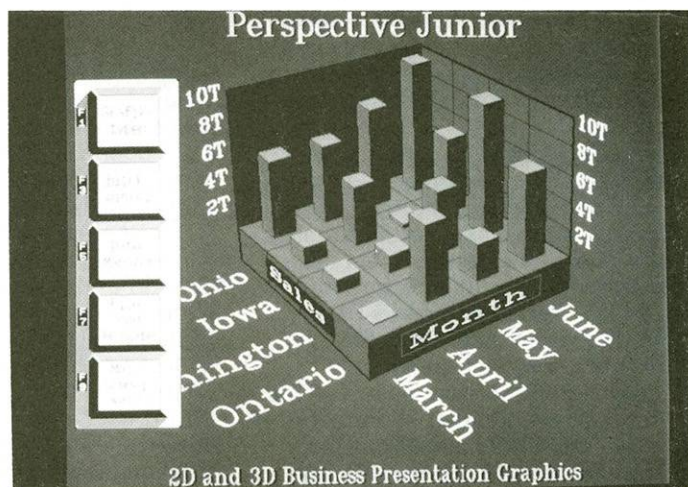
Präsentationsgrafik: Programm zur Aufbereitung und Darstellung von Zahlenwerten (Statistiken) in grafischer Form (Bild 6).

Tabellenkalkulation: Programm zur Berechnung von mathematischen Zusammenhängen innerhalb sogenannter Arbeitsblätter (Spreadsheets).

Textverarbeitung: Programm zum Schreiben und Aufbereiten von Texten. Ersetzt die Schreibmaschine und bietet viele zusätzliche Funktionen, zum Beispiel die nachträgliche Formatierung von Texten und das nachträgliche Austauschen, Verändern oder Löschen von Textpassagen.

Utility: Hilfsprogramm, das den Anwender bei kleineren Problemlösungen unterstützt (zum Beispiel Taschenrechner, Notizblock, Telefonverzeichnis und Terminkalender).

Bild 6.
Zur übersichtlichen Darstellung von Daten eignen sich Präsentationsgrafiken



Programmierung:

Den Computer mit vielen unverständlichen Worten dazu zu veranlassen, das zu tun, was man mit wenigen einfachen Handgriffen schneller selbst erledigen könnte.

Com-Datei: Ausführbares Programm, das sich auf einen 64 KByte großen Speicherbereich beschränkt.

Compiler: Ein Compiler übersetzt einen Quelltext einer höheren Programmiersprache, wie beispielsweise Cobol, Fortran oder Pascal, in für den Prozessor verständliche Maschinenbefehle. Die Ausführungszeit kompilierter Programme ist im Vergleich zu gleichartigen mit einem Interpreter interpretierten Programmen wesentlich kürzer.

Editor: Ein Programm zum Schreiben, Ändern und Speichern von Texten.

Exe-Datei: Ausführbares Programm, das eine beliebige Länge haben darf.

Integrierte Entwicklungsumgebung: Faßt den Compiler, Linker und Editor unter einer gemeinsamen Benutzeroberfläche zusammen. So läßt sich ein Programm schreiben, übersetzen und zu einer lauffähigen Com- oder Exe-Datei zusammenführen, ohne die hierfür erforderlichen Entwicklungswerkzeuge und Hilfsprogramme getrennt aufrufen zu müssen.

Interpreter: Ein Interpreter ist ein Programm zur schrittweisen Übersetzung von Quelltexten in lauffähigen Maschinencode. Bei jedem Programmstart wird der Übersetzungs-

vorgang erneut durchgeführt. Der Compiler hingegen übersetzt den gesamten Quelltext auf einmal und führt ihn dann aus.

Linker: Ein Linker, auch Binder genannt, ist ein Programm, das bereits compilierte Programmteile zu einem lauffähigen Programm zusammenfaßt.

Programmiersprachen: Künstliche Sprachen zur Programmierung des Computers, beispielsweise Basic (Beginner's All-purpose Symbolic Instruction Code), C, Assembler oder Pascal.

Toolbox: Eine Sammlung kleinerer Hilfsprogramme (Routinen).

Tools: Werkzeuge, die bei der Programmentwicklung unterstützen.

(gb)

Die MS-DOS-Werkstatt

Für den PC-Einsteiger ist der Umgang mit dem Betriebssystem MS-DOS keineswegs eine Selbstverständlichkeit. Die häufig gebrauchten »interne« Befehle (wie »dir«, »copy«, »del« und »ren«) sind deshalb ebenso Gegenstand dieser Werkstatt wie die selten benutzten externen Befehle zur Festplattenverwaltung (»append«, »join«, »subst« oder »assign«).

Das Betriebssystem Ihres PC bekommen Sie meist nur umständlich mit Hilfe eines Handbuchs in Griff. In dieser Werkstatt erleben Sie das Betriebssystem in der Praxis, um eine Vorstellung davon zu erhalten, wie Sie Ihr System professionell einrichten können.

Für die Teilnahme an dieser »MS-DOS-Werkstatt« benötigen Sie einen PC, der entweder eine formatierte Festplatte hat oder Disketten formatieren kann. Ihr Handbuch hat dazu sicher die notwendigen Informationen bereit. Ferner sollte MS-DOS schon installiert sein. Falls Sie das betreffende Verzeichnis nicht »\dos« genannt haben, so setzen Sie an den entsprechenden Stellen einfach den von Ihnen gewählten Verzeichnisnamen ein. Sollten Sie im weiteren Verlauf der Werkstatt auf Befehle stoßen, die Ihr Computer nicht versteht, so verfügen Sie nicht über die MS-DOS-Version 3.3 – Ihr System kann demzufolge den Befehl auch nicht ausführen. Sie werden aber trotzdem den Ausführungen folgen können.

Leider arbeitet wohl nicht jeder Anwender bereits mit der neuesten Version von MS-DOS. Diese Fassung mit der Nummer 4.01 bietet hinsichtlich der Möglichkeiten zur Datenverwaltung auf Festplatten und zur Bedieneroberfläche einige Neuerungen – etwa Festplattenpartitionen über 32 MByte Kapazität. Selbstverständlich können Sie sämtliche hier angesprochenen Anweisungen unter MS-DOS 4.01 verwenden. Um herauszufinden, welche Version Sie benutzen, genügt es, am MS-DOS-Prompt – das heißt bei eingeschaltetem PC nach dem Zeichen »C:>« –

ver einzugeben und mit [Enter] zu bestätigen. Übrigens werden fast alle Befehle von MS-DOS mit der Enter-Taste abgeschlossen, erst dann kommen

Sie auch zur Ausführung. Neben den Unterschieden im Leistungsbereich der verschiedenen MS-DOS-Versionen, sorgt auch die auf den verschiedenen PCs unterschiedliche Benennung der Tasten für Verwirrung: Im folgenden wird [Enter] für eine Taste geschrieben, die auch als [Return], [↵] oder Eingabetaste bezeichnet sein kann. [Ins] heißt neuerdings Einfügen, [Del] heißt [Löschen] oder [Entf], [Ctrl] ist auch als [Strg] bekannt, [Ctrl-Break] als [Abbr], [PrtSc] als [Druck] und [Esc] als [Eing. Löschen]. [Home] wurde zu [Pos1], [End] zu [Ende], [PgUp] zu [Bild ↑], [PgDn] zu [Bild ↓], [Shift] zu [Umschalt] und so weiter. Die Tasten [Alt], [Ctrl] und [Shift] werden einzeln so gut wie nie benutzt – sie kommen meist als Tastenkombinationen zur Programmsteuerung, zur Eingabe von Optionen oder zur Ausführung doppelter, dreier oder vierfacher Tastaturbelegung vor. Halten Sie etwa mit der rechten Hand [Alt] so lange gedrückt, bis Ihre Linke nacheinander die [9] und die [2] angestrichelt hat (jeweils aus dem rechten numerischen Tastenblock), so ergibt sich beim Loslassen der »Backslash« [\]. Diese Tastenfolge ist kurz beschrieben als [Alt-92].

Das Fehlen eines deutschen Tastaturtreibers vermittelt erste Unlustgefühle

Wenn von MS-DOS die Rede ist, so stellt sich – zumal für den Einsteiger – die Frage, wie dieses ominöse Programm zu starten ist. Viele Anwender, die beruflich am PC beschäftigt sind, bekommen nämlich das Betriebssystem nie zu Gesicht, da beim Systemstart unverzüglich Anwendungen geladen werden, die keinen Ausstieg vorsehen. Aber fabrikneue

PCs sehen auf der MS-DOS-Startdiskette beziehungsweise auf der Festplatte vor, daß Sie erst an der Kommandoaufforderungszeile Ihres Betriebssystems landen, die so aussieht

C:>

Meist ist dann noch nicht einmal klar, wie Sie ein »ü«, »ä«, »ö«, »ß«, »?«, »\«, »y«, »z« oder einige weitere Zeichen anbringen. Dazu ist erst der entsprechende Tastaturtreiber zu laden: Bis zur Version 3.2 von MS-DOS waren Sie nach der Eingabe der beiden Befehlszeilen

```
cd \dos
kezbgr
```

Herr Ihrer Tastatur, wobei statt »kezbgr« »keybgr« am Monitor erscheint. MS-DOS 3.3 verlangt statt »kezbgr«

```
kezb gr,850,C:\dos\keyboard.sys
```

wobei angenommen wird, daß sich »keyboard.sys« im Verzeichnis »c:\dos« aufhält. Falls Ihre Tastatur trotz alledem ein »y« für ein »z« und Sie zum Narren hält, dann ist ein Blick ins Handbuch ganz nützlich. Das Betriebssystem MS-DOS für den PC koordiniert Ein- und Ausgabegeräte wie Monitor, Laufwerke, Tastatur, Drucker und Modem. Der Anwender muß seine Programme ausführen können und das Betriebssystem hat die Dateien auf dem Laufwerk zu organisieren.

Der PC kennt die Diskettenlaufwerke A und B, das Festplattenlaufwerk C und des weiteren reale oder »imaginäre« Laufwerke D bis Z. Die »imaginären« Laufwerke können etwa im Arbeitsspeicher als »RAM-Disk« konfiguriert sein, oder Sie weisen einem Verzeichnis per »join« einen Laufwerksbuchstaben zu.

Innerhalb eines Laufwerks sorgen Verzeichnisse (auch »Ordner« genannt) für Ordnung im Chaos der Dateien und Programme. Dadurch, daß Sie Ordner anlegen, hält sich

```
C>dir

Volume in Laufwerk C ist ALOIS'PC
Verzeichnis von C:\GEMBOOT

.                <DIR>          13.05.87    9.14
..               <DIR>          13.05.87    9.14
SNAPSHOT        13312         20.08.86   15.36
SNAPSHOT RSC    1452         20.01.86   17.25
CALCLOCK ACC    28672         20.03.87   15.17
GEM EXE         64512         20.03.87   15.17
GEM RSC         5992         20.03.87   15.17
MOUSE COM       7854         15.04.87   10.28
8 Dateien      794624 Bytes frei

C>prompt $p$g
C:\GEMBOOT>
```

Bild 1. Das eingestellte Verzeichnis machen Sie mit einem dieser beiden Befehle ausfindig

MS-DOS auch eine ganze Menge Arbeit vom Leib: Zum einen können Sie nämlich zu einem bestimmten Zeitpunkt nur in einen Ordner Einsicht nehmen (mit dem Dir-Befehl »directory«, englisch für »Verzeichnis«) beziehungsweise es aktivieren oder »aktuell« schalten. Zum anderen beschränkt sich MS-DOS dann aber auch darauf, Dateien oder Programme, mit denen Sie auf irgendeine Weise arbeiten wollen (anzeigen, löschen, kopieren, etc.), ohne besondere Vorkehrungen Ihrerseits nur in diesem aktuellen Verzeichnis zu suchen.

MS-DOS kennt für seine gesamte Arbeit vier Arten von Befehlen: interne und externe Systembefehle, Anwenderprogramme und Treiber. Eine bestimmte Art von Anwenderprogrammen, Benutzeroberflächen genannt, können die Befehlsgewalt des Betriebssystems teilweise übernehmen. Vor allem die häufig gebrauchten internen und externen Befehle werden durch Programme wie PC-Tools, Norton Commander, Ergo, Star Manager und andere nachgebildet und bedienungsfreundlicher gestaltet.

»dir« und »prompt« können das aktuelle Verzeichnis mitteilen

Wie wissen Sie aber überhaupt, in welchem Verzeichnis Sie sich gerade befinden? Die umständliche Methode des Nachforschens bedient sich des Dir-Befehls (Bild 1), eine einfachere Methode bietet das Prompt-Zeichen. Das Prompt-Zeichen, auch Bereitschaftsanzeige genannt, läßt sich nämlich so einstellen, daß es nicht nur das immerwährende »C>« ausgibt, sondern in vielen Variationen auf den Bildschirm auszugeben ist. Gestalten Sie Ihren Prompt in der Autoexec-Datei als

```
prompt $p$g
```

so wird mit der Laufwerksangabe gleich das aktuelle Verzeichnis mitgeliefert. Aus »C>« wird unverzüglich »C:\>« oder »C:\text>« – je nachdem, in welchem Verzeichnis Sie sich befinden.

Nun kann der Prompt-Befehl noch viel mehr. In einigen Fällen müssen Sie sich der Systemdatei »ansi.sys« bedienen, wollen Sie diese Fähigkeiten nutzen. Den Aufenthaltsort der Systemdatei geben Sie in der Konfigurationsdatei »config.sys« mit

```
device=C:\dos\ansi.sys
```

bekannt. Nach einem Systemstart und den entsprechenden »Ansi-Sequenzen« können Sie Ihren System-Prompt gar »kunterbunt« gestalten

oder bestimmte Befehle auf die Funktionstasten legen.

Angenommen, Sie wollen mehrere Dateien löschen, sind dazu ohne die genannten Benutzeroberflächen etwa folgende Befehlszeilen notwendig:

```
del brief1.txt
del datei1.dbf
del tabelle3.wks
del bild56.pcx
del start.bat
del graf.pic
```

Der Befehl »del« (delete, englisch für löschen) mit darauffolgenden Dateinamen läßt nämlich jeweils die genannte Datei verschwinden.

Ein bißchen viel Aufwand, um sechs Dateien von der Festplatte zu entfernen. Einen Weg zurück, also ein »undelete«, gibt es übrigens unter MS-DOS nicht. Dazu ist ein »Dietrich« erforderlich, wie ihn nur spezielle »Utilities« (Hilfsprogramme) bieten (PC-Tools, Star Manager, Norton Utilities).

Der Komfort der Benutzeroberflächen macht es überflüssig, sich allzu intensiv mit den vielen Optionen der gebräuchlichsten MS-DOS-Befehle auseinanderzusetzen, besonders da besagte Benutzeroberflächen auch schon für wenig Geld (insbesondere in der »Shareware«) zu haben sind.

Um gar nicht erst in die Verlegenheit zu kommen, »falsche« Dateien zu löschen, ist von Anfang an eine sorgfältige Organisation der Festplatte oder der Disketten notwendig. Dazu gehören sinnvolle Unterverzeichnisse für die verschiedenen Dateiartern ebenso wie aussagekräftige Dateinamen.

Zugriff auf Programme – entweder direkt über Verzeichniswechsel oder über »path«

Aber auch MS-DOS hält ein wenig Komfort über Funktions- und Cursor-Tasten bereit, beispielsweise zum Editieren seiner Kommandozeile: [F1] wiederholt zeichenweise das vorhergehende Kommando, [F3] reiht den Rest des früheren Kommandos auf, [Esc] bricht die Eingabe ab, [Del] löscht das Zeichen auf dem Cursor, [Cursor-Links] oder [Cursor-Rechts] zaubern die überstrichenen Zeichen hervor oder löschen diese.

MS-DOS verlangt bei der Befehlsabarbeitung vom Anwender einige Aufmerksamkeit. Interne Befehle wie »dir«, »copy«, »del«, »type« und so

```
C:\>ver
```

```
MS-DOS Version 3.20
```

```
C:\>vol
```

```
Volume in Laufwerk C ist ALOIS'PC
```

```
C:\>time
```

```
Aktuelle Zeit ist 10.39.56,59
```

```
Neue Zeit eingeben:
```

```
C:\>date
```

```
Aktuelles Datum ist Sam 26.08.1989
```

```
Neues Datum eingeben (tt-mm-jj):
```

```
C:\>set
```

```
COMSPEC=C:\command.com
```

```
DISPLAY=MDMONO
```

```
PATH=C:\;.;C:\MSDOS;C:\BATCH;\UTIL;
```

```
PROMPT=$p$g
```

```
C:\>path
```

```
PATH=C:\;.;C:\MSDOS;C:\BATCH;\UTIL;
```

```
C:\>\dos\chkdsk A:
```

```
Volume DOSEXTRA9 erzeugt 10 Aug 1989 18.34
```

```
362496 Bytes verfügbarer Platz auf Diskette
```

```
0 Bytes in 1 versteckte Dateien
```

```
53248 Bytes in 10 Benutzerdateien
```

```
309248 Bytes frei auf der Diskette
```

```
655360 Bytes Speicherplatz insgesamt
```

```
568960 Bytes fre
```

Bild 2. Die internen Befehle benötigen keinen Pfad zu ihrem Aufruf, wohl aber die externen, falls sie nicht im aktuellen Verzeichnis abgelegt sind


```
C:\TEMP\COPI>dir
```

Volume in Laufwerk C ist ALOIS'PC
Verzeichnis von C:\TEMP\COPI

.	<DIR>	26.08.89	12.11
..	<DIR>	26.08.89	12.11
PCGLOBE	TXT	4897	31.08.89 13.02
PRESSE1	TXT	5632	31.08.89 13.03
PCO	TXT	2419	27.08.89 13.07
UNTER	<DIR>	26.08.89	12.14
6 Datei(en)		729088	Bytes frei

```
C:\TEMP\COPI>xcopy *.* A: /s /w /D:30/08/89
```

Zum Kopieren beliebige Taste drücken

Quelldateien werden gelesen...

PCGLOBE.TXT

PRESSE1.TXT

2 Datei(en) kopiert

```
C:\TEMP\COPI>dir A:
```

Volume in Laufwerk A hat keinen Namen
Verzeichnis von A:\

PCGLOBE	TXT	4897	31.08.89 13.02
PRESSE1	TXT	5632	31.08.89 13.03
2 Datei(en)		351232	Bytes frei

Bild 3. Der erweiterte Copy-Befehl (»xcopy« für »extended copy«) kopiert sogar gemäß Datumsvorgaben

weiter lassen sich von jedem Laufwerk und von jedem Verzeichnis aus ohne besondere Umstände aufrufen. Gleichgültig also, welches das aktuelle Verzeichnis oder Laufwerk ist, mit

time

erfahren Sie immer die Systemuhrzeit, beziehungsweise Sie werden aufgefordert, diese Zeit neu zu setzen. Entsprechendes gilt für das Datum (Bild 2) mit:

date

Diese beiden Befehle sind also »omnipräsent« beziehungsweise »intern«. Anders sieht es bei den externen MS-DOS-Befehlen aus. Gesetzt den Fall, Sie haben als aktuelles Verzeichnis »C:\«, also das Stammverzeichnis Ihrer Festplatte, angemeldet und wollen eine Diskette formatieren (natürlich legen Sie eine fabrikneue, leere Diskette in den Diskettenschlitz!):

```
format A: /4
```

ist hier der einzugebende Befehl, da Sie eine Double-Density-Diskette auf Ihrem AT-Laufwerk formatieren wollen. Die Meldung »Befehl oder Dateiname falsch« irritiert Sie jedoch gewaltig. Der Grund dafür liegt darin, daß MS-DOS nur im aktuellen Verzeichnis nach der angegebenen Datei »format.exe« sucht und »format.exe« sich im Verzeichnis »C:\dos« befindet. Auf zwei Wegen kommen Sie jedoch ans Ziel:

```
cd \dos
```

```
format A: /4
```

ist der eine Weg, der erst den Wech-

sel in das entsprechende Verzeichnis vornimmt. Mit

```
\dos\format A: /4
```

führt MS-DOS nur den gewünschten Befehl aus, ohne das aktuelle Verzeichnis anzugeben. Diese Art der Befehlseingabe ist immer dann sinnvoll, wenn Sie anschließend im gerade aktuellen Verzeichnis weitere Aktivitäten vorhaben und wenn es sich zwischendurch nur um externe MS-DOS-Befehle oder um singuläre Kommandodateien handelt, die keine weiteren Dateien in anderen Verzeichnissen zu ihrer Abarbeitung erfordern. Ansonsten ist der Weg in das entsprechende Verzeichnis nicht immer zu umgehen.

Gebräuchliche interne Befehle, Druckbefehle und die Konfigurationsdatei

Dateien sind das, was Sie mit dem Befehl »dir« angezeigt bekommen. Da dies oft mehr ist, als Ihr Bildschirm fassen kann, rollen die oberen Zeilen dann so schnell aus dem Blickfeld, daß der Anwender keine Chance hat, mitzulesen. Probieren Sie deshalb gleich einmal die folgenden Optionen, die hier Abhilfe schaffen:

```
dir C:\
```

```
dir C:\ /w
```

```
dir A:
```

```
dir /p
```

```
dir C:\dos\*.exe
```

```
C:\TEMP\COPI>dir
```

Volume in Laufwerk C ist ALOIS'PC
Verzeichnis von C:\TEMP\COPI

.	<DIR>	26.08.89	12.11
..	<DIR>	26.08.89	12.11
PRESSE1	TXT	5632	31.08.89 2.27
PCGLOBE	TXT	4886	31.08.89 2.53
PCO	TXT	2417	27.08.89 6.36
UNTER	<DIR>	26.08.89	12.14
6 Datei(en)		731136	Bytes frei

```
C:\TEMP\COPI>copy . A:
```

.\PRESSE1.TXT

.\PCGLOBE.TXT

.\PCO.TXT

3 Datei(en) kopiert

```
C:\TEMP\COPI>dir A:
```

Volume in Laufwerk A hat keinen Namen
Verzeichnis von A:\

PRESSE1	TXT	5632	31.08.89 2.27
PCGLOBE	TXT	4886	31.08.89 2.53
PCO	TXT	2417	27.08.89 6.36
3 Datei(en)		348160	Bytes frei

Bild 4. Der Copy-Befehl ignoriert Unterverzeichnisse und vermeidet Rückfragen

```
dir C:\dos\*.com /w
```

```
dir C:\dos\for*.*
```

Sie erfahren auf diese Weise, daß auch einzelne Dateien oder Dateigruppen angezeigt werden. Jedoch ist die gezielte Dateianzeige mitunter aufwendig. Angenommen Sie suchen eine Datei, die irgendwo im Namen eine »1« enthält. Dann sind dazu die Befehle:

```
dir 1*.*
```

```
dir ?1*.*
```

```
dir ??1*.*
```

```
dir ???1*.*
```

```
dir ???1*.*
```

```
dir ?????1*.*
```

```
dir ?????1*.*
```

```
dir ?????1*.*
```

hilfreich. Ziemlich verwirrend und doch sehr einfach, diese verschiedenen Befehle, Fragezeichen und Sterne. Das Fragezeichen verkörpert dabei als »Joker« irgendein und nur ein Zeichen im Dateinamen. Der Stern steht als »Wildcard« für irgendeine Zeichenfolge ab der betreffenden Stelle bis zum Punkt. Der Punkt trennt nämlich den Dateinamen, der achteinstellig sein darf, von der Namensergänzung, die aus bis zu drei Stellen bestehen darf. Als Zeichen nicht erlaubt sind in Dateinamen:

»«, »/«, »\«, »[«, »]«, »<«, »>«, »+«, »., »:, »;, »=«, »>«. Diese Zeichen werden etwa zur Bezeichnung des Laufwerks (»C:«), des Verzeichnisses (»\«), der Namensergänzung (».«) oder zur Ein- und Ausgabeumleitung (»<«, »>«) verwendet.

Gebräuchliche Namenserverweiterungen sind unter anderem etwa »txt« für Textdateien, »bat« für Kommando-Stapeldateien, »exe« und »com« für lauffähige Programme und »sys« für Gerätetreiber.

Wollen Sie etwa in eine dieser Dateien Einsicht nehmen, so kann das mitunter sehr verwirrend sein. Befehle wie

```
cd \ dos
type format.exe
```

zeigen nur, daß Sie nicht versuchen sollten, diese Exe-Dateien so zu lesen. Textdateien lassen sich hingegen ganz problemlos mit dem Type-Befehl begutachten

```
type brief.txt
oder mit
type brief.txt > prn
copy brief.txt prn
```

drucken – zwei Methoden, schnell eine ausgewählte Datei auf dem Drucker auszugeben. Ist die Datei kleiner als eine Bildschirmseite, so ist der Ausdruck mit [Shift-PrtSc] noch schneller – sofern Ihr Drucker »online« ist. Aufwendiger wäre

```
\ dos \ print brief.txt
```

denn dafür muß erst ein externes Programm geladen werden.

Und falls Sie noch keine Textdateien zum Ansehen ausfindig machen konnten, probieren Sie es doch mal mit

```
cd \
type config.sys
```

und siehe da – Sie haben eine Systemdatei vor sich, die lesbar ist. In diese Konfigurationsdatei können Sie sogar weitere Treiber einbinden, falls Sie vielleicht eine »RAM-Disk« einrichten, den Bildschirmtreiber »ansi.sys« verwenden wollen oder ein CD-ROM-Laufwerk angeschlossen haben.

Angenommen, Sie wollen im erweiterten Arbeitsspeicher Ihres AT einen Bereich festlegen, der etwa Ihre Textverarbeitung aufnimmt, so fügen Sie der Konfigurationsdatei die Zeile

```
device=C:\dos\ramdrive.sys 350 /E
```

an. Diese Zeile bemerkt Ihr AT aber erst nach erneutem Systemstart. Beim Systemstart sucht MS-DOS unter anderem auch nach einer automatischen Startdatei namens »auto-exec.bat«. In diese Datei, die einzelne MS-DOS-Befehle nacheinander ausführt, schreiben Sie dann, welche Programmdateien Sie in dieses »elektronische« Laufwerk (RAM-Disk) kopieren wollen. Da Sie beispielsweise nur eine Festplatte C haben, ist das RAM-Laufwerk automatisch von MS-DOS mit »D« benannt.

Angenommen, Ihre kleine Textverarbeitung heißt »editor.exe« und sie befindet sich im Verzeichnis »C:\Texte«, so wissen Sie nach

```
copy C:\text\editor.exe D:
```

```
D:
editor
```

was ein elektronisches Laufwerk ist. Und damit Sie ein solches Laufwerk nicht in bleibender, unschöner Erinnerung behalten, sollten Sie entweder Ihre Textzeugnisse mit einem »realen« Pfadnamen versehen, also etwa »C:\text\dateiname«, oder Sie gestalten den Aufruf des Editors als Stapeldatei:

```
@echo off
cls
```

```
D:
editor
copy *.* C:\text
C:
```

Natürlich können Sie den Editor auch mit

```
D:editor
```

zur Aktivität bewegen, falls er keine weiteren Dateien für seinen Ablauf benötigt. Es soll demonstriert werden, daß Sie in RAM-Disks enorm schnelle Zugriffe auf Dateien haben. Dies ist vorteilhaft bei häufig benötigten Programm- oder Datendateien.

```
C:\MSDOS>tree

PFADVERZEICHNIS FÜR DATENTRÄGER ALOIS'PC

Pfad: C:\GEMGRAF
Unterverzeichnisse: Keine

Pfad: C:\GEMDRAW
Unterverzeichnisse: Keine

Pfad: C:\GEMAPPS
Unterverzeichnisse: Keine

Pfad: C:\GEMDESK
Unterverzeichnisse: Keine

Pfad: C:\GEMBOOT
Unterverzeichnisse: Keine

Pfad: C:\MSDOS
Unterverzeichnisse: Keine

Pfad: C:\GEMSTART
Unterverzeichnisse: Keine

Pfad: C:\GEMSYS
Unterverzeichnisse: Keine

Pfad: C:\WORD
Unterverzeichnisse: Keine

Pfad: C:\UTIL
Unterverzeichnisse:  MANAGER
                        NORT

Pfad: C:\UTIL\MANAGER
Unterverzeichnisse: Keine

Pfad: C:\UTIL\NORT
Unterverzeichnisse: Keine

Pfad: C:\MPLAN
Unterverzeichnisse: Keine

Pfad: C:\GWBAS
Unterverzeichnisse:  BAN

Pfad: C:\GWBAS\BAN
Unterverzeichnisse: Keine

Pfad: C:\SPIELE
Unterverzeichnisse:  BASPIELE
                        FARB
                        PIANO

Pfad: C:\SPIELE\BASPIELE
Unterverzeichnisse: Keine

Pfad: C:\SPIELE\FARB
```

Bild 5. Der Tree-Befehl liefert bis MS-DOS 3.3 verhältnismäßig unübersichtliche Verzeichnisbäume

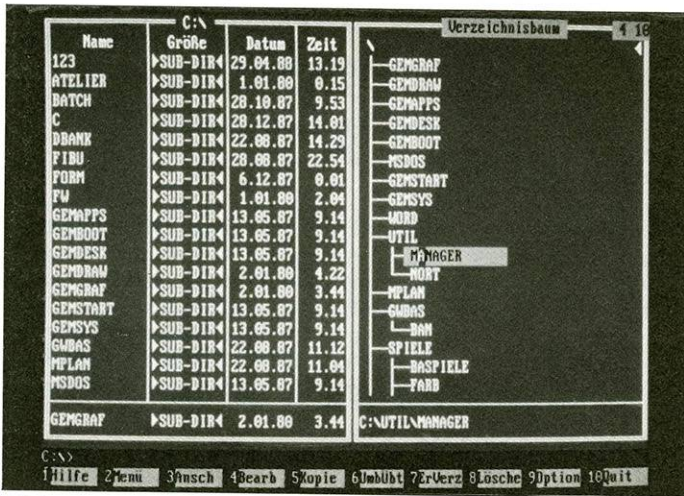


Bild 6. Eine Benutzeroberfläche für MS-DOS (Norton Commander) zeichnet den Verzeichnisbaum übersichtlicher

Doch solche elektronischen Laufwerke können die Daten nicht auf Dauer speichern, denn mit Ausschalten des PC sind alle Daten verloren. Vor dem Abschalten des PC ist also ein Kopiervorgang der Daten auf ein dauerhaftes Medium wie eine Festplatte oder eine Diskette zwingend.

Die Arbeit mit der Festplatte erfordert für eine geschickte Verwaltung mehr als nur die Befehle zum Anlegen, Wechseln und Löschen von Verzeichnissen:

```
md \bat
cd \bat
...
del \tempor\*,*
[J]a
cd \
rd \tempor
```

sondern auch elegante Programmaufrufe. Mehr als ein Dutzend Befehle zur Festplattenverwaltung helfen hierbei. Aber auch die »Systemvariablen« sind dafür gut.

Um ein bestimmtes (externes oder anwenderorientiertes) Programm zu starten, muß in das entsprechende Unterverzeichnis übergewechselt werden. Dies geschieht mit der Anweisung »cd \Verzeichnis«. Mit der Zeit verursacht dies eine ganze Menge fehleranfälliger Tipparbeit, und jedesmal ist die Überlegung anzustellen, wo denn nun welche Datei untergebracht ist. MS-DOS stellt für diese Gelegenheit den Path-Befehl zur Verfügung, der verzeichnisübergreifend nach Befehlsdateien sucht. Befehlsdateien sind solche, die mit der Namensergänzung »exe«, »com« oder »bat« ausgerüstet sind. Um also den Befehl zum Kopieren von Disketten (»diskcopy«) zu erreichen, müssen Sie künftig nicht mehr

```
C:\dos\diskcopy A: B:
eingeben, sondern es ist zweckmäßig, zu Beginn der Sitzung mit
path C:\;C:\dos;C:\text;C:\dbank;
C:\bat;C:\spiele;C:\util;
```

die Befehlsdateien gesucht werden, falls sich diese nicht im aktuell geschalteten Verzeichnis befinden. Angenommen, Sie arbeiten im Verzeichnis »\text« und wollen eine Diskette duplizieren, dann genügt der Befehl

```
diskcopy A: B:
um MS-DOS im dritten Verzeichnis
fündig werden zu lassen: Nach
»\text« wird das Stammverzeichnis
gesichtet, danach ist »\dos« an der
Reihe, in dem sich vermutlich die ex-
ternen MS-DOS-Befehle befinden.
```

Mit Systemvariablen bestimmen Sie die »Grundeinstellung« Ihres Computers

Dieser Path-Befehl für den Suchpfad sollte nicht allzu intensiv genutzt werden, denn die Suchzeiten werden bei einer vollen Festplatte dann sehr lang, wenn Sie eine Befehlsdatei am Ende eines langen Pfades abgearbeitet haben wollen. Gesetzt den Fall, Sie haben den Pfad

```
path C:\;C:\dos;C:\text;C:\dbank;
C:\bat;C:\spiele;C:\util;
```

definiert und wollen das Programm »snapshot.com« aus dem Verzeichnis »\util« aktivieren, so dauert das nicht nur auf einem langsameren PC seine Zeit. Starten Sie in diesem Fall statt

```
snapshot
einfach mit
\util\snapshot
```

Natürlich wollen Sie nicht nur das Betriebssystem auf Ihrer Festplatte haben. Eine Textverarbeitung, eine Datenbank, ein Grafik- und ein Kalkulationsprogramm oder alles zusammen in einem Paket gehören neben vielen Hilfsprogrammen zur Grundausstattung. Die Installation auf der Festplatte besorgen viele Programme von selbst per »Install-« oder »Ein-

richtungsdiskette« (»install.exe«, »instal.com«, »setup.exe« oder ähnlich benannt). Entweder legen die Programme für sich selbst ein Unterverzeichnis mit vorbestimmten Namen an, oder Sie haben Mitspracherecht bei dieser Namensvergabe. Jedenfalls ist es der Organisationsstruktur der Festplatte förderlich, wenn jedes Anwendungsprogramm seinen eigenen Ordner (Verzeichnis) hat.

Anwendungsprogramme wollen erst auf Ihrer Festplatte installiert sein

Sind Programme, die Sie auf Ihrer Festplatte haben wollen, nicht mit einem Installationsprogramm ausgerüstet, so empfiehlt sich das »Herüberholen« durch die Befehlsfolge

```
C:
md \util
cd \util
\dos\xcopy A:*.*/s
```

wobei es sich in diesem Beispielfall um ein Utility handeln soll, das mit etwaigen Hilfsdateien allein auf der Diskette in Laufwerk A untergebracht ist. Falls noch nicht vorhanden, erweitern Sie den Verzeichnisbaum um das Verzeichnis »util«. Der Befehl »xcopy« mit dem Parameter »/s« holt alle Dateien auf der Diskette mitsamt den eventuellen Unterverzeichnissen auf die Festplatte in das Verzeichnis »util«. Eine andere Option für den Xcopy-Befehl ist in Bild 3 zu sehen. Meist kommen Sie allerdings auch mit dem allgegenwärtigen Copy-Befehl aus, der jedoch auf ein einziges Kommando hin keine Unterverzeichnisse überträgt (Bild 4).

MS-DOS hat einen Befehl, der Ihnen alle Verzeichnisse und deren Unterverzeichnisse anzeigt. Leider ist dieser Befehl »tree« (Bild 5) bis zur Version 3.3 fast nicht zu gebrauchen, da er die Baumstruktur nur umständlich benennt. Gängige Benutzeroberflächen dagegen »zeichnen« den Baum (Bild 6) richtiggehend. Inzwischen ist jedoch auch die neueste MS-DOS-Version dazu in der Lage.

Der Aufruf von Programmen, mit denen Sie regelmäßig arbeiten, ist natürlich gesondert zu organisieren. Selbstverständlich können Sie, um etwa das Textprogramm zu starten, jedesmal so vorgehen:

```
cd \word
word
```

um danach in die Tabellenkalkulation zu gelangen:

```
cd \tabcalc
lotus
```

um auf diese Weise von Programm zu Programm zu »marschieren«. Mit Sta-

Tips & Tricks gesucht!

Die DOS International ist eine Zeitschrift zum Mitmachen. Vor allem die Tips & Tricks-Rubriken leben von Ihren Beiträgen, Ihren Ideen und Programmen. Sie müssen kein Schriftsteller sein, damit ein Beitrag aus Ihrer Feder (beziehungsweise Ihrer Tastatur) veröffentlicht wird. Einschicken können Sie alles, was irgendwie mit der Programmierung und Bedienung des PC zusammenhängt. Das sind zum Beispiel wichtige Hinweise zu Anwendungs-Software, trickreiche Stapeldateien, interessante Basic-, Assembler-, C-, oder Pascal-Routinen, aber auch Tips zur PC-Hardware. Für jeden veröffentlichten Trick gibt es – je nach Länge und Qualität – ein angemessenes Honorar.

Bitte beachten Sie beim Vorbereiten einer Einsendung die folgenden Hinweise, damit der Beitrag in der Redaktion schnell und unproblematisch bearbeitet werden kann:

- Legen Sie eine Diskette bei mit allen zum Trick gehörigen Dateien (Quelltext, ausführbares Programm, Beschreibung).
- Das Anschreiben sollte in kurzen Stichpunkten Ihre Idee beschreiben.
- Diskette und Anschreiben versehen Sie bitte mit Ihrer genauen Adresse und mit dem Programmnamen.
- Sie ersparen uns viel Arbeit, wenn Sie Quelltexte auf 44 Zeichen pro Zeile formatieren. Steuerzeichen sollten nicht enthalten sein.
- Die Programmbeschreibung sollte als unformatierter ASCII-Text vorliegen.

Spätestens zwei Wochen, nachdem Ihre Einsendung bei uns eingegangen ist, erhalten Sie schriftlich Bescheid.

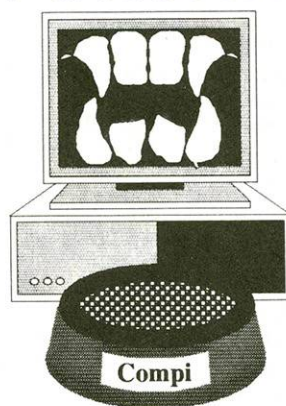
Unsere Adresse:



**Redaktionsbüro
Everts & Hagedorn
Redaktion DOS International
Stichwort Tips & Tricks
Hans-Stießberger-Straße 3
8013 Haar/München**

Wir freuen uns über jede Zuschrift.

Futter...



...für den Computer !

- **Standardsoftware**
- **Spiele**
- **Public Domain**
- **günstiges Leasing**
- **große Auswahl**
- **knallhart kalkuliert**

• **Gratis-Info Anfordern:**
oder Katalog mit über 100 Seiten
& ausführlichen Beschreibungen
für DM 5.- (wird bei Kauf verrechnet)

PSM • Postfach 82 05 05
8 München 82 • Tel. 439 28 87



PC-Anwendungspraxis

Schluß mit komplizierten Bedienungsanleitungen und trockenen Fachbüchern! Schluß mit der lästigen Rumfragelei!

- Der neue Fernlehrgang PC-Anwendungspraxis eröffnet Ihnen freie Bahn zu den modernen Computeranwendungen!
- Er vereint alles, was Sie zum selbstständigen Studium zu Hause benötigen. 12 leichtverständliche Lehrbriefe, 4 ausgezeichnete Anwenderprogramme, ein Set raffinierter Lerndisketten und eine wirksame Studienbetreuung.
- Er bildet Sie systematisch und gründlich vom Einsteiger zum versierten PC-Profi aus: Kompetenz im Umgang mit MS-DOS, Gewandtheit im Software-Einsatz und Sicherheit in allen professionellen PC-Anwendungen.

BON ☐ Senden Sie mir den 1. Lehrbrief für 3 Wochen kostenlos u. unverbindlich zum Teststudium.
☐ Senden Sie mir die Informationsschrift.

Name

Straße/Nr.

PLZ/Ort

Mein PC-Modell

Einsenden an Lehrinstitut DE238

Christiani • 7750 Konstanz • Postfach

DECOS - Briefsimulationen Spiel und Spaß in der 4. Dimension



Briefsimulationsspiele werden nach großen Erfolgen in USA nun auch bei uns immer beliebter. Bereits mehrere Tausend Bundesbürger lassen, in vergangene Zeiten zurückversetzt, die Geschichte neu aufleben und sind als Könige, Kaiser und Feldherren schon eifrig am erobern, kämpfen, besetzen und regieren. Und es ist ganz einfach: jeder Teilnehmer unserer Briefsimulationen sendet alle 14 Tage seine Befehle an den Spielleiter und erhält nach Computerauswertung neue Berichte mit den aktuellen Spielständen. Steigen Sie ein bei den realistischen Simulationen "Historie 1800", "Historie 1800 - Kolonialismus" und dem spannenden Westernspiel "Deadwood". Ein Riesenspaß ist allen Teilnehmern sicher!

Wir senden Ihnen gerne unser unverbindliches Info zu.
DECOS GmbH • Postfach 7 • 6382 Friedrichsdorf
Telefon 06175 - 7276

peldateien hingegen geschieht solch ein Aufruf nicht nur fehlerlos und schnell, sondern auch die dazugehörige Eintipparbeit wird unkompliziert übergeben. Sie müssen die Befehlsfolge nur einmal fehlerfrei in einer Textdatei speichern. In Zukunft reicht dann der Name der Textdatei, um etwa das Kalkulationsprogramm zu aktivieren. Angenommen, Sie verfrachten zweckmäßigerweise alle Stapeldateien in das Verzeichnis »\bat« und nennen folgenden Text »k.bat«

```
@echo off
cls
cd \tabcalc
lotus
cd \bat
cls
for %I in (*.bat) do echo %I
so genügt die Eingabe von
k
```

um die Kalkulation zu starten und sich nach dem Start wieder im Ausgangsverzeichnis zu befinden. Alles weitere zu den Stapeldateien ist in einer eigenen Werkstatt zusammengefaßt.

Kopierbefehle im Praxistest: »copy« contra »replace«

Nachdem Sie in Ihrer Textverarbeitung Briefe oder Berichte verfaßt haben, soll es darum gehen, diese Dateien auf Diskette zu »sichern« (Backup-Kopie) oder für irgendwelche Zwecke auf einem solchen Datenträger transportfähig zu machen. Natürlich bietet sich der Copy-Befehl an:

```
cd \text
copy *.txt A: \
```

Doch dieser Befehl hat eine sehr unangenehme Eigenschaft, denn er überschreibt ohne Warnung gleichnamige Dateien auf dem Ziellaufwerk.

Ein anderer Befehl gewährleistet hier in Verbindung mit verschiedenen Parametern mehr Sicherheit. Angenommen, Sie haben einen ganzen Tag an Ihren Abrechnungen gesessen und danach auch glücklich eine Kopie auf der Sicherungsdiskette angelegt. Doch am zweiten Tag, als Sie an der Datei weiterarbeiten, nimmt das Verhängnis seinen Lauf – eine falsche Taste ruiniert fast den ganzen Text! Nervös geworden, beenden Sie die Tragödie und speichern Ihren Text ordnungsgemäß, bevor Sie das Textprogramm verlassen. Zumindest haben Sie ja noch die intakte Sicherungskopie vom vorigen Tag auf Diskette, und bevor Sie diese wieder auf die Festplatte überspielen, wollen Sie noch schnell die anderen Textdateien sichern, und das erledigen Sie wiederum mit dem Copy-Befehl – , doch

```
C:\TEMP\COPI>dir
```

```
Volume in Laufwerk C ist ALOIS'PC
Verzeichnis von C:\TEMP\COPI
```

.	<DIR>	26.08.89	12.11
..	<DIR>	26.08.89	12.11
PCGLOBE	TXT	4897	31.08.89 13.02
PRESSE1	TXT	5632	31.08.89 13.03
PCO	TXT	2419	27.08.89 13.07
UNTER	<DIR>	26.08.89	12.14
6 Datei(en)		704512 Bytes frei	

```
C:\TEMP\COPI>dir A:
```

```
Volume in Laufwerk A hat keinen Namen
Verzeichnis von A:\
```

PCGLOBE	TXT	4585	29.08.89 13.37
1 Datei(en)		357376 Bytes frei	

```
C:\TEMP\COPI>replace *.txt A: \ /w /p /a
```

```
REPLACE Version 3.20 (C) Copyright Microsoft Corp. 1985, 1986
```

Beliebige Taste drücken, um Dateien hinzuzufügen k

```
A:\PRESSE1.TXT hinzufügen ? (J/N) j
A:\PRESSE1.TXT wird hinzugefügt
A:\PCO.TXT hinzufügen ? (J/N) j
A:\PCO.TXT wird hinzugefügt
```

```
2 Datei(en) hinzugefügt
```

```
C:\TEMP\COPI>dir A:
```

```
Volume in Laufwerk A hat keinen Namen
Verzeichnis von A:\
```

PRESSE1	TXT	5632	31.08.89 13.03
PCGLOBE	TXT	4585	29.08.89 13.37
PCO	TXT	2419	27.08.89 13.07
3 Datei(en)		348160 Bytes frei	

Bild 7. Der Replace-Befehl kann um vorhandene Dateien »herumkopieren«

halt, eigentlich wäre ja der Replace-Befehl der richtige gewesen.

Bild 7 zeigt, wozu der Replace-Befehl (»replace«, englisch für »ersetzen«) gut sein kann: Sie kopieren alle Textdateien von der Festplatte auf Diskette, wobei der Parameter »/A« veranlaßt, daß nur die im Zielverzeichnis unbekannten Dateien kopiert werden – Ihre intakte Sicherheitskopie auf Diskette wird demnach von der gleichnamigen defekten Datei auf der Festplatte nicht überschrieben. Natürlich kann der Copy-Befehl auch eleganter eingesetzt werden – zusammen etwa mit dem Rename-Befehl (»rename«, englisch für »Umbenennen«) hätten Sie eine Vater-Sohn-Sicherung (einfaches Backup) zustande gebracht:

```
ren A:*.txt A:*.sik
copy C:\text\*.txt A: \
```

aber spätestens dann hätten Sie bemerken müssen, daß beispielsweise die Datei »Briefe« nur noch als »briefe.sik« brauchbar ist.

So wie er hier verwendet wird, ist mit dem Umbenennungsbefehl jedoch kein dauerhaftes Sicherungssy-

stem durchzuführen. Besser ist die Kommandofolge

```
copy A:*.txt A:*.sik
copy C:\text\*.txt A: \
```

Der nächste Programmlauf hätte nun jedoch auch diese brauchbare Sicherungsdatei mit der Endung »sik« überschrieben.

Die Umgebungsvariablen: alles Einstellungssache, die Sie mit »set« befehlen

Das Betriebssystem, soviel ist bis hierher deutlich geworden, ist vor allem ein Programm, das dem Anwender eine Eingabezeile zur Verfügung stellt, in der dieser seine Wünsche durch Befehle mitteilt. Diese Befehle betreffen direkte Aktionen oder nur Einstellungen, die von Fall zu Fall nützlich sein können.

Die Einstellungen sind in der Konfigurationsdatei »config.sys« unterzubringen, Sie übergeben diese Infor-


```

C:\>dir w.*

Volume in Laufwerk C ist ALOIS'PC
Verzeichnis von C:\

Datei nicht gefunden

C:\>dir \batch\w.*

Volume in Laufwerk C ist ALOIS'PC
Verzeichnis von C:\BATCH

W          BAT          165      1.01.88    0.44
          1 Datei(en)      776192 Bytes frei

C:\>type w.bat
Datei nicht gefunden

C:\>append \batch

C:\>type w.bat
@echo off
CLS
echo      ... SIE KOMMEN GLEICH ZU WORD
echo      -----
C:
cd \word
word %1 /G /H
cd \batch
tv

```

Bild 8. Mit dem Append-Befehl heben Sie zwar die Verzeichnisstruktur nicht auf, aber die Dateien sind plötzlich zu mancherlei Aktionen bereit

mationen per Set-Befehl in die »Umgebung« (Deutsch für »Environment«), oder Sie benutzen gegebenenfalls einen Verbundbefehl (»join«, »assign« oder ähnliche). Die Systemumgebung ist ein kleiner Speicherbereich, dessen Größe Sie festlegen können. Der Befehl

set

bringt den Inhalt auf den Bildschirm (Bild 2). In diesen Speicherbereich können Sie nicht nur die Pfadvariable, den Kommandoprompt oder den Aufenthaltsort des Kommandoprozessors mit dem Set-Befehl plazieren, sondern auch Variable, die Sie selbst festlegen. Meist ist allerdings die Bekanntgabe des Kommandoprozessors (»command.com«) ungleich wichtiger, da manche Programme diesen elementaren Bestandteil des laufenden Systems aus dem Arbeitsspeicher verdrängen. Angenommen, Ihr »Kommandeur« ist in das Stammverzeichnis Ihrer Festplatte kopiert worden, dann verhindert die Zeile

```
set comspec=C:\command.com
```

eingangs Ihrer PC-Sitzung wohl des öfteren die Meldung »Command.com« falsch oder fehlend. System angehalten«.

Neben den bekannten Einstellungen (»path«, »prompt«, etc.) können Sie also auch eigene Informationen hinterlegen – zumindest für die ungestörte Sitzung sind diese Informationen dann von anderen Programmen (vor allem von Stapeldateien und Anwendungsprogrammen) abrufbar. Geht es um die Verzeichnisse, die gelöscht werden diese Zeichenketten

durch die leere Zeichenkette (gar kein Zeichen)

```
set Variablenname=
```

Um beispielsweise den Pfad zu löschen, ist dann diese Anweisung nützlich:

```
set path=
```

Kurz gesagt ist eine Variable ein »Variablenname«, der stellvertretend für eine andere Kombination von Zeichen oder eine Zahl steht. Ein Beispiel soll dies veranschaulichen

```
set Tag=heute
```

In diesem Beispiel wird der Variablen »Tag« das Wort »heute« zugeordnet. In einer Stapeldatei, die Sie gewöhnlich aufrufen, können Sie dann etwa bestimmte Befehlsfolgen davon abhängig machen, ob Sie den genannten Set-Befehl heute schon abgesetzt haben:

```
if "%tag%"=="heute" goto nichtgestern
```

Sie können auch beliebigen anderen Variablen, deren Namen Sie frei erfinden können, beliebige andere Werte zuordnen. Falls Sie dies allerdings in größerem Ausmaß planen, so müssen Sie den Umgebungsbereich vergrößern gemäß

```
set comspec=C:\command.com /E=4000
```

Eine »Systemvariable« unterscheidet sich von einer »beliebigen« Variable dadurch, daß das Betriebssystem von vornherein erwartet, daß die Variable bestimmte Werte enthält, die für Betriebssystemfunktionen verwendet werden. Im Falle von »path« erwartet MS-DOS, daß der Anwender unter diesem Variablennamen alle Unter-

verzeichnisnamen speichert, in denen MS-DOS nach Programmen suchen soll, falls ein aufgerufenes Programm nicht im gerade aktuellen Unterverzeichnis auffindbar ist.

Der Funktionsweise von »path« sehr ähnlich ist die MS-DOS-Anweisung »append«. Allerdings werden damit keine Programmsuchwege festgelegt, sondern der Suchweg für normale Datendateien wird gespeichert. »append« kann sowohl als Systemvariable als auch als externer MS-DOS-Befehl Verwendung finden.

```
append /E
```

nimmt den folgenden Append-Befehl in die Systemumgebung auf, so daß Sie sich etwa nach

```
append \batch
```

aus irgendeinem Verzeichnis heraus die Dateien in »\batch« ansehen können (Bild 8):

```
type Dateiname
```

Falls sich eine Datei nicht im aktuellen Verzeichnis befindet, sucht MS-DOS daraufhin alle mit »append« festgelegten Verzeichnisse nach der Datei »Dateiname« durch und gibt sie schließlich auf dem Bildschirm aus, falls es sie in »\batch« findet.

Mit »append« findet Ihr Programm seine Dateien auch in anderen Verzeichnissen

Ebenso wie bei »path« müssen Sie in Kauf nehmen, daß bei jedem neuen Aufruf von »append« der alte Suchpfad gelöscht wird. Soll dies absichtlich geschehen, so geben Sie

```
append ;
```

ein.

Es gibt Anwenderprogramme (etwa Spiele), die verlangen, daß Sie eine bestimmte Diskette in ein bestimmtes Laufwerk legen. Das kann lästig sein – nicht nur, weil Sie die Diskette Ihrer Freundin geliehen haben – sondern weil Sie Ihre Daten generell gerne auf der Festplatte hätten. Mit »assign« vertauschen Sie die Kennbuchstaben der verschiedenen Laufwerke. Um beispielsweise mit dem störrischen Spielprogramm zurechtzukommen, geben Sie einfach die Zeile ein

```
assign a=c
```

Alle Zugriffe auf das Diskettenlaufwerk A werden hierdurch bei sauber programmierter Software auf die Festplatte C umgeleitet (Bild 9). Um die Laufwerkszuordnungen wieder rückgängig zu machen, geben Sie einfach nochmals »assign« ein – sofern dieses externe MS-DOS-Programm durch die Laufwerkverketung überhaupt noch erreichbar ist.

Bei

assign c=a

müssen Sie schon die Systemdiskette (natürlich in A einzulegen) oder die Tastenfolge [Ctrl-Alt-Del] bemühen, um die Laufwerke wieder auseinanderzukoppeln. Diese Tasten führen eine vollständige Neuorganisation des Arbeitsspeichers herbei, die nicht immer gewünscht wird. Ferner ist zu erwähnen, daß der Assign-Befehl durch sehr hardwarenahe Programmierung unterlaufen werden kann – nicht nur das Digger-Spiel, sondern auch die MS-DOS-eigenen Befehle »format« oder »diskcopy« erkennen ihn nicht an.

Der Subst-Befehl kann mehr, als nur lange Pfadangaben abzukürzen

Verzeichnisnamen einzugeben, ist eine aufwendige Prozedur. Ältere Programme weigern sich mitunter sogar, Pfadangaben vor Dateinamen zu akzeptieren. Auch beim Laden und Speichern ist es unkomfortabel, gar dritte oder vierte Unterverzeichnisnamen aufführen zu müssen. Der Append-Befehl hilft hier auch nicht, da es nicht darum geht, eine Datei zu finden, sondern »ordnungsgemäß« zu verwalten. Dies gelingt mit dem Befehl »subst«, der ein Verzeichnis wie eine Diskette behandelt:

subst D: \text\briefe

ersetzt (»substitute«, englisch für »ersetzen«) den Verzeichnisnamen durch das viel kürzere »d:«. Alle Aktionen, die Sie fortan auf das Laufwerk D beziehen, das natürlich nicht real existieren muß (etwa als zweite Festplatte), werden in Wirklichkeit im Verzeichnis »c:\text\briefe« (Bild 10) stattfinden. Natürlich können Sie auch

subst a: \text\briefe

ansetzen, aber dabei wird das eigentliche Diskettenlaufwerk A »verdeckt«, solange »subst« aktiv ist. Das heißt, Sie können derweil nicht von der Diskette lesen oder darauf schreiben! Mit dem Parameter »/D« können einzelne Zuweisungen durch »subst« wieder rückgängig gemacht werden, wobei Sie den Laufwerksbuchstaben mit angeben müssen:

subst a: /D

Nun kennt MS-DOS von sich aus nur Laufwerke bis zum Buchstaben E. Wollen Sie weitere benützen, so müssen Sie Ihr Vorhaben in der Konfigurationsdatei (»config.sys«) bekanntmachen, und zwar mit:

lastdrive=Buchstabe

wobei der Buchstabe auch ein »Z« sein kann.

Den umgekehrten Weg schlägt »join« ein: Ein Laufwerk wird als Verzeichnis behandelt, das zwar leer zu sein hat, nicht so jedoch das Laufwerk. Angenommen, an Ihrem PC arbeiten mehrere Benutzer, und Sie sind es leid, von jedem einzelnen den Dateimüll dauernd von der Festplatte zu leeren. Oder Sie lagern gerne selber Dateien oder Programme auf Diskette aus. Dazu richten Sie also ein leeres Verzeichnis ein

md \leer

und verbinden es mit dem Diskettenlaufwerk (Bild 11)

join A: C:\leer

dann können Ihre Anwendungsprogramme entweder die Ausgabe via »\leer« an die jeweils eingelegte Dis-

kette leiten, oder Sie holen sich von dort Programm- oder Datendateien. Erwerben Sie eines Tages eine größere Festplatte, auf der alle Daten Platz haben, dann brauchen Sie Ihre Programme nicht »umzustricken«, sondern nur den vorangestellten Join-Befehl wegzulassen.

Bei größeren Datenbanken ist dieses Verfahren vielleicht nicht geschickt, bei Texten oder Kalkulationen genügen aber eventuell sogar Double-Density-Disketten, die die einzelnen Mitbenutzer in Eigenregie verwalten können.

Auch bei »join« dient der Parameter »/D« dazu, eine Zuordnung wieder zu löschen. Die Syntax ist die gleiche wie bei »subst«.

Beachten Sie auch, daß ein Inhaltsverzeichnis mit »dir« bei einem mit »join« verknüpften Unterverzeichnis am Schluß immer den noch freien

C:\GEMDRAW>dir A:

Volume in Laufwerk A ist DMS_TURBO50
Verzeichnis von A:\

INSTDMS	BAT	639	3.05.89	14.15
DMS	BAT	31	3.05.89	13.50
DMS1	EXE	3648	3.05.89	13.48
DMS2	EXE	41216	3.05.89	13.48
MIRROR	FIL	6144	3.05.89	14.29
ASSIGN	COM	1524	15.04.87	10.28
TES	BAT	338	1.01.80	5.37
			7 Datei(en)	303104 Bytes frei

C:\GEMDRAW>dir C:

Volume in Laufwerk C ist ALOIS'PC
Verzeichnis von C:\GEMDRAW

.	<DIR>		2.01.80	4.22
..	<DIR>		2.01.80	4.22
DRAW	APP	78336	1.08.84	0.13
DRAW	RSC	10789	28.06.85	10.57
			4 Datei(en)	772096 Bytes frei

C:\GEMDRAW>assign A=C

C:\GEMDRAW>dir A:

Volume in Laufwerk A ist ALOIS'PC
Verzeichnis von A:\GEMDRAW

.	<DIR>		2.01.80	4.22
..	<DIR>		2.01.80	4.22
DRAW	APP	78336	1.08.84	0.13
DRAW	RSC	10789	28.06.85	10.57
			4 Datei(en)	772096 Bytes frei

C:\GEMDRAW>dir C:

Volume in Laufwerk C ist ALOIS'PC
Verzeichnis von C:\GEMDRAW

.	<DIR>		2.01.80	4.22
..	<DIR>		2.01.80	4.22
DRAW	APP	78336	1.08.84	0.13
DRAW	RSC	10789	28.06.85	10.57
			4 Datei(en)	772096 Bytes frei

C:\GEMDRAW>assign

Bild 9. Ein Laufwerk dominiert mit »assign« ein anderes


```

C:\TEXT>dir D:
Ungültige Laufwerksangabe

C:\TEXT>subst D: \text\brieft

C:\TEXT>dir D:

Volume in Laufwerk D ist ALOIS'PC
Verzeichnis von D:\

.                <DIR>          1.01.80   9.14
..               <DIR>          1.01.80   9.14
BILD2    EXE        512    1.01.80   9.40
CARMEN   TXT        181    1.01.80   0.03
BEWERB   TXT       3072    1.01.80   0.53
MONIKA   TXT       1536    1.01.80   0.54
PATTY    TXT       5632    1.01.80   0.56
ASTRID   TXT       5120    16.08.89  15.03
      8 Datei(en)       727040 Bytes frei

C:\TEXT>subst
D: => C:\TEXT\BRIEF
C:\TEXT>subst D: /d

C:\TEXT>dir D:
Ungültige Laufwerksangabe

```

Bild 10. Mit »subst« sind auch lange Verzeichnisnamen ganz kurz

Speicherplatz des Speichermediums anzeigt, in dem sich das Verzeichnis wirklich (physikalisch) befindet. Zwar können Sie in dem Unterverzeichnis so viele Daten ablegen, wie auf den »angehängten« Datenträger passen, die Anzeige »Bytes frei« im Inhaltsverzeichnis bleibt davon aber unberührt (Bild 11).

Sie werden verstehen, daß in dieser Einsteiger-Werkstatt viele MS-DOS-Befehle nicht vorgeführt werden

Die mit externen Programmen vorgenommenen Einstellungen gelten ohne Zurücknahme bis zum Systemabsturz

Im Umgang mit den hier beschriebenen Befehlen ist außerdem unbedingt zu beachten, daß nicht alle MS-DOS-Befehle korrekt funktionieren, solange Zugriffspfade umgeleitet sind. Beispielsweise beachten »chkdsk«, »fdisk«, »diskcopy« und einige andere hardwarenahen Befehle keine Laufwerks- oder Verzeichnisverbunde mit »join«, »subst« oder »assign«. Sollten Sie also diese Befehle benutzen wollen, so setzen Sie zuvor alle Zugriffspfade auf die Standardwerte zurück, um keine bösen Überraschungen zu erleben. Die Wirkungs-dauer der Verbundeinstellungen geht ohne Rücknahme der Befehle bis zum Systemabsturz – sei dieser nun manuell, durch höhere Gewalt oder von Programmfehlern herbeigeführt. Stellen Sie auf jeden Fall vor Gebrauch dieser Verbundbefehle sicher, daß Sie auch ohne Neustart des Computers wieder rückgängig gemacht werden können.

können. Denn was nutzen Ihnen – im Moment jedenfalls noch – Befehle wie »backup« oder »restore«, mit denen Sie den Inhalt Ihrer Festplatte auf Diskette sichern und wieder zurückschreiben, wo doch zum jetzigen Zeitpunkt Ihre Festplatte noch fast leer ist. Auch der Zeileneditor »edlin« verlangt einiges an Einfühlungsvermögen von Ihnen, und so wurde dieses Thema in der Hoffnung weggelassen, daß Sie sich baldmöglichst einen komfortablen »Full-screen-Editor« anschaffen, der Ihren Schreibkünsten einen gesamten Bildschirm zur Verfügung stellt (Norton-Editor, Personal-Editor, PC-Write oder ähnliche Programme). Weitere MS-DOS-Befehle sind an anderer Stelle dieser DOS-Extra-Ausgabe erläutert. Wieder andere Befehle sind doppelt vorhanden (wie »delete« und »erase«), veraltet oder mit Vorsicht zu genießen (»fcbs«, »fastopen«), nur für Programmierer interessant (»exe2bin«, »debug«, »stacks«), einmalig (»fdisk«) oder in speziellen Gerätekonfigurationen wichtig (etwa »share«).

(al)

```

C:\LEER>dir

Volume in Laufwerk C ist ALOIS'PC
Verzeichnis von C:\LEER

.                <DIR>          1.01.80   7.57
..               <DIR>          1.01.80   7.57
      2 Datei(en)       778240 Bytes frei

C:\LEER>join A: \leer

C:\LEER>dir

Volume in Laufwerk C hat keinen Namen
Verzeichnis von C:\LEER

NEDITOR   COM        2992    18.02.87  12.16
PE         EXE       45696    8.03.85  12.02
PE         PRO        4019   19.05.87   2.21
PE         SIN         229   16.10.86  12.09
EDIT       EXE       58112    9.05.83  12.00
XEDIT      EXE     110080    3.09.85  12.50
PE         HLP        9216   28.02.88  14.18
RPED       EXE       4804   20.08.86   9.57
      8 Datei(en)       778240 Bytes frei

C:\LEER>join
A: => C:\LEER

C:\LEER>join A: /d

C:\LEER>join

C:\LEER>dir

Volume in Laufwerk C ist ALOIS'PC
Verzeichnis von C:\LEER

.                <DIR>          1.01.80   7.57
..               <DIR>          1.01.80   7.57
      2 Datei(en)       778240 Bytes frei

```

Bild 11. Der Befehl »join« überträgt scheinbar den »Inhalt« der gerade eingelegten Diskette in ein leeres Festplattenverzeichnis

Die Befehlsstapel-Werkstatt

Mit mehreren Befehlen kann MS-DOS zwar nicht gleichzeitig umgehen, doch als Stapel hintereinander akzeptiert es sie allemal. Und davon profitieren Sie in dieser Werkstatt: Ihre Festplatte erhält ein komfortables Menüsystem, und den Drucker steuern Sie ab sofort ganz zwanglos.

Die meistverkaufte Programmiersprache für den PC ist neben GW-Basic eine oft nicht beachtete Sprache, die mit MS-DOS ins Haus kommt: die Batch-Sprache. Das Betriebssystem beschränkt sich nämlich nicht nur darauf, einen Befehl auszuführen, sondern es kann sich auch eine ganze Reihe von zusammen übergebenen Befehlen »merken« und der Folge nach abarbeiten. Neben den zulässigen Befehlen, die Sie in der Eingabezeile übergeben, dienen spezielle Wörter der Batch-Sprache dazu, den Programmablauf zu steuern (siehe »Befehlsreferenz Stapelbefehle«). Ist ein solcher Befehlsblock einmal angelaufen, kann der Anwender ihn nur noch stoppen, wenn es spezielle Tastenkombinationen ergreift: [Ctrl-C] oder [Ctrl-Break] sind die Tasten, die die Meldung »Stapel Job beenden (J/N?)« provozieren. Falls Sie ohne das in Listing 1 angegebene Programm arbeiten, das situative Interaktionen erlaubt, ist es vor allem beim wiederholten Batchdurchlauf sinnvoll, diesen durch diese Notbremse anzuhalten.

Noch ein Wort zu den verschiedenen MS-DOS-Versionen: Je nach Version und Land erhalten Sie unterschiedliche Meldungen. Um kein vollständiges Compendium schreiben zu müssen und weil die Meldungen sich ähneln, wird jeweils nur auf eine Version eingegangen, ohne diese ausdrücklich zu nennen. Sind allerdings neue Befehle vorhanden – MS-DOS 3.3 hat zum Beispiel Neues zu bieten –, so ist dies vermerkt.

Die Reihenfolge, nach der MS-DOS Programmdateien sucht, ist nicht ganz belanglos

MS-DOS hat für die Betriebsart per Batch-Sprache eine Textdatei vorgesehen, die ein spezielles »Namensanhängsel« haben muß: Die Endung »bat« steht für »batch«, was hierzulande auch als »Stapel« bezeichnet wird. Mit diesen speziellen Batch-Da-

teien können Sie ständig wiederkehrende Befehlsabfolgen automatisieren oder, indem Sie Parameter übergeben, auch variable Abarbeitungen veranlassen. Letztendlich lassen sich die Stapeldateien mit einem kleinen Programm interaktionsfähig gestalten, so daß etwa in einem Menüsystem die gewünschte Befehlsfolge per Tastendruck auszuwählen ist (Listing 1).

Eine »autoexec.bat« verhindert nicht nur die Standardabfrage nach Zeit und Datum

Das Grundprinzip der Stapeldateien besteht darin, daß das Betriebssystem Dateien mit der Namensergänzung »bat« so ausführt, als ob jede Zeile ein eigener Befehl wäre. Somit können Sie eine Reihe von verschiedenen Befehlen untereinander schreiben, die dann nacheinander abgearbeitet werden. Leerzeilen werden ignoriert – MS-DOS sendet dazu weder Fehlermeldungen noch unternimmt es Ausführungsversuche. Der Name, den Sie für eine Stapeldatei vergeben, ist nicht ganz belanglos – bei doppelt abgelegten Dateinamen kommt es auf die Namensergänzung an. Und schließlich gibt noch es eine automatisch ausgeführte Startdatei, die den vorgeschriebenen Namen »autoexec.bat« tragen muß, da sie sonst nicht ihrer Bestimmung gemäß ausgeführt werden kann. Diese Start-

datei muß nicht zwingend vorhanden sein, doch dient sie als willkommene Anlaufstelle, um entweder Einstellungen für die Sitzung am PC vorzunehmen oder eine Anwendung beim Systemstart gleich mitzuladen. Diese Datei verhindert auf Wunsch auch die von MS-DOS automatisch ausgeführte Abfrage von Datum und Uhrzeit.

MS-DOS kennt interne Befehle wie »dir« oder »type«, die der Kommando-Interpreter ohne weitere Umstände ausführt, und externe Befehle, die es erst in einem Verzeichnis suchen muß. Wenn Sie dem Interpreter Zeichen übergeben, überprüft er zunächst, ob die Zeichenfolge einen internen Befehl zur Ausführung auffordert; ist dies nicht der Fall – etwa bei der Eingabe von

```
xcopy A: \ *.* B: /S
```

wird im aktuellen Verzeichnis oder in einem Verzeichnis, das in der Path-Direktive bekanntgemacht wurde, nach einer gleichlautenden Datei mit dem Namenszusatz »com« gesucht. Ist auch davon nichts zu entdecken, wird nach Dateien mit der Endung »exe« geforscht. Findet MS-DOS eine Datei namens »xcopy.exe«, wird diese geladen und der Eingabezusatz »A: \ *.* B: /S« an dieses Programm weitergegeben.

Sollte auch keine entsprechende Exe-Datei vorhanden sein, beginnt die Suche nach Bat-Dateien. Bei einem Treffer arbeitet der Kommando-Interpreter nun Zeile für Zeile der Datei nacheinander ab. Die beschriebenen Zeilen müssen demzufolge wieder einen Befehl enthalten, den MS-DOS behandeln kann – andernfalls wird am Bildschirm die wenig erfreuliche Meldung ausgegeben:

Befehl oder Dateiname falsch

oder, falls die Parameter unzutreffend sind, ein entsprechender Hinweis.

Die Vorgehensweise des Betriebssystems interessiert zwar an sich weniger, doch gibt die Reihenfolge einen Hinweis für die Namensgebung: Sollten Sie eine Stapeldatei mit demselben (Vor-) Namen wie eine der zunächst gesuchten Dateien benannt

Eingabe	Meldung	Zusatzeingabe
C:\BAT> del A: *.*	Sind Sie sicher (J/N)? Datei nicht gefunden	[j] [Enter]
C:\BAT> del A: *.* < jenter	Sind Sie sicher (J/N)? Datei nicht gefunden	
C:\BAT> del A: *.* < jenter > nul	Datei nicht gefunden	

Bild 1. Durch Befehlsumleitungen werden Stapeldateien komfortabel und übersichtlich

haben, wird diese nur dann ausgeführt, wenn Sie den Aufruf mit vollem Namen vornehmen. Um beim gewählten Beispiel zu bleiben, müßte »xcopy« mit

```
xcopy.bat A:\*.* B:/s
```

eingegeben werden, woraufhin eine so benannte Stapeldatei abgearbeitet würde.

Der übliche Einsatz von Stapeldateien ist vielfältig: Sie können etwa einen englischen Befehl eindeutschend oder abkürzen, ganze Befehlsfolgen mit einem einzigen, kleinen Befehl auslösen, Stapelprogramme verketteten, Funktionstasten mit speziellen Aufgaben belegen, die Bereitschaftsanzeige von MS-DOS mit wichtigen Informationen anreichern, eine Menüauswahl anbieten, Ihren Drucker steuern und vieles mehr. Ein Auszug aus dieser Vielfalt wird im folgenden vorgestellt.

Flexible Stapeldateien mit Parametern: Korrekte Abarbeitung erst mit aktuellem Aufruf

Vorher aber noch ein Wort zu den Parametern. MS-DOS gestattet einigen seiner Befehle, eine ganze Reihe von Parametern mitzuführen. Selbst »fremden« Programmen, zum Beispiel jeder beliebigen Anwendersoftware, kann MS-DOS Parameter mit auf den Weg geben, um eine genauere Spezifikation zu erreichen. So können Sie Word mit dem Parameter

```
word /l
```

starten oder ein Verzeichnis mit

```
dir B:*.com /w
```

selektiv einsehen. Mit Hilfe von Parametern können Sie also denselben Befehl auf verschiedene Daten anwenden, ohne die Datei jedesmal neu editieren zu müssen. Ihre Batch-Dateien erhalten dieselben Fähigkeiten durch die sogenannte auswechselbare Parameterform. Ein auswechselbarer Parameter (auch Blindparameter genannt) ist ein spezielles Symbol, das aus einem Prozentzeichen und einer einzelnen Ziffer, zum Beispiel »%1«, besteht. Folglich können Sie zehn dieser Parameter einsetzen. Während der Ausführung wechselt MS-DOS dann das Symbol durch einen Parameter aus, den Sie bei Aufruf der Stapeldatei festlegen müssen. Naheliegenderweise repräsentiert »%1« also den zuerst eingegebenen Parameter, »%2« den zweiten und so weiter.

Das folgende Beispiel, das Sie auch in weitere Tricks der Batch-Verarbeitung einführen soll, löscht nicht nur

das Stammverzeichnis Ihrer Disketten, sondern auch ganze Unterverzeichnisse einschließlich weiterer Unterverzeichnisse. Da die Batch-Datei nicht wissen kann, wie das Unterverzeichnis der soeben eingelegten Diskette heißt, müssen Sie dessen Namen beim Aufruf mitübergeben:

```
löschr textdat seribrie
```

veranlaßt die Stapeldatei namens »löschr.bat«, das Unterverzeichnis »textdat« und dessen Unterverzeichnis »seribrie« zu eliminieren, obwohl diese Verzeichnisnamen nur als »%1« und »%2« in der Stapeldatei vorkommen (Listing 2).

Nun aber zum eigentlichen Gegenstand dieser Batch-Datei. Die Aufgabe besteht beispielsweise darin: Sie besitzen ein Diskettenlaufwerk, und es ist fast stündlich erforderlich, gebrauchte Disketten schnell wieder von Dateien zu befreien. Sie könnten dies beispielsweise durch den Befehl

```
del A:*.*
```

Sind Sie sicher (J/N)?

```
[J]
```

realisieren, den Sie jedesmal bei eingelegter Diskette übergeben. Mit [F3] könnten Sie zwar den Löschbefehl einfach wiederholen, doch das Ärgertliche daran ist das Warten auf die Sicherheitsmeldung »Sind Sie sicher (J/N)?«, die mit zweimaligem Tastendrücken – [J] und [Enter] – beantwortet werden muß. Bei vorhandenen Unterverzeichnissen müßten Sie dazu jedesmal diese mit angeben, dort alle Dateien löschen, das Unterverzeichnis löschen, etc. Fraglich ist dann jedoch immer noch, ob tatsächlich alle Dateien von der Diskette gelöscht wurden oder ob nicht vielleicht doch noch ein gefülltes Unterverzeichnis dem Löschvorgang widerstanden hat. Eine kleine Batch-Datei »löschr.bat« (Listing 2) nimmt Ihnen diese Arbeit ab.

Numerische Parameter gibt es nur zehn, benannte Parameter können Sie beliebig viele einsetzen

Dabei nützen Sie nicht nur den Effekt, daß die Stapeldatei Parameter übernimmt, sondern auch, daß MS-DOS in der Batch-Verarbeitung ebenfalls eine Eingabeumleitung (< » und > ») erkennt. Die Umleitung nach »nul« verhindert beispielsweise die Meldungs Ausgabe am Bildschirm, die Eingabe von »jenter« erspart das Drücken der Tasten [J] und [Enter] für die Sicherheitsabfrage (Bild 1). Allerdings müssen Sie zuvor eine kleine Datei dieses Namens anlegen, in der diese Tastenkombination gespeichert

ist. Ferner muß diese Datei im gleichen Verzeichnis abgelegt sein, in dem sich auch die Datei »löschr.bat« befindet.

Wenn Sie Listing 2 näher betrachten, dann fallen Ihnen bestimmt viele Befehle auf, die Sie weder von der MS-DOS-Diskette noch vom täglichen Umgang mit dem Prompt-Zeichen her kennen.

Zusätzlich zu den zehn numerischen (oder auswechselbaren) Parametern sind auch sogenannte benannte Parameter gestattet. Wenn ein benannter Parameter auf eine Datei oder Umgebungsvariable bezogen ist, wird sein Name zwischen zwei Prozentzeichen gestellt. Die Werte dieser Namensparameter werden nicht beim Aufruf festgelegt, sondern MS-DOS ruft den Wert jedes benannten Parameters von seiner »Umgebung« ab. Die Größe der Umgebung legen Sie ab Version 3.2 in einer Zeile von »config.sys« fest, und zwar mit

```
shell=C:\command.com /e:4000 /p
```

falls C das Startlaufwerk ist.

Ein Beispiel, das den Einsatz der Umgebungsvariablen zeigt: Wollen Sie etwa nur einmal wöchentlich eine Programmdatei starten, und zwar montags, dann halten Sie einfach Ihr Systemdatum aktuell und fügen diese Zeile in Ihre automatische Startdatei ein:

```
echo | more | date > readdate.bat
```

Die beim Date-Befehl übliche Meldung wird folglich in die Datei »readdate.bat« umgeleitet. Die Zeilenschaltung übernimmt »more.com«. Das Filterzeichen zwischen den Befehlen erzeugen Sie mit [Alt-124]. Falls diese Batch-Datei nun aufgerufen wird, also mit

```
readdate
```

sucht MS-DOS nach einer Datei namens »aktuelle«. Einer Batch-Datei mit dem Namen »aktuelle« geben Sie den Inhalt

```
echo off
if %datum%==.Mon goto end
set datum=%3
if not %datum%==.Mon goto end
woechtl.com
:end
```

wobei hiermit das vierte Wort der Date-Meldung den Wochentag anzeigt. Mit diesen beiden Stapeldateien und der ausgeführten Zeile in der Autoexec-Datei wird das Programm »woechtl.com« nur einmal die Woche aufrufbar – und das nur am Montag. In der Umgebungsvariablen »datum«, die MS-DOS die ganze Sitzung für Sie aufbewahrt, ist nämlich nach dem ersten Aufruf von »aktuelle.bat« das Tagesdatum eingeflochten. Beim zweiten Aufruf dieser Stapeldatei ist der benannte Parameter »%datum%« festgelegt als »Mon«, »Die« und so weiter – jeder andere Wochentag als

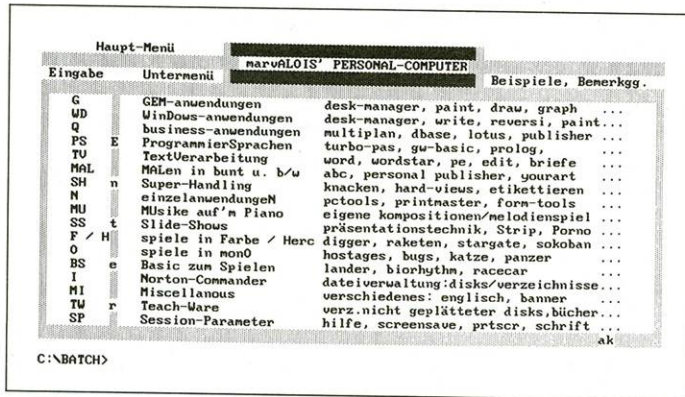


Bild 2. Bei langen Texten wie diesem bietet sich statt zeilenweisen Echos eher eine separate Textdatei an

Montag wird durch die zweite If-Abfrage ohnedies ausgesiebt. Haben Sie in Dateinamen schon ein Prozentzeichen eingebracht, dann ist es für die korrekte Abarbeitung der Stapelbefehle unerlässlich, dieses Zeichen zu verdoppeln

echo Kopiere Datei »alpha%1%1.lbm«
meldet, daß die Datei »alpha%1.lbm«
kopiert wird.

Tips und Tricks zu Batch-Dateien: Stapel ruft Stapel und so weiter

Listing 3 zeigt noch eine weitere Besonderheit, mit der Stapeldateien miteinander umgehen – zumindest bis einschließlich MS-DOS 3.2: Ruft eine Stapeldatei eine andere auf, ist eine »automatische« Rückkehr zur ersten nicht mehr möglich. Zwar wird während des Aufrufs von »batch1.bat«

```
...
echo hier endet die 1. Batch-Datei
batch2.bat
```

die Abarbeitung der Befehle vom ersten Stapel an »batch2« übergeben, doch damit beendet die erste Datei auch ihr Kommandodasein, das heißt, eventuell folgende Stapelbefehle in »batch1.bat« werden ignoriert. Nicht so unter MS-DOS 3.3, wenn Sie den Aufruf eines »untergeordneten« Befehlsstapels mit dem Kommando »call Stapeldatei Parameter« einleiten:

```
echo hier endet die 1. Batch-Datei
call batch2.bat
echo MS-DOS 3.3 arbeitet hier weiter ...
```

»batch2.bat« beendet hier also nicht den Stapel der aufrufenden Datei. Unter MS-DOS 3.2 und darunter verhilft jedoch ein einfacher Trick, eine solche Befehlsfolge nicht durch die untergeordnete Batch-Datei unterbrechen zu lassen. Sie rufen am Ende der untergeordneten Stapeldatei einfach wieder die aufrufende Stapeldatei auf, denn MS-DOS kann sehr wohl eine Batch-Datei nach der anderen aktivieren, aber eben »von selbst«

nicht zur rufenden Batch-Datei zu rückspringen.

Nun besteht aber nach wie vor ein Problem, denn die Verarbeitung des übergeordneten Stapels soll ja nicht wieder von vorne beginnen. Sie soll statt dessen an der Stelle fortgesetzt werden, die dem Aufruf des untergeordneten Stapels folgt. Hier helfen nur korrekt gesetzte Marken, diesbezügliche Parameter und deren Abfrage, um den Kommando-Interpreter am entsprechenden Befehl wieder einsetzen zu lassen. Die übergeordnete Datei »batch1.bat« sieht dann etwa so aus:

```
if "%2"=="weiter" goto weiter
...
batch2.bat
:weiter
echo hier setzt der Kommandointerpreter
echo ein, wenn »batch2.bat« mit einem
echo treffenden Aufruf versehen wird
...
```

In »batch2.bat« wird nun als »Ausgangszeile« der Wiederaufruf von »batch1.bat« in folgender Weise gestaltet:

```
batch1 batch1 weiter
»batch2.bat« ist damit vollständig als
Unterprogramm von »batch1.bat«
realisiert. »batch1.bat« wird, unge-
achtet der Tatsache, daß es zwischen-
durch eine andere Batch-Datei auf-
ruft, vollständig und einmalig abge-
arbeitet.
```

Call, Command oder Konstruktionen – alle rufen einen Unterprogramm-Stapel

Eine andere Version, Stapelprogramme so miteinander zu verbinden, daß ein Wiederaufruf des ersten Programms erfolgt, besteht darin, dem zweiten Programm einen eigenen Kommandoprozessor zuzuweisen. Das funktioniert ungefähr folgendermaßen:

```
echo hier ist Stapeldatei 1
command /c stapel2.bat
echo hier ist wieder Stapeldatei 1
```

Sie rufen also nicht nur eine zweite Stapeldatei auf, sondern erst einmal ein neues Exemplar des Kommandoprozessors, der sich an die Abarbeitung von »stapel2.bat« macht. Der Parameter »/c« läßt nur die Ausführung eines einzigen Befehls (in diesem Fall ist dies ein zusammengefaßter Befehlsstapel) zu, so daß nach Bearbeitung von »stapel2.bat« eine Rückkehr zur vorherigen Kopie von »command.com« erfolgt.

Allerdings kann dieses Vorgehen leicht einen Systemabsturz zur Folge haben, sofern residente Programme durch den zweiten Kommandoprozessor geladen wurden. Nach Beendigung dieses Stapels gibt zwar der zweite »command.com« seinen Speicher wieder frei, nicht aber das residente Programm, das quasi eine Lücke in die Belegung des Arbeitsspeichers reißt.

Batch-Dateien gehen übrigens viel schneller vonstatten, wenn Sie diese auf ein virtuelles Laufwerk, also eine RAM-Disk, legen. Natürlich ist dies nur empfehlenswert, wenn Sie diese Datei (und alles, was mit ihr zu tun hat, also etwa die Texte, Filter, Umleitungs- und Programmdateien) erst in dieses Laufwerk kopieren. Eine kleine RAM-Disk, gegebenenfalls im Speicherbereich oberhalb von 640 KByte, dürfte genügen, um Ihre Arbeit mit Stapeldateien erfolgreich zu gestalten.

Nicht nur in Stapeldateien ist es oft sinnvoll, die Ein- oder Ausgabedaten außer zur Standardeinheit auch auf den Drucker, in eine Datei oder in eine »virtuelle Einheit« (nul) umzuleiten. Solche Transaktionen wurden hier teilweise schon verwendet, ohne jedesmal ihre Bedeutung zu erwähnen:

- Das Zeichen »>« weist die Ausgaben auf das nachstehende »Gerät«.
- Ist damit eine Datei bezeichnet, wird diese neu angelegt, selbst wenn sie schon existiert.
- Mit dem Zeichen »<« erhält ein vorstehendes Programm seine Eingaben von der angegebenen Einheit – sinnvoll ist hier vor allem eine Datei.
- Mit den Zeichen »>>« werden die Programmausgaben an die zuletzt erfolgte Ausgabe angehängt. Wollen Sie etwa wissen, wann Ihr PC eingeschaltet wurde, dann führt folgende Batch-Datei, die Sie in die Autoexec-Datei einfügen, bei batteriegepufferten Systemen für Sie das Logbuch:

```
path C: \; \dos \; \bat
echo System eingeschaltet am: >>
log.dat
date | find "Aktuelles" >> log.dat <
cr.dat
time | find "Aktuelle" >> log.dat <
cr.dat
```

»cr.dat« ist dabei eine kleine Datei, die nur ein Return-Zeichen lang ist

und die dieses Return am angegebenen Ort durch die »Einleitung« (<) in das voranstehende Programm einspeist. Natürlich müssen Sie auch diese Batch-Befehle an Ihre MS-DOS-Version anpassen. So erscheint etwa auf Peacock-ATs im Time-Befehl anstelle von »Aktuelle Uhrzeit:« nur die Meldung »Uhrzeit:«.

Eine Stapeldatei zusammenzustellen, ist nicht weiter schwer. Dies gelingt mit den meisten Editoren und Textverarbeitungen, Sie müssen lediglich darauf achten, die Datei als ASCII-Datei – ohne jegliche Formatierungen – auf dem Datenträger zu speichern. Natürlich eignet sich dazu auch der Zeileneditor Edlin von MS-DOS oder aber – falls die Datei nicht zu umfangreich ausfällt – ein Kopiermechanismus namens »copy con«, der alle Tastatureingaben in eine Datei umleitet, die im folgenden Beispiel »stapel.bat« heißen soll.

Das Zusammenstellen einer Stapeldatei erfordert nicht immer einen ASCII-Editor

Diese Sonderform des Copy-Befehls hat den Vorteil, daß Sie nicht erst umständlich einen geeigneten Editor starten müssen, sondern er steht, ebenso wie die anderen internen MS-DOS-Befehle, allzeit bereit: Sie müssen dabei allerdings den Text zeilenweise an den PC übergeben, und Korrekturen sind nur innerhalb der aktuellen Zeile ausführbar. Sie müssen auch jede Befehlszeile mit [Enter] abschließen; ist die Zeile so beendet, haben Sie auch keine Korrekturmöglichkeit mehr. Sollte sich also in einer bereits abgeschlossenen Zeile ein Fehler eingeschlichen haben, so müssen Sie nach einem Abbruch mit [Ctrl-C] vollständig neu mit der Eingabe beginnen. Ist Ihre Stapeldatei vollständig, löst [Ctrl-Z] (oder [F6]) den Kopiervorgang aus, so daß diese Kommandofolge nun zur Verfügung steht.

```
copy con stapel.bat
rem demonstration einer Kopie
rem von der Tastatur
echo off
cls
type menue.txt
prompt $p$g
uhr.com
[Ctrl-Z]
```

Jede Zeile ist, wie erwähnt, mit [Enter] zu beenden. Die Meldung »1 Datei(en) kopiert« bestätigt dann, daß Ihre Zeilen heil auf der Platte angekommen sind. Wichtig für das Zusammenbinden von Stapeldateien ist hier, daß Sie selbst die letzte Zeile noch mit einem [Enter] abschließen.

MS-DOS erkennt den Befehl sonst nicht als vollständig übergeben an.

Um eine Stapeldatei nicht nur per Variable flexibel zu schreiben, sondern den Ablauf vom Benutzer aktuell steuern zu lassen, gibt es verschiedene Verfahren:

Ein Menüsystem auf Basis der Stapelverarbeitung: Ein Programm nur mit dem Anfangsbuchstaben laden

– Es werden für die in Frage kommenden Eventualitäten jeweils spezielle Stapeldateien geschrieben. Diese Vorgehensweise ist in der vorgestellten Menüanwendung verwirklicht. Ein Vorteil ist etwa die leichtere Pflege der Kommandodateien bei auftretenden Veränderungen (Sie wollen von »Word« auf »PC-Write« umsteigen und müssen dies Ihrer Batch-Datei mitteilen), Nachteile sind der vermehrte Speicherplatzbedarf durch viele kleine Dateien und eine umständlichere Bedienung gegenüber dem zweiten Verfahren.

– Eine Alternative dazu verlangt eine spezielle Programmdatei, die die Interaktionen dem Betriebssystem bekanntgibt und es zwingt, die daraus abgeleitete Stapelfolge in Angriff zu nehmen. Auf diese Weise genügen eine oder wenige Stapeldateien, um mehrere Aktionen abzudecken.

In Ihrer Batch-Datei zur Druckersteuerung wollen Sie etwa jeweils wahlweise Zugriff auf den deutschen oder den amerikanischen Zeichensatz haben, je nachdem, welcher bereits eingeschaltet ist. Listing 3 zeigt, wie sich dies unter MS-DOS 3.2 geschickt lösen läßt: als Stapelprogramm, mit dem Sie je nach Tastendruck amerikanisch oder deutsch drucken können.

Verwenden Sie als Steuerprogramm entweder »ask.exe« aus dem Norton-Integrator-Paket oder »taste.com«, das Sie gemäß Listing 1 mit dem MS-

DOS-Hilfsprogramm »Debug« eingeben.

Für die Darstellung des Menütextes stehen zwei Verfahren zur Wahl. Sie geben den Text durch die Echo-Befehle auf dem Bildschirm wieder, oder Sie holen mit »type Dateiname« einen Menütext auf den Monitor. Das erste Verfahren eignet sich bei kurzen Texten (Listing 3) und bei schnellen Computern, das zweite bei langen Texten (Bild 2) und bei PCs mit gemächlicherer Gangart.

Falls Sie Ihre gesamte Festplatte mit Batch-Dateien verwalten, ist es nicht sinnvoll, mit nur einer einzigen Batch-Datei auskommen zu wollen. Mehrere kleinere Stapeldateien sind dafür prädestiniert, diese Aufgabe zu übernehmen. Stellen Sie einer solchen Datei nur eine Aufgabe – etwa die Diskettenpflege mit Formatieren, Diskette löschen, Disketten kopieren (Bild 2) oder die Druckersteuerung (Listing 4).

Die Methode der Textausgabe in Stapeldateien: »Echo« kontra »type«

Auch der Menütext selbst kann sehr ansprechend gestaltet werden – einsteigergerichtet (Bild 2) oder orientiert an professionellen Vorbildern (Bild 3). Jedenfalls muß der Aufruf mitangegeben sein (und natürlich die entsprechenden Batch-Dateien) oder, falls Sie einen interaktiven Batch vom Stapel lassen, die Tasten, die entsprechende Unterstapeldateien aktivieren: Für Bild 3 muß etwa eine Batch-Datei namens »mp.bat« vorhanden sein, die in der Lage ist, Multiplan zu starten und wieder zu diesem Menü zurückzukehren. Für Bild 4 genügt eine Stapeldatei, die die aufgezeichneten Tasten abfragt und weiterbehandelt.

Sollten Sie ein hierarchisches Menüsystem aufgebaut haben, dann müssen Sie auch in der Batch-Datei vermerken, wie die nächsthöhergelegenen Menüs erreichbar sind – bei-

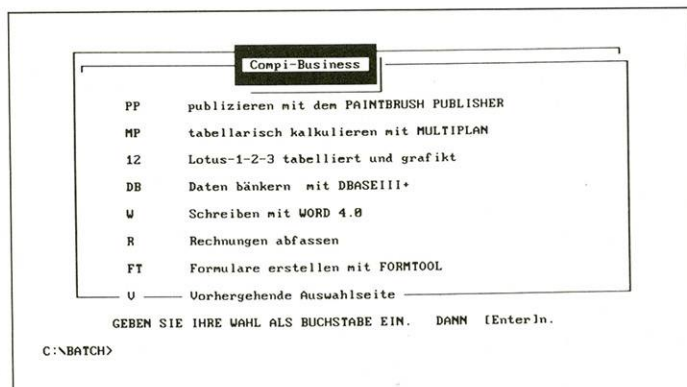


Bild 3. Schlicht, doch professionell wirkt dieses Menü

Druckersteuermenu			
ZEICHENSATZE	ZEILENABSTAND	DIVERSES	BUCHSTABENGRÖßE
[a] Standard [b] IBM [c] IBM #1 [d] IBM #2	(Standard 1/6) [k] 1/8 inch [l] 7/32 inch [m] 1/4 inch [n] 1/2 inch	[r] Papierende- sensor ein [s] PES aus	[v] doppelt hoch [w] vierf. hoch [x] löschen [y] doppelt hoch, einfach breit [z] löschen
DRUCKFORMEN	FORMATIEREN	DIPS	
[e] Breit ein [f] aus [g] Schatten ei [h] aus [i] Fett ein [j] aus	[o] linksbündig [p] zentriert [q] rechtsbündig	[t] Wagenrück- lauf mit ZU [u] UR ohne Zeilenvor- schub (ZU)	[1] ENDE [2] INFO [3] TEST

Bild 4. Mit »taste.com« werden Ihre Wünsche auf Tastendruck Wirklichkeit

spielsweise »v« für das in einer Baumstruktur vorhergehende Menü (Bild 3). Und wenn Sie noch innerhalb des Pfades (am besten in »\bat«) eine Datei plazieren, die das Hauptmenü anbringt, dann haben Sie für alle Fälle vorgesorgt. Zur Namensgebung bietet sich »v.bat« an:

```
echo off
cd \bat
cls
type menu.txt
prompt $p$g
```

Von hier aus können Sie sich jederzeit wieder durch das Menü an die Stelle arbeiten, an der Ihre Batch-Datei (noch) versagt hat oder an der Sie diese aus den Augen verloren haben. Die Batch-Datei kann sich auch selbst aus den Augen verlieren, »Batch Datei fehlt« heißt die entsprechende Fehlermeldung.

Nun haben Batch-Dateien und Menütexte einen Nachteil: Sie sind oft nicht gerade aktuell. Gerade in der Einsteigerphase wird die Festplatte mit allerlei Programmen angefüllt – so ist man es oft leid, alle neuen Programme auch im Menütext zu vermerken. Beispielsweise nimmt das Verzeichnis »\Spiel« rasend schnell zu. Um trotzdem den Überblick zu behalten, empfiehlt sich eine Auswahlliste mit dem Dir-Befehl oder mit Programmen aus der Public-Domain-Szene zusammenzustellen, die eine kompaktere Auflistung gestatten (Bild 5). Mit folgender Stapeldatei ist Ihnen vielleicht auch geholfen, um eine aktuelle Auswahl zu treffen:

```
echo off
cls
cd \spiel
prompt [Alt-255]
echo Es wird umgeleitet ...
echo Hier sind die abgelegten Spiele: >
spiel.dat
for %I in (*.exe, *.com, *.bat) do echo
%I >> spiel.dat
echo Lade Dir ein Spiel >> spiel.dat
cls
type spiel.dat | more
prompt $p$g
```

Dieser Vorschlag ist zwar sicherlich nicht die allerschnellste Lösung, aber auf ATs haben Sie trotzdem in Se-

kunden eine aktuelle Liste, die aufgrund des More-Befehls mehr als 23 Vorschläge beinhalten darf, ohne unübersichtlich zu sein. Bei weniger als 23 Spieldateien empfiehlt sich keine langwierige Umleitung und damit auch kein CIs- und Type-Befehl. Durch das zwischenzeitliche Unsichtbarmachen des Kommando-Prompts erhalten Sie eine tadellose Liste, die im For-Befehl mit dem Datei-Jokerzeichen »*« arbeitet.

Sehr sinnvoll ist es, wenn Ihnen das Hauptmenü sofort nach dem Einschalten des PC zur Verfügung steht – ohne es jedesmal umständlich aufzurufen. Sorgen Sie einfach dafür, daß diese Aufgabe die automatische Startdatei »autoexec.bat« übernimmt: Angenommen, Sie laden nur den deutschen Tastatortreiber und haben Ihre sämtlichen Stapeldateien (samt Zubehör, das heißt einschließlich der Menütexte) im Unterverzeichnis »\bat« untergebracht, dann sieht Ihre »autoexec.bat«, die Sie natürlich im Stammverzeichnis des Startlaufwerkes angelegt haben, so aus:

```
echo off
keybgr > nul
path C: \; \bat; \util
cd \bat
cls
type hauptmenu.txt
prompt $p$g
```

Um die Meldungen, die MS-DOS gewöhnlich ausgibt, zu unterdrücken, ist die Umleitung in ein Dummy-Gerät nützlich. Diesen, besonders in Stapelarbeiten oft erwünschten Effekt, erzeugen Sie mit dem Größerzeichen, das auf das scheinbare Gerät »nul« zeigt. Der Pfad ist auf das Stammverzeichnis und auf die Unterverzeichnisse »\bat« und »\util« gerichtet. Das Hauptmenü wird auf den Bildschirm gebracht und der Kommando-Prompt mit dem aktuellen Verzeichnis ausgegeben.

Batch-Dateien haben die unangenehme Eigenschaft, nicht unter allen Umständen vollständig ausgeführt zu werden. So zum Beispiel beim Schneider-MS-DOS 3.2 auch die Stapeldatei »a.bat«, die auf Laufwerk A

umschalten und das dortige Verzeichnis gleich mit ausgeben soll:

```
echo off
cls
A:
dir
```

Rufen Sie diese Befehlsfolge aus dem Verzeichnis auf, in dem sie abgelegt ist (»\bat«), erhalten Sie auch richtig das erwartete Ergebnis Ihrer Bemühungen. Aus dem Verzeichnis »\msdos« aufgerufen, beunruhigt Sie hingegen die Meldung »Batch Datei nicht gefunden«, nachdem eben gerade diese Batch-Datei auf Laufwerk A umgeschaltet hat – allerdings wird kein Verzeichnis mehr ausgegeben.

Eine anderes Ablaufhindernis kann bei der Verwendung von Diskettenlaufwerken eintreten. Wenn Sie während der Verarbeitung einer Stapeldatei die Diskette, auf der sich die Datei befindet, aus dem Laufwerk nehmen, wird Ihnen mitgeteilt, daß die Weiterverarbeitung davon abhängt, ob Sie die entsprechende Diskette wieder in das Laufwerk zurücklegen.

»taste.com« informiert den Stapel darüber, welche Taste Sie gedrückt haben

Nun kennen Sie aber immer noch kein Programm, das es zuläßt, eine Stapeldatei auf einfachen Knopfdruck hin zu steuern. Das erledigt das Programm »taste.com«. Es löscht den Tastaturpuffer, wartet auf ein Zeichen von der Tastatur und übergibt den ASCII-Wert dieses Zeichens als Wert für den Errorlevel-Parameter an die Stapeldatei, aus der »taste.com« aufgerufen wurde. Auf diese Weise leitet etwa die Taste 1 einen Errorlevel von 49 weiter, den Sie mit dem Kommando »if errorlevel 49 Befehl« dann auch »weiterbefehlen« können (Listing 3).

»Errorlevels« sind also keine Fehler, sondern Codes, die ein Programm dem Betriebssystem übergibt. Diese Codes lassen dann eine bedingte Weiterverarbeitung zu (mit dem Befehl »if errors level...«).

Mit dem System-Utility »debug.com« können Sie sich diese Datei aneignen. Entweder Sie erzeugen eine Textdatei gemäß Listing 1 – wobei die letzte Zeile mit [Enter] abgeschlossen werden muß – und kopieren diese Datei in Ihr MS-DOS-Verzeichnis. Die Befehlszeile

```
debug < taste.deb
```

macht dann im Handumdrehen die gewünschte »taste.com« daraus. Oder Sie »debuggen« interaktiv nach folgendem Muster: Den Aufruf

debug
schließen Sie mit [Enter] ab, daraufhin präsentiert sich das Minuszeichen als Bereitschaftsanzeige. Falls Sie dieses Programm nicht auf Ihrer Festplatte finden, legen Sie die entsprechende Betriebssystemdiskette ins Laufwerk und schalten darauf um. Ihre weiteren Eingaben sind:

```
n taste.com
r cx
0009
a
mov ax,0c08
int 21
mov ah,4c
int 21
[Enter]
w
q
```

Auch hier beschließen Sie wieder jede Zeile mit dem gewohnten [Enter]. Wenn Sie jetzt das Verzeichnis Ihrer Dateien auflisten, finden Sie darin die neue Datei »taste.com«, die ganze neun Byte lang ist. Kopieren Sie sich diese Datei in Ihr Verzeichnis »\bat«, von hier aus kann sie häufig eingesetzt werden. Arbeiten Sie mit »Scheck«, einer Eingabehilfe für Listings aus der DOS International, dann sind Sie auch in der Lage, folgenden einzeiligen Hexdump zur Lauffähigkeit zu bewegen:

```
[C53] 1: B8 08 0C CD 21 B4 4C CD 21
```

In den Listings 3 und 4 ist exemplarisch vorgeführt, wie diese neue Datei sinnvoll eingesetzt werden kann: »taste.com« liefert die Errorlevels, die Sie unbedingt in absteigender Reihenfolge abfragen müssen, weil sonst eine einwandfreie Funktion der Programmsteuerung nicht gewährleistet ist. Da Falscheingaben nicht abgefangen werden, sind auch andere Tasten nicht folgenlos: Alle kleinen Buchstaben von e bis u geben in Listing 3 den deutschen Zeichensatz zurück – die Zahlen oder Großbuchstaben bauen nur das Menü neu auf – eine ASCII-Tabelle klärt Sie schnell über die Hintergründe auf. Natürlich könnten Sie die nicht vom Menü festgesetzten Tasten auch durch weitere Abfragen in den Griff bekommen.

if errorlevel 101 goto weiter
würde – an entsprechender Stelle eingeordnet – etwa bewirken, daß Kleinbuchstaben von e bis u keine vielleicht unerwünschten Schalter setzten könnten.

Die Steuerung des Druckers ist nicht nur für Einsteiger eine immer wiederkehrende Aufgabe. Mit der Stapelsprache von MS-DOS haben Sie auch dieses Problem im Griff: Durch Umleitung auf das Gerät »prn« können beispielsweise die Echo-Texte an den Drucker geleitet werden. Und wenn Sie diese Texte geschickterweise mit Steuerzeichen füllen, dann sind die Tage gezählt, an denen Sie sich stundenlang mit der Druckereinstellung beschäftigten.

```
echo [Alt-1-2] > prn
```

sendet etwa einen Seitenvorschub an den Drucker, wenn Sie das ASCII-Zeichen [Alt-12] oder [Ctrl-L] in Ihrem Editor unterbringen können.

Eine Druckersteuerung als Stapeldatei: Fettdruck auf Tastendruck

Die Methode, die Sie in Listing 4 sehen, bedient sich allerdings einer Datei »pr.com«, die den Einsatz von leichten Steuerzeichen umgeht, indem nur deren ASCII-Code anzugeben ist (Listing 5).

Ein wichtiges Kriterium für eine gute Druckersteuerung ist die Anpassung an verschiedene Drucker – »druck.bat« arbeitet mit jedem Drucker zusammen. Das Programm besteht aus den Dateien »druck.bat« (Hauptprogramm, Listing 4), »schirm.txt« (Menübild, Bild 4), »dructest.txt« (dient der sofortigen Erfolgskontrolle), »druck.txt« (zeigt die verwendeten Steuersequenzen), »taste.com« (fragt den ASCII-Wert der gedrückten Taste ab, Listing 1) und »pr.com« (Public-Domain-Software, die die Steuersequenzen an den Drucker sendet, Listing 6). Nach dem Aufruf der Batch-Datei wird das Echo ausgeschaltet und mit »anfang« eine Marke für die Schleifenbildung gesetzt. Es ist wichtig, daß der Befehl »cls« nach der Zeile »anfang« steht; hierdurch ist gewährleistet, daß der Bildschirm keine unerwünschten Überreste nach einer Verzweigung anzeigt. Nun wird mit

```
type schirm.txt lmore
```

das Menübild aufgerufen. Durch die Verwendung des Filters »more«, ist es möglich, auch ein Menü zu verwenden, das länger als eine Seite ist. »more.com« oder »more.exe« (je nach MS-DOS-Version) muß sich im Zugriffspfad befinden. Durch das Programm »taste.com« wird der ASCII-

Wert der ersten gedrückten Taste als Errorlevel an das Betriebssystem zurückgegeben. Die Errorlevels werden abgefragt, das Programm verzweigt zu den entsprechend angegebenen Sprungmarken.

Sie müssen bei Erweiterungen des Programms unbedingt darauf achten, daß die Errorlevels – wie bereits erwähnt – in absteigender Reihenfolge abgefragt werden (eine Einschränkung von MS-DOS). Für die Anordnung der Sprungmarken (»:name«) gilt das nicht, hier ist die Reihenfolge beliebig.

Für die nach Eingabe einer [3] aufgerufene Datei »dructest.txt« können Sie natürlich einen beliebigen Text verwenden. Bei Eingabe einer [2] erhalten Sie einen Überblick über alle verwendeten Drucker-codes.

Das Programm »pr.com« schickt beliebige Steuersequenzen an den Drucker. Es läßt sich natürlich auch, ohne in »druck.bat« eingebunden zu sein, direkt vom Betriebssystem aus einsetzen. Die Anpassung von »druck.bat« an Ihren Drucker ist, wie Sie sehen, kein Problem, alle Steuersequenzen finden Sie in Ihrem Druckerhandbuch.

Wichtig ist es jedoch, wenn Sie das Programm installieren, die korrekten Pfadnamen zu verwenden. Auch wenn die Dateien in einem Verzeichnis stehen, das im Path-Befehl angegeben wurde, wird es zu Fehlern kommen, da dieser Befehl des Betriebssystems nur für die Dateitypen »bat«, »com« und »exe« gilt. Sie schaffen Abhilfe, wenn Sie bei allen Type-Befehlen explizit den Pfad mit angeben – etwa:

```
type c:\drucker\schirm.txt
```

So können Sie aus jedem Verzeichnis heraus Ihren Drucker beliebig steuern.

Natürlich konnte diese Werkstatt noch nicht auf alle Eigenheiten, Tricks oder Ausbaumöglichkeiten von Stapelbefehlen eingehen. Doch Sie wissen jetzt, daß MS-DOS ein leistungsfähiger »Hochstapler« ist!

(Alexander Stürtz/al)

Bild 5. Aktuelle Auswahllisten erhalten Sie auch mit den Befehlen »dir /w *.exe« oder noch kompakteren Verzeichnisanzeigen (»ad.com«)

Ihre Wahl fiel auf Spiele auf'm Mono-Schirm:									
BASRUN	EXE	BUGS	EXE	DAME	EXE	EAU	EXE	GOBBLE	EXE
HOSTAGES	EXE	PITFALL	EXE	PONG	EXE	SKIING	EXE	SPIEL	EXE
WORM	EXE								
	11								
FLIEGEN	BAT	FROSCH	BAT	GOMOKU	BAT	OLIS	BAT	RAKETEN	BAT
REVERSI	BAT	SCHACH	BAT	SEESCHLA	BAT	STARGATE	BAT	TANK	BAT
	10								
=====									
Spielwahl: Vorname des Spiels und [Enter]-Taste klopfen.									
Will gar nicht gespielt sein, dann geradewegs eintippen: U [Enter].									
C:\SPIELE>									


```
[S02] 1: n taste.com
[8L0] 2: r cx
[9F0] 3: 0009
[240] 4: a
[ZU1] 5: mov ax,0c08
[320] 6: int 21
[3E1] 7: mov ah,4c
[320] 8: int 21
[000] 9:
[E50] 10: w
[250] 11: q
[000] 12:
```

Listing 1. »taste.deb« als Eingabelisting für den Debugger

```
[DA1] 1: echo off
[AU0] 2: :oben
[PN0] 3: cls
[ER6] 4: echo Gebrauchte Diskette in A: einlegen
[A97] 5: echo Zu löschende U-Verzeichnisse angeben
[VU4] 6: echo -----
[MW0] 7: pause
[8L3] 8: del A:.* < jenter > nul
[NT3] 9: if not %2==. goto loesch2
[Z32] 10: goto loesch1
[UC1] 11: :loesch2
[EV4] 12: del A:\%1\%2\.* < jenter > nul
[DL1] 13: rd A:\%1\%2
[TC1] 14: :loesch1
[CT3] 15: if not %1==. goto loesch0
[MZ1] 16: goto abgang
[SC1] 17: :loesch0
[LB4] 18: del A:\%1\.* < jenter > nul
[Y31] 19: rd A:\%1
[E81] 20: :abgang
[PN0] 21: cls
[LR6] 22: echo So sieht es nun auf der Diskette aus
[VZ0] 23: dir A:
[MP4] 24: echo -----
[3C6] 25: echo Abbrechen mit [Ctrl-C], ansonsten
[MW0] 26: pause
[GH1] 27: goto oben
```

Listing 2. »lösch.bat« ist eine Stapeldatei, die Ihre Disketten in Laufwerk A von Dateien und Unterverzeichnissen befreit

```
[DA1] 1: echo off
[HW2] 2: if %2==. goto erst
[H41] 3: goto %2
[U06] 4: rem beim Zweitaufwurf wird der Befehls-
[8N6] 5: rem ablauf bei »weiter« fortgesetzt
[6W0] 6: :erst
[PN0] 7: cls
[U66] 8: echo Erst das Schreibgerät anpowern.
[S54] 9: echo -----
[MW0] 10: pause
[NC1] 11: :weiter
[PN0] 12: cls
[8M7] 13: echo [d] deutschen Zeichensatz einschalten
[ZD7] 14: echo [a] amerikan. Zeichensatz einschalten
[8L6] 15: echo [w] Schriftattribute einschalten
[Z15] 16: echo [v] vorhergehendes Menü
[VU4] 17: echo -----
[XU7] 18: echo Die gewünschte Taste zu drücken genügt
[SU1] 19: \bat\taste
[9U4] 20: if errorlevel 119 goto attrib
[2L3] 21: if errorlevel 118 v.bat
[C25] 22: if errorlevel 100 goto deutsch
[7S4] 23: if errorlevel 97 goto amerik
[TK2] 24: dru dru weiter
[PP6] 25: rem Bei Tasten, deren ASCII-Code kleiner
[DW6] 26: rem als 97 ist wird durch obige Anweisung
[7N6] 27: rem ein »goto weiter« herbeigeführt
[YC1] 28: :deutsch
[PN0] 29: cls
[X67] 30: echo Deutscher Zeichensatz wird eingerenkt
[Y44] 31: \druck\setprn [27]R[1]
[TK2] 32: dru dru weiter
[VA1] 33: :amerik
[PN0] 34: cls
[147] 35: echo Amerikan. Zeichensatz wird eingerenkt
[W44] 36: \druck\setprn [27]R[0]
[TK2] 37: dru dru weiter
[2B1] 38: :attrib
[HA3] 39: \druck\druattr.exe
[TK2] 40: dru dru weiter
```

Listing 3. »dru.bat« ist eine Stapeldatei, die sich selbst wieder aufruft

```
[DA1] 1: echo off
[X91] 2: :anfang
[PN0] 3: cls
[FX2] 4: type schirm.txt
[779] 5: echo Bitte die Auswahl in Buchstaben oder Z
ahlen eingeben
[NW0] 6: taste
[VU3] 7: if errorlevel 122 goto z
[RU3] 8: if errorlevel 121 goto y
[LU3] 9: if errorlevel 120 goto x
[2V3] 10: if errorlevel 119 goto w
[VU3] 11: if errorlevel 118 goto v
[UU3] 12: if errorlevel 117 goto u
[PU3] 13: if errorlevel 116 goto t
[KU3] 14: if errorlevel 115 goto s
[EU3] 15: if errorlevel 114 goto r
[AU3] 16: if errorlevel 113 goto q
[6U3] 17: if errorlevel 112 goto p
[2U3] 18: if errorlevel 111 goto o
[YT3] 19: if errorlevel 110 goto n
[DU3] 20: if errorlevel 109 goto m
[9U3] 21: if errorlevel 108 goto l
[SU3] 22: if errorlevel 107 goto k
[U3] 23: if errorlevel 106 goto j
[XT3] 24: if errorlevel 105 goto i
[TT3] 25: if errorlevel 104 goto h
[NT3] 26: if errorlevel 103 goto g
[HT3] 27: if errorlevel 102 goto f
[DT3] 28: if errorlevel 101 goto e
[9T3] 29: if errorlevel 100 goto d
[GP3] 30: if errorlevel 99 goto c
[EP3] 31: if errorlevel 98 goto b
[CP3] 32: if errorlevel 97 goto a
[8N3] 33: if errorlevel 96 goto 3
[6N3] 34: if errorlevel 95 goto 2
[U94] 35: if errorlevel 49 goto ende
[D80] 36: :3
[EL2] 37: rem ** Drucktest
[Y3] 38: type dructest.txt>lpt1
[812] 39: goto :anfang
[A80] 40: :2
[CS1] 41: rem ** Info
[PN0] 42: cls
[LD3] 43: type info.txt|more
[7F7] 44: echo -----
[MW0] 45: pause
[812] 46: goto :anfang
[R90] 47: :a
[5D4] 48: rem ** Standardzeichensatz
[4D1] 49: pr 27 116 0
[812] 50: goto :anfang
[U90] 51: :b
[BK3] 52: rem ** IBM-Zeichensatz
[5D1] 53: pr 27 116 1
[812] 54: goto :anfang
[X90] 55: :c
[GU3] 56: rem ** IBM-Zeichensatz #1
[W21] 57: pr 27 54
[812] 58: goto :anfang
[0A0] 59: :d
[KU3] 60: rem ** IBM-Zeichensatz #2
[Y21] 61: pr 27 55
[812] 62: goto :anfang
[3A0] 63: :e
[XE3] 64: rem ** Breitdruck ein
[EE1] 65: pr 27 87 1
[812] 66: goto :anfang
[6A0] 67: :f
[0G3] 68: rem ** Breitdruck aus
[DE1] 69: pr 27 87 0
[812] 70: goto :anfang
[9A0] 71: :g
[714] 72: rem ** Schattendruck ein
[981] 73: pr 27 71
[812] 74: goto :anfang
[CA0] 75: :h
[D14] 76: rem ** Schattendruck aus
[C81] 77: pr 27 72
[812] 78: goto :anfang
[FA0] 79: :i
[GA3] 80: rem ** Fettdruck ein
[Z81] 81: pr 27 69
[812] 82: goto :anfang
[KA0] 83: :j
[PA3] 84: rem ** Fettdruck aus
[P21] 85: pr 27 70
[812] 86: goto :anfang
[NA0] 87: :k
[DS4] 88: rem ** 1/8 inch Zeilenabstand
[331] 89: pr 27 48
[812] 90: goto :anfang
[SA0] 91: :l
[5Y4] 92: rem ** 7/72 inch Zeilenabstand
[531] 93: pr 27 49
[812] 94: goto :anfang
```



```

[VA0] 95: :m
[XR4] 96: rem ** 1/4 inch Zeilenabstand
[YD1] 97: pr 27 51 48
[812] 98: goto :anfang
[YA0] 99: :n
[NE4] 100: rem ** 1/2 inch Zeilenabstand
[GE1] 101: pr 27 51 96
[812] 102: goto :anfang
[1B0] 103: :o
[4F3] 104: rem ** linksbündig
[3K1] 105: pr 27 97 48
[812] 106: goto :anfang
[4B0] 107: :p
[6M2] 108: rem ** zentriert
[FE1] 109: pr 27 97 49
[812] 110: goto :anfang
[7B0] 111: :q
[A93] 112: rem ** rechtsbündig
[AE1] 113: pr 27 97 50
[812] 114: goto :anfang
[AB0] 115: :r
[7N4] 116: rem ** Papierendesensor ein
[111] 117: pr 27 57
[812] 118: goto :anfang
[DB0] 119: :s
[GP4] 120: rem ** Papierendesensor aus
[031] 121: pr 27 56
[812] 122: goto :anfang
[GB0] 123: :t
[G76] 124: rem ** Wagenrücklauf + Zeilenvorschub
[YB1] 125: pr 27 53 1
[812] 126: goto :anfang
[LB0] 127: :u
[T07] 128: rem ** Wagenrücklauf ohne Zeilenvorschub
[UB1] 129: pr 27 53 0
[812] 130: goto :anfang
[PB0] 131: :v
[9X4] 132: rem ** doppelt hohe Buchstaben
[XC1] 133: pr 27 104 1
[812] 134: goto :anfang
[TB0] 135: :w

```

```

[VB4] 136: rem ** 4fach hohe Buchstaben
[UC1] 137: pr 27 104 2
[812] 138: goto :anfang
[WB0] 139: :x
[UG5] 140: rem ** löschen hohe Buchstaben
[KC1] 141: pr 27 104 0
[812] 142: goto :anfang
[2B0] 143: :y
[AP5] 144: rem ** doppelt hoch, einfach breit
[3H1] 145: pr 27 119 49
[812] 146: goto :anfang
[2C0] 147: :z
[SA2] 148: rem ** löschen
[1H1] 149: pr 27 119 48
[812] 150: goto :anfang
[9U0] 151: :ende
[PN0] 152: cls

```

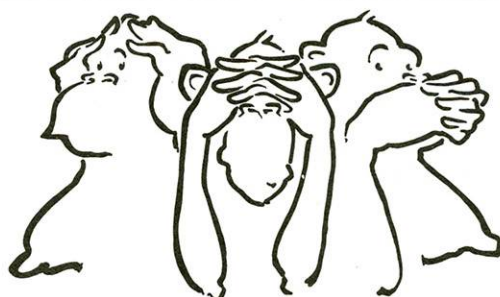
Listing 4. Eine Druckersteuerung, die als Stapeldatetei realisiert ist

```

[MK5] 1: 32 ED 8A 0E 80 00 0A C9 74 35 B8 81 00 8B F0 32
[TT5] 2: DB 32 D2 AC 2C 30 72 11 3C 09 7F 0D FE C3 86 C2
[AT5] 3: B4 0A F6 E4 02 D0 EB 0D 90 0A DB 74 08 B4 05 CD
[2S5] 4: 21 32 D2 32 DB E2 DC 0A DB 74 04 B4 05 CD 21 CD
[424] 5: 20 00 00 00 00 00 00 00 00 00 00 00 00 00 00
[024] 6: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
[024] 7: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
[024] 8: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
[024] 9: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
[024] 10: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
[024] 11: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
[024] 12: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
[024] 13: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
[024] 14: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
[024] 15: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
[024] 16: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00

```

Listing 5. »pr.com« als Hexdump für die Eingabe mit SCheck



WALDSTERBEN, WIESO? IST DOCH ALLES GRÜN.

Irrtum. Denn der Wald ist zu über 50 Prozent geschädigt.

Auf den ersten Blick sehen unsere Wälder gesund aus. Aber jeder zweite Baum ist krank. Dies konnte auch die bisherige Umweltpolitik nicht verhindern. Halbherzige Kompromisse helfen höchstens das schlechte Gewissen zu beruhigen.

Wenn Sie wissen wollen, was zur Rettung unserer letzten Waldgebiete getan werden muß, dann füllen Sie den Coupon aus. Wir beantworten Ihre Frage, sagen Ihnen, wo wir uns noch engagieren, warum Sie uns dabei helfen können.

Ja, ich will von ROBIN WOOD wissen, was zur Rettung unserer letzten Waldgebiete getan werden muß.

Name, Vorname

Straße

Wohnort

dmv

Weil ich helfen will, lege ich 3 Mark in Briefmarken bei.
Coupon bitte einsenden an
ROBIN WOOD, Postfach 10 21 22, 2800 Bremen 1

ROBIN WOOD

Gewaltfreie Aktionsgemeinschaft für Natur und Umwelt e.V.

Die Tips-und-Tricks-Fundgrube

Auf diesen Seiten finden Sie einige ausgewählte Tips und Tricks, die Einsteigern das Leben leichter machen. Besonders nützlich sind dabei zum Beispiel die Umrechnungsroutinen für die Zahlensysteme »Dezimal«, »Hexadezimal« und »Binär«. Aber Sie erfahren auch weitere nützliche Dinge, beispielsweise wie die Daten einer versehentlich mit Cola begossenen Diskette doch noch zu retten sind und vieles mehr.

Hilfe bei defekten 5 1/4-Zoll-Disketten

Wenn Sie versehentlich Kaffee oder Limonade über Ihre Diskettensammlung schütten, dann muß das noch nicht das Ende Ihrer Daten bedeuten. Meistens wird die Diskettenoberfläche durch dieses Unglück nicht beschädigt, vielmehr klebt die Magnetscheibe dann an der Hülle fest und läßt sich nicht mehr drehen.

Eine so verschmutzte Hülle können Sie auf jeden Fall wegwerfen. Doch zuvor lohnt es sich, die Daten der Diskette zu retten. Nehmen Sie eine Schere und schneiden Sie vorsichtig den Rand der Hülle auf einer Seite ab. Versuchen Sie dabei, so nah wie möglich am äußeren Rand zu bleiben, sonst beschädigen Sie die Magnetscheibe.

Dann öffnen Sie die Hülle und probieren, ob Sie die Diskette ohne Gewalt mit Daumen und Zeigefinger herausziehen können. Eventuell müssen Sie eine weitere Kante abschneiden, um die Magnetscheibe besser von der Stoffbeschichtung lösen zu können. Vergessen Sie für diesen Augenblick, daß Sie normalerweise niemals mit den Fingern die Diskettenoberfläche berühren dürfen. Sie dürfen schon, nur wirkt sich dies sehr nachteilig auf die Lebensdauer des Datenträgers aus. Im Moment geht es aber nur darum, die Diskette wenigstens noch einmal lesen zu können, um eine Kopie anzufertigen.

Wenn Sie es geschafft haben, kommt das nächste Problem. Es kann sein, daß die Oberfläche so stark verschmutzt ist, daß Sie die Schreib-/Leseköpfe des Laufwerks gefährden würden. Eine Diskette läßt sich unter fließendem, lauwar-men Wasser mit einem weichen Lappen und einem Spritzer Spülmittel vorsichtig säubern. Die meisten Rückstände von zuckerhaltigen (klebrigen) Getränken können Sie so entfernen. Achten Sie bei dieser Prozedur unbedingt darauf, daß die Diskette nicht geknickt werden darf.

Nachdem Sie die Magnetscheibe also gereinigt haben, gibt es zwei Möglichkeiten, sie dem PC anzubieten: Sicherer ist es, eine alte Diskette zu opfern. Schneiden Sie die Hülle wie oben beschrieben auf, und ersetzen Sie die Magnetscheibe durch die eben präparierte. Ganz Mutige können die »nackte« Diskette aber auch ohne Hülle ins Laufwerk schieben – vorsichtig natürlich.

Der letzte Schritt besteht nun darin, mit dem Diskcopy-Programm von MS-DOS eine Kopie der Diskette anzufertigen. Wenn Sie Glück haben, geht dies sogar ohne Lesefehler.

(tr)

Was heißt eigentlich »Hexadezimal«?

Eigentlich rechnet ein PC nicht in dem uns geläufigen, dezimalen Zahlensystem (0, 1, 2, ..., 9, 10, 11, ...), sondern in Hexadezimal und in Binär. Als PC-Besitzer kommen Sie zwangsläufig mit diesen Systemen einmal in Kontakt.

Wer andere Zahlensysteme verstehen will, muß sich erstmal mit dem Dezimal-, auch »Zehnersystem«, beschäftigen. Betrachten Sie zum Beispiel die Zahl Fünfhundertdreißig (573). Diese Zahl besteht aus drei einzelnen Ziffern, die jeweils einen eigenen Wert repräsentieren. 573 läßt sich auch so schreiben:

$$\begin{aligned} 573 &= \\ 5 \times 100 &+ \\ 7 \times 10 &+ \\ 3 \times 1 & \end{aligned}$$

Die Zahl 573 setzt sich also aus 5 Hundertern, 7 Zehnern und 3 Einern zusammen. Die Faktoren 100, 10 und 1 lassen sich auch als Zehnerpotenzen formulieren:

$$\begin{aligned} 573 &= \\ 5 \times 10^2 &+ \\ 7 \times 10^1 &+ \\ 3 \times 10^0 & \end{aligned}$$

Jede Zahl mit 0 potenziert ergibt 1, also ist 10^0 , ebenso wie 2^0 oder 16^0 , mit 1 gleichzusetzen.

Der Name des »Dezimalsystems« ist vom lateinischen Wort »decem« abgeleitet, das »Zehn« bedeutet. Jede Stelle einer Zahl im Zehnersystem kann ja zehn verschiedene Werte (von 0 bis 9 sind es zehn Ziffern) annehmen.

Ein Computer »kennt« auf unterster Prozessorebene keine Zahlen von 0 bis 9, sondern nur 1 und 0. »1« bedeutet »Strom an« und »0« bedeutet »Strom aus«. Um einem Computer das Rechnen beizubringen, benötigt man also ein Zahlensystem, das nur aus diesen beiden Werten (Zuständen) besteht. Jede nur denkbare Zahl, jedes Zeichen, jeder Befehl muß auf irgendeine Weise durch diese beiden Zahlen ausgedrückt sein.

Eben dies ist das Binärsystem (von der lateinischen Vorsilbe »bi« abgeleitet, die »aus zwei Einheiten bestehend« bedeutet). Jede Position einer Binärzahl – ein »Bit« (englisch »bit« = »kleinste Einheit«) – besteht also aus den Zahlen 0 oder 1 und stellt Vielfache einer Zweierpotenz dar. Die binäre Zahl 100111bin setzt sich zum Beispiel so zusammen:

$$\begin{aligned} 100111\text{bin} &= \\ 1 \times 2^5 &+ \\ 0 \times 2^4 &+ \\ 0 \times 2^3 &+ \\ 1 \times 2^2 &+ \\ 1 \times 2^1 &+ \\ 1 \times 2^0 & \end{aligned}$$

An diesem Beispiel haben Sie auch gleich gelernt, wie man binäre in dezimale Zahlen umrechnet. Die binäre Zahl 100111bin ergibt also:

$$\begin{aligned} 1 \times 2^5 &+ 0 \times 2^4 + 0 \times 2^3 + 1 \times 2^2 + 1 \times 2^1 + 1 \times 2^0 = \\ 2^5 &+ 2^2 + 2^1 + 2^0 = \\ 32 &+ 4 + 2 + 1 = \\ 39 & \end{aligned}$$

Von Bits und Bytes

Den Begriff »Byte« kennt eigentlich jeder PC-Besitzer. Der Hauptspeicher des Rechners oder die Kapazität der Festplatte wird in Byte, genauer gesagt in KByte und MByte gemessen. Ein Byte ist eine Speichereinheit, die Werte von 0 bis 255 annehmen kann. Warum gerade 255? Ganz einfach, die Zahl 255 besteht im Binärsystem aus acht Einsen (11111111bin) und ist die größte Zahl, die sich mit 8 Bit darstellen läßt. Ein 8-Bit-Prozessor kann also mit Zahlen von 0 bis maximal 255 rechnen.

Die Einheit »1 KByte« (sprich »kah-beit«) entspricht 1024 Byte. Auch wenn viele Anwender häufig von »Kilo-Byte« sprechen, so ist dieser Ausdruck augenommen falsch. Denn die Einheit »Kilo« bedeutet ja »1000«, und ein KByte sind 1024 und nicht 1000 Byte. Demnach passen auf eine 360-KByte-Diskette auch nicht 360000, sondern 368640

Byte (360 x 1024). Und ein PC hat nicht 640000 Byte Hauptspeicher, sondern etwas mehr, nämlich 655360. Zurück zu den Zahlensystemen eines Computers. Auch das Hexadezimalsystem (griechisch »hexa« = »sechs«) baut auf dem gleichen Prinzip wie das Binärsystem auf. Bei einer hexadezimalen Zahl jedoch hat jede Stelle 16 Werte von 0 bis 15. Um die Werte 10 bis 15 zu repräsentieren, verwendet man die Buchstaben »A« bis »F«. Das heißt, daß eine Ziffer einer Hexadezimalzahl die Zeichen 0 bis 9 und »A« bis »F« annehmen kann. Ein »A« ist mit dem dezimalen Wert 10 gleichzusetzen, »B« bedeutet 11, »C« 12 und so weiter. Eine Position innerhalb einer Hexadezimalzahl stellt also nicht Potenzen von 10 oder von 2, sondern von 16 dar. Betrachten Sie sich als Beispiel die folgende Zahl:

1FChex =
1 x 16² +
F x 16¹ +
C x 16⁰

Mit den Buchstaben »F« und »C« läßt sich leider etwas schwer rechnen. Ersetzen Sie sie durch normale Zahlen nach folgender Tabelle:

A = 10, B = 11, C = 12, D = 13, E = 14, F = 15

Als »Eselsbrücke« merken Sie sich zum Beispiel: »C« ist Zwölf, »D« ist Dreizehn, »F« ist Fünfzehn. Die Beispielszahl 1FChex läßt sich jetzt leicht ins Zehnersystem umrechnen:

1FChex =
1 x 16² +
15 x 16¹ +
12 x 16⁰
= 256 + 15 x 16 + 12
= 256 + 240 + 12
= 508

Fällt Ihnen etwas auf? 16² ergibt 256, das ist um 1 mehr als die vorher erklärte Einheit »Byte«. Zufall? Keineswegs, denn mit zwei Stellen einer Hexadezimalzahl lassen sich insgesamt 255 Werte darstellen. Die größte zweistellige Hexzahl ist FFhex. Und das ergibt

FFhex = 15 x 16 + 15 = 255

Sie haben jetzt gelernt, wie Sie Zahlen aus anderen Systemen ins Dezimalsystem zurückrechnen. Viel schwieriger ist der umgekehrte Weg in ein anderes Rechensystem. Als Beispiel soll die Zahl 45054 dienen, die ins Hexadezimalsystem umzurechnen ist. Dazu prüfen Sie, wie oft diese Zahl durch die höchste Potenz von 16 teilbar ist. Sie wissen, daß 16⁴ bereits 65536 ergibt, beginnen die Überprüfung also bei 16³ (= 4096):

45054 / 16³ = 10, ...
45054 - 10 x 16³ = 4094 (Rest)
4094 / 16² = 15, ...
4094 - 15 x 16² = 254 (Rest)
254 / 16¹ = 15, ...
254 - 15 x 16¹ = 14 (Rest)
14 / 16⁰ = 14

Sie erhalten also die Ziffern 10, 15, 15 und 14 des hexadezimalen Zahlensystems, und das wiederum ergibt AF-FEhex.

Für das binäre Zahlensystem gehen Sie genauso vor, arbeiten dann aber mit Zweierpotenzen. Als Beispiel dient die Zahl 215dez:

215 / 2⁷ = 1, ...
215 - 1 x 2⁷ = 87 (Rest)
87 / 2⁶ = 1, ...
87 - 1 x 2⁶ = 23 (Rest)
23 / 2⁵ = 0, ...
23 / 2⁴ = 1, ...
23 - 1 x 2⁴ = 7 (Rest)
7 / 2³ = 0
7 / 2² = 1, ...
7 - 1 x 2² = 3 (Rest)
3 / 2¹ = 1, ...
3 - 1 x 2¹ = 1 (Rest)

1 / 2⁰ = 1

Unser Ergebnis lautet also:

215dez = 11010111bin

(tr)

Zahlensysteme umrechnen

Umrechnungsroutinen für die drei Zahlensysteme Dezimal, Hexadezimal und Binär können Sie immer brauchen. Um aus einer Dezimal- eine Hexadezimalzahl zu erhalten, gibt es die Basic-Funktion »hex\$«:

```
print hex$(240)
```

Eine Dezimalzahl zu konvertieren, zum Beispiel in eine Binärzahl, ist allerdings erheblich schwieriger. In den Listings 1 bis 3 sehen Sie Umrechnungsroutinen für alle Zahlensysteme. Bei Listing 1 (»dez.bas«) geben Sie eine dezimale Zahl ein und erhalten das Äquivalent in Hexadezimal und in Binär. Listing 2 (»hex.bas«) rechnet eine hexadezimale Zahl nach Dezimal und Binär um. Und Listing 3 (»bin.bas«) erwartet die Eingabe einer Binärzahl und zeigt daraufhin das dezimale und hexadezimale Format der Zahl.

Die Routinen sind für GW-Basic ausgelegt, funktionieren aber auch mit jedem anderen Basic-Dialekt. Beachten Sie aber, daß die Hex\$-Funktion bei Zahlen über 65535 teilweise recht seltsame Ergebnisse liefert. Während GW-Basic bei

```
print hex$(65536)
```

einen »Überlauf« signalisiert, meldet SBasic beispielsweise das Ergebnis »8000«. Die Hex\$-Funktion ist also nur für Werte unterhalb 65536 geeignet. Dementsprechend geben die Listings auch nur bis zu dieser Grenze korrekte Werte aus.

Wenn Sie den SBasic-Compiler aus der DOS Extra 6'89 besitzen, können Sie sich aus den Listings drei nützliche Utilities für Ihren PC basteln. Ersetzen Sie die jeweils erste Zeile der Listings.

In Listing 1 schreiben Sie:

```
10 d=val(command$(1))
```

In Listing 2 ersetzen Sie die Zeile 10 durch:

```
10 h$=command$(1)
```

Und in Listing 3 lautet die Zeile 10 dann:

```
10 b$=command$(1)
```

Diese drei Programme übersetzen Sie mit dem SBasic-Compiler. Danach können Sie jederzeit direkt auf der MS-DOS-Eingabeebene eine Zahl umrechnen. Wenn Sie beispielsweise 1F0hex konvertieren möchten, tippen Sie ein:

```
hex 1F0
```

Auf dem Bildschirm erhalten Sie daraufhin die entsprechenden Werte in Dezimal und Hexadezimal.

(tr)

```
[CZ3] 10 INPUT "Dezimalzahl: ",D
[F61] 20 PRINT
[ML1] 30 H$=HEX$(D)
[C57] 40 WHILE LEN(H$)/2<>INT(LEN(H$)/2) : H$="0"+H$ :
WEND
[YZ2] 50 PRINT "    hex: ";H$
[BP4] 60 FOR I=(LEN(H$)*4-1) TO 0 STEP -1
[572] 70 A=INT(D/2^I)
[UY2] 80 B$=B$+CHR$(48-(A=1))
[CN1] 90 D=D-A*2^I
[H71] 100 NEXT I
[EG3] 110 PRINT " binaer: ";B$
```

Listing 1. Umrechnen einer Dezimalzahl ins Binär- und Hexadezimalformat mit »dez.bas«


```
[CD3] 10 INPUT "Hexzahl: ",H$
[F61] 20 PRINT
[RS2] 30 FOR I=1 TO LEN(H$)
[RC3] 40 D=ASC(RIGHT$(H$,I))-48
[GP2] 50 IF D>9 THEN D=D-7
[303] 60 IF D>15 THEN D=D-32
[762] 70 X=X+D*16^(I-1)
[G51] 80 NEXT I
[N43] 90 PRINT "dezimal: ";X
[CW4] 100 FOR I=(LEN(H$)*4-1) TO 0 STEP -1
[UF2] 110 A=INT(X/2^I)
[HX2] 120 B$=B$+CHR$(48-(A=1))
[7W1] 130 X=X-A*2^I
[X71] 140 NEXT I
[UG3] 150 PRINT " binaer: ";B$
```

Listing 2. »hex.bas« rechnet Hexadezimalzahlen ins Dezimal- und Binärsystem um

```
[6S3] 10 INPUT "Binaerzahl: ",B$
[F61] 20 PRINT
[ZR2] 30 FOR I=1 TO LEN(B$)
[GX4] 40 D=D+(ASC(RIGHT$(B$,I))-48)*2^(I-1)
[A51] 50 NEXT I
[733] 60 PRINT "dezimal: ";D
[W11] 70 H$=HEX$(D)
[M57] 80 WHILE LEN(H$)/2<>INT(LEN(H$)/2) : H$="0"+H$ : WEND
[603] 90 PRINT " hex: ";H$
```

Listing 3. Mit »bin.bas« konvertieren Sie Binärzahlen nach Dezimal und Hexadezimal

Wie ist der Zeichensatz aufgebaut?

Jeder Buchstabe, jede Zahl und jedes Sonderzeichen Ihres PC ist aus vielen Einzelpunkten zusammengesetzt. Mit Hilfe eines kurzen Basic-Programms wird diese »Punktmatrix« ausgelesen und vergrößert auf dem Bildschirm dargestellt.

In Listing 4 sehen Sie, wie so etwas gemacht wird. Das Programm »zeichen.bas« liest die Punktmatrix aus und ersetzt jeden Einzelpunkt durch ein ausgefülltes Kästchen. Es ist recht interessant, sich einmal genau anzusehen, wie eigentlich ein »@« oder ein »ß« aufgebaut sind.

Wenn Ihnen der Durchlauf der Zeichen zu schnell geht, dann fügen Sie folgende Zeile in das Programm ein:

```
196 while inkey$="" : wend
```

Dadurch wartet der GW-Basic-Interpreter nach der Ausgabe jedes Zeichens auf einen beliebigen Tastendruck, und Sie haben so Gelegenheit, die Grafik genau zu analysieren.

Übrigens sind im ROM nur die ersten 128 des 256 Zeichen großen »Character-Satzes« gespeichert. Die restlichen sind von der Wahl des verwendeten Tastaturreibers beziehungsweise der gewählten Code-Seite (ab MS-DOS 3.3) abhängig. Der Zugriff auf die Matrix dieser erweiterten Zeichen ist allerdings mit erheblichem Programmieraufwand verbunden und unter GW-Basic nicht sinnvoll. (tr)

```
[BP6] 10 REM Zeichensatz auslesen und anzeigen
[GX5] 20 REM für alle IBM-kompatiblen PCs
[TT0] 30 REM
[K64] 100 DEF SEG=&HFF00:ADR=&HA6E
[EH2] 110 FOR I=0 TO 127
[YT5] 115 PRINT CHR$(218)STRING$(8,196)CHR$(191)
[VD4] 120 FOR X=ADR+I*8 TO ADR+I*8+7
[WR5] 130 BYTE=PEEK(X):B$="":REM 8 Space
[4W3] 140 FOR BIT=8 TO 1 STEP -1
[MM6] 150 IF BYTE MOD 2 THEN MID$(B$,BIT,1)=CHR$(219)
[Z53] 160 BYTE=INT(BYTE/2)
[4W1] 170 NEXT BIT
[SC4] 180 PRINT CHR$(179);B$;CHR$(179)
[MK1] 190 NEXT X
[VU5] 195 PRINT CHR$(192)STRING$(8,196)CHR$(217)
[4C1] 200 PRINT
[HG1] 210 NEXT I
```

Listing 4. »zeichen.bas« zeigt Ihnen den Zeichensatz Ihres PC

Systeminfos unter GW-Basic

Am Anfang des Basic-Datensegments sind einige interessante Speicheradressen zu finden. Dort erfahren Sie zum Beispiel die Größe des Gesamtarbeitsspeichers und die Anzahl der angeschlossenen Diskettenlaufwerke.

Wenn Sie Basica anstatt GW-Basic besitzen (bei PC-DOS), dann können Sie zusätzlich herausfinden, welche Zeilennummer gerade bearbeitet wird und in welcher Zeile zuletzt ein Fehler auftrat. In »info.bas« (Listing 5) sehen Sie, wie das gemacht wird. Die Zeilen ab 400 brauchen Sie nicht abzutippen, wenn Sie nur GW-Basic besitzen. Die entsprechenden Befehle würden sonst unsinnige Werte ergeben. (tr)

```
[DW5] 100 REM Zeitangabe in Zeiteinheiten
[U12] 110 DEF SEG=64
[AG8] 120 ZE=PEEK(108)+256*PEEK(109)+(PEEK(110)+256*PEEK(111))*65536!
[2K4] 130 PRINT "Zeiteinheiten: ";ZE
[CH0] 140 '
[2V4] 200 REM Gesamtspeicherplatz
[MW1] 210 DEF SEG=0
[RD4] 220 GS=PEEK(1043)+256*PEEK(1044)
[416] 230 PRINT "Gesamtspeicher: ";GS;"KByte"
[EH0] 240 '
[VL5] 300 REM Anzahl Diskettenlaufwerke
[PW1] 310 DEF SEG=0
[DW9] 320 IF PEEK(1040) AND 1=0 THEN LW=0 ELSE LW=((PEEK(1040) AND 192)/64)+1
[AM6] 330 PRINT "Anzahl Diskettenlaufwerke: ";LW
[GH0] 340 '
[Z06] 400 REM Aktuelle Basic-Zeilenummer
[XP5] 405 REM (nur Basica, nicht GW-Basic!)
[Z02] 410 DEF SEG=64
[YZ3] 420 NR=PEEK(46)+256*PEEK(47)
[816] 430 PRINT "Aktuelle Zeilennummer: ";NR
[KH0] 440 '
[NP6] 500 REM Zeilennummer mit letztem Fehler
[ZP5] 505 REM (nur Basica, nicht GW-Basic!)
[Z22] 510 DEF SEG=64
[H84] 520 FZ=PEEK(839)+256*PEEK(840)
[AE5] 530 PRINT "Letzte Fehlerzeile: ";FZ
```

Listing 5. Mit GW-Basic und dem Programm »info.bas« erfahren Sie einiges über die Ausstattung Ihres Rechners

Tip für »Viel-Löcher«

Oft löscht ein PC-Anwender viel zu schnell eine Datei mit »del« oder »erase«, nur um eine halbe Stunde später zu der Erkenntnis zu kommen, daß der Inhalt vielleicht doch noch ganz nützlich gewesen wäre. Am sichersten ist es, ein Verzeichnis auf der Festplatte einzurichten, das nur der Aufnahme von alten Dateien dient:

```
mkdir c:\alt
```

Dann legen Sie sich eine Stapeldatei »loesche.bat« an:

```
copy con loesche.bat
@echo off
copy %1 c:\alt
del %1
[F6] [Enter]
```

Anstatt »del text.bak« einzutippen, verwenden Sie ab jetzt »loesche text.bak«. Das Ergebnis, nämlich aus dem aktuellen Verzeichnis die Datei »text.bak« zu entfernen, haben Sie so auch erreicht. Allerdings haben Sie die Gewißheit, daß die Datei im Unterverzeichnis »c:\alt« noch existiert. Alle vier oder fünf Wochen schauen Sie dann kurz das Verzeichnis durch und löschen die ältesten Dateien, um die Speicherkapazität der Festplatte nicht unnötig zu belasten. (tr)

DOS INTERNATIONAL

12 X IM JAHR MONAT FÜR MONAT NEU

Im Inland bzw. West-Berlin	Im europäischen Ausland	Im außereuropäischen Ausland
6 Ausg. = 42,-	6 Ausg. = 55,-	6 Ausg. = 75,-
12 Ausg. = 82,50	12 Ausg. = 110,-	12 Ausg. = 150,-

DOS EXTRA

4 X IM JAHR 4 X POWERPROGRAMME FÜR IHREN PC

Im Inland bzw. West-Berlin	Im europäischen Ausland	Im außereuropäischen Ausland
4 Ausgaben 60,- DM	4 Ausgaben 70,- DM	4 Ausgaben 80,- DM

Widerrufsrecht: Jeder Abonnent hat das Recht, seine Bestellung innerhalb einer Woche beim DMV-Verlag, Postfach 250, 3440 Eschwege, schriftlich zu widerrufen. Die rechtzeitige Absendung des Widerrufsschreibens genügt zur Fristwahrung.

Bitte benutzen Sie die Bestellkarte.

DMV-VERLAG · POSTFACH 250 · 3440 ESCHWEGE

DMV
Daten- und
Medienverlag

Tips und Tricks zum Prompt-Befehl

Der Prompt-Befehl gehört schon seit der ersten Version von MS-DOS zum Befehlsrepertoire des Betriebssystems. Neben den interessanten Standardanwendungen erleichtert er zusammen mit dem Ansi-Bildschirmtreiber den Umgang mit der Kommandoebene wesentlich.

Das Prompt-Kommando gehört zu den sinnvollsten Bestandteilen des MS-DOS-Betriebssystems. Mit ihm kontrollieren Sie vorwiegend das sogenannte »Prompt«, das heißt, die Systemanfrage, die in der Kommandozeile anzeigt, daß MS-DOS auf eine Eingabe wartet. Die Standard-Einstellung beim Einschalten des Computers ist so programmiert, daß Sie erfahren, welches Laufwerk zur Zeit aktiv ist. Das sieht beispielsweise so aus:

C>

Die Prompt-Funktion versetzt Sie in die Lage, zusammen mit den zugehörigen Parametern, die in Tabelle 1 aufgeführt sind, eine informative Systemanzeige zu realisieren. Um Ihnen das Verfahren an einem Beispiel zu verdeutlichen, zeigen die folgenden Ausführungen, wie dies zu bewerkstelligen ist.

Als erstes interessiert neben dem aktiven Laufwerk der aktuelle Pfad. Denn es passiert oft, daß Sie nicht mehr genau wissen, in welchem Unterverzeichnis Sie sich befinden. In diesem Fall ist eine Orientierungshilfe mit dem nachfolgenden Größer-als-Zeichen sehr sinnvoll. Diesem Zweck dient die Befehlszeile

prompt \$p\$g

In vielen Fällen ist es hilfreich, neben dem Verzeichnis auch die Zeit zu wissen. In diesem Fall hilft die Eingabe

prompt \$t\$ \$p\$g

Dabei ergeben sich jedoch Probleme. Erstens sind die Sekunden- und Hundertstelsekunden-Angaben von zweitrangiger Bedeutung, zweitens wirkt die Konstruktion unübersichtlich, und drittens raubt sie einen Teil des ohnehin spärlichen Platzes in der Kommandozeile. Mit dem \$h\$-Parameter, der den Cursor um die gewünschte Anzahl von Stellen zurücksetzt, umgehen Sie dieses Hindernis. So führt der Befehl

prompt \$t\$ \$h\$ \$h\$ \$h\$ \$h\$ \$h\$ \$h\$ \$h\$ \$p\$ \$g\$

zum gewünschten Ergebnis.

Prompt-Funktionen		
Zeichen- kette	Bedeutung	ASCII- Zeichenfolge
\$_	Wagenrücklauf und Zeilenvorschub	13, 10
\$ \$	Dollarzeichen	36
\$b	senkrechte Linie	124
\$d	aktuelles Datum	-
\$e	Escape-Sequenz (Tabelle 2)	27
\$g	Größer-Zeichen	62
\$h	Backspace	8
\$l	Kleiner-Zeichen	60
\$n	aktuelles Laufwerk	-
\$p	aktuelle Pfadangabe	-
\$q	Gleichheitszeichen	61
\$t	Systemzeit	-
\$v	Nummer der DOS-Version	-

Tabelle 1. Der Prompt-Befehl kann zusammen mit einer Vielzahl von Parametern eingesetzt werden

Die Fähigkeiten des Prompt-Befehls sind mit diesen Variationen aber noch nicht ausgereizt. Nach der Einbindung des Ansi-Treibers benutzen Sie die Prompt-Funktion auch dazu, Steuersequenzen an diesen Betriebssystemteil weiterzuleiten.

Der Ansi-Treiber, der erweiterte Bildschirm-Steuerfunktionen zur Verfügung stellt, befindet sich zusammen mit anderen Dateien auf der MS-DOS-Systemdiskette. Er wurde vom amerikanischen Ansi-Komitee (»Ansi« steht für »American National Standard Institute«) normiert und bildet somit die Grundlage für viele Computer.

Eine wichtige Aufgabe des Treibers liegt darin, die Tastaturbelegung zu ändern, um damit beispielsweise die Funktionstasten umzudefinieren.

Der Ansi-Treiber wird über die Anweisung

device=ansi.sys

die sich in der Config-Datei befinden muß, in das Betriebssystem eingebunden.

Eine Ansi-Befehlssequenz beginnt immer mit einem Escape-Zeichen (ASCII-Code 27) oder beim Prompt-Kommando entsprechend mit der Anweisung »\$e[«, die der Treiber automatisch in den Esc-Code umwandelt. Tabelle 2 zeigt übersichtlich, welche Hilfsmittel der Ansi-Treiber bietet. Sie können ihn sogar dazu verwenden, einen anderen als den aktiven Bildschirmmodus anzuwählen (Tabelle 4) oder das Schriftattribut festzulegen (Tabelle 5).

Mit diesen Angaben sind Sie in der Lage, Ihre Systemanzeige weiter zu verbessern. Der Befehl mit den zugehörigen Parametern

prompt \$e[7m\$ \$h\$ \$h\$ \$h\$ \$h\$ \$h\$ \$h\$ \$h\$ / \$p\$ \$e[0m\$g

sorgt für noch mehr Übersichtlichkeit, indem er die Systemanzeige invertiert.

Neben der Verbesserung der Systemanzeige dient das Prompt-Kommando auch – wie eben schon erwähnt – zur Umbelegung der Tastatur. In der entsprechenden Ansi-Zeichenfolge geben Sie zuerst den Tastencode der zu belegenden Taste an, der mit einem Semikolon abschließt, und danach den Text in Anführungszeichen beziehungsweise den Tastatur-Code, mit dem der Ansi-Treiber die Taste belegt soll. Zum Abschluß folgt ein »p«. Der Ta-

Prompt-Steuerzeichen (ANSI)	
Zeichenkette	Bedeutung
\$e[6n	Treiber gibt eine Escape-Sequenz mit aktueller Cursor-Position aus
\$e[#A	Cursor um # Zeilen nach oben bewegen
\$e[#B	Cursor um # Zeilen nach unten bewegen
\$e[#C	Cursor um # Spalten nach rechts bewegen
\$e[#D	Cursor um # Spalten nach links bewegen
\$e[H	Cursor in die linke obere Ecke des Bildschirms setzen
\$e[#;#H	Cursor in angegebene Zeile (erstes #) und Spalte (zweites #) setzen
\$e[2J	Bildschirm löschen und Cursor in die linke obere Ecke setzen
\$e[K	Bildschirm von Cursor-Position bis Zeilenende löschen
\$e[#;#f	Cursor in angegebene Zeile (erstes #) und Spalte (zweites #) setzen
\$e[#h	Bildschirmmodus # setzen (Tabelle 3)
\$e[#l	Bildschirmattribut setzen (siehe Tabelle 4)
\$e[#;#;...;#m	
\$e[#;#p	Scancode oder Zeichenkette (zweites #) der Taste mit Scancode (erstes #) zuweisen
\$e[s	aktuelle Cursor-Position zwischenspeichern
\$e[u	gespeicherte Cursor-Position wiederherstellen

Tabelle 2. Die wichtigsten Angaben des Ansi-Treibers auf einen Blick

Scan-Code	Sondertaste	Scan-Code	Sondertaste
129	Alt-0	116	Ctrl-Cursor-rechts
120	Alt-1	117	Ctrl-End
121	Alt-2	94	Ctrl-F1
122	Alt-3	95	Ctrl-F2
123	Alt-4	96	Ctrl-F3
124	Alt-5	97	Ctrl-F4
125	Alt-6	98	Ctrl-F5
126	Alt-7	99	Ctrl-F6
127	Alt-8	100	Ctrl-F7
128	Alt-9	101	Ctrl-F8
30	Alt-A	102	Ctrl-F9
48	Alt-B	103	Ctrl-F10
46	Alt-C	119	Ctrl-Home
32	Alt-D	118	Ctrl-PgDn
18	Alt-E	132	Ctrl-PgUp
33	Alt-F	114	Ctrl-PrtSc
104	Alt-F1	72	Cursor hoch
105	Alt-F2	75	Cursor links
106	Alt-F3	77	Cursor rechts
107	Alt-F4	80	Cursor runter
108	Alt-F5	83	Del
109	Alt-F6	79	End
110	Alt-F7	59	F1
111	Alt-F8	60	F2
112	Alt-F9	61	F3
113	Alt-F10	62	F4
34	Alt-G	63	F5
35	Alt-H	64	F6
23	Alt-I	65	F7
36	Alt-J	66	F8
37	Alt-K	67	F9
38	Alt-L	68	F10
50	Alt-M	71	Home
49	Alt-N	82	Ins
24	Alt-O	81	PgDn
25	Alt-P	73	PgUp
16	Alt-Q	84	Shift-F1
19	Alt-R	85	Shift-F2
31	Alt-S	86	Shift-F3
20	Alt-T	87	Shift-F4
22	Alt-U	88	Shift-F5
47	Alt-V	89	Shift-F6
17	Alt-W	90	Shift-F7
45	Alt-X	91	Shift-F8
21	Alt-Y	92	Shift-F9
44	Alt-Z	93	Shift-F10
115	Ctrl-Cursor-links	15	Shift-Tab

Tabelle 3. Bei der Umbelegung der Tastatur helfen Ihnen die Scan-Codes der PC/AT-Tastatur

VirusDoktor

Eine Programmsammlung, die Sie gegen alle bekannten PC-Computer-Viren wappnet

Auch schon betroffen?

- Schäden durch infizierte, unbrauchbare Programmdateien?
- Datenverlust durch formatierte und zerstörte Speichermedien?
- Zeitverlust und Ärger durch Verstellen der Systemkonfiguration?



TEST SIEGER
PC Plus 9/89

Name: VirusDoktor
Preis: 98 Mark
Anbieter: DMV-Verlag
Kurzinfo: Das Programm checkt zuverlässig und äußerst schnell. Die durchgängige Logik, die CMOS-RAM-Backup-Möglichkeit und der niedrige Preis sprechen ebenfalls für das Programm.
Schutzeffekt: hoch
Killmechanismus: nein
Handhabung: sehr gut

Prädikat
+++++

1 PLUS-Punkt = ungenügend,
10 PLUS-Punkte = hervorragend

VirCheck -

Ein Kontrollprogramm der Superklasse!

- Kontrolle Ihrer Programme auf Längenänderung, Attributierung etc.
- Spezieller Check für virusbedingte Abänderung des Programmcodes
- Überwachung von Umbenennungen und Verschiebungen von Dateien
- Beliebige Auswahl der kontrollierten Programmgruppen
- Hohe Effektivität und Geschwindigkeit durch variables SETUP
- Komfortables, menügesteuertes, grafisches Installationsprogramm

VirSperr -

dreimal Schutz vor Virusinfektion

- Drei Programme zum Absichern Ihrer Dateien vor Veränderungen
- Ausführlicher Report über versuchte Zugriffe auf Ihre Dateien
- Virussicheres Fixieren von gesetzten Read-Only-Attributen
- Ein- und Ausschalten des Schreibschutzes per Tastendruck

Und als Zugabe

CMOS-BKP - Konfigurationssicherung für AT- und AT-kompatible Rechner

VirusDoktor

Fünf wertvolle Programme zum Preis von einem plus ausführliche Programmanleitung inklusive Grundwissen über Computerviren

Für alle MS-DOS-Computer

99,- DM (unverbindliche Preisempfehlung)

Wenn Sie über den DMV-Bestellservice bestellen, gilt folgendes:

Inland:		Ausland:	
Einzelpreis	99,- DM	Einzelpreis	99,- DM
zzgl. Versandkosten	4,- DM	zzgl. Versandkosten	6,- DM
Endpreis	103,- DM	Endpreis	105,- DM

Bitte benutzen Sie die Bestellkarte.

DMV-Verlag · Postfach 250 · 3440 Eschwege

Bildschirmmodi für die Prompt-Steuerzeichen

Parameter Modus

0	monochrom, 40x25 Zeichen
1	Farbe, 40x25 Zeichen
2	monochrom 80x25 Zeichen
3	Farbe, 80x25 Zeichen
4	Farbgrafik, 320x200 Bildpunkte
5	Monochromgrafik, 320x200 Bildpunkte
6	Monochromgrafik, 640x200 Bildpunkte

Tabelle 4. Ein Steuerzeichen bestimmt den Bildschirmmodus

stencode entspricht für die Standardzeichen dem ASCII-Wert der Taste.

So läßt sich jeder PC mit dem Befehl

prompt \$e[88;85p

ein X für ein U vormachen. Als Beispiel zur Verdeutlichung des vermittelten Sachverhalts dient Listing 1, das sowohl die eckigen Klammern auf die Tastenkombination »Control-Q« und »Control-W« als auch das oft benutzte Backslash-Zeichen auf das weniger wichtige Doppelkreuz legt.

Neben der Standard-Zuweisung über die ASCII-Codes kann der Ansi-Treiber auch Sondertasten wie beispielsweise die Funktionstasten umbelegen. Die Sondertasten erreichen Sie über die sogenannten »Scan-Codes«, die in Tabelle 3 zusammengefaßt sind. Als Anwendungsbeispiele finden Sie in Listing 2 eine Batch-Datei, die Sie in die Lage versetzt, über Funktionstasten alle von Ihnen oft benutzten Programme aufzurufen. Listing 3 erleichtert hingegen den Umgang mit den Standardkommandos des Be-

Prompt-Zeichenattribute (ANSI)

Befehlsfolge Attribut

Bemerkung

\$e[0m	normal	-
\$e[1m	intensiv	-
\$e[4m	unterstrichen	nur Monochromadapter
\$e[5m	blinkend	nur Monochromadapter
\$e[7m	invertiert	-
\$e[8m	unsichtbar	nur Monochromadapter
\$e[30m	Textfarbe schwarz	nur CGA, EGA und VGA
\$e[31m	Textfarbe rot	nur CGA, EGA und VGA
\$e[32m	Textfarbe grün	nur CGA, EGA und VGA
\$e[33m	Textfarbe gelb	nur CGA, EGA und VGA
\$e[34m	Textfarbe blau	nur CGA, EGA und VGA
\$e[35m	Textfarbe lila	nur CGA, EGA und VGA
\$e[36m	Textfarbe kobaltblau	nur CGA, EGA und VGA
\$e[37m	Textfarbe weiß	nur CGA, EGA und VGA
\$e[40m	Hintergrundfarbe schwarz	nur CGA, EGA und VGA
\$e[41m	Hintergrundfarbe rot	nur CGA, EGA und VGA
\$e[42m	Hintergrundfarbe grün	nur CGA, EGA und VGA
\$e[43m	Hintergrundfarbe gelb	nur CGA, EGA und VGA
\$e[44m	Hintergrundfarbe blau	nur CGA, EGA und VGA
\$e[45m	Hintergrundfarbe lila	nur CGA, EGA und VGA
\$e[46m	Hintergrundfarbe kobaltblau	nur CGA, EGA und VGA
\$e[47m	Hintergrundfarbe weiß	nur CGA, EGA und VGA

Tabelle 5. Bei der Ausgabe von Texten ist man nicht auf Standard-Schriftattribute angewiesen

triebssystems. Bei allen vorgestellten Beispielen ist es am sinnvollsten, die Befehle in die Autoexec-Datei einzubinden.

(Markus Zietlow/wn)

Name	prompt1.bat
Sprache	MS-DOS
Grafikkarte	jede
Besonderheiten	keine

```
[7N0] 1: rem
[796] 2: rem Programm, das die eckigen Klammern
[R27] 3: rem auf die Control-Q und Control-W-Taste
[T67] 4: rem der Tastatur und das Backslash-Zeichen
[4W5] 5: rem auf die Doppelkreuztaste legt
[7N0] 6: rem
[ZR2] 7: prompt $e[17;"["p
[7S2] 8: prompt $e[23;"]"p
[MH2] 9: prompt $e[35;92p
[7W1] 10: prompt $p$g
```

Listing 1. Die Tastaturumbelegung über die Prompt-Funktion erleichtert den Zugriff auf Sonderzeichen

Name	prompt2.bat
Sprache	MS-DOS
Grafikkarte	jede
Besonderheiten	keine

```
[7N0] 1: rem
[2E6] 2: rem Beispiel einer Batch-Datei, die die
[BE7] 3: rem Funktionstasten mit rem Programmnamen
[TS1] 4: rem belegt
[7N0] 5: rem
[WK3] 6: prompt $e[0;84;"Fa";13p
[5Y3] 7: prompt $e[0;85;"Word";13p
[LY5] 8: prompt $e[0;86;"C:\Windows\Win";13p
[014] 9: prompt $e[0;87;"dBase";13p
[464] 10: prompt $e[0;88;"Lotus";13p
[DN3] 11: prompt $e[0;89;"l13";13p
[7W1] 12: prompt $p$g
[DA1] 13: echo off
[PN0] 14: cls
```

```
[GW7] 15: echo »Sft« entspricht der »Shift«-Taste
[YS0] 16: echo.
[BX8] 17: echo Sft&F1: F&A -Sft&F2: MS Word 4.0-
Sft&F3: MS Windows 2.0
[G69] 18: echo Sft&F4: dBase III-Sft&F5: Lotus 123 -
Sft&F6: Laplink V3.0
```

Listing 2. Ein Beispiel für die Programmauswahl über Funktionstasten in einer Autoexec-Datei

Name	prompt3.bat
Sprache	MS-DOS
Grafikkarte	jede
Besonderheiten	keine

```
[7N0] 1: rem
[2E6] 2: rem Beispiel einer Batch-Datei, die die
[YZ6] 3: rem Funktionstasten mit MS-DOS-Befehlen
[TS1] 4: rem belegt
[7N0] 5: rem
[AS3] 6: prompt $e[0;59;"dir";13p
[YF3] 7: prompt $e[0;60;"dir "p
[RR3] 8: prompt $e[0;62;"copy "p
[VM3] 9: prompt $e[0;63;"chdir "p
[VN3] 10: prompt $e[0;64;"mkdir "p
[EP3] 11: prompt $e[0;65;"rmdir "p
[ZY3] 12: prompt $e[0;66;"format "p
[DN3] 13: prompt $e[0;67;"erase "p
[TP3] 14: prompt $e[0;68;"mode "p
[7W1] 15: prompt $p$g
[DA1] 16: echo off
[PN0] 17: cls
[AK9] 18: echo Die Belegung der Funktionstasten mit M
S-DOS-Befehlen
[YS0] 19: echo.
[7B7] 20: echo F1 : DIR [Return] - F2 : DIR
- F4 : COPY
[287] 21: echo F5 : CHDIR - F6 : MKDIR
- F7 : RMDIR
[ZX6] 22: echo F8 : FORMAT - F9 : ERASE
- F10: MODE
```

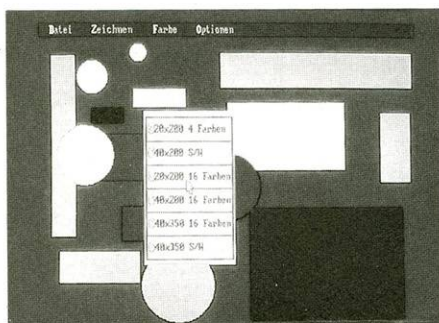
Listing 3. Das Prompt-Kommando übernimmt beim Einsatz der Standard-MS-DOS-Befehle viel Routinearbeit

DOS SOFTWARE TOOLS

JETZT NEU
Drei
Toolboxen
für GW-Basic

GW-Basic-Toolboxen

Warum kompliziert, wenn's doch so einfach geht! Dieser Wahlspruch gilt ab sofort auch für GW-Basic und SBasic, dem Compiler aus DOS-Extra 6/89. Verwenden Sie die neuen, leistungsfähigen GW-Basic-Toolboxen, und das Programmieren macht doppelt so viel Spaß. Trotz professioneller Leistung brauchen Sie auf Komfort nicht zu verzichten: Die Bedienung aller GW-Basic-Toolboxen gleicht der Bedienung des Basic-Interpreters "bis aufs Haar". Sie können problemlos mehrere GW-Basic-Toolboxen gleichzeitig laden und nutzen, denn es werden im Basic-Speicherbereich nicht mehr als zwei Bytes belegt. Mit jeder GW-Basic-Toolbox erhalten Sie ein komplettes, leistungsfähiges Beispielprogramm.



"Oberfläche"

Best.-Nr. 6009/6010

Mit dieser Toolbox realisieren Sie in GW-Basic eine komfortable zu bedienende Benutzeroberfläche mit Pull-down-Menüs und kompletter Mausunterstützung. Die Leistungsmerkmale:

- * Vollständige Maussteuerung im Text- und Grafikmodus
- * Spezielle Menüleisten für häufig benötigte Eingaben
- * Popup-Menüs mit Meldungen bis zu fünf Zeilen Länge
- * Position, Farben und Rahmen aller Menüs frei wählbar
- * Unterstützung aller Bildschirmmodi von GW-Basic
- * Inklusive einem fertigen Malprogramm als Beispiel

Jede Toolbox nur 69,- DM*

"DOS-Funktionen"

Best.-Nr. 6011/6012

Mit mehr als 30 DOS-Funktionen erlaubt Ihnen diese Toolbox von GW-Basic aus einen komfortablen Zugang zum MS-DOS-System. Die Leistungsmerkmale:

- * Kompatibel zu MS-DOS von Version 2.1 bis 4.01
- * Laufwerke wechseln, aktuelles Laufwerk ermitteln etc.
- * Suchen nach Dateien mit den Wildcards "*" und "?"
- * Beschleunigte Bildschirmausgabe über Systemfunktionen
- * Zugang zu wichtigen, internen Daten des MS-DOS-Systems
- * Benutzung des Druckprogramms Print von GW-Basic aus
- * Inklusive Beispielprogramm: Eine komplette DOS-Shell

"Hercules-Grafik"

Best.-Nr. 6013/6014

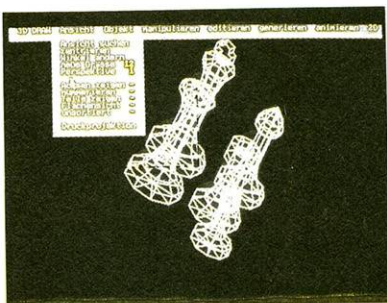
Mit dieser Toolbox sind die Zeiten vorbei, in denen von GW-Basic nur der Textmodus der Hercules-Grafikkarte unterstützt wurde: Mit der GW-Basic-Toolbox »Hercules-Grafik« programmieren Sie unter GW-Basic nun auch im Grafikmodus. Die Leistungsmerkmale:

- * Routinen zum Ein- und Ausschalten des Grafikmodus
- * Zwei verfügbare Grafikseiten mit Umschaltmöglichkeit
- * Funktionen für Linie, Bogen, Kreis, Ellipse und Viereck
- * Ausfüllen von Flächen mit beliebigen Punktmustern
- * Speichern und Restaurieren des Grafikbildschirms
- * Setzen, Löschen und Abfragen von einzelnen Bildpunkten
- * Emulation der GW-Basic-Befehle »View« und »Window«
- * Inklusive Beispielprogramm: Präsentationsgrafik

3D-Draw

Das komfortable CAD-Programm »3D-Draw« bietet verdeckte Flächen und, als Besonderheit, Lichtquellsimulation. Mit der integrierten Animationsroutine erzeugen Sie beliebig lange Filme. Die Leistungsmerkmale:

- * Ansicht durch Eingabe der Projektionswinkel frei wählbar * Auf Wunsch lassen sich drei verschiedene 2D-Ansichten erzeugen, die am unteren Bildschirmrand eingeblendet werden * 2D-Editor zur schnellen Konstruktion von Objekten * Integrierter Taschenrechner für Konstruktionsdaten * Einstellbare Größe und perspektivische Verzerrung * Kombination von getrennt konstruierten Teilen * 3D-Funktion als Manipulationswerkzeug * Punkte, Kanten und Flächen sind frei editierbar * Drei Raumachsen mit 1600-Ebenen-Editor * Verdeckte Flächen * Lichtquellsimulator * Animationsroutine mit Größen- und Perspektivveränderung von beliebiger Länge; ein eigener Animationslinker ist eingeschlossen * Verformen der Objekte durch Zufallszahl oder Verknüpfung mit mathematischen Formeln * Erzeugung von Rotations- und Translationskörpern und modifizierten Duplikaten * Programmierer-Schnittstelle, Objekte lassen sich mit einem Textprogramm oder Editor generieren * Vielzahl von Ausgabeformaten: 3D-Draw, Autocad-Script, Postscript, EPS-Format und HPGL-Plot-Format * Alle gängigen Grafikkarten werden unterstützt, komplette Ausnutzung der EGA-Palette mit 64 Farben



Systemvoraussetzungen:

Computer: PC, XT, AT oder Kompatibler mit mindestens 640 KByte Arbeitsspeicher, ein oder zwei Diskettenlaufwerke beziehungsweise ein Diskettenlaufwerk und eine Festplatte.

Grafikkarten: Hercules-, CGA-, EGA- oder VGA-Karte

Ausgabemedien: Epson-kompatibler Drucker, Ausgabe von Postscript-Dateien in Verbindung mit Postscript-Emulator, Postscript-fähiger Laserdrucker, Plotter mit HPGL-Standard.

Sonstiges: Microsoft-kompatible Maus unbedingt erforderlich, MS-DOS ab Version 3.0

nur 99,- DM*

Best.-Nr. 6007/6008

Resident-Software-Manager

Menügesteuertes Löschen und Nachladen von residenten TSR-Programmen löst Ihre Speicherplatzprobleme: RSM

- * zeigt den noch freien Arbeitsspeicher * registriert geladene residente Programme * erinnert an die Aktivierungstastenkombination * informiert über benutzte Interrupt-Vektoren * löscht beliebige, speicherresidente Programme * lädt mehrere TSR-Programme in einem Arbeitsgang * bietet umfassende, komfortable Menüsteuerung * besitzt einen universellen Maustreiber * belegt lediglich 15 KByte Speicherplatz * informiert mit einem umfangreichen Handbuch

Segment	Programm	Bytes	Befehl	(Vektoren mit Tab)	RSN	z.B.
0000	MAN128	14944	Residenter Programmteil des RSM			Laden
1100	KEYBR	2362	Deutscher Tastaturreis			489
1226	KEYBK	2280	Tastaturbeschleuniger			APPEND
1283	KEY-PAGE	710	Tastaturradar in Batch-Dateien			1 ASCII
1285	DROPTASK	464	<Ctrl>(<Alt>(<Shift>(<RSif>))			BREAKOUT
1381	ROGEDIT	2864	Kommandoeditor			2 CLOCK
1387	COMPRESS	12114	Copy-Compress: <Alt>-S			COMMON
1678	MOUSE	5084	Maustreiber			5 DOSEDIT
1706	PCTOOLS	65888	Disk-Utility: <Ctrl>(<Esc>)			5 DOSEDIT
2776	SPEED	394	DOS-Speedup			5 DOSEDIT
2818	GRAFTABL	1312	Grafikzeichnen			5 DOSEDIT
2863	GRAPHICS	3448	Grafik-Hardcopy			5 DOSEDIT
2938	APPEND	1776	DOS-Suchpfad für Batchdateien			5 DOSEDIT
2980	ASCII	21368	Sonderzeichen-Pickup: <Alt>-S			5 DOSEDIT
2984	CLOCK	880	Bildschirm			5 DOSEDIT
2719	HYPERKEY	7496	Tastaturradar: <Shift>(<F1>)			5 DOSEDIT
4186	LX	7908	Schreibtreiber für 9-Adre			5 DOSEDIT
5407	PAGES	416	Videoschreibtreiber: <Alt>(<F1>)			5 DOSEDIT
5404	QUICKDIR	55952	Directory-Utility: <Alt>(<F2>)			5 DOSEDIT

Für alle PCs, XTs, ATs und Kompatible mit mindestens 256 KByte Arbeitsspeicher

Best.-Nr. 6001/6002

nur 69,- DM*

PC-Utilities

nur 69,- DM*

Best.-Nr. 6003/6004

Call: verschachtelt Aufrufe von Batch-Dateien. Code: Paßwortverschlüsselung. Makeroom: löscht Dateien gemäß Suffix. Setdir, Restdir: aktivieren nach Programmaufruf das Original-Verzeichnis. Ty: Hex-Dump, Binärfeld oder Zeilennummer auf dem Drucker oder Bildschirm aus. Extras: »Scopy«, das professionelle Kopierprogramm, »Filed«, der Editor, dem nichts verborgen bleibt.

Softreset: Spannen Sie Ihre Nerven nicht weiterhin auf die Folter! Mit Softreset bleibt der komplette Speicherinhalt auch nach einem Reset erhalten und kann auf Diskette oder Festplatte gespeichert werden. Zeilteile PC: Keine Angst vor Geschwindigkeit! Mit Zeilteile PC verlangsamen Sie Ihren Computer bis zu einem Bruchteil der Taktfrequenz eines 4.77-MHz-PC. Mit Boot-Option! Extras: Der universell einsetzbare AT-Tastaturreis »T6« mit Quellcode. Passen Sie Ihre Tastatur an Ihre persönlichen Bedürfnisse an!

MS-DOS-Utilities

nur 49,- DM*

Best.-Nr. 6005/6006

* Sämtliche Programme sind wahlweise auf 5 1/4-Zoll- oder 3 1/2-Zoll-Diskette erhältlich. (Bitte bei Bestellung unbedingt angeben.) Unabhängig von der Anzahl der bestellten Artikel berechnen wir für das Inland 4,- DM bzw. für das Ausland 6,- DM Porto und Verpackung. Unverbindliche Preisempfehlung. Bitte benutzen Sie die Bestellkarte.

Wie sag ich's meiner Batch-Datei

Leider bietet die Batch-Sprache von MS-DOS keine Möglichkeit, eine Eingabe des Benutzers abzufragen. Dieses Problem ist hinlänglich bekannt und so verwundert es nicht, daß schon eine Anzahl an Lösungen dafür existiert: Die kürzeste davon nur sechs Byte lang.

Um einer Stapeldatei etwas mitzuteilen wird im allgemeinen ein Com-Programm geschrieben, das die Eingabe des Benutzers liest und das Zeichen der Fehlervariablen »errorlevel« zuweist. Diese Variable läßt sich dann in der Batch-Datei abfragen. Die kürzeste Lösung hierzu lautet:

```
mov ah,08h
int 21h
mov ah,4ch
int 21h
```

Das Programm »input.com« (Listing 1 und 2) bietet Ihnen allerdings mehr Komfort. Die Syntax beim Aufruf ist

```
input ["<Dialogstring>",<zulässige Zeichen>]
```

Sie dürfen den Dialogstring einschließlich zugehörigem Komma natürlich auch weglassen, wenn Sie die Eingabe am Zeilenanfang wünschen.

Der zweite Vorteil des Programms ist die Angabe der zulässigen Zeichen. Die Variable »errorlevel« enthält nach der Eingabe nicht mehr die gedrückte Taste, sondern die

Nummer in der Parameterzeile, beginnend ab eins. Somit entfällt die umständliche Abfrage auf gültige Eingaben in der Batch-Datei. Auch um Groß- und Kleinschreibung brauchen Sie sich nicht mehr zu kümmern, denn Input konvertiert alle Eingaben in Großschrift. Zur Verdeutlichung folgt als Beispiel ein kleines Menü:

```
@echo off
:menue
cls
echo (B)asic
echo.
echo (D)atenfernübertragung
echo.
echo (T)extverarbeitung
echo.
input "Ihre Wahl (ESC=MS-DOS): ",bdt^[
if errorlevel==4 goto ende
if errorlevel==3 goto text
if errorlevel==2 goto dfue
gwbasic
goto menue
:dfue
procomm
goto menue
:text
word
goto menue
:ende
echo.
```

Wie Sie an diesem Beispiel sehen, hat das Hochzeichen »^« eine besondere Bedeutung: Es steht als Kennzeichen vor einem Ctrl-Code (ASCII-Wert 0 bis 31). Diese Zeichen werden nicht automatisch ausgegeben, denn ein beim Drücken der ESC-Taste erscheinender Pfeil nach links ist nicht sinnvoll.

(Wolfgang Lorenz/zi)

Name	input.asm
Sprache	MASM ab Version 4.0, TASM
Grafikkarte	jede
Besonderheiten	keine

```
[384] 1: ; Programmname: INPUT.ASM
[WY3] 2: ; (C) 1989 Everts & Hagedorn
[8U3] 3: ; DOS International
[463] 4: ; -----
[0Y2] 5: ; Assemblieren mit
[3V3] 6: ; Microsoft Assembler:
[AC2] 7: ; MASM INPUT;
[YC2] 8: ; LINK INPUT;
[UD4] 9: ; EXE2BIN INPUT INPUT.COM
[P30] 10: ;
[G33] 11: ; Turbo Assembler:
[4B2] 12: ; TASM INPUT
[2Y2] 13: ; TLINK INPUT /T
[463] 14: ; -----
[000] 15: ;
[H22] 16: Code Segment
[EA2] 17: Assume CS:Code
[A61] 18: Org 100h
[000] 19:
[201] 20: Begin:
[K95] 21: ;Länge der Kommandozeile holen
[DL1] 22: Mov SI,128
[CV0] 23: Lodsb
[KF1] 24: Mov CL,AL
[6F1] 25: Xor CH,CH
[V36] 26: ;Führende Leerzeichen überlesen
[TH1] 27: LeseLeer:
[0N1] 28: Call GetCh
[2G6] 29: ;Wenn kein Parameter angegeben ist, dann
[UC4] 30: ;kurze Erklärung ausgeben
[712] 31: Je Anleitung
[ZD1] 32: Cmp AL,' '
[LX1] 33: Je LeseLeer
[FY5] 34: ;Ist das erste Zeichen ein Hochkomma?
[YB1] 35: Cmp AL,34
[BT2] 36: Jne LeseZeichen
[836] 37: ;Wenn ja, dann Dialogstring lesen und
[6H1] 38: ;ausgeben
[442] 39: LeseString:
[0N1] 40: Call GetCh
[7C6] 41: ;Bei unerwartetem Ende des Textes raus
[451] 42: Je Ende
```

```
[Y75] 43: ;Endekennzeichen ist Hochkomma
[YB1] 44: Cmp AL,34
[B02] 45: Je LeseKomma
[RN4] 46: ;Ansonsten Zeichen ausgeben
[MF1] 47: Mov DL,AL
[V91] 48: Mov AH,2
[381] 49: Int 21h
[RL2] 50: Jmp LeseString
[5W5] 51: ;Solange lesen, bis Komma oder Ende
[CT1] 52: LeseKomma:
[0N1] 53: Call GetCh
[451] 54: Je Ende
[RE1] 55: Cmp AL,', '
[G92] 56: Jne LeseKomma
[0N1] 57: Call GetCh
[451] 58: Je Ende
[WC6] 59: ;Position der gültigen Zeichen sichern
[L52] 60: LeseZeichen:
[401] 61: Dec SI
[RF1] 62: Mov DI,SI
[P11] 63: Inc CX
[Eh1] 64: Mov BX,CX
[UB1] 65: Eingabe:
[VU4] 66: ;Auf Benutzereingabe warten
[1M1] 67: Mov AH,08h
[381] 68: Int 21h
[K94] 69: ;Steuerzeichen überlesen
[BC1] 70: And AL,AL
[MG2] 71: Jne KeinSteuer
[1M1] 72: Mov AH,08h
[381] 73: Int 21h
[V02] 74: Jmp Eingabe
[A22] 75: KeinSteuer:
[206] 76: ;Zeichen in Großbuchstaben wandeln
[P02] 77: Call Uppcase
[642] 78: ;Und sichern
[MF1] 79: Mov DL,AL
[AG7] 80: ;Position der gültigen Zeichen zurückholen
[6G1] 81: Mov SI,DI
[PH1] 82: Mov CX,BX
[NV1] 83: Mov BP,4C00h
[0D6] 84: ;Eingegebenes Zeichen mit zulässigen
[SE3] 85: ;Zeichen vergleichen
[PS1] 86: Vergleich:
[6L4] 87: ;Positionszeiger erhöhen
[W01] 88: Inc BP
[0N1] 89: Call GetCh
[B06] 90: ;Zeichen war nicht korrekt, dann neu
[XF1] 91: ;eingeben
[MK1] 92: Je Eingabe
[P02] 93: Call Uppcase
[084] 94: ;If Ctrl-Zeichen prüfen
```


HYPERKEY 4.0

Der Tastatur-Manager

```

[VH1] 95:  Cmp AL, ''
[242] 96:  Jne KeinCtrl
[6F6] 97:  ;Buchstaben holen und nach 0..31 wandeln
[0N1] 98:  Call GetCh
[MK1] 99:  Je Eingabe
[7A1] 100: And AL, 31
[NV2] 101: ;Eingabe korrekt?
[1L1] 102: KeinCtrl:
[2D1] 103:  Cmp AL, DL
[U82] 104:  Jne Vergleich
[NL1] 105:  Mov AH, 02h
[NV6] 106: ;Zeichen auf Bildschirm ausgeben, wenn es
[2E3] 107: ;kein Steuercode ist
[UB1] 108:  Cmp AL, 32
[VM1] 109:  Jb NewLine
[3B1] 110:  Int 21h
[XE1] 111: NewLine:
[3X4] 112: ;Zeile mit CRLF abschließen
[KD1] 113:  Mov DL, 13
[3B1] 114:  Int 21h
[CD1] 115:  Mov DL, 10
[3B1] 116:  Int 21h
[DM5] 117: ;Index der Eingabe zurückliefern
[VG1] 118:  Mov AX, BP
[3B1] 119:  Int 21h
[WR7] 120: ;Beim Fehlen von Parametern kurze Erklärung
[DH1] 121: ;anzeigen
[DU1] 122: Anleitung:
[663] 123:  Mov DX, Offset Text
[3M1] 124:  Mov AH, 09h
[3B1] 125:  Int 21h
[XT0] 126: Ende:
[BW1] 127:  Mov AX, 4C00h
[3B1] 128:  Int 21h
[000] 129:
[1K7] 130: ;Unterprogramm holt Zeichen in AL und setzt
[MS3] 131: ;bei Ende das Zero-Flag
[E11] 132: GetCh:
[MF1] 133:  Xor AL, AL
[S02] 134:  Jcxz GetEnd
[CV0] 135:  Lodsb
[H01] 136:  Dec CX
[BC1] 137:  And AL, AL
[2A1] 138: GetEnd:
[3P0] 139:  Ret
[000] 140:
[0S7] 141: ;Unterprogramm wandelt AL in Großschrift um
[TU6] 142: ;unter Berücksichtigung der deutschen
[BG1] 143: ;Umlaute
[4A1] 144: Uppcase:
[1G1] 145:  Cmp AL, 'a'
[6Z1] 146:  Jb KeinKlein
[LH1] 147:  Cmp AL, 'z'
[3Z1] 148:  Ja KeinKlein
[UB1] 149:  Sub AL, 32
[3P0] 150:  Ret
[SR1] 151: KeinKlein:
[VM1] 152:  Cmp AL, 'ä'
[UP1] 153:  Jne KeinAE
[DP1] 154:  Mov AL, 'Ä'
[3P0] 155:  Ret
[G81] 156: KeinAE:
[HN1] 157:  Cmp AL, 'ö'
[PR1] 158:  Jne KeinOE
[3R1] 159:  Mov AL, 'Ö'
[3P0] 160:  Ret
[C91] 161: KeinOE:
[LN1] 162:  Cmp AL, 'ü'
[2S1] 163:  Jne KeinUE
[5R1] 164:  Mov AL, 'Ü'
[S91] 165: KeinUE:
[3P0] 166:  Ret
[000] 167:
[296] 168: Text DB 13, 10, 'INPUT ["Dialogstring",] '
[VS4] 169:  DB 'Zulässige Zeichen', 13, 10, '$'
[000] 170:
[TF1] 171: Code Ends
[SF1] 172:  End Begin

```

Listing 1. Der Quelltext des Utility »input.asm«

```

[BN5] 1: BE 80 00 AC 8A C8 32 ED E8 81 00 74 73 3C 20 74
[KG5] 2: F7 3C 22 75 1F E8 74 00 74 6D 3C 22 74 08 8A D0
[GT5] 3: B4 02 CD 21 EB EF E8 63 00 74 5C 3C 2C 75 F7 E8
[9P5] 4: 5A 00 74 53 4E 8B FE 41 8B D9 B4 08 CD 21 22 C0
[2V5] 5: 75 06 B4 08 CD 21 EB F2 E8 4A 00 8A D0 8B F7 8B
[9K5] 6: CB BD 00 4C 45 E8 34 00 74 E0 E8 38 00 3C 5E 75
[GF5] 7: 07 E8 28 00 74 D4 24 1F 3A C2 75 E8 B4 02 3C 20
[1N5] 8: 72 02 CD 21 B2 0D CD 21 B2 0A CD 21 8B C5 CD 21
[9L5] 9: BA B5 01 B4 09 CD 21 B8 00 4C CD 21 32 C0 E3 04
[WH5] 10: AC 49 22 C0 C3 3C 61 72 07 3C 7A 77 03 2C 20 C3
[2H5] 11: 3C 84 75 03 B0 8E C3 3C 94 75 03 B0 99 C3 3C 81
[RG5] 12: 75 02 B0 9A C3 0D 0A 49 4E 50 55 54 20 5B 22 44
[SK5] 13: 69 61 6C 6F 67 73 74 72 69 6E 67 22 2C 5D 20 5A
[EG5] 14: 75 6C 84 73 73 69 67 65 20 5A 65 69 63 68 65 6E
[UX0] 15: 0D 0A 24

```

Listing 2. »input.com« als Hexdump-Listing

HYPERKEY Plus Makroprozessor 4.0 INS

Taste	Name	Makro	Status	Chaining	Aufnehmen	Pausen	Kopieren	Beenden
Alt-LShift-RShift-F6		Noch besser - Die neue Version 4.0	*					
F7		"Directory"- das erste Makro	*					
Alt-C		"Sondertasten"-Makro	*					
Alt-LShift-RShift-A		"Aufnahme"-Makro	*					
LShift-RShift-P		"Pause"-Makro	*					

Makro Editor

Taste: Status:

Name: Chaining:

Makro:

Cursor: Code: Zeichen:

Cursorstasten: Makro editieren F10:Menu Esc:Abbrechen Return:Speichern+Ende

Nutzen Sie die Vorteile Ihres PC!

Mit dem speicherresidenten Makroprozessor HYPERKEY können alle Tastaturtypen mit beliebigen, verkettbaren Zeichenfolgen belegt werden: Anwendungen, Kürzel, Programmschablonen, etc. erscheinen auf Tastendruck. Komfortables Editieren, Abspielen, Speichern, Sortieren und Suchen der Belegungen erfolgt über eine ergonomische Benutzeroberfläche mit dreistufiger Auswahlliste und kontextsensitiver Hilfestellung. Minimaler Speicherplatzbedarf wird durch optimiertes Programmlayout, automatische Expanded-Memory-Unterstützung, integrierte DeInstall- und Speicherreservierungsfunktion und den Mini-Makroprozessor TINYKEY/PLAYBACK gewährleistet.

Weiterhin neu in der Version 4.0:

- Jetzt auch Zusammenarbeit mit Programmen, die im PC-Grafikmodus arbeiten und zwischenzeitlich den Tastaturpuffer löschen
- Unterstützung von MF-2-Tastaturen und Einstellung aller Tastaturtypen von XT/AT in Deutsch/Englisch durch einen Treiber-Generator
- Unterscheidung der Shifttasten rechts/links und beliebige Kombinationsmöglichkeiten der Belegungen mit ALT und CONTROL
- Verbesserte Editiermöglichkeiten für Makros: Kopieren, Packen, Verschieben und Kontrolle der Speicher-Ressourcen
- Steuerung und Konfigurieren von HYPERKEY durch Aufrufparameter, zum Beispiel für das automatische Laden einer Belegung beim Start
- Makro-Abspielkontrolle aus externen Programmen mit frei wählbarer, ständig sichtbarer Statusanzeige auf dem Monitor
- Quit-and-Play-Funktion und Ausgabe der Makrobelegungen auf dem Drucker oder in eine Textdatei
- Konvertierungsprogramm für Tastaturbelegungen von Hyperkey Plus auf die neue Version und ausführliche Bedienungsanleitung in Deutsch
- Inklusive universell verwendbarem Maustreiber

für alle PC/XT/AT unter MS-DOS

mit minimal 256 KByte RAM, DOS ab Version 2.11 bis 3.3

Und das alles zum alten Preis!

Zudem bieten wir gegen Einsendung des Handbuchs von Hyperkey Plus ein super-günstiges Upgrade auf HYPERKEY 4.0 an.

HYPERKEY 4.0

5 1/4"-Disk. Best-Nr. 229 3 1/2"-Disk. Best-Nr. 235

99,- DM (unverbindliche Preisempfehlung)

Wenn Sie über den DMV-Bestellservice bestellen, gilt folgendes:

Inland:		Ausland:	
Einzelpreis	99,- DM	Einzelpreis	99,- DM
zzgl. Versandkosten	4,- DM	zzgl. Versandkosten	6,- DM
Endpreis	103,- DM	Endpreis	105,- DM

Upgrade von 3.0

5 1/4"-Disk. Best-Nr. 2601 3 1/2"-Disk. Best-Nr. 2602

29,- DM (unverbindliche Preisempfehlung)

Wenn Sie über den DMV-Bestellservice bestellen, gilt folgendes:

Inland:		Ausland:	
Einzelpreis	29,- DM	Einzelpreis	29,- DM
zzgl. Versandkosten	4,- DM	zzgl. Versandkosten	6,- DM
Endpreis	33,- DM	Endpreis	35,- DM

Bitte benutzen Sie die Bestellkarte.

Datenstapelei

Ein Datenbankprogramm mit Stapeldateien

Für die kleine Adressen- und Telefonnummernverwaltung benötigen Sie kein umfangreiches Datenbankprogramm. Eine trickreiche Kombination von Stapeldateien und ein paar MS-DOS-Hilfsprogramme genügen dazu völlig. Der Vorteil dieser Methode besteht darin, daß Sie die Programme jederzeit an Ihre persönlichen Vorstellungen anpassen können.

»DBTom« ist eine richtige Miniaturdatenbank, die zeigt, daß einfachste Hilfsmittel – in diesem Falle nur die MS-DOS-Befehle »copy«, »find«, »sort«, »more« und »edlin« – genügen, um eine komplette Datenverwaltung aufzubauen. Nun sind Programme, die einfach nur eine Textdatei durchsuchen, zwar nicht besonders schwierig zu schreiben, aber haben Sie schon einmal versucht, aus einer Adreßkartei alle die Daten herauszusuchen, die sowohl »x« als auch »y« oder »z«, nicht aber »a« oder »b« enthalten?

Der Autor sah keinen Grund, für einfaches Textsuchen eine große Datenbank zu laden und die Texte zu konvertieren oder sich durch ellenlange Listen der Fundstellen zu kämpfen. So entstand dieses kurze Programmpaket. Um es nutzen zu können, müssen Sie zunächst einmal die Listings 1 bis 10 sowie die Hilfstexte (Bild 1 bis 4) abtippen, die von den einzelnen Programmteilen bei Bedarf per Type-Kommando ausgegeben werden. Die Textdateien geben Sie am besten mit einer Textverarbeitung ein, notfalls genügt auch der MS-DOS-Editor »Edlin«. Die Textdateien haben keine Namenserverweiterung.

Alle Dateien kommen in ein Unterverzeichnis namens »\dbtom1«. Wenn Sie die Datenbank einmal auf einen anderen Rechner portieren möchten, dann kopieren Sie die Listings 1 bis 10 und die entsprechenden Textdateien auf eine Diskette. Nach der Eingabe von

dbtom1 c:

installieren sich die Stapeldateien selbständig in einem neuen Unterverzeichnis auf der Festplatte.

Wenn Sie möchten, können Sie »DBTom« auch auf einer Diskette installieren, auf einer Festplatte läuft es aber natürlich wesentlich schneller. Gestartet wird die ganze Datenbank mit

dbtom1 <Datei>

wobei Sie für »Datei« den Namen einer Datenbank (zum Beispiel »Adress«) eingeben. Existiert diese noch nicht, dann wird sie automatisch angelegt. Das Prompt-Symbol ändert sich in eine kurze Hinweismeldung, um Ihnen zu zeigen, daß Ihr PC gerade für die Datenbankanwendung vorbereitet ist. Falls Sie einmal die Fehlermeldung

Umgebungsbereich erschöpft

erhalten sollten, müssen Sie folgende Zeile in Ihre Systemdatei »config.sys« einfügen und den Rechner neu hochfahren:

device command.com /p /e:1024

Dadurch erhöhen Sie den Platz für Umgebungsvariablen auf 1024 Byte. Beachten Sie bitte auch, daß Sie das Unter-

Sie können folgende Befehle benutzen:

DBTOM1	[d:]	Beginnt die Arbeit mit dem Micro-Datenbanksystem
DATEI	file	Setzt neue Datenbank aktiv
DRUCKE	text	Druckt alle oder einzelne Daten auf dem Drucker aus
SUCHE	text	Gibt alle oder einzelne Daten auf dem Bildschirm aus
ZÄHLE		Zählt die Anzahl der eingegebenen Datensätze
ERGÄNZE	xxx	Fügt einzelne oder mehrere Daten ein
LÖSCHE	xxx	Löscht einzelne oder mehrere Daten
ENDE		Beendet die Arbeit mit der Datenbank

Mit "Befehl ?" können Sie eine kurze Erklärung zu dem Befehl abrufen.

Bild 1. Die Datei »Befehle« enthält eine Übersicht über die Kommandos der Mini-Datenbank. Bitte geben Sie diesen Text mit einem Editor oder einer Textverarbeitung ein.

verzeichnis »\dbtom1« nicht verlassen dürfen, wenn Sie noch mit der Datenbank weiterarbeiten möchten.

Selbst wenn Sie das Verzeichnis in den MS-DOS-Pfad eingebunden haben, erwarten die Programme bestimmte Zusatzdateien im aktuellen Verzeichnis, und das muß dann »\dbtom1« lauten.

Durch die vielen, umfangreichen Hilfstexte und die Kommentare innerhalb der Programme ist das ganze System selbsterklärend. Insbesondere der einfache, klare Aufbau der vielseitig einsetzbaren Datei »suche.bat« zeigt, wie mit einfachen Mitteln eine vollständige Textabfrage realisiert werden kann. Die restlichen Dateien sind eigentlich nur der Vollständigkeit halber beigefügt.

»suche.bat« in Verbindung mit der Hilfsdatei »Suche« ist der eigentliche Kern des Systems. Die einzelnen Parameter werden in dem Hilfstext »Suche« (Bild 4) ausführlich erläutert. Zunächst wird in »suche.bat« der obligate Test, ob der Anwender eine Erklärung wünscht, durchgeführt. Von früheren, vorzeitig mit [Ctrl-C] abgebrochenen Aufrufen übriggebliebene Suchdateien »dbsuch.alt« werden nun gelöscht. Dann wird eine neue Suchdatei »dbsuch« entweder ohne oder mit Zeilennummern angelegt und zur Suche sogleich als alte Suchdatei »dbsuch.alt« deklariert. Schließlich wird getestet, ob der Anwender überhaupt einen Suchtext angegeben hat. Falls nicht, wird ab der Marke »:alles« die gesamte Datei seitenweise ausgegeben.

Nun geht es ab der Marke »:suche« weiter. In einer Schleife bis vor die Marke »:gefunden« werden die Suchparame-

KORRIGIERE.BAT ermöglicht die Korrektur von Daten mit Hilfe des MS-DOS-Editors EDLIN
Syntax: KORRIGIERE

Tastenkombis:
" - " bewegen die Schreibmarke vor / zurück
F1 F2x F3 ein / bis Zeichen x / alle Zeichen übernehmen
F4x bis Zeichen x löschen
Ins Einfg Einfügemodus anschalten (bis andere Befehlstaste gedrückt wird)

Eingabekommandos: (a,b,c stehen für die Nummern der Zeilen)
C Kopieren der Zeilen a-b nach c a,b,c,C
D Löschen der Zeile a oder von a bis b aD oder a,bD
E Bearbeiten von Zeile a a
E Ende der Korrektur E
I Einfügen von Zeilen vor Zeile a (Ende mit CTRL-C) aI
L Anzeigen der Zeilen a-b aL
M Verschieben der Zeilen a-b nach c a,bL
P Anzeigen der Zeilen a-b seitenweise a,b,CM
Q Verlassen des Editors ohne Übernahme der Änderungen a,bP
R Ersetzen von Text t1 durch t2 in den Zeilen a-b a,b,Rt1F6t2
?R Ersetzen von Text t1 durch t2 in den Zeilen a-b a,b,?Rt1F6t2
S Suchen von Text t in den Zeilen a-b a,b,St
?S Suchen von Text t in den Zeilen a-b a,b,?St
T Übernehmen der Daten aus Datei b ab Zeile a aTb
Seltener benutzt: (Nur bei Dateien über 65529 Zeilen oder 65535 Zeichen)
A Laden weiterer a Zeilen aA
W Speichern von a Zeilen, um weitere laden zu können aW

Bild 2. Einen Hilfstext für den Befehl »korrigie.bat« enthält die Textdatei »Korrigie«

LÖSCHE.BAT löscht ein oder mehrere Daten
Syntax: LÖSCHE [(SatzNr (SatzNr)) | /T text | /R | /![]]

Beispiele:
LÖSCHE Alles wird gelöscht.
LÖSCHE /R Der letzte Aufruf von Löschen wird rückgängig gemacht
LÖSCHE /T Schade Alle Familien Schade werden gelöscht
LÖSCHE 4 Der Datensatz Nummer 4 wird gelöscht
LÖSCHE 5 6 7 Die Datensätze 5 bis 7 einschließlich werden gelöscht
LÖSCHE /! Datensätze mit Nummern ausgeben und Datei sortieren
LÖSCHE /! Datensätze mit Nummern auf dem Drucker ausgeben
LÖSCHE /T Schade, Melanie Der (oder die) Einträge über Melanie Schade werden gelöscht

Bild 3. In der Textdatei »Lösche« steht ein Hilfstext zum Befehl »lösche.bat«

ter analysiert, und die Suche wird folgendermaßen ausgeführt: Die Oder-Funktion (»/O«) benutzt immer die alte Suchdatei. Die anderen Optionen erhalten zunächst eine aktualisierte Fassung von »dbsuch.alt«, indem (sofern bereits etwas gesucht wurde) die alte Suchdatei gegen die neu angelegte ausgetauscht wird. Dann prüft »suche.bat« auch den Parameter »/N« und verzweigt gegebenenfalls. Ist kein Parameter angegeben worden, so handelt es sich um eine Und-Verknüpfung. Das bedeutet, daß aus den bereits gefundenen Daten all jene herausgesucht werden, die auch den nächsten, in der Kommandozeile angegebenen Text enthalten.

Ist ein »Nicht-Parameter« (»/N«) in der Parameterzeile enthalten, so wird dieser zunächst mittels »shift« (Marke »:wegfiltern«) aus der Parameterliste gelöscht. Dann werden aus den bereits gefundenen Daten mittels des Find-Befehls all jene Datenzeilen herausgesucht, die den angegebenen Text nicht enthalten.

Wenn der Anwender die Oder-Funktion aktiviert hat (»/O«), so wird ab der Marke »:zufuegen« noch einmal auf eine Kombination von »oder« und »nicht« (»/O« und »/N«) getestet. Falls dieses gewünscht wurde, geht es bei »:nichtzufuegen« weiter. Ansonsten werden aus der alten Datei »dbsuch.alt« die Datenzeilen zu den bereits gefundenen in der Datei »dbsuch« hinzugefügt. Weil MS-DOS leider keinen Befehl zur Erkennung von doppelten Datenzeilen kennt, erscheinen hierdurch natürlich Daten unter Umständen mehrfach. Aber dies kann, je nach Anwendung, ja durchaus erwünscht sein.

Schließlich treffen sich alle Bearbeitungswege wieder bei der Marke »:weiter«. Dort wird mit »shift« der gerade gesuchte Text aus der Parameterliste gelöscht (und damit ne-

```
SUCHE.BAT sucht einzelne oder mehrere Daten.
Syntax: SUCHE [/Z] { [/O [/N] ] Text }
Parameter:  /Z Zeilennummern der Datenbank anzeigen
            /O Oder-Verknüpfung von Daten
            /N Nicht, Ausschluß von Daten
Kein Parameter: Und-Verknüpfung von Daten

Beispiele:
*****
*** Alle Daten suchen / drucken:
(DRUCKE ist eine SUCHE mit Ausgabe auf dem Drucker)
SUCHE

*** UND-Verknüpfung:
Benutzen Sie bitte folgenden Aufruf, wenn Sie Daten suchen, die sowohl den Text "dat1" als auch "dat2" als auch "datx" usw. enthalten:
SUCHE dat1 dat2 datx ...
Um Herrn oder Frau Schade aus Braunschweig zu suchen:
SUCHE Schade Braunschweig
(Sie dürfen abkürzen, wenn die Abkürzung eindeutig ist.)

*** Ausschließung:
Wenn die Daten einen bestimmten Text (z.B. "dat1") NICHT enthalten sollen, geben Sie bitte "/N dat1" ein.
Also alle Braunschweiger außer allen Familien Schade:
SUCHE Braunschweig /N Schade oder
SUCHE /N Schade Braunschweig

*** ODER-Verknüpfung:
Sollen die Daten den Text "dat1" und "dat2" ODER "dato" enthalten, geben Sie statt "dato" bitte "/O dato" ein.
Alle Familien namens Schade und alle Frauen namens Melanie findet also:
SUCHE Schade /O Melanie

Die Abarbeitung der Suchformel geschieht von links nach rechts. Ein ODER greift dabei auf die Daten der letzten UND-Verarbeitung zurück.
```

Bild 4. Auch der Hilfstext für den Befehl »suche.bat« kommt in eine eigene Textdatei namens »Suche«

benbei die Gesamtzahl der Parameter und Suchtexte auf über zehn Texte erhöht). Nach einer Abfrage, ob der Anwender noch einen Parameter angegeben hat, wird entweder nach diesem Text gesucht, oder das Programm gibt ab der Marke »:gefunden« die gefundenen Einträge sortiert und seitenweise aus. Die Datei »dbsuch« bleibt dabei erhalten, um sie bei Bedarf mittels »type« oder »print« noch

Ist Ihr Programm der HIT?

Der DMV-Verlag sucht ständig nach neuer, interessanter Software zur Aufnahme in unser Softwaresortiment.

Dabei ist es einerlei, ob Sie nun ein Anwendungs- oder ein Spielprogramm geschrieben haben.

Der DMV-Verlag bietet Ihnen sein Software-Know-how an!

24-Nadeldrucker - ein Problem?

Gleich welchen Computer-Typ Sie benutzen, für keinen 24-Nadeldrucker gibt es in jeder Software einen geeigneten Druckertreiber, der alle 24 Nadeln steuert. Die wenigen SW-Treiber sind teilweise kompliziert zu installieren und blockieren den Rechner für neue Aufgaben. Um diesem Problem zu begegnen, haben wir eine Hardware-Lösung entwickelt, die durch Auslagern der Berechnungsroutine in einen integrierten Puffer unseres Hardware-Emulators die Konvertierung schnell und komfortabel handhabt. Selbstverständlich haben wir die Gelegenheit genutzt und weitere Features eingebaut, die ein noch bequemer Arbeiten ermöglichen. Um die weltweit unterschiedlichen Systeme von Schnittstellen ausreichend zu berücksichtigen, haben wir für den C64/128-Bereich das **PRINTERFACE**. Für parallele Anschlüsse (Centronics), vorkommend z.B. bei Schneider CPC, Atari, AMIGA oder PCs, bieten wir den **BITMASTER** an. Bei den NEC-Pinwritern P2200/P2+ bzw. P6/7+ sowie dem CITIZEN SWIFT 24 gibt es eine Integrallösung namens **PRINTERJET**.

Gerne schicken wir Ihnen bei Einsendung eines DIN-A4-Kuverts, frankiert mit DM 4,- und unter Beilage dieser Anzeige, eine komplette Infomappe mit zahlreichen Testberichten zu.

DOS Extra 9/90

R K T GmbH

Postfach 71 08 44 • 8000 München 71
Tel. 0 89 - 79 51 10 • Fax 0 89 - 7 59 51 50

einmal ausgeben zu können oder um sie nach einer Umbenennung zur Grundlage weiterer Suchaktionen zu machen.

Der eigentliche Trick, der dafür sorgt, daß bei geeigneter Abfragetechnik mit dieser Methode tatsächlich alle Abfragen realisiert werden können, liegt also in den Oder-Befehlen, die nicht auf die letzten, sondern auf frühere Ergebnisse der Suche zurückgreifen.

Mit ein wenig Boolescher Algebra lassen sich hiermit sogar Abfragen wie beispielsweise

Suche alle Artikel (aus einer Literaturverwaltung), die die Stichworte Grafik und EGA betreffen, nicht aber die Artikel über Hercules und Maussteuerung unter Grafik

formulieren. Zum besseren Verständnis: das umgangssprachliche »Grafik und EGA« ist gleichbedeutend mit der logischen Verknüpfung »Grafik oder EGA«. Die Aussage »nicht (Hercules und Maus)« wird ausgeklammert, was die äquivalente Formulierung »nicht Hercules oder nicht Maus« ergibt. Somit lautet der Suchbefehl:

suche Grafik /O EGA /N Hercules /O /N Maus

(Thomas Stein/tr)

```
[FD1] 1: @echo off
[4V4] 2: rem Aufrufmodul für DBTOM1
[8A7] 3: rem DBTOM1.BAT bereitet die Micro-Datenbank
[AA7] 4: rem "DBTOM1" durch einen Verzeichniswechsel
[SK5] 5: rem vor oder installiert "DBTOM1"
[PN0] 6: CLS
[MBL] 7: echo
[7T4] 8: echo
[S85] 9: echo
[XW6] 10: echo
[826] 11: echo
[7T4] 12: echo
[W46] 13: echo
[7T4] 14: echo
[ECL] 15: echo
[YS0] 16: echo.
[B34] 17: if %1 == .? goto erklarung
[LP4] 18: if %1 == .b: goto installieren
[LP4] 19: if %1 == .B: goto installieren
[MP4] 20: if %1 == .c: goto installieren
[MP4] 21: if %1 == .C: goto installieren
[NP4] 22: if %1 == .d: goto installieren
[NP4] 23: if %1 == .D: goto installieren
[F56] 24: if exist %1\DBTOM1\DBTOM1 goto aufruf
[CS4] 25: if exist %1\DBTOM1 goto aufruf
[000] 26:
[BA2] 27: :INSTALLIEREN
[W87] 28: echo Das Verzeichnis %1\DBTOM1 wird erzeugt
[4N8] 29: echo und alle Dateien werden von Laufwerk A
: kopiert
[MW0] 30: pause
[NF1] 31: CTTY NUL
[DY1] 32: md %1\DBTOM1
[5D3] 33: copy /A a:*. * %1\DBTOM1
[2P3] 34: copy /A a:DBTOM1.BAT %1\
[2E7] 35: copy /B a:NEUDATEI.DAT %1\DBTOM1\NEUDATEI.D
AT
[BF6] 36: copy /B a:NEUDATEI.DAT %1\DBTOM1\DBTOM1
[YD1] 37: CTTY CON
[G90] 38: A:
[2DC] 39: for %%a in (*.*) DO if not exist %1\DBTOM1\
%%a echo Fehler! %%a nicht installiert!
[X60] 40: %1
[WW4] 41: del %1\DBTOM1\DBTOM1.BAT >NUL
[000] 42:
[CC1] 43: :AUFRUF
[HE2] 44: CD %1\DBTOM1 >NUL
[FS6] 45: if not exist NEUDATEI.DAT goto erklarung
[UH4] 46: if not %1 == . set DBTOM1DAT=%1
[ZN7] 47: if %1\DBTOM1DAT% == . set DBTOM1DAT=%1\
DBTOM1
[XBH] 48: if not exist %1\DBTOM1DAT% echo Die Datei "%1\
DBTOM1DAT%" existiert nicht. Bitte benutzen
Sie den Befehl "DATEI"!
[XSD] 49: if exist %1\DBTOM1DAT% echo DBTOM1 wird mit d
er Datensammlung "%1\DBTOM1DAT%" aktiviert.
[0U7] 50: if %1\DBTOM1PROMPT% == . set DBTOM1PROMPT=%P
ROMPT%
[3BA] 51: PROMPT DBTOM1 $b Micro-Datenbanksystem, Uhr
zeit: $t am $d$ $n$g
[FH1] 52: goto ENDE
[000] 53:
[722] 54: :ERKLARUNG
[WK4] 55: echo
[20E] 56: echo |Falls Sie DBTOM1 installieren wollen,
geben Sie bitte das Laufwerk an,
[NPD] 57: echo |auf das die Micro-Datenbank-Verwaltun
g kopiert werden soll. Also z.B.:
[N9D] 58: echo |DBTOM1 C: = kopiert das System von de
r Diskette (Laufwerk A:) nach C:
[7L7] 59: echo |
[DYD] 60: echo |Sie können später dann mit \DBTOM1 [D
ateiname] eine beliebige Daten-
[WAA] 61: echo |bank erstellen und bearbeiten.
[7L7] 62: echo |
```

```
[0NA] 63: echo |Syntax: DBTOM1 [ Laufwerk: | Datei
|
[7L7] 64: echo |
[6YD] 65: echo |Wenn Sie DBTOM1 nicht installieren mö
chten, wechseln Sie ggf. auf das
[6RD] 66: echo |Laufwerk, auf dem DBTOM1 bereits exis
tiert (meist A:, manchmal auch C:),
[RCA] 67: echo |und rufen Sie erneut DBTOM1 auf.
[PL4] 68: echo |
[000] 69:
[9U0] 70: :ENDE
```

Listing 1. »dbtom1.bat« startet die Mini-Datenbank

```
[FD1] 1: @echo off
[UC3] 2: rem Modul von DBTOM1
[UT6] 3: rem DATEI.BAT setzt neue Datenbank aktiv
[5G3] 4: if %1 == . goto meldung
[D04] 5: if %1 == ? goto erklarung
[000] 6:
[V44] 7: if exist %1 goto altdat
[UP1] 8: :NEUDATEI
[B17] 9: echo Neue Micro-Datenbank "%1" wird erzeugt
[SF4] 10: copy /B NEUDATEI.DAT %1 >NUL
[EP1] 11: :ALTDAT
[D66] 12: echo Datenbank "%1" wird aktiv gesetzt.
[ML2] 13: set DBTOM1DAT=%1
[FH1] 14: goto ende
[000] 15:
[1E1] 16: :MELDUNG
[ZY4] 17: if %1\DBTOM1DAT% == . goto inaktiv
[R56] 18: echo Zur Zeit ist die Micro-Datenbank
[9A4] 19: echo "%1\DBTOM1DAT%" aktiviert
[FH1] 20: goto ende
[000] 21:
[ME1] 22: :INAKTIV
[6H7] 23: echo Zur Zeit ist das Micro-Datenbanksystem
[AP4] 24: echo DBTOM1 nicht aktiviert!
[FH1] 25: goto ende
[000] 26:
[722] 27: :ERKLARUNG
[YS3] 28: echo
[UKC] 29: echo DATEI.BAT gibt den Namen der aktiven D
atenbank aus oder setzt eine neue
[H0C] 30: echo Datei aktiv; d.h. daß sich alle folgen
den Befehle auf diese Datenbank
[292] 31: echo beziehen.
[ZCC] 32: echo Syntax: DATEI [[Pfad\]Dateiname]
(Dateiname ohne Erweiterung!)
[YS0] 33: echo.
[LK9] 34: echo DATEI Gibt den Namen der
aktiven Datenbank aus
[PBB] 35: echo DATEI C:\DBTOM1 Setzt neue Datei D
BTOM1 in Laufwerk C: aktiv.
[MWA] 36: echo Falls diese Datei
noch nicht existiert, so wird sie
[625] 37: echo automatisch angele
gt.
[MW0] 38: pause
[F32] 39: type BEFEHLE
[000] 40:
[9U0] 41: :ENDE
```

Listing 2. Mit »datei.bat« wechseln Sie die Datendatei


```

[FD1] 1: @echo off
[UC3] 2: rem Modul von DBTOM1
[UX5] 3: rem ERGÄNZE.BAT ergänzt einzelne
[KR3] 4: rem oder mehrere Daten
[B34] 5: if .%1 == .? goto erklärung
[E45] 6: if not .%1 == . goto direkteingabe
[000] 7:
[9K1] 8: :EINLESEN
[6AL] 9: echo -----
[T56] 10: echo Bitte geben Sie das in %DBTOM1DAT%
[KG7] 11: echo einzufügende Datum ein. Drücken Sie am
[CS6] 12: echo Ende der Eingabezeile die Enter-Taste
[3C7] 13: echo Drücken Sie [F6] und [Enter], wenn Sie
[S16] 14: echo den Eingabemodus beenden wollen.
[BY5] 15: echo Geben Sie jetzt einen beliebigen
[T97] 16: echo Eintrag ein, oder nur [F6] und [Enter]
[G24] 17: echo für keine Eingabe:
[YS0] 18: echo.
[X96] 19: copy /A %DBTOM1DAT%+con DBTOM1.TMP >NUL
[PR4] 20: sort <DBTOM1.TMP >%DBTOM1DAT%
[073] 21: del DBTOM1.TMP >NUL
[YS0] 22: echo.
[4G5] 23: echo Die Eingabe wurde eingefügt
[M33] 24: echo in %DBTOM1DAT%.
[FH1] 25: goto ende
[000] 26:
[2F2] 27: :DIREKTEINGABE
[2Y6] 28: rem In Eingabeschleife jeweils 5 Parameter
[WK2] 29: rem abarbeiten.
[NV4] 30: rem Vorsicht bei den Kommata!
[VL3] 31: rem (%x ignoriert sie)
[LZ1] 32: set DBTOM1=
[2D1] 33: :DIREKT1
[VL4] 34: set DBTOM1=%DBTOM1% %1 %2 %3 %4 %5
[H94] 35: for %%a in (1 2 3 4 5) DO SHIFT
[E24] 36: if not .%1 == . goto DIREKT1
[000] 37:
[XV5] 38: echo In %DBTOM1DAT% wird eingefügt:
[702] 39: echo %DBTOM1%
[WT4] 40: echo %DBTOM1%>%DBTOM1DAT%
[LZ1] 41: set DBTOM1=
[N36] 42: type %DBTOM1DAT% | SORT >%DBTOM1DAT%
[FH1] 43: goto ende
[000] 44:
[722] 45: :ERKLÄRUNG
[PN0] 46: cls
[MBL] 47: echo
[6W7] 48: echo ERGÄNZE.BAT fügt ein Datum in die
[C46] 49: echo Datenbank ein.
[MG5] 50: echo Syntax:
[XW7] 51: echo ERGÄNZE Eingabe menügesteuert
[775] 52: echo oder
[C98] 53: echo ERGÄNZE datum Datum wird eingefügt
[ECL] 54: echo
[F32] 55: type BEFEHLE
[000] 56:
[9U0] 57: :ENDE

```

Listing 3. Zum Eintippen von Datensätzen verwenden Sie »ergänze.bat«

```

[FD1] 1: @echo off
[UC3] 2: rem Modul von DBTOM1
[W37] 3: rem SUCHE.BAT sucht ein oder mehrere Daten
[B34] 4: if .%1 == .? goto erklärung
[WK6] 5: if exist DBSUCH.ALt del DBSUCH.ALt >NUL
[VU4] 6: if .%1 == ./z goto zeilennummern
[VU4] 7: if .%1 == ./Z goto zeilennummern
[7R4] 8: copy %DBTOM1DAT% DBSUCH >NUL
[H93] 9: goto parmauswerten
[000] 10:
[4L2] 11: :ZEILENUMMERN
[GW0] 12: SHIFT
[716] 13: find /v /n "^^" <%DBTOM1DAT% >DBSUCH
[BK2] 14: :PARMAUSWERTEN
[U33] 15: if .%1 == . goto alles
[P94] 16: ren DBSUCH DBSUCH.ALt >NUL
[000] 17:
[VC1] 18: :SUCHEN
[XW3] 19: if %1 == /o goto zufuegen
[XW3] 20: if %1 == /O goto zufuegen
[R77] 21: if exist DBSUCH copy DBSUCH DBSUCH.ALt >NUL
[YB4] 22: if %1 == /n goto wegfiltern
[YB4] 23: if %1 == /N goto wegfiltern
[CC4] 24: FIND "%1" <DBSUCH.ALt >DBSUCH
[V32] 25: goto weiter
[R32] 26: :WEGFILTERN
[GW0] 27: SHIFT
[8Z4] 28: FIND /V "%1" <DBSUCH.ALt >DBSUCH
[V32] 29: goto weiter
[KM1] 30: :ZUFUEGEN
[GW0] 31: SHIFT
[FV4] 32: if %1 == /n goto nichtzufuegen
[FV4] 33: if %1 == /N goto nichtzufuegen
[FH4] 34: FIND "%1" <DBSUCH.ALt >>DBSUCH
[V32] 35: goto weiter
[8L2] 36: :NICHTZUFUEGEN
[GW0] 37: SHIFT
[X35] 38: FIND /V "%1" <DBSUCH.ALt >>DBSUCH
[NC1] 39: :WEITER
[GW0] 40: SHIFT
[724] 41: if not .%1 == . goto SUCHEN
[8H1] 42: :GEFUNDEN
[A83] 43: del DBSUCH.ALt >NUL
[AD3] 44: SORT | MORE <DBSUCH
[FH1] 45: goto ende
[000] 46:
[P11] 47: :ALLES
[L22] 48: more <DBSUCH
[FH1] 49: goto ende
[000] 50:
[722] 51: :ERKLÄRUNG
[PN0] 52: cls
[V02] 53: more <SUCHE
[MW0] 54: pause
[F32] 55: type BEFEHLE
[000] 56:
[9U0] 57: :ENDE

```

Listing 5. Für die Suche nach Datensätzen gibt es »suche.bat«

```

[FD1] 1: @echo off
[UC3] 2: rem Modul von DBTOM1
[8Y4] 3: rem KORRIGIERE.BAT ruft EDLIN
[5H5] 4: rem zur Korrektur der Daten auf
[LR4] 5: if not .%1 == . goto erklärung
[000] 6:
[GH6] 7: echo Bitte beachten Sie die Befehle des
[YM6] 8: echo MS-DOS-Editors EDLIN entsprechend
[946] 9: echo Ihrem Handbuch. Verlassen Sie den
[A15] 10: echo Editor mit [E] [Enter]
[WL2] 11: edlin %DBTOM1DAT%
[FH1] 12: goto ende
[000] 13:
[722] 14: :ERKLÄRUNG
[PN0] 15: cls
[PV2] 16: more <KORRIGIERE
[MW0] 17: pause
[F32] 18: type BEFEHLE
[000] 19:
[9U0] 20: :ENDE

```

Listing 4. Datensätze werden mit »korrigie.bat« geändert

```

[FD1] 1: @echo off
[UC3] 2: rem Modul von DBTOM1
[375] 3: rem ZÄHLE.BAT zählt die Daten
[B34] 4: if .%1 == .? goto erklärung
[BV5] 5: echo Anzahl der Zeilen in der Datei:
[9P4] 6: find /c /v "^^" %DBTOM1DAT%
[M87] 7: echo Anzahl der zuletzt gefundenen Zeilen:
[804] 8: find /c /v "^^" DBSUCH
[FH1] 9: goto ende
[000] 10:
[722] 11: :ERKLÄRUNG
[PN0] 12: cls
[6AL] 13: echo -----
[C66] 14: echo ZÄHLE.BAT zählt die Daten in der
[H95] 15: echo Datei. Geben Sie einfach ein:
[YS0] 16: echo.
[572] 17: echo ZÄHLE
[YS0] 18: echo.
[F32] 19: type BEFEHLE
[000] 20:
[9U0] 21: :ENDE

```

Listing 6. Die Zahl der Datensätze bestimmen Sie mit »zähle.bat«


```

[FD1] 1: @echo off
[UC3] 2: rem Modul von DBTOM1
[324] 3: rem LÖSCHE.BAT löscht Daten
[4K4] 4: if %1 == . goto allesloeschen
[164] 5: if %1 == ? goto erklaerung
[PH4] 6: if %1 == /t goto textloeschen
[PH4] 7: if %1 == /T goto textloeschen
[3E4] 8: if %1 == /r goto rueckgaengig
[3E4] 9: if %1 == /R goto rueckgaengig
[MG3] 10: if %1 == /! goto zaehlen
[9H3] 11: if %1 == /!! goto drucken
[000] 12:
[9K1] 13: :LOESCHEN
[V43] 14: echo %1d >DBTOM1.TMP
[F63] 15: echo e >>DBTOM1.TMP
[PB5] 16: edlin %DBTOM1DAT% <DBTOM1.TMP >NUL
[073] 17: del DBTOM1.TMP >NUL
[GW0] 18: SHIFT
[N74] 19: IF not %1 == . goto loeschen
[FH1] 20: goto ende
[000] 21:
[0G2] 22: :ALLESLOESCHEN
[RM6] 23: echo Wenn Sie nicht alle Daten verlieren
[KW6] 24: echo wollen, drücken Sie jetzt [Ctrl-C]!
[M64] 25: echo (Danach [J] oder [Y])
[MW0] 26: pause
[RT7] 27: if exist %DBTOM1DAT%.BAK del %DBTOM1DAT%.BAK
      K >NUL
[EX4] 28: ren %DBTOM1DAT% %DBTOM1DAT%.BAK
[T16] 29: copy /B NEUDETEI.DAT %DBTOM1DAT% >NUL
[FH1] 30: goto ende
[000] 31:
[TA2] 32: :TEXTLOESCHEN
[2X6] 33: echo Es werden folgende Daten gelöscht:
[CL5] 34: find "%2" %DBTOM1DAT% | SORT | MORE
[5K3] 35: echo Wenn Sie diese Daten nicht verlieren
[HV6] 36: echo wollen, drücken Sie jetzt [Ctrl-C]
[M64] 37: echo (Danach [J] oder [Y])
[MW0] 38: pause
[RT7] 39: if exist %DBTOM1DAT%.BAK del %DBTOM1DAT%.BAK
      K >NUL
[MP5] 40: ren %DBTOM1DAT% %DBTOM1DAT%.BAK >NUL
[XG7] 41: find /V "%2" <%DBTOM1DAT%.BAK | SORT >%DBTOM1DAT%
      MIDAT%
[FH1] 42: goto ende
[000] 43:
[E92] 44: :RUECKGAENGIG
[N15] 45: ren %DBTOM1DAT% DBTOM1.TMP >NUL
[5E8] 46: echo Die Daten aus "%DBTOM1DAT%" werden res
      tauriert.
[MP5] 47: ren %DBTOM1DAT%.BAK %DBTOM1DAT% >NUL
[8M5] 48: ren DBTOM1.TMP %DBTOM1DAT%.BAK >NUL
[N36] 49: type %DBTOM1DAT% | SORT >%DBTOM1DAT%
[FH1] 50: goto ende
[000] 51:
[1D1] 52: :ZAEHLEN
[N36] 53: type %DBTOM1DAT% | sort >%DBTOM1DAT%
[NR4] 54: find /N /V "^^" %DBTOM1DAT%
[FH1] 55: goto ende
[000] 56:
[6D1] 57: :DRUCKEN
[N36] 58: type %DBTOM1DAT% | sort >%DBTOM1DAT%
[246] 59: echo Die Daten aus "%DBTOM1DAT%" werden
[VU6] 60: echo numeriert auf dem Drucker ausgegeben.
[XG5] 61: find /N /V "^^" %DBTOM1DAT% >PRN
[FH1] 62: goto ende
[000] 63:
[722] 64: :ERKLAERUNG
[PN0] 65: cls
[552] 66: more <LÖSCHE
[MW0] 67: pause
[F32] 68: type BEFEHLE
[000] 69:
[9U0] 70: :ENDE

```

Listing 7. »lösche.bat« dient dem Entfernen von Datensätzen

```

[FD1] 1: @echo off
[UC3] 2: rem Modul von DBTOM1
[4G6] 3: rem DRUCKE.BAT sucht Daten wie SUCHE.BAT.
[SY6] 4: rem Die Ausgabe erfolgt auf dem Drucker
[B34] 5: if %1 == .? goto erklaerung
[000] 6:
[VC1] 7: :SUCHEN
[SR5] 8: echo ***** SUCHE %1 %2 %3 %4 %5 %6 %7 %8 %9
      >PRN
[3V6] 9: echo Es wird gesucht nach %1 %2 %3 %4 %5 %6
      %7 %8 %9
[WF5] 10: %COMSPEC% /C SUCHE %1 %2 %3 %4 %5 %6 %7 %8
      %9
[A85] 11: echo Gefundene Einträge werden
[3B5] 12: echo auf dem Drucker ausgegeben.

```

```

[PW2] 13: SORT <DBSUCH >PRN
[FH1] 14: goto ende
[000] 15:
[722] 16: :ERKLAERUNG
[PN0] 17: cls
[RR6] 18: echo Die Parameter für diesen Befehl sind
[C46] 19: echo die gleichen, wie die für den nach-
[LR6] 20: echo folgend aufgeführten Befehl SUCHE.
[YP6] 21: echo Allerdings können nur (inkl. /O und
[VU5] 22: echo /N) 9 Wörter angegeben werden,
[FZ5] 23: echo der Rest wird ignoriert. Sollen
[X76] 24: echo kompliziertere Suchen gedruckt
[U27] 25: echo werden, benutzen Sie bitte den Befehl
[EF6] 26: echo SUCHE und drucken Sie die gefundenen
[S87] 27: echo Einträge dann mit COPY DBSUCH PRN aus.
[MW0] 28: pause
[V02] 29: more <SUCHE
[MW0] 30: pause
[F32] 31: type BEFEHLE
[000] 32:
[9U0] 33: :ENDE

```

Listing 8. Mit »drucke.bat« geben Sie Datensätze auf einem Drucker aus

```

[FD1] 1: @echo off
[UC3] 2: rem Modul von DBTOM1
[U75] 3: rem HILFE.BAT gibt eine einfache
[5K3] 4: rem Hilfsmeldung aus
[PN0] 5: cls
[C55] 6: echo D a t e n v e r w a l t u n g
[463] 7: echo DBTOM1 v1.1
[XF5] 8: echo geschrieben von Tom Stein
[YS0] 9: echo.
[ZY4] 10: if %DBTOM1DAT% == . goto inaktiv
[000] 11:
[MX6] 12: echo Zur Verwaltung der Micro-Datenbank
[XL5] 13: echo "%DBTOM1DAT%" mit MS-DOS-Mitteln
[V32] 14: goto weiter
[000] 15:
[ME1] 16: :INAKTIV
[3Y6] 17: echo Zur Verwaltung einer noch mit "DATEI"
[S87] 18: echo zu benennenden Datenbank unter MS-DOS
[000] 19:
[NC1] 20: :WEITER
[4G6] 21: echo Es stehen die folgenden Befehle als
[6A6] 22: echo Als Stapeldateien zur Verfügung:
[F32] 23: type BEFEHLE

```

Listing 9. Eine Übersicht für alle Befehle bekommen Sie durch die Stapeldatei »hilfe.bat«

```

[FD1] 1: @echo off
[MC3] 2: rem Modul von BDTOM1
[G75] 3: rem ENDE.BAT beendet die Arbeit
[LR4] 4: if not %1 == . goto erklaerung
[VA6] 5: echo Aktive Datenbank war: "%DBTOM1DAT%"
[EHO] 6: cd \
[FL3] 7: PROMPT %DBTOM1PROMPT%
[413] 8: set DBTOM1PROMPT=
[FD2] 9: set DBTOM1DAT=
[7KL] 10: echo
[387] 11: echo | Die Micro-Datenbanksystem DBTOM1
[PT6] 12: echo | ist deaktiviert worden.
[LGL] 13: echo
[FH1] 14: goto ende
[000] 15:
[722] 16: :ERKLAERUNG
[6AL] 17: echo
[0A6] 18: echo ENDE.BAT beendet die Arbeit mit dem
[XR4] 19: echo Micro-Datenbanksystem.
[B03] 20: echo Syntax: ENDE
[F32] 21: type BEFEHLE
[000] 22:
[9U0] 23: :ENDE

```

Listing 10. »ende.bat« beendet die Arbeit mit der Mini-Datenbank

Stapeldatei mit Pfiff

Wie oft kommt es vor, daß man einen langen Befehl oder gar eine Sequenz von einigen Befehlen immer wieder eintippen muß. Natürlich könnte man sich in einem solchen Fall erst eine kleine Stapeldatei anlegen und sie dann immer bei Bedarf aufrufen. Einfacher geht es jedoch mit »Quickbat«.

»Quickbat« ist die Bezeichnung einer raffinierten Stapeldatei, die aus Gründen der einfacheren Bedienung allerdings nur mit dem Minuszeichen aufgerufen wird. Die Stapeldatei (Listing 1) müssen Sie also unter dem Namen »-bat« speichern. Haben Sie nun einen MS-DOS-Befehl, den Sie später vielleicht noch einmal brauchen, dann setzen Sie mit »Quickbat« einfach nur das Minuszeichen vor diesen Befehl, und schon wird er normal ausgeführt. Gleichzeitig aber ist der Befehl gespeichert und kann immer wieder ausgeführt werden, indem Sie jetzt einfach nur das Minuszeichen und anschließend die Return-Taste tippen. Dabei dürfen Sie zwischendurch beliebige andere MS-DOS-Befehle verwenden. Natürlich müssen Sie mindestens ein Leerzeichen zwischen den Befehlen und dem Minuszeichen lassen, damit das Ganze auch funktioniert.

Quickbat speichert ganze Befehlsfolgen

Soweit ist Quickbat schon ganz nützlich. Das Programm leistet aber noch viel mehr. Quickbat kann genauso einfach eine ganze Folge bis zu acht Befehlen auf einmal

speichern und immer wieder ausführen, ohne daß Sie eine neue Stapeldatei schreiben müssen. Um diese Makro-Fähigkeit zu nutzen, müssen Sie alle Befehle in eine einzige Zeile schreiben und dazwischen jeweils wieder das Minuszeichen verwenden. Die Eingabe sieht dann so aus:

- Erster Befehl - Zweiter Befehl - - Letzter Befehl

Mit [Enter] wird abgeschlossen. Die Befehle werden bei dieser Art der Eingabe sofort zum ersten Mal abgearbeitet. Später können Sie dann durch Eingabe des Minuszeichens allein die ganze Befehlsfolge immer wieder aufrufen. Alle Befehle der Makro-Folge bleiben so lange gespeichert, bis sie durch eine neue Eingabe ersetzt werden oder bis der Computer abgeschaltet wird.

Soweit keines der Zeichen »<«, »>«, »>>«, »>>>«, »>>>>«, »>>>>>« oder »%« vorkommt, läßt sich jede Art von MS-DOS-Befehlen auf diese Weise speichern und ausführen. Erlaubt sind dabei sogar Labels und Goto-Anweisungen wie in einer normalen Stapeldatei.

Das ist schon alles Wissenswerte, um Quickbat mit großem Nutzen einzusetzen. Es gibt bei Quickbat eine einzige Option, aufzurufen mit »-?«; wozu diese gut ist, können Sie sicher erraten, ausprobieren oder aus dem Listing ersehen.

Die Funktionsweise von Quickbat ist einfach. Die Befehlsfolge wird als Eingabeparameter eingelesen, an den einzelnen Minuszeichen aufgeteilt und jeder Teil in einer Environment-Variablen gespeichert. Beim ersten wie bei allen späteren Aufrufen werden die Variableninhalte der Reihe nach MS-DOS zur Abarbeitung vorgelegt.

Ungewöhnlich ist dabei die Verwendung einer Art von Indizierung mit normalen Environment-Variablen. Auch wird von keinem externen Befehl Gebrauch gemacht, so daß es genügt, wenn die Datei »-bat« in einem der mittels Path-Anweisung angegebenen Verzeichnissen vorliegt, um Quickbat jederzeit zu Diensten zu haben.

(Wolf-Rainer Bartels/hi)

Name	- .bat
Sprache	MS-DOS
Grafikkarte	jede
Besonderheiten	keine

```
[YA2] 1: :: ** Quickbat **
[AS7] 2: :: (c) 1988,1989 Redaktionsbüro Everts & Ha
gedorn
[646] 3: :: Redaktionsbüro DOS International
1
[ZD7] 4: :: -bat Zweck siehe :help WR Bartels
31.1.88
[DA1] 5: echo off
[7W2] 6: if %1==? goto exec
[5R2] 7: if %1==? goto help
[WMA] 8: for %%$ in (%I I II III IIII IIIII IIIIII I
IIIIII IIIIIII) do set %%$=
[000] 9:
[KV0] 10: :next
[U82] 11: if %1==? shift
[C23] 12: if %1==? goto next
[7W2] 13: if %1==? goto exec
[1T1] 14: set %I=%I%I
[CS6] 15: if not %I%==IIIIIIII set %I%I= goto
ENDE
[PD1] 16: set %Z=%1
[973] 17: if %2==? goto store
[E13] 18: if %2==? goto store
[000] 19:
[LV0] 20: :more
[GW0] 21: shift
[V12] 22: set %Z=%Z% %1
[973] 23: if %2==? goto store
[E13] 24: if %2==? goto store
[GW0] 25: shift
[V12] 26: set %Z=%Z% %1
[E13] 27: if %2==? goto store
[BN3] 28: if not %2==? goto more
```

```
[000] 29:
[Z31] 30: :store
[IS1] 31: set %I%=%Z%
[GW0] 32: shift
[EV6] 33: if not %2==? if not %I%==IIIIIIII goto ne
xt
[000] 34:
[EV0] 35: :exec
[RV3] 36: for %%$ in (%I %Z) do set %%$=
[P03] 37: if %I%==? goto help
[791] 38: echo on
[TD0] 39: %I%
[EL0] 40: %II%
[WP0] 41: %III%
[LW0] 42: %IIII%
[L41] 43: %IIIII%
[8A1] 44: %IIIIII%
[PD1] 45: %IIIIIII%
[DL1] 46: %IIIIIIII%
[S27] 47: rem Höchstanzahl von 8 Befehlen erreicht!
[FH1] 48: goto ENDE
[000] 49:
[CU0] 50: :help
[PN0] 51: cls
[6P7] 52: if %I%==? echo FEHLER! Keine Befehle gespe
ichert!
[YS0] 53: echo.
[08D] 54: echo QUICKBAT: Eingeben von (max.8) Befehle
n/Programmen in einer Kommandozeile.
[YS0] 55: echo.
[KM6] 56: echo Eine Utility der DOS International
[YS0] 57: echo.
[UFC] 58: echo Syntax 1 " - Befehl - Befehl - ... [E
nter] " für 1. Ausführung der Befehle.
[2NC] 59: echo Zwischen den Befehlen ein fr
eistehendes " - " statt [Enter] setzen.
[TS8] 60: echo " < > | = ; , % " dürfen ni
cht verwendet werden.
[82A] 61: echo Beispiel " - edlin test.bat - del tes
t.bak - test [Enter] "
[YS0] 62: echo.
```

Das Listing »Quickbat« ist eine wertvolle Hilfe für jeden MS-DOS-Anwender


```
[KSB] 63: echo Syntax 2 " - [Enter] " für wiederholt
        e Ausführung der Befehle.
[YSØ] 64: echo.
[RF7] 65: echo Syntax 3 " - ? [Enter] " für diese Ta
        fel.
[YSØ] 66: echo.
[SCB] 67: echo "SET [Enter]" zeigt die einzelnen Bef
        ehle hinter I= , II= , III= ...
```

```
[ØLA] 68: echo "SET I...=Neuer Befehl [Enter]" erset
        zt den I...-ten Befehl.
[ØØØ] 69:
[9UØ] 70: :ENDE
```

Das Listing »Quickbat« ist eine wertvolle Hilfe für jeden MS-DOS-Anwender

Erläuterungen zum Listing

1-4: Zeilen, die mit zwei Doppelpunkten beginnen, werden weder als Befehl noch als Label interpretiert und selbst mit »echo on« nicht gezeigt.

5: »echo off« steht nicht ganz links; so stört es weniger beim Ablauf.

6: Falls kein Aufrufparameter mitgegeben wurde, Sprung zur Abarbeitung der schon gespeicherten Befehlsfolge.

7: Falls ein Fragezeichen als Parameter auftaucht, zeige Hilfetafel.

8: Wird diese Zeile erreicht, enthalten die Parameter »%1«, »%2«, »%3« etc. die neu zu speichernden Befehle, getrennt durch Minuszeichen. Sämtliche benötigten Variablen werden mit einer For-Schleife geleert. For-Schleifen arbeiten sehr viel schneller als Einzelbefehle.

10: Dieses Label wird immer angesprungen, wenn bei der wortweisen Abarbeitung der Eingabezeile ein Minuszeichen auftaucht, wenn also ein neuer Befehl anfängt.

11, 12: Ein oder mehrere Minuszeichen werden überlesen.

13: Ist jetzt die Eingabezeile leer, werden die Befehle aufgerufen, sonst wird gespeichert.

14: Die Indexvariable »I« wird mit einem String belegt, der aus ihrem bisherigen Inhalt (anfangs leer) und einem daran angehängten »I« besteht. Mit jedem Durchlauf dieser Zeile verändert sich also der Inhalt von »I« nacheinander in »I«, »II«, »III« etc. Mit »set %I% = ...« wird dann (weiter unten) bei jedem Durchlauf eine andere Variable belegt, als erste Variable »I«, dann Variable »II« und so weiter bis zur achten und letzten Variablen »IIIIII«.

15: Vorausgesetzt, die Indexvariable ist noch nicht die letzte, wird in die für den nächsten Durchlauf vorgesehene Variable erst einmal vorsorglich ein »goto ende« gespeichert; dies wird gegebenenfalls später wieder überschrieben.

16: Das erste Wort des Befehls wird in der Zeilenvariablen »\$Z« zwischengespeichert.

17, 18: Ist das nächste Wort ein Minuszeichen oder folgen keine Angaben mehr, so wird zu »store« gesprungen (der ganze Befehl bestand nur aus einem Wort), sonst weiter.

20: Dieses Label wird später besprochen.

21: »shift«, um alle restlichen Befehlswörter und Minuszeichen um einen Platz vorzurücken.

22: Die Variable »\$Z« wird neu belegt. An ihren eigenen Inhalt (das erste Befehlswort) wird ein Leerzeichen und das zweite Befehlswort angehängt.

23, 24: Prüfen, ob ein Minuszeichen oder nichts folgt; ist dies der Fall, so ist der Befehl vollständig in der Variablen »\$Z« zwischengespeichert.

25-28: Das nächste Wort wird durch »shift« herangebracht, an »\$Z« angehängt und daraufhin überprüft, ob es das letzte Wort ist. Ist immer noch nicht das letzte Wort erreicht, wird zu »more« (Zeile 20) zurückgesprungen und alles wiederholt. Der Sprung zu »more« hätte schon vier Zeilen früher vorgenommen werden können, der Ablauf wäre jedoch langsamer, denn bei jedem »goto« liest MS-DOS die Stapeldatei von Anfang an ein, um nach dem Label zu suchen, was bedeutend länger dauert als das Abarbeiten aufeinanderfolgender Zeilen.

30: Wenn nun zu »store« gesprungen wird, enthält die Variable »\$Z« einen kompletten Einzelbefehl, der selbst aus mehreren Wörtern bestehen kann.

31: Der in »\$Z« gespeicherte Einzelbefehl wird jetzt in die Variable übertragen, deren Bezeichnung (schon in Zeile 14) mit der Index-Variablen »I« definiert wurde (»I«, »II«, »III«, ...).

32: Minuszeichen mittels Shift-Befehl überlesen.

33: Ist »%2« nun nicht leer, liegt noch ein ganzer Befehl vor, also alles nochmals ab Zeile 10 abarbeiten, vorausgesetzt, es ist noch eine Variable frei.

35: Die Befehlsfolge ist in Einzelbefehle aufgegliedert und gespeichert; die Ausführung kann beginnen. Gleiches gilt, wenn in Zeile 6 festgestellt wurde, daß es sich um einen Wiederaufruf handelt und direkt hierher gesprungen wurde.

36: Schnell noch zwei nicht mehr benötigte Variablen aus dem Environment entfernen. Sie sollten keinen Environment-Platz verschwenden, denn MS-DOS reserviert normalerweise nur 160 Byte für den Umgebungsbereich.

37: Vielleicht wurde Quickbat zwecks Befehlswiederholung ohne Parameter aufgerufen, obwohl noch nichts gespeichert wurde. Dann hilft nur noch die Hilfetafel.

38: Damit bei der Ausführung der einzelnen gespeicherten Befehle auf dem Bildschirm verfolgt werden kann, welche Befehle gerade abgearbeitet werden, wird auf »echo on« umgeschaltet.

39-46: Die Inhalte der Befehlsvariablen kommen der Reihe nach zur Ausführung. In der letzten belegten Variablen steht »goto ende« (siehe Zeile 15).

47: Bestand die Befehlsfolge aus acht oder noch mehr Befehlen, so steht in der achten und letzten Variablen »IIIIII« kein »goto ende«, und eine Bemerkung warnt auf dem Bildschirm, daß eventuell weitere Befehle abgeschnitten wurden.

48: Damit die folgende Hilfetafel auch übersprungen wird, wenn in der achten Variablen noch ein Befehl steht, kommt hier ein »goto ende« zum guten Ende.

50-68: Hilfetafel

70: Das Ende

»delq.com« macht das Datenlöschen sicher

Dateien mit dem Befehl »del *.*« zu löschen, kann sich als gefährliches Unterfangen entpuppen. Besonders seit MS-DOS-Version 3.2, denn hier geht es noch »einfacher« – und »schwupps« sind die Dateien weg. Das kleine Basic-Programm »delq.com« baut hier eine Sicherheitskontrolle ein.

Zweifelsohne ist es sehr praktisch, mit dem Befehl »copy.« alle Dateien eines Verzeichnisses zu kopieren. Aber es geht auch umgedreht: So genügt auch der kurze MS-DOS-Befehl »del.« vollkommen, um alle Dateien eines Verzeichnisses zu löschen – und das geht oft schneller, als Sie denken.

Sie sehen auf dem Bildschirm die Meldung

Alle Dateien im Verzeichnis werden gelöscht
Sind Sie sicher (J/N)?

und reflexartig ist auch schon die Taste [J] gedrückt.

Damit es gar nicht erst soweit kommt, benutzen Sie in Zukunft einfach das kleine Programm »delq.com«. Es arbeitet genau wie der MS-DOS-Befehl »del.com«. Der wesentliche Unterschied liegt in der Sicherheitsabfrage: Auch wenn Sie

delq *.*

eingeben, versichert sich das Programm vor dem Löschen einer Datei durch eine Nachfrage, ob Sie die betreffende Datei auch wirklich löschen wollen. Sie geben dann also [J] für »ja« oder [N] für »nein« ein. Jede andere Taste als [J] wird ebenfalls als »nein« interpretiert.

Wie kommen Sie nun in den Genuß dieses kleinen Utilities? Sind Sie stolzer Besitzer eines Assemblers (zum Beispiel von MASM oder einem ähnlichen), dann können Sie den Assembler-Quelltext (Listing 1) eingeben. Etwas kürzer ist das Listing 2. Wenn Sie keinen Assembler besitzen, rufen Sie GW-Basic beziehungsweise Basica auf und geben dieses Listing in Ihren Computer ein. Es ist für GW-Basic (Basica) geschrieben, funktioniert aber auch mit Quick- oder Turbo-Basic. Nach dem Aufruf des Basic-Programms wird die Datei »delq.com« erzeugt. Wenn Sie alle Datenzeilen richtig eingegeben haben, erhalten Sie die Meldung, daß die Datei »delq.com« erzeugt wurde. Andernfalls zeigt Ihnen das Programm an, in welcher Zeile Sie einen Fehler gemacht haben.

(hw)

Name	delq_asm.asm
Sprache	Assembler
Grafikkarte	jede
Besonderheiten	keine

```

[9Y5] 1: ; DELQ.ASM - Datei löschen mit Abfrage
[XX4] 2: ; (c) 1989, Little Softwarehouse
[DB7] 3: ; und Redaktionsbuero Everts & Ha
      gedorn
[000] 4:
[403] 5: KommStrLen equ 80H
[KU3] 6: ; Offset des Namen in DTA
[2P2] 7: NameInDta equ 9EH
[000] 8:
[EV1] 9: BELL EQU 7
[PT1] 10: TAB EQU 9
[DT1] 11: LF EQU 10
[8U1] 12: CR EQU 13
[042] 13: SPACE EQU ' '
[000] 14:
[SD2] 15: code segment
[3U4] 16: ASSUME cs:code,ds:code,es:code
[A61] 17: ORG 100H
[000] 18:
[Y81] 19: Anfang:
[321] 20: jmp Start
[000] 21:
[HK4] 22: AntwStr DB 'Soll die Datei $'
[485] 23: SelStr DB ' gelöscht werden? $'
[AG7] 24: Syntax DB 'Aufruf: delq <Dateiname>','CR
      ,LF,$'
[HK9] 25: KeinStr DB 'Keine Dateien mit diesem Namen
      vorhanden!','BELL
[WR2] 26: CrLf DB CR,LF,$'
[X97] 27: DelErr DB 'Fehler beim Löschen','CR,LF,B
      ELL,$'
[000] 28:
[054] 29: SkipWhiteSpace PROC NEAR
[0M1] 30: SkipLoop:
[7F6] 31: ; Zeichen in AL laden, SI inkrementiere
      n
[CV0] 32: lodsb
[WB5] 33: ; Anzahl Zeichen um 1 erniedrigen
[741] 34: dec cx
[PL2] 35: ; Leerzeichen ...
[472] 36: cmp al,SPACE
[962] 37: je SkipLoop
[6T4] 38: ; ... oder [Tab] überspringen
[RU1] 39: cmp al,TAB
[962] 40: je SkipLoop
[095] 41: ; Zurück zum gelesenen Zeichen
[SS1] 42: dec si
[FP3] 43: ; Anzahl aktualisieren
[CS1] 44: inc cx
[3P0] 45: ret
[ZD3] 46: SkipWhiteSpace ENDP
[000] 47:
[ZL4] 48: SkipToWhiteSpace PROC NEAR

```

```

[542] 49: SkipToLoop:
[7F6] 50: ; Zeichen in AL laden, SI inkrementiere
      n
[CV0] 51: lodsb
[WB5] 52: ; Anzahl Zeichen um 1 erniedrigen
[741] 53: dec cx
[PL2] 54: ; Leerzeichen ...
[472] 55: cmp al,SPACE
[5L2] 56: je SkipToExit
[LP2] 57: ; ... oder [Tab] ...
[RU1] 58: cmp al,TAB
[5L2] 59: je SkipToExit
[HD6] 60: ; ... oder [Enter] beendet diese Routin
      e
[6T1] 61: cmp al,CR
[5L2] 62: je SkipToExit
[4U2] 63: jmp SkipToLoop
[K32] 64: SkipToExit:
[SS1] 65: dec si
[CS1] 66: inc cx
[3P0] 67: ret
[8S3] 68: SkipToWhiteSpace ENDP
[000] 69:
[V53] 70: AscizOut PROC NEAR
[L91] 71: push bx
[T95] 72: ; BX wird für die Suche gebraucht
[3W1] 73: mov bx,dx
[T42] 74: AscizSearch:
[B25] 75: ; Null am Ende des Strings suchen
[U83] 76: cmp BYTE PTR [bx],0
[G32] 77: je AscizEnd
[851] 78: inc bx
[PU2] 79: jmp AscizSearch
[3K1] 80: AscizEnd:
[U94] 81: ; Dos braucht '$' für Ausgabe
[2K3] 82: mov BYTE PTR [bx],'$'
[UH3] 83: ; Dos "String ausgeben"
[EF1] 84: mov ah,9
[N81] 85: int 21H
[3W3] 86: ; Null wieder eintragen
[CA3] 87: mov BYTE PTR [bx],0
[D71] 88: pop bx
[3P0] 89: ret
[4A2] 90: AscizOut ENDP
[000] 91:
[E21] 92: Start:
[596] 93: ; Länge des Strings in Kommandozeile
[AE3] 94: mov si,KommStrLen
[F52] 95: mov cl,[si]
[U95] 96: ; CX ist die Länge der Parameter
[VE1] 97: mov ch,0
[WD1] 98: cmp cx,1
[9N4] 99: ; Sprung wenn kein Parameter
[FM2] 100: jbe SyntaxErr
[5P5] 101: ; SI zeigt auf Anfang der Parameter
[X61] 102: inc si
[YF6] 103: ; [Tab] und Leerzeichen überspringen
[BE3] 104: call SkipWhiteSpace
[KX3] 105: ; Anfang des Namen merken
[WV1] 106: mov dx,si
[MV3] 107: call SkipToWhiteSpace

```

Listing 1. »delq.asm«, dieses Listing schützt Sie vor bösen Überraschungen


```

[MX4] 108: ; [Enter] durch [0] ersetzen
[LA3] 109: mov BYTE PTR [si],0
[GX3] 110: ; DOS "Erste Datei suchen"
[VT1] 111: mov ah,4EH
[BF1] 112: mov cx,0
[N81] 113: int 21H
[X74] 114: ; Sprung falls keine Datei
[0S1] 115: jc NoFile
[PL1] 116: FindNext:
[UH3] 117: ; Dos "String ausgeben"
[EF1] 118: mov ah,9
[B04] 119: ; Erste Hälfte der Frage
[C44] 120: mov dx,OFFSET AntwStr
[N81] 121: int 21H
[Y55] 122: ; Offset des Namen fuer Ausgabe
[G22] 123: mov dx,NameInDta
[UB2] 124: call AscizOut
[UH3] 125: ; Dos "String ausgeben"
[EF1] 126: mov ah,9
[XF4] 127: ; Zweite Hälfte der Frage
[5T3] 128: mov dx,OFFSET SelStr
[N81] 129: int 21H
[AC5] 130: ; DOS "Tastatur-Eingabe mit Echo"
[ZP1] 131: mov ah,01H
[N81] 132: int 21H
[KP2] 133: ; Zeichen retten
[F91] 134: push ax
[UH3] 135: ; Dos "String ausgeben"
[EF1] 136: mov ah,9
[0W1] 137: ; Neue Zeile
[1C3] 138: mov dx,OFFSET CrLf
[N81] 139: int 21H
[LT3] 140: ; Zeichen resaturieren
[971] 141: pop ax
[B86] 142: ; groß geschrieben für den Vergleich
[3R1] 143: and al,5FH
[TP1] 144: cmp al,'J'
[W82] 145: jne NoDelete
[1E3] 146: ; DOS "Datei Löschen"
[BRI] 147: mov ah,41H
[G22] 148: mov dx,NameInDta
[N81] 149: int 21H
[3D3] 150: ; Fehler beim Löschen
[742] 151: jc DelError
[EK1] 152: NoDelete:
[8G4] 153: ; DOS "Nächste Datei suchen"
[ZT1] 154: mov ah,4FH
[N81] 155: int 21H
[5N5] 156: ; Sprung, wenn es noch Dateien gibt
[W92] 157: jnc FindNext
[YV0] 158: Exit:
[BM3] 159: ; DOS "Prozeß beenden"
[332] 160: mov ax,4C00H
[N81] 161: int 21H
[ZV1] 162: SyntaxErr:
[SF4] 163: ; Aufrufsyntax beschreiben
[PU3] 164: mov dx,OFFSET Syntax
[UX1] 165: jmp Error
[T91] 166: NoFile:
[7V3] 167: ; Meldung "Keine Dateien"
[G24] 168: mov dx,OFFSET KeinStr
[UX1] 169: jmp Error
[8L1] 170: DelError:
[000] 171: ; Meldung "Loeschfehler"
[2X3] 172: mov dx,OFFSET DelErr
[NR3] 173: mov dx,OFFSET DelErr
[931] 174: Error:
[UH3] 175: ; Dos "String ausgeben"
[EF1] 176: mov ah,9
[N81] 177: int 21H
[BM3] 178: ; DOS "Prozeß beenden"
[532] 179: mov ax,4C01H
[GG2] 180: ; Fehler-Code = 1
[N81] 181: int 21H
[BW1] 182: code ENDS
[UP1] 183: END Anfang

```

Listing 1. »delq.asm«, dieses Listing schützt Sie vor bösen Überraschungen

Name	delq_bas.bas
Sprache	GW-Basic
Grafikkarte	jede
Besonderheiten	keine

```

[HE5] 1: 100 REM Programm erzeugt DELQ.COM
[G06] 2: 101 REM (c) 1989, Little Softwarehouse
[P88] 3: 102 REM und Redaktionsbuero Evert
      s.& Hagedorn
[UH4] 4: 110 OPEN "DELQ.COM" AS #1 LEN = 1
[YG2] 5: 120 FIELD #1,1 AS A$

```

```

[6T2] 6: 130 PRUEFSUMME = 0
[6H2] 7: 140 FOR I% = 1 TO 38
[K53] 8: 150 ZEILENSUMME% = 0
[CR2] 9: 160 FOR J% = 1 TO 8
[RW1] 10: 170 READ K%
[M95] 11: 180 PRUEFSUMME = PRUEFSUMME+K%
[XR5] 12: 190 ZEILENSUMME% = ZEILENSUMME%+K%
[E53] 13: 200 LSET A$ = CHR$(K%)
[ZR1] 14: 210 PUT #1
[BZ1] 15: 220 NEXT J%
[5H3] 16: 230 READ ZEILENPRUEF%
[7Y6] 17: 240 IF ZEILENSUMME%=ZEILENPRUEF% GOTO 270
[P46] 18: 250 BEEP: PRINT "Fehler: Zeile";300+10*I%
[MD1] 19: 260 STOP
[5P1] 20: 270 NEXT I%
[FBI] 21: 280 CLOSE
[TV4] 22: 290 IF PRUEFSUMME=29866 GOTO 310
[T86] 23: 300 BEEP: PRINT "Fehler in Datei!": STOP
[V46] 24: 310 PRINT "Datei DELQ.COM erzeugt": END
[AA5] 25: 320 DATA 233,185, 0, 83,111,108,108, 32, 8
      60
[KB5] 26: 330 DATA 100,105,101, 32, 68, 97,116,101, 7
      20
[CB5] 27: 340 DATA 105, 32, 36, 32,103,101,108,148, 6
      65
[BD5] 28: 350 DATA 115, 99,104,116, 32,119,101,114, 8
      00
[AA5] 29: 360 DATA 100,101,110, 63, 32, 36, 65,117, 6
      24
[NC5] 30: 370 DATA 102,114,117,102, 58, 32,100,101, 7
      26
[WB5] 31: 380 DATA 108,113, 32, 60, 68, 97,116,101, 6
      95
[KA5] 32: 390 DATA 105,110, 97,109,101, 62, 13, 10, 6
      07
[6A5] 33: 400 DATA 36, 75,101,105,110,101, 32, 68, 6
      28
[NC5] 34: 410 DATA 97,116,101,105,101,110, 32,109, 7
      71
[9D5] 35: 420 DATA 105,116, 32,100,105,101,115,101, 7
      75
[9B5] 36: 430 DATA 109, 32, 78, 97,109,101,110, 32, 6
      68
[HE5] 37: 440 DATA 118,111,114,104, 97,110,100,101, 8
      55
[145] 38: 450 DATA 110, 33, 7, 13, 10, 36, 70,101, 3
      80
[5D5] 39: 460 DATA 104,108,101,114, 32, 98,101,105, 7
      63
[GD5] 40: 470 DATA 109, 32, 76,148,115, 99,104,101, 7
      84
[T55] 41: 480 DATA 110, 13, 10, 7, 36,172, 73, 60, 4
      81
[MA5] 42: 490 DATA 32,116,250, 60, 9,116,246, 78, 9
      07
[UA5] 43: 500 DATA 65,195,172, 73, 60, 32,116, 10, 7
      23
[305] 44: 510 DATA 60, 9,116, 6, 60, 13,116, 2, 3
      82
[9G5] 45: 520 DATA 235,240, 78, 65,195, 83,139,218,12
      53
[775] 46: 530 DATA 128, 63, 0,116, 3, 67,235,248, 8
      60
[195] 47: 540 DATA 198, 7, 36,180, 9,205, 33,198, 8
      66
[G65] 48: 550 DATA 7, 0, 91,195,190,128, 0,138, 7
      49
[875] 49: 560 DATA 12,181, 0,131,249, 1,118, 80, 7
      72
[FL5] 50: 570 DATA 70,232,185,255,139,214,232,193,15
      20
[375] 51: 580 DATA 255,198, 4, 0,180, 78,185, 0, 9
      00
[185] 52: 590 DATA 0,205, 33,114, 65,180, 9,186, 7
      92
[L45] 53: 600 DATA 3, 1,205, 33,186,158, 0,232, 8
      18
[DA5] 54: 610 DATA 187,255,180, 9,186, 19, 1,205,10
      42
[R55] 55: 620 DATA 33,180, 1,205, 33, 80,180, 9, 7
      21
[N85] 56: 630 DATA 186,107, 1,205, 33, 88, 36, 95, 7
      51
[1C5] 57: 640 DATA 60, 74,117, 9,180, 65,186,158, 8
      49
[F95] 58: 650 DATA 0,205, 33,114, 23,180, 79,205, 8
      39
[M95] 59: 660 DATA 33,115,202,184, 0, 76,205, 33, 8
      48
[3B5] 60: 670 DATA 186, 38, 1,235, 10,144,186, 65, 8
      65
[Z65] 61: 680 DATA 1,235, 4,144,186,110, 1,180, 8
      61
[665] 62: 690 DATA 9,205, 33,184, 1, 76,205, 33, 7
      46

```

Listing 2. Mit diesem Basic-Listing erzeugen Sie die Datei »delq.com«

Pulldown-Komfort unter GW-Basic

Pulldown-Menüs sehen gut aus und sind ein wichtiger Bestandteil eines guten Programms. Das Utility »pulldown.bas« zeigt Ihnen, wie Sie in Basic Pulldown-Menüs konstruieren. Einer Einbindung in eigene Programme steht nichts im Wege, da »pulldown.bas« dafür bewußt offen gehalten wurde.

Um das Prädikat »gut« zu erhalten, sollte ein Programm schon mit Pulldown-Menüs aufwarten können. Das in GW-Basic geschriebene Programm »pulldown.bas« bringt Ihnen die Programmierung eigener Pulldown-Menüs nahe, so daß Sie auch eigene Programme mit Pulldown-Menüs ausstatten können.

In den Zeilen 120 bis 190 wird zunächst der übliche Arbeitsbereich eingerahmt, anschließend das Menükästchen gezeichnet und durch Gosub-Einsprünge beschriftet. Für die Auswahl im Menü sind nur drei Tasten zugelassen: [Cursor aufwärts], [Cursor abwärts] und [Enter]. Diesen Tasten wird in den Zeilen 220 bis 240 jeweils eine String-Variable (\$) zugewiesen, die aus dem Tastencode besteht. Dadurch verkürzen sich anschließend die Pro-

grammzeilen, was wiederum eine bessere Übersicht gewährleistet.

Die Tastaturabfrage beginnt ab Zeile 260. Werden nun die Cursor-Tasten betätigt, so springt das Programm per »go-sub« zuerst in die Zeilen 430 bis 460. Das hell hinterlegte Feld im Menü wird durch eine dieser Zeilen in normaler Schrift überschrieben. Der zweite Sprung führt schließlich in die Zeilen 480 bis 510. Mit diesen Zeilen wird das angewählte Feld wiederum hell hinterlegt überschrieben. Anschließend beginnt die Tastaturabfrage erneut.

Mit Betätigung der Enter-Taste gelangen Sie in Zeile 520. Hier ist Raum für die eigentlichen Programme, die gestartet werden sollen.

Nachdem nun das Prinzip klar ist, dürfte es Ihnen nicht schwerfallen, dieses oder ein ähnlich aufgebautes Menü allen Ihren Basic-Programmen voranzustellen. Dazu haben Sie lediglich folgende Zeilen zu ersetzen:

```
540 load "Programm.001"
550 load "Programm.002"
560 load "Programm.003"
```

»Programm.001« bis »Programm.003« enthält jeweils den Namen des von Ihnen gewünschten Programms. In die Zeilen 430 bis 500 tragen Sie die Beschreibung Ihrer Programme ein.

»pulldown.bas« wurde bewußt offen gehalten, um Platz für Ihre eigenen Anwendungen zu lassen. Es läßt sich auch ohne großen Aufwand in andere Basic-Dialekte übertragen. Sie sehen, Pulldown-Menüs selbst zu erstellen, ist gar nicht so schwer, wie Sie vielleicht gedacht haben.

(Manfred Stövesandt/wn)

Name	pulldown.bas
Sprache	GW-Basic
Grafikkarte	jede
Besonderheiten	keine

```
[SA5] 10 'Pulldown.bas - geschrieben von
[EY3] 20 'Manfred Stövesandt
[4B6] 30 '(c) Redaktionsbüro Everts & Hagedorn
[3U1] 100 KEY OFF
[B41] 110 CLS
[3E5] 120 '***** Bildaufbau
[9RF] 130 LOCATE 2,3:PRINT STRING$(76,220):FOR I=3 TO
22:LOCATE I,3:PRINT CHR$(219):LOCATE I,78:PRINT
CHR$(219)
[DK6] 140 LOCATE 23,3:PRINT STRING$(76,223):NEXT
[GVE] 150 LOCATE 2,9:PRINT STRING$(21,32):LOCATE 1,8:C
OLOR 0,7:PRINT CHR$(201);STRING$(20,205);CHR$(1
87)
[GRB] 160 FOR I=2 TO 11:LOCATE I,8:PRINT CHR$(186):LOCA
TE I,29:PRINT CHR$(186):NEXT
[T88] 170 LOCATE 12,8:PRINT CHR$(200);STRING$(20,205);
CHR$(188)
[ZR7] 180 LOCATE 3,9:PRINT " PULLDOWN-MENÜ ":COLOR
2
[EK6] 190 GOSUB 480:GOSUB 440:GOSUB 450:GOSUB 460
[SW2] 200 'Aufbau fertig
[PC5] 210 'Stringzuweisung der Tasten
[HC3] 220 CD$=CHR$(0)+CHR$(80):
[GA7] 225 '$-Variable=Code für Cursor Down - Taste
[4L6] 230 CU$=CHR$(0)+CHR$(72):'dto für Cursor Up
[BV5] 240 CR$=CHR$(13):'dto für Enter - Taste
[2V4] 250 'Tastenabfrage und Auswahl
[SL8] 260 A$=INKEY$:IF A$=CD$ THEN GOSUB 430:GOSUB 490
:GOTO 290
[KE4] 270 IF A$=CR$ THEN A=1:GOTO 520
[4R1] 280 GOTO 260
[EL8] 290 A$=INKEY$:IF A$=CD$ THEN GOSUB 440:GOSUB 500
:GOTO 330
```

```
[PZ6] 300 IF A$=CU$ THEN GOSUB 440:GOSUB 480:GOTO 260
[5E4] 310 IF A$=CR$ THEN A=2:GOTO 520
[ØR1] 320 GOTO 290
[DL8] 330 A$=INKEY$:IF A$=CD$ THEN GOSUB 450:GOSUB 510
:GOTO 370
[CØ7] 340 IF A$=CU$ THEN GOSUB 450:GOSUB 490:GOTO 290
[ME4] 350 IF A$=CR$ THEN A=3:GOTO 520
[RP1] 360 GOTO 330
[1N8] 370 A$=INKEY$:IF A$=CU$ THEN GOSUB 460:GOSUB 500
:GOTO 330
[72C] 380 IF A$=CR$ THEN LOCATE 5,15:COLOR 27,0:PRINT
"E N D E":COLOR 2:LOCATE 18,1:END
[GR1] 390 GOTO 370
[W43] 400 'Ende der Auswahl
[Z55] 410 'Hier kommen die Einsprünge
[EY4] 420 'Teil 1 für Normalschrift
[K67] 430 LOCATE 6,9:PRINT " Programm 1 ":RETU
RN
[S67] 440 LOCATE 7,9:PRINT " Programm 2 ":RETU
RN
[Y67] 450 LOCATE 8,9:PRINT " Programm 3 ":RETU
RN
[8L7] 460 LOCATE 10,9:PRINT " Arbeit beenden ":RET
URN
[BC4] 470 'Teil 2 mit hellem Grund
[8BA] 480 COLOR 0,7:LOCATE 6,9:PRINT " Programm 1
":COLOR 2:RETURN
[FBA] 490 COLOR 0,7:LOCATE 7,9:PRINT " Programm 2
":COLOR 2:RETURN
[UAA] 500 COLOR 0,7:LOCATE 8,9:PRINT " Programm 3
":COLOR 2:RETURN
[NPA] 510 COLOR 0,7:LOCATE 10,9:PRINT " Arbeit beend
en ":COLOR 2:RETURN
[823] 520 'Unterprogramm
[FM5] 530 'Bereich für eigene Programme
[L77] 540 LOCATE 18,40:PRINT "Hier können Sie Ihr"
[5CA] 550 LOCATE 19,40:PRINT "Programm Nr.";A;"einbinde
n."CHR$(26)" ENTER"
[UL4] 560 IF INKEY$ <> CHR$(13) THEN 560
[DTB] 570 FOR I=18 TO 19:LOCATE I,40:PRINT STRING$(34,
32):NEXT:ON A GOTO 260,290,330
[BS2] 580 'Programmende
```

Mit dem GW-Basic-Listing »pulldown.bas« gestalten Sie eigene Pulldown-Menüs

Briefbogen-Druckerei für daheim

Kaum ein PC-Besitzer wird seine Korrespondenz noch mit Hilfe einer »altmodischen« Schreibmaschine erledigen, sondern er benutzt ein mehr oder weniger komfortables Textverarbeitungsprogramm. Der Briefkopf glänzt dabei meist nicht gerade durch Originalität. Das GW-Basic-Programm »briefkpf.bas« macht Schluß mit diesem Einheitsbrei.

Jeder, der einen PC oder AT sein eigen nennt, möchte auch seine Korrespondenz mit dem Computer erledigen. Das reine Briefeschreiben stellt mit einem PC und dem passenden Textverarbeitungsprogramm auch kein Problem dar, aber wenn es an das Ausgestalten des Briefkopfs geht, streiken viele Textprogramme. Es fehlen ihnen einfach einige Funktionen, mit denen sich auch der Briefkopf in optisch ansprechender Weise und mit dem »ganz persönlichen Touch« gestalten ließe.

Mit dem GW-Basic-Programm »briefkpf.bas« und einem Epson-kompatiblen Drucker können Sie nun Ihre Kreativität in individuelle Briefköpfe umsetzen und brauchen sich nicht mehr auf Ihre Textverarbeitung zu verlassen.

Name	briefkpf.bas
Sprache	GW-Basic
Grafikkarte	jede
Besonderheiten	Benötigt Epson-kompatiblen Drucker

```

[GG4] 10 REM Programm BRIEFKPF.BAS
[A07] 20 REM (c) Redaktionsbüro Everts & Hagedorn
[1X4] 30 REM Autor: Hans Bachem, Köln
[194] 40 LPRINT CHR$(27);"3";CHR$(36);
[HS1] 50 GOSUB 1000
[4P9] 60 CLS:LOCATE 10,10:INPUT "Anzahl der zu druckend
en Bögen: ";A
[DF2] 70 IF A=0 THEN END
[602] 80 FOR I=1 TO A
[NA4] 90 LPRINT CHR$(27);"w";CHR$(1);
[4G4] 100 LPRINT CHR$(27);"w";CHR$(1);
[SF4] 110 LPRINT CHR$(27);"q";CHR$(2);
[VV3] 120 LPRINT :LPRINT PNAME$
[SF4] 130 LPRINT CHR$(27);"q";CHR$(0);
[DG4] 140 LPRINT CHR$(27);"w";CHR$(0);
[GG4] 150 LPRINT CHR$(27);"w";CHR$(0);
[ZR5] 170 LPRINT:LPRINT CHR$(27);"-";CHR$(1);
[423] 180 LPRINT SPACE$(80)
[YC4] 190 LPRINT CHR$(27);"-";CHR$(0);
[WF4] 200 LPRINT CHR$(27);"!";CHR$(16);
[BM1] 210 LPRINT
[FHG] 220 LPRINT CHR$(15);CHR$(27);CHR$(83);CHR$(49);CH
R$(27);CHR$(51);CHR$(15);      Minischrift f
ür Absenderzeile
[FB8] 230 LPRINT :LPRINT "          ";CHR$(27);"-";
CHR$(1);AZEIL$
[HC4] 240 LPRINT CHR$(27);"-";CHR$(0);
[M77] 250 LPRINT CHR$(18);CHR$(27);"T";CHR$(27);CHR$(50
);
[SG4] 260 LPRINT CHR$(27);"w";CHR$(1);
[HE7] 270 LPRINT:LPRINT:LPRINT TAB(55)STR$:LPRINT
[YH4] 280 LPRINT TAB(55)PLZ$;" ";ORT$
[B03] 290 LPRINT :LPRINT
[5U4] 300 LPRINT TAB(55)"Telefon ";TEL$

```

Obwohl das Programm für Epson-kompatible Drucker entworfen wurde, ist es kein Problem, die Druckzeilen auch auf andere Drucker anzupassen. Allerdings gibt es heutzutage fast keinen Drucker mehr, der die Epson-Codes nicht versteht – ein Standard ist schon eine feine Sache. Ob Sie Einzelblatteinzug oder Endlospapier bevorzugen, spielt für »briefkpf.bas« keine Rolle. Bei Endlospapier ist lediglich durch das etwas längere Papierformat die Lochmarkierung auf die Oberkante von DIN A4 ausgerichtet. Hier kann eine Anpassung in Zeile 400 erfolgen, indem Sie das Schleifenende um 1 erhöhen.

In Zeile 90 bis 110 verwendet »briefkpf.bas« für den Namen eine Kombination aus doppelter Schrifthöhe, Breitschrift und Schattenschrift.

Eine Anpassung an Ihren eigenen Drucker ist problemlos zu bewerkstelligen

Selbstverständlich können Sie im Unterprogramm ab Zeile 1000 die Befehle »line input« durch Konstanten, also Ihren eigenen Namen und Ihre eigene Adresse ersetzen. So müssen Sie diese Daten nicht bei jedem Lauf neu eingeben.

Wie wäre es, einige dieser Briefbögen als Gastgeschenk bei Einladungen mitzubringen? Die Bögen lassen sich nämlich nicht nur in einen Drucker einspannen, sondern auch in Schreibmaschinen!

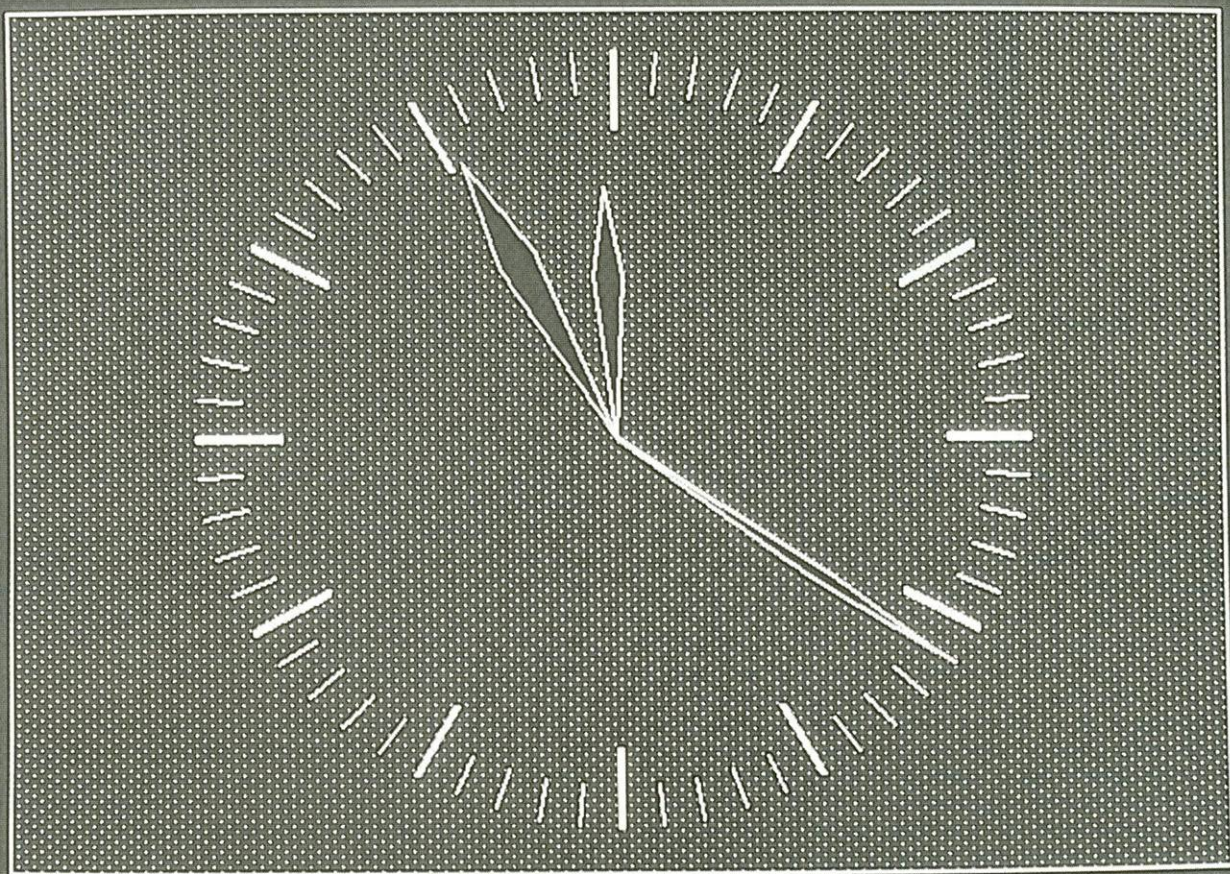
(Hans Bachem/wn)

```

[8G4] 310 LPRINT CHR$(27);"w";CHR$(0);
[807] 320 LPRINT :LPRINT "
";
[BL4] 330 LPRINT CHR$(27);"-";CHR$(1);
[W13] 340 LPRINT SPACE$(80)
[PC4] 350 LPRINT CHR$(27);"-";CHR$(0);
[2N5] 360 LPRINT:LPRINT TAB(55)ORT$;"den"
[YZ1] 370 LPRINT "-"
[ZM1] 380 LPRINT
[334] 390 LPRINT "          Betrifft:"
[T25] 400 FOR X=1 TO 8:LPRINT :NEXT X
[K22] 410 LPRINT " "
[CT2] 420 LPRINT CHR$(12)
[VG1] 430 NEXT I
[0N1] 440 GOTO 60
[E31] 1000 CLS
[R25] 1001 REM Hier beginnt die Dateneingabe
[W58] 1010 COLOR 0,7:LOCATE 2,35:PRINT " BRIEFKPF ":COL
OR 7,0
[LYH] 1020 PRINT :PRINT :PRINT "Druck-Programm für Brie
fköpfe zum Druck auf EPSON - LQ und kompatiblen D
ruckern"
[7K1] 1030 PRINT
[0U7] 1040 PRINT"          Autor: Hans B
achem, Köln
[TN2] 1050 PRINT :PRINT
[MX6] 1060 LOCATE 9,10:LINE INPUT "          Name: ",NA
M$
[ZC9] 1070 LOCATE 11,2:LINE INPUT "Name für Absenderzei
le: ",NAM2$
[L08] 1080 LOCATE 13,10:LINE INPUT "          Straße: ",S
TRA$
[427] 1090 LOCATE 15,10:LINE INPUT "          PLZ: ",P
LZ$
[G17] 1100 LOCATE 17,10:LINE INPUT "          Ort: ",O
RT$
[PB7] 1110 LOCATE 19,10:LINE INPUT "          Telefon: ",T
EL$
[F65] 1120 PNAME$=SPACE$(38-LEN(NAM$))+NAM$
[VY6] 1130 AZEIL$=" "+NAM2$+" "+STR$+" "+PLZ$+" "+OR
T$+" "
[BV1] 1140 RETURN

```

Listing. Ihre Briefbogen-Druckerei für daheim:
»briefkpf.bas«



Analoguhr für Selfmade-Designer

Es ist ja nicht so, daß digitale Uhren auf einmal gänzlich »out« wären, aber man verbindet mit ihnen immer moderne, zweckmäßige Technologie. Vielleicht ist es typisch für den derzeitigen Trend zur Humanisierung der Computer-Mensch-Schnittstelle: Heute zählt nicht mehr die Leistung allein, sondern ebenso die Anwenderfreundlichkeit.

Sie haben also den Entschluß gefaßt, ein Programm zur Anzeige einer Analoguhr in Turbo Pascal 4.0 zu schreiben. Wenn Sie sich allerdings schon die Mühe machen und Eigeninitiative entwickeln, so soll das Ergebnis wenigstens besser werden als die bisherigen Vorgänger- und Beispielprogramme. Außerdem halten Sie sich mit einem eigenen Programm alle Wege für nachträgliche Erweiterungen offen.

Der Grundgedanke ist, mit einem Grafikprogramm (zum Beispiel »Paintbrush«) eine Uhr selbst zu malen, die dann von Ihrem Programm zum Leben erweckt wird. So umgehen Sie elegant das Problem der Fabrikatfestlegung: Der eine möchte lieber ein Modell aus der Schweiz, der andere muß unbedingt ein Titan-Gehäuse haben... Die »künstlerische Gestaltung« der Uhr richtet sich also nach Ihrem persönlichen Geschmack, den Rest erledigt das Programm

(siehe Bild oben). Wenn es darum geht, die Uhrzeit auf dem Bildschirm darzustellen, müssen Sie die aktuelle Zeit natürlich erst einmal kennen. Hier nimmt Ihnen Turbo Pascal die Arbeit ab, denn in der Unit »Dos« ist eine Prozedur »GetTime« enthalten, die die aktuelle Systemzeit an das Programm übergibt. Nach dem Aufruf durch »GetTime(stunde,minute,sekunde,hundertstel)« befindet sich in den Variablen »stunde«, »minute«, »sekunde« die aktuelle Uhrzeit (alle Variablen vom Typ »Word«). Welchen Zeigerstellungen entsprechen diese Zeitangaben nun? Bild 1 gibt darüber Auskunft:

Das Zifferblatt selbst entspricht einem Vollkreis. Jeder Zeiger wird durch einen Winkel genau in seiner Lage bestimmt. Nach der Winkeldefinition, die übrigens auf die Babylonier zurückgeht, ist ein Grad als der dreihundert-sechzigste Teil des Vollkreises definiert. Der Sekundenzeiger kann nur sechzig Stellungen einnehmen, denn nach sechzig Sekunden soll der Zeiger wieder dort angekommen sein, wo er gestartet war. In jeder Sekunde legt der Sekundenzeiger also einen Winkel von sechs Grad zurück.

Es ist allgemein üblich, nach jeder Sekunde den Sekundenzeiger einfach um sechs Grad weiterzusetzen. Ganz korrekt ist dies natürlich nicht, denn die Zeit steht ja in der Zwischenzeit nicht still, sondern läuft kontinuierlich weiter. Um jedoch das »Geflacker« – das durch das Zeichnen und Löschen des Zeigers unweigerlich entsteht – nicht ins Unerträgliche zu steigern, begnügen Sie sich auf das Korrigieren der Uhr zu jeder vollen Sekunde. Damit ist Ihr Programm keineswegs minderwertig, sondern befindet sich mit dieser Darstellungsart durchaus in bester Gesellschaft; achten Sie einmal darauf, wenn Sie andere analoge Uhren sehen. Die meisten Armbanduhren stellen

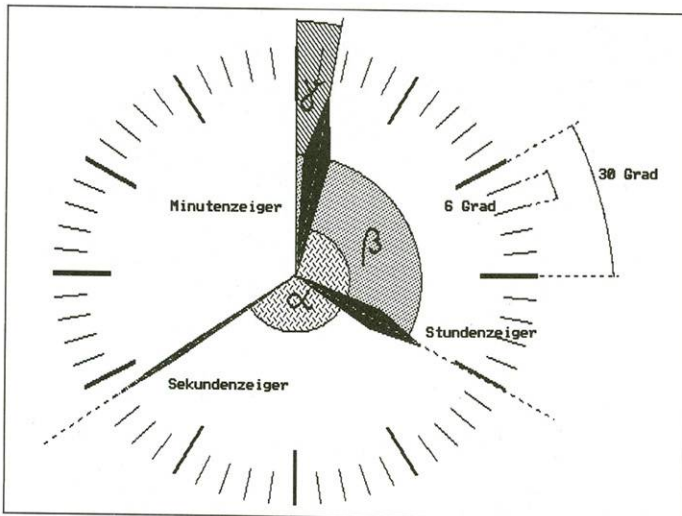


Bild 1. Die Uhrzeit muß durch die Winkel der Uhrzeiger ausgedrückt werden

den Sekundenzeiger ebenfalls derart ruckweise weiter – wenn auch nicht aus den obengenannten Gründen. Wenn Sie den Winkel »Null Grad« in der Zwölf-Uhr-Stellung festlegen, läßt sich dem Sekundenzeiger direkt ein Winkel zuordnen, indem Sie die aktuelle Sekunde mit sechs multiplizieren. Bei der Ermittlung der Winkel von Minuten- und Stundenzeiger könnten Sie theoretisch ebenso verfahren: Sie rücken den Stundenzeiger einfach nur zu jeder vollen Stunde vor, den Minutenzeiger zu jeder vollen Minute. So nimmt der Stundenzeiger aber nur zwölf Positionen ein, womit Sie dann doch eine zu grobe Verallgemeinerung vollzogen hätten. Beim Stundenzeiger sind Stellungen zwischen den vollen Stunden durchaus geläufig; es besteht ja auch ein beträchtlicher Unterschied zwischen neun Uhr, neun Uhr dreißig oder gar einer Minute vor zehn. In allen Fällen würde Ihre Uhr – vorausgesetzt, Sie vereinfachten die Darstellung so drastisch – auf neun Uhr stehen. Hier muß man also differenzieren.

Die digitalen Werte müssen durch analoge Zeigerstellungen ausgedrückt werden

Zunächst gilt die grobe Abschätzung: Der Winkel des Minutenzeigers ergibt sich durch Multiplikation von sechs mit der aktuellen Minute, weil es maximal sechzig Minuten gibt. Pro Minute bewegt sich also der Minutenzeiger um sechs Grad. Währenddessen vollführt der Sekundenzeiger eine ganze Umdrehung. Sechzig Sekunden laufen also in sechs Grad ab. Sie müssen folglich zu der Gradzahl des Minutenzeigers noch die aktuelle Sekunde mal ein Sechzigstel von sechs Grad addieren.

Ähnlich der Stundenzeiger: Die Grobabschätzung für den Winkel erhält man durch Multiplikation der aktuellen Stunde mit dreißig, denn bei zwölf Stellungen in dreihundertsechzig Grad ergibt sich alle dreißig Grad eine volle Stunde. Innerhalb der dreißig Grad zwischen zwei vollen Stunden macht der Minutenzeiger eine volle Umdrehung, es vergehen also sechzig Minuten. Bei dreißig Grad pro Stunde bleiben für eine Minute der sechzigste Teil von dreißig Grad, also ein halbes Gradmaß. Sicherlich ließe sich jetzt auch noch der Sekundenzeiger im Stundenzeiger ausdrücken, jedoch käme für jede Sekunde nur ein Winkel von 0,008333 Grad dazu – bei so kleinen Winkeln werden Sie auf Ihrem Bildschirm sicher keine Veränderung mehr bemerken.

Damit wissen Sie nun, welche Uhrzeit welchen Zeigerstellungen entspricht. Jetzt wollen Sie natürlich die Zeiger der Uhr auf dem Bildschirm darstellen. Dazu müssen Sie

zunächst einmal in den Grafikmodus umschalten. Ab Turbo Pascal 4.0 läßt sich mit den folgenden Zeilen eine einfache Initialisierung der Grafik erreichen:

```
DetectGraph(GraphDriver, GraphMode);
InitGraph(GraphDriver, GraphMode, GraphPath);
```

Die Variablen »GraphDriver« und »GraphMode« sind Integer-Typen, und die Variable »GraphPath« bezeichnet den Verzeichnispfad, unter dem Sie bei der Installation Ihrer Turbo-Pascal-Version die Dateien »*.bgi« (Grafiktreiber) und »*.chr« (Fonts) abgelegt haben. Somit erkennt Turbo Pascal die Art Ihrer Grafikkarte selbständig. Ist jedoch ein Fehler aufgetreten, konnte die Initialisierung der Grafikkarte also nicht korrekt erfolgen, so wird die Variable »GraphResult« auf einen entsprechenden Wert gesetzt (siehe Handbuch). Sie müssen in diesem Zusammenhang nur wissen, daß alles in Ordnung ist, sofern diese Variable den Wert Null angenommen hat (siehe Listing).

Ist die Grafikkarte initialisiert, so wird als nächstes der Uhrhintergrund hergestellt. Im Listing geschieht dies durch die Prozedur »UhrHintergrund«. Im einfachsten Fall löscht diese Prozedur nur den Bildschirm oder legt ein Muster über den ganzen Schirm. Für Besitzer von Hercules- und EGA-Grafikkarten ist im Programm jedoch auch die Alternative implementiert, eine Grafikdatei in den Bildschirmspeicher zu laden. Die Grafik muß im einfachen Bit-Format vorliegen, wobei die einzelnen »Farbseiten« (im Falle der EGA-Karte) nacheinander gespeichert sein müssen.

Ist der Hintergrund zufriedenstellend gestaltet, geht es an die Einteilung des Zifferblattes durch jene kleinen Striche, die jede volle Stunde oder Minute andeuten und so eine leichte Orientierung auf dem Zifferblatt zulassen. Diese Striche soll das Programm zeichnen, weil man sie von Hand in einem Malprogramm meistens nicht genau genug ziehen kann. Für das Ziehen von Strichen ist Turbo Pascal mit vielen Prozeduren ausgestattet. Bedienen Sie sich der einfachsten: »Line(x1,y1,x2,y2)«. Dieser Befehl zieht eine Linie von dem Punkt mit den Koordinaten x1,y1 zu einem anderen Punkt mit den Koordinaten x2,y2. Dabei beeinflussen Sie mit dem Befehl »SetLineStyle(style, pattern, thickness)« das Aussehen der Striche.

Ihre Einteilungsstriche sollen normale, durchgezogene Linien sein. Setzen Sie deshalb die ersten beiden Parameter dieser Anweisung auf Null. Allein der dritte Parameter hat für den gewünschten Zweck eine Bedeutung. Er gibt die Stärke der zu ziehenden Linien an. Da die Stundeneinteilungen von den Minuteneinteilungen optisch abgehoben werden sollen, wählen Sie für die Strichstärke bei jeder vollen Stunde die Stärke drei, ansonsten Stärke eins. Aber wie kommen Sie überhaupt zu den Anfangs- und Endpunkten der Einteilungsstriche? Bild 1 enthält die Antwort.

Für Berechnungen rund um den Kreis: Die Sinus- und Kosinus-Funktion

Die trigonometrischen Funktionen Sinus und Kosinus sind für dieses Vorhaben wie geschaffen (Bild 2). Die Sinus-Funktion gibt die y-Koordinate einer Strecke im Winkel »alpha« wieder, die Kosinus-Funktion die x-Koordinate. Beide Funktionen sind für den Einheitskreis definiert, das heißt, die Länge des hier verwendeten Zeigers ist eins. Um beliebige Zeigerlängen zu verwenden, multiplizieren Sie einfach den Funktionswert mit der gewünschten Länge.

Ein Problem ergibt sich dennoch: Der Winkel »alpha« liegt bei der Definition von Sinus und Kosinus an anderer Stelle als bei der Uhr. Das Gradmaß Null war im vorliegenden Beispiel bei zwölf Uhr festgelegt worden – tatsächlich

liegt das Gradmaß Null («alpha» gleich Null) aber bei drei Uhr. Da das Zifferblatt keine Zahlen trägt, sondern nur symmetrisch verteilte Einteilungsstriche, spielt es in diesem Fall keine Rolle, ob das Gradmaß Null bei zwölf oder bei drei Uhr liegt. Bei der Anzeige der Uhrzeit ist diese Tatsache allerdings sehr wohl von Bedeutung.

Die Realisation der Zifferblattanzeige läßt sich im Listing an der Prozedur »Zifferblatt« nachvollziehen. Da dieselben Sinus- und Kosinus-Werte für Anfangs- und Endpunkt jeder Einteilungslinie benötigt werden, legen Sie die Werte zu Beginn der Prozedur in den Variablen »c« (für Kosinus) und »s« (für Sinus) ab. Der Computer berechnet diese Funktionen allerdings in einem anderen System, dem sogenannten Bogenmaß. Hier wird der Winkel durch den Teilumfang festgelegt. Weil der Umfang des Vollkreises 2π (Kreiszahl $\pi = 3,1416$) beträgt, entspricht also π -hundertachtzigstel einem Grad im Gradmaß. Für eine korrekte Berechnung der Kosinus- beziehungsweise Sinus-Werte ist daher eine Multiplikation mit dem Quotienten $\pi/180$ notwendig. Der Einfachheit halber wird im ganzen Programm ein Sechzig-Grad-Maß verwendet, der Vollkreis also in sechzig gleich große Teile unterteilt. Dies bietet sich an, weil in unserem Zeitsystem sowohl maximal sechzig Sekunden als auch sechzig Minuten benutzt werden. Einer solchen Einteilung kommen sechs Grad im normalen Gradmaß gleich, weswegen Sie den Quotienten $\pi/180$ noch mit sechs multiplizieren. Und so kommen Sie schließlich zu der im Programm verwendeten Anweisung

```
s:=Sin( i * pi / 30)
```

Die Variable »i« durchläuft nun alle Werte von null bis neunundfünfzig, so daß sich der Kreis schließt (insgesamt sechzig Einteilungen).

Jetzt geht es an die Realisierung der Zeiger. Als Selbstprogrammierer streben Sie natürlich eine etwas bessere Uhr an, sie soll richtige Zeiger besitzen – nicht nur einen dünnen, wackeligen Strich. Ferner sollen die Zeiger nicht nur als Drahtrahmen im Bild stehen, sondern eine richtige Fläche aufweisen, das heißt, mit einer Farbe oder einem Muster ausgefüllt sein. Bei diesen Anforderungen bietet sich das von Turbo Pascal unterstützte Konzept der Polygon-Definition an. Polygone sind nichts anderes als normale Vielecke. Die Zeiger sollen die in Bild 1 gezeigte Form aufweisen (am Ende des Beitrags werden Sie in der Lage sein, diese Form gegebenenfalls zu verändern, falls sie Ihnen nicht gefällt).

Die Polygon-Definition in der von Turbo Pascal vorgeschriebenen Art verläuft genauso, als ob Sie die Figur von Hand zeichnen würden. Sie setzen also den Stift im Uhrmittelpunkt auf, ziehen ihn zur linken Ecke, dann zur Spitze, zur rechten Ecke und schließlich wieder zum Uhrmittelpunkt. Dieser Linienzug umfaßt fünf Eckpunkte, also definieren Sie in der Prozedur »ZeitAnzeige«:

```
TYPE ZeigerPolygon = ARRAY[1..5] OF PointType;  
VAR Zeiger = ARRAY[1..3] OF ZeigerPolygon;
```

Der vordefinierte Typ »PointType« entspricht einem Verbund aus den Intervariablen »x« und »y« für die x- und y-Koordinate des jeweiligen Punkts. Das Feld »Zeiger« stellt nun für jeden Zeiger (Sekunden, Minuten und Stundenzeiger) einen Polygonzug der Länge fünf zur Verfügung. Zu Beginn des Ausführungsteils der Prozedur »ZeitAnzeige« werden folglich auch als erstes die Zeigerpolygone definiert. In Abhängigkeit von der jeweiligen Zeigerlänge wird die Breite des Zeigers festgelegt. Jetzt läßt sich der Zeiger sehr leicht zeichnen. Mit der Prozedur »FillPoly(Anzahl,Polygonzug)« wird ein Polygon mit dem Namen »Polygonzug«, das aus »Anzahl« Ecken besteht, auf dem Bildschirm mit dem aktuellen Füllmuster gezeichnet. Nun müssen die Zeiger nur noch in Bewegung gesetzt werden. Gerade diese Aufgabe stellt die größte Herausforderung dar. Es ergeben sich zwei Probleme: Wie lassen Sie den Zeiger Ihrer Uhr rotieren, und wie sorgen Sie dafür, daß

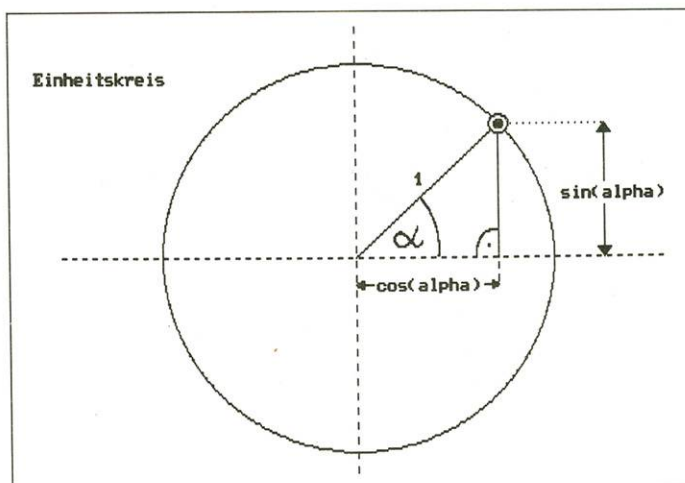


Bild 2. Bei allen kreisrunden Angelegenheiten unentbehrlich: Die Sinus- und Kosinus-Funktionen

der Bildschirmhintergrund durch die Bewegung der Zeiger nicht zerstört wird? Als aufmerksamer Leser des Kurses »Drahtmodelle im Eigenbau« in der DOS International bereitet Ihnen die Rotation natürlich keine Probleme. Bei der Drehung um einen Winkel »alpha« ergeben sich folgende neue Koordinaten:

```
Neu-X = Cos(alpha)*x-Sin(alpha)*y  
Neu-Y = Sin(alpha)*x+Cos(alpha)*y
```

Dieser Rotationsausdruck muß auf alle Eckpunkte eines Uhrzeigers angewendet werden. Dabei bleiben End- und Anfangspunkt natürlich immer im Uhrmittelpunkt, weswegen Sie sich die Berechnung dieser beiden Eckpunkte ersparen können (Prozedur »ZeichneZeiger«).

Mehr Schwierigkeiten bereitet der Hintergrund. Einige Grafikkarten bieten die Möglichkeit, zwei unabhängige Grafikseiten zu bearbeiten. Mit einer derartigen Grafikkarte (Hercules oder EGA) speichern Sie den Hintergrund (ohne Uhr) auf der verborgenen Seite und kopieren bei jeder Zeigerveränderung den Hintergrund auf die angezeigte Seite, um schließlich die neuen Zeigerpositionen einzutragen. Die Realisation dieses Verfahrens zeigt das gewünschte Ergebnis; allerdings in den meisten Fällen mit einem störenden Flackern beim Bildaufbau. Da es sich bei der Übertragung einer vollen Bildschirmgrafikseite unter Umständen um sehr große Datenmengen handelt, läßt sich diese Aktion oft mit dem Auge verfolgen. Und so stört den Betrachter zwischen zwei Sekunden immer ein lästiger Bildschirmaufbau.

Ein größerer Aufwand wird durch eine ruhigere Darstellung belohnt

Ganz ohne diese Technik – das sei vorweg gesagt – geht es natürlich nicht. Es soll versucht werden, die Übertragungszeit auf programmtechnischem Wege zu verkürzen. Zunächst könnten Sie dazu übergehen, nicht den ganzen Bildschirm zu kopieren, sondern nur den Bereich, in dem die Uhr zu sehen ist. Genauer gesagt, nur das Quadrat, in das die Uhr eingefast ist, denn das runde Zifferblatt allein läßt sich nur unter noch größerem Aufwand übertragen. Bei einer bildschirmfüllenden Uhr bringt dieses Verfahren jedoch noch nicht den erwarteten Erfolg.

Gehen Sie daher einen anderen Weg: Es wird nur der rechteckige Bereich aufgefrischt, in dem eine Zeigerbewegung stattfindet. Des weiteren wollen Sie nicht jede Sekunde alle Zeiger neu zeichnen lassen, sondern nur bei Bedarf. So ist zum Beispiel das Vorrücken des Minutenzeigers nur alle fünf Sekunden und das Weiterstellen des Stundenzeigers jede volle Minute sinnvoll, weil innerhalb

dieser Zeiträume die Vorwärtsbewegung dieser Zeiger so minimal ausfällt, daß sie auf dem Bildschirm kaum wahrzunehmen ist.

Dieses Konzept stellt allerdings einige Ansprüche an die Programmierung: Denn jetzt darf der Sekundenzeiger nicht einfach mit dem ursprünglichen Hintergrund »überpinselt« werden, sondern es ist zu berücksichtigen, daß eventuell ein anderer Zeiger, der gerade nicht korrigiert wird, unter dem Sekundenzeiger liegt und folglich dort stehen bleiben und nicht gegen den Hintergrund ausgetauscht werden soll.

Die Lösung dieses Problems sieht folgendermaßen aus: Direkt bevor Sie einen Zeiger zeichnen, kopieren Sie das Rechteck um ihn herum an eine andere Stelle im Speicher. Erst jetzt wird der Zeiger gezeichnet. Ferner müssen Sie darauf achten, daß der Zeiger zuerst gezeichnet wird, der zuletzt weiterrückt. Beim Aktualisieren der Uhrzeit ersetzen Sie nun, diesmal beim »schnellsten« Zeiger angefangen, den Bildschirmbereich des Zeigers durch Ihre vorher angefertigte Kopie.

So arbeiten Sie sich im Bedarfsfall bis zum langsamsten Zeiger, zum Stundenzeiger, vor. Durch dieses Vorgehen gewährleisten Sie, daß im Moment des Übertragens der Kopie auf den Bildschirm genau derselbe Schirmaufbau herrscht wie direkt nach dem letzten Zeichnen des Zeigers. Wäre dies nicht der Fall, so gäbe es bei bestimmten Zeigerkonstellationen immer Flächen, die falsch oder gar nicht korrigiert würden.

In Turbo Pascal stehen die Prozeduren »GetImage(Pointer,Size)« und »PutImage(Pointer,Size)« zur Verfügung, die von einem rechteckigen Bildschirmausschnitt eine Kopie anfertigen. Diese Prozeduren sind in Maschinensprache geschrieben, deswegen eignen sie sich vorrangig für eine solche Animation.

Der beschriebene Algorithmus ist im Listing gut nachvollziehbar. In der Prozedur »ZeitAnzeige« sind zwei Repeat-Schleifen geschachtelt. Die innere Schleife bildet eine Zeitverzögerung bis zu jeder vollen Sekunde. Die äußere Schleife stellt dann die Zeiger nach – solange Sie keine Taste betätigen. Die Variablen »Zeigerstellen[Zeigernummer]« vom Typ Boolean geben der Prozedur »ZeichneZeiger« an, ob der jeweilige Zeiger erneuert werden soll oder

nicht. Das Kriterium für das Weiterrücken des Minutenzeigers (Zeiger Nummer zwei) ist »sekunde mod 5 = 0«. Dieser Ausdruck ist nur dann wahr, wenn der Rest der Division der aktuellen Sekunde durch fünf gleich Null ist, also alle fünf Sekunden. Falls Ihnen eine Periode von fünf Sekunden zum »Auffrischen« der Minutenzeiger zu lang ist, so ändern Sie an dieser Stelle die Bedingung nach Ihren Wünschen.

Das Aktualisieren der Zeiger wird vollständig von der Prozedur »ZeichneZeiger« übernommen. Hier werden zunächst bei den in Frage kommenden Zeigern die neuen Koordinaten ermittelt und die Maße des neuen zu sichernden Rechtecks bestimmt. Dann werden die Zeiger in der Reihenfolge »erst schnell, dann langsam« vom Bildschirm genommen, um sie anschließend in der korrigierten Fassung und in umgekehrter Reihenfolge wieder auf den Schirm zu bringen.

Nahezu alle Eigenschaften der Uhr lassen sich leicht an Ihre Bedürfnisse anpassen

Der Rest des Programms verschafft Ihnen, ohne daß Sie sich erst näher in dessen Funktionsweise vertiefen müssen, (nur) noch einen größtmöglichen Einfluß auf die Gestalt der Uhr. Sie können eine automatische Anpassung an Ihre Grafikkarte vornehmen lassen oder selbst die benötigten Werte von Hand eingeben.

Von Bedeutung sind darüber hinaus nur die Konstanten »GraphPath« und »X_Faktor« beziehungsweise »Y_Faktor«. In »GraphPath« müssen Sie den Verzeichnispfad zu Ihren BGI-Dateien eintragen. Mit den Faktorvariablen korrigieren Sie die x- und y-Koordinaten des Programms so, daß ein Kreis auch als Kreis auf Ihrem Bildschirm erscheint. Bedingt durch die verschiedenen Verhältnisse der verfügbaren Bildschirmpunkte in der Horizontalen und in der Vertikalen kann es sein, daß Sie hier die Werte anpassen müssen. Im allgemeinen reichen aber auch die automatisch ermittelten Werte für eine gute Darstellung aus.

(Frank Höppner/zi)

Name	clock.pas
Sprache	Turbo Pascal ab Version 4.0
Grafikkarte	jede
Besonderheiten	Das Laden von Bildern ist nur für die EGA- und Hercules-Karte vorgesehen

```

[P15] 1: ( Programmname: CLOCK.PAS )
[LG4] 2: ( (C) 1989 Everts & Hagedorn )
[ZB4] 3: ( DOS International )
[000] 4:
[8K6] 5: (CLOCK.PAS - Analoguhr auf dem Bildschirm
[YP3] 6: Autor: Frank Höppner
[EC5] 7: Lauffähig ab Turbo Pascal 4.0 mit
[TK6] 8: beliebiger Grafikkarte; das Laden einer
[417] 9: konvertierten Grafik wurde jedoch nur für
[NH6] 10: Hercules und EGA implementiert.
[000] 11:
[A93] 12: USES Graph,Crt,Dos;
[000] 13:
[6X0] 14: CONST
[LT5] 15: GraphPath : STRING = 'A:\CLOCK';
[HC5] 16: Picture : STRING = 'UHR.EGA';
[6F5] 17: ( Default: Uhrhintergrunddatei )
[376] 18: Minuten_Einteilung: BOOLEAN = TRUE;
[586] 19: Stunden_Einteilung: BOOLEAN = TRUE;
[606] 20: SelbstAnpassung : BOOLEAN = FALSE;
[GA4] 21: Y_Faktor : REAL = 1.0;
[SA4] 22: X_Faktor : REAL = 1.5;
[BT4] 23: UhrRadius : INTEGER = 150;

```

```

[G45] 24: Minuten_Laenge : INTEGER = 20;
[V55] 25: Stunden_Laenge : INTEGER = 35;
[7M6] 26: ZeigerLaenge : ARRAY[1..3] OF INTEGER
[8T1] 27: : (140,120,90);
[NM6] 28: ZeigerBreite : ARRAY[1..3] OF INTEGER
[P41] 29: : (2,8,7);
[AY4] 30: UrSprungX : INTEGER = 359;
[UX4] 31: UrSprungY : INTEGER = 173;
[NH3] 32: WeckZeit : RECORD
[VV0] 33: hour,
[NS2] 34: minute : INTEGER;
[4M0] 35: END
[5K3] 36: = (hour:10; minute:45);
[000] 37:
[RN0] 38: VAR
[M52] 39: GraphDriver,
[K14] 40: GraphMode : INTEGER;
[000] 41:
[MW4] 42: ( ***** )
[000] 43:
[FR3] 44: PROCEDURE GrundWerte;
[FZ1] 45: VAR c : CHAR;
[000] 46:
[L84] 47: FUNCTION JaNein:BOOLEAN;
[XZ1] 48: VAR j : CHAR;
[MU0] 49: BEGIN
[P31] 50: REPEAT
[6Y2] 51: j:=UPCASE(ReadKey)
[Z23] 52: UNTIL j IN ['J','N'];
[5Z1] 53: WRITELN(j);
[MZ1] 54: JaNein:=j='J'
[ZN0] 55: END;
[000] 56:

```



```

[MU0] 57: BEGIN
[DB1] 58:   ClrScr;
[896] 59:   Writeln('CLOCK.PAS - analoge Uhr auf '+
[CX2] 60:   'dem Bildschirm');
[M16] 61:   Writeln('(C) 1989 DOS International');
[XF1] 62:   Writeln;
[Z26] 63:   Writeln('Parameter-Einstellungen');
[XF1] 64:   Writeln;
[A26] 65:   Writeln('Wollen Sie die (P)rogramm'+
[3Y4] 66:   'konstanten verwenden, eine '+
[1X2] 67:   '(a)utomatische');
[M35] 68:   Write('Ausrichtung oder eine '+
[664] 69:   '(m)anuelle Eingabe der '+
[XK3] 70:   'Parameter vornehmen? ');
[P31] 71:   REPEAT
[5X2] 72:   c:=UPCASE(ReadKey)
[W23] 73:   UNTIL c IN ['P','A','M',#13];
[RY1] 74:   Writeln(c);
[XF1] 75:   Writeln;
[Y91] 76:   CASE c OF
[UX0] 77:   'P' :
[LN1] 78:   'M' : BEGIN
[XU4] 79:   WRITE('BGI-Pfad. . . . .
. . ');
[H03] 80:   READLN(GraphPath);
[0H6] 81:   WRITE('Minuten-Einteilung anzei
gen: ');
[XX4] 82:   Minuten_Einteilung:=JaNein;
[5H6] 83:   WRITE('Stunden-Einteilung anzei
gen: ');
[Z44] 84:   Stunden_Einteilung:=JaNein;
[LP6] 85:   WRITE('Streckungsfaktor X-Richt
ung: ');
[GW2] 86:   READLN(X_Faktor);
[NP6] 87:   WRITE('Streckungsfaktor Y-Richt
ung: ');
[HW2] 88:   READLN(Y_Faktor);
[195] 89:   WRITE('Radius der Uhr. . . .
. . ');
[E23] 90:   READLN(UhrRadius);
[WD6] 91:   WRITE('Länge der Minuten-Strich
e . . ');
[B24] 92:   READLN(Minuten_Laenge);
[3E6] 93:   WRITE('Länge der Stunden-Strich
e . . ');
[G24] 94:   READLN(Stunden_Laenge);
[7F6] 95:   WRITE('Länge des Sekundenzeiger
s . . ');
[604] 96:   READLN(ZeigerLaenge[1]);
[GF6] 97:   WRITE('Länge des Minutenzeigers
. . ');
[804] 98:   READLN(ZeigerLaenge[2]);
[RF6] 99:   WRITE('Länge des Stundenzeigers
. . ');
[A04] 100: READLN(ZeigerLaenge[3]);
[9D6] 101: WRITE('Breite des Sekundenzeige
rs. ');
[924] 102: READLN(ZeigerBreite[1]);
[ZA6] 103: WRITE('Breite des Minutenzeiger
s . . ');
[B24] 104: READLN(ZeigerBreite[2]);
[V96] 105: WRITE('Breite des Stundenzeiger
s . . ');
[D24] 106: READLN(ZeigerBreite[3]);
[ZS6] 107: WRITE('X-Koordinate Uhrmittelpu
nkt: ');
[C33] 108: READLN(UrsprungX);
[0T6] 109: WRITE('Y-Koordinate Uhrmittelpu
nkt: ');
[D33] 110: READLN(UrsprungY);
[DM5] 111: WRITE('Weckzeit. . . . .Stu
nde: ');
[PS3] 112: READLN(WeckZeit.hour);
[6S4] 113: WRITE(' . . . . .Min
ute: ');
[824] 114: READLN(WeckZeit.minute);
[ZN0] 115: END;
[TE0] 116: 'A',
[CP4] 117: #13 : SelbstAnpassung:=TRUE;
[ZN0] 118: END;
[U52] 119: IF (c<>'P') AND
[GV3] 120: ((GraphDriver=Ega64) OR
[GR4] 121: (GraphDriver=HercMono)) THEN
[MU0] 122: BEGIN
[534] 123: IF GraphDriver=Ega64 THEN
[E83] 124: GraphMode:=Ega64Lo;
[TW4] 125: WRITE('Bilddatei . . . . .');
[ZL2] 126: READLN(picture);
[ZN0] 127: END;
[ZN0] 128: END;
[000] 129:
[UH3] 130: PROCEDURE Anpassung;
[MU0] 131: BEGIN
[882] 132: GraphPath:='';
[NM4] 133: Minuten_Einteilung:= TRUE;
[RN4] 134: Stunden_Einteilung:= TRUE;

```

```

[A94] 135: UrsprungX:=GetMaxX DIV 2;
[F94] 136: UrsprungY:=GetMaxY DIV 2;
[PB2] 137: Y_Faktor:=1.0;
[345] 138: X_Faktor:=GetMaxX/GetMaxY*0.7;
[5G5] 139: UhrRadius:=Round(0.1*UrsprungY);
[AA6] 140: Minuten_Laenge:=Round(0.1*UhrRadius);
[EB6] 141: Stunden_Laenge:=Round(0.2*UhrRadius);
[TP6] 142: ZeigerLaenge[1]:=Round(0.95*UhrRadius);
[4G6] 143: ZeigerLaenge[2]:=Round(0.8*UhrRadius);
[PP6] 144: ZeigerLaenge[3]:=Round(0.65*UhrRadius);
[0P6] 145: ZeigerBreite[1]:=Round(0.02*UhrRadius);
[FP6] 146: ZeigerBreite[2]:=Round(0.05*UhrRadius);
[EP6] 147: ZeigerBreite[3]:=Round(0.04*UhrRadius);
[EY6] 148: WeckZeit.hour:=1; ( Wecker abgestellt )
[ZN0] 149: END;
[000] 150:
[0H4] 151: PROCEDURE UhrHinterGrund;
[000] 152:
[7M4] 153: PROCEDURE RestoreEGAGraph;
[KR4] 154: ( Laden einer EGA-Grafik nach
[K86] 155: J. Gallenbach; DOS International 4'89 )
[PW2] 156: VAR count : BYTE;
[AS1] 157: fl : FILE;
[ZW4] 158: xx : ARRAY[0..$7FFF] OF BYTE
[123] 159: ABSOLUTE $A000:$0000;
[MU0] 160: BEGIN
[C93] 161: ASSIGN(fl,picture);
[BM1] 162: RESET(fl);
[D93] 163: FOR count:=0 TO 3 DO
[MU0] 164: BEGIN
[L92] 165: port[$3ce]:=1;
[L92] 166: port[$3cf]:=0;
[L82] 167: port[$3c4]:=2;
[8Y3] 168: port[$3c5]:=1 shl count;
[T83] 169: blockread(fl,xx,128)
[ZN0] 170: END;
[HD1] 171: CLOSE(fl)
[ZN0] 172: END;
[000] 173:
[2R5] 174: PROCEDURE RestoreHerculesGraph;
[2E6] 175: ( Hercules-Grafik nach selbem System )
[PW2] 176: VAR count : BYTE;
[AS1] 177: fl : FILE;
[ZW4] 178: xx : ARRAY[0..$7FFF] OF BYTE
[523] 179: ABSOLUTE $B000:$0000;
[MU0] 180: BEGIN
[C93] 181: ASSIGN(fl,picture);
[BM1] 182: RESET(fl);
[KC3] 183: BlockRead(fl,xx,256);
[HD1] 184: CLOSE(fl)
[ZN0] 185: END;
[000] 186:
[MU0] 187: BEGIN
[322] 188: ClearDevice;
[WV2] 189: IF picture='' THEN
[MU0] 190: BEGIN
[Y25] 191: Rectangle(0,0,GetMaxX,GetMaxY);
[WV4] 192: Setfillstyle(widedotfill,1);
[MG2] 193: FloodFill(2,2,1);
[BB4] 194: Setfillstyle(solidfill,1)
[SB5] 195: ( hier können Sie das Muster der
[L05] 196: Zeiger der Uhr festlegen - siehe
[D54] 197: Turbo-Manual )
[4M0] 198: END
[F14] 199: ELSE CASE GraphDriver OF
[XL5] 200: HercMono : RestoreHerculesGraph;
[904] 201: Ega64 : RestoreEGAGraph
[4M0] 202: END
[ZN0] 203: END;
[000] 204:
[L64] 205: PROCEDURE Ziffernblatt;
[B42] 206: VAR i,laenge,
[6V1] 207: xvon,yvon,
[3A3] 208: xbis,ybis : INTEGER;
[472] 209: c,s : REAL;
[MU0] 210: BEGIN
[F94] 211: IF Minuten_Einteilung OR
[974] 212: Stunden_Einteilung THEN
[MK2] 213: FOR i:=0 TO 59 DO
[MU0] 214: BEGIN
[RA2] 215: c:=Cos(i*pi/30);
[CB2] 216: s:=Sin(i*pi/30);
[HY5] 217: xbis:=Round(UhrRadius*C*X_Faktor)+
[2X1] 218: UrsprungX;
[N06] 219: ybis:=Round(UhrRadius*S*Y_Faktor)+
[1Y1] 220: UrsprungY;
[ZM2] 221: IF (i MOD 5 = 0) AND
[974] 222: Stunden_Einteilung THEN
[MU0] 223: BEGIN
[C53] 224: SetLineStyle(0,0,3);
[BW3] 225: Laenge:=Stunden_Laenge
[4M0] 226: END
[7G5] 227: ELSE IF Minuten_Einteilung THEN
[MU0] 228: BEGIN
[653] 229: SetLineStyle(0,0,1);

```

Listing. Mit dem Programm »clock.pas« bringen Sie auf dem Bildschirm Ihre eigene Analoguhr zum Laufen


```

[9V3] 230: Laenge:=Minuten_Laenge
[4M0] 231: END
[1H2] 232: ELSE Laenge:=0;
[6G2] 233: IF Laenge>0 THEN
[MU0] 234: BEGIN
[BB5] 235:   xvon:=Round((UhrRadius-Laenge)*
[MX3] 236:     C*X_Faktor)+UrsprungX;
[DB5] 237:   yvon:=Round((UhrRadius-Laenge)*
[UY3] 238:     S*Y_Faktor)+UrsprungY;
[HB4] 239:   Line(xvon,yvon,xbis,ybis)
[4M0] 240: END
[4M0] 241: END
[ZN0] 242: END;
[000] 243:
[HX3] 244: PROCEDURE ZeitAnzeige;
[D36] 245: TYPE ZeigerPolygon = ARRAY[1..5] OF
[KW1] 246:   PointType;
[763] 247:   Rechteck = RECORD
[3W1] 248:     x1,y1,x2,y2 :
[PD1] 249:       INTEGER
[ZN0] 250:   END;
[5T3] 251:   ZeigerStatus = RECORD
[4U1] 252:     grad : REAL;
[2U2] 253:     stellen : BOOLEAN;
[ZN0] 254:   END;
[HK4] 255: VAR stunde,minute,sekunde,
[8H4] 256:   hundert,lastsec : WORD;
[BH3] 257:   i : INTEGER;
[472] 258:   NeuerZeiger,
[515] 259:   Zeiger : ARRAY[1..3] OF
[SK2] 260:     ZeigerPolygon;
[U62] 261:   HilfsFlaeche,
[235] 262:   Flaeche : ARRAY[1..3] OF
[BG1] 263:     Rechteck;
[SM5] 264:   ImagePointer : ARRAY[1..3] OF
[TF1] 265:     POINTER;
[4T5] 266:   ZeigerStellen : ARRAY[1..3] OF
[9D2] 267:     ZeigerStatus;
[000] 268:
[6A4] 269: PROCEDURE ZeichneZeiger;
[SV2] 270: VAR a,b : INTEGER;
[9S1] 271:   s,c : REAL;
[MU0] 272: BEGIN
[NF2] 273:   FOR b:=1 TO 3 DO
[NP5] 274:     IF ZeigerStellen[b].stellen THEN
[MU0] 275:       BEGIN
[ML5] 276:         s:=Sin((ZeigerStellen[b].grad-15)*
[H11] 277:           pi/30);
[ZK5] 278:         c:=Cos((ZeigerStellen[b].grad-15)*
[H11] 279:           pi/30);
[C25] 280:         HilfsFlaeche[b]:=Flaeche[b];
[5D5] 281:         Flaeche[b].x1:=Pred(UrsprungX);
[8D5] 282:         Flaeche[b].y1:=Pred(UrsprungY);
[6D5] 283:         Flaeche[b].x2:=Succ(UrsprungX);
[9D5] 284:         Flaeche[b].y2:=Succ(UrsprungY);
[PF2] 285:         FOR a:=2 TO 4 DO
[MU0] 286:           BEGIN
[LX3] 287:             NeuerZeiger[b][a].x:=
[KG4] 288:               Round((c*Zeiger[b][a].x-s*
[PM4] 289:                 Zeiger[b][a].y)*X_Faktor)+
[ZK1] 290:                 UrsprungX;
[RX3] 291:             NeuerZeiger[b][a].y:=
[OG4] 292:               Round((s*Zeiger[b][a].x+c*
[SM4] 293:                 Zeiger[b][a].y)*Y_Faktor)+
[1Y1] 294:                 UrsprungY;
[944] 295:             IF NeuerZeiger[b][a].x <
[Z22] 296:                 Flaeche[b].x1 THEN
[NH2] 297:                 Flaeche[b].x1:=
[HE4] 298:                 Pred(NeuerZeiger[b][a].x)
[D75] 299:             ELSE IF NeuerZeiger[b][a].x >
[103] 300:                 Flaeche[b].x2 THEN
[RH2] 301:                 Flaeche[b].x2:=
[8M4] 302:                 Succ(NeuerZeiger[b][a].x);
[C44] 303:             IF NeuerZeiger[b][a].y <
[003] 304:                 Flaeche[b].y1 THEN
[PH2] 305:                 Flaeche[b].y1:=
[KE4] 306:                 Pred(NeuerZeiger[b][a].y)
[H75] 307:             ELSE IF NeuerZeiger[b][a].y >
[203] 308:                 Flaeche[b].y2 THEN
[SH2] 309:                 Flaeche[b].y2:=
[SE4] 310:                 Succ(NeuerZeiger[b][a].y)
[4M0] 311:             END
[ZN0] 312:           END;
[000] 313:         END;
[NF2] 314:       FOR b:=1 TO 3 DO
[3L5] 315:         IF (ZeigerStellen[b].stellen AND
[8R4] 316:           (ImagePointer[b]<>NIL) THEN
[Z34] 317:           WITH HilfsFlaeche[b] DO
[MU0] 318:             BEGIN
[LH2] 319:               PutImage(x1,y1,
[PE3] 320:                 ImagePointer[b],0);
[XB4] 321:               FreeMem(ImagePointer[b],
[XL3] 322:                 ImageSize(x1,y1,x2,y2))
[ZN0] 323:             END;
[000] 324:           END;
[E73] 325:         FOR b:=3 DOWNT 1 DO
[NP5] 326:           IF ZeigerStellen[b].stellen THEN

```

```

[173] 327: WITH Flaeche[b] DO
[MU0] 328: BEGIN
[P34] 329:   GetMem(ImagePointer[b],
[SN3] 330:     ImageSize(x1,y1,x2,y2));
[Z83] 331:   GetImage(x1,y1,x2,y2,
[K83] 332:     ImagePointer[b]);
[6H4] 333:   FillPoly(5,NeuerZeiger[b])
[4M0] 334: END
[ZN0] 335: END;
[000] 336:
[MU0] 337: BEGIN
[3G2] 338:   FOR i:=1 TO 3 DO
[MU0] 339:     BEGIN
[953] 340:       Zeiger[i][1].x:=0;
[WW2] 341:       Zeiger[i][2].x:=
[DH4] 342:         Round(0.65*ZeigerLaenge[i]);
[0X2] 343:       Zeiger[i][3].x:=
[X44] 344:         Round(ZeigerLaenge[i]);
[4X2] 345:       Zeiger[i][4].x:=
[DH4] 346:         Round(0.65*ZeigerLaenge[i]);
[T53] 347:       Zeiger[i][5].x:=0;
[C53] 348:       Zeiger[i][1].y:=0;
[1X5] 349:       Zeiger[i][2].y:=ZeigerBreite[i];
[M53] 350:       Zeiger[i][3].y:=0;
[GY5] 351:       Zeiger[i][4].y:=ZeigerBreite[i];
[W53] 352:       Zeiger[i][5].y:=0;
[AU3] 353:       ImagePointer[i]:=NIL;
[506] 354:       NeuerZeiger[i][1].x:=UrsprungX;
[C06] 355:       NeuerZeiger[i][1].y:=UrsprungY;
[906] 356:       NeuerZeiger[i][5].x:=UrsprungX;
[G06] 357:       NeuerZeiger[i][5].y:=UrsprungY;
[TS5] 358:       ZeigerStellen[i].stellen:=TRUE;
[ZN0] 359:     END;
[000] 360:
[2W6] 361:     GetTime(stunde,minute,sekunde,hundert);
[722] 362:     lastsec:=Sekunde;
[6K5] 363:     ZeigerStellen[1].grad:=sekunde;
[395] 364:     ZeigerStellen[2].grad:=minute+
[9W1] 365:       sekunde/60;
[8H5] 366:     ZeigerStellen[3].grad:=stunde*5+
[SU1] 367:       minute/60*5;
[DF2] 368:     ZeichneZeiger;
[000] 369:
[P31] 370:   REPEAT
[000] 371:     REPEAT
[P31] 372:       GetTime(stunde,minute,sekunde,
[G95] 373:         hundert)
[4G1] 374:     UNTIL sekunde<>lastsec;
[654] 375:     lastsec:=sekunde;
[722] 376:
[000] 377:     ZeigerStellen[1].grad:=sekunde;
[6K5] 378:     ZeigerStellen[2].grad:=minute+
[395] 379:       sekunde/60;
[9W1] 380:     ZeigerStellen[3].grad:=stunde*5+
[8H5] 381:       minute/60*5;
[SU1] 382:
[000] 383:     ZeigerStellen[2].stellen:=
[PN4] 384:       (sekunde MOD 5)=0;
[4X2] 385:     ZeigerStellen[3].stellen:=
[UN4] 386:       (sekunde=0);
[NY1] 387:     ZeichneZeiger;
[DF2] 388:     IF (WeckZeit.minute=minute) AND
[0A5] 389:       (WeckZeit.hour=stunde) THEN
[8P4] 390:       Sound(440)
[MG1] 391:     ELSE NoSound
[H62] 392:
[000] 393:   UNTIL keypressed;
[K13] 394: END;
[ZN0] 395:
[000] 396: BEGIN (Hauptprogramm)
[9D4] 397:
[000] 398:   DetectGraph(GraphDriver,GraphMode);
[A86] 399:
[000] 400:   GrundWerte;
[W32] 401:
[000] 402:   InitGraph(GraphDriver,GraphMode,
[LK5] 403:     GraphPath);
[7X1] 404:   IF GraphResult<>0 THEN
[WN3] 405:     BEGIN
[MU0] 406:       ClrScr;
[DB1] 407:       Writeln;
[XF1] 408:       Writeln('#7'Graphik-Initialisierung '+'
[G36] 409:         'erfolglos! Falscher Pfad?'#7);
[8Y4] 410:       Writeln;
[XF1] 411:       Readln;
[991] 412:       Halt
[RP0] 413:     END;
[ZN0] 414:
[000] 415:     IF SelbstAnpassung THEN Anpassung;
[SV5] 416:     UhrHinterGrund;
[DW2] 417:     ZiffernBlatt;
[WA2] 418:     ZeitAnzeige;
[A62] 419:
[000] 420:   CloseGraph
[TU1] 421:
[000] 422:
[KN0] 423: END.

```

Listing. Mit dem Programm »clock.pas« bringen Sie auf dem Bildschirm Ihre eigene Analoguhr zum Laufen

Grafik- konvertierung Schraffur statt Farbe

Zusammen mit der Microsoft-Maus und auch mit einigen EGA-Grafikkarten werden das Programm »Paintbrush« oder ähnliche Grafikprogramme (»Dr. Halo«) geliefert. Diese Programme versetzen Sie in die Lage, auf relativ komfortable Weise Grafiken zu erzeugen, zu speichern und auszu-
drucken. Für den Programmierer, der die Grafiken in eigenen Programmen nutzen möchte, erweist sich als Nachteil, daß die Bilddateien in komprimiertem Format vorliegen. Dieses Handicap kann jedoch behoben werden.

Wer mit Grafikprogrammen wie zum Beispiel »Paintbrush« oder »Dr. Halo« erzeugte Bilder (beziehungsweise Bilddateien) in selbstgeschriebenen Programmen (zum Beispiel dem Analoguhrprogramm in dieser Ausgabe) verwenden möchte, steht fast immer vor dem Problem, mit den komprimierten Dateien nicht viel anfangen zu können, weil die Daten nicht dem tatsächlichen Inhalt des Grafikspeichers entsprechen. Zwar existieren Konvertierungsprogramme, doch übersetzen sie die Bilddateien meist nur vom Format eines Grafikprogramms in das eines anderen – der Inhalt des Grafikspeichers im »Klartext« bleibt dennoch verborgen.

Für die Hercules-Karte gibt es schon mehrere Programme auf dem Markt, die das aktuelle Bild auf Tastendruck »einfangen« und speichern. Doch auch auf einer EGA-Grafikkarte in der Betriebsart 640x200 Punkte mit 16 Farben funktioniert dieses (Paintbrush nutzt diesen Modus). Damit bekommen Sie ferner die Möglichkeit, Bilder, die mit einer EGA-Grafikkarte erzeugt wurden, auf eine Hercules-Karte zu übertragen.

Die Programmierung der EGA-Grafikkarte ist eine sehr komplexe Thematik. Es werden daher nur die unmittelbar im Programm »savescr.asm« (Listing 1) eingesetzten Funktionen erläutert. Der Speicher der EGA-Grafikkarte ist in vier Ebenen aufgeteilt, die der Adressierung nach »hintereinander liegen« (von Adresse A000:0000hex bis A000:FFFFhex; bei 640x200 Punkten nur bis A000:3FFFhex). Diese Ebenen lassen sich getrennt lesen und getrennt – oder in Kombinationen – beschreiben. In

jeder der Ebenen ist jeweils ein Bit für die 640x200 Punkte vorgesehen. Insgesamt ergeben sich also 4 Bit pro dargestelltem Punkt. Somit sind 2^4 (gleich 16) verschiedene Farben realisierbar.

Welche Ebenen Sie lesen beziehungsweise beschreiben, teilen Sie der Grafikkarte über deren Register mit, die Sie über Port-Adressen erreichen. Zum Lesen schreiben Sie in Port 3CEhex (Graphic Address Register) eine 4, um das Read-Map-Register zu adressieren. Als nächstes wird in Port 3CFhex ein Wert zwischen 0 und 3 (für die Ebenen 0 bis 3) geschrieben. Beim Lesen der Speicheradressen A000:0000hex bis A000:3FFFhex erhalten Sie nun die grafischen Daten der selektierten Ebene.

Da die Ebenen übereinandergelagert die Bitmuster für farbige Punkte ergeben, ist es sinnvoll, auch mehrere Ebenen gleichzeitig mit demselben Wert zu beschreiben (zum Beispiel Ebene 0 und 2 für Farbe Nummer 5; dies entspricht dem Binärmuster »0101«). Genau dies wurde bei der EGA-Karte realisiert.

Ein Maschinenprogramm sichert problemlos Ihre EGA-Grafik als Datei

Bei der Programmierung der Register zum Beschreiben der Ebenen werden Bit 0 bis 3 (also vier einzelne Bit) und nicht die Werte 0 bis 3 (darstellbar durch zwei Bit) verwendet. Zuerst bestimmen Sie das Register 1 (»Enable Set/Reset Register«), indem Sie Port 3CEhex den Wert 1 zuweisen. Anschließend schreiben Sie über Port 3CFhex den Wert 0 in dieses Register und übertragen dadurch die Kontrolle über die zu beschreibenden Seiten (Ebenen) ausschließlich dem Map-Mask-Register.

Als nächstes wählen Sie mit der Zuweisung des Wertes 2 an den Port 3C4hex das genannte Register und setzen jeweils ein Bit (0 bis 3) über Port 3C5hex, um in nur eine Speicherseite zu schreiben. Die Selektion der zu lesenden und der zu beschreibenden Ebene ist übrigens unabhängig voneinander. So kann man beispielsweise in Turbo Pascal bei geeigneter Registerprogrammierung mit

```
mem[$A000:n] := mem [$A000:n]
```

Daten einer Seite in eine andere kopieren.

Das in Assembler geschriebene Programm »savescr.asm« (Listing 1) setzt sich aus einem residenten und einem »flüchtigen« Installationsteil zusammen. In letzterem wird der Interrupt 5 (PrtSc-Behandlung) auf den residenten Teil »umgeleitet« und über den MS-DOS-Interrupt 27hex der Speicher für diesen Teil freigehalten.

Die Interrupt-Behandlungsroutine eröffnet im aktuellen Verzeichnis Ihrer Festplatte oder Diskette eine Datei mit

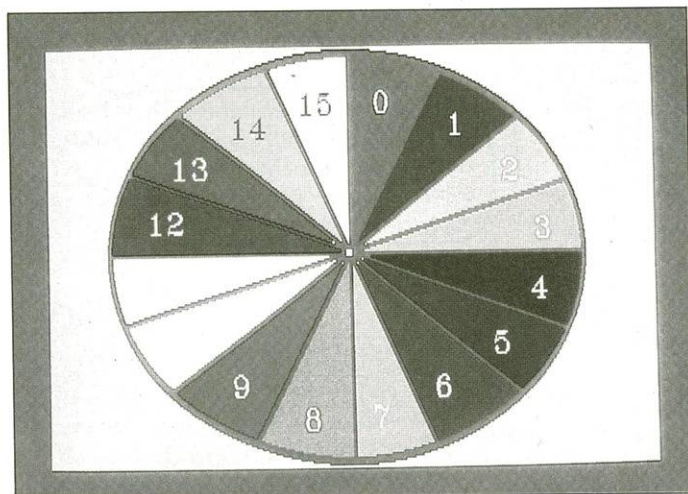


Bild 1. »Ega2hgc« erzeugt aus einer EGA-Grafik mit 16 Farben...

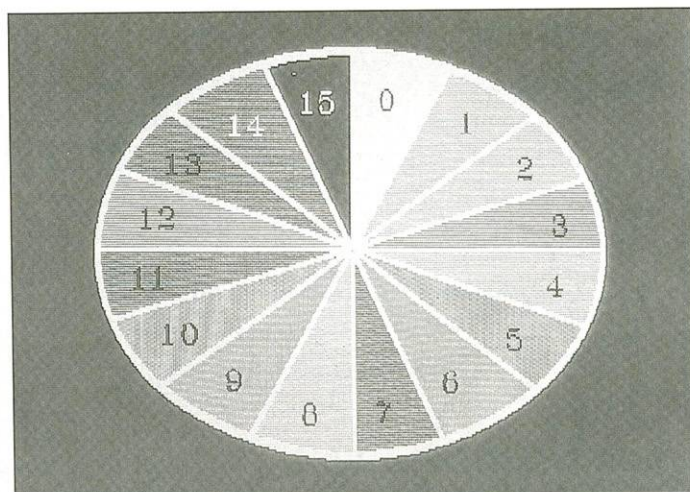


Bild 2. ...eine Hercules-Grafik mit entsprechenden Schraffuren

Farbe 0 (Schwarz)	Farbe 1 (Blau)	Farbe 2 (Grün)	Farbe 3 (Türkis)
Farbe 4 (Rot)	Farbe 5 (Fuchsin)	Farbe 6 (Braun)	Farbe 7 (Hellgrau)
Farbe 8 (Dunkelgrau)	Farbe 9 (Hellblau)	Farbe 10 (Hellgrün)	Farbe 11 (Helles Türkis)
Farbe 12 (Hellrot)	Farbe 13 (Helles Fuchsin)	Farbe 14 (Gelb)	Farbe 15 (Weiß)

Tabelle 1. Die Anordnung der Farbebenen der EGA-Karte (die Angaben in Klammern beziehen sich auf die jeweilige Standardfarbe)

dem Namen »bild.pix« und schreibt nacheinander die Bytes der vier EGA-Bildebenen hinein. Sie müssen allerdings sicherstellen, daß auf Ihrem Speichermedium genügend freier Speicherplatz (128 KByte) zur Verfügung steht.

Um Grafiken nicht eventuell zu zerstören, wurde auf die Ausgabe einer Fehlermeldung bei Problemen mit der Datei verzichtet. Das Schließen der Datei ist sehr wichtig, damit keine Daten verlorengehen. Das ladende Programm »loadscr.pas« (Listing 2) wurde hier in Turbo Pascal 4.0 realisiert. Versierte Programmierer sollten in der Lage sein, das Programm anhand der Kommentare in andere Programmiersprachen zu übertragen. Das Maschinenprogramm starten Sie von der Betriebssystemebene aus mit

savescr [Enter]

Wenn Sie im Anschluß daran die Tastenkombination [Shift-PrtSc] betätigen, speichert die Routine das aktuelle Bild. Der Dateiname wird dabei nicht verändert. Falls Sie also mehrere Bilder auf Ihrer Festplatte (oder Diskette) ablegen möchten, müssen Sie der Datei »bild.pix« mit dem MS-DOS-Kommando »rename« einen anderen Namen zuweisen, weil das Programm sonst jeweils die letzte Datei überschreibt. Beachten Sie bitte, daß die Routine nicht gegen einen wiederholten Aufruf gesichert ist. Falls Sie »savescr.com« daher ohne vorhergehenden Warmstart (Tastenkombination [Ctrl-Alt-Del]) oder Betätigen der Reset-Taste an Ihrem PC (sofern vorhanden) mehrfach laden, so belegt das Programm jeweils zirka 300 Byte Ihres Hauptspeichers.

Bei der Verwendung mit »Paintbrush« sind einige Besonderheiten zu beachten:

- Vor dem Speichern des aktuellen Bildes ist der Menüpunkt »Show Screen« zu wählen, denn Sie haben sicherlich kein Interesse daran, das Bild mit dem Rahmen des Zeichenprogramms zu sichern.

A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P

Tabelle 2. So sind die Buchstaben A bis P den 2x2-Schraffurmustern zugeordnet

– »Paintbrush« lädt das speicherresidente Utility »Frieze«, mit dem man Bilder im MS-Format speichern kann. Um Ihre Routine »Savescr« zu aktivieren, müssen Sie deshalb nach Betätigung der Tastenkombination [Shift-PrtSc] noch die Option »PrtSc« auswählen.

Wenn Sie Titelseiten oder generell Bilder zu eigenen Programmen erzeugen, so tun Sie dies sicherlich im »besten« Grafikmodus – sofern die hardwaremäßigen Voraussetzungen erfüllt sind, also im EGA-Modus. Die Grafikprogramme bieten jedoch leider keine Möglichkeit, die oft aufwendig zusammengestellte Grafik in den Hercules-Standard zu überführen. Auf Computern, die nur mit der Hercules-Grafikkarte ausgerüstet sind, lassen sich diese Bilder folglich nicht einsetzen.

Durch Manipulation der Bilddatei werden EGA-Bilder auch für die Hercules-Karte zugänglich

Für diese und ähnliche Situationen wurde das Programm »Ega2hgc« (sprich: »EGA to HGC«) in Turbo Pascal konzipiert. Es lädt die Datei »bild.pix« in den Arbeitsspeicher und stellt die 16 Farben (Tabelle 1) getrennt auf dem Hercules-Grafikbildschirm dar (Bild 1). Sie wählen nun aus, welcher Farbe welche Schraffur zugeordnet werden soll. Die Schraffuren sind sehr einfach und haben eine Matrix von nur 2x2 Punkten (Tabelle 2). Bei einer größeren Auswahl würde jedoch die Auflösung des Bildes zu sehr darunter leiden (grobe Rasterung).

Um die Schraffur-Farbzuordnung zu treffen, drücken Sie eine der Buchstabentasten [A] bis [P], wenn ein blinkender Rahmen um das betreffende Teilbild erscheint. Der Rechner erzeugt nun ein Bild mit den gewählten Schattierungen (Bild 2). Ist dies geschehen, so betätigen Sie die Taste [J], um das neue Bild mit dem Namen »herc.pix« zu speichern, oder [N], falls Sie dies nicht wünschen. Um den Grafikbildschirm nicht zu zerstören, wurde auf eine Abfrage mit Text verzichtet; die Reihenfolge der Eingaben ist dennoch klar ersichtlich.

(Jens Gallenbacher/zi)



Name	savescr.asm
Sprache	MASM ab Version 4.0, TASM
Grafikkarte	EGA
Besonderheiten	keine

```

[9V4] 1: ; Programmname: SAVESCR.ASM
[WY3] 2: ; (C) 1989 Everts & Hagedorn
[8U3] 3: ; DOS International
[000] 4:
[C85] 5: ;by J. Gallenbacher; Bad Homburg
[506] 6: ;Assemblieren mit MASM, LINK, EXE2BIN
[000] 7:
[AD2] 8: code segment
[HW4] 9: ;MASM-Header für COM-Dateien
[8U3] 10: assume cs:code, ds:code
[A61] 11: org 100h
[SH2] 12: start: jmp beg
[817] 13: ;der nichtresidente Teil steht am Schluß
[000] 14:
[M32] 15: int5 equ 5h
[9X4] 16: ;INT 5 soll 'umgebogen' werden
[F43] 17: intadr equ int5 * 4
[3U4] 18: ;die Adresse des Interrupts
[PT2] 19: savlen equ 0400h
[EZ5] 20: ;Länge einer zu speichernden Ebene
[000] 21:
[B73] 22: ;bei 640x200 Punkten:
[YR3] 23: ;16 KByte = 4000hex Byte
[000] 24:
[E93] 25: data db 'BILD.PIX', 0
[LW6] 26: ;das Bild wird unter diesem Namen abgelegt
[5X1] 27: interrupt:
[PP6] 28: ;diese Routine wird bei PrtSc aufgerufen
[6P0] 29: sti
[FH4] 30: ;alle Interrupts freigeben
[PC1] 31: push ax
[NK3] 32: ;die Register sichern
[TC1] 33: push bx
[WC1] 34: push cx
[ZC1] 35: push dx
[BC1] 36: push ds
[000] 37:
[SP1] 38: mov ax, cs
[UP1] 39: mov ds, ax
[Z16] 40: ;das Datensegment auf das Codesegment
[EU4] 41: ;legen, um 'data' zu erreichen
[7Y1] 42: lea dx, data
[HR1] 43: xor cx, cx
[SV1] 44: mov ah, 3ch
[BK5] 45: ;Ausgabedatei 'BILD.PIX' eröffnen
[381] 46: int 21h
[Y41] 47: jc ende
[TW3] 48: ;bei Disk-Fehler beenden
[000] 49:
[5R1] 50: mov bx, ax
[TC1] 51: push bx
[DK5] 52: ;Dateinummer nach bx übertragen
[642] 53: ;und sichern
[T12] 54: mov dx, 03ceh
[ZV5] 55: ;EGA auf Lesen der Seite 0 einstellen
[XC1] 56: mov al, 4
[LP1] 57: out dx, al
[U12] 58: mov dx, 03cfh
[NC1] 59: mov al, 0
[LP1] 60: out dx, al
[YE2] 61: mov cx, savlen
[F54] 62: ;savlen Bytes schreiben
[8C3] 63: ;ab Adresse $A000:$0000
[NR1] 64: xor dx, dx
[452] 65: mov ax, 0a000h
[UP1] 66: mov ds, ax
[U11] 67: mov ah, 40h
[2E5] 68: ;MS-DOS-Funktion 40hex aufrufen
[VM3] 69: ;(= in Datei schreiben)
[381] 70: int 21h
[000] 71:
[T12] 72: mov dx, 03ceh
[ZW5] 73: ;EGA auf Lesen der Seite 1 einstellen
[XC1] 74: mov al, 4
[LP1] 75: out dx, al
[U12] 76: mov dx, 03cfh
[RC1] 77: mov al, 1
[LP1] 78: out dx, al
[T31] 79: pop bx
[BD4] 80: ;Dateinummer restaurieren
[TC1] 81: push bx
[YE2] 82: mov cx, savlen
[F54] 83: ;savlen Bytes schreiben
[NR1] 84: xor dx, dx
[452] 85: mov ax, 0a000h

```

```

[UP1] 86: mov ds, ax
[U11] 87: mov ah, 40h
[381] 88: int 21h
[000] 89:
[T12] 90: mov dx, 03ceh
[5W5] 91: ;EGA auf Lesen der Seite 2 einstellen
[XC1] 92: mov al, 4
[LP1] 93: out dx, al
[U12] 94: mov dx, 03cfh
[TC1] 95: mov al, 2
[LP1] 96: out dx, al
[T31] 97: pop bx
[BD4] 98: ;Dateinummer restaurieren
[TC1] 99: push bx
[YE2] 100: mov cx, savlen
[F54] 101: ;savlen Bytes schreiben
[NR1] 102: xor dx, dx
[452] 103: mov ax, 0a000h
[UP1] 104: mov ds, ax
[U11] 105: mov ah, 40h
[381] 106: int 21h
[000] 107:
[T12] 108: mov dx, 03ceh
[8W5] 109: ;EGA auf Lesen der Seite 3 einstellen
[XC1] 110: mov al, 4
[LP1] 111: out dx, al
[U12] 112: mov dx, 03cfh
[VC1] 113: mov al, 3
[LP1] 114: out dx, al
[T31] 115: pop bx
[BD4] 116: ;Dateinummer restaurieren
[TC1] 117: push bx
[YE2] 118: mov cx, savlen
[F54] 119: ;savlen Bytes schreiben
[NR1] 120: xor dx, dx
[452] 121: mov ax, 0a000h
[UP1] 122: mov ds, ax
[U11] 123: mov ah, 40h
[381] 124: int 21h
[T31] 125: pop bx
[BD4] 126: ;Dateinummer restaurieren
[YV1] 127: mov ah, 3eh
[S13] 128: ;Datei schließen
[381] 129: int 21h
[000] 130:
[CY3] 131: ;Register restaurieren
[B52] 132: ende: pop ds
[X31] 133: pop dx
[V31] 134: pop cx
[T31] 135: pop bx
[R31] 136: pop ax
[GR0] 137: iret
[ZC6] 138: ;Interrupt-Service-Routine verlassen
[000] 139:
[DC4] 140: ;Ende des residenten Teils
[000] 141:
[NV5] 142: messge db 13, 10, 'Treiber erfolgreich '
[CL3] 143: db 'installiert.', 13, 10
[RU4] 144: db 'ACHTUNG! Bei Disk-Fehlern '
[KD4] 145: db 'erfolgt keine Meldung!',
[RV1] 146: db 13, 10, 10, '$'
[000] 147:
[GT5] 148: ;Warnung bei Aufruf des Programms
[000] 149:
[KP2] 150: beg: xor ax, ax
[WP1] 151: mov es, ax
[Y03] 152: ;ES mit $0000 laden
[6P0] 153: sti
[6N7] 154: ;Interrupts während der Veränderung eines
[YM4] 155: ;Interruptvektors sperren
[XD2] 156: mov di, intadr
[SG5] 157: ;Adresse in Indexregister laden
[YY2] 158: lea bx, interrupt
[H46] 159: ;neuen Vektor (Ofs) in die Adresse laden
[ET2] 160: mov es:[di], bx
[N81] 161: add di, 2
[F36] 162: ;neuen Vektor (Seg) in die Adresse laden
[SP1] 163: mov ax, cs
[BT2] 164: mov es:[di], ax
[EM0] 165: cli
[FY4] 166: ;Interrupts wieder zulassen
[GA2] 167: lea dx, messge
[D86] 168: ;Warnung über MS-DOS-Funktion 09hex
[6H1] 169: ;ausgeben
[VC1] 170: mov ah, 9
[381] 171: int 21h
[GA2] 172: lea dx, messge
[816] 173: ;Programm beenden und dabei alles bis
[KC6] 174: ;messge resident im Speicher belassen
[N81] 175: int 27h
[000] 176:
[G55] 177: ;MASM-Schluß für COM-Dateien
[ZP1] 178: code ends
[4K1] 179: end start

```

Listing 1. »savescr.asm« speichert einen EGA-Bildschirminhalt als binäres Abbild

Name	loadscr.pas
Sprache	Turbo Pascal ab Version 4.0
Grafikkarte	EGA, Hercules-Grafikkarte
Besonderheiten	keine

```

[7C5] 1: ( Programmname: LOADSCR.PAS )
[LG4] 2: ( (C) 1989 Everts & Hagedorn )
[ZB4] 3: ( DOS International )
[000] 4:
[646] 5: ( RESTORE-Routine für EGA-Bilder )
[VT5] 6: ( by J. Gallenbacher, Bad Homburg )
[000] 7:
[VT3] 8: program restoregraph;
[000] 9:
[RR2] 10: uses graph, crt;
[000] 11:
[5R4] 12: type string80 = string [80];
[000] 13:
[K83] 14: var gm, gd : integer;
[000] 15:
[YB6] 16: procedure restgraph (name : string80);
[PW2] 17: var count : byte;
[AS1] 18: fl : file;
[5X4] 19: xx : array [0..$7fff] of byte
[123] 20: absolute $a000:$0000;
[000] 21:
[MU0] 22: begin
[BU2] 23: assign (fl, name);
[LK5] 24: ( Bilddatei eröffnen )
[HR1] 25: reset (fl);
[TS4] 26: for count := 0 to 3 do begin
[6F5] 27: ( Seite 0-3 beschreiben )
[SL2] 28: port [$3ce] := 1;
[UL2] 29: port [$3cf] := 0;
[VL6] 30: ( EGA auf Schreiben der Seite count )
[FP4] 31: ( einstellen )
[6K2] 32: port [$3c4] := 2;
[NC4] 33: port [$3c5] := 1 shl count;
[Z83] 34: blockread (fl, xx, 128)
[K46] 35: ( interessante Farb-Effekte ergeben
[UY5] 36: sich, wenn hier die Reihenfolge des
[ZL5] 37: Ladens durcheinandergebracht wird
[CE4] 38: (z.B. durch '1 shl (3-count)').
[FY4] 39: 16 KByte der Datei in den Grafik-
[KK5] 40: speicher laden, bei EGA-Karten mit
[B96] 41: mehr RAM kann der Wert erhöht werden,
[KW6] 42: um größere Auflösungen zu speichern
[H66] 43: (gleichzeitig das Assembler-Programm
[9B4] 44: modifizieren!)
[ZN0] 45: end;
[WE1] 46: close (fl)
[KX5] 47: ( Bilddatei schließen )
[ZN0] 48: end;
[000] 49:
[MU0] 50: begin
[SL1] 51: gd := ega64;
[LL1] 52: gm := ega640;
[F16] 53: ( 640 x 200 Bildpunkte, 16 Farben (wie
[627] 54: ( von PAINTBRUSH genutzt) selektieren )
[G73] 55: initgraph (gd, gm, '');
[KR3] 56: restgraph ('bild.pix');
[LM4] 57: ( Grafik laden )
[Y44] 58: repeat until keypressed;
[TU1] 59: closegraph
[D15] 60: ( Grafik schließen )
[KN0] 61: end.

```

Listing 2. »loadscr.pas« macht die EGA-Bilddatei Ihren eigenen Programmen zugänglich

Name	ega2hgc.pas
Sprache	Turbo Pascal ab Version 4.0
Grafikkarte	EGA, Hercules-Grafikkarte
Besonderheiten	keine

```

[985] 1: ( Programmname: EGA2HGC.PAS )
[LG4] 2: ( (C) 1989 Everts & Hagedorn )
[ZB4] 3: ( DOS International )
[000] 4:
[HE6] 5: ( Enhanced to Hercules Graphic Adapter )
[4W5] 6: ( by J. Gallenbacher, Bad Homburg )
[000] 7:
[MY2] 8: program egatohga;
[000] 9:
[RR2] 10: uses graph, crt;
[000] 11:
[5R4] 12: type string80 = string [80];
[PS5] 13: graphmem = array [0..$3fff] of byte;

```

```

[ZB3] 14: graphpnt = graphmem;
[9E6] 15: ( Speicherarray für Grafikdaten )
[1A6] 16: ( als Pointer, um genügend Speicher )
[4N5] 17: ( zur Verfügung zu haben. )
[336] 18: graphfull = array [0..3] of graphpnt;
[5G5] 19: ( 4 Ebenen werden benötigt )
[000] 20:
[K83] 21: var gm, gd : integer;
[RE1] 22: ch : char;
[663] 23: egamem : graphfull;
[BA5] 24: ( Pointer auf EGA-Grafik )
[CB3] 25: c, x, y, xl, yl : word;
[666] 26: ( Index-Variablen für Farbe und )
[5K4] 27: ( Koordinaten )
[B85] 28: colpatrn : array [0..15] of byte;
[SU6] 29: ( Zuordnungstabelle Farbe -> Schraffur )
[ZG2] 30: herfile : file;
[KM5] 31: hergraph : array [0..$7fff] of byte
[523] 32: absolute $b000:$0000;
[7S6] 33: ( Array auf Hercules-Speicherbereich )
[000] 34:
[116] 35: procedure restgraph (name: string80;
[GB3] 36: var xx: graphfull);
[NY5] 37: ( Einlesen der EGA-Grafik in den )
[ZB3] 38: ( Speicher )
[PW2] 39: var count : byte;
[7F1] 40: fl : file;
[MU0] 41: begin
[PR5] 42: ( Reservieren des Speichers )
[S82] 43: new (xx [0]);
[V82] 44: new (xx [1]);
[Y82] 45: new (xx [2]);
[192] 46: new (xx [3]);
[BU2] 47: assign (fl, name);
[N65] 48: ( Grafikdatei öffnen )
[HR1] 49: reset (fl);
[TS4] 50: for count := 0 to 3 do begin
[Z45] 51: ( 4 Farbebenen einlesen )
[R05] 52: blockread (fl, xx [count], 128)
[ZN0] 53: end;
[WE1] 54: close (fl)
[8L5] 55: ( Grafikdatei schließen )
[ZN0] 56: end;
[000] 57:
[MU0] 58: begin
[SD2] 59: gd := hercmono;
[6P2] 60: gm := hercmonohi;
[ZS5] 61: ( Auf Herculesgrafik schalten )
[G73] 62: initgraph (gd, gm, '');
[ZW4] 63: restgraph ('bild.pix', egamem);
[UX4] 64: ( EGA-Grafik einlesen )
[RE5] 65: ( Die 16 Farbebenen getrennt )
[H94] 66: ( darstellen )
[3K3] 67: for c := 0 to 15 do begin
[VF5] 68: ( Schleife für die Farbebene )
[KS2] 69: for yl := 0 to 86 do
[U96] 70: ( Jedes Teilbild hat 87 Punkte in y- )
[7X3] 71: for xl := 0 to 179 do begin
[PH5] 72: ( und 180 Punkte in x-Richtung )
[4P2] 73: y := yl * 200 div 87;
[GL2] 74: x := xl * 32 div 9;
[YL5] 75: ( Umrechnung der x/y-Werte der )
[ZV5] 76: ( Herculesgrafik in die der EGA )
[9K5] 77: if (ord (egamem[0][y*80 + x div 8]
[W63] 78: and (1 shl (7 - x mod 8)) <> 0)
[0X4] 79: = (c shr 0) and $01) and
[8Z3] 80: (ord (egamem[1][y*80 + x div 8]
[263] 81: and (1 shl (7 - x mod 8)) <> 0)
[3X4] 82: = (c shr 1) and $01) and
[8Z3] 83: (ord (egamem[2][y*80 + x div 8]
[273] 84: and (1 shl (7 - x mod 8)) <> 0)
[6X4] 85: = (c shr 2) and $01) and
[8Z3] 86: (ord (egamem[3][y*80 + x div 8]
[6L2] 87: and (1 shl (7 - x mod 8)) <> 0)
[E54] 88: = (c shr 3) and $01)
[SG1] 89: then putpixel (xl + (c mod 4)
[A02] 90: * 180, yl + (c
[3K4] 91: div 4) * 87, 1);
[7A4] 92: ( Wenn die Bits in den Ebenen
[BT4] 93: der EGA-Grafik der Farbe 'c'
[ZS4] 94: entsprechen, dann setze den
[EC4] 95: Punkt am Bildschirm, jeweils
[KB3] 96: in den für 'c' vorgesehenen
[4M0] 97: Bereich )
[ZN0] 98: end;
[C96] 99: end;
[1B6] 100: ( Abfragen, in welcher Schraffur die )
[M35] 101: ( jeweiligen Farben dargestellt )
[3K3] 102: ( werden sollen (A - P) )
[P31] 103: for c := 0 to 15 do begin
[P31] 104: repeat
[Z46] 105: ( Kennlichmachen der abgefragten )
[TR5] 106: ( Farbe (blinkendes Rechteck) )
[2P3] 107: rectangle ((c mod 4) * 180,
[4V1] 108: (c div 4) * 87,
[FG2] 109: (c mod 4) * 180 + 179,

```

Listing 3. »ega2hgc.pas« konvertiert EGA-Bilddateien in Hercules-Bilddateien


```

[YD2] 111:      (c div 4) * 87 + 86);
[F12] 112:      setcolor (0);
[2P3] 113:      rectangle ((c mod 4) * 180,
[4V1] 114:      (c div 4) * 87,
[FG2] 115:      (c mod 4) * 180 + 179,
[YD2] 116:      (c div 4) * 87 + 86);
[XX1] 117:      setcolor (1)
[K13] 118:      until keypressed;
[DM3] 119:      ch := upcase (readkey);
[RV4] 120:      ( Schraffur zuordnen )
[TF3] 121:      if ch in ['A'..'P'] then
[GE4] 122:      colpatrn [c] := ord (ch) - 65
[5L3] 123:      until ch in ['A'..'P'];
[LB6] 124:      ( Wiederholung bei unzulässiger )
[8Y3] 125:      ( Eingabe )
[ZN0] 126:      end;
[LK2] 127:      clearviewport;
[UX4] 128:      ( Schirm löschen )
[000] 129:
[VZ5] 130:      ( Die einzelnen Farben nacheinander
[TN5] 131:      ( durch Schraffuren ersetzen und
[LB6] 132:      mit der Herculesgrafik darstellen )
[FP2] 133:      for c := 0 to 15 do
[K24] 134:      if colpatrn [c] <> 0 then
[5W5] 135:      ( Schraffur 0 = Farbe wird nicht )
[PD4] 136:      ( dargestellt )
[HU2] 137:      for y1 := 0 to 347 do
[1X3] 138:      for x1 := 0 to 719 do begin
[3H5] 139:      ( 348 Punkte in y- und 720 Punkte )
[PL5] 140:      ( in x-Richtung darstellen )
[3N2] 141:      y := y1 * 50 div 87;
[LG2] 142:      x := x1 * 8 div 9;
[4G5] 143:      ( Umrechnung auf die EGA-Punkte )
[XR4] 144:      ( Das Bild wird gestreckt )
[8Z2] 145:      if (((x1 and 1) = 0) and
[NN2] 146:      ((y1 and 1) = 0) and
[0V3] 147:      ((colpatrn [c] and 1) <> 0))
[MC0] 148:      or
[XE2] 149:      (((x1 and 1) = 1) and
[NN2] 150:      ((y1 and 1) = 0) and
[1V3] 151:      ((colpatrn [c] and 2) <> 0))
[MC0] 152:      or
[WE2] 153:      (((x1 and 1) = 0) and
[TN2] 154:      ((y1 and 1) = 1) and
[3V3] 155:      ((colpatrn [c] and 4) <> 0))
[MC0] 156:      or
[XE2] 157:      (((x1 and 1) = 1) and
[TN2] 158:      ((y1 and 1) = 1) and
[TX3] 159:      ((colpatrn [c] and 8) <> 0)))
[WL0] 160:      and
[AZ3] 161:      (ord (egamem [0] * y * 80 +
[HT1] 162:      x div 8) and
[6F3] 163:      (1 shl (7 - x mod 8)) <> 0) =
[MP2] 164:      (c shr 0) and $01) and
[EZ3] 165:      (ord (egamem [1] * y * 80 +
[HT1] 166:      x div 8) and
[6F3] 167:      (1 shl (7 - x mod 8)) <> 0) =
[NP2] 168:      (c shr 1) and $01) and
[KZ3] 169:      (ord (egamem [2] * y * 80 +
[HT1] 170:      x div 8) and
[6F3] 171:      (1 shl (7 - x mod 8)) <> 0) =
[PP2] 172:      (c shr 2) and $01) and
[PZ3] 173:      (ord (egamem [3] * y * 80 +
[HT1] 174:      x div 8) and
[6F3] 175:      (1 shl (7 - x mod 8)) <> 0) =
[E52] 176:      (c shr 3) and $01)
[3T3] 177:      then putpixel (x1, y1, 1);
[W45] 178:      ( Wenn die Bits in den Ebenen der
[XB4] 179:      EGA-Grafik der Farbe 'c' ent-
[TD5] 180:      sprechen und der Punkt auch von
[335] 181:      der Schraffur her gesetzt sein
[HP4] 182:      soll, dann setze den Punkt am
[FH5] 183:      Bildschirm. Das Programm sollte
[TY4] 184:      unbedingt mit der Einstellung
[4D3] 185:      'Options / Compiler /
[HH3] 186:      Boolean Evaluation ->
[GC2] 187:      Short Circuit'
[WF4] 188:      compiliert werden.
[ZN0] 189:      end;
[KW1] 190:      write(#7#7#7);
[D25] 191:      ( Piepton zeigt das Ende an )
[152] 192:      ch := readkey;
[0D4] 193:      if upcase (ch) <> 'N' then begin
[RH4] 194:      assign (herfile, 'HERC.PIX');
[D95] 195:      ( Abfrage, ob das Bild unter dem )
[0L5] 196:      ( Namen 'HERC.PIX' gespeichert )
[T64] 197:      ( werden soll )
[G23] 198:      rewrite (herfile);
[WP5] 199:      blockwrite (herfile, hergraph, 256);
[0A5] 200:      ( Speicherinhalt sichern )
[RD2] 201:      close (herfile)
[ZN0] 202:      end;
[TU1] 203:      closegraph
[DA4] 204:      ( Grafik beenden )
[KN0] 205:      end.

```

Listing 3. »ega2hgc.pas« konvertiert EGA-Bilddateien in Hercules-Bilddateien

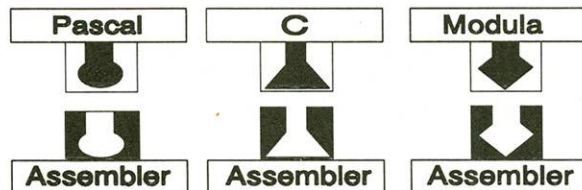
Programmieren mit System!
Sparen Sie bis zu 90% an Zeit und Aufwand!

H.A.I.-Tech

Hochsprachen-Assembler-Interface

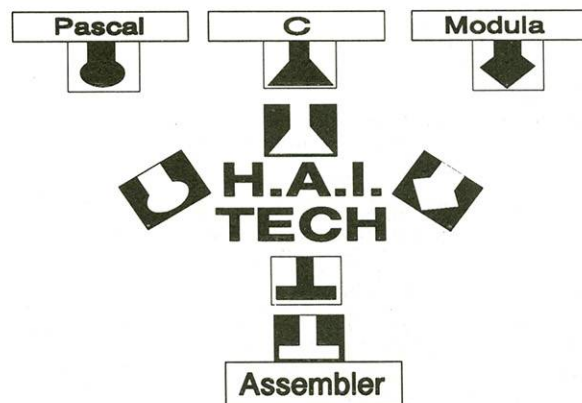
früher:

Für die gleiche Aufgabe mußte für jede Sprache, jeden Compiler eine neue Anbindung in Assembler geschrieben werden.



heute:

H.A.I.-Tech erledigt problemlos die Anbindung der verschiedenen Sprachen. Sie brauchen für jede Aufgabe nur noch eine Anbindung über die einheitliche und komfortable H.A.I.-Tech-Schnittstelle.



Folgende Compiler werden von H.A.I.-Tech direkt unterstützt:

Lahey Fortran F77L,	ab Version 2.1	Microsoft Pascal,	ab Version 3.2
Microsoft Fortran	4.0	Oregon Pascal-2	2.1
Ryan/McFarland Fortran	2.1	Turbo C	2.0
Prospero Fortran	1.2	Microsoft C	4.0
Turbo Pascal	4.0	JPI Modula-2	1.0

Anpassungen auf andere Compiler können über "externe Makros" selbst vorgenommen werden.

Als Beispiele zur Einarbeitung und als wertvoller Anfangsbestand werden Ihnen fertige Module von MS-DOS-Funktionen mitgeliefert: INT 21h (soweit sinnvoll) und die BIOS-Schnittstelle INT 10h bis 17h (außer 15h) für Bildschirm, Systemkonfiguration, Speicherbereiche, Diskettenzugriff, Festplatte, serielle Schnittstelle, Tastatur und Drucker.

Zum Erlernen der Makros steht ein ausführliches deutschsprachiges Handbuch von über 120 Seiten Umfang zur Verfügung.

Die Assemblerprogrammierung verliert mit H.A.I.-Tech einen Großteil ihrer üblichen Schrecken

Voraussetzungen: MASM oder Kompatible ab Version 4.0

Reviewt in toolbox 8'89

H.A.I.-Tech

Hochsprachen-Assembler-Interface

199,- DM (unverbindliche Preisempfehlung)

Wenn Sie über den DMV-Bestellservice bestellen, gilt folgendes:

Inland:		Ausland:	
Einzelpreis	199,- DM	Einzelpreis	199,- DM
zzgl. Versandkosten	4,- DM	zzgl. Versandkosten	6,- DM
Endpreis	203,- DM	Endpreis	205,- DM

DMV-Verlag · Postfach 250 · 3440 Eschwege



SCheck 2.02: Checksummer für sichere Listing-Eingabe

Die Programme in der DOS International und der Sonderheftreihe DOS Extra werden nach einem speziellen Verfahren mit Prüfsummen versehen und abgedruckt. Das für die Überprüfung notwendige Prüfsummenprogramm »SCheck« liegt jetzt auch als Version 2.02 für GW-Basic und SBasic vor.

Alle in der DOS International und DOS Extra abgedruckten Programme sind mit Prüfsummen und Zeilennummern versehen. Jede Programmzeile beginnt mit einer dreistelligen Prüfsumme. Sie ist in eckige Klammern gesetzt und wird beim Abtippen nicht mit eingegeben. Nach der Klammer folgt ein Leerzeichen, und darauf der eigentliche Text der Programmzeile (nur bei Basic-Programmen) oder eine Zeilennummer, die Sie ebenfalls nicht mit eingeben, und darauf erst der Programmtext.

Nur diesen letzten Teil der abgedruckten Zeile geben Sie beim Abtippen ein.

Bei allen Programmzeilen wird im Abdruck nach jeweils 55 Zeichen ein Zeilenumbruch vorgenommen. Die umgebrochene Teilzeile erhält dabei keine Prüfsumme. Bei der Eingabe geben Sie umgebrochene Zeilen in einem Stück ein, drücken also nicht die Enter-Taste, wenn ein Zeilenumbruch im Ausdruck steht, sondern nur, wenn eine neue Zeilennummer auftaucht.

Beim Abtippen von Programmen sind Tippfehler kaum zu vermeiden. Mit dem Checksummerprogramm »SCheck« überprüfen Sie die abgetippten Listings auf eventuelle Fehler. Das Programm erlaubt nicht nur die Diagnose von fehlerhaft eingegebenen Zeilen, sondern mittels eines integrierten Editors auch gleich die Korrektur der fehlerhaften Zeilen. SCHeck wird unter GW-Basic mit

run "scheck"

gestartet. Als erstes wählen Sie die gewünschte Prüfsummenversion. In der vorliegenden DOS-Ausgabe brauchen Sie immer »SCheck Version 2.0«. Nur wer Programme aus Ausgaben der DOS International vor Oktober 1988 eingeben will, muß den ebenfalls integrierten Prüfsummenalgorithmus »SCheck Version 1.x« aufrufen. Danach geben Sie den gewünschten Prüfmodus – normalerweise »Standard« – an und abschließend den Programmnamen des Listings, so wie Sie es auf dem Datenträger abgelegt haben.

Achtung, Basic-Listings müssen dabei im ASCII-Format vorliegen und dürfen nicht im Token-Format gespeichert sein. Geben Sie dazu im GW-Basic-Interpreter beim Speichern des Programms »name« den Befehl

SAVE"name",A

ein, um dieses Format zu erzwingen. Zuvor sollten Sie sich aber sicherheitshalber eine Kopie Ihres Listings anfertigen. SCHeck arbeitet zwar nicht mit dem Original, sondern mit einer temporären Zwischendatei und verändert die Originaldatei erst ganz zum Schluß, nach dem erfolgreichen Abschluß der Arbeit. Dennoch sollten Sie auch ein noch so geringes Risiko eines Datenverlusts, etwa durch Fehler beim Speichern, ausschalten. Wie Sie im Bild sehen, zeigt SCHeck die aktuelle Zeile stets in einem besonderen Bildschirmfenster an. Die Prüfsumme erscheint links unter dem

Fenster. Beim Überprüfen Ihres Listings vergleichen Sie die angezeigte Prüfsumme mit der in der DOS-Ausgabe in eckigen Klammern abgedruckten Prüfsumme. Rechts daneben finden Sie die aktuelle Zeilennummer und den Dateinamen. Im unteren Fenster werden Fehlermeldungen und Hinweise auf die weitere Bearbeitung ausgegeben. Im abgebildeten Fall können Sie mit [Enter] die Zeile übernehmen oder mit [E] den Editor aufrufen.

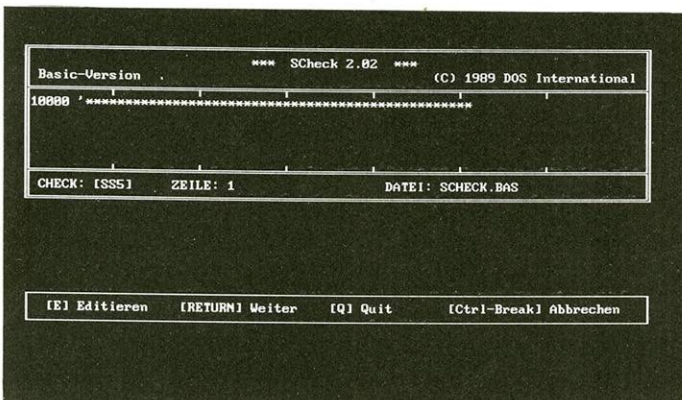
Der Editor dient der Korrektur fehlerhafter Zeilen. Er arbeitet immer im Einfügemodus, so daß fehlerhafte Eingaben extra gelöscht werden müssen. Sie können also nichts wesentlich überschreiben. Mit den Cursor-Tasten steuern Sie den Cursor im Editierfenster. [Home] bringt den Cursor an den Anfang der angezeigten Zeile, [End] entsprechend ans Ende.

Löschen können Sie das Zeichen links vom Cursor mit der großen Löschtaaste oberhalb der Enter-Taste, [Del] entfernt das Zeichen unter dem Cursor. Mit [Enter] übernehmen Sie die korrigierte Zeile. Steuerzeichen wie zum Beispiel Tabulatoren dürfen in einer Zeile nicht verwendet werden. Haben Sie so das gesamte Programm oder auch nur »verdächtige« Teile davon durchgearbeitet, so speichert ein letzter Druck auf die Enter-Taste das korrigierte Listing. SCHeck wird beendet, und das fehlerfreie Programmlisting steht zu Ihrer Verfügung.

Wenn Sie die Prüfsummen mit abtippen, kann SCHeck Ihr Listing automatisch überprüfen

SCHeck erlaubt aber noch einen zweiten Weg, das Listing korrekt einzutippen. Dabei geben Sie die Prüfsumme gleich bei der Eingabe ein. Die eckigen Klammern lassen Sie aber weg. Hinter der dreistelligen Prüfsumme fügen Sie ein Leerzeichen ein und danach den normalen Programmtext dieser Zeile. Wenn Sie den Text so aufbereitet vorliegen haben, können Sie die automatische Überprüfung mit SCHeck nutzen. Rufen Sie dazu die Option »Mit Prüfsumme« auf.

SCHeck verwandelt das Listing nun in eine Quelldatei für den jeweiligen Compiler oder Interpreter, indem die drei Prüfsummenstellen und das Leerzeichen vom Rest der Zeile abgetrennt werden. Aus der so entstandenen Programmzeile berechnet SCHeck die Prüfsumme und ver-



Der Arbeitsbildschirm von SCHeck 2.02 in Aktion

gleicht sie mit der zuvor abgetrennten Summe, die Sie eingegeben haben. Falls ein Unterschied auftritt, haben Sie Gelegenheit, die fehlerhafte Zeile manuell zu korrigieren. Nach einem solchen Durchlauf von SCHECK steht wieder das fehlerfreie Programm, diesmal allerdings ohne Prüfsumme, auf der Diskette. Auch hier gilt, daß Sie nicht mit dem Original arbeiten sollten, sondern nur mit einer Kopie, um im Zweifelsfall nochmal von vorne beginnen zu können.

SCHECK verfügt aber noch über eine weitere Option. Die von Fall zu Fall abgedruckten hexadezimalen Listings können Sie nämlich mit SCHECK im Binärmodus direkt in lauffähige Programme umwandeln. Dazu müssen Sie das Hexdump-Listing mit einem ganz normalen Editor (zum Beispiel Turbo Pascal, Wordstar oder Edlin) als Textdatei

eingeben. Anschließend überprüfen Sie mit SCHECK zunächst die Korrektheit der abgetippten Daten wie bei jedem anderen Listing. Befindet sich das Programm korrekt auf Diskette, so wird die Option »Hexdump« aufgerufen. SCHECK wird nun dazu veranlaßt, eine Hexdump-Textdatei in ein ablauffähiges Programm umzuwandeln. SCHECK erzeugt in jedem Fall eine Datei mit der Namensweiterung »bin«. Um daraus eine lauffähige Datei zu machen, müssen Sie sie mit dem Ren-Befehl von MS-DOS umbenennen:

```
ren <name>.bin <name>.com
```

Falls eine Exe-Datei erzeugt werden soll (bitte entnehmen Sie dies der jeweiligen Programmbeschreibung) lautet der entsprechende Befehl:

```
ren <name>.bin <name>.exe
```

Sie sehen, mit SCHECK 2.02 ist das korrekte Abtippen von Programmlis-

tings kein bißchen komplizierter geworden – aber durch einen verbesserten Prüfsummen-Algorithmus um einiges sicherer.

(ev/ma)

SCHECK mit SBasic compilieren

Um die Eingabehilfe SCHECK mit dem SBasic-Compiler compilieren zu können, müssen Sie lediglich aus Programmzeile 10070 die Zeichenfolge »key off:« löschen und die Zeile 10720 sowie 11240 vollständig entfernen. Schließlich müssen Sie noch die Programmzeile 11170 in

```
11170 GOTO 10750
```

umändern und die Zeile

```
10735 IF EOF(1) THEN 11160
```

neu in das Programm einfügen.

Name	SCHECK.bas
Sprache	GW-Basic und SBasic
Grafikkarte	jede
Besonderheiten	Checksummer für Listings in DOS International und DOS Extra

```
[SS5] 10000 *****
[F29] 10010 ' * SCHECK - Prüfsummengenerator für DOS-Lis
[ML4] 10020 ' *
[807] 10030 ' * Konzeption: Volker Everts
[AC7] 10040 ' * Basic-Version: Markus Zietlow
[XS5] 10050 *****
[4U0] 10060 '
[FL7] 10070 KEY OFF: DEFINT A-O,Q-Z: DEFSNG P: COLOR 7,
[5G8] 10080 T = 1: TEMP = 2: F = 3'Ein-/Ausgabekanäle f
[332] 10090 FALSE = 0
[902] 10100 TRUE = -1
[EY7] 10110 ZEICHEN$ = "0123456789ABCDEFGHIJKLMNPRSTUVWXY
[9H2] 10120 LAUF = FALSE
[6B2] 10130 BIN = FALSE
[851] 10140 CLS
[9S1] 10150 PRINT
[K07] 10160 PRINT CHR$(201); STRING$(78, 205); CHR$(187
);
[Z5C] 10170 PRINT CHR$(186); STRING$(28, 32); "**** Sch
eck 2.02 ***"; STRING$(29, 32); CHR$(186);
[0KE] 10180 PRINT CHR$(186); " Basic-Version
(C) 1989 DOS Internatio
nal "; CHR$(186);
[C5C] 10190 PRINT CHR$(204); "
"; CHR$(185);
[DR3] 10200 FOR ZAEHLER = 1 TO 5
[Zy6] 10210 PRINT CHR$(186); STRING$(78, 32); CHR$(18
6);
[CR2] 10220 NEXT ZAEHLER
[84C] 10230 PRINT CHR$(204); "
"; CHR$(185);
[00D] 10240 PRINT CHR$(186); " CHECK: ZEILE:
DATEI:
"; CHR$(186);
[P07] 10250 PRINT CHR$(200); STRING$(78, 205); CHR$(188
);
[2D2] 10260 LOCATE 20, 1
[B17] 10270 PRINT CHR$(218); STRING$(78, 196); CHR$(191
);
```

```
[007] 10280 PRINT CHR$(179); STRING$(78, 32); CHR$(179)
;
[C27] 10290 PRINT CHR$(192); STRING$(78, 196); CHR$(217
);
[4V1] 10300 COLOR 7
[9F1] 10310 Y = 7
[9V6] 10320 LOCATE 7, 31: PRINT "SCHECK Version 1.x"
[PR6] 10330 LOCATE 8, 31: PRINT "SCHECK Version 2.0"
[M7H] 10340 LOCATE 21, 4: PRINT "Bitte wählen Sie mit d
en Pfeiltasten aus und bestätigen Sie durch [Ente
r] ";
[2PD] 10350 COLOR 14, 1: FOR Q = 30 TO 50: LOCATE Y, Q:
PRINT CHR$(SCREEN(Y, Q)): NEXT Q: COLOR 7, 0
[UP6] 10360 TASTE$ = INKEY$: IF TASTE$ = "" THEN 10360
[P96] 10370 IF TASTE$ <> CHR$(0) + CHR$(72) THEN 10400
[4ND] 10380 IF Y <> 7 THEN FOR Q = 30 TO 50: LOCATE Y,
Q: PRINT CHR$(SCREEN(Y, Q)): NEXT Q: Y = Y - 1
[532] 10390 GOTO 10350
[1A6] 10400 IF TASTE$ <> CHR$(0) + CHR$(80) THEN 10430
[1ND] 10410 IF Y <> 8 THEN FOR Q = 30 TO 50: LOCATE Y,
Q: PRINT CHR$(SCREEN(Y, Q)): NEXT Q: Y = Y + 1
[232] 10420 GOTO 10350
[F35] 10430 IF TASTE$ <> CHR$(13) THEN 10360
[8N2] 10440 FOR Q = 7 TO 8
[U16] 10450 LOCATE Q, 30: PRINT STRING$(21, 32)
[KN1] 10460 NEXT Q
[8Y2] 10470 VERSION = Y - 6
[GV1] 10480 COLOR 7
[NF1] 10490 Y = 7
[MA5] 10500 LOCATE 7, 36: PRINT "Standard"
[4A6] 10510 LOCATE 8, 34: PRINT "Mit Prüfsumme"
[C65] 10520 LOCATE 9, 37: PRINT "Hexdump"
[V7H] 10530 LOCATE 21, 4: PRINT "Bitte wählen Sie mit d
en Pfeiltasten aus und bestätigen Sie durch [Ente
r] ";
[9PD] 10540 COLOR 14, 1: FOR Q = 30 TO 50: LOCATE Y, Q:
PRINT CHR$(SCREEN(Y, Q)): NEXT Q: COLOR 7, 0
[3R6] 10550 TASTE$ = INKEY$: IF TASTE$ = "" THEN 10550
[4B6] 10560 IF TASTE$ <> CHR$(0) + CHR$(72) THEN 10590
[BND] 10570 IF Y <> 7 THEN FOR Q = 30 TO 50: LOCATE Y,
Q: PRINT CHR$(SCREEN(Y, Q)): NEXT Q: Y = Y - 1
[E32] 10580 GOTO 10540
[GA6] 10590 IF TASTE$ <> CHR$(0) + CHR$(80) THEN 10620
[CND] 10600 IF Y <> 9 THEN FOR Q = 30 TO 50: LOCATE Y,
Q: PRINT CHR$(SCREEN(Y, Q)): NEXT Q: Y = Y + 1
[B32] 10610 GOTO 10540
[P35] 10620 IF TASTE$ <> CHR$(13) THEN 10550
[GN2] 10630 FOR Q = 7 TO 9
[126] 10640 LOCATE Q, 30: PRINT STRING$(21, 32)
[TN1] 10650 NEXT Q
[WC4] 10660 IF Y = 8 THEN LAUF = TRUE
[NA4] 10670 IF Y = 9 THEN BIN = TRUE
[RU5] 10680 LOCATE 12, 54: PRINT STRING$(25, 32)
[8DF] 10690 GOSUB 13150: LOCATE 21, 12: PRINT "Bitte ge
ben Sie den Dateinamen der gewünschten Datei ein
";
[9P8] 10700 Y = 12: X = 54: LAENGE = 24: MENGE$ = "alle
": GOSUB 10890
[N6D] 10710 IF (TASTE$ = CHR$(27)) OR (Q$ = "") THEN LO
CATE 12, 54: PRINT STRING$(25, 32): GOTO 10680
[LM3] 10720 ON ERROR GOTO 11160
```

Listing. »SCHECK.bas« für GW-Basic


```
[C74] 10730 OPEN QS FOR INPUT AS #T
[732] 10740 GOTO 11230
[302] 10750 COLOR 12
[2A6] 10760 LOCATE 8, 30: PRINT "D A T E I F E H L E R"
[VV1] 10770 COLOR 7
[5GC] 10780 GOSUB 13150: LOCATE 21, 21: PRINT "Drücken
Sie bitte eine beliebige Taste ";
[B83] 10790 WHILE INKEY$ = ""
[GX2] 10800 PRINT CHR$(7);
[CE1] 10810 WEND
[6U5] 10820 LOCATE 8, 30: PRINT STRING$(21, 32)
[842] 10830 GOTO 10680
[2V0] 10840 '
[184] 10850 '*****
[UD6] 10860 '* Unterprogramm zur Dateneingabe *
[384] 10870 '*****
[6V0] 10880 '
[GG1] 10890 QS = ""
[EF2] 10900 LOCATE Y, X
[FS3] 10910 PRINT QS; CHR$(219);
[GR6] 10920 TASTE$ = INKEY$: IF TASTE$ = "" THEN 10920
[2K5] 10930 IF (TASTE$ <> CHR$(8)) THEN 10990
[224] 10940 IF LEN(QS) = 0 THEN 10920
[X44] 10950 QS = LEFT$(QS, LEN(QS) - 1)
[6N2] 10960 LOCATE Y, X
[V44] 10970 PRINT QS; CHR$(219); " ";
[8A2] 10980 GOTO 10920
[T98] 10990 IF TASTE$ = CHR$(13) THEN LOCATE Y, X: PRIN
T QS; " ";
[639] 11000 IF (TASTE$ = CHR$(13)) OR (TASTE$ = CHR$(27
)) THEN RETURN
[C55] 11010 IF (ASC(TASTE$) < 32) THEN 10920
[NX4] 11020 IF MENGE$ <> "zahl" THEN 11050
[ZC8] 11030 IF (ASC(TASTE$) < 48) OR (ASC(TASTE$) > 5
7) THEN 10920
[892] 11040 GOTO 11070
[PU4] 11050 IF MENGE$ = "alle" THEN 11070
[E86] 11060 IF INSTR(MENGE$, TASTE$) = 0 THEN 10920
[YF6] 11070 IF LEN(QS) < LAENGE THEN QS = QS + TASTE$
[T22] 11080 GOTO 10900
[6H8] 11090 Y = 10: X = 65: LAENGE = 5: MENGE$ = "zahl"
: GOSUB 10890
[BV5] 11100 IF (TASTE$ = CHR$(27)) THEN RETURN
[6U0] 11110 '
[NK2] 11120 '*****
[TR3] 11130 '* Fehler-Routine *
[RK2] 11140 '*****
[AU0] 11150 '
[KN1] 11160 CLOSE
[ZG2] 11170 RESUME 10750
[DU0] 11180 '
[8G2] 11190 '*****
[TH3] 11200 '* Hauptprogramm *
[4G2] 11210 '*****
[BU0] 11220 '
[KP3] 11230 COLOR 7: GOSUB 13150
[N43] 11240 ON ERROR GOTO 0
[K62] 11250 DATEI$ = QS
[DF7] 11260 GROSS$ = DATEI$: GOSUB 13040: DATEI$ = GROS
S$
[3W3] 11270 I = INSTR(DATEI$, ".")
[6K7] 11280 IF I = 0 THEN DATEI$ = DATEI$ + ".": I = LE
N(DATEI$)
[U64] 11290 EXT$ = MID$(DATEI$, I, 4)
[8T4] 11300 DATEI$ = LEFT$(DATEI$, I - 1)
[HN5] 11310 LOCATE 12, 54: PRINT DATEI$: EXT$
[HB4] 11320 TEMPNAME$ = DATEI$ + ".$$$"
[SE6] 11330 OPEN TEMPNAME$ FOR OUTPUT AS #TEMP
[CE7] 11340 IF BIN THEN OPEN DATEI$ + ".BIN" FOR OUTPUT
AS #F
[CP7] 11350 IF VERSION = 2 THEN NUMBER = 1 ELSE NUMBER
= 0
[KYC] 11360 IF BIN THEN GOSUB 13150: LOCATE 21, 25: PRI
NT ">>> Binärdatei wird erzeugt <<<"
[TE3] 11370 WHILE NOT EOF(T)
[X24] 11380 LINE INPUT #T, ZEILE$
[LDB] 11385 WHILE RIGHT$(ZEILE$, 1) = " ": ZEILE$ = LEFT$
(ZEILE$, LEN(ZEILE$) - 1) : WEND
[333] 11390 WEITER = FALSE
[WT3] 11400 IF WEITER THEN 11500
[WN6] 11410 GOSUB 11820 'Gelesene Zeile anzeigen
[TS4] 11420 IF BIN THEN WEITER = TRUE
[ZT3] 11430 IF WEITER THEN 11500
[AM6] 11440 CH$ = INKEY$: IF CH$ = "" THEN GOTO 11440
[3G3] 11450 ERLAUBT = FALSE
[ZH8] 11460 IF CH$ = CHR$(13) THEN ERLAUBT = TRUE: WE
ITER = TRUE
[N1C] 11470 IF (CH$ = "e") OR (CH$ = "E") THEN ERLAUB
T = TRUE: GOSUB 12060 'Zeile bearbeiten
[8ME] 11480 IF (CH$ = "q") OR (CH$ = "Q") THEN IF NOT
LAUF THEN GOSUB 12420: IF SCHLUSS THEN GOTO 1163
0 'Quit
[F57] 11490 IF ERLAUBT THEN GOTO 11400 ELSE GOTO 1144
0
[8Y3] 11500 PRINT #TEMP, ZEILE$
[DU3] 11510 IF NOT BIN THEN 11610
[D99] 11520 WHILE LEFT$(ZEILE$, 1) = " ": ZEILE$ = MI
D$(ZEILE$, 2) : WEND
```

```
[364] 11530 IF ZEILE$ = "" THEN 11610
[K75] 11540 B = VAL("&H" + LEFT$(ZEILE$, 2))
[TSH] 11550 IF (B < 0) OR (B > 255) THEN GOSUB 13150:
LOCATE 21, 22: PRINT "+++++ ABBRUCH WEGEN DATENF
EHLER +++++": LOCATE 25, 1: END
[HN1] 11560 BY = B
[TK3] 11570 PRINT #F, CHR$(BY);
[YA4] 11580 ZEILE$ = MID$(ZEILE$, 3)
[M99] 11590 WHILE LEFT$(ZEILE$, 1) = " ": ZEILE$ = MI
D$(ZEILE$, 2) : WEND
[T92] 11600 GOTO 11530
[DY3] 11610 NUMBER = NUMBER + 1
[8E1] 11620 WEND
[S7B] 11630 WHILE NOT EOF(T): LINE INPUT #T, ZEILE$: PR
INT #TEMP, ZEILE$: WEND
[4P1] 11640 CLOSE
[8U5] 11650 OPEN DATEI$ + EXT$ FOR INPUT AS #1
[X06] 11660 OPEN DATEI$ + ".BAK" FOR OUTPUT AS #2
[XGB] 11670 WHILE NOT EOF(1): LINE INPUT #1, ZEILE$: PR
INT #2, ZEILE$: WEND: CLOSE
[6D5] 11680 OPEN TEMPNAME$ FOR INPUT AS #1
[576] 11690 OPEN DATEI$ + EXT$ FOR OUTPUT AS #2
[UGB] 11700 WHILE NOT EOF(1): LINE INPUT #1, ZEILE$: PR
INT #2, ZEILE$: WEND: CLOSE
[713] 11710 KILL TEMPNAME$
[GTN] 11720 GOSUB 12560: LOCATE 12, 9: PRINT STRING$(11
, 32): LOCATE 12, 25: PRINT STRING$(21, 32): LOCA
TE 12, 53: PRINT STRING$(27, 32) 'Statuszeile lösc
hen
[494] 11730 IF BIN THEN EXT$ = ".BIN"
[N4H] 11740 GOSUB 13150: LOCATE 21, (80 - (31 + LEN(DAT
EI$ + EXT$))) / 2: PRINT "Die Datei "; DATEI$ +
EXT$; " wurde gespeichert."
[CE2] 11750 LOCATE 25, 1
[H51] 11760 END
[4V0] 11770 '
[TH3] 11780 '*****
[B65] 11790 '* Gelesene Zeile anzeigen *
[NH3] 11800 '*****
[2V0] 11810 '
[BBB] 11820 IF LAUF THEN CHECK$ = "[" + LEFT$(ZEILE$, 3
) + "": ZEILE$ = MID$(ZEILE$, 5)
[MDN] 11830 FOR Q = 1 TO LEN(CHECK$): IF ASC(MID$(CHECK
$, Q, 1)) > 96 AND ASC(MID$(CHECK$, Q, 1)) < 123
THEN MID$(CHECK$, Q, 1) = CHR$(ASC(MID$(CHECK$, Q
, 1)) - 32)
[3P1] 11840 NEXT Q
[B57] 11850 LINKS = 2: OBEN = 6: RECHTS = 79: UNTEN = 1
0
[CX4] 11860 LOCATE 12, 26: PRINT NUMBER
[Z45] 11870 GOSUB 12560 'Fenster löschen
[WX8] 11880 GOSUB 12660 'Zeile innerhalb des Fensters an
zeigen
[366] 11890 GOSUB 12800 'Check-Summe errechnen
[0M2] 11900 LOCATE 12, 10
[HB2] 11910 PRINT PS$
[HM7] 11920 IF BIN THEN GOTO 12000 'Binärdatei erzeugen
[N8C] 11930 IF LAUF THEN GOTO 11970 'Übereinstimmung der
Prüfsummen feststellen
[X74] 11940 GOSUB 13150: LOCATE 21, 4
[LKD] 11950 PRINT "[E] Editieren [RETURN] Weiter
[Q] Quit [Ctrl-Break] Abbrechen"
[152] 11960 GOTO 11990
[ATP] 11970 GOSUB 13150: IF PS$ = CHECK$ THEN LOCATE 21
, (80 - (22 + LEN(STR$(NUMBER)))) / 2: PRINT "Prü
fsumme OK für Zeile"; NUMBER: WEITER = TRUE: GOTO
12000
[1NH] 11980 LOCATE 21, 4: PRINT CHR$(7); "Prüfsummenfeh
ler! [E] Editieren [RETURN] Weiter [Ctrl-Break
] Abbrechen"
[XM2] 11990 LOCATE 12, 15
[KV1] 12000 RETURN
[5U0] 12010 '
[MK2] 12020 '*****
[4H3] 12030 '* Zeile eingeben *
[PK2] 12040 '*****
[9U0] 12050 '
[HRD] 12060 GOSUB 13150: LOCATE 21, 17: PRINT "Ändern S
ie die Programmzeile [Enter] Ende"
[Z45] 12070 GOSUB 12560 'Fenster löschen
[7B5] 12080 GOSUB 12660 'Zeile darstellen
[FL3] 12090 LOCATE OBEN, LINKS
[7G3] 12100 X = 1: PRINT " ";
[8E2] 12110 CH$ = INKEY$
[FE6] 12120 IF LEFT$(CH$, 1) <> CHR$(0) THEN GOTO 12220
[4N3] 12130 CH$ = RIGHT$(CH$, 1)
[4NE] 12140 IF CH$ = "M" THEN IF X > LEN(ZEILE$) THEN
PRINT CHR$(7); : GOTO 12110 ELSE GOSUB 12340: X
= X + 1
[KZC] 12150 IF CH$ = "K" THEN IF X = 1 THEN PRINT CHR
$(7); : GOTO 12110 ELSE GOSUB 12340: X = X - 1
[YZC] 12160 IF CH$ = "H" THEN IF X < 79 THEN PRINT CH
R$(7); : GOTO 12110 ELSE GOSUB 12340: X = X - 78
[YCF] 12170 IF CH$ = "P" THEN IF X + 77 > LEN(ZEILE$)
THEN PRINT CHR$(7); : GOTO 12110 ELSE GOSUB 1234
0: X = X + 78
[TV7] 12180 IF CH$ = "G" THEN IF X <> 1 THEN GOSUB 12
340: X = 1
```

Listing. »Scheck.bas« für GW-Basic


```

[41C] 12190 IF CH$ = "O" THEN IF X <> LEN(ZEILE$) + 1
      THEN GOSUB 12340: X = LEN(ZEILE$) + 1
[8PM] 12200 IF CH$ = "S" THEN IF X <= LEN(ZEILE$) THEN
      N ZEILE$ = LEFT$(ZEILE$, X - 1) + MID$(ZEILE$, X
      + 1): GOSUB 12660 ELSE PRINT CHR$(7); : GOTO 1211
      0
[G92] 12210 GOTO 12240
[XXN] 12220 IF CH$ = CHR$(8) THEN IF X > 1 THEN ZEILE$
      = LEFT$(ZEILE$, X - 2) + MID$(ZEILE$, X): GOSUB 1
      2660: GOSUB 12340: X = X - 1 ELSE PRINT CHR$(7);
      : GOTO 12110
[LZN] 12230 IF CH$ >= " " THEN IF LEN(ZEILE$) = 255 THEN
      N PRINT CHR$(7); : GOTO 12110 ELSE ZEILE$ = LEFT$
      (ZEILE$, X - 1) + CH$ + MID$(ZEILE$, X): GOSUB 12
      660: X = X + 1
[7YB] 12240 LOCATE OBEN + INT((X - 1) / 78), LINKS - 1
      + X - INT((X - 1) / 78) * 78: PRINT "█":
[LK5] 12250 IF CH$ <> CHR$(13) THEN GOTO 12110
[648] 12260 IF LAUF THEN ZEILE$ = MID$(CHECK$, 2, 3) +
      " " + ZEILE$
[XL6] 12270 GOSUB 12340: GOSUB 12660 'Zeile anzeigen
[2W1] 12280 RETURN
[NU0] 12290 '
[CP3] 12300 '*****
[2H5] 12310 '* Zeichen wiederherstellen *
[EP3] 12320 '*****
[LU0] 12330 '
[UM9] 12340 LOCATE OBEN + INT((X - 1) / 78), LINKS - 1
      + X - INT((X - 1) / 78) * 78
[YXA] 12350 IF MID$(ZEILE$, X, 1) = " " THEN PRINT " " E
      LSE PRINT MID$(ZEILE$, X, 1)
[4W1] 12360 RETURN
[RU0] 12370 '
[0A2] 12380 '*****
[3X2] 12390 '* Quit (Ende) *
[W92] 12400 '*****
[NU0] 12410 '
[U63] 12420 SCHLUSS = FALSE
[KYB] 12430 SICHER$ = "": FOR Q = 2 TO 79: SICHER$ = SI
      CHER$ + CHR$(SCREEN(21, Q)): NEXT Q
[YDD] 12440 GOSUB 13150: LOCATE 21, 16: PRINT "Bitte dr
      ücken Sie [J], um die Arbeit zu beenden "
[GA6] 12450 CH$ = INKEY$: IF CH$ = " " THEN GOTO 12450
[R3B] 12460 IF (CH$ = "j") OR (CH$ = "J") THEN SCHLUSS
      = TRUE: PRINT #TEMP, ZEILE$
[BA2] 12470 GOSUB 13150
[P78] 12480 IF NOT SCHLUSS THEN LOCATE 21, 2: PRINT SIC
      HERS
[BW1] 12490 RETURN
[651] 12500 END
[TU0] 12510 '
[6R2] 12520 '*****
[U24] 12530 '* Fenster löschen *
[8R2] 12540 '*****
[XU0] 12550 '
[044] 12560 FOR Q = OBEN TO UNTEN
[253] 12570 LOCATE Q, LINKS
[8B6] 12580 PRINT STRING$(RECHTS - LINKS + 1, 32)
[2N1] 12590 NEXT Q
[AW1] 12600 RETURN
[XU0] 12610 '
[265] 12620 '*****
[D98] 12630 '* Inhalt von »zeile$« im Fenster anzeigen
      *
[465] 12640 '*****
[1V0] 12650 '

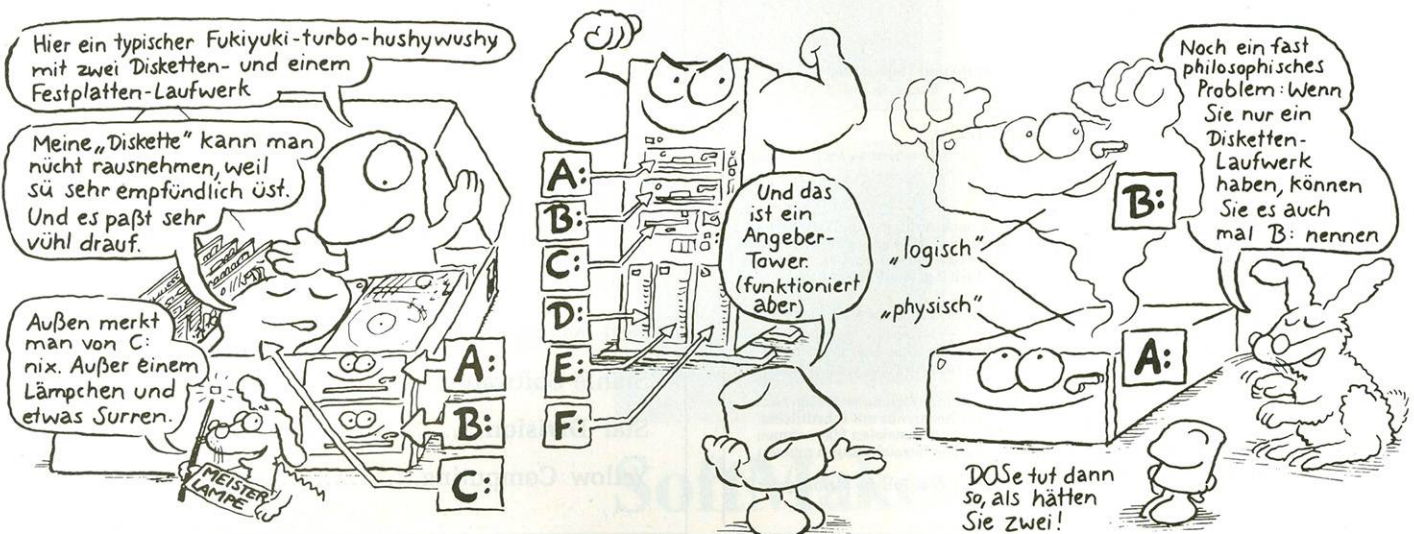
```

```

[SR2] 12660 F.X = 0: F.Y = 0
[CC5] 12670 FOR ZAEHLER = 1 TO LEN(ZEILE$)
[F85] 12680 LOCATE OBEN + F.Y, LINKS + F.X
[UH5] 12690 PRINT MID$(ZEILE$, ZAEHLER, 1);
[SK2] 12700 F.X = F.X + 1
[597] 12710 IF F.X >= RECHTS - 1 THEN F.X = 0: F.Y =
      F.Y + 1
[6S2] 12720 NEXT ZAEHLER
[RX6] 12730 LOCATE OBEN + F.Y, LINKS + F.X: PRINT " ";
[KW1] 12740 RETURN
[5V0] 12750 '
[AH4] 12760 '*****
[ET6] 12770 '* Zeile auf Check-Summe überprüfen *
[CH4] 12780 '*****
[9V0] 12790 '
[F42] 12800 P = 0: A = 1
[8SB] 12810 WHILE (MID$(ZEILE$, A, 1) = " ") OR (MID$(Z
      EILE$, A, 1) = CHR$(9)): A = A + 1: WEND
[374] 12820 FOR I = A TO LEN(ZEILE$)
[4CX] 12830 IF MID$(ZEILE$, I, 1) >= " " THEN GROSS$
      = MID$(ZEILE$, I, 1): IF VERSION = 1 THEN P = ((I
      - A + 1) AND 7) * ASC(GROSS$) + P: GOTO 12920 EL
      SE GOSUB 13040: P = (((I - A + 1) AND 3) + 1) * A
      SC(GROSS$) + P: GOTO 12920
[S22] 12840 COLOR 12
[5N6] 12850 LOCATE 8, 30: PRINT "D A T E I F E H L E
      R"
[022] 12860 COLOR 7
[X7C] 12870 LOCATE 21, 11: PRINT " Drücken S
      ie bitte eine beliebige Taste ";
[CC3] 12880 WHILE INKEY$ = " "
[D93] 12890 PRINT CHR$(7);
[5L1] 12900 WEND
[R16] 12910 LOCATE 8, 30: PRINT STRING$(21, 32)
[0P1] 12920 NEXT I
[KF2] 12930 PSS$ = "[ ]"
[AM2] 12940 FOR I = 2 TO 4
[K25] 12945 IF P > 32767 THEN P = P - 32768
[5T7] 12950 MID$(PSS$, I, 1) = MID$(ZEICHEN$, (P AND 3
      1) + 1, 1)
[H22] 12960 P = FIX(P / 32)
[5P1] 12970 NEXT I
[YW1] 12980 RETURN
[HV0] 12990 '
[ZX3] 13000 '*****
[K06] 13010 '* In Großbuchstaben umwandeln *
[934] 13020 '*****
[AU0] 13030 '
[92N] 13040 FOR Q = 1 TO LEN(GROSS$): IF ASC(MID$(GROSS
      $, Q, 1)) > 96 AND ASC(MID$(GROSS$, Q, 1)) < 123
      THEN MID$(GROSS$, Q, 1) = CHR$(ASC(MID$(GROSS$, Q
      , 1)) - 32)
[8T8] 13050 IF MID$(GROSS$, Q, 1) = "ä" THEN MID$(GROSS
      $, Q, 1) = "Ä"
[NU8] 13060 IF MID$(GROSS$, Q, 1) = "ö" THEN MID$(GROSS
      $, Q, 1) = "Ö"
[UU8] 13070 IF MID$(GROSS$, Q, 1) = "ü" THEN MID$(GROSS
      $, Q, 1) = "Ü"
[DN1] 13080 NEXT Q
[YV1] 13090 RETURN
[BU0] 13100 '
[Y22] 13110 '*****
[9N4] 13120 '* Meldezeile löschen *
[003] 13130 '*****
[FU0] 13140 '
[ZT5] 13150 LOCATE 21, 2: PRINT STRING$(78, 32)
[ZV1] 13160 RETURN

```

Listing. »SCheck.bas« für GW-Basic



Impressum

Herausgeber

Christian Widuch

Leitung des Redaktionsbüros

Volker Everts (ev), Andreas Hagedorn (hg)

Chefredakteur

Martin Aschoff (ma)

Stellvertretende Chefredakteurin

Eva-Maria Hierlmeier (hi)

Redaktion

Alois Kneisle (al), Manfred Kohlen (mk), Werner Niedermaier (wn), Thomas Röder (tr), Andrea Scheding (as), Herwig Weihe (hw), Stefan Willmeroth (sw), Johann Zieger (zi)

Schlußredaktion

Barbara Mooser, Karl-Heinz Peter, Markus Stein, Andrea Voß

Redaktionsassistentin

Claudia Beer, Dorothea Lehner

Koordination

Stefan Ritter

Produktionsleitung

Gerd Köberich

Bereichsleitung

Christian Heckmann (Fotografie/Layout)

Claudia Küllmer (Fotosatz/Lektorat)

Margarete Schenk, Helmut Skoupy (Montage/Reprografie)

Layout

Yvonne Hendricks

Fotografie

Klaus Jatho

Fotosatz

Silvia Führer

Lektorat

Dagmar Wilhelm-Ballhaus

Montage/Reprografie

Manuela Eska, Andrea Gundlach

Werbegestaltung

Mohamed Hawa, Petra Küch

Anzeigenverkaufsleitung

Wolfgang Schnell

Anzeigenverkauf für PLZ 1,4,5

Gerlinde Rachow (0 56 51) 80 09 - 53

Wolfgang Brill (0 56 51) 80 09 - 51

Anzeigenverkauf für PLZ 2 + 3

DMV-Verlagsbüro Hamburg

Ohlsdorfer Straße 34, 2000 Hamburg 60

Sylvia Ehrenpfordt, Tel. (0 40) 46 12 33, Telefax: (0 40) 47 43 10

Anzeigenverkauf für PLZ 6 - 8

DMV-Verlagsbüro München

Zaunkönigsweg 2c, 8000 München 82,

Telefon: (0 89) 4 39 10 87, Telefax: (0 89) 4 39 10 80

Leitung: Britta Fiebig

Anzeigenverkauf: Monika Schöbel, Jens Dhein, Peter Schätzle

Anzeigenverwaltung und Disposition

Andrea Giese, Karina Ehrlich, Beate Kranz

Anschrift Redaktion:

Redaktionsbüro Everts & Hagedorn

Redaktion DOS International,

Hans-Stießberger-Str. 3, 8013 Haar/München

Telefon: (0 89) 46 10 01-0

Telefax: (0 89) 46 10 01-90

Anschrift Verlag:

DMV Daten und Medien-Verlag

Widuch GmbH & Co. KG

Fuldaer Straße 6, 3440 Eschwege

Telefon: (0 56 51) 80 09-0

Telefax: (0 56 51) 80 09-33

Vertrieb

Verlagsunion Erich Pabel-Arthur Moewig KG (VPM)

Friedrich-Bergius-Straße 20

6200 Wiesbaden

Druck

Druckerei Jungfer, 3420 Herzberg

Bezugspreise

DOS Extra erscheint viermal jährlich.

Einzelpreis DM 18,-, 145 6S, 18,- sfr.

Abonnementpreise

Die Preise verstehen sich grundsätzlich einschließlich Porto und Verpackung.

Inland: Bezugspreis 4 Ausgaben: DM 60,-

Europäisches Ausland: Bezugspreis 4 Ausgaben: DM 70,-

Außereuropäisches Ausland: Bezugspreis 4 Ausgaben: DM 80,-

Bankverbindungen:

Postcheck Frankfurt/M: Kto.-Nr.: 23043-608

Raiffeisenbank Eschwege:

BLZ: 522 603 85, Kto.-Nr.: 245 7008

Die Abonnementbestellung kann innerhalb einer Woche nach Auftrag beim DMV-Verlag, Postfach 250, 3440 Eschwege, schriftlich widerrufen werden. Zur Wahrung der Frist reicht der Poststempel. Das Abonnement verlängert sich automatisch um 4 Ausgaben, wenn es nicht mindestens 6 Wochen vor Ablauf beim Verlag schriftlich gekündigt wird.

Manuskripte, Programme:

Bitte an das Redaktionsbüro Everts & Hagedorn, Redaktion DOS International, Hans-Stießberger-Straße 3, 8013 Haar/München, senden.

Für unverlangt eingesandte Manuskripte und Datenträger sowie Fotos übernehmen der Verlag und die Redaktion keine Haftung. Die Zustimmung zum Abdruck wird vorausgesetzt.

Das Urheberrecht für veröffentlichte Manuskripte liegt ausschließlich beim Verlag. Nachdruck sowie Vervielfältigung oder sonstige Verwertung von Texten nur mit schriftlicher Genehmigung des Verlages. Für veröffentlichte Programme und Basteileien übernehmen wir weder Gewähr noch Haftung. Schaltungen und verwendete Bezeichnungen müssen nicht frei von gewerblichen Schutzrechten sein.

Namentlich gekennzeichnete Fremdbeiträge geben nicht in jedem Fall die Meinung der Redaktion wieder.

Unsere Telefonnummer hat sich geändert!

0 56 51/8 09-0

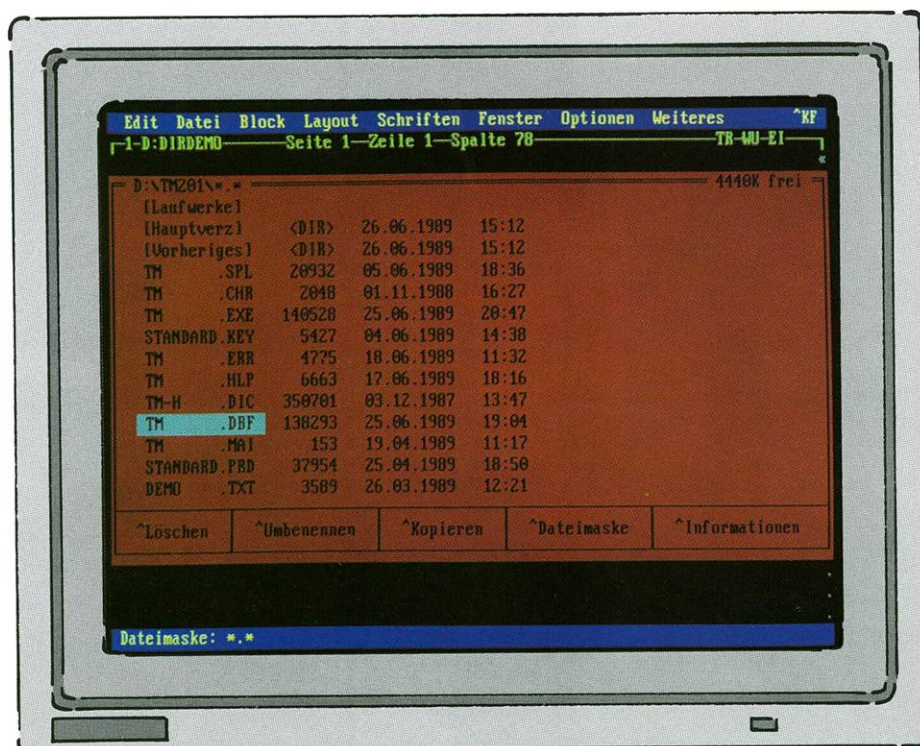
Ab 1. Januar 1990
erreichen Sie den DMV-Verlag
während der üblichen
Bürozeiten unter der oben
genannten Rufnummer.

Die Inserenten

Abacomp.....	9
Bath Beach Software.....	105
Christiani.....	113
Computer Solutions.....	77
CPS Computertechnik.....	2
Decos.....	113
DMV.....	17,37,39,55,59,89,129,133
Interest Verlag.....	34
Münzenloher.....	63
Pähler-Soft.....	113
PD Service Lage.....	9
PDS Service.....	79
RKT.....	137
Signum Medien Verlag.....	27
Soft Maker.....	163
Stähle Software.....	9
Star Division.....	164
Yellow Computing.....	63

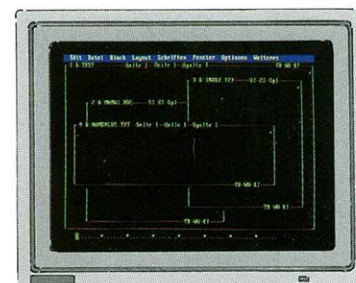


TextMaker 2.01 eröffnet neue Dimensionen!



Und das zum sagenhaft günstigen Preis von

DM 248.-*



Das Textprogramm TextMaker eignet sich nicht nur zum Briefeschreiben. Nein, TextMaker bringt Ordnung auch in Ihr ganz privates Dateienchaos!

Entweder Sie geben Ihren Dateien endlich vernünftige Namen . . . oder Sie verwenden den Dateimanager von TextMaker 2.01. Und plötzlich finden Sie verschollene Dateien wieder.



Coupon

- ☐ Schicken Sie mir bitte eine Demo-Diskette. 10-Mark-Schein liegt bei.
- ☐ 5.25"-Format
- ☐ 3.5"-Format

Name _____

Straße _____

Ort _____

- ☐ TextMaker 2.01 ist genau, was ich suche! Schicken Sie mir ein Exemplar.
- ☐ Verrechnungsscheck anbei
- ☐ Nachnahme

Datum _____ Unterschrift _____

* unverbindliche Preisempfehlung

SoftMaker

Grabbestraße 9 · 8500 Nürnberg 90
☎ 09 11/30 33 33 · Fax 09 11/30 37 96

In Österreich: Otto Rödinger
Klosterneuburger Straße 5 · A-1203 Wien
Telefon 02 22/33 43 08



Star Manager 2.0. Die starke DOS-Hilfe.

Star Manager 2.0 bringt Ihr gesamtes DOS in Schwung. Von 2.11 bis 4.01 sogar im Netzwerk. Schwierige Abläufe von DOS faßt Star Manager 2.0 zusammen und vereinfacht sie. Funktionen, die DOS nicht bietet, ergänzt er. Sogar gelöschte Files stellt er wieder her. Er optimiert die Geschwindigkeit Ihrer Platte, automatisiert Programmstarts und bietet Ihnen Editoren für Texte, Dateien, Disketten, Festplatten etc.

Star Manager verzichtet auf komplizierte Parameter, er reagiert auf Ihre Menü-Anweisungen. Backup, Restore, Formatieren,

Dateien kopieren und alle übrigen DOS-Funktionen erledigt er wesentlich schneller, als Sie es in DOS sonst schaffen.

Star Manager 2.0, die zugkräftige Hilfe für Ihr DOS, erhalten Sie für 99 Mark*; ab 1. September '89 für 198 Mark*.

Coupon ausschneiden, auf eine Postkarte kleben oder in einen Briefumschlag stecken und senden an: STAR DIVISION, Zum Elfenbruch 5-11, 2120 Lüneburg.



STAR DIVISION

INFO-COUPON

Bitte senden Sie mir mehr Informationen über Star-Manager 2.0 und das weitere Software-Angebot von STAR DIVISION.

Name, Vorname

Straße, Platz

PLZ, Ort

DOS Extra 9

* Unverbindliche Preisempfehlung