DOS EXTRA Nr. 15 – 1991

DOS EXTRA

18,– DM 145 öS 18,– sfr 55,– Fmk

Sonderausgabe DOS EXTRA

Nr. 15 1991

ISSN 0934 – 2842

DMV-Verlag

Eine Publikation der
DOS INTERNATIONAL

SBasic 2.0

**Zum Abtippen:
GW-Basic-Compiler
der Extraklasse**

Programmierung

**SKeybgr 4.0:
Tastatur im Griff**

**Prokey:
Eingaben von Geisterhand**

Turbo Pascal im Überblick

Grundlagen

MS-DOS für Einsteiger
Batch-Dateien für Profis
Tips zur Druckerinstallation

Tips, Tricks und Tools
für Einsteiger und Fortgeschrittene

Leistungsfähige Software zum attraktiven Preis

DMV Software

Das komplette Software-Paket für die komfortable Auftragsbearbeitung

Sie suchen eine kompetente Software-Lösung, um Ihre **Aufträge** zu bearbeiten, um **Angebote, Rechnungen, Lieferscheine, Mahnungen** und **Gutschriften** zu erstellen und Ihren **Artikelbestand** zu kontrollieren?

Sie sind der Meinung, daß eine **Kundenverwaltung,** ein **Textprogramm** für Korresponenz und **Serienbriefe** mit Textbausteinen das Leistungsspektrum abrunden muß?

Sie möchten darüber hinaus problemlos auf **Präsentationsgrafiken** für **Umsatzstatistik, Offene Posten** usw. zurückgreifen?

Dann bietet DMV-Faktura die komplette und durchdachte Mittelstandslösung:
Ein breites **Leistungsspektrum,** erprobte **Anwenderfreundlichkeit,** problemlose **Installation** und eingebaute **Sicherheit.**

Systemvoraussetzung:
IBM PC/XT/AT oder kompatibler mit mindestens 640 KByte RAM, Festplatte, MS-DOS ab Version 3.3

DMV-Faktura **DM 398,–***

Faktura DMV



DMV Atelier



Professionelles Grafik-Design

Vektorgrafik auf Ihrem PC

DMV-Atelier bietet Farben- und Schriftgestaltung in professioneller Qualität. Aufbauend auf Schrift, Rechteck-, Kreisfunktion und Bezierkurven lassen sich alle gewünschten Objekte darstellen:

- umfangreiche Manipulation der Schriften und Grafiken: Drehen, Kopieren, Verschieben, Skalieren, Spiegeln, Überlagern, Rastern und Färben
- Schriften in beliebigen Größen
- fertige Grafiken in erweiterbarer Symbolbibliothek
- 4 Schriften standardmäßig
- Einlesen von Pixelgrafiken (PCX, IMG, TIFF)
- Ausgabe an Drucker oder Datei, auch PostScript
- Trace-Funktion zur Umsetzung von Pixelbildern
- Kompatibilität zu DTP-Programmen

Systemvoraussetzungen:
PC/XT/AT mit 640 KByte RAM, Festplatte, Maus, EMS-Speicher wird unterstützt. Druckertreiber: 9- und 24-Nadeldrucker, HP-kompatibler Laserdrucker

DMV-Atelier **DM 249,–***

Zusatzfonts **DM 99,–***
41 Schriften mit Installation

Symbolbibliothek **DM 99,–***
Zusätzliche Symbole

Die Benutzeroberfläche.

ERGO 3.0 arbeitet unter dem modernsten Standard mit kompletter Maussteuerung in verschiebbaren Fenstern mit Scollbalkenmenüs.

ERGO 3.0 im Überblick:
Dateien: Kopieren, Löschen, Umbenennen, Verschieben, Verstecken, Attribute ändern, wiederherstellen. **Verzeichnisse:** Anlegen, Löschen, Umbenennen, Verstecken, Verschieben. **Disketten:** Formatieren, Optimieren, Kopieren. **Sektor-Editor:** Suchen, Ändern, Abspeichern.

Text-Editor: Blockkommandos, Merkerfunktionen, Suchen und Ersetzen, Einrücken, Spaltenblock und Tabulator. **Optionen:** Paßwort, Zugriffsrechte, Druckercodes, Farben, Zeit/Datum setzen, Systeminfo. **TSR:** Laden und Entfernen speicherresidenter Programme. **Anwendermenü:** Anlegen individueller Menüfenster. **OS-Shell:** Sofort in DOS - mit Exit zurück in ERGO. **Hilfe:** Online-Manual, ASCII-Tabelle, Kalender, Rechner.

Minimalkonfiguration:
PC/XT/AT mit mindestens 384 KByte RAM, MS-DOS ab Version 3.0

Ergo 3.0 **Nur DM 199,– ***

ERGO 3.0



DMV-Virenschutz-System



Virus Doktor PC

Sichert Programm-Dateien gegen Virusangriffe ■ meldet und unterbindet Manipulationsversuche ■ verhindert das "Umbiegen" von Read-only-Attributen ■ überwacht Dateiintegrität durch Bildung von Prüfsummen ■ speichert und restauriert Inhalt des CMOS-Setups ■ sperrt auf Wunsch Schreibzugriff

VirusDoktor PC **DM 99,–***

Virus Finder PC

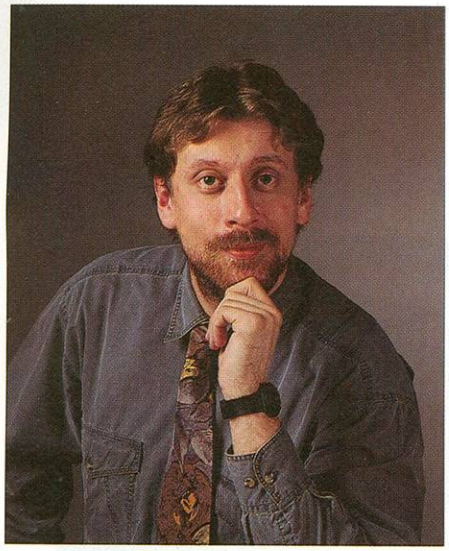
Findet über **300** bekannte Boot- und Linkviren ■ stellt Infektionen fest ■ löscht befallene Dateien physikalisch ■ läßt sich problemlos auf den neuesten Stand bringen. ■ hindert infizierte Programme an der Weiterverbreitung von Viren ■ bietet menügesteuerte Benutzerführung ■ bequemer. Aufruf über die Kommandozeile (Kommandozeilenviren) ■ Dazu Spezialprogramm zum Aufspüren selbstmodifizierter Viren.

VirusFinder PC **DM 149,–***

* Unabhängig von der Anzahl der bestellten Programme berechnen wir für das Inland DM 4,– bzw. für das Ausland DM 6,– Porto und Verpackung. Bitte benutzen Sie die Bestellkarte.

DMV-Verlag • Postfach 250 • 3440 Eschwege

DMV Daten- und Medienverlag



Lernprozeß Computer

»(A)bort, (R)etry, (I)gnore?«, so lautet die Meldung, die manchen Computerbesitzer beinahe um den Verstand gebracht hat, im Original. Und so hat sie als feststehender Ausdruck Eingang in den Sprachschatz des PC-Freaks gefunden. Aber auch die eingedeutschte Fassung neuerer Betriebssystemversionen ((A)bbruch, (W)iederholen, (F)ehler ?) ist nicht dazu angetan, arglosen Anwendern den Umgang mit dem Computer zu erleichtern. Und diese Meldung ist nur ein Beispiel für die zahlreichen unerwünschten Ergebnisse, die beim mehr oder minder sachgerechten Umgang mit den verschiedenen Produkten rund um den Computer auftreten können.

Nun wäre es aber unfair, die Schuld am Mißlingen mancher Aktion von vornherein pauschal der Hard- und Softwareindustrie in die Schuhe zu schieben. Das ausführlichste Handbuch ist nicht umfangreich genug, alle Eventualitäten zu berücksichtigen, die, faßt man alle möglichen Kombinationen von Hard- und Software zusammen, auftreten können.

Schuld ist nicht zuletzt die Fülle an Peripherie und Software, die, eben auf Grund der großen Angebotspalette, viel Spielraum für falschen oder unsachgemäßen Umgang bietet. Jeder Computeranwender weiß, daß es mit dem Auspacken und Aufstellen des Computers nicht getan ist. Software und Peripheriegeräte wollen installiert und angepaßt werden. Später kommen dann auch noch Erweiterungen und neue Programme hinzu. Kurz: Der Computer befindet sich in einem ständigen Veränderungs- beziehungsweise Wachstumsprozeß. Selbstverständlich wachsen mit der Vielfalt auch die Fehlerquellen.

Gerade der erfahrene Anwender wird, je nach persönlicher Einstellung, selbst sehr leicht zu einer Fehlerquelle. Zahlreich sind die Namen derjenigen »Profis«, die das neue Produkt, sei es nun ein Drucker oder ein Anwendungsprogramm, mal eben auf die schnelle »anstöpseln« beziehungsweise »reinladen«, nur »um mal zu sehen, was es bringt«. Installiert, daß heißt, vernünftig auf die anderen Systemkomponenten angepaßt, wird dann später, wenn es nötig wird. Man hat ja im Moment mal wieder keine Zeit. »Später« ist dann entweder »nie«, oder man hat im entscheidenden Moment gerade nicht das Handbuch zur Verfügung, und irgendwie kann man ja ein bißchen tricksen, aber später, spätestens nächste Woche, da wird man dann einmal vernünftig installieren ...

Schließlich lernt man, mit unbefriedigenden Ergebnissen zu leben. Sinn der Sache ist das natürlich nicht. Und müßte auch nicht sein. Schlimm ist es, wenn man auf optimale Ergebnisse verzichten kann, aber auch nicht, es kann sogar ganz nützlich werden:

Fehler im Umgang mit dem Computer und der ihn umgebenden Hard- und Software unterlaufen nicht nur dem blutigen Anfänger. Selbst eingefleischte Profis sind gegen die vielzitierten »Tücken des Objekts« nicht gefeit. Der Unterschied zwischen beiden ist, daß der Einsteiger durch Mißerfolge und Bedienungsschwierigkeiten schnell frustriert und demotiviert wird, während der erfahrene Anwender in der Regel den Fehler (zum Teil sogar intuitiv) erkennen und beheben kann, ja oft sogar, kaum ist die falsche Taste gedrückt, schon leise flucht, weil ihm schon im selben Moment bewußt wird, daß sich gerade keine Diskette in Laufwerk A befindet.

Die nötige Erfahrung aber gewinnt man, das zeigt schon das Leben, aber nicht zuletzt durch Fehler und Rückschläge.

Damit Sie, egal ob Einsteiger oder Fortgeschrittener, zumindest über die häufigsten Hürden leichter hinwegkommen, finden Sie in dieser Ausgabe Hilfen zur Hard- und Software-mäßigen Problemlösung. Darüber hinaus enthält die Ausgabe natürlich noch eine Fülle wichtiger Grundlageninformationen, Tips & Tricks und interessanter Programmlistings.

Viel Erfolg und Spaß mit der DOS Extra 15/91 wünscht Ihnen



Ulrich Eike

INHALT

DOS EXTRA
15/91

ANWENDUNG

MS-DOS transparent 6

Einsteigerlexikon 14

Sterne, Sonnen und Planeten 18

Dem Rat folgt die Tat 20

Befehle, die vom Stapel laufen 26

HARDWARE

Bunt oder nicht; hier keine Frage 42

Stichworte zur Computergrafik 46

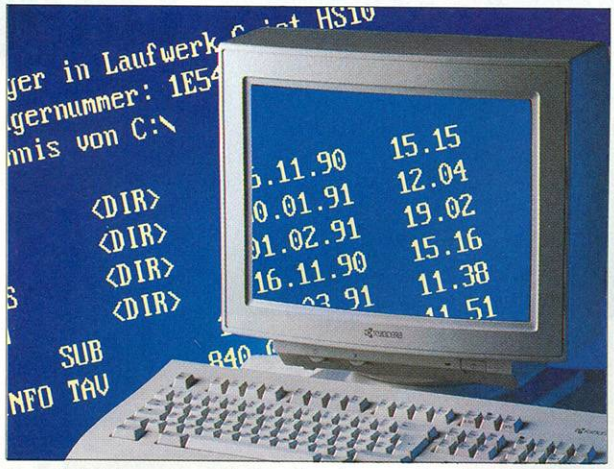
Jetzt gibt es Druck 50

Netzarchitektur 54

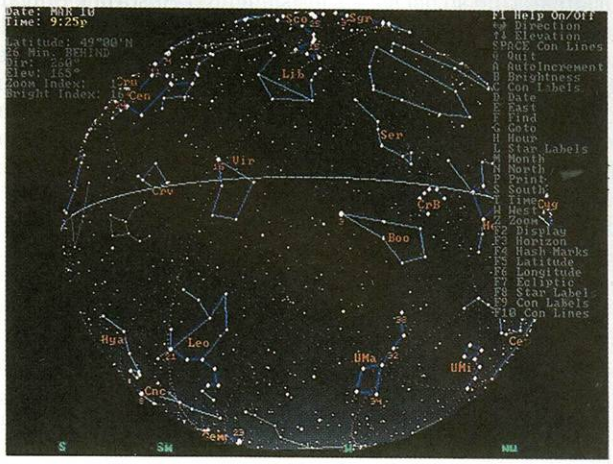
PROGRAMMIERUNG

Die Gedanken des Computers 56

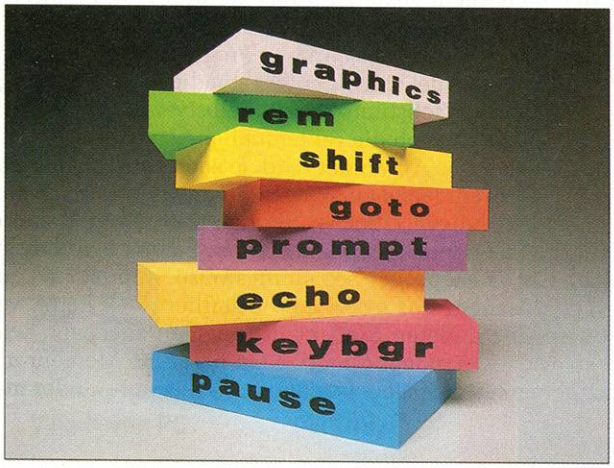
Grafik mit GW-Basic 68



Das Betriebssystem ist das Fundament für die Arbeit mit dem Computer. Über die Grundlagen und Funktionsweisen von MS-DOS und DR-DOS lesen Sie ab Seite 6



Gutes muß nicht teuer sein. Wie Sie für wenig Geld neue Welten entdecken, erfahren Sie auf Seite 18



Stapeldateien erleichtern den Umgang mit dem Computer wesentlich. Hinweise, wie Sie Batchdateien effektiv programmieren, finden Sie ab Seite 26

Durch die Praxis zum Erfolg 74

Turbo Pascal auf einen Blick 80

Kontrollierte Datenflut 90

TIPS & TRICKS

Tips & Tricks für Einsteiger 96

LISTINGS

Der programmierte Untergang 104

SCheck 110

Skeybgr in neuer Version 4.0 114

Eingaben wie von Geisterhand 124

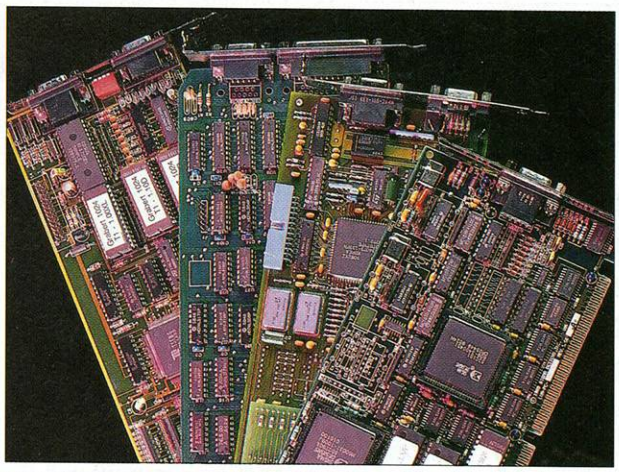
SBasic 2.0: Der »Nachbrenner« für GW-Basic 134

RUBRIKEN

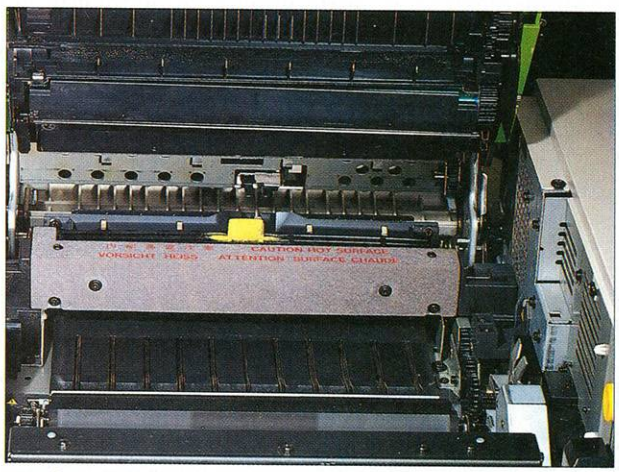
Editorial 3

Bücher 41

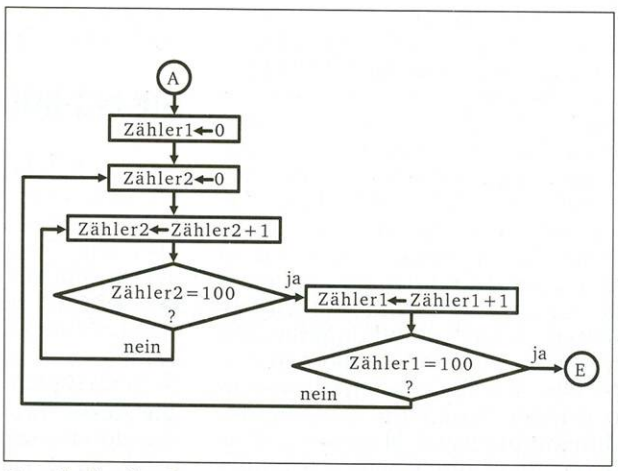
Impressum 166



Gerade auf dem Gebiet der Grafikkarten schreitet die Entwicklung schnell voran. Über die verschiedenen Standards, Vor- und Nachteile lesen Sie ab **Seite 42**



Beschwerlich ist oft der Weg, bis der Drucker optimal konfiguriert ist. Den richtigen Umgang mit den »Mäuseklavieren«, wie DIP-Schalter oft genannt werden, schildert der Beitrag auf **Seite 50**

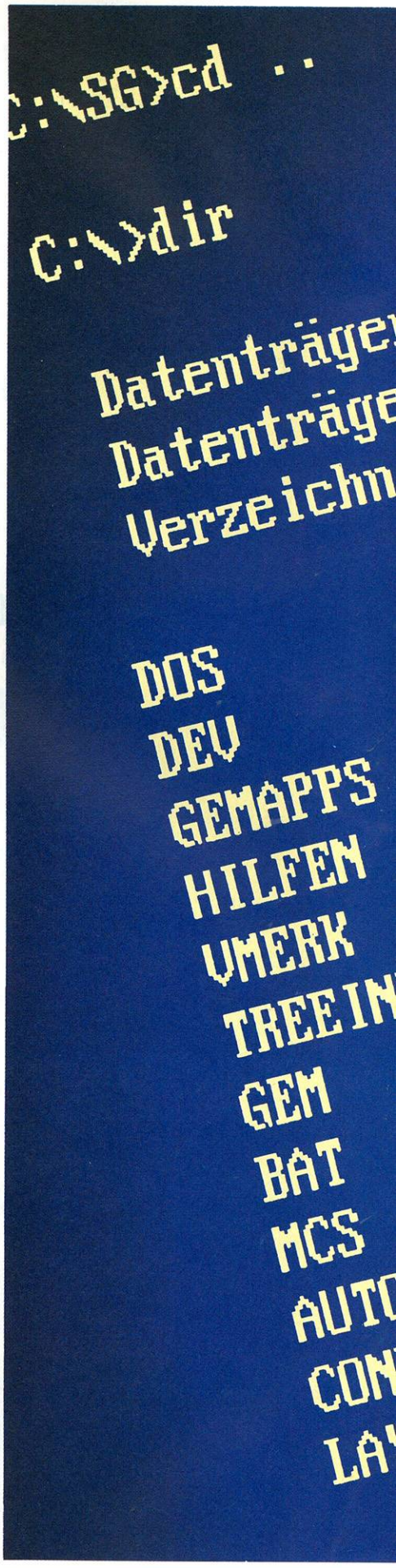


Gerade in der Programmierung ist das richtige Grundlagenwissen Voraussetzung für den schnellen Erfolg. Wichtige Informationen finden Sie ab **Seite 56**

MS-DOS/DR-DOS

MS-DOS transparent

Beim Einstieg in die Computerwelt ist die Beschäftigung mit einem noch unbekannten Betriebssystem nicht einfach. Daher erhalten Sie hier einen Überblick, wie die einzelnen Elemente des Computers und des Betriebssystems MS-DOS miteinander in Wechselwirkung stehen. Auch Hinweise zu DR-DOS fehlen nicht.



Zunächst können Sie dem sich mit »A>« oder »C>« meldenden Betriebssystem noch kein Verständnis entgegenbringen. Welche Fülle an technischer Erfahrung, Leistungsfähigkeit und Flexibilität darin enthalten sind, bleibt verborgen. Mit jedem MS-DOS-Kapitel betreten Sie nicht nur erneut eine neue Welt, sondern auch selbst unlösbar scheinende Probleme lassen sich dann bewältigen. Um vorerst wichtige Wechselwirkungen zwischen PC und Betriebssystem zu verstehen, werfen Sie nun anhand von alltäglichen Beispielen einen Blick »hinter die Kulissen« und klären Vorgänge sowie damit zusammenhängende Begriffe.

Die Aufgaben von MS-DOS

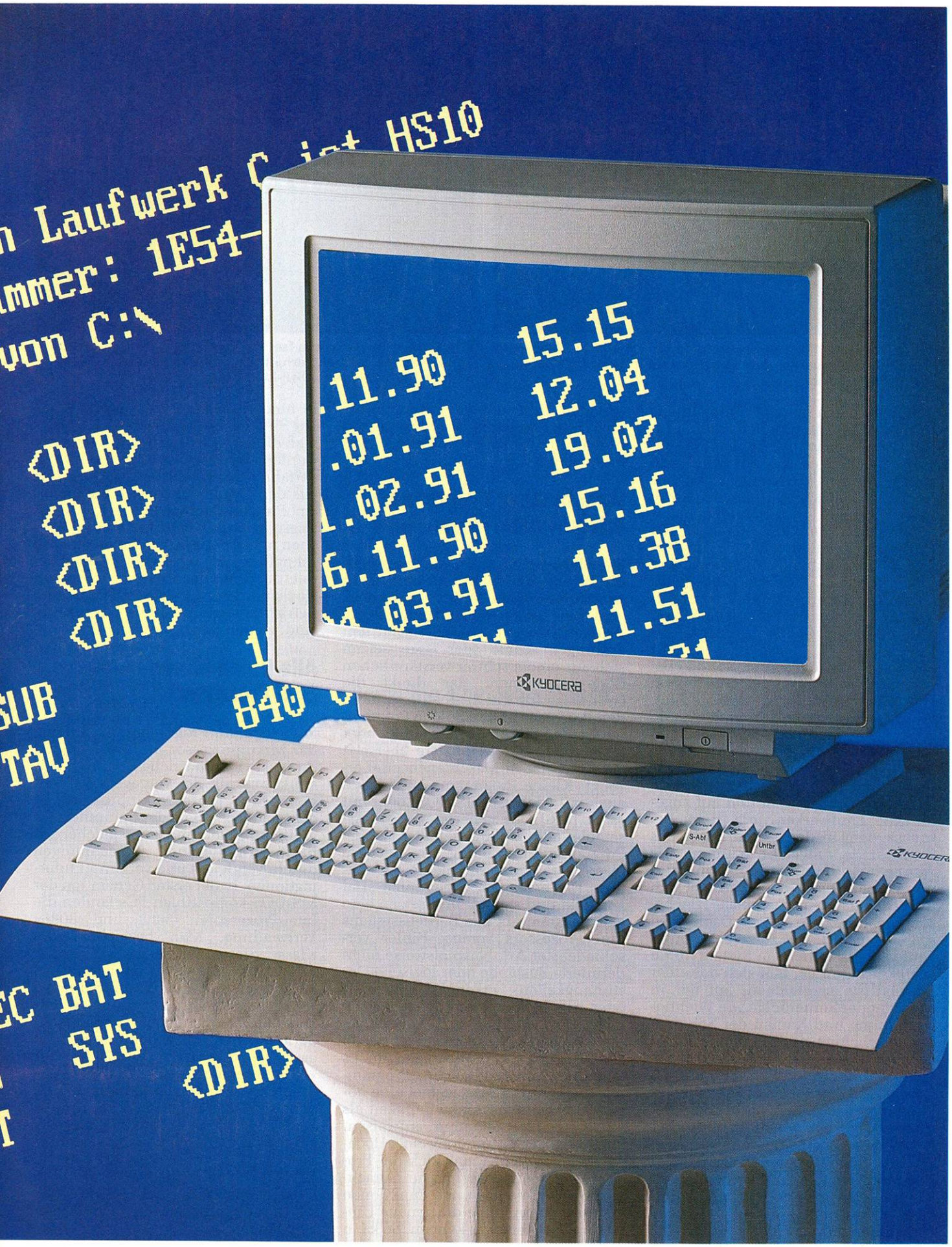
Dem Betriebssystem MS-DOS (»Disk Operating System«) obliegt die Aufgabe zwischen Anwender, Computer und Anwendungsprogrammen zu vermitteln. Es bildet gewissermaßen die Schnittstelle zwischen der Ebene der Schaltkreise, also dem elektronischen Teil, und dem Anwender. Alle grundsätzlichen Aufgabenstellungen sind in diesem System verankert. MS-DOS nimmt Benutzereingaben entgegen und wertet sie aus. Es erkennt Anfragen und veranlaßt beispielsweise den Datenaustausch zwischen Zentraleinheit und externen Geräten, etwa das Lesen und Schreiben von Daten auf einen Datenträger. Zusammen mit dem BIOS (»Basic-Input-/Output-System«) nimmt es Fehlersituationen bei angeschlossenen Geräten und Programmen wahr und bietet Korrekturalternativen. Ohne das Bindeglied Betriebssystem gäbe es nicht diese Fülle von komplexen Programmen zu fast allen Einsatzgebieten, denn ohne MS-DOS müßte sich jedes Programm selbst um die elementarsten Funktionen kümmern. Daher besteht der »Kern« des Betriebssystems aus einer Fülle von Programmen, den »Interruptroutinen«, die bestimmte Bereiche bewältigen. Programmierer, die auf diese bereits bestehenden Programme zugreifen, ersparen sich nicht nur Programmierarbeit, sondern stellen auch sicher, daß Sie die entstandenen Programme auf Computern mit anderer (von Ihrem Programm eventuell nicht unterstützten) Ausstattung einsetzen können. Diese Interrupts werden auch von MS-DOS-Dienstprogrammen verwendet.

Das BIOS ist unter anderem für den Start des Betriebssystems nach dem Einschalten des PC verantwortlich und liegt in Form eines Festwertspeichers (Inhalt nicht veränderbar: »ROM« oder »EPROM«) vor. Bild 1 zeigt einen solchen Festwertspeicher. MS-DOS dagegen wird geladen und belegt Teile des Hauptspeichers (Inhalt veränderbar: »RAM«, siehe Bild 2). Das Laden des Betriebssystems bietet den Vorteil, daß sich neue Versionen veränderten Rahmenbedingungen anpassen lassen. So wird ab MS-DOS 4.01 die 32-MByte-Grenze für Festplattenpartitionen durchbrochen, um der Größe neuer Speichermedien gerecht zu werden (eine Partition ist ein frei festlegbarer Teil der Datenkapazität einer Festplatte, die einem Laufwerksbuchstaben zugeordnet ist).

Die MS-DOS-Befehlszeile

Ist MS-DOS komplett geladen, befindet man sich in der Eingabezeile des Betriebssystems, in der Sie Befehle über die Tastatur eingeben können. Jede Befehlseingabe schließen Sie mit Drücken der Entertaste ab. Damit bestätigen Sie gleichzeitig Ihre Befehlseingabe und übergeben diese an das Betriebssystem.

Die Auswertung übernimmt der Kommandointerpreter »command.com«. Er prüft, ob es sich bei der Eingabe um eine Anweisung handelt, die nicht nachgeladen werden muß (interner Befehl) und führt sie sofort aus. Findet er das Kommando intern nicht, geht er davon aus, daß es sich um einen externen Befehl (Dienstprogramm) handelt.


Laufwerk C ist HS10
mmer: 1E54-
von C:\
<DIR>
<DIR>
<DIR>
<DIR>
15.15
12.04
19.02
15.16
11.38
11.51
SUB
TAV
840
EC BAT
SYS
<DIR>
KYOCERA
S-Abf
Unterbr

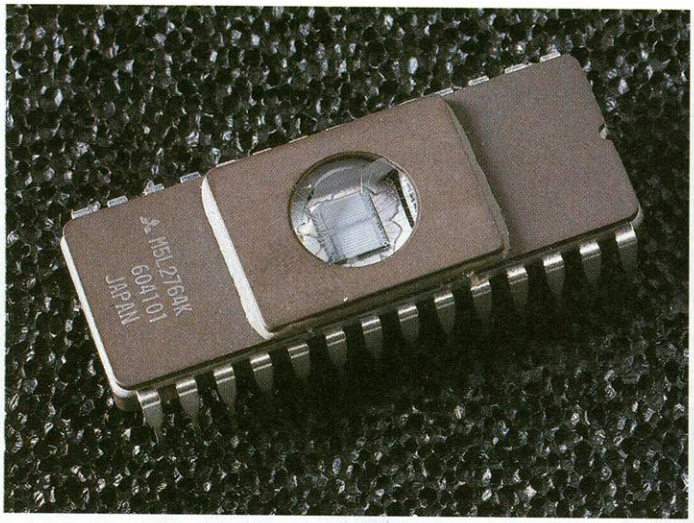


Bild 1. Ein »EPROM« (programmierbarer Festwertspeicher) enthält ein Programm, das dem Computer sofort und ohne Laden zur Verfügung steht

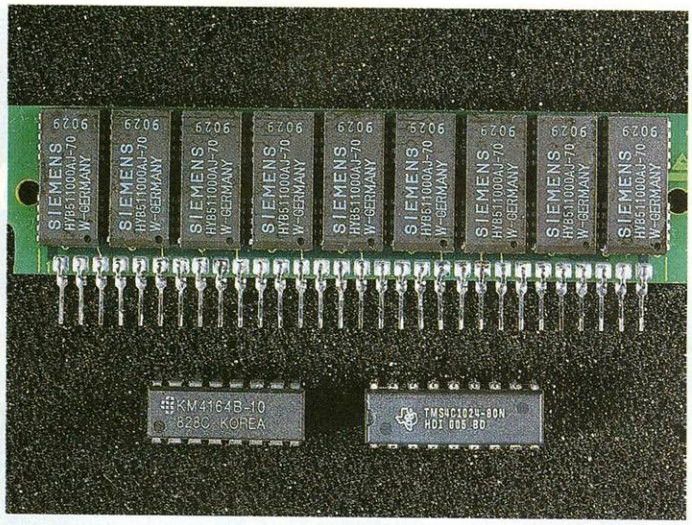


Bild 2. Mehrere Arten von frei les- und beschreibbarem Haupt- und Zusatzspeicher sind erhältlich, unter anderem 256-KBit- und 1-MBit-RAMs sowie SIP-Module

Die Suche nach einem Programm mit dem eingegebenen Namen beginnt im aktuellen Verzeichnis. Stellen Sie sich ein Verzeichnis als eine Art Schublade vor, in der sich Werkzeuge (Programme) und Hilfsstoffe (Dateien) sowie weitere Fächer (Verzeichnisse) befinden. Dieser Symbolik gemäß ist ein Laufwerk ein anderes Zimmer, in dem sich Schubladen mit verschiedenem Inhalt befinden. MS-DOS geht grundsätzlich davon aus, daß sich Kommandos auf die aktuelle Position beziehen, also auf das gewählte Laufwerk und Verzeichnis. So gesehen, verändert das Betriebssystem seine Position erst, wenn es vom Anwender oder einem Programm Anweisung dazu erhält. Wurde nicht in ein Unterverzeichnis gewechselt, befindet man sich im Hauptverzeichnis. Verläuft die Suche nach einem Programmnamen im aktuellen Verzeichnis negativ, sucht MS-DOS in Verzeichnissen und auf zugehörigen Laufwerken, die dem Betriebssystem in der letzten Phase des Systemstarts mitgeteilt wurden. Die Anweisung, die diese Funktion übernimmt, heißt »path« und befindet sich in der Textdatei »autoexec.bat«. Der besondere Charakter dieser Datei wird später erläutert. Die Verzeichnis- und Laufwerksliste, die dem System durch »path« bekannt ist, läßt sich mit einer Inventarliste vergleichen, auf der in absteigender Reihenfolge von Wichtigkeit Räume und Namen der Schubladen verzeichnet sind, in denen zu suchen ist. Wird auch nach dem Durchsuchen der eventuellen Fundstellen (gemäß der Inventarliste) kein solches Programm gefunden, gibt das Betriebssystem die Meldung »Falscher Befehl oder Dateiname!« aus.

War die Suche erfolgreich, wird das Programm geladen und ausgeführt. Es handelt sich um eine Datei mit der Endung »com« oder »exe« oder um eine Stapeldatei mit der Endung »bat«. Die maximal achtstelligen Programmnamen lassen keinen Rückschluß auf den Aufbau zu, wohl aber die dreistelligen Endungen.

Verschiedene Programmarten: »com«, »exe«, »bat«

Stapeldateien bestehen aus Textzeilen, die Befehle enthalten und einfach zu generieren, lesen und zu verändern sind. Eine Kommando-Shell wertet die Befehle einzeln aus. Dateien mit den Endungen »com« oder »exe« setzen sich aus einem schwer verständlichen Code zusammen, der direkt die Grundlage für die Bearbeitung durch den Prozessor bildet. Solche Programme werden mit einem Assembler oder einer Hochsprache (letztere übersetzt ein Compiler in Maschienencode) erzeugt. Kleine Programme lassen sich auch mit »debug« (MS-DOS) oder mit »sid« (DR-DOS) generieren. Ein Assembler oder Makroassembler ist eine Programmierumgebung, in der Sie elementare Prozessorbefehle verwenden. Die erzeugten Programme sind kompakt und schnell und setzen eine genaue Kenntnis des Betriebssystems und Prozessors voraus. Fehler verschiedenster Art, beispielsweise nicht definierte Zustände oder logische Unstimmigkeiten, erkennt, reklamiert oder behebt hier keine Systemebene mehr. Ein Compiler setzt Programme, die in einer Hochsprache geschrieben sind, in Prozessoranweisungen um. Es entstehen umfangreichere Programme, die je nach Einsatzgebiet eine höhere Bearbeitungsgeschwindigkeit als ein Assemblerprogramm aufweisen. Der Vorteil der Verwendung einer Hochsprache liegt darin, daß sich der Programmierer auf wesentliche Sprachelemente, Befehle und Funktionen verlassen kann, ohne die genaue Umsetzung im System zu kennen. So kann in einer Hochsprache beispielsweise der Sinus eines Arguments im Bogenmaß durch eine Anweisung berechnet werden. In Assembler ist vorerst das Verfahren zur Berechnung eines Sinus aus den Grundrechenarten aufzubauen. Programmentwicklung und Testphase in einer Hochsprache ist verglichen mit Assembler problemloser und schneller, und viele Fehler erkennt bereits der Compiler. Zu den wichtigsten Hochsprachen für PCs, die mit dem Betriebssystem MS-DOS ausgestattet sind, zählen Basic, Pascal und C.

Alle Daten führen zum Prozessor

Rufen Sie ein Programm auf, arbeitet der Mikroprozessor (»CPU«, »Central Processing Unit«, siehe Bild 3) die Instruktionen in der vorgegebenen Reihenfolge ab. Dieser Mikroprozessor ist das technische Herz des Computers. Er greift auf Daten im Hauptspeicher zu, führt grundlegende Berechnungen durch, vergleicht und verknüpft Informationen. In der ersten Generation der MS-DOS-kompatiblen PCs fanden die Intel-Prozessoren »8088« und »8086« Verwendung. Man bezeichnet die Klasse dieser Geräte als »PC« oder »XT«. Die nächste Computerfamilie, »AT«, umfaßt den Mikroprozessortyp »80286« (seltener auch »80186«). Seit Verwendung der 80386- und 80486-Serie hat sich als Benennung für den jeweiligen Computer »AT-386« oder »AT-486« eingebürgert. All diesen Mikroprozessoren ist gemeinsam, daß sie fähig sind, den Befehlssatz des Urahns 8088 zu bearbeiten. Dies gilt auch für alternative Mikroprozessoren der V-Serie beispielsweise V20 oder V30 von NEC. Der Unterschied zwischen den einzelnen CPUs besteht in Befehlsvorrat, Geschwindigkeit und technischen Fähigkeiten.

In der Kürze liegt die Würze

Arbeitsspeicher ist die Bezeichnung für den internen Speicher eines Computers (vergleiche RAM). Der Zugriff auf die Dateien 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 benötigt. Wird der Computer ausgeschaltet, geht der Inhalt des Arbeitsspeichers verloren.

AT ist die Abkürzung von »Advanced Technology«. Mit AT werden PCs bezeichnet, die mit den 80286- oder auch 80386-Prozessoren und einer höheren Taktfrequenz arbeiten.

Batch-Dateien, auch als Stapeldateien bezeichnet, stellen eine Sammlung von Betriebsbefehlen dar, die »am Stück« abgearbeitet werden.

Partitionieren bedeutet, Festplatten in logische »Unterlaufwerke« einzuteilen, die vom Betriebssystem als eigenständige Laufwerke behandelt werden.

RAM ist die Abkürzung für »Random Access Memory« (Schreib-Lese-Speicher oder Speicher mit wahlfreiem Zugriff), siehe Arbeitsspeicher.

RAM-Disk ist ein 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.

ROM ist die Abkürzung für Read-Only-Memory (Nur-Lese-Speicher oder Festwertspeicher). Der Inhalt des ROM-Speichers ist unveränderbar und bleibt auch nach dem Ausschalten des Computers erhalten. Auf seinen Inhalt kann nur lesend zugegriffen werden. Beim PC sind im ROM wichtige Daten abgelegt, die für den Start und Betrieb unbedingt notwendig sind.

XT ist die Abkürzung für »Extended Technology«. Die Bezeichnung XT wurde 1983 von IBM für PCs eingeführt, die im Gegensatz zu den IBM-PCs eine 10-MByte-Festplatte besaßen.

Das Betriebssystem MS-DOS nutzt nur jenen Befehlsvorrat, über den ein 8088-Prozessor verfügt, selbst wenn die in Ihrem Computer enthaltene CPU weitaus leistungsfähigere und schnellere Alternativen bietet. Der Grund dafür ist im Bereich der Werterhaltung von Programmen (»Software«) und technischer Ausstattung (»Hardware«) zu suchen. Üblicherweise erfordert ein neuer Prozessor und/oder eine verbesserte Computerarchitektur neue Programme und spezielle Hardware. Ist aufgrund höherer Anforderungen ein neuer PC erforderlich, sind kostspielige Anschaffungen für den alten sinnlos. Das Konzept der PC-kompatiblen Geräte mit dem Betriebssystem MS-DOS stellt sich daher zur Aufgabe, sowohl die Verwendung von Soft- und Hardware auf zukünftigen Computersystemen der 80x86-Familie zu gewährleisten, als auch dem Käufer die Chance einzuräumen, bisherige Computer individuell gestiegenen Ansprüchen anzupassen. Investitionen verlieren so nicht an Wert; Neuentwicklungen und technische Innovationen werden nicht verzögert.

Die Kehrseite der Medaille: Durch Beschränkung auf den Befehlssatz des 8088 bleiben Geschwindigkeit und technische Vorteile fortschrittlicherer Prozessortypen auf der Strecke. Nur die höhere Taktfrequenz (Bearbeitungsgeschwindigkeit der Befehle) würde genutzt, gäbe es nicht bespielsweise Programme zur Speicherverwaltung, die für Anwendungssoftware und Betriebssystem zumindest einen Teil der Fähigkeiten neuerer 80x86-Prozessoren erschließen.

Bedachte Neuinvestitionen

Oft ist kein neuer Computer notwendig, um gestiegenen Bedürfnissen gerecht zu werden. Zahlreiche Anwendungsprogramme, darunter Grafik-, Datenbank- und Textverarbeitungssysteme, brauchen für umfangreiche Tätigkeiten mehr Speicherplatz als im Hauptspeicher verfügbar ist. Meist lagern diese Programme dann Teile auf die Festplatte aus und bearbeiten die Daten dort. Um in solchen Fällen wesentliche Geschwindigkeitssteigerungen zu erzielen, können Sie zwar eine schnellere Festplatte und einen passenden Festplattencontroller sowie einen Computer mit leistungsfähigerem Prozessor erwerben. Eine Aufrüstung des Zusatzspeichers mit weiteren RAM-Bausteinen (RAM = lesbarer und beschreibbarer Speicher) und gegebenenfalls einem Programm zur Nutzung dieses Erweiterungsspeichers (falls nicht im Lieferumfang der Anwendung enthalten) würde unserem Beispiel gemäß mindestens den gleichen Zeitgewinn erbringen. Die hierfür erforderlichen Ausgaben machen nur einen Bruchteil des Preises eines neuen PC aus. Weitere Beispiele für kostengünstigen Leistungszuwachs sind Festplatten, bei denen Sie durch Austausch des Festplatten- und Disketten-Controllers eine merklich höhere Datendurchsatzquote erzielen. Konsultieren Sie bei solchen Fragen den Fachhandel. Die von Ihnen verwendeten Steckkarten besitzen eventuell schon für das Gesamtsystem optimale Werte, wenn ein vorhandener Coprozessor nicht genutzt wird.

Die Bezeichnungen »Hauptspeicher« und »Erweiterungsspeicher« oder »Zusatzspeicher« wurden genannt. Dieser begrifflichen Unterscheidung entsprechen beachtliche Tatsachen: Der Arbeitsspeicher kann durchaus zwei MByte (mehr als zwei Millionen Zeichen) umfassen, wovon ihm allerdings als Hauptspeicher maximal 640 KByte (gemäß unserem Beispiel nur etwa ein Drittel des Speichers) zur Verfügung stehen. Diese 640-KByte-Grenze setzte Anfang der achtziger Jahre der gebräuchliche 8088-Prozessor fest. Damals war die Beschränkung auf einen 640-KByte-Hauptspeicher und eine maximale Partitionsgröße (externer Speicherplatz für ein Laufwerk) von 32 MByte eher theoretischer Natur, da interne und externe Speichermedien zum einen (für heutige Verhältnisse) sehr teuer und die Programme entsprechend klein dimensioniert waren. Technische Leistungssteigerung und Preisverfall sowie immer mehr internen und externen Speicherplatz belegende Programme ließen die ehemals hypothetischen Grenzen zu wahren Nahtstellen werden.

Haupt- und Erweiterungsspeicher

Ein Programm arbeitet grundsätzlich im Hauptspeicher. Nicht nur das Betriebssystem liegt in diesem Bereich, auch das Programm selbst und je nach Anwendung eventuell sehr viele zu verarbeitende Daten. Der Erweiterungs- oder Zusatzspeicher wird nur von einem Teil der kommerziellen Programme verwendet, und auch diese Systeme können dort nur Daten ablegen und bei Bedarf schnell darauf zugreifen. Das Betriebssystem selbst erlaubt jedoch nicht, Programme direkt aus diesem Bereich zu bearbeiten. Häufig dienen als Zusatzspeicher RAM-Disks und Cachespeicher. RAM-Disks sind Laufwerke, die physikalisch nicht bestehen, im Speicher aber als solche behandelt werden. Lese- und Schreibzugriffe sind sehr viel schneller als solche auf richtigen Laufwerken oder schnellsten Festplatten. Der Inhalt von RAM-Disks geht mit dem Ausschalten des Computers oder durch ein Reset verloren. Sie lassen sich daher nur als schnelle Zwischenspeicher nutzen. Cachespeicher weisen die Ei-

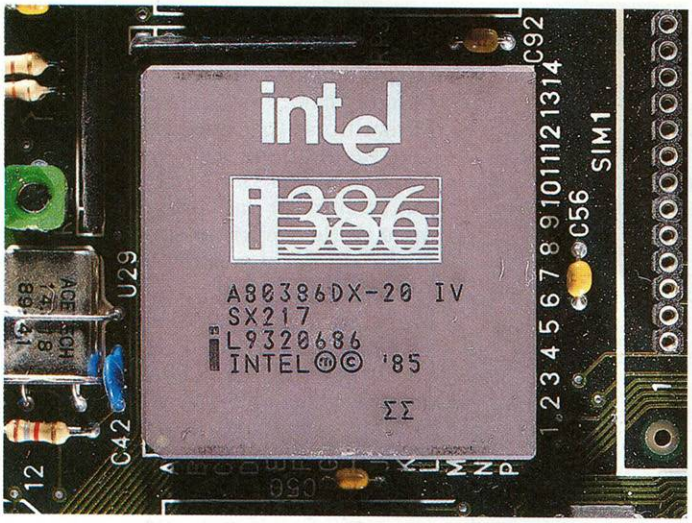


Bild 3. Das Gehirn des Computers besteht aus einem Mikroprozessor der Intel-80x86-Familie

genschaften auf, sich Zugriffe auf Programme und Daten zu »merken« und die eingelesenen Informationen zu speichern. Das Konzept dieser Technik besteht darin, auf Programme, Programmteile oder Daten mehrmals nacheinander zuzugreifen. Das Cacheprogramm versorgt das laufende Programm mit den im RAM zwischengespeicherten Daten.

Es gibt zwei Arten von Erweiterungsspeicher: »Expanded Memory«, ein Erweiterungsspeicher, oft auch Upper Memory genannt, der vom Ende des Hauptspeichers (640 KByte) bis zur 1-MByte-Speichergrenze reicht und insgesamt 384 KByte umfaßt. »Extended Memory«, ein Zusatzspeicher, der sich im Bereich oberhalb der 1-MByte-Grenze befindet. Des weiteren existieren Programme, die »Extended Memory« in »Expanded Memory« umsetzen. Nur wenige Programme, die in der Lage sind, Zusatzspeicher zu nutzen, erlauben den Einsatz beider Erweiterungsarten.

Das Betriebssystem als »offenes Konzept«

Beide Konzepte brauchen sogenannte Treiberprogramme oder Einheitentreiber, die Sie in der Datei »config.sys« festlegen. Die Besonderheiten dieser Systemsteuerdatei werden noch erläutert. Wird ein solches Treiberprogramm aktiviert (es kann beispielsweise in einem größeren Anwendungsprogramm enthalten sein und sich bei der Installation automatisch in »config.sys« eintragen), so wird dem Betriebssystem vermittelt, daß es diese Speicherbereiche nutzen kann. Mit dieser Technik verfügen MS-DOS und alle Programme prinzipiell über den Zugriff auf solche Erweiterungen, unabhängig davon, welcher Prozessor vorliegt oder wie stark der Computer mit zusätzlichem Speicher ausgebaut wurde. Durch diesen »indirekten« Einsatz wird zwar Effizienz eingebüßt, dafür bleiben aber ältere Programme funktionsfähig.

Zur Anpassung an die zuvor schon beschriebenen Unterschiede zwischen verschieden ausgestatteten Computern sowie zum Einstellen zahlreicher weiterer »Umgebungsbedingungen«, die von Anwender zu Anwender differieren, dienen die beiden Dateien »autoexec.bat« und »config.sys«. Sie legen unter anderem den landesspezifischen Zeichensatz, das Datumformat, Suchpfade, die Kommunikation mit Zusatzgeräten, die Auswahl von Betriebszuständen und die Definition von Systemvariablen dort fest. Dadurch kann der gleiche Computertyp in verschiedenen Ländern eingesetzt werden, im angelsächsischen Sprachraum beispielsweise ohne Umlaute und mit dem Datumformat »Monat.Tag.Jahr«, im deutschsprachigen Einzugsgebiet mit Umlauten und dem Datumsformat »Tag.Monat.Jahr«. Gleichermaßen passen Sie den Drucker dem landesspezifischen Zeichensatz an. Auf diese Weise festgelegte Werte berücksichtigen MS-DOS-Dienstprogramme ebenso wie auch Anwendungsprogramme, die über zeichensatz- oder datumsorientierte Funktionen verfügen. Weiterhin bietet dieses Verfahren jedem Anwender genügend Freiraum, um sowohl die Verzeichnisstruktur nach eigenen Wünschen festzulegen, als auch nicht-standardmäßige Zusatzgeräte oder Einschubkarten zu verwenden. Das Betriebssystem wird zum offenen Konzept.

Da die beiden Dateien »autoexec.bat« und »config.sys« von Bedeutung sind, lohnt es sich, deren Aufgaben näher zu betrachten. »config.sys« enthält grundlegende Anweisungen bezüglich der Systemeinstellungen, die in dieser Datei zwar variabel definierbar sind, nach Ausführung aber als unveränderliche Werte vorliegen, bis Sie den Computer ausschalten.

Systemkonfiguration mit »config.sys« und »autoexec.bat«

Die Befehlsausführung der »config.sys« ist somit ein wesentlicher Bestandteil beim Starten des Betriebssystems. Jede Änderung wird erst beim nächsten Start (das heißt, bei Reset oder erneutem Einschalten) gültig. Eine zentrale Rolle bei den Config-Anweisungen spielt »device«. Über eine Zuweisung der Form

```
device=[Laufwerk][Pfad]Treibername.sys
[Parameter]
```

werden als »Treiber« bezeichnete Programme geladen und aktiviert. Diese ändern oder erweitern die Fähigkeiten des Betriebssystems auf speziellen Gebieten. So können nicht-standardmäßige Steckkarten, Zusatzgeräte oder besondere Ausstattungen des Computers mittels dieser Treiber angesprochen und verwendet werden (beispielsweise Grafikkarten oder Scanner). Das Treiberprogramm schließt also eine Lücke für den Einsatz einer noch nicht im Betriebssystem verankerten Technologie. Ebenfalls in »config.sys« werden die Voreinstellungen dafür getroffen, wieviele Dateien sich gleichzeitig in Bearbeitung befinden dürfen (»files = «) und wieviele Pufferbereiche für eingelesene Daten angelegt werden sollen (»buffers = «). Bei Textverarbeitungsprogrammen machen sich diese Pufferbereiche dann bemerkbar, wenn Sie mehrfach einige Seiten vor- und zurückblättern. Statt eines erneuten Zugriffs auf einen Datenträger werden die benötigten Informationen aus dem Puffer bezogen, was wesentlich schneller erfolgt. Beide Einstellungen benötigen jedoch, abhängig von der festgelegten Anzahl, Speicherplatz aus dem Hauptspeicher. Erst ab MS-DOS-Version 4.x läßt sich zusätzlich vorhandener Speicherausbau hierfür nutzen. Pro definierter Einheit werden bei »files« 48 Byte und bei »buffers« 512 Byte benötigt. Diese Werte erscheinen niedrig, summieren

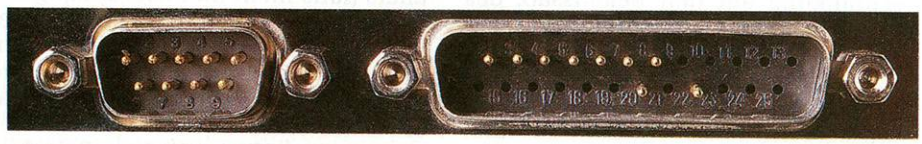


Bild 4. Den Anschluß einer seriellen Schnittstelle erkennt man an der Anschlußleiste in Form eines 9- oder 25poligen Steckers

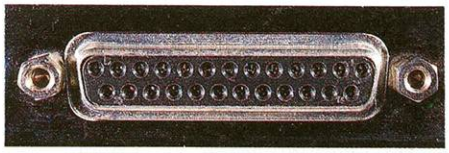


Bild 5. Die parallele Schnittstelle ist eine unverwechselbare 25polige Buchse

sich aber bei zu groß gewählten Angaben. Für den eigenen Bedarf empfiehlt sich der Test mit unterschiedlichen Werten. Als »goldenes Mittelmaß« bewährten sich für »files« und »buffers« jeweils 25. Unter den weiteren sieben Konfigurationsbefehlen sind »break« (Programmunterbrechungen mit [Strg-C] oder [Strg-Untbr] auch bei Lese- und Schreibzugriffen prüfen), »country« (Definition von Datum-, Zeit- und Währungsformat) und »drivparm« (Festlegen von speziellen Parametern, üblicherweise bei Diskettenlaufwerken oder Festplatten) hervorzuheben. Die restlichen Befehle dienen Aufgaben, die eher Internas zugerechnet und nur in wenigen Bereichen genutzt werden können.

Nachdem MS-DOS durch »config.sys« ein Teil der Systemeinstellungen übermittelt wurde, lädt das Betriebssystem den Kommandointerpreter »command.com«. Dieser enthält die als »internen« Befehle bezeichneten Kommandos und kümmert sich um die Auswertung von Eingaben sowie das Laden und Starten von »externen« Befehlen und Programmen.

»autoexec.bat« – die letzte Phase des Systemstarts

Zu diesem Zeitpunkt fehlen noch einige wichtige Konfigurationsanweisungen in Form von Umgebungsvariablen des Betriebssystems oder durch ausführbare Dateien mit den Endungen »com« oder »exe«. Im Unterschied zur »config.sys« enthält »autoexec.bat« nur interne und externe MS-DOS-Befehle, die Sie in gleicher Form auch in der Betriebssystem-Eingabezeile eingeben können. Der Vorteil von »autoexec.bat« liegt darin, daß diese Datei dann automatisch ausgeführt wird, wenn sie sich beim Starten des Systems im Hauptverzeichnis des Startlaufwerks befindet. Somit können Sie »autoexec.bat« alle Befehle und Anweisungen ausführen lassen, die nach dem Einschalten des Computers oder nach einem Reset erforderlich sind. Mit Ausnahme der automatischen Bearbeitung ist »autoexec.bat« eine gewöhnliche Stapelverarbeitungsdatei (siehe hierzu den Beitrag »Befehle, die vom Stapel laufen« in diesem Sonderheft). Ein elementarer Bestandteil von »autoexec.bat« ist das Setzen des Suchpfades für ausführbare Dateien sowie im deutschsprachigen Raum das Aktivieren des Tastaturtreibers (»keybgr«, ab DOS 3.3 »keyb gr« oder mit Computer oder Tastatur gelieferte alternative Tastaturtreiber). Diesem Programm obliegt es, Sonderzeichen, Umlaute, Klammern sowie »y« und »z« an die uns gewohnten Positionen auf der Tastatur zu rücken.

Sehr empfehlenswert ist auch die Verwendung von »prompt« mit den Parametern »pg«. »$p« weist die MS-DOS-Eingabezeile an das aktuelle Laufwerk und den aktuellen Pfad auszugeben, »$g« hängt das »>«-Zeichen an, beispielsweise »c:\>«. In der MS-DOS-Eingabezeile (»Eingabeaufforderung«) weiß der Anwender daher sofort, welches Laufwerk und welches Verzeichnis aktiviert ist. Das ist beim Löschen oder Kopieren von Dateien von Bedeutung. Sofern benötigt, werden Programme zum Betrieb von Netzwerk oder Maus mit speziellen Werten initialisiert. Verschiedene Betriebsarten der seriellen (Bild 4) und parallelen (Bild 5) Schnittstelle sowie der Grafikkarte legen Sie mit »mode« fest. Wer anstelle der Betriebssystemeingabezeile (»Shell«) sein Lieblingsprogramm, eine Oberfläche, einen Dateimanager oder eine Standardanwendung aktivieren möchte, kann dies ebenfalls mit »autoexec.bat« managen.

Bearbeiten von Textdateien

Je nachdem, welche Programme Sie besitzen, sollten Sie die beiden Dateien mit einem ASCII-Editor anlegen oder überarbeiten. Sofern Sie mit einem Textverarbeitungsprogramm arbeiten, das Texte ohne Formatierung speichert, greifen Sie darauf zurück. Da aber nicht alle Anwender über solche Programme verfügen, wird im folgenden davon ausgegangen, daß nur die im Lieferumfang des Betriebssystems enthaltenen Programme verfügbar sind. Die Bearbeitung des Inhalts der Textdateien »autoexec.bat« und »config.sys«, besonders das Korrigieren einzelner Zeilen, ist umständlich. Zum Erzeugen einer Textdatei verwenden Sie den Copy-Befehl, indem Sie als Quelle die Tastatur und als Ziel eine Datei angeben (siehe Beitrag »Befehle, die vom Stapel laufen« in diesem Sonderheft – auch bei folgenden Befehlen und Techniken, sofern sie Ihnen noch nicht bekannt sind):

```
copy con a:\autoexec.bat
```

Generieren Sie sicherheitshalber die Datei zuerst auf Laufwerk »A:« oder »B:« mit einer eingelegten neuen Diskette, um nicht versehentlich die Originaldateien zu überschreiben, und kopieren Sie zusätzlich die Dateien »autoexec.bat« und »config.sys« vom Startlaufwerk auf eine Sicherheitsdiskette. Treten nun Probleme auf, können Sie auf die ursprünglichen Werte zurückgreifen. Mit Betätigen der Taste [F6] oder [Strg-Z] und [Enter] schließen Sie die Dateneingabe ab. Jede Befehlszeile kann solange überarbeitet werden bis Sie die Taste [Enter] betätigen.

Diese Praktik ist nur zum Anlegen kleiner Dateien sinnvoll. Bestehende Dateien können Sie mit »copy con« nicht bearbeiten oder korrigieren. Bei Änderungen oder Erweiterungen von ASCII-Dateien leistet (sofern kein anderer Editor oder kein Textverarbeitungsprogramm verfügbar ist) »edlin« diese Dienste. Obwohl »edlin« unter anderem die Funktionen Text suchen, ersetzen sowie Teile davon löschen, einfügen, verschieben und kopieren bietet, werden nur seine Grundfunktionen erläutert, da die Bedienung dieses zeilenorientierten Editors sehr gewöhnungsbedürftig ist und oft in unerwünschten Ergebnissen oder Fehleingaben endet.

Sie rufen »edlin« mit dem Namen und Zugriffspfad der zu generierenden oder zu bearbeitenden Datei auf, beispielsweise

```
edlin a:\autoexec.bat
```

Ein Sternchen signalisiert, daß sich »edlin« im Befehlsmodus befindet. Wird zuvor die Meldung »Ende der Eingabedatei!« ausgegeben, so ist nicht ein Fehler unterlaufen, sondern eine bereits bestehende Datei mit dem angegebenen Namen eingelesen worden. Die Eingabe von [l], gefolgt von [Enter], zeigt in diesem Falle alle vorhandenen Textzeilen mit der entsprechenden Zeilennummer an. Wird eine Zahl angegeben, gelten alle weiteren Befehle für beziehungsweise ab dieser Zeile.

Nehmen wir an, Sie kopierten (zusätzlich zu einer Sicherheitskopie auf einer weiteren Diskette) die Datei »autoexec.bat« ins Hauptverzeichnis in Laufwerk »A:«, und die entsprechende Diskette liegt im verriegelten Laufwerk. Der Aufruf

```
edlin a:\autoexec.bat
```

gefolgt von [l] und [Enter] könnte dann einen wie in Bild 6 gezeigten Inhalt darstellen. Zeile 2 initialisiert eine Grafikkarte. Die in Zeile 6 und 7 aufgerufenen Programme beschleunigen die Tastatur und schalten den Zifferblock der Tastatur zur Verwendung als Cursortasten um. Zeile 10 ruft einen alternativen Tastaturtreiber auf, und Zeile 11 aktiviert die Maus mit weiteren Werten. All diese Programme gehören nicht zum MS-DOS-Lieferumfang, sondern liegen den jeweiligen Zusatzgeräten bei. Andere wurden als Hilfsprogramme gekauft oder selbst programmiert.

Da jeder Computer anders ausgestattet ist, weisen »autoexec.bat« und »config.sys« auf jedem Computer andere Inhalte auf. Wie der Eintrag »share« in

Zeile 9 zeigt, werden auch Dienstprogramme des Betriebssystems nur von Fall zu Fall verwendet. In diesem Beispiel ist ein Netzwerk eingesetzt und »share« daher erforderlich. Angenommen, der Suchpfad für ausführbare Dateien soll um das Verzeichnis »pack« auf Laufwerk »D:« erweitert werden. Nun brauchen Sie nicht allen dort befindlichen Programmen Pfad- und Laufwerksbezeichnung voranstellen, um den Zugriff auf diese Dateien zu garantieren. Nehmen wir weiter an, daß in diesem Verzeichnis Komprimierungs- und zugehörige Hilfsprogramme enthalten sind, die überall aufrufbar sein sollen.

Der Stern in Zeile 1 zeigt an, daß alle Befehle für beziehungsweise ab dieser Zeile gültig sind. Da sich »path=« zum Setzen von Suchpfaden in Zeile 5 befindet, geben Sie [5] und [Enter] ein. Der Inhalt dieser Zeilennummer wird daraufhin ausgegeben und die Folgezeile weist durch »5:*« die gewünschte aktuelle Zeile sowie die Bereitschaft zur Zeilenbearbeitung aus. Die Eingabe der Zeilennummer schaltet auch aus dem Befehls- in den Bearbeitungsmodus um. Alle Befehlstasten, die in der MS-DOS-Eingabezeile verfügbar sind, können zur Bearbeitung der Textzeile eingesetzt werden. Möchten Sie an das bisherige Zeilenende »d:\pack;« anfügen, kopieren Sie mit [F3] alle Zeichen aus dem Eingabepuffer in die aktuelle Zeile. Die Schreibmarke befindet sich bereits am Ende und es bleibt daher nur noch die Eingabe von »d:\pack;« mit abschließendem [Enter]. Zur Rückversicherung, daß die Korrektur auch ausgeführt wurde, drücken Sie nochmals [l] und [Enter], um alle Textzeilen darzustellen. [e] mit folgendem [Enter] schließt die Textbearbeitung mit »edlin« ab und speichert alle Änderungen. Mit [q] anstatt [e] verlassen Sie den Zeileneditor nach einer Abfrage ohne den Text zu sichern. Das Original bleibt unverändert. Anstelle der Bearbeitung einer speziellen Textzeile durch Eingabe der zugehörigen Nummer nach dem Stern ist es auch erlaubt, Zeilen komplett zu entfernen oder neue einzufügen. Die Anweisungen »d« für Löschen oder »i« für Einfügen nutzen wiederum die aktuelle Position bei »edlin«. Zeilenposition und Anweisung lassen sich aber auch kombiniert angeben. Beispielsweise löscht »8,11 d« die Zeilen 8 bis 11. »3 i« fügt nach der bisherigen Zeile 2 einen Eintrag ein. Beachten Sie hier jedoch, daß der Einfügemodus solange erhalten bleibt, bis mit [Strg-C] die Texteingabe beendet und in den Befehlsmodus zurückgeschaltet wird.

An diesen Beispielen der Bearbeitung von ASCII-Dateien mit »edlin« wird deutlich, daß Sie trotz leistungsfähiger Kommandos dieses Zeileneditors besser auf einfacher zu bedienende Programme zurückgreifen.

Die bereits erwähnten Text- und ASCII-Dateien (Abkürzung für »American Standard Code for Information Interchange«, eine Zeichensatznormierung) dürfen nicht mit Dateien verwechselt werden, die unter anderem Text zusätzlich aber Steueranweisungen für externe Geräte wie beispielsweise Drucker enthalten.

Der PC-Zeichensatz

Das Umschalten von Schriftarten und -größen bei Druckern erfolgt je nach Druckerfamilie (Matrix-, Laser- oder Postscript-Laserdrucker) mit weitgehendst genormten Steuersequenzen, die zur Befehlsumschaltung unter anderem Zeichen verwenden, die in Text- oder ASCII-Dateien nicht enthalten sein dürfen. Der MS-DOS-Befehl »type« wird beispielsweise in seiner Funktionsfähigkeit beeinträchtigt, wenn bestimmte Zeichen im Text enthalten sind, die als Steueranweisungen mißverstanden werden. Das 26. Zeichen des PC-Zeichensatzes wird bei ASCII-Dateien als Dateiende (»EOF«, »End of File«) interpretiert. Alle Zeichen nach einem Dateiendekennzeichen werden von Befehlen und Programmen, die ASCII-Dateien voraussetzen, nicht erkannt und auch nicht bearbeitet. Der ASCII-Zeichensatz enthält alle Zeichen von 32 bis 127, darin enthalten sind Groß- und Kleinbuchstaben, Zahlen und Sonderzeichen wie etwa »%«. Diese Zeichen geben Sie mit [Taste] oder [Umschalt-Taste] ein. Die Steuersequenzen sind im PC-Zeichensatz von 0 bis 31 enthalten und lassen sich teilweise mit [Strg-Taste] erreichen. Unter diesen Steuersequenzen befinden sich Codes zur Durchführung eines Rückschrittes auf das vorherige Zeichen ([Strg-H], »Backspace«, Position 8 im Zeichensatz), zur Ausgabe eines Tones ([Strg-G], »Bell«, Position 7 im Zeichensatz) oder zur Einleitung einer Steuersequenz ([Strg-[], »Escape«, Position 27 im Zeichensatz). Die Zeichen größer als 127 werden auch als erweiterter ASCII-Zeichensatz bezeichnet. Dieser weist Zeichen des griechischen Alphabets, Linien und Rahmenzeichen – und die deutschen Umlaute sowie »ß« auf. Da die deutschen Umlaute in fast jeder Textdatei (beispielsweise auch bei Textausgaben oder Kommentaren in Stapeldateien) vorkommen und auch korrekt bearbeitet werden können, sollte der Begriff »ASCII-Datei« entsprechend in »Dateiinhalt aus ASCII- und erweitertem ASCII-Zeichensatz« umbenannt werden. Aber selbst dann ist die Umschreibung nicht ganz korrekt, weil Wagenrücklauf und Zeilenschaltung, die die einzelne Zeilen voneinander trennen, dem Bereich der Steuerzeichen entstammen (Codes 13 und 10). Gerade bei Computern stimmt die Lebensweisheit: Keine Regel ohne Ausnahmen. Eine ganze Reihe Druckerprobleme hängen direkt damit zusammen, daß die Umlaute außerhalb des ASCII-Zeichensatzes liegen.

```
A:\>edlin autoexec.bat
Ende der Eingabedatei!
*l
   1: *@echo off
   2: tvgabio
   3: cls
   4: set comspec=c:\command.com
   5: path=c:\;c:\tools;c:\dos;
      c:\batch;c:\tasm;
   6: keyfast
   7: numoff
   8: prompt $p$g
   9: share
  10: ckeygr
  11: mouse /ld /s100
*
```

Bild 6. »edlin« stellt den Inhalt einer Datei mit Zeilennummern dar

Lernen Sie Programme nutzen

An dieser Stelle vermutet man, der PC-Zeichensatz sei eine fixe Einrichtung und alles weitere läßt sich daraus ableiten. Aber es hängt vom jeweils aktiven Programm ab, welche Bedeutung den einzelnen Zeichen zukommt. Trotz Normierung auf zahlreichen Gebieten umschließt jedes »Zeichen« mehrere Bedeutungen. Ein Code kann das bezeichnete Zeichen codieren, ebenso jedoch eine Assembleranweisung, einen Zahlenwert (oder einen Teil davon) sowie ein Element aus einer Tabelle von Werten, die eine übergeordnete Einheit festlegen (unter anderem zum Aufbau von Bildschirm- und Druckerzeichensätzen hoher Auflösung). Diese relative Bedeutung ist abhängig vom Einsatzgebiet des Programms. Sie bildet eine wesentliche Kenntnis im Umgang mit PCs. Letztlich bestehen alle Programme und Informationen aus einem Alphabet von 256 Zeichen – und nur deren Anordnung und Zusammenhang entscheidet darüber, ob das Ergebnis ein Spiel, eine komplexe Datenbankanwendung, ein verschlüsselter Text oder wirres und sinnloses Durcheinander von Werten ergibt.

Lernen Sie die einzelnen Unterscheidungen kennen und deuten und zu Ihrem Vorteil nutzen!

(Herbert Buckel/Gerda Hutt/gb)

ColoRIX



Grafik-Design mit himmlischen Möglichkeiten

ColoRIX ist ein einfach zu bedienendes Grafikwerkzeug. Es ist für nahezu jeden Anwendungsbereich geeignet. Sowohl private Anwender als auch Grafiker, Ingenieure, Kartographen, Designer, Fotografen, Programmierer, technische Autoren, Lehrer und Professoren setzten das ausgereifte und ausgefeilte Programm erfolgreich ein.

ColoRIX eignet sich hervorragend zur nachträglichen Bildbearbeitung, gleichgültig ob ein Bild im Format PCX/PCC, GIF, IMG, WPG (bitmapped), TIFF, TARGA oder im Textformat vorliegt: ColoRIX unterstützt die gängigsten Grafikformate. Einem Zusammenspiel mit weit verbreiteten Programmen wie Word 4.0 und 5.0, PC Paintbrush, AutoCad, AutoDesk Animator, Ventura Publisher, Word Perfect und Windows (BMP-Dateien) steht somit nichts im Wege.

Daten und Fakten zu ColoRIX:

Import- und Export-Dateitypen: PCX/PCC, GIF, IMG, WPG (bitmapped), TIFF, Targa, BMP, Text.

Maximale Auflösung und Farben: VGA-Grafikkarten mit 256 KByte-Videospeicher, 640 x 400 Punkte und 256 Farben. Mit 512 KByte, 800 x 600 Punkte und 256 Farben und mit einem MByte 1024 x768 Punkte und 256 Farben.

Zeichenwerkzeuge: Pinsel, Radierer, Sprühdose, Glätter (Anti-Aliasing), Smudge (Weichzeichner).

Zeichenelemente: Linie, Viereck oder Rechteck, Kreis, Ellipse, Ein- oder Zwei-Punkt-Kurven, Füllen, Gießen (Füllen mit Muster).

Bildmanipulation: Horizontales und vertikales Spiegeln, Rotieren (in Ein-Grad-Schritten), Kippen, Vergrößern und Verkleinern.

Farbmanipulation: HSV und RGB für individuelle Farbanpassung und über HSV globale Farbpaletteneinstellung.

Muster und Farbverläufe: Maximal 16 (frei definierbar).

Zoom-Funktion: Vier- oder achtfache Vergrößerung von Teilbereichen einer Zeichnung.

Text-Funktionen: 35 integrierte, skalierbare Fonts, einstellbare Attribute, Farbwahl über aktuelle Farbpalette, Texteditor mit Import- und Export-Funktion von Textblöcken.

Zusätzliche Programme: **PrintRIX,** Druckprogramm für ColoRIX, gleichzeitig Bildschirmfoto-Utility. **Present und Presentc,** Diashow und Diashow-Compiler. **VRIX,** programm- und sprachenabhängiges Bildschirmladeutility. **RIXLATE,** Konvertierungsutility für Bildschirmformate und Bildschirmauflösungen. **RES-TEST,** Grafikkarten Testutility.

Systemvoraussetzungen: 640 KByte Arbeitsspeicher, MS-DOS ab Version 2.0, EGA- oder VGA-Grafikkarte, Festplatte, Maus.

ColoRIX **nur 299,-***

Für faszinierende Raytracing-Grafik:

LIGHTNING 3D **nur 199,-***

* Unabhängig von der Anzahl der bestellten Programme berechnen wir für das Inland DM 4,-- bzw. für das Ausland DM 6,-- Porto und Verpackung. Bitte benutzen Sie die Bestellkarte.

DMV-Verlag • Postfach 250 • 3440 Eschwege

DMV Daten- und Medienverlag

Grundlagen

Einsteigerlexikon

Jeder PC-Besitzer steht nach dem Erwerb seines ersten PC vor dem gleichen Problem. Fachausdrücke, Abkürzungen und Anglizismen erschweren den Einstieg in die interessante und unterhaltsame Computertechnik. Das Einsteigerlexikon schafft hier rasche Abhilfe.

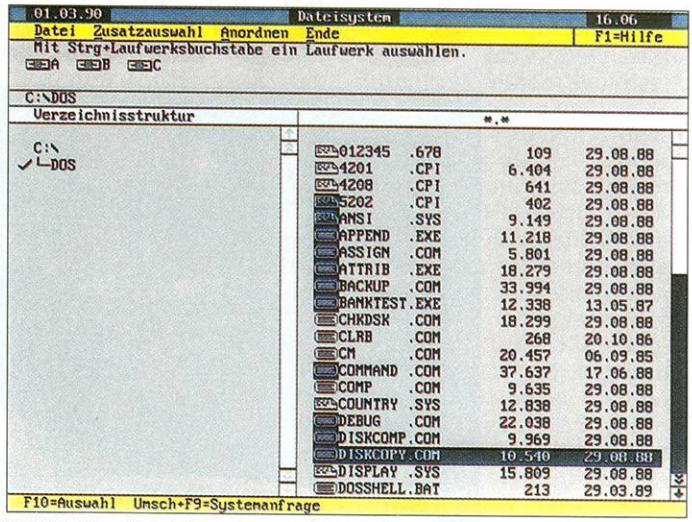


Bild 1. Das Betriebssystem MS-DOS bietet in seiner neuesten Version eine übersichtliche Benutzeroberfläche

Akustikkoppler: Mit dem Akustikkoppler überträgt man Daten über die Telefonleitung. Der Akustikkoppler wird an die serielle Schnittstelle des PC angeschlossen.

Alphanumerische Variablen: Eine alphanumerische *Variable* ist ein Platzhalter, dessen Inhalt aus beliebigen Zeichen bestehen kann. Im Gegensatz zu *numerischen Variablen* sind auch Buchstaben und Sonderzeichen erlaubt. Den Inhalt einer alphanumerischen Variablen bezeichnet man als *Zeichenkette* oder *String*.

Anschluß: Ein-/Ausgabekanal zu und von einem Computer für die Ansteuerung von peripheren Geräten (*Port*)

ANSI: ANSI (»American National Standard Institute«) ist eine amerikanische Behörde, die für Normierungen auf dem Gebiet der Datenverarbeitung zuständig ist (ähnlich dem DIN-Institut in der Bundesrepublik Deutschland). Das ANSI hat beispielsweise den *ASCII*-Zeichensatz und die Standardbefehlssätze von vielen *Programmiersprachen* festgelegt.

Anweisung: *Kommandos* in einer *Programmiersprache*, die den Computer veranlassen, notwendige Operationen durchzuführen.

Anwendungsprogramm: Der Oberbegriff »Anwendungsprogramm« faßt alle Programme zusammen, die ein Benutzer bedienen kann, ohne eine *Programmiersprache* zu beherrschen. Typische Anwendungsprogramme sind *CAD-Programme*, *Datenbankprogramme*, *Tabellenkalkulationsprogramme* und *Textverarbeitungsprogramme*.

Arbeitsspeicher: Interner Speicher des PC. Der Arbeitsspeicher ist ein schneller Schreib- und Lesespeicher, daher häufig die Bezeichnung »*RAM*« für Arbeitsspeicher. Er ist unterteilt in kleine Speicherplätze. Diese sind durchnumeriert und daher einzeln adressierbar. Alle Programme und Daten werden in den Arbeitsspeicher geladen, bevor man sie anwenden beziehungsweise bearbeiten kann. Der Zugriff auf die einzelnen Speicherbereiche des Arbeitsspeichers geschieht wesentlich schneller als auf Datenträgern wie Diskette oder Festplatte, da er computerintern, das heißt ohne lange Übertragungswege, erfolgt. Der Inhalt des Arbeitsspeichers ist »flüchtig«, das heißt, er geht verloren, wenn Sie den Computer ausschalten.

Array: Begriff aus der Mathematik für ein Zahlenfeld (siehe *Feld*)

ASCII: ASCII (»American Standard Code of Information Interchange«) ist ein 7-Bit- beziehungsweise 8-Bit-Zeichencode, der von den meisten Computern für den Austausch von Informationen benutzt wird. ASCII wurde vom *ANSI* als Standardzeichensatz festgelegt.

Assembler: Ein Umwandlungsprogramm, das in *Assembler-Programmiersprache* geschriebene Programme in Maschinencode übersetzt. Nur ein in Maschinencode übersetztes Programm wird vom Computer verstanden.

Assembler-Programmiersprache: In der Assembler-Programmiersprache sind die Befehle an die Architektur des Prozessors angepaßt. Daher unterscheiden sich die Assembler-Sprachen von Prozessor zu Prozessor, zum Beispiel 8080-Mikroprozessor-Assembler-Sprache und so weiter.

Basic ist die weltweit bekannteste Programmiersprache

Basic: Basic (»Beginner's All Purpose Symbolic Instruction Code«) ist die weitverbreitetste *Programmiersprache* der Welt. Sie wurde in den sechziger Jahren am Dartmouth-College von John Kemeny und Thomas Kurtz zur Demonstration von grundlegenden Programmstrukturen konzipiert. Basic gilt als zeitsparendes Mittel zur Programmentwicklung und ist eine sehr einfach zu lernende Programmiersprache. Im PC-Bereich hielt Basic durch den Microsoft-Standard *GW-Basic* Einzug. Neben der großen Verbreitung im Heimbereich hat Basic auch im semiprofessionellen und professionellen Bereich durch die Entwicklung moderner Basic-Dialekte wie *Quick Basic* einen bemerkenswerten Aufschwung erlebt. Das Listing zeigt ein typisches Basic-Programm.

Befehl: siehe *Kommando*

Begrenzungszeichen: Begrenzungszeichen werden benutzt, um den Anfang beziehungsweise das Ende einer Dateneinheit auf einem Speichermedium zu kennzeichnen. Üblicherweise verwendet man Kommata, Leerzeichen, Punkte und Semikola zur Trennung und Gliederung von Daten. In vielen *Programmiersprachen* hat sich der Einsatz des Return-Zeichens (*ASCII*-Code 13) als Begrenzungszeichen durchgesetzt.

Benutzeroberfläche: Eine Benutzerführung, die die Anwendung eines Programms oder den Aufruf von Befehlen erleichtert. Um den Bildschirm übersichtlich zu gestalten, sind in modernen Programmen die Benutzeroberflächen grafisch aufgebaut – mit Fenster und Menüs – und unterstützen eine *Maus*. Die Benutzeroberfläche von *MS-DOS* (Bild 1) gilt als gutes Vorbild.

Betriebssystem: Ein Betriebssystem besteht aus mehreren *Programmen*, die sowohl die Ein- und Ausgabe von Daten als auch die Systemfunktionen steuern und somit die Grundlage für alle internen Vorgänge bilden.

Boolesche Logik: Unter diesem Begriff faßt man alle logischen *Verknüpfungsoperationen* der *Programmiersprachen*

zusammen. Tabelle 2 zeigt die bekanntesten booleschen Operatoren. Variablen mit dem *Datentyp* Boolean können nur zwei Werte annehmen: wahr (englisch »true«), das heißt, eine Bedingung trifft zu, und falsch (englisch »false«), das heißt, eine Bedingung trifft nicht zu. Die Funktion der Vergleichsoperationen wurde 1864 von dem englischen Mathematiker George Boole festgelegt.

C: »C« ist eine leistungsfähige, systemnahe Programmiersprache. Die Systemprogrammierer Brian W. Kenningham und Dennis M. Ritchie begannen 1972 mit der Entwicklung dieser neuen Programmiersprache unter der Zielsetzung, eine flexible Hochsprache für die Programmierung eines ebenso flexiblen Betriebssystems zu entwerfen. Da das geplante Betriebssystem »Unix« nicht an einen Computertyp gebunden war, sondern auf beliebige Rechner übertragbar sein sollte, wurde auch beim Sprachkonzept auf strikte Maschinenunabhängigkeit geachtet. Darüber hinaus sollte die geplante Sprache schnell und maschinennah sein, um ein leistungsfähiges Betriebssystem wie Unix realisieren zu können. Das erfolgreiche Ergebnis war die Programmiersprache C. Die Sprache C ist ein hervorragendes Werkzeug für die Programmierung von Betriebssystemen.

»Computer Aided Design« findet im Flugzeug- und Autobau Verwendung

CAD: CAD ist die englische Abkürzung für »Computer Aided Design«. Mit CAD-Programmen kann man technische Zeichnungen anfertigen oder Modelle konstruieren.

Compiler: Ein Compiler wie *Turbo Pascal* oder *Quick Basic* übersetzt einen Quelltext nach erfolgreicher Fehlerbeseitigung in einen maschinenspracheähnlichen Code (»Objekt-Code«) oder sogar direkt in Maschinensprache. Ein auf diese Weise übersetztes Programm erreicht wesentlich höhere Geschwindigkeiten als ein *Interpreter*.

Coprozessor: Der sogenannte Arithmetikprozessor unterstützt die Zentraleinheit (englisch: »CPU«, Abkürzung für »Central Processing Unit«) bei arithmetischen Operationen und steigert die Arbeitsgeschwindigkeit von Programmen durch Übernahme bestimmter Funktionen, auf die er spezialisiert ist. Der Coprozessor wird nicht von allen *Programmen* unterstützt.

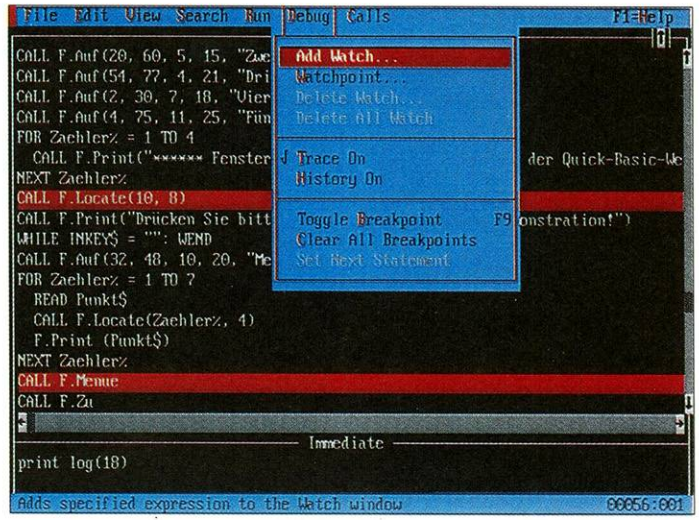


Bild 2. Der Quick-Basic-Compiler gehört zu den bekanntesten Basic-Varianten auf dem Markt

Der Coprozessor unterstützt den Prozessor bei mathematischen Aufgaben

Cursor: Wird auch Schreibmarke oder Positionsanzeiger genannt. Der Cursor markiert durch einen Pfeil oder ein kleines Rechteck die aktuelle Schreibposition auf dem Bildschirm. Er läßt sich in Anwendungsprogrammen oft mit den Cursortasten (auch Pfeiltasten genannt) bewegen (siehe *Maus*).

Datenbank: Datenbanken verwendet man, um große Datenbestände zu verwalten. Sie greifen durch spezielle Verfahren schnell auf bestehende Dateien zu und erlauben eine sinnvolle Datenbetreuung mit Hilfe von komfortablen Kommandos.

Datentyp: Fast alle in einem Programm enthaltenen Informationen sind eingeteilt in die grundlegenden Datentypen: Integer-Zahlen, Fließkommawerte, Text und Wahrheitswerte. Die genauen Beschreibungen der Datentypen finden Sie unter den Stichworten *Boolesche Logik*, *Fließkommavariablen*, *Integervariablen* und *Alphanumerische Variablen*.

Feld: Ein Feld oder *Array* ist eine strukturierte Sammlung von Daten oder festen Werten. Ein Feld setzt sich aus mehreren *Variablen* zusammen. Einerseits kann jedes Element des Feldes unter demselben Variablennamen angesprochen werden – der einzige Unterschied besteht im Indexwert. Andererseits sind Sie in der Lage, jedes Feldelement so zu bearbeiten wie eine Einzelvariable des gleichen Variablentyps. Die Angabe der Nummer wird auch als *Index* bezeichnet.

Fließkommavariablen: Die Fließkommavariablen sind eine Teilmenge der *numerischen Variablen*. Fließkommavariablen können mit rationalen Werten arbeiten, (siehe *Variable*).

Grafikkarten: Eine Grafikkarte ist eine Rechnerbaugruppe, die die Text- oder Grafikinformationen in eine für den Monitor verständliche Form umwandelt. Auflösung (Punktdichte) und Farbigkeit der Darstellung hängen sowohl von der verwendeten Grafikkarte als auch vom Monitor ab. Im Laufe der Zeit sind viele Grafikkarten auf den Markt gekommen (»MDA«, »Hercules«, »CGA«, »EGA« und »VGA«). Die leistungsfähigste Grafikkarte ist zur Zeit die »VGA« mit 640x480 Punkten (Grafikmodus) und 256 darstellbaren Farben, die aus 262144 Farben gewählt werden können. Die EGA-Karte hat eine Auflösung von 640x350 in 16 Farben, die CGA-Karte arbeitet mit

Typ	Wertebereich	Aufbau
Byte	0 bis 255	8 Bit vorzeichenlos
Integer	-32768 bis 32767	16 Bit mit Vorzeichen
LongInt	-2147483648 bis 2147483647	32 Bit mit Vorzeichen
ShortInt	-128 bis +127	8 Bit mit Vorzeichen
Word	0 bis 65535	16 Bit vorzeichenlos

Tabelle 1. Die Integer-Variablen lassen sich grundsätzlich in fünf verschiedene Datentypen einteilen

Operator	Operation	Operandentyp	Ergebnistyp
and	logische Verknüpfung	Boolean	Boolean
not	logische Negation	Boolean	Boolean
or	logisches Oder	Boolean	Boolean
xor	logische Antivalenz	Boolean	Boolean

Tabelle 2. Boolesche Operationen sind von elementarer Bedeutung für Vergleiche und Bedingungen in Programmiersprachen

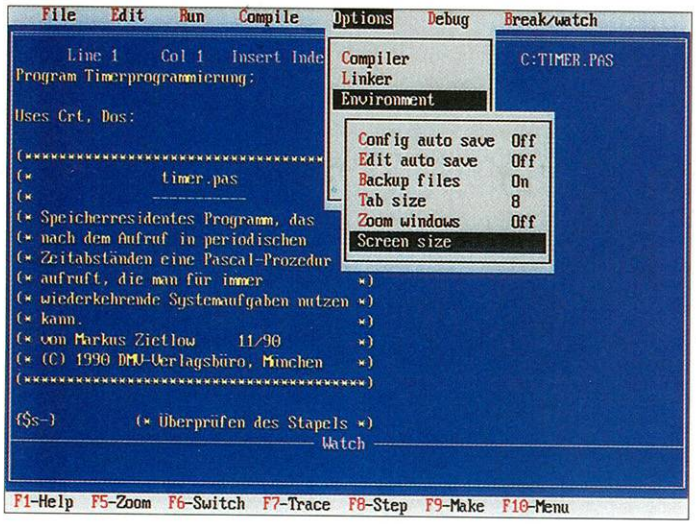


Bild 3. Mit Turbo Pascal eröffnet sich dem PC-Anwender die Welt der strukturierten Programmierung zu einem günstigen Preis

640x200 und kann nur vier Farben darstellen.

GW-Basic: GW-Basic ist der bekannteste *Interpreter* für PCs. Er ist in Kombination mit dem *Betriebssystem* MS-DOS im Lieferumfang eines PC enthalten und eignet sich besonders gut für den schnellen Einstieg in das Programmieren. Der Wortbestandteil »GW« (»Graphics and Windows«) weist darauf hin, daß GW-Basic schon in seiner ursprünglichen Form Grafikbefehle enthält und ein (eingeschränktes) Arbeiten mit Fenstern erlaubt.
GW-Basic gilt als Standard im PC-Bereich. Auf dem Befehlssatz dieses Basic-Interpreters basieren moderne *Compiler* wie beispielsweise *Quick Basic*.

Index: Der Index bestimmt innerhalb eines *Feldes* ein Feldelement. Dieses Element wird wie eine *Variable* angesprochen.

Integervariablen: Die Integervariablen sind eine Teilmenge der *numerischen Variablen*. Integervariablen können nur ganze Zahlen aufnehmen. Eine Aufstellung der meist verwendeten Integertypen mit den zugeordneten Wertebereichen finden Sie in Tabelle 1 (*Variable*).

Interpreter: Ein Interpreter übersetzt ein Hochsprachenprogramm während der Ausführung Befehl für Befehl in Maschinensprache. Weil jedoch beim Ausführen eines *Befehls* neben der Übersetzung und den gesamten Fehlerabfangroutinen auch noch die vom jeweiligen *Kommando* benötigten Systemroutinen aufgerufen werden müssen, ist die erreichbare Verarbeitungsgeschwindigkeit in den meisten Fällen vergleichsweise gering. Ein Interpreter ist wesentlich langsamer als ein *Compiler*. Der bekannteste Vertreter dieses Programmiersprachen-Genres ist *GW-Basic*.

Kommando: Ein Programmiersprachenkommando oder *Befehl* ist ein vordefiniertes Wort aus einer Folge von Zeichen, das den Computer veranlaßt, bestimmte Operationen wie die Ausgabe einer Zeichenkette durchzuführen. Jedes Kommando wird vom *Compiler* oder *Interpreter* übersetzt.

LCD: LCD ist die englische Abkürzung für »Liquid Crystal Display« (deutsch: »Flüssigkristallanzeige«). LCD-Monitore sind im Laptop-Bereich häufiger, da sie leicht sind und wenig Strom verbrauchen. Allerdings ist die Qualität der Darstellung nicht so gut wie auf Standardmonitoren.

Listing: siehe *Programm*

Maus: Aufgrund der äußeren Form und der Farbe als »Maus« bezeichnetes Eingabegerät zur Steuerung des *Cursors* beziehungsweise Mauszeigers auf dem Bildschirm. Die Maus ist mit einer Rollkugel ausgestattet, mit der auf einer glatten Oberfläche (zum Beispiel einer Tischfläche) die Bewegungen der Maus auf den Bildschirm übertragen werden.

MS-DOS: DOS ist die Abkürzung für Disk Operating Systems, also ein Betriebssystem, das von der Firma Microsoft (MS) für 16-Bit-Prozessoren entwickelt wurde und durch den Einsatz im IBM-PC große Verbreitung fand.

Numerische Variablen: Eine numerische *Variable* ist ein Platzhalter, dessen Inhalt ausschließlich aus Zahlen, dem Vorzeichen und dem Dezimalpunkt bestehen kann. Im Gegensatz zu *alphanumerischen Variablen* sind weder Buchstaben noch Sonderzeichen erlaubt.

Pascal: Pascal ist eine der bekanntesten und weitverbreitetsten Programmiersprachen, die ihren Namen zu Ehren des französischen Philosophen und Mathematikers Blaise Pascal (1623-1662) trägt. Im Jahre 1971 hat Nikolaus Wirth diese Programmiersprache entworfen, die als Weiterentwicklung von Algol 60 angesehen wird. Pascal gilt als zeitsparendes Mittel zur Programmentwicklung und ist eine leicht erlernbare Programmiersprache. Die Vorzüge von Pascal sind die *strukturierte Programmierung* und die Vielseitigkeit. Auf dem PC-Sektor fand Pascal über den Borland-Standard *Turbo Pascal* Einzug in den Computerbereich. Im semiprofessionellen Bereich und Heimbereich hat die Programmiersprache durch die Weiterentwicklung und die Realisierung moderner Konzepte einen bemerkenswerten Erfolg erlebt.

Über die Ein- und Ausgabekanäle des Computers findet die Kommunikation statt

Port: Ein-/Ausgabekanal eines Computers zum Anschluß einer Kommunikationsleitung (siehe *Anschluß*)

Programm: Die Zusammenfassung von mehreren Anweisungen, die dazu dienen, komplexe Probleme zu lösen. Ein Programm wird auch als *Listing* bezeichnet.

Programmiersprache: Eine Programmiersprache ist ein künstlicher Dialekt, der dazu gedacht ist, einen kleinen Befehlssatz zur Verfügung zu stellen, mit dem man *Programme* schreibt. Programmiersprachen sind beispielsweise *Assembler*, *Basic*, *C*, *Pascal*.

Prozedur: Eine Prozedur ist das gleiche wie ein *Unterprogramm*.

Quelltext: Ein Quelltext ist die Definition eines *Programms*.

Quick Basic: Der moderne Basic-Dialekt Quick Basic von Microsoft ist weit verbreitet. Der *Compiler* arbeitet mit einer optisch aufwendigen, menügeführten Benutzeroberfläche und kann auch mit *GW-Basic* konzipierte Programme bearbeiten. Der integrierte Editor ist Wordstar-kompatibel und überzeugt durch Schnelligkeit und seine vielfältigen Funktionen zur Bearbeitung des eingegebenen Quelltextes. Quick Basic präsentiert zu jedem benötigten Themenkomplex ein Pulldown-Menü (Bild 2).

RAM: Englische Abkürzung für »Random Access Memory« (deutsch: »Schreib-Lese-Speicher«). RAM ist ein Speicherbaustein. *Arbeitsspeicher* sind meistens aus RAM-Bausteinen aufgebaut (siehe *ROM*).

ROM: Englische Abkürzung für »Read Only Memory« (deutsch: »Reiner Lesespeicher«). Der Inhalt des ROM ist im Gegensatz zum *RAM* »nicht flüchtig«, das heißt, die gespeicherten Daten bleiben auch nach dem Abschalten des Computers erhalten. Das ist unerläßlich, da auf dem ROM die grundlegenden System- und Startroutinen gespeichert sind. Der Anwender greift

auf die ROM-Informationen nur lesend zu, er kann sie nicht verändern.

Scanner: Ein Scanner ist ein Eingabegerät, das auf Vorlagen befindliche Grafiken oder Schriften einliest und als Kopien im PC ablegt. Moderne Texterkennungsprogramme ergänzen Scanner. Diese Hilfsprogramme wandeln die als »Bild« vorhandenen Schriften in ASCII-Texte um, die der Anwender auf diese Weise mühelos archivieren und weiterverarbeiten kann.

Scanner digitalisieren grafische Vorlagen

Schleife: Eine Schleife ist gleichbedeutend mit einer Folge von Computerinstruktionen, die so lange wiederholt werden, bis eine vorgegebene Bedingung erfüllt ist. Um die Effizienz zu erhöhen, können Sie mehrere Schleifen ineinander schachteln.

String: Das Wort »String« ist die englische Bezeichnung für eine *Zeichenkette*.

Strukturierte Programmierung: Die strukturierte Programmierung ist eine Programmiertechnik, die den Erfolg von *Pascal* entscheidend mitgetragen hat. Pascal erlaubt es, aufgrund der zur Verfügung gestellten Datenstrukturen Algorithmen direkt in die Programmiersprache umzusetzen. Da man komplexe Programme aus kleineren modularen Einheiten zusammensetzen kann, erhält man leicht lesbare Quelltexte.

Tabellenkalkulation: Eine Tabellenkalkulation berechnet anhand mathematischer Formeln Bezüge zwischen einzelnen Datenfeldern. Dieser Typ von *Anwendungsprogrammen* eignet sich beispielsweise, um große Zahlenmengen zu verwalten.

Textdatei: In einer Textdatei sind alle Datensätze mit einer variablen Länge und unterschiedlichen Struktur hintereinander auf dem Speichermedium abgelegt (siehe *Begrenzungszeichen*).

Turbo Pascal: Turbo Pascal ist der bekannteste *Compiler* für die Programmiersprache *Pascal*. Turbo Pascal wartete als erstes Pascal mit einer fenster- und menüorientierten Benutzeroberfläche auf (Bild 3). Turbo Pascal unterstützt verschiedene *Bildschirmkarten* und den Großteil der ANSI-Pascal-Funktionen.

Textverarbeitung: Mit einem guten Textverarbeitungsprogramm kann man Texte erfassen, am Bildschirm korrigieren, Zeichenketten suchen, verschieben und kopieren, Geschriebenes formatieren und archivieren. Die meisten Textverarbeitungsprogramme, wie beispielsweise MS-Word, stellen darüber hinaus eine Rechtschreibkorrektur, einen Thesaurus (Rechtschreibhilfe) und vielfältige Bearbeitungsfunktionen zur Verfügung (Bild 4).

Textverarbeitungsprogramme gehören zu den meistverwendeten Anwendungen

Unterprogramm: Es gibt viele Situationen, in denen ein Abschnitt eines *Programms* mehrfach benutzt werden muß. In diesem Fall helfen *Schleifen* nicht weiter. Für diese Zwecke gibt es Unterprogramme. Ein Unterprogramm wird durch die Angabe seines Namens aktiviert und kehrt nach Ausführung der in ihm enthaltenen *Befehle* in den aufrufenden Programmteil zurück. Unterprogramme machen ein Programm übersichtlicher.

Variable: Variablen sind einheitliche Zeichengruppen, die als Platzhalter für bestimmte Werte in Programmen fungieren.

Variablentyp: Der Typ einer *Variablen* bestimmt, welche Werte in ihr abgelegt werden können. Die Größe von *numerischen Variablen* und *alphanumerischen Variablen* ist abhängig vom *Compiler* und ihrer Deklaration (*Integervariablen*, *Fließkommavariablen*).

Vergleichsoperation: Jede Vergleichsfunktion wird benutzt, um zwei Datenfelder zu vergleichen und bei Bedarf das Ergebnis in einem Datenfeld abzulegen. In *Programmiersprachen* gibt es mehrere Vergleichsoperationen. Die gebräuchlichen Operatoren sind gleich, ungleich, kleiner, kleinergleich, größer und größergleich. Mit den booleschen Operatoren lassen sich Vergleiche verknüpfen.

Jedes Inhaltsverzeichnis läßt sich in Unterverzeichnisse aufteilen

Verzeichnis: Ein Verzeichnis enthält Angaben über alle auf einem Speichermedium in einem Inhaltsverzeichnis gespeicherten Dateien. Jeder Eintrag gibt den Namen, die Größe, das Erstellungsdatum und die Erstellungszeit einer Datei an.

Verzweigung: Eine Verzweigung setzt die Programmausführung an der angegebenen Stelle fort. Zu diesem Zweck benutzt man Sprungbefehle, die die Art des Programmaufrufs bestimmen.

Zeichenkette: Eine Zeichenkette ist eine Folge von beliebigen ASCII-Codes (*alphanumerische Variablen*).

(Markus Zietlow/ba)

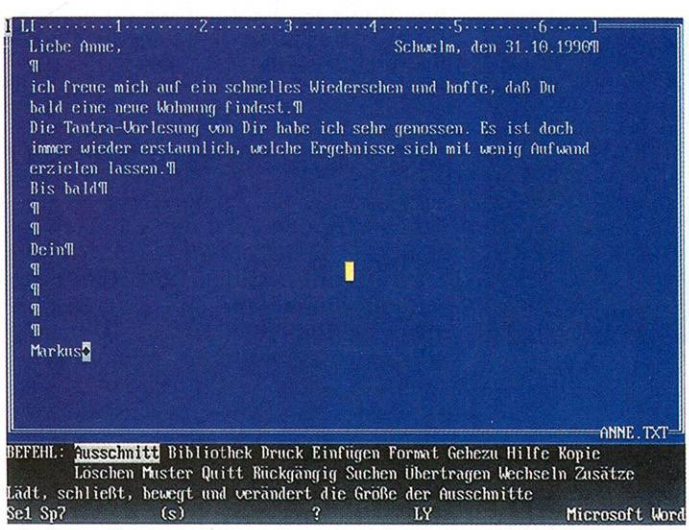


Bild 4. Das Textverarbeitungssystem MS-Word setzte mit seiner einfachen Benutzeroberfläche einen neuen Standard

```
Name.......... EINST-L.BAS

Sprache....... Basic

Grafikkarte..... jede

[RMØ] 1ØØØ '
[7E3] 1Ø1Ø 'Rekursions-Test
[ZMØ] 1Ø2Ø '
[PZ3] 1Ø3Ø ZAHL = 8: FAKULTAET = 1
[AØ7] 1Ø4Ø PRINT "Die Fakultät von"; ZAHL; "ist:";
[G32] 1Ø5Ø GOSUB 111Ø
[F73] 1Ø6Ø PRINT FAKULTAET
[941] 1Ø7Ø END
[RNØ] 1Ø8Ø '
[FH2] 1Ø9Ø 'Rekursion
[UMØ] 11ØØ '
[E45] 111Ø FAKULTAET = FAKULTAET * ZAHL
[ER2] 112Ø ZAHL = ZAHL - 1
[FM4] 113Ø IF ZAHL > Ø THEN GOSUB 111Ø
[BV1] 114Ø RETURN
```

Listing. Ein typisches Basic-Programm zeigt den leicht verständlichen Aufbau und die einfache Struktur

Astronomieprogramme

Sterne, Sonnen und Planeten

Dieses Thema ist so alt wie die Welt, und es beschäftigt selbst in der modernen Computerwelt viele Hobby-Astronomen. Die Shareware gestattet diesen »modernen Himmelsstürmern« mit zahlreichen Programmen und Utilities einen Blick in die Sternenwelt. Dieser Beitrag widmet sich zwei besonders interessanten Programmen.

Gute Astronomieprogramme sind nicht nur in der Shareware, sondern auch auf dem kommerziellen Markt gefragt, denn der nächtliche Sternenhimmel fasziniert nicht nur gelernte Astronomen. Ohne geeignetes Gerät und je nach Wolkenbedeckung sowie atmosphärischen Verhältnissen ist die Sicht in den heimatlichen Sternenhimmel häufig schlecht. Daher ist es wichtig, daß in einem Astronomieprogramm auch Funktionen implementiert sind, mit denen Sie bekannte Sterne direkt auf den Bildschirm holen, die sichtbaren Sterne heranzoomen oder auf die wichtigsten begrenzen können. Natürlich ist es unabdingbar, daß bei der Standortauswahl eine Liste zur Verfügung gestellt wird, die reichhaltig ist und rund um den Globus reicht. Der Anspruch an Gravitationssimulator-Programme umfaßt die Simulation der möglichen Planeten-/Sonnenkonstellationen, wobei Masse, Ablaufgeschwindigkeit und Gravitationskonstante einstellbar sind. Die hier präsentierten Sharewareprogramme »Gravitiy« und »Skyglobe« werden diesen Ansprüchen bis auf einen gerecht: Die Standortliste bei Skyglobe ist begrenzt. Bezugsquelle, Registrierungspreis und Grafikkartenanforderungen der beiden Programme finden Sie in der abgebildeten Tabelle.

Skyglobe – Sternenatlas

Skyglobe weckt Erinnerungen an die Entdeckungsreisen der »Enterprise«: Den Weltraum mit seinen unendlichen Weiten. Wir schreiben das Jahr 1991. Zu fernen Sternen und Galaxien kann man trotz hohem Technologiefortschritts noch lange nicht reisen. Auch wenn es sich die Weltall-Fans noch so sehr wünschen. Ein kleiner Trost auf dem Weg dahin bietet Skyglobe (Bild 1). Dieser elektronische Himmelsatlas bringt die Sterne in eindrucksvoller Art und Weise auf den »heimatlichen« PC-Bildschirm. Das Programm beinhaltet die Koordinaten von 3250 Sternen. Voraussetzung für den Betrieb ist eine Grafikkarte, die kompatibel zum Hercules-, CGA- oder EGA-Modus ist. Die Darstellungsqualität ist bei allen Grafikmodi ordentlich; der EGA-Modus mit 16 Farben läßt

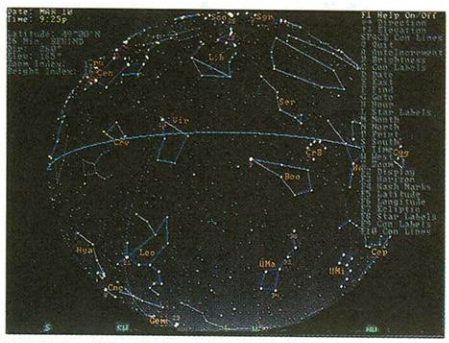


Bild 1. »Skyglobe« ist ein elektronischer Sternenatlas mit großem Funktionsumfang

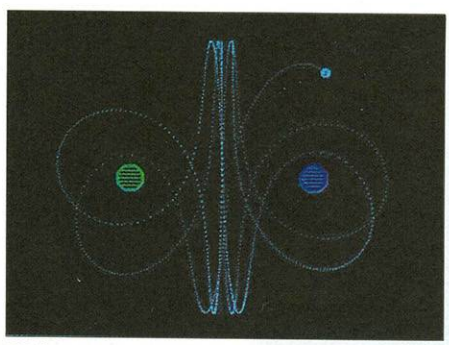


Bild 2. Mit »Gravity« lassen sich Planeten-Orbits simulieren, wie diese Schmetterlingskonstellation

nichts zu wünschen übrig. Trotz umfangreicher Funktionen beansprucht Skyglobe lediglich 300 KByte Speicherkapazität – das Programm kann problemlos auf einem 360-KByte-Diskettenlaufwerk betrieben werden. Nach Programmstart werden Sie zunächst aufgefordert, Ihren Standort anzugeben. Die Darstellung der Sterne am Bildschirm erfolgt dann exakt so, wie Sie sie am nächtlichen Himmel sehen. Die Standortauswahlliste beinhaltet zahlreiche Städte rund um den Globus. Für die BRD ist die Anwahl von Frankfurt aus am geeignetsten. Wer sich für 15 Dollar registrieren läßt, erhält vom Programmautor eine erweiterte Auswahlliste mit dem eigenen Wohnort beziehungsweise der nächstgelegenen Stadt. Die Darstellung der Sternenkarte am Bildschirm erfolgt gemäß der aktuellen Zeit der PC-Uhr. Mit fortlaufender Zeit bewegt sich der sichtbare Ausschnitt am Bildschirm automatisch weiter. Die Bedienung von Skyglobe erfolgt ausschließlich über die Tastatur. Die Funktionsbelegung der Tasten kann am rechten Bildschirmrand jederzeit per F1-Taste ein- oder ausgeblendet werden. Am linken Bildschirmrand finden Sie Informationen bezüglich aktueller Zeit, Breitengrad und der eingestellten Programmkonfiguration. Auch diese Informationen können Sie nach Belieben ausblenden, so daß ausschließlich der Sternenhimmel zu sehen ist. Über die Pfeiltasten läßt sich der sichtbare Ausschnitt verschieben, und so können Sie auch Sterne betrachten, die zur aktuellen Zeit am Himmel nicht zu sehen sind. Daß Skyglobe einer papierenen Sternenkarte weitaus überlegen ist, zeigt sich schnell. Je nach Wolkenbedeckung und Atmosphärenverhältnissen sind nicht immer alle Sterne am nächtlichen Himmel zu sehen. Per Tastendruck können Sie nun in 16 Stufen einstellen, welche Sterne Sie betrachten wollen. Bei höchster Stufe erscheinen alle sichtbaren Sterne, bei kleinster nur jene mit größter Helligkeit.

Auf Tastendruck können Sie auch die Verbindungslinien der Sternenbilder ein- und ausschalten. Eine Zoomfunktion für die nähere Betrachtung der gewählten Sterne ist ebenfalls vorhanden.

300 Sterne auf Tastendruck

Zum Auffinden einer der 300 wichtigsten gespeicherten Sterne aktivieren Sie die Findfunktion. Am oberen Bildrand erscheint dann eine Liste sämtlicher Sterne, aus der Sie den gewünschten auswählen. Nach Anwahl dreht Skyglobe die Sterndarstellung automatisch so, daß der gesuchte Stern in die Kartenmitte rückt. Zusätzlich wird sein Name eingeblendet. Der aktuelle Bildschirminhalt kann jederzeit durch Druck auf die P-Taste zu Papier gebracht werden. Hierfür wird lediglich ein Epson-kompatibler Drucker

vorausgesetzt. Skyglobe wurde für höchstmögliche Ausführungsgeschwindigkeit komplett in Assembler programmiert.
Abgesehen vom fehlenden Standort (insofern Sie nicht in Frankfurt wohnen) wird Skyglobe in der Sharewareversion mit vollem Funktionsumfang geliefert. Hobby-Astronomen erhalten mit Skyglobe ein fantastisches Werkzeug, das mit viel Liebe zum Detail programmiert wurde. Die englische Dokumentation enthält neben den Funktionsbeschreibungen auch eine Liste all der Sterne, die von der Skyglobe-Findfunktion unterstützt werden.

Name	Funktion	Grafikkarte	Registrierung	Bezugsquelle[1)]
Skyglobe	elektronischer Sternenatlas	Hercules, CGA, EGA	15 Dollar	DSB
Gravity	Gravitationssimulator	Hercules, CGA, EGA, VGA	10 Dollar	DSB

[1)]Bezugsquelle:
DSB = Deutsche Software Bibliothek, Verlag Nane Jürgensen, 8038 Gröbenzell

Wissenswertes über Skyglobe und Gravity auf einen Blick

Gravity-Gravitationssimulator

Gravity ist ein Gravitationssimulator, mit dem Sie die Orbits sämtlicher im Universum gegebenen Planeten-/Sonnenkonstellationen simulieren können. Das Programm unterstützt alle Grafikkarten von Hercules bis VGA. Die englische Dokumentationsdatei beschränkt sich auf 4000 Zeichen Umfang. Mehr wird auch nicht benötigt, da sich Gravity dank Pull-down-Menüs und Hilfefunktion einfach handhaben läßt. Für erste Betrachtungen und Experimente stehen mehrere Beispielkonstellationen zur Auswahl. Bild 2 zeigt eine sogenannte Schmetterlingskonstellation, bei der sich ein Planet in der Anziehungskraft zweier Sonnen bewegt (ausgehend vom abgebildeten Status wird er noch ein paar Runden drehen und dann in eine der beiden Sonnen stürzen). Die Bewegung der Himmelskörper am Bildschirm erfolgt realistisch mit Beschleunigungsmomenten. Über den integrierten Editor konfigurieren Sie die Masse und Ausgangsgeschwindigkeit eines jeden Planeten individuell.

Auf diese Art und Weise lassen sich auch andere Simulationen erzeugen, die selbstverständlich gespeichert und später wieder abgerufen werden können. Im Utility-Menü nehmen Sie globale Einstellungen bezüglich Gravitationskonstante und Zeitablauffaktor vor. Ist die eigens gebastelte Konstellation sehr umfangreich, empfiehlt es sich, in einen geringer auflösenden Grafikmodus umzuschalten, um die Ausführungsgeschwindigkeit zu erhöhen. Mathematische Coprozessoren werden unterstützt. Gravity ist ein interessantes Utility zum Experimentieren. Darüber hinaus eignet es sich auch hervorragend für astronomische Schulungszwecke. Der Funktionsumfang ist in der Sharewareversion nicht eingeschränkt.

(Michael Nickles/ba)

DMV
PROGRAMMIERER-TOOLS

GraphBas PC

BGI-Grafikerweiterung für Microsoft Quick Basic & Turbo Basic

GraphBas PC erschließt Ihrem Basic-Compiler bisher ungeahnte Grafikwelten.

Was Turbo Pascal und Turbo C berühmt gemacht hat, gilt jetzt auch für Basic:

- Volle Nutzung der BGI-Vektorfonts
- Editor für eigene Schriften, Logos und Pictogramme
- 3D-Balken- und Tortendiagramme, kinderleicht aufgerufen
- Farben, Füllmuster und Figuren mit Assemblerroutinen
- einfachste Handhabung durch Quellcodes-Einbindung
- Unterstützung aller Grafikkarten auf dem PC

GraphBas PC je DM 99,–*
BGI-Grafikerweiterung für Microsoft Quick oder Turbo Basic

MaskEdit PLUS 5.0

Professioneller Maskengenerator liefert Quellcodes Ihrer Wahl – Statt Listings tippen Menüpunkte auswählen.

Schnelle Bildschirmmasken für Ihre Programme:

- hierachische Menüs
- verschiedene Ausgabefenster
- 8 Rahmentypen, 16 Farben, Schattenwurf
- ASCII-, dBase-, Binär- und ANSI-Ausgabeformate als Standard
- optimale Treiber für die Ausgabesprache in Ihrer Wahl
- alle Treiber im Quelltext
- einfaches, aber effizientes Arbeiten mt über 50 Funktionen
- Komfort durch kontextbezogene Hilfen

Zusatztreiber-Auswahl: Turbo Pascal 3/4/5, Turbo C und Microsoft C, Topspeed-Modula-2, Turbo basic, Quick Basic

MaskEdit Plus, ein Zusatztreiber DM 99,–*
MaskEdit Plus, zwei Zusatztreiber DM 149,–*
MaskEdit Plus, drei Zusatztreiber DM 199,–*

* Unabhängig von der Anzahl der bestellten Produkte berechnen wir für das Inland DM 4,– und für das Ausland DM 6,– Porto und Verpackung. Bitte benutzen Sie die Bestellkarte.

DMV-Verlag • Postfach 250 • 3440 Eschwege

DMV Daten- und Medienverlag

Troubleshooting-Werkstatt

»Error-Handling«: Dem Rat folgt die Tat

Es soll unter den Computerbenutzern Leute geben, für die es bei Problemen mit ihrer Hard- und Software erst richtig interessant wird. Falls Sie nicht zu diesem Kreis gehören, zeigen wir Ihnen, wie Sie auftretende Fehler beheben oder diese grundsätzlich vermeiden.

Die unfreundlichste Meldung, die ein PC von sich gibt, ist immer noch die Errormeldung. Diese tritt in den unterschiedlichsten Varianten auf – manchmal bemüht sich das Gerät nicht einmal mehr, den Besitzer mit dieser Unfreundlichkeit zu konfrontieren und stürzt ab.

Nicht immer muß der Grund der Fehlermeldung gleich in einem Hardwaredefekt zu suchen sein. Vielleicht erweist sich die Ursache letztlich nur als eine falsche Einstellung der Hardware – vielleicht haben Sie beim Aufruf eines Programms nicht die richtigen Parameter übergeben oder die vorhandene Gerätekonfiguration ist für die gestellten Aufgaben ungeeignet?

Der Anfänger ist meist nicht in der Lage, dem Problem allein mit Hilfe der Handbücher nachzugehen. Daß selbst Profis mitunter ins Schwitzen kommen, werden Sie feststellen, wenn Sie sich bei ihnen Rat und Hilfe holen wollen.

Es ist ratsam, beim Kauf von Hard- und Software größten Wert auf komplette Handbücher zu legen, am besten in Deutsch verfaßte. Achten Sie auch darauf, daß die Unterlagen mit dem gekauften Produkt übereinstimmen. Gerade bei den »Handheftchen« von No-name-Fabrikaten erleben Sie nicht nur ein völlig neues Sprachgefühl, sondern auch produktspezifische Wunder – etwa ein Handbuch zum Scanner, obwohl Sie einen PC vor sich haben.

Natürlich ist es undenkbar, gegen alle Tücken der Technik gewappnet zu sein oder allen nur erdenklichen Fehlern im voraus zu begegnen.

Fehlertypen: Neben den mitunter vorkommenden Hardwarefehlern sind es vor allem Anschaffungs-, Bedienungs-, Interdependenz- und Softwarefehler, die Errormeldungen zeitigen.

- Hardwarefehlern ist manchmal zu begegnen. Einen kaputten Monitor können Sie ohne Datenverlust ersetzen, eine »gestrandete« Festplatte nicht mehr.

Hardwarefehler kosten mindestens die Ersatzteilkosten der entsprechenden Komponente. Vorausgesetzt, Sie haben Ihr Gerät in vorbildlicher Weise gelagert und behandelt, sind Hardwarefehler hersteller-, zufalls- oder altersbedingt und manchmal reparabel.

```
shell=c:\command.com c:\ /p /e:1024
files=120
buffers=20
lastdrive=K
history=on, 256, off
country=049,,c:\drdos\country.sys
device=C:\himem.sys
device=c:\windows\mouse.sys /Y
device=C:\windows\smartdrv.sys 768
* device=C:\windows\ega.sys
```

Bild 1. Eine korrekte »config.sys« kann einiges dazu tun, damit Ihr PC fehlerfrei läuft – hier etwa die Treiber für »Windows« (»Smartdrv«, »Himem«) und »Ensemble« (»Files«) laden

- Softwarefehler: Kein Softwarehaus ist in der Lage, fehlerfreie Produkte zu entwickeln – aber die bekanntgewordenen Fehler können in der Folgeversion eliminiert werden. Begegnen Ihnen solche Fehler, sollten Sie diese beim Hersteller »registrieren« lassen. Gegen Softwarefehler hilft meist ein Update auf dieses Produkt. Mitunter ist aber die Lösung selbst herbeizuführen, da Software in vielen Fällen auf Ausnahmesituationen nicht vorbereitet ist – etwa wenn die Diskette voll, der Drucker nicht angeschaltet oder der Maustreiber nicht geladen ist. Selbstverständlich ist es kein Softwarefehler, wenn Sie eine notwendige Konfigurationsdatei eigenhändig gelöscht haben. Aber es ist ein Softwarefehler, dieses Manko nicht mit der entsprechenden Meldung zu quittieren.

- Anschaffungsfehler sind solche, die bei Inkompatibilitäten enden: Was kann etwa Ihre VGA-Karte dafür, daß Sie ihr einen Hercules-Monitor aufzwingen? Oder was kann das EGA-Malprogramm dafür, daß Sie Ihre CGA-Karte nicht ersetzen? Solche Fehler auszumerzen, kostet Geld – Ihr Geld (oder das der geschätzten Firma).

- Konfigurations- oder Interdependenzfehler können hard- oder softwareseitig auftreten: So kann beispielsweise ein Programm abstürzen, weil es nicht korrekt installiert wurde, oder weil ein residentes Programm den Arbeitsspeicher streitig macht. Oder der Rechner stürzt ab, weil er im Turbomodus fährt. Die Lösung ist oftmals einfach, wenn der Grund bekannt ist. Vor allem softwareseitig ist durch stufenweise Zurücknahme der Treiber- oder TSR-Programme schnell das verursachende Element gefunden.

- Meistens gibt Ihnen das Softwareprodukt die Chance, es korrekt und ohne Bedienungsfehler zu handhaben. Aber das Handbuch war mal wieder zu dick, die Konzentration schon beim Teufel oder die Hilfe in Englisch. Natürlich ist es ein Bedienungsfehler, wenn Sie eine später wieder benötigte Datei löschen. Meist genügen aber ein paar Tricks, und die »Errors« gehören der Vergangenheit an. Beispielsweise eine korrekte Nachinstallation oder das Unterlassen eines residenten Programmstarts. Oder Sie haben Ihre Adreßverwaltung resident geladen, was den Arbeitsspeicher zur Arbeitsverweigerung veranlaßt.

Andere Bedienungsfehler beziehen sich auf die Hardware: Sie vergessen etwa, den Drucker mit »Online« zu versehen oder die Tastatur anzuschließen. Sie haben das Druckerkabel auf »Lpt2« gesteckt, das Programm aber angewiesen, auf »Lpt1« zu drucken.

Welcher Fehlertyp gegebenenfalls letztendlich vorliegt, ist nur in genauer Kenntnis der Geräte- und Softwarekonfiguration zu eruieren. Nichtsdestotrotz sind häufige Fehlertypen zu entdecken. Diese typischen Fehler betreffen selbstverständlich auch solche, die nicht vom Anwender verursacht sind. Beispielsweise ist die serielle Schnittstelle bei bestimmten Herstellern vorzugsweise schon nach wenigen Stunden kaputt. In dieser Werkstatt kann es aber nur um Fehlerarten gehen, die vom Anwender ursächlich beeinflußt und beeinflußbar sind.

Aufgrund der unterschiedlichen Hard- und Softwarekonfiguration sind natürlich keine eindeutigen Angaben der Fehlermeldungen möglich. Es wurden deshalb – der allgemeinen Verständlichkeit wegen – deutsche Umschreibungen gewählt.

Ein PC besteht bekanntlich aus vielen Komponenten, die erst einmal miteinander vertraut gemacht werden müssen. Dazu gibt es zum einen entsprechende Schnittstellen – entweder hardware- oder treibermäßig realisiert. Herzstücke dieser Steuerung sind BIOS, Betriebssystem und CMOS-RAM, das die Setup-Daten behält. Wenn bei den Hardwarefehlern die

XT-Computer vernachlässigt werden, so liegt das daran, daß deren Hardware nicht per Programm zu konfigurieren ist. Dies muß nicht unbedingt ein Nachteil sein, was Ihnen sicher mancher NEAT-Board-Besitzer bestätigen kann, der nach diversen Experimenten mit den vielfältigen Einstelloptionen vor einem schwarzen Bildschirm saß und als letzte Rettung die Pufferbatterie für den Konfigurationsspeicher kurzschloß, um die falschen Einstellungen zu löschen. Hätte er sein Handbuch gelesen, wäre ihm vielleicht aufgefallen, daß man durch Drücken der Taste [Einfg] beim Starten die Grundeinstellungen wieder aktivieren kann. Deshalb sollten Sie in jedem Fall immer erst zum Handbuch greifen, bevor Sie Defekte suchen, die gar keine sind.

● Computer startet nicht von Diskette: Obwohl Sie eine Systemdiskette eingelegt haben, will Ihr AT diese nicht für den Startup benutzen. Hier sind zwei Fehlerquellen denkbar:

○ Ihre Diskette ist entweder defekt oder gar keine Systemdiskette. Abhilfe schafft eine (neu hergestellte) Systemdiskette.

○ Ihr AT hat das Diskettenlaufwerk falsch konfiguriert. Beispielsweise nimmt er an, er habe ein 360-KByte-Laufwerk, während Sie eine 1,2-MByte-Systemdiskette eingelegt haben. Abhilfe kann hier entweder eine 360-KByte-Systemdiskette leisten oder – auf längere Sicht erfolgreicher – ein korrigiertes Setup.

● Falsche Setup-Daten beim AT: Leider konnten sich die BIOS-Hersteller bis heute nicht auf einen einheitlichen Aufruf einigen. In den meisten Fällen ist es fest im EPROM integriert, und Sie erreichen es entweder über die Taste [Entf], die während des Hochfahrens gedrückt werden muß, oder über die Kombination [Strg-Alt-Esc]. Bei manchen Computern benötigen Sie dazu eine spezielle Software, mit deren Hilfe die notwendigen Einstellungen vorgenommen werden – etwa »setup.com«. Dies ist oftmals bei Boards mit NEAT-Technologie der Fall. Dort befindet sich im EPROM meist ein zusätzliches Diagnoseprogramm, und das Programm für die Einstellungen der NEAT-Chips hat deshalb keinen Platz mehr.

Beachten Sie auch, daß bis zum Setup noch kein Tastaturtreiber geladen wurde und die amerikanische Tastaturbelegung aktiv ist. Deshalb sind noch [Z] und [Y] vertauscht – bei der Abfrage (Y/N) ist also mit [Z] zu bestätigen. Ferner stimmen die Satzzeichen noch nicht mit der Tastatur überein. Für die spezielle Handhabung des eigenen Setup-Programms schlagen Sie in Ihren Handbüchern nach.

● Uhrzeit oder Datum stimmen nicht: Da Computer der XT-Reihe in der Grundversion über keine Hardware-Uhr verfügen (nicht so beim Schneider-PC 1640), benötigen Sie zur Einstellung der Uhrenkarte in diesen Fällen ein individuelles Programm des jeweiligen Herstellers. Ab dem 286-AT gehört die Uhr zum Standard. Ihre Einstellung erfolgt im Setup-Menü und ist von der jeweiligen Version abhängig. Bei den meisten Geräten können Sie die Zeit auch über die MS-DOS-Befehle »time« und »date« korrigieren, die dann ins CMOS-RAM übernommen werden. Die falschen Zeitangaben können ihre Ursache darin haben, daß die Batterien zur Neige gehen – sie also nicht mehr fähig sind, die Daten »aufhalten« zu können.

Bei einem 8086- oder 8088-PC ist es normal, daß die Zeitangaben nicht korrekt sind – schließlich haben sie ein Uhrwerk, das nur im eingeschalteten Zustand läuft und ohne genaue Zeitangaben stets am 1.1.1980 beginnt. Hier hilft nur, die beiden Befehle

```
time
date
```

in die »autoexec.bat« mitaufzunehmen und bei jeder Sitzung von Hand zu aktualisieren.

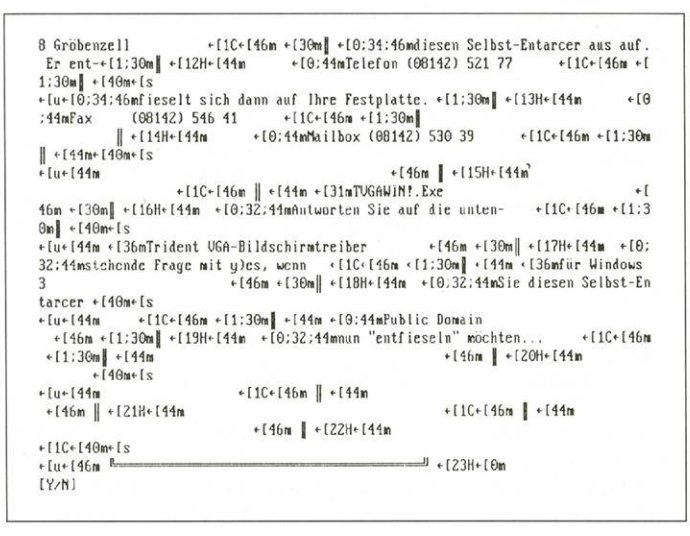


Bild 2. Dies ist kein Hexdump, sondern eine Textdatei, die den »ansi.sys« schmerzlich vermißt

● Festplatte fährt nicht mehr hoch: Dies könnte mehrere Ursachen haben. Beispielsweise:

○ Kommandointerpreter fehlt oder ist fehlerhaft: Dies bedeutet, daß die Datei »command.com« nicht gefunden oder nicht gelesen werden kann. Diese sollte sich normalerweise im Hauptverzeichnis der Festplatte (c:\) befinden. Legen Sie deshalb eine Systemdiskette in das Laufwerk A und starten Ihren Computer neu. Danach wird die Datei »command.com« mit dem Befehl

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

wieder auf die Festplatte geschrieben. Nach dem erneuten Starten des Computers sollte sich das Betriebssystem ohne Schwierigkeiten laden lassen.
Wichtig ist dabei jedoch, daß die beiden unsichtbaren Systemdateien (bei MS-DOS sind dies »msdos.sys« und »io.sys«) sowie »command.com« von der gleichen DOS-Version sind. Andernfalls werden Sie mit einer Fehlermeldung darauf aufmerksam gemacht.
Abhilfe schafft auf Dauer der Shell-Befehl, in dem Sie angeben, wo sich Ihr Kommandoprozessor befindet:

```
shell=c:\command.com
```

beugt etwa dem Fall vor, daß Sie von Laufwerk A aus gestartet und auf der Festplatte Programme gefahren haben.

● Kein Betriebssystem gefunden: Wenn dieser Fehler auftaucht, wird meist die Brachialmethode ergriffen und die Festplatte neu formatiert. In der Regel sind sämtliche Daten verloren. Oft ist aber eine Neuformatierung unnötig, wenn Sie zuerst versuchen, alle eventuellen einfachen Fehler auszuschalten. Gehen Sie dazu folgendermaßen vor:

○ Starten Sie den Computer mit einer Systemdiskette und rufen als erstes das MS-DOS-Programm »fdisk« auf. Damit läßt sich feststellen, ob die Festplatte noch angesprochen werden kann. Ist dies der Fall, lesen Sie beim dritten Punkt weiter.

○ Wenn »fdisk« meldet, daß es keine Festplatte finden oder nicht darauf zugreifen kann, wäre es im Interesse Ihrer Daten und Programme auf jeden Fall besser, auf die Hilfe eines Fachmanns zurückzugreifen. Sie können sich zwar selbst an die Fehlersuche wagen, dazu sollten Sie aber über die dafür notwendigen Hardwarekenntnisse verfügen.

○ Handelt es sich bei dem Computer um einen 286-AT oder höher, kann die Ursache im einfachsten Fall ein falscher Eintrag im Konfigurationsspeicher sein. Um dies zu überprüfen, brauchen Sie jedoch die genauen Daten Ihrer Festplatte – also die Anzahl der Schreib-/Leseköpfe, der Zylinder und der Sektoren.
Für die gängigsten Festplattentypen gibt es Festvorgaben, die im Setup

eingestellt werden. Dies ist nur dann der Fall, wenn es sich um Festplatten mit dem MFM-Aufzeichnungsverfahren handelt. Die Schwierigkeiten beginnen jedoch bei Festplatten mit RLL-, ESDI-, SCSI- oder IDE- (AT-Bus-) Verfahren, die nicht mit 17 Sektoren arbeiten.

Ein paar Hinweise sind jedoch auf alle Platten und Controllertypen anwendbar: Bei der Benutzung von zwei Festplatten ist auf die richtige Adressierung der einzelnen Festplatten zu achten. Diese erfolgt meist mit Jumpern oder Dipschaltern.

○ Hier gilt wiederum für MFM- und RLL-Festplatten folgendes:

Entfernen Sie bei einem Laufwerk den sogenannten »Pull-Up-Widerstand«: Bei Seagate-Platten liegt er beispielsweise in der Nähe der Steckeranschlüsse und ist ockerfarben.

Je nach Aussehen des 34adrigen Kabels müssen die Laufwerke verschieden adressiert werden: Bei einem gedrehten Kabel werden beide Laufwerke auf »Drive Select 1«, bei einem normalen Kabel das zweite Laufwerk auf »Drive Select 2« gestellt.

Laufwerk 1 kommt immer an das Kabelende, Laufwerk 2 an den mittleren Stecker.

○ SCSI-Laufwerke werden in der Regel im Setup nicht angemeldet. Das Vorhandensein einer Festplatte wird beim Starten durch den Controller erkannt. Sie sind zwar in der Lage, eine SCSI-Festplatte auch neben einer normalen MFM-Platte zu betreiben, jedoch garantieren die Hersteller nicht für eine fehlerfreie Funktion solcher Kombinationen.

○ Bei ESDI-Festplatten werden die Daten der Festplatte auf einer eigenen Systemspur der Platte selbst gespeichert. Normalerweise werden diese im Setup mit der Einstellung 1 angemeldet. Das BIOS weiß nun, daß eine Festplatte vorhanden ist. Die genauen Daten und Parameter werden dann von der Festplatte gelesen.

Interrupt	belegt durch den Baustein (Computertyp)
IRQ 0	System Timer
IRQ 1	Tastatur
IRQ 2	Cascade (nur AT)
IRQ 3	Seriell 2
IRQ 4	Seriell 1
IRQ 5	Festplatte
IRQ 6	Floppy
IRQ 7	Drucker
IRQ 8	Echtzeituhr (nur AT)
IRQ 9	frei (nur AT)
IRQ 10	frei (nur AT)
IRQ 11	frei (nur AT)
IRQ 12	frei (nur AT)
IRQ 13	Coprozessor (nur AT)
IRQ 14	Festplatte (nur AT)
IRQ 15	frei (nur AT)

Tabelle 1. Die 15 Hardware-Interrupts mit der üblichen Belegung

```
      SPEICHERBELEGUNGS-TABELLE
   ▪ Mit dem »Commander« und mit
     ihm gestartetes »Snapsho«

... Länge Hex+Dez   Name
    =====  ======  ==================
... 188F0 = 100592  System
... 01390 =   5008
...
... 000A0 =    160
... 032A0 =  12960  C:\NC.EXE
... 000A0 =    160
... 00100 =    256
... 000A0 =    160  Freigegeben
... 01350 =   4944  Freigegeben
... 000A0 =    160
... 00B80 =   2944  C:\SNAPSHO.COM
... 01390 =   5008
=====================================
...
Nächste Programm-Adresse: 2518
Freier RAM-Speicher     : 314640 Bytes

      SPEICHERBELEGUNGS-TABELLE
   ▪ Mit deinstalliertem »Commander«

... Länge Hex+Dez   Name
    ===== ======  ==================
... 188F0 = 100592  System
... 01390 =   5008
...
... 000A0 =    160  Freigegeben
... 04870 =  18544  Freigegeben
... 000A0 =    160
... 00B80 =   2944  Name unbekannt
=====================================
...
Nächste Programm-Adresse: 23DE
Freier RAM-Speicher     : 317780 Bytes

      SPEICHERBELEGUNGS-TABELLE
   ▪ Mit von neuem geladenen »Commander«

... Länge Hex+Dez   Name
    =====  ======  ==================
... 188F0 = 100592  System
... 01390 =   5008
...
... 000A0 =    160
... 000A0 =    160
... 00100 =    256
... 000A0 =    160  Freigegeben
... 04600 =  17920  Freigegeben
... 000A0 =    160
... 00B80 =   2944  Name unbekannt
... 032A0 =  12960  C:\NC.EXE
... 01390 =   5008
=====================================
...
Nächste Programm-Adresse: 2843
Freier RAM-Speicher     : 306530 Bytes
```

Bild 3. Residente Programme sind ein steter Gefahrenherd für eine PC-Sitzung – hier die Dokumentation einer Speicherfresserei durch diese Nesthocker: Beim zweiten Commander-Start wurde die Speicherlücke nicht wieder gefüllt

○ Eine Neuheit und zugleich Ausnahme bilden die AT-Bus-Festplatten: Sie benötigen keinen Controller im alten Sinn, diesen haben Sie bereits auf der Festplatte selbst integriert, benötigt wird nur eine Karte, die den Datenbus des PC an die Festplatte anpaßt. Dadurch erklärt sich auch der niedrige Anschaffungspreis. Mittlerweile sind bereits Festplatten mit über 100 MByte Speicher für unter 1000 Mark erhältlich.

Zu beachten ist, daß die LED-Anzeige für den Zugriff beispielsweise bei Seagate-Festplatten erst mit Hilfe eines Jumpers auf externe Anzeige umgestellt werden muß. Außerdem ist die eingebaute LED dann an einem Anschlußbein durchzutrennen. Dies ist dann nötig, wenn Sie die Zugriffe durch eine (meist rote) LED an der Frontseite des Computers anzeigen lassen wollen, die am Controller angeschlossen wird.

○ Konnte »fdisk« ohne Fehlermeldung gestartet werden, lassen Sie sich die Partitionstabelle anzeigen. Wichtig ist dabei, daß die Partition für das Laufwerk C auch aktiv ist. Dies erkennen Sie am Buchstaben »A« in der Spalte »Status«. Fehlt dieser, ist sie nicht als aktive Boot-Partition gekennzeichnet und muß erst aktiviert werden, da der Computer sonst nicht weiß, auf welchem Laufwerk er das Betriebssystem findet.

○ Sollte sich die Festplatte nun immer noch weigern, ihren Dienst zu versehen, kann dies auch am zerstörten Bootsektor liegen. Hier können Sie versuchen, das Betriebssystem komplett neu auf die Festplatte zu übertragen. Dies gelingt Ihnen durch:

```
sys c:
```

Dieses Programm schreibt den Bootsektor neu und kopiert die Systemdateien auf das Laufwerk C.

○ Waren alle Versuche bisher erfolglos, hilft nur noch eine Neuformatierung. Wenn ein Zugriff auf die Platte erlaubt ist, können Sie zuvor die darauf befindlichen Daten auf Disketten (oder eine zweite Festplatte) sichern. Danach bleibt allein noch

```
format c: /s
```

übrig, und Ihr Computer muß sich nach einem anschließenden Reset wieder mit der Aufforderung nach Eingabe des Datums und der Uhrzeit melden. Nun schreiben Sie noch die geretteten Daten auf die Festplatte zurück.

● Diskette läßt sich nicht lesen: Auch in diesem Fall kann die Ursache an einem falschen Eintrag im Konfigurationsspeicher (CMOS-RAM) liegen. Überprüfen Sie, ob die Eintragungen im Setup mit den vorhandenen Laufwerken übereinstimmen. Ist alles korrekt, können Probleme folgende Ursachen haben:

○ 3 1/2-Zoll-Laufwerk (1,44 MByte): Bei älteren BIOS-Versionen funktionieren 1,44-MByte-Laufwerke noch nicht. Hier hilft nur der Austausch gegen eine neue BIOS-Version. Auch ältere Diskettencontroller können noch keine 1,44-MByte-Laufwerke ansprechen. Auch hier rettet Sie nur eine Neuanschaffung.

Da die Laufwerke selbständig erkennen, ob es sich um HD-Disketten handelt, können auch nur solche auf 1,44 MByte formatiert werden. Bei DD-Disketten ist dies nicht möglich.

Schwierigkeiten bereitet auch das Formatieren von 720-KByte-Disketten in 1,44-MByte-Laufwerken. Lösung bringt eine Stapeldatei. Diese kann den Namen »form720.bat« tragen und muß folgende Zeile enthalten (Laufwerk B, 3 1/2 Zoll):

```
format b: /n:9 /t:80
```

Dadurch ersparen Sie sich die Eingabe und das Merken der Parameter. Bei DR-DOS genügt

```
format b: /f:720
```

○ 3 1/2-Zoll-Laufwerke (720 KByte) bereiten in der Regel keinerlei Schwierigkeiten mit Controllern. Sie lassen sich auch auf XT-Computern

problemlos einbauen. Zum Formatieren benötigen Sie die gleiche Stapeldatei, wie beim 1,44-MByte-Laufwerk beschrieben. Sollte Ihr Controller dies jedoch nicht verarbeiten, müssen Sie ihm mitteilen, daß das Laufwerk B 720-KByte-Disketten verarbeiten kann. Die Datei »config.sys« wird dafür um die Zeile

```
device=c:\dos\driver.sys /d:1 /f:2
```

erweitert. Angenommen, Sie haben schon die Diskettenlaufwerke A und B (5 1/4 und 3 1/2 Zoll) sowie die Festplatte C. Dann wird durch den Eintrag ein »virtuelles« Laufwerk D angelegt. Auf diesem können Sie nun 720-KByte-Disketten formatieren. Lesen ist aber auch auf Laufwerk B erlaubt.

○ 5 1/4-Zoll-Laufwerke (360 KByte): Wollen Sie ein Laufwerk diesen Typs in einem 286-PC oder höher einbauen, kann es Probleme mit der richtigen Erkennung des Diskettenformats geben. Warum ein altes 360-KByte-Laufwerk in einen AT einbauen? Auf den ersten Blick macht dies keinen Sinn, da ja auch ein 1,2-MByte-Laufwerk dieses Format lesen und schreiben kann. Vielleicht besitzen Sie jedoch noch jede Menge alter 360-KByte-Disketten. Diese lassen sich mit einem 1,2-MByte-Laufwerk oft nicht mehr vollständig lesen, da hier der Schreib-/Lesekopf wesentlich schmäler ist. In diesem Fall leistet das 360-KByte-Laufwerk weiterhin gute Dienste.

Diskettenlesefehlern auf der Spur: Nicht immer ist die Diskette schuld

○ 5 1/4-Zoll-Laufwerke (1,2 MByte): Dieser Laufwerkstyp gehört zur Standardausrüstung der AT-Serie. Das Laufwerk muß wiederum im Setup richtig eingestellt werden. Bei Verwendung von zwei Laufwerken ist auch hier auf die Art des Anschlußkabels zu achten. Normalerweise sind für den Anschluß 1 die Adern 10 bis 16 gedreht. Dies hat den Vorteil, daß beide Laufwerke gleich selektiert werden können. In diesem Fall ist dies »Drive Select 1«. Besitzen Sie ein ungedrehtes Kabel (teilweise bei Commodore), so muß Laufwerk A auf »Drive Select 0« und Laufwerk B auf »Drive Select 1« betrieben werden.

Achten Sie unbedingt darauf, daß die Drive-Select-Bezeichnungen nicht mit den Festplattenlaufwerken identisch sind. Seagate-Festplatten zählen in den Handbüchern von DS1 bis DS4, auf Diskettenlaufwerken von NEC und TEAC finden Sie die Bezeichnungen DS0 bis DS3.

Auf keinen Fall dürfen Sie die Anschlußkabel für Festplatte und Diskettenlaufwerk verwechseln. Beide sind zwar 34adrig, jedoch befinden sich die Drive-Select-Leitungen an unterschiedlichen Pins: Bei der Festplatte sind dies die Pins 26 bis 28 oder 25 bis 29, wenn die Masseleitungen mitgedreht werden.

Genauere Einzelheiten entnehmen Sie Ihren Handbüchern.

Wer schon einmal versucht hat, die Erweiterungssteckplätze seines Computers vollständig zu nutzen, wird festgestellt haben, daß es mit »Karte rein und funktioniert« in aller Regel nicht getan ist.

Dem stehen drei Punkte im Wege, die zuvor gelöst sein wollen. Dies sind

- der Interrupt, den die Karte benutzt,
- die Adresse, unter der sie angesprochen wird
- und der Speicherbereich, den sie belegt.

Für die Belegung der Interrupts (Tabelle 1) und Adressen (Tabelle 2) gibt es einen allgemeingültigen Standard: Eine zusätzliche Steckkarte muß in der Regel auf einen noch freien IRQ gelegt werden. Im Einzelfall werden Sie trotzdem um das Ausprobieren nicht herumkommen, da auch Mehrfachbelegungen denkbar sind. Keinesfalls können Sie dabei etwas zerstören. Das schlimmste, was passieren kann, ist, daß Ihr Computer nicht mehr startet – was aber durch Ausbau der betreffenden Karte wieder behoben wird. Oder es können Daten auf der Festplatte überschrieben werden. Treffen Sie deshalb auch hier Vorsorge, indem Sie vorab Ihre Daten sichern, was eigentlich vor allen Eingriffen in den PC selbstverständlich sein sollte.

● Grafikkarten: Die Vielzahl der unterschiedlichen Standards läßt eine genaue Angabe von Fehlerursachen nicht zu. Trotzdem seien zu jedem Grafikstandard ein paar grundsätzliche Hinweise gegeben:

○ Computer auf richtigen Modus stellen: Dabei sind Sie auf die Hilfe des Handbuches angewiesen. Bei Computern ab dem 286er muß entweder ein Jumper auf dem Motherboard eingestellt oder aber im Setup-Menü die Grafikkarte angegeben werden. Der Jumper erlaubt die Einstellung »Mono« oder »Color«. Das Setup-Programm unterscheidet die Farbkarten meist noch in »CGA« oder »EGA/VGA«. In der Praxis hat sich jedoch herausgestellt, daß die Grafikkarten normalerweise automatisch richtig erkannt werden und deshalb die Einstellung ohne Bedeutung ist.

```
@echo off
path c:\bat;c:\drdos;c:\;c:\tools
set temp=c:\windows\temp
keyb gr
saveall
\tools\cnslock n
verify off
prompt $p$g
cls
cd \bat
type ausw.txt
```

Bild 4. Die »autoexec.bat« ist nicht mehr als eine »Gedächtnisstütze« – Sie könnten diese Befehle auch alle von Hand eingeben

Schnittstelle	Adresse
Lpt1 (Hercules)	03BChex
Lpt1 (normal)	0378hex
Lpt2	0278hex
Com1	03F8hex
Com2	02F8hex
Com3	03E8hex
Com4	02E8hex

Tabelle 2. Die standardmäßigen Adressen für die parallelen und seriellen Schnittstellen

○ Print-Screen-Funktion in MS-DOS: Eine Hardcopy-Ausgabe des Bildschirms im Textmodus ist bei allen Grafikkarten zu erzeugen. Probleme bereiten dabei nur bestimmte Grafikzeichen, die je nach Zeichensatztabelle auch als Steuerzeichen dienen.

Sie sind auch in der Lage, von CGA-Grafikbildschirmen eine Hardcopy zu generieren. Dazu sollten Sie zuvor die beiden MS-DOS-Programme »graftabl« und »graphics« geladen haben.

○ Programme starten nicht – falscher Modus: Nicht jede Software, die im Grafikmodus arbeitet, unterstützt auch alle gängigen Auflösungen. Die Programme fragen jedoch normalerweise die aktuelle Auflösung ab und melden nötigenfalls Inkompatibilitäten. Beim überwiegenden Teil der Software läßt sich die vorhandene Grafikkarte bei der Installation einstellen. Dies ist aber leider, vor allem bei Spielen, des öfteren nicht der Fall. Achten Sie darum schon vor dem Kauf darauf, ob das gewünschte Programm auf Ihrer Grafikkarte lauffähig ist.

Eine Karte macht noch keinen PC: Probleme mit Zusatzkarten und Geräten

● Monitore: Hierbei ist als wichtigstes Kriterium zu nennen, daß der Monitor auf jeden Fall zur vorhandenen Grafikkarte passen muß. Wenn auch die meisten Monitorhandbücher (bis auf sehr wenige Ausnahmen) etwas kläglich im Umfang sind, so finden Sie darin normalerweise zumindest die Einstellregler und ihre Funktion.

Deshalb nur ein paar Hinweise, die für fast alle Monitore zutreffen und im Handbuch nur unzureichend beschrieben sind.

○ Öffnen Sie auf keinen Fall als Nichtfachmann Ihr Gerät. Der Monitor benötigt zum Betrieb Hochspannung, die im Kilovolt-Bereich liegt. Ein falscher Handgriff kann deshalb tödlich sein.

○ Bild läuft durch: Suchen Sie in diesem Fall an Ihrem Monitor den Regler für die Synchronisation. Ist an Ihrem Gerät keiner angebracht, hilft nur der Fachmann.

○ Bildgröße stimmt nicht: Auch dafür gibt es an den meisten Geräten entsprechende Regler. Testen Sie die Einstellung in allen Grafikmodi, und korrigieren Sie die Größe nötigenfalls.

○ Hercules-Monitor zeigt keine Helligkeitsunterschiede: Hier müssen Sie mit den zwei Reglern für Kontrast und Helligkeit das Bild so einstellen, daß die Textzeilen mit halber sich von denen mit voller Intensität unterscheiden.

● Modemkarte: Da das Modem über eine eigene serielle Schnittstelle (auf der Modemkarte implementiert) angesteuert wird, ist darauf zu achten, welche anderen Schnittstellen bereits vorhanden sind. Diese sind normalerweise »com1« und »com2«. In den meisten Fällen wird die Karte deshalb als »com3« installiert.

Ob dafür der IRQ4 (Tabelle 1) benutzt werden kann oder ein anderer, liegt an der Karte. Hier sollten Sie im Handbuch nachschlagen, welche Interrupts Sie verwenden können.

● Joystick-Karte: Der Einbau ist normalerweise mit keinen Schwierigkeiten verbunden, da hierbei kaum Überschneidungen von Adreßbereichen vorkommen. Wichtig ist jedoch, daß Sie nur eine Karte installiert haben. Sollten Sie beispielsweise schon über einen Gameport verfügen und noch eine Soundblaster-Karte einbauen, so müssen Sie auf jeden Fall den Gameport auf dieser Karte mit Hilfe des Jumpers ausschalten.

● Maus: Die zwei verbreitesten Betriebsarten sind die Microsoft- und Mouse-System-Maus. Besitzt Ihre Maus eine eigene Steckkarte, so darf diese wiederum nicht mit einem bereits benutzten Interrupt kollidieren.

Die wenigsten Probleme bereitet deshalb eine serielle Maus. Sie wird an einer freien Schnittstelle angeschlossen und der entsprechende Maustreiber mit der Parameterangabe der Schnittstellennummer gestartet. Als Beispiel wird die Genius-Maus an »com2« mit dem Aufruf

```
gmouse 2
```

installiert. Zu beachten ist natürlich vor dem Kauf, daß alle Programme, die Sie mit Mausunterstützung laufen lassen wollen, auch Ihr Fabrikat zulassen.

● Abstürze und ihre Ursachen: Ein schneller PC mag für Ihre Anwendungen vorteilhaft sein, manchmal entstehen dadurch aber ungewollt Schwierigkeiten:

○ Wenn beim Zugriff auf Schnittstellen oder Peripheriegeräten das Programm plötzlich abstürzt, deutet dies meist auf Timingprobleme. Die entsprechende Schnittstelle ist dann nicht für die eingestellte Geschwindigkeit angelegt. Gerade Computer, die eine Wahl der Bustaktrate erlauben, sind davon betroffen. Diese beträgt normalerweise 8 MHz. Sie läßt sich jedoch bei NEAT-Boards auch auf höhere Taktraten einstellen. Sollten Sie nicht über speziell dafür ausgelegte Karten verfügen, ist eine Taktrate von 10 MHz normalerweise das »problemlose« Maximum. Jedoch kann auch dafür keine Gewähr übernommen werden, da üblicherweise nur 8 MHz garantiert sind.

Stürzt Ihr PC ab, während Sie Texte eingeben, kann das daran liegen, daß die Speicherchips beispielsweise den Prozessortakt von 25 MHz nicht bewältigen – mitunter hilft das Runterschalten mit dem Turboknopf.

»Unmotivierte« Abstürze können vielfältige Ursachen haben:

○ Residente Programme, die sich stören. Also raus mit ihnen, jeweils nacheinander starten und prüfen, ob gerade diese Residenz für das Unheil verantwortlich war. Mitunter genügt es, die Startreihenfolge der residenten Programme zu verändern.

○ Das Programm ist auf Ihrem System gar nicht lauffähig. Prüfen Sie im Handbuch: Vielleicht ist Ihr Rechenkünstler nicht kompatibel genug.

● Festplatte voll: Auch 40- oder 80-MByte-Platten sind im Windows-Zeitalter schnell ausgeschöpft. Abhilfe schaffen verschiedene Aktionen:

○ Datei-Wildwuchs eindämmen: Sie löschen beispielsweise alle Tmp-Dateien, die von irgendwelchen Programmabstürze her rühren, ebenso alle alten Sicherungskopien (»*.sik«). Sie lagern unwichtige Dateien auf Diskette aus. Sie legen Archive an, die etwa mit »Lha« oder »Arj« erzeugt werden. Sie komprimieren stauchungsfähige Exe- oder Com-Dateien mit »Pklite« (natürlich vorher sichern). Sie überarbeiten Ihre Bilddateien und löschen gegebenenfalls die Bildergalerie Ihrer eingescannten Ex-Freundin etc.

○ Festplatte von Dateifragmenten befreien: Um von vornherein Datenverlusten vorzubeugen und um die Festplatte von Dateifragmenten zu befreien, die sich im Laufe der Zeit angesammelt haben, empfiehlt es sich, ab und zu das DOS-Programm »chkdsk« (Checkdisk) in Aktion treten zu lassen. Wird es in der Form

```
chkdsk /f
```

aufgerufen, durchsucht es die gerade aktuelle Festplatte nach Dateifragmenten. Wurden welche gefunden, wird deren Anzahl angezeigt und abgefragt, ob sie in normale Dateien umgewandelt werden sollen. Wird diese Frage mit »Ja« beantwortet, legt »chkdsk« Dateien, die die Erweiterung »chk« tragen, im Hauptverzeichnis der jeweiligen Festplatte ab. Diese sind nun mit

```
del \*.chk
```

zu löschen. Gerade bei Programmen, die ständig Dateien überschreiben, beispielsweise Datenbanken oder Textverarbeitungen, ist diese Prozedur des öfteren notwendig.

Die Datei »config.sys« ist nach »command.com« (und den versteckten Systemdateien) die wichtigste, um Ihren Computer richtig zu starten. In ihr werden alle Parameter gesetzt, die verändert werden sollen (Bild 1). Fehler, die dabei auftreten können:

● Seltsame Steuerzeichen auf dem Bildschirm (Bild 2): Sie haben vergessen, die Datei »ansi.sys« einzubinden, oder sie konnte nicht gefunden werden. Der Übersichtlichkeit wegen sollten Sie alle (nicht unbedingt notwendigen) Dateien des Betriebssystems, sofern Sie eine Festplatte besitzen, in einem eigenen Unterverzeichnis speichern – etwa »\dos«. Der richtige Eintrag in die Datei »config.sys« muß dann lauten:

```
device = c:\dos\ansi.sys
```

Nun können die Escape-Sequenzen des ANSI-Standards genutzt werden. Achten Sie unbedingt darauf, daß jede Zeile in der Datei »config.sys« mit einem »Return« abgeschlossen wird.

● Falsche Anzeigeform von Datum, Währung und Dezimalzahlen: Hierzu können Sie mit der Zeile

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

auf die deutsche Schreibweise umschalten. Dadurch wird das Datum in der deutschen Form »Tag-Monat-Jahr« dargestellt.

● Laufwerke ab »G« lassen sich nicht ansprechen: Da MS-DOS ohne zusätzliche Angaben nur maximal sechs Laufwerke ansprechen kann, müssen Sie mitteilen, daß Sie mehr benutzen wollen. Dafür setzen Sie das größtmögliche Laufwerk mit

```
lastdrive = j
```

auf den neuen Wert. Hierdurch werden im angegebenen Beispiel zehn Laufwerke angesprochen. Der größte Buchstabe, den Sie eingeben können, ist das »Z«. Daraus ergibt sich auch, daß mit MS-DOS maximal 26 Laufwerke zu verwalten sind.

Speicherprobleme: Wichtig ist, sich vor der Anschaffung eines Programms zu überzeugen, ob es auch mit dem vorhandenen Speicher läuft. Starten Sie deshalb das Programm »chkdsk«, und Sie erhalten eine Übersicht über die aktuelle Speicherkapazität und wieviel davon im Augenblick für Anwendungsprogramme zur Verfügung steht. Mit den neueren MS-DOS-Version ab MS-DOS 4.0 und DR-DOS 5.0 sind Sie mittlerweile in der Lage, auch den Bereich zwischen 640 KByte und 1 MByte sinnvoll zu nutzen (»hidos« respektive »himem«). Durch spezielle Programme lassen sich beispielsweise der Tastatur- oder Bildschirmtreiber in

diesen Bereich auslagern. Sogar das MS-DOS selbst findet dort seinen Platz. Dadurch bekommen Sie die normalen 640 KByte fast gänzlich für Programmanwendungen frei.

Bei Speichererweiterungen über 1 MByte ist es wichtig zu wissen, welches Verfahren Ihre Software unterstützt:

○ Erfordert Ihre Software Expansionsspeicher, auch »EMS« genannt, so benötigen Sie in jedem Fall eine spezielle Speicherkarte oder einen Computer, der diese Speicherart unterstützt. Dies ist beispielsweise bei Computern mit NEAT-Technologie der Fall. Außerdem ist eine Treibersoftware nötig, welche die Verwaltung übernimmt.

Sollte trotzdem die ominöse Meldung auf dem Monitor erscheinen:

● Nicht genug Speicher – etwa nach RAM-Disk-Installation (bei AT-Besitzern mit 1 MByte Speicher sehr häufig) –, so bedenken Sie:

○ Da die oberen 384 KByte unter MS-DOS normalerweise nicht verwaltet werden, wird dieser Speicher sinnvollerweise als RAM-Disk benutzt. Nur wird vergessen, das auch dem Treiber mitzuteilen. Dieser nimmt sich den Speicher aus den unteren 640 KByte (normaler MS-DOS-Bereich). Vergessen Sie deshalb auf keinen Fall, den Parameter »/e« an das Ende der Zeile zu stellen. Damit weiß »vdisk.sys« oder »ramdrive.sys«, daß es auf den Speicher über 640 KByte zugreifen soll. Eine korrekte Zeile für eine 384-KByte-große RAM-Disk lautet dann:

```
device = \dos\ramdrive.sys 384 /e
```

Aber auch ohne RAM-Disk kann der Speicher nicht ausreichend groß sein: Moderne Anwendungsprogramme sind selbstherrlich und dulden oftmals keine anderweitigen Treiber in den 640 KByte. Diese also erforderlichenfalls nicht mehr laden – weder über »config.sys«, wo Sie die entsprechenden Zeilen »aussternen«

```
*device = \dos\ramdrive.sys 384 /e
```

noch als Treiberprogramme.

Sie haben auch die Möglichkeit, mittels eines Hidos- oder Himem-Treibers einige KByte in den oberen MS-DOS-Bereich auszulagern.

○ Ein anderes Problem ergibt sich bei residenten Programmen, die den Arbeitsspeicher »zumüllen«: Danach ist der für die eigentliche Anwendung benötigte Platz nicht mehr frei. Hier hilft entweder eine Deinstallation – diese gelingt mit der Option »/d« oder »/u« – oder ein völliger Neustart.

Bei der Deinstallation entstehen mitunter auch Lücken im Arbeitsspeicher: Sie haben etwa den Norton Commander als Verzeichnismanager laufen, starten mit seiner Hilfe ein residentes Programm (Bild 3, links), verlassen den Commander (Bild 3, Mitte) – und schon ist ein unbenutzbares Loch im Speicher, das der Commander auch dann nicht wieder besetzt, wenn Sie ihn erneut aufrufen (Bild 3, rechts).

● Environmentspeicher zu klein: Wer seine Festplatte sauber strukturiert und alle Programme in einzelne Unterverzeichnisse kopiert, steht bald vor einem Problem (die dies nicht tun, stehen vor einem anderen, viel größeren). Soll auf die Programme aus allen Verzeichnissen zugegriffen werden, muß dazu (in der Datei »autoexec.bat«) mit der Angabe »path« darauf verwiesen werden (Bild 4). Die maximale Anzahl der Zeichen darf jedoch dabei 128 nicht überschreiten. Trotzdem kann diese Zeile bald zu einem Monstrum anwachsen und der diesbezügliche Speicher – wenn auch noch jede Menge Set-Angaben gemacht wurden – dafür zu klein werden. Dieser läßt sich jedoch leicht vergrößern: Tragen Sie dazu die Zeile

```
shell = command.com c:\ /e:2048 /p
```

in die Datei »config.sys« ein. Damit stehen 2 KByte Evironmentspeicher zur Verfügung. Dies sollte für alle Fälle ausreichen.

● Anwenderprogramm startet nicht: Meist geben Programme eine Meldung aus, wenn ihnen an der Konfiguration etwas nicht paßt. Oft sind aber auch im Handbuch die Installations- und Starthinweise nochmals zu überfliegen. Es könnte etwa sein, daß die

○ Files-Puffer zu klein sind: Dabei ist der Wert »files = 8« in der »config.sys« Grundeinstellung und muß nicht eingetragen werden. Benötigt Ihre Software einen höheren Wert, so können Sie diesen beispielsweise mit

```
files = 30
```

ändern. »Ensemble« benötigt etwa »files = 120«. Die 30er Einstellung reicht für die meisten Anwendungen aus. Pro Werterhöhung werden dafür 48 Byte vom Speicher abgezwackt.

● Anwendungsprogramm läuft nicht wie gewünscht: Bei der letzten PC-Sitzung war die Menüzeile noch blau, heute ist sie gelblich und kaum zu sehen. Oder die Voreinstellung bei den Dateinamen ist »doc« – wo Sie doch nur Txt-Dateien bearbeiten. Solche »Fehler« lassen darauf schließen, daß Sie eine wichtige Konfigurationsdatei gelöscht oder aus dem Verzeichnis bewegt haben – in der falschen Annahme, daß solche Ini- oder Cfg-Dateien nur beim Setup notwendig sind. Oder Sie haben nur vergessen, die neugewählte Konfiguration zu speichern. Die Datei »autoexec.bat« kann eine ganze Menge an Konfigurationsarbeit abnehmen (Bild 4):

● MS-DOS-Befehle oder Programme werden beim Aufruf nicht gefunden: Wenn Sie mit einer Festplatte arbeiten, sollten Sie auf jeden Fall immer mit dem Kommando »path« die Verzeichnisse angeben, in denen sich die Programme befinden, die Sie während Ihrer Sitzung gebrauchen. Angenommen, Sie arbeiten mit Windows, benötigen ferner eine Menge Tools und transiente MS-DOS-Programme, dann empfiehlt sich etwa ein Pfad

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

der nur voraussetzt, daß Ihre Programmdateien auch in den bezeichneten Verzeichnissen beheimatet sind. Dadurch können Sie alle erwähnten Programme aus jedem anderen Unterverzeichnis starten.

● Die Tastatur bringt keine deutschen Umlaute: Um die Tastatur auf den richtigen Zeichensatz einzustellen, ist es notwendig, den entsprechenden Tastaturtreiber zu laden. Der dafür erforderliche Eintrag in der Autoexec-Datei lautet:

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

Besitzen Sie eine ältere MS-DOS-Version, in der jeder länderspezifische Tastaturtreiber ein eigenes Programm ist, so reicht die Eingabe von »keybgr« – in diesem Fall ohne Leerzeichen – aus. Sollten Sie für bestimmte Programme gern die amerikanische Tastaturbelegung walten lassen wollen, können Sie jederzeit mit [Strg-Alt-F1] darauf umschalten. Mit der Kombination und [Strg-Alt-F2] sind Sie wieder im deutschen Tastaturtreiber.

● In welchem Verzeichnis ist MS-DOS gelandet? Wenn Sie direkt mit MS-DOS arbeiten (also ohne Verzeichnismanager), verlieren Sie sehr leicht die Orientierung und wissen nicht mehr, in welchem Unterverzeichnis Sie sich befinden. Nehmen Sie deshalb auf jeden Fall die Zeile

```
prompt $p$g
```

in Ihre »autoexec.bat« auf. Dadurch wird der System-Prompt um die Angabe des Pfades, in dem Sie sich gerade befinden, erweitert.

● Im Grafikmodus keine deutschen Umlaute: Wenn diese zwar im Textmodus vorhanden sind, beim Grafikmodus jedoch statt dessen nur Klammern oder merkwürdige Zeichen erscheinen, fehlt der entsprechende Treiber »graftabl«. Sie laden diesen am einfachsten bereits beim Starten automatisch mit. Dazu tragen Sie

```
c:\dos\graftabl
```

in die Datei »autoexec.bat« ein.

Trotz aller Tips und Hinweise ist es ratsam, im Zweifelsfall auf die Hilfe eines Fachmanns zu vertrauen – erhältlich etwa bei der »Hotline« Ihres Händlers. Gute Handbücher sowie Hinweise und Tips können zwar bei einfachen Problemen ausreichen, doch »müssen« sie alle auftretenden Fehlerquellen und Ursachen letztendlich nicht behandeln können.

(Helmut Stult/al)

Info:
Arj: Computer Solutions, 8018 Grafing
Pklite: PD-Service Lage, 4937 Lage



MS-DOS/DR-DOS

Befehle, die vom Stapel laufen

Sie verwenden Stapeldateien unter MS-DOS für zahlreiche Einsatzbereiche. Neben der Anpassung von Programm und Hardware an Betriebszustände können diese Dateien komplexe Befehlsabläufe zusammenfassen oder durch geschickte Nutzung der Fähigkeiten von MS-DOS »neue Befehle« für den täglichen Einsatz schaffen.

Schon kurze Zeit nach der ersten Kontaktaufnahme mit dem Betriebssystem MS-DOS bemerkt der Anwender, daß sich bestimmte Standardaufgaben herauskristallisieren. Die Gemeinsamkeit bei derartigen Routinetätigkeiten besteht sehr oft in einer Vielzahl von Einzelbefehlen, die Sie zur Durchführung stets in der gleichen Reihenfolge eingeben. Anfangs reichen eventuell noch handschriftliche Notizen, um den Programmaufruf mit erforderlichen Zusatzinformationen (»Parametern«) im Gedächtnis zu behalten. Mit zunehmendem Einsatz des Computers und einer damit verbundenen besseren Ausnutzung der Programm- und Gerätefähigkeiten sind Eingaben längerer Befehlsketten nötig, die Sie dem Betriebssystem zur Bewältigung einer Aufgabe mitteilen. Aus dieser Situation erwächst der Wunsch, mit einem einzigen, leicht zu merkenden Kommando alle Schritte durchzuführen, die mit der gewünschten Aufgabenstellung zusammenhängen. Diesem Anliegen trägt das Betriebssystem MS-DOS durch die Stapelverarbeitungsdateien, auch Batch-Dateien genannt, Rechnung. Sie erkennen diese Dateien mit der Endung »bat«, beispielsweise bei der Anzeige eines Verzeichnisinhalts mit dem Befehl »dir«. Eine Batch-Datei kann alle Befehle und Aufrufparameter enthalten, die auch in der MS-DOS-Eingabezeile verwendet werden.

Für die erfolgreiche Bearbeitung von Stapelverarbeitungsdateien ist es wesentlich, die zur Bearbeitung der Aufgabe relevanten Befehle und MS-DOS-Dienstprogramme zu kennen. In Tabelle 1 sind die wichtigsten dieser Befehle zusammengestellt. Im Handbuch zu Ihrer Betriebssystemversion finden Sie eine vollständige Aufstellung aller Befehle mit Parametern, bei denen Sie von Fall zu Fall entscheiden, ob sie für Ihre Bedürfnisse nötig sind.

Folgende Übersicht legt besonderen Wert auf den praktischen Nutzen der Befehle im alltäglichen Umgang mit dem Computer, deshalb sind nicht alle mit diesem Kommando verbundenen Optionen beschrieben, dafür aber Einsatzgebiete, die das Handbuch nicht oder nur schwer verständlich erläutert.

»cd« – Change Directory
Rufen Sie »cd« ohne Parameter auf, wird das aktuelle Laufwerk und Verzeichnis ausgegeben, ansonsten wechseln Sie beispielsweise mit

```
cd dos
```

zum angegeben Verzeichnis »dos«. Ist das Verzeichnis nicht vorhanden, erfolgt eine Fehlermeldung. Der Befehl »cd« dient zusammen mit der Anwahl des gewünschten Laufwerks zum Positionieren auf ein Verzeichnis. Besonders sinnvoll ist der Einsatz, wenn Suchpfade minimiert werden. Um ein Programm, beispielsweise auf Festplatte, aus einem beliebigen aktuellen Verzeichnis und beliebigem Laufwerk aufzurufen, müssen Sie MS-DOS entweder einen Suchpfad dorthin mitteilen (siehe Befehl »path«), oder Sie definieren genau dieses Laufwerk und dieses Unterverzeichnis mit seinem aktuellen Pfad. Mit

```
cd \beispiel
```

wechseln Sie in das Unterverzeichnis »beispiel«, bezogen auf das Hauptverzeichnis. Sofern die Stapeldatei zum Aufruf des Programms fertiggestellt ist, reicht es aus, das Verzeichnis mit dieser Batch-Datei im Suchpfad anzugeben.

»chkdsk« – Check Disk
Dieser oft weniger beachtete Befehl überprüft den Datenträger und gibt wichtige Systeminformationen aus. In zahlreichen Situationen (Absturz eines Programms während des Schreibzugriffs, Stromausfall, versehentliches Abschalten des Computers trotz laufender Anwendung) kann es zu Beschädigungen von Dateien kommen, die äußerlich nicht ersichtlich sind. Der Dir-Befehl liefert dann beispielsweise einen optisch korrekten Wert bei der Dateigröße. Auf dem Datenträger jedoch ist die Datei dann nicht mehr vollständig erhalten. »chkdsk« erkennt dieses Problem und macht darauf aufmerksam. Finden Sie defekte Dateien, sollten Sie »chkdsk« mit dem Parameter »/f« (fix, reparieren) aufrufen, um das Problem zu beseitigen. Erkennen Sie eine solche Situation frühzeitig, existiert vielleicht noch eine Sicherheitskopie eines Textes oder Kalkulationsblattes, die Sie anstelle der defekten Datei verwenden. Bei beschädigten Programmdateien ergeben sich ohne Korrektur mit Sicherheit unerklärliche Systemabstürze, die in der Regel mit Datenverlusten korrespondieren. »chkdsk« sollten Sie daher am besten täglich einsetzen, am günstigsten als festen Bestandteil der »autoexec.bat«, einer speziellen Stapeldatei, die das System beim Start automatisch ausführt.

Informant in Sachen Speicherauslastung

Eine der angedeuteten, wichtigen Systeminformationen von »chkdsk« betrifft die Größe des Hauptspeichers und des verbleibenden freien Bereiches. Ab MS-DOS Version 4.01 liefert der Befehl »mem« unter anderem diese Werte; Anwender früherer Versionen mußten eine Datenträgerprüfung vornehmen, um festzustellen, ob ihr freier Hauptspeicher zur Verwendung eines bestimmten Programms groß genug bemessen ist. Sofern Sie nicht völlig sicher sind, lohnt sich also der Vergleich der Systemwerte von »mem« oder »chkdsk« mit den Softwarevoraussetzungen laut Katalog, Händlerauskunft oder Produktangaben, bevor Sie ein Programm erwerben, das mehr Haupt- oder Festplattenspeicherplatz voraussetzt, als die Hardware Ihnen überhaupt zur Verfügung stellt. Dadurch können Sie Software unter Umständen nicht verwenden. »chkdsk« läßt sich aber auch einsetzen, um alle Datei- und Verzeichnisnamen auf dem Datenträger zu ermitteln. Der Befehlszusatz »/v« zeigt alle bearbeiteten Dateien und Verzeichnisse an. Da in allen Pfadangaben – und nur dort – die Zeichenfolge »:\« (»\« entspricht [Alt-9-2]) auftaucht, kann man sich diesen Effekt zunutze machen. Der Aufruf

```
chkdsk /v ¦ find ":\" > inhalt.txt
```

legt im aktuellen Verzeichnis eine Datei mit dem Namen »inhalt.txt« an und leitet alle Datei- und Verzeichnisnamen des gegenwärtigen Laufwerks dorthin. Die Befehlsverkettung »¦«, den Befehl »find« und die Datenumleitung »>« finden Sie in diesem Artikel weiter hinten im Detail beschrieben.

»cls« – Clear Screen
Mit »cls« löschen Sie den Bildschirm. Diesen Befehl verwendet man häufig am Beginn einer Stapeldatei, um Verwirrungen mit vorherigen System- oder Programmeldungen zu vermeiden. Am Ende einer Batch-Datei sollten Sie den Befehl vermeiden, da Sie sonst Fehler oder Warnungen nicht erkennen können.

»copy« – Kopie
Mit »copy« kopieren Sie eine Datei oder Dateigruppe in eine oder mehrere Dateien beziehungsweise auf ein angegebenes Gerät, oder Sie übertragen Dateien zu einer logischen Einheit. »copy« ist ein vielseitiger und leistungsfähiger Befehl. Nicht nur Dateien lassen sich von Verzeichnis zu Verzeichnis oder auf ein anderes Laufwerk übertragen. Eine wichtige Funktion ist das Senden von Dateien zu

Befehl	Kurzbeschreibung der Funktion
cd	Wechsel zum angegebenen Verzeichnis
chkdsk	Datenträger prüfen und wichtige Systeminformationen ausgeben
cls	Bildschirm löschen
copy	Datei oder Dateigruppen kopieren
date	Setzen und Ausgeben des Datums
del	Datei oder Dateigruppen löschen
dir	Informationen über Verzeichnisse, Dateien und den Datenträger
diskcopy	Kopieren (und Formatieren) von Disketten
erase	Datei oder Dateigruppen löschen
find	Suchen von Zeichenketten
format	Vorbereiten eines neuen Datenträgers zur Verwendung unter dem Betriebssystem MS-DOS
keyb	Tastaturtreiber aktivieren
md	Verzeichnis generieren
mode	Betriebszustand für seriell und parallel angeschlossene Geräte setzen, Videomodus festlegen
more	Daten seitenweise auf dem Bildschirm ausgeben
path	Setzen, Ausgeben und Löschen des Suchpfades für ausführbare Dateien
prompt	Erscheinungsbild der MS-DOS-Eingabeaufforderung bestimmen
rd	Verzeichnis löschen
ren	Datei oder Dateigruppen umbenennen
replace	Aktualisieren von Verzeichnissen
set	MS-DOS-Umgebungsvariablen anzeigen, erzeugen oder löschen
sort	Sortieren von Daten
time	Setzen und Ausgeben der Uhrzeit
type	Inhalt einer Datei ausgeben
verify	Aktivieren oder Deaktivieren der Fehlerprüfung bei Schreibzugriff auf einen Datenträger
xcopy	Kopieren von Dateien und Verzeichnisstrukturen
z:	Laufwerk mit der Bezeichnung »z:« wird zum aktuellen Laufwerk

Tabelle 1. Die wichtigsten MS-DOS-Befehle und Dienstprogramme für Stapelverarbeitungsaufgaben

Bezeichnung	Funktion
aux	gleiche Funktion wie »com1« (Standardschnittstelle für serielle Geräte)
com1	erste serielle Schnittstelle
com2	zweite serielle Schnittstelle
com3	dritte serielle Schnittstelle (ab DOS 3.3)
com4	vierte serielle Schnittstelle (ab DOS 3.3)
con	Bildschirm und Tastatur
lpt1	erste parallele Schnittstelle (Druckeranschluß 1)
lpt2	zweite parallele Schnittstelle (Druckeranschluß 2)
lpt3	dritte parallele Schnittstelle (Druckeranschluß 3)
nul	physikalisch nicht existente Einheit für spezielle Ausgabeumleitungen
prn	gleiche Funktion wie »lpt1« (Standardschnittstelle für Drucker)

Tabelle 2. MS-DOS-Gerätenamen und Einheiten-Bezeichnungen

Schnittstellen. Die Ausgabe der Datei »autoexec.bat« im Hauptverzeichnis der Festplatte »c:« auf den Drucker an der ersten parallelen Schnittstelle erfolgt durch

```
copy c:\autoexec.bat lpt1:
```

Handelt es sich um einen Laserdrucker, können Sie mit

```
echo^L>lpt1:
```

einen Seitenvorschub ([Alt-1-2] beziehungsweise [Strg-L]) erzwingen. Verwenden Sie den Copy-Befehl mit Schnittstellen- oder Gerätenamen, geht der Befehl davon aus, daß es sich um ASCII-Dateien handelt. ASCII-Dateien enthalten lediglich die Zeichen von dezimal 32 bis 127, also Buchstaben, Zahlen und einige Sonderzeichen. Für Dateien, die den vollen Zeichenvorrat (256 Zeichen, darunter nicht darstellbare Zeichen und Steuersequenzen) brauchen, wie dies beispielsweise bei Binärdateien der Fall ist, muß ein Befehlszusatz »copy« mitteilen, daß es sich hierbei nicht um ASCII-Daten handelt. Mit der Befehlssequenz

```
copy /b druckdat.txt lpt1:
```

lassen sich Dateien mit Text und Steuersequenzen für den Drucker oder auch Binärdaten korrekt übermitteln.
»copy« erlaubt weiter, Dateigruppen in eine Datei zu kopieren. Die Anweisungsfolge

```
copy *.bat gesamt.txt
```

kopiert alle Stapeldateien des aktuellen Verzeichnisses in der Reihenfolge, in der sie auf den Datenträger kopiert oder dort erzeugt wurden, hintereinander in die Datei »gesamt.txt«. Wenn die Zeichen »*« und »?« als Platzhalter für Dateigruppen zu viele Ergebnisse liefern, lassen sich auch einzelne Dateien miteinander verbinden und unter neuem Namen ablegen.

```
copy test.bat+test2.bat gesamt.bat
```

Diese Anweisung verbindet lediglich »test.bat« mit »test2.bat« und legt deren Inhalt in der beim Befehl aufgeführten Reihenfolge in der Zieldatei »gesamt.bat« ab. Weiterhin können Sie auch Daten von einem Gerät beziehen und dann in eine Datei übertragen oder auf eine andere Einheit senden. Beispielsweise holt

```
copy con test.bat
```

Daten zeilenweise von der Tastatur und leitet sie in die Datei »test.bat«, bis Sie [Strg-Z] beziehungsweise [F6] und [Enter] eingeben. Dies ist eine der Methoden, Stapeldateien zu erzeugen. Für kleinere Aufgaben reicht diese Technik aus, da Sie Fehler bei der Eingabe in der noch nicht mit [Enter] beendeten Zeile so wie in der Kommandozeile üblich korrigieren. Sie sollten jedoch komplexere Dateien mit einem ASCII-Editor, einem Textverarbeitungsprogramm, das ohne Steuerzeichen (»unformatiert«) speichern kann, oder dem MS-DOS-Dienstprogramm »edlin« schreiben, da diese Textpassagen suchen, ersetzen oder Teile verschieben können.

Selbst Datum und Zeit der Speicherung einer Datei, wie mit dem Dir-Befehl ausgewiesen, läßt sich mit »copy« beeinflussen. Die Befehlssequenz

```
copy /b test.bat+,
```

kopiert die Datei »test.bat« vollständig in sich selbst und paßt das Datum dem Systemdatum an. Der Zusatz »/b« ist eigentlich nur für Binärdateien wichtig, schadet bei ASCII-Dateien aber nicht. Ohne diesen Zusatz oder mit dem Parameter »/a« aufgerufen, wird die Datei nur bis zu dem Dateiendekennzeichen ([Strg-Z], [Alt-2-6]) bearbeitet. »/b« ignoriert dieses Dateiendekennzeichen und kopiert soviele Zeichen, wie der Dir-Befehl ausweist. Wollen Sie Datum- und Zeiteintrag der Datei »test.bat« gleichzeitig ändern, lautet der entsprechende Befehl:

```
copy /b test.bat+,,
```

Vor Aufruf eines solchen Befehls können Sie mit »date« das Datum und mittels »time« die Systemzeit nach Belieben setzen und anschließend auch korrigieren. Nicht nur Kombinationen aus den zuvor gezeigten Fähigkeiten des Copy-Befehls sind möglich, eine weitere Bereicherung bringt die bei den Stapelkommandos beschriebene Befehlskonstruktion »for ... in (...) ... do«, wie noch ausgeführt wird.

»date« – Datum
Der Date-Befehl gibt das Datum aus. Gegebenenfalls können Sie damit auch das Datum setzen, was für Computer ohne Echtzeituhr wichtig ist.

»del« – Delete
Der Del-Befehl löscht eine Datei oder mit der Eingabe von Platzhaltern ganze Dateigruppen. Nach der Eingabe von

```
del *.*
```

gibt das Betriebssystem eine Sicherheitsabfrage aus.

»dir« – Directory
Der Dir-Befehl bietet Information über Verzeichnisse, Dateien, Laufwerke und verfügbaren Speicherplatz auf dem Datenträger.

»diskcopy«
Mit »diskcopy« kopieren Sie komplette Disketten. Sie können dabei als Zieldiskette eine neue, unformatierte Diskette verwenden, da zuvor der neue Datenträger formatiert wird.

»erase«
Der Erase-Befehl ist mit dem Del-Befehl identisch.

»find«
Mit »find« erhalten Sie einen Filter zum Suchen von Zeichenketten. Wie das Beispiel »chkdsk« zeigt, lassen sich mit »find« verblüffende Ergebnisse erzielen. Filter bedeutet, daß dieses Programm mit der Befehlsverkettung »|« ([Alt-1-2-4]) die Ausgabe eines anderen Programms als Eingabe heranzieht und darin die angegebene Zeichenkette sucht. Auf Befehlsverkettung wird noch ausführlicher eingegangen. Da beim Dir-Befehl das Datum in der Form »Tag.Monat.Jahr« ausgegeben wird, lassen sich alle Dateien des aktuellen Tages, eines bestimmten Monats oder speziellen Jahres im gegenwärtigen Verzeichnis auffinden. Nehmen wir an, das aktuelle Verzeichnis enthält alle Ihre Textdateien, und Sie erinnern sich nicht mehr an den Namen des gesuchten Textes, wissen aber noch, daß er im März 1991 entstanden ist. Die Anweisung

```
dir *.txt | find ".03.91"
```

findet alle Dateien aus diesem Zeitraum. Mit dem Parameter »/v« werden alle Zeilen ausgegeben, in denen die gesuchte Zeichenfolge nicht gefunden wurde,

```
dir *.txt | find /v ".03.91"
```

gibt demnach alle Dateien aus, die nicht im März 1991 entstanden sind (allerdings auch die sonst bei der Dir-Ausgabe üblichen Informationen). Suchen Sie eine Zeichenfolge in einer Datei, verwenden Sie den Find-Filter mit dem Type-Befehl, beispielsweise:

```
type info.txt | find "Helmut"
```

Mit der Eingabe dieser Zeile finden Sie in diesem Text »Helmut«.

»format«
Der Format-Befehl bereitet einen neuen Datenträger zur Verwendung unter dem Betriebssystem MS-DOS vor. Um

spezielle Parameter nicht immer nachzuschlagen oder sich zu merken, empfiehlt sich die Verwendung einer einzeiligen Stapeldatei für jeden Anwendungsfall, mit einem Namen, den Sie nicht vergessen. Beispielsweise schreiben Sie eine Stapeldatei mit dem sinngebenden Namen »360.bat«, die die Zeile enthält

```
format a:/4
```

Damit formatieren Sie dann Ihre 360-KByte-Disketten auf einem 1,2-MByte-Laufwerk ohne Systemspuren.

»keyb«
Mit »keyb« aktivieren Sie den Tastaturtreiber. Die Festlegung erfolgt üblicherweise beim Systemstart in der Datei »autoexec.bat«.

»md« – Make Directory
Mit der Eingabe von »md« legen Sie ein Verzeichnis auf Ihrem Datenträger an. Hierbei sind die üblichen Vereinbarungen von MS-DOS einzuhalten, die auch bei der Vergabe von Dateinamen gelten. Bis zu acht Zeichen dürfen Sie vor und drei weitere nach dem Punkt eingeben.

»mode«
»mode« ist ein sehr leistungsfähiges Dienstprogramm zur Einstellung von Betriebszuständen bei seriell oder parallel angeschlossenen Geräten sowie des gewünschten Bildschirmmodus. Bei Druckern an der parallelen Schnittstelle läßt sich die Zeilenlänge und der Zeilenabstand bei Ausgaben festlegen, und bei seriell angesteuerter Hardware ändert »mode« die Parameter (Übertragungsgeschwindigkeit, Parität und dergleichen). Auch können Sie Daten zwischen parallelen und seriellen Schnittstellen umleiten. So taucht »mode« in zahlreichen Befehlskombinationen auf:

```
mode co40
```

schaltet auf vierzig Zeichen je Zeile mit Farbdarstellung um,

```
mode 80
```

beziehungsweise

```
mode co80
```

auf die gewohnte 80-Zeichen-Darstellung. »mode lpt1« bewirkt einige Standardeinstellungen beim Druckbetrieb über die erste parallele Schnittstelle.

```
mode lpt2:132
```

teilt dem Drucker am zweiten Druckeranschluß mit, daß die Druckausgabe mit komprimierter Schrift (132 Zeichen pro Zeile) erfolgen soll. Ein weiteres Betätigungsfeld für den Mode-Befehl ist die Umleitung von Daten, die an die parallele Schnittstelle gesendet werden, zu einer definierten, seriell angeschlossenen Einheit. Dies ist beispielsweise dann der Fall, wenn Sie einen Laser- oder Matrixdrucker an eine serielle Schnittstelle anschließen. Sie haben beispielsweise Ihren Laserdrucker an der »com2:« angeschlossen, und möchten die Druckausgaben der Standarddruckerschnittstelle »lpt1« dorthin umleiten, verwenden Sie den Befehl:

Symbol	Beschreibung der Funktion
¦	([Alt-1-2-4]) Filter zur Verkettung von Befehlen. Die Ausgaben eines Programms dienen als Eingabe für das folgende.
<	([Alt-6-0]) Quelle für Dateneingaben festlegen.
>	([Alt-6-2]) Ziel von Datenausgaben festlegen. Eine eventuell existierende Datei gleichen Namens wird überschrieben.
>>	Ziel von Datenausgaben festlegen. Im Unterschied zu »>« werden beim Schreiben der Informationen in eine Datei die neuen Daten angefügt.

Tabelle 3. Alle Filter- und Dateiumleitungssymbole auf einen Blick

```
mode lpt1:=com2:
```

Solche wesentlichen Initialisierungen binden Sie in die Stapeldatei »autoexec.bat« ein. Eventuell ist beim letzten Beispiel eine weitere Einstellung erforderlich – das Setzen der Übertragungsparameter für den Betrieb der seriellen Schnittstelle. Dazu lautet der Befehl beispielsweise

```
mode com2:1200,n,8,1
```

für ein Modem, das mit 1200 Baud, ohne Parität, mit acht Datenbits und einem Stoppbit arbeitet. Sind derartige Einstellungen erforderlich, finden Sie im Handbuch des jeweiligen Geräts einen Vermerk dazu.
Die Datenumleitung von »lptx« nach »comx« können Sie wiederum mit folgendem Befehl rückgängig machen:

```
mode lpt1:
```

Dies ist nur dann sinnvoll, wenn sich an diesem Druckerport tatsächlich ein weiterer Drucker befindet, und die Umleitung zu einem seriell angeschlossenen Drucker nur zeitweilig den Vorrang über diese Schnittstelle besitzen soll.

»more«
Dieser Befehl bewirkt eine seitenweise Ausgabe von Daten auf den Bildschirm. Sie ziehen nochmals das Beispiel bei »find« heran und gehen davon aus, daß mehr Dateien im März 1991 entstanden sind, als eine Bildschirmseite darstellen kann. Damit Sie die ausgegebenen Fundstellen seitenweise lesen können, hängen Sie bei dem Beispiel unter »find« noch »¦ more« an:

```
dir *.txt ¦ find ".03.91" ¦ more
```

An Hand dieser Anweisung sehen Sie auch, daß Sie Filter durchaus mehrfach hintereinander einsetzen dürfen.

»path«
Mit dem Path-Befehl setzen Sie den aktuellen Suchpfad für die Ausführung Ihrer Com-, Exe- und Bat-Dateien. Nur mit der Eingabe von »path« erhalten Sie eine Anzeige des aktuellen Pfades. Die umfassende Path-Angabe ist mitentscheidend, wie häufig Sie die Meldung »Falscher Befehl oder Dateiname!« lesen müssen. Alle im Path-Befehl festgelegten Verzeichnisse durchsucht das System in der Reihenfolge der Aufstellung nach einem eingegebenen Programmnamen, den es im aktuellen Verzeichnis nicht gefunden hat. Bei der Path-Anweisung sollte mindestens das Hauptverzeichnis des Startlaufwerks beziehungsweise die Festplatte mit den Systemspuren sowie das Verzeichnis mit den Betriebssystem-Hilfsprogrammen genannt sein, beispielsweise

```
path=c:\;c:\dos;
```

Weiterhin lohnt es sich, ein Verzeichnis für Ihre Hilfsprogramme sowie eines für Stapeldateien anzulegen. Die zugehörige Befehlszeile könnte dann wie folgt lauten:

```
path=c:\;c:\dos;d:\hilfen;c:\batch;
```

In diesem Fall wäre das Verzeichnis mit den Hilfsprogrammen auf Laufwerk »D:« zu finden. Angenommen, Sie haben ein Grafikprogramm »graf.exe« sowie zahlreiche Bilddateien, Treiberprogramme und ähnliches auf Laufwerk »D:« im Verzeichnis »grafik« eingerichtet, wie dies üblicherweise Installationsprogramme bewältigen, und Sie verwenden weiterhin die letzte Path-Anweisung. Es reicht, eine kleine Stapeldatei zu schreiben, die dieses Programm immer korrekt startet, unabhängig davon, welches Laufwerk und Verzeichnis gerade gültig ist:

```
@echo off
d:
cd\grafik
graf
```

Diese Stapeldatei legen Sie im Unterverzeichnis »c:\batch« an. Die erste Zeile unterdrückt dabei die Ausgabe der Befehle aus der Stapeldatei, »d:« wechselt zum D-Laufwerk, »cd\grafik« springt über das Hauptverzeichnis ins Unterverzeichnis »grafik«, und »graf« startet schließlich das eigentliche Programm. Beim Cd-Befehl ist das Zeichen »\« (»Backslash«, [Alt-9-2]) sehr wichtig, denn ohne Sprung über das Hauptverzeichnis (also lediglich »cd grafik«) wäre es nicht mehr beliebig, in welchem Verzeichnis Sie arbeiten, bevor Sie die Stapeldatei aufrufen. Das Ergebnis wäre die gelegentlich auftretende Fehlermeldung »Ungültiges Verzeichnis!«, gefolgt von der Meldung »Falscher Befehl oder Dateiname!«. »graf« ließe sich in dieser Situation mit der nicht vollständig korrek-

ten Stapeldatei nicht starten. Durch die Aufnahme des Verzeichnisses mit allen Stapeldateien in den Path-Befehl ergeben sich mehrere Vorteile: Es wird nur wenig Speicherkapazität des MS-DOS-Umgebungsbereiches (Standardeinstellung 128 Zeichen) belegt, alle wichtigen Programme sind von beliebiger Stelle erreichbar (ein Teil davon wird direkt in den aufgeführten Verzeichnissen gefunden, bei den verbliebenen wird über Stapeldateien der korrekte Aufruf gewährleistet) und schließlich befindet sich in den über »path« erreichbaren Verzeichnissen kein Datenballast. Letzteres wäre dann der Fall, wenn das zuvor beschriebene Verzeichnis »grafik« direkt in die Path-Anweisung aufgenommen worden wäre. Das Betriebssystem würde dann bei jedem Programmnamen, den es im aktuellen Verzeichnis nicht finden kann, die komplette Verzeichnisaufstellung absuchen. Bei umfangreichen Unterverzeichnissen mit lediglich einer ausführbaren Datei ergäbe sich dadurch eine unnötig lange Wartezeit, bis der entsprechende Befehl ausgeführt oder eine Fehlermeldung ausgegeben würde. Hieraus ergibt sich übrigens eine weitere Schlußfolgerung: Die Reihenfolge der Aufführung aller Verzeichnisse, auf die das Betriebssystem direkt zugreifen kann, sollten Sie so wählen, daß Sie das am häufigsten verwendete Verzeichnis an den Anfang und weniger oft gebrauchte eher ans Ende setzen. Die Festlegung der Path-Umgebung erfolgt üblicherweise beim Systemstart in der Datei »autoexec.bat«.

»prompt«
Mit dem DOS-Prompt erfolgt eine Anpassung der MS-DOS-Eingabeaufforderung (üblicherweise »a>« oder »c>«) und der dortigen Verwendung von Farben nach individuellen Wünschen. Besonders interessant ist »$p« zur Ausgabe des aktuellen Laufwerks und Verzeichnisses bei jeder MS-DOS-Eingabeaufforderung. Gefolgt von »$g« ergibt sich ein typisches Erscheinungsbild der Betriebssystem-Eingabezeile:

```
prompt $p$g
```

Die Festlegung erfolgt üblicherweise in der Datei »autoexec.bat« beim Systemstart.

»rd« – Remove Directory
Mit »rd« werden Verzeichnisse gelöscht.

»ren« – Rename
»ren« benennt Dateien um. Es unterstützt die Ersatzzeichen »*« und »?« – somit können Sie auch eine Gruppe von Dateien umbenennen:

```
ren *.doc *.txt
```

Durch diese Eingabe erhalten alle Dateien im aktuellen Verzeichnis die Endung »txt« anstelle »doc«.

»replace«
Sie aktualisieren damit Verzeichnisinhalte. Mit diesem oft unterschätzten Befehl lassen sich Daten oder Programme ersetzen. Ihre Textdateien liegen beispielsweise in mehreren Unterverzeichnissen verteilt auf Festplatte »C:« vor. Sicherheitskopien der Texte sind im Hauptverzeichnis einer Diskette gespeichert. Nun haben Sie einen oder mehrere Texte auf einer Diskette korrigiert und erweitert. In einem solchen oder ähnlich gelagerten Fall versagt der Copy-Befehl, da Sie ja nicht alle Texte von Laufwerk »A:« in ein Unterverzeichnis kopieren wollen. Möglicherweise bestehen auch auf Laufwerk »C:« in einem der Unterverzeichnisse aktuellere Textversionen als auf der Diskette. Das manuelle Vergleichen von Einzeldateien bei umfangreichen Verzeichnissen kann – selbst mit Ausdrucken der Verzeichnisausgabe – zu Schwerarbeit ausufern. Eine praktische Alternative bietet hierbei die Replace-Anweisung. Sie beseitigen das geschilderte Problem durch die Eingabe folgender Kommandozeile:

```
replace a:\*.txt c:\ /s /d /r /p
```

Mit dieser Syntax suchen Sie alle Dateien mit der Endung ».txt« des Laufwerks »A:« in einem beliebigen Verzeichnis (»/s«) auf der Festplatte »C:«. Wird dort eine Datei gleichen Namens entdeckt, wird sie ersetzt, sofern die Version von Laufwerk »A:« neueren Datums ist (»/d«). Der Parameter »/p« verlangt noch eine Bestätigung vor dem Aktualisieren einer Datei und »/r« bewirkt, daß auch schreibgeschützte Dateien übertragen werden. Aus diesem Beispiel geht hervor, daß einige Befehle sehr viele Parameter besitzen, die man schwer in Erinnerung behalten kann. »replace« bietet noch weitere Schalter zur Steuerung an, beispielsweise »/a«, um nur Dateien aufzunehmen, die im Zielverzeichnis noch nicht vorhanden sind. Diesen Parameter können Sie jedoch nicht zusammen mit »/d« verwenden. Es liegt nun an der Aufgabenstellung, die geeignete Option zu wählen, oder in einer Stapeldatei beide Konstruktionen nacheinander einzusetzen. Taucht eine Situation auf, die wieder auftreten wird, lohnt es sich, selbst für einzelne Befehlszeilen Stapeldateien anzulegen.

»set«
Die Set-Anweisung zeigt Umgebungsvariablen an, erzeugt sie oder löscht sie. Im MS-DOS-Umgebungsbereich sind für das Betriebssystem wichtige Informationen gespeichert, unter anderem die Werte von »prompt«, »path« und »comspec«. Auch einige Programme setzen hier Variablen oder lassen sich dadurch in ihrer Funktion beeinflussen (beispielsweise Komprimierungsprogramme wie »lharc« oder »pkzip« durch die Variablen »tmp« und »pktmp«). »set« kann auch von Stapeldateien zur Zwischenspeicherung von Informationen verwendet werden. Die Kommandozeile

```
set tmp=e:
```

können Sie aus einer Batch-Datei folgendermaßen verwenden:

```
dir %tmp%
```

Die entsprechende Variable ist also vorher und nachher direkt von einem Prozentzeichen umschlossen.

»sort«
Filter zum Sortieren von Daten. Verwenden Sie nochmals das bei »find« und »more« beschriebene Beispiel zum Filtern der Textdateien, die im März 1991 entstanden sind. Gehen Sie davon aus, daß selbst diese Unterteilung noch zu viele Ergebnisse liefert. Da auch »sort« ein Filter ist, können Sie ihn mit »find« und »more« beliebig kombinieren. Des weiteren sind Sie in der Lage, zwei Schalter von »sort« ebenfalls zu verwenden: »/r« kehrt die

Befehl	Beschreibung der Funktion
call	Eine Stapeldatei ruft eine weitere zur Befehlsausführung als Unterprogramm auf. Nachdem alle Befehlszeilen in der aufgerufenen Stapeldatei bearbeitet wurden, wird der weitere Ablauf mit der Zeile nach dem Call-Aufruf fortgesetzt (ab MS-DOS 3.3).
echo	Dient der Ausgabe individueller Meldungen von Stapeldateien. Mit dem Parameter »off« läßt sich die Ausgabe von Systemmeldungen unterbinden.
for ... in (...) ... do	Befehlskonstrukt zur Bearbeitung von Aufzählelementen oder mehrerer, ausgewählter Dateien mit einem Befehl
goto	Fortsetzung der Bearbeitung an der Position einer Sprungmarke (Label)
if	Bedingte Ausführung von Befehlen
pause	Erzwingt die Eingabe eines Tastendrucks vom Anwender
rem	Einleitung einer Kommentarzeile
shift	Rückt alle bei Aufruf in der Befehlszeile übergebenen Parameter um eine Position auf.
@	Als erstes Zeichen einer Befehlszeile unterdrückt dies die Ausgabe der Befehlszeile, selbst wenn »echo« eingeschaltet sein sollte (ab MS-DOS 3.3)

Tabelle 4. Alle Stapelbefehle auf einen Blick

Sortierreihenfolge um. Mit dem Zusatz »/+Zahl« legen Sie fest, ab welcher Spalte Sie sortieren wollen. Sehen Sie sich die Ausgabe des Dir-Befehls genauer an, stellen Sie fest, daß die Programmnamen von der ersten bis zur achten Spalte, die Endungen von der zehnten bis zur zwölften Spalte, die Programmgröße ab Position 14, das Datum ab Stelle 24 und der Zeiteintrag schließlich ab Spalte 34 abgelegt sind. Daraus kombinieren Sie die Suche nach den Textdateien vom März 1991 mit einer nach Datum und Zeit abwärts sortierten Ausgabe. Die zuletzt entstandenen Textdateien sind ganz am Anfang zu finden:

```
dir *.txt ¦ find ".03.91" ¦ sort /r /+24 ¦ more
```

Schreiben Sie mit diesen Befehlen und ihrer Syntax eine Stapeldatei, beispielsweise »datdir.bat«.

»time«
Dieser Befehl bestimmt die Tageszeit (wichtig für Computer ohne Echtzeituhr) und gibt sie bei Befehlseingabe aus.

»type«
Dieser Befehl gibt den Inhalt einer Datei aus. Standardmäßig erfolgt die Anzeige auf dem Bildschirm. Sie können die Ausgabe jedoch auch auf eine serielle Schnittstelle, einen Drucker oder in eine Datei umlenken. »type« erlaubt keine Stellvertreterzeichen, für die Sie auch die Bezeichnung »Jokerzeichen« oder »Wildcards« finden. Beim Type-Befehl müssen Sie Dateinamen also eindeutig angeben. Durch die Befehlskonstruktion »for ... in (...) ... do«, die später bei Stapelkommandos beschrieben wird, läßt sich das Hindernis der nicht erlaubten Platzhalterzeichen umgehen. Der Einsatz von »type« ist auf ASCII-Dateien beschränkt – zumindest bringt er nur dort sinnvolle Ergebnisse. Die Ausgabe der Datei »test.txt« auf einen an der ersten parallelen Schnittstelle angeschlossenen Drucker erfolgt demnach mit:

```
type test.txt >lpt1:
```

oder

```
type test.txt >prn:
```

Mit »type test.txt« sehen Sie also den Inhalt der Datei »test.txt« auf Ihrem Bildschirm.

»verify«
Mit »verify on« aktivieren Sie die Fehlerprüfung nach jedem Schreibzugriff auf einen Datenträger, mit »verify off« deaktivieren Sie diese Prüfung. Die Eingabe von »verify« ohne Parameter zeigt den Status von »verify« an. Das Betriebssystem geht, solange Sie nichts anderes definieren, davon aus, daß keine Prüfung stattfindet. Die Festlegung sollte in der Datei »autoexec.bat« beim Systemstart erfolgen.

»xcopy«
Der Befehl kopiert Dateien und Verzeichnisstrukturen. Auch dieses Dienstprogramm wird in seiner Leistungsfähigkeit oftmals unterschätzt. »copy« kopiert lediglich Dateien oder Dateigruppen, wobei es Unterverzeichnisse nicht berücksichtigt. Wollen Sie auch diese übertragen, könnte prinzipiell der Befehl »backup« zur Anwendung kommen. Dies ist jedoch nur dann empfehlenswert, wenn das zu übertragende Verzeichnis mit seinen Unterverzeichnissen die Speicherkapazität des Zieldatenträgers überschreitet. Zwei Nachteile verschlagen »backup« auf den letzten Platz der zur Verfügung stehenden Kopieralternativen: Zum einen erzeugt »backup« eine einzelne Datei auf dem Zieldatenträger, die Sie nicht mehr direkt verwenden können, weil Sie sie zuerst mit »restore« wieder in die einzelnen Dateien umwandeln müssen. Der andere besteht in der Problematik der Inkompatibilität von Backup- und Restore-Versionen verschiedener Betriebssysteme. Der Datenaustausch zwischen PCs mit unterschiedlichen MS-DOS-Versionen ist dadurch mit »backup« nicht oder nur eingeschränkt machbar.

Das sind starke Argumente für den Einsatz von »xcopy«. Auch dieser Befehl wartet mit einer ganzen Palette von Parametern auf. Rufen Sie sich das beim Path-Befehl verwendete Beispiel des Grafikprogramms mit Namen »graf.exe« auf Laufwerk »D:« im Verzeichnis »grafik« in Erinnerung. Nehmen Sie weiterhin an, daß außer zahlreichen Bilddateien, Zeichensätzen, Druckertreibern und so weiter auch etliche Unterverzeichnisse enthalten sind. Es bestehe beispielsweise für jede Grafikauflösung ein eigenes Unterverzeichnis. Nun wollen Sie das komplette Verzeichnis »grafik« auf die Festplattenpartition »E:« verschieben. Dies könnte erforderlich sein, wenn auf Laufwerk »D:« der verbleibende Speicherplatz zu gering zur weiteren Arbeit mit dem Grafikprogramm ist und alternativ auf »E:« ausreichende Reserven bereitstehen. Eine einzige Zeile bewältigt das Übertragen des kompletten Verzeichniszweigs:

```
xcopy d:\grafik e: /s /e
```

Der Parameter »/s« schließt dabei die Unterverzeichnisse von »grafik« mit ein, und »/e« kopiert auch leere Verzeichnisse, die eventuell das Programm »graf« trotz fehlender Dateien benötigen könnte. Leider stellt das Betriebssystem MS-DOS keinen analogen Befehl zum Löschen von Verzeichnissen mit enthaltenen Unterverzeichnissen zur Verfügung. Das Programm »direx.com« in DOS Extra Ausgabe 14, ab Seite 24, löst dieses Problem. Somit ist zum Entfernen eines solchen Verzeichnisastes nötig, jedes einzelne Verzeichnis mit »cd« anzuspringen, dessen Inhalt mit »del *.*« zu löschen und solange mit »cd« tiefer einzutauchen, bis keine weitere Ebene mehr folgt. Aus der direkt darüberliegenden Ebene (»cd..«) löschen Sie dann mit »rd Verzeichnisname« Stück um Stück auch jedes einzelnen Unterverzeichnisses. Wieviel Arbeit »xcopy« gegenüber »copy« eingespart hat, erkennen Sie daran, daß Sie bei Verwendung von »copy« die einzelnen Verzeichnisse zuvor mit »md Verzeichnisname« (Verzeichnis erzeugen) und mit »cd Verzeichnisname« (Verzeichnis anwählen) spiegelbildlich zu deren Löschung hätten bearbeiten müssen.

»z:«
Umschalten auf das Laufwerk mit der Bezeichnung »Z:« können Sie naturgemäß nur dann, wenn ein Laufwerk mit dieser Bezeichnung vorliegt. Ansonsten erhalten Sie die Fehlermeldung »invalid drive specification«, was bedeutet, daß Sie einen ungültigen Laufwerksnamen gewählt haben.

MS-DOS-Gerätenamen und deren Eigenschaften

Gelegentlich tauchen bei der Übersicht über die wichtigsten MS-DOS-Befehle und Dienstprogramme Geräte- und Einheitennamen auf, beispielsweise bei »copy« und »type«. In Tabelle 2 sind sie in einer Kurzübersicht zusammengestellt. Einige Geräte- und Einheitennamen stellen wertvolle Bereicherungen für Stapelverarbeitungsaufgaben dar. »nul« ist gewissermaßen ein logisches Nichts. Die Funktion von »nul« wird klarer, wenn Sie sich vorstellen, daß Sie bestimmte unwichtige oder störende Meldungen unterdrücken wollen. Gäbe es keinen »Mülleimer« wie die Einheit »nul«, so müßten Sie die Ausgaben in eine Datei umleiten und anschließend wieder löschen. Der Befehl

```
copy c:\texte\*.txt a:\
```

gibt beim Kopieren die jeweiligen Programmnamen und am Ende die Anzahl der übertragenen Dateien aus. Möchten Sie dies verhindern, hängen Sie einfach an die Kommandozeile »>nul« an:

```
copy c:\texte\*.txt a:\ >nul
```

Die Systemmeldungen sind nun verschwunden. Die Umleitung von System- und Programmeldungen »ins Nichts« sollten Sie dann vermeiden, wenn das betreffende Programm während des Ablaufs Eingaben von Ihnen benötigt, da auch diese Informationen sonst das »nul« »verschlucken« würde. Aus Tabelle 2 geht hervor, daß das Betriebssystem MS-DOS insgesamt vier serielle (»comx:«) und drei parallele (»lptx:«) Schnittstellen erkennt. Häu-

fig verwenden Sie Computer nur mit einer seriellen und einer parallelen Schnittstelle, Erweiterungen müssen dann Sie selbst oder ein Fachhändler vornehmen. Aus den Systemunterlagen geht hervor, wieviele und welche Anschlußoptionen Ihr PC besitzt. Betriebssystemversionen vor 3.3 können nur zwei serielle Schnittstellen ansprechen, was bei der Vielzahl von Geräten, die diesen Anschluß verwenden (Maus, Modem, Akkustikkoppler und so weiter), natürlich zu wenig war.
Die Einheit »con« stellt die Standardein-/-ausgabeeinheit dar (Tastatur und Bildschirm). Ohne daß Sie dies jeweils ausdrücklich erwähnen müssen, setzen die einzelnen Dienstprogramme es voraus. So entspricht

```
type test.txt
```

der ausführlicheren Notation

```
type test.txt >con
```

MS-DOS erkennt dabei automatisch, daß es für Eingaben die Tastatur und für Ausgaben den Bildschirm verwenden soll. Um die Leistungsfähigkeit der Betriebssystem-Befehle und -Dienstprogramme hervorzuheben, sei hier kurz der Befehl »ctty« erwähnt. Er erlaubt es, die Standardein-/-ausgabeeinheit zu ändern, was für alle Befehle Gültigkeit hat. So lassen sich beispielsweise bei Verwendung von zwei Computern, die Sie über die serielle Schnittstelle mit einem sogenannten »Nullmodemkabel« verbunden und deren Schnittstellenparameter Sie mit dem Mode-Befehl korrekt eingestellt haben, alle Ein- und Ausgaben auf MS-DOS-Ebene von einem anderen Computer aus realisieren:

```
ctty com2:
```

Alle Ein- und Ausgaben werden somit über die zweite serielle Schnittstelle des betreffenden Computers abgewickelt. Ohne speziellere Angaben wird sich dann

```
type test.txt
```

wie

```
type test.txt >com2:
```

verhalten. Der gezielte Einsatz von Gerätenamen erschließt für zahlreiche »alltägliche« Befehle neue Qualitäten.

Datenumleitung und Filter

Eine Art der Datenumleitung, das Größerzeichen (»>«) und das Befehlsverkettungszeichen (»¦«, [Alt-1-2-4]) sind Ihnen bei einigen Befehlen schon begegnet. In Tabelle 3 sind alle denkbaren Datenumleitungen zusammengestellt. Betrachten Sie zuerst das Befehlsverkettungszeichen »¦« genauer. Sie setzen es ein, um gewissermaßen eine Brücke zwischen zwei Befehlen herzustellen. Der einzige zu beachtende Punkt besteht darin, daß ein Befehl, der mit einem weiteren durch »¦« verbunden ist, von dessen »Vorgänger« Information bezieht. Der verkettete Befehl muß also in der Lage sein, die eintreffenden Daten zu bearbeiten, wie dies bei den drei MS-DOS-Filtern »sort«, »more« und »find« der Fall ist. Am Beispiel von »chkdsk« hat sich jedoch gezeigt, daß durchaus auch ungewöhnliche Befehlskonstruktionen bei Befehlsverkettungen Verwendung finden. Später machen Sie bei »echo« und bei der »Erzeugung neuer Befehle« noch weiteren Gebrauch von dieser Technik.
Das bekannteste Datenumleitungssymbol ist »>«, welches die Ausgaben eines Programms auf eine spezifizierte Geräteeinheit oder in eine Datei leitet. Folgt auf das »>«-Symbol (und gegebenenfalls Leerzeichen) keine gültige Gerätebezeichnung, so wird eine Datei dieses Namens im aktuellen Verzeichnis angelegt. Dabei wertet das Symbol Laufwerks- und Pfadangaben bei Vorhandensein ebenfalls aus. Existiert bereits eine Datei dieses Namens, so wird der vorherige Inhalt überschrieben, die Informationen gehen also verloren. Sofern Sie das Anhängen von weiteren Daten an eine schon bestehende Datei wünschen, müssen Sie die Datenumleitung »>>« einsetzen.

```
dir ¦ find ".03.91" > c:\texte\1991.txt
```

leitet alle Dateiangaben des aktuellen Verzeichnisses, in denen das Datum »März 1991« enthalten ist, in die Datei »1991.txt« auf Laufwerk »C:« in das Verzeichnis »texte«. Mit dem Befehl würden Sie eine dort bestehende Datei »1991.txt« allerdings überschreiben.

```
dir ¦ find ".04.91" >> c:\texte\1991
.txt
```

Diese Befehlszeile hängt an die bereits bestehende Datei »1991.txt« alle Dateiangaben des aktuellen Verzeichnisses, in denen das Datum »April 1991« in der Notation des Dir-Befehls (»04.91«) enthalten ist, an.
Die letzte noch ausstehende Datenumleitung betrifft die Festlegung des Dateneingabewegs mit dem Symbol »<«. Alle Eingaben, die sonst über die Tastatur erfolgen würden, werden dann von der so spezifizierten Geräteeinheit oder aus einer Datei erwartet. Wird eine Datei benannt, geht erst nach Dateiende die Kontrolle wieder an die Tastatur über. Sind in der Datei fehlerhafte oder unvollständige Eingabedaten, bleibt das System im Wartezustand. In diesem Fall müssen Sie es durch einen Reset beziehungsweise das Aus- und Wiedereinschalten des Computers neu starten. Eine weitere Einschränkung, die zwar generell alle Datenumleitungen betrifft, aber hier besonders elementar ins Gewicht fällt, besteht in der Art und Weise, wie einzelne Programme Aus- und Eingaben handhaben. Es kann also durchaus geschehen, daß ein von Ihnen erworbenes Hilfsprogramm beispielsweise seine Bildschirmmeldungen nicht umleiten läßt. Auch beim MS-DOS-Programm »format« haben Programmierer einer eventuellen Datenumleitung sinnvollerweise entgegengewirkt, da der Datenverlust enorm sein könnte. Versorgen Sie ein Programm mit »konservierten Tastenanschlägen« aus einer anderen Datei, müssen Sie grundsätzlich höchste Vorsicht walten lassen. Nur in den wenigsten Fällen sollten Sie Programme auf diese Art »fernsteueren«, da die Benutzereingaben üblicherweise ihre vorrangige Berechtigung haben. Eine Ausnahme bilden die beiden Befehle »date« und »time«. Zum Setzen neuer Werte ist die Eingabe des Anwenders erforderlich. Soll jedoch nur eine Ausgabe der aktuellen Informationen erfolgen, können sie normalerweise nicht eingesetzt werden. Gerade bei Stapelverarbeitungsaufgaben ist es jedoch nützlich, wenn Sie automatisch Datum und Uhrzeit in eine Datei schreiben können. Benutzen Sie also hier die Anwendung der Dateneingabe aus einer anderen Datei. Mit »copy con Dateiname« stellen Sie also eine Datei her:

```
copy con enter.txt
```

Deren Inhalt besteht lediglich aus einem [Eingabe]. Zur Speicherung wird [F6] und nochmals [Eingabe] eingegeben. Sie rufen nun »date« und »time« mit

```
date <enter.txt
```

beziehungsweise

```
time <enter.txt
```

auf und verlangen keine manuelle Eingabe mehr. Die Umleitung von »date« und »time« in ein Datei, gefolgt von weiteren Informationen könnte also folgendermaßen aussehen:

```
date <enter.txt >neudat.txt
time <enter.txt >>neudat.txt
dir >>neudat.txt
```

Wer sich an der überflüssigen Zeile stört, kann sie mit dem Filter »find« beseitigen. Sofern es nicht auf Anhieb funktioniert, sehen Sie sich den zuvor schon beschriebenen Find-Befehl noch einmal genauer an.
Außer den schon bekannten Filtern »sort«, »more« und »find«, die alle ihre Eingaben auch direkt aus einer Datei beziehen können, ist der Einsatz der Dateneingabeumleitung besonders beim MS-DOS-Zeileneditor »edlin« und dem Systemhilfsprogramm »debug« (MS-DOS) beziehungsweise »sid« (DR-DOS 5.0) sehr interessant.
Wollen Sie beispielsweise die Datei »test.txt« seitenweise auf dem Bildschirm darstellen, können Sie anstelle von

```
type test.txt ¦ more
```

auch die schnellere Variation

```
more < test.txt
```

verwenden. Bei allen Programmen, die Tastenanschläge aus einer Datei beziehen, ist es erforderlich, daß auch der Befehl zur Rückkehr zum Betriebssystem enthalten ist, denn erst dort geht die Kontrolle wieder an die Tastatur über. Vorher ist keine Eingabe mehr erlaubt.
Trotz einiger Gefahren, die sich durch Vorsicht, Überlegung und einem »Trockentest« mit unwichtigen Daten umgehen lassen, erschließt die Umleitung »<« neue Einsatzgebiete.
Speziell zur Unterstützung von Stapelverarbeitungsaufgaben stellt MS-DOS einige weitere Anweisungen zur Verfügung, die Sie in Tabelle 4 im Gesamtüberblick finden.

»call«
setzen Sie bei MS-DOS ab Version 3.3 ein. Eine Stapeldatei ruft eine weitere zur Befehlsausführung als Unterprogramm auf. Nachdem alle Befehlszeilen in der aufgerufenen Stapeldatei bearbeitet wurden, wird der weitere Ablauf mit der Zeile nach dem Call-Aufruf fortgesetzt. In früheren MS-DOS-Versionen können Sie dies durch »command /c Stapeldateiname.bat« zwar simulieren, allerdings laden Sie damit einen neuen Kommandointerpreter, der weiteren Hauptspeicher und Ladezeit benötigt. »call« findet beispielsweise in Stapeldateien Anwendung, in denen Teile identisch sind. Anstatt diese Anweisungen in jede Batch-Datei erneut aufzunehmen, reicht ein Aufruf mit »call« aus. Der Vorteil besteht weniger in der ersparten Eingabearbeit oder dem Ausschalten von Fehlern, die hierbei entstehen könnten. Vielmehr reicht bei veränderten Umgebungswerten (beispielsweise Laufwerk- und Pfadangaben) eine einmalige Korrektur in einer Datei aus, um alle Stapeldateien, die die entsprechende »bat«-Datei aufrufen mit einem Schlag anzupassen. Im Zusammenhang mit der Befehlskonstruktion »for ... in (...) ... do« wird »call« mit einer weiteren Bedeutung auftreten.

»echo«
Den »Echo-Befehl« verwenden Sie, um Meldungen aus Stapeldateien auszugeben, beispielsweise

```
echo Sortieren der Daten in Arbeit!
```

Die Darstellung erfolgt standardmäßig auf dem Bildschirm, durch Umleitung jedoch auch auf anderen gültigen Geräteeinheiten wie »lptx«, »comx« oder »nul«. Die Technik, einzelne Zeichen oder Zeichenketten an ein bestimmtes Gerät zu senden, ist besonders bei Druckern interessant. Laserdruckerbesitzer kennen sicher das Problem, daß der Drucker die Daten schon empfangen hat, sie aber noch nicht ausgibt (beispielsweise nach einer Bildschirmhardcopy mit der Tastenfolge [Umschalt-Druck]). Ursache dafür ist eine fehlende Steuersequenz, die den Drucker anweisen würde, das Blatt auszuwerfen. Dieser Steuercode (Seitenvorschub) ist das Zeichen [Alt-1-2] oder [Strg-L]. Mit »echo« schicken Sie es nachträglich zum Drucker:

```
echo^L >lpt1:
```

Durch den Zusatz »on« oder »off« schalten Sie die Ausgabe von Systemmeldungen ein oder aus. Auch Filter- und Umleitungssymbole (»|«, »<«, »>« und »>>«) können Sie auswerten. Beim Start einer Stapeldatei nimmt das System immer »echo« als »on« an. Stellen Sie »echo off« ein, bleiben alle in der Stapeldatei folgenden Anweisungen »unsichtbar«. Sie sehen nur noch die Zeile »echo off« am Bildschirm. Wollen Sie diese Meldung auch nicht mehr erhalten, schaffen Sie Abhilfe durch das Voranstellen des Zeichens »@« ([Alt-6-4], erst ab MS-DOS 3.3 verfügbar), welches die Ausgabe der entsprechenden Zeile unterdrückt.

Echo erspart Ihnen mitunter die Eingabetaste

Verwenden Sie lediglich »echo« oder »echo« gefolgt von Leerzeichen, meldet das Betriebssystem den aktuellen Zustand des Schalters. Damit verbunden stellt sich die Frage, wie es machbar wäre, Leerzeilen aus Batch-Dateien auszugeben? Die Lösung besteht in einem weder auf Drucker, noch dem Bildschirm darstellbaren Zeichen: [Alt-2-5-5]. Es erfüllt den gleichen Zweck wie ein Leerzeichen und bringt die MS-DOS-Eingabeprüfung nicht durcheinander.
Exotisch mutet der Einsatz von »echo« bei Befehlsverkettungen an, aber es lohnt sich, auch diesen Winkelzug zu kennen. Manche Programme erwarten eine Benutzereingabe oder ein [Eingabe], wie etwa der Aufruf von »date« und »time«. Sie finden das Beispiel bei der Dateneingabeumleitung »<« genauer beschrieben. Einen weiteren Weg, die Eingabe von [Eingabe] zu simulieren, ohne jedoch eine externe Datei und weitere Vorkehrungen zu benötigen, beschreiten Sie mittels der Eingabe von »echo«:

```
echo.|date
```

beziehungsweise

```
echo.|time
```

Wichtig ist hier, daß Sie nach »echo« und vor dem Befehlsverkettungszeichen »|« keinerlei andere Zeichen, auch kein Leerzeichen verwenden. In dieser Situation nämlich würde außer [Eingabe] noch ein weiteres Zeichen »erzeugt« werden, und die Datum- beziehungsweise Zeiteingabe wäre damit ungültig. Schlimmer jedoch ist der Nebeneffekt, daß die Eingabe wiederholt werden muß. Aufgrund der Befehlsverkettung ist die Tastatur blockiert, und die letzte Rettung, den Computer aus diesem Zustand zurückzuholen, besteht in der Eingabe von [Strg-Untbr].

»for ... in (...) ... do«
Diese sehr leistungsfähige Befehlssequenz dient zur Bearbeitung mehrerer, ausgewählter Dateien oder Aufzählungen mit einem Befehl. Mit »for« spezifizieren Sie eine Umgebungsvariable, die je Durchgang ein weiteres Element der Aufzählung oder eine weitere Datei aufnimmt. »in« regelt die Auswahl der Dateien, wobei die Platzhalterzeichen »*« und »?« oder auch Aufzählungen von Dateien, Laufwerken, Stapel- und Umgebungsparameter erlaubt sind. Der Befehl nach »do« kommt schließlich zur Ausführung. In diesem Teil müssen Sie auch die in »for« benutzte Umgebungsvariable wieder angeben. Wenden Sie »for ... in (...) ... do« in der MS-DOS-Eingabezeile an, steht nur ein einzelnes Prozentzeichen vor dem Variablennamen – innerhalb einer Stapeldatei sind zwei Prozentzeichen erforderlich. Von den Befehlen, die sich lediglich im Batch-Betrieb einsetzen lassen abgesehen, ist dies einer der wenigen Unterschiede zwischen der MS-DOS-Befehlsebene und der Ausführung des gleichen Kommandos in Stapeldateien. Sehen Sie sich ein Beispiel an:

```
for %%a in (*.doc) do type %%a >prn
```

Diese Anweisung gibt alle Dokumentationsdateien (geläufige Endung der Textdateien, die Sie häufig bei englischsprachigen Programmen aus dem Public-Domain-Bereich oder der Shareware vorfinden) des aktuellen Verzeichnisses und Laufwerks mit Endung »doc« auf dem Drucker aus. Auf MS-DOS-Ebene eingegeben, sieht die Zeile wie folgt aus:

```
for %a in (*.doc) do type %a >prn
```

Wie schon erwähnt, lassen sich nach »in« auch Aufzählungen einsetzen:

```
for %%a in (c: d: e: f:) do dir %%a /w
```

beziehungsweise als direkte Eingabe für das Betriebssystem:

```
for %a in (c: d: e: f:) do dir %a /w
```

Durch diese Aufzählung wird nacheinander von den Laufwerken »C:« bis »F:« das Inhaltsverzeichnis ausgegeben. Wie Sie am Schalter »/w« ersehen, können Sie hier sowohl Parameter übergeben als auch Befehlsverkettungen und Datenumleitungen gebrauchen. Vielleicht haben Sie jedoch bei den beiden Beispielen eine potentielle Schwachstelle der Befehlskonstruktion entdeckt: Nach »do« können Sie nur eine Anweisung verwenden. Doch der Schein trügt: »call« betritt wieder die Bühne und leitet entweder den jeweiligen Programmnamen oder das jeweilige Element der Aufzählung

Der direkte Weg zu besseren Programmen



Prüfen Sie Ihr Nachschlagewerk in Ruhe zu Hause. Wenn Sie nicht überzeugt sind, schicken Sie dieses mit den Disketten innerhalb von 10 Tagen an uns zurück. Damit ist die Sache für Sie erledigt.

Lösungen in Turbo-Pascal 4.0, 5.0, 5.5 und 6.0

Dieses neue Nachschlagewerk bietet Ihnen u. a.

- geprüfte Musterlösungen und raffinierte Utilities
- detailliertes Know-how zum Turbo-Pascal-System
- hardwarenahe Programmierung
- Betriebssystemroutinen
- Programmierkurse für optimales Software-Engineering

Aus dem Inhalt:

- **DOS-Systemroutinen,** z. B. Diskettenmonitor, Druckeransteuerung, Verwaltung von Harddisk und Diskette, Schnittstelle zu MS-DOS, Datenschutz…
- **Module für eine grafische Benutzeroberfläche** zur optischen Aufwertung eigener Programme oder zur Programmierung von Systemsteuerungen à la Windows;
- **raffinierte Utilities,** z. B. Schnittstellenutilities (u. a. Turbo-Inliner für TASM und MASM, Lesen von dBase III, III+ und IV-Dateien), Grafikutilities, Dateiverwaltungs-Utilities.
- **Turbo-Pascal-Modulbibliotheken,** u. a. String- und DOS-Routinen (z. B. Dialogverarbeitung im Grafikmodus, Desktop-Steuerung), Mausfunktionen und Interrupts;
- **Finanzbuchhaltung,** inkl. Lagerverwaltung, Musterlösungen für Mathematik, Statistik, Betriebswirtschaft, Grafik…;
- **Turbo-Maustreiber** zur komfortablen Steuerung Ihrer Programme mit der Maus;
- **Hardwarenahe Programme,** u. a. Expanded Memory, Disk-Optimizer, Floppy-Tuning;

Fordern Sie noch heute an:

„Musterprogramme, Utilities und Modulbibliotheken für raffinierte Lösungen in Turbo-Pascal"

Praktischer Ringbuchordner, Format DIN A4, Grundwerk ca. 600 Seiten, inkl. drei 5 ¼"-Disketten, Bestell-Nr.: 2500, Preis: DM 98,–, wahlweise zwei 3 ½"-Disketten (gegen DM 4,– Aufpreis).

Alle 2 – 3 Monate erhalten Sie Erweiterungsausgaben zum Grundwerk mit jeweils ca. 130 Seiten, inkl. Diskette, zum Preis von DM 59,– (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;
- liefert Ihnen praxiserprobte Programme und raffinierte Utilities auf Diskette.

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 5.0; ausführliche Erläuterungen zur Funktion der Systemkomponenten (z. B. E/A-System oder Boot Record); systematischer DOS-Intensivkurs;
- **DOS-Kommandos:** alphabetisch gegliederte Befehlsbeschreibungen mit praktischen Anwendungsbeispielen;
- **Tools:** detaillierte Erläuterungen der DOS-Werkzeuge; mit kompletten Tools (z. B. Texteditor "TE");
- **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"

Praktischer Ringbuchordner, Format DIN A4, Grundwerk ca. 750 Seiten, inkl. zwei 5 ¼"-Disketten, Bestell-Nr.: 5000, Preis: DM 98,–, wahlweise 3 ½"-Diskette (gegen DM 2,– Aufpreis).

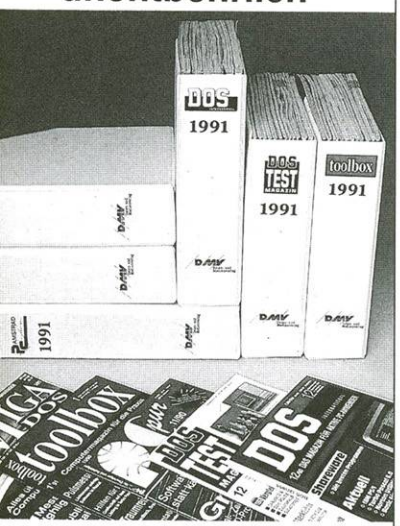
Alle 2 – 3 Monate erhalten Sie Erweiterungsausgaben zum Grundwerk mit jeweils ca. 130 Seiten, inkl. Diskette, zum Preis von DM 59,– (Abbestellung jederzeit möglich).

Römerstraße 16
W-8901 Kissing
Tel. 0 82 33 / 2 11-0
Fax 0 82 33 / 2 11-2 99

INTEREST-VERLAG
Fachverlag für Special Interest Publikationen und Anwendersoftware

Für Ihre Anforderung verwenden Sie bitte nebenstehende Bestellkarte!

Zum Sammeln unentbehrlich



Die DMV-Sammelordner

PC-Software

vom

DMV Versandservice

- leistungsstark
- praxisbewährt
- preisbewußt

Abo-Order DOS International

Hiermit bestelle ich **»DOS International«** für mindestens

☐ **12 Ausgaben** ☐ **6 Ausgaben**

Preise: **BRD**	12 Ausgaben	82,50 DM
	6 Ausgaben	42,– DM
Auslandspreise: **Europa**	12 Ausgaben	110,– DM
	6 Ausgaben	55,– DM
Außereuropäisches Ausland	12 Ausgaben	150,– DM
	6 Ausgaben	75,– DM

Lieferung erfolgt ab der nächsterreichbaren Ausgabe.

Abo-Order DOS DATABOX PLUS

Hiermit bestelle ich **»DOS DATABOX PLUS«** für mindestens

☐ **12 Ausgaben** ☐ **6 Ausgaben**

Diskettenformat: ☐ IBM 5 1/4" ☐ IBM 3 1/2"

Inlandspreise:	12 Ausgaben 300,– DM, 6 Ausgaben 150,– DM
Europa:	12 Ausgaben 320,– DM, 6 Ausgaben 160,– DM
Außereurop. Ausland:	12 Ausg. 360,– DM, 6 Ausg. 180,– DM

Lieferung erfolgt ab der nächsterreichbaren Ausgabe.

Lieferanschrift

☐ Geschäft ☐ Privat

Vorname, Name

Firma (nur wenn Lieferadresse)

Straße, Nr./Postfach

PLZ/Ort *(Bitte genaue Anschrift angeben!)*

Dieses Abonnement verlängert sich automatisch um 6 bzw. 12 Ausgaben, wenn es nicht sechs Wochen vor Ablauf schriftlich gekündigt wird.

Datum 1. Unterschrift (bei Minderjährigen des gesetzlichen Vertreters)

☐ **Bequem und bargeldlos durch Bankabbuchung**

Bankleitzahl

Konto-Nr./-Inhaber

Geldinstitut

☐ **Gegen Rechnung** – zahlbar innerhalb zwei Wochen nach Erhalt. (Bitte keine Vorauszahlung leisten — Rechnung abwarten.)

Bitte unbedingt auf der Vorderseite eine zweite Unterschrift leisten!

PC-Software

5 1/4"	3 1/2"		DM
2282 ☐	2281 ☐	**ConText EASY,**	99,–
2407 ☐	2408 ☐	**ConText PRO,**	199,–
2415 ☐	2416 ☐	**ConText-ADREVA**	49,–
2419 ☐	2420 ☐	**ConText-Zeicheneditor**	99,–
2421 ☐	2422 ☐	**ConText-Rechtschreibhilfe**	69,–
2650 ☐	2651 ☐	**DMV-Power-Pack**	349,–
256 ☐	257 ☐	**PC-VirusDoktor**	99,–
2903 ☐	2904 ☐	**PC-VirusFinder**	149,–
2905 ☐	2906 ☐	**GraphBas PC,** Turbo Basic	99,–
2907 ☐	2908 ☐	**GraphBas PC,** Microsoft Quick Basic	99,–
2522 ☐	2521 ☐	**ERGO 3.0,** DOS-Benutzeroberfläche	199,–
2702 ☐	2703 ☐	**DMV-Show-Manager**	199,–
238 ☐	239 ☐	**MaskEdit Plus** mit einem Treiber	99,–
2613 ☐	2614 ☐	**MaskEdit Plus,** Paket mit zwei Treibern	149,–
2621 ☐	2622 ☐	**MaskEdit Plus,** Paket mit drei Treibern	199,–

Turbo Pascal 3/4/5	Turbo C Microsoft C	Top-Speed Modula-2	Turbo Basic	Quick Basic
☐ 2632	☐ 2633	☐ 2634	☐ 2635	☐ 2636

2720 ☐	2721 ☐	**WinBasic**	299,–
2724 ☐	2725 ☐	**ColoRIX**	299,–
6031 ☐	6032 ☐	**ProfEd**	99,–
227 ☐	233 ☐	**BCI DisAsm/I86,** (MS-DOS)	199,–
6001 ☐	6002 ☐	**RSM-Manager**	69,–
6007 ☐	6008 ☐	**3D-Draw, Version 1.0**	99,–
6017 ☐	6018 ☐	**3D-Draw Professional**	199,–
6019 ☐	6020 ☐	**Navigator 4.5**	69,–
6025 ☐	6026 ☐	**Lightning 3D, 1.5**	199,–

5 1/4"	3 1/2"		DM
6035 ☐	6036 ☐	**Fontline**	299,–
6027 ☐	6028 ☐	**DOS-CAD 3.0**	199,–
2700 ☐	2701 ☐	**DMV-Faktura**	398,–
2710 ☐	2711 ☐	**DMV-Atelier**	249,–
2712 ☐	2713 ☐	**DMV-Atelier** Zusatzfonts	99,–
2714 ☐	2715 ☐	**DMV-Atelier** Symbole	99,–
20506 ☐	20507 ☐	**PC-Software pur,** Festplatten-Optimierer	69,–
20508 ☐	20509 ☐	**PC-Software pur,** MasterDat Datenbank	69,–
20510 ☐	20511 ☐	**PC-Software pur,** MS-DOS Lernprogramm	69,–
___ ☐	___ ☐	**toolbox Spezial Nr.** ___	35,–
		Kurzbez.: ___	
2361 ☐	2362 ☐	**MAUSALL Plus**	35,–
1371 ☐	1372 ☐	**PC-Spiele-Sampler**	49,–
248 ☐	249 ☐	**Fraktal-Generator 3D,** MS-DOS	69,–
1374 ☐	—	**Populous PC**	69,95
1375 ☐	1376 ☐	**Bards Tale II**	39,95
1377 ☐	1378 ☐	**Starflight II**	49,95
1379 ☐	1380 ☐	**Indianapolis 500**	49,95
1381 ☐	1382 ☐	**Oil Imperium**	39,95
1383 ☐	1384 ☐	**Loom**	69,95
1385 ☐	1386 ☐	**Zak McKracken**	69,95
1387 ☐	1388 ☐	**Maniac Mansion**	69,95
1389 ☐	—	**Jet Pack**	169,–
1401 ☐	—	**Digi-Joy**	59,–
1295 ☐	1296 ☐	**Memo**	69,–
1297 ☐	1298 ☐	**Trademaster**	49,–

+ Porto/Verpackung

(Inland 4,– DM, Ausland 6,– DM) ___ DM = Gesamtbetrag: ___ DM

Ihre Anforderungskarte

Bitte abtrennen oder ausschneiden!

Ja, senden Sie mir sofort

(Bitte Zutreffendes ankreuzen)



___ Expl. **„Musterprogramme, Utilities und Modulbibliotheken für raffinierte Lösungen in Turbo-Pascal"**

Grundwerk ca. 600 Seiten, DIN A4, inkl. drei 5 1/4"-Disketten, Bestell-Nr.: 2500, Preis: DM 98,–

___ Expl. **„DOS-Systemprogramierung für IBM-PCs und Kompatible"**

Grundwerk ca. 750 Seiten, DIN A4, inkl. zwei 5 1/4"-Disketten, Bestell-Nr.: 5000, Preis: DM 98,–

Disketten-Service:

☐ Ich wünsche die Disketten im 3 1/2"-Format (DM 2,– Aufpreis pro Diskette)

Zu jedem dieser Grundwerke erhalten Sie alle 2 – 3 Monate Erweiterungsausgaben mit je ca. 130 Seiten, inkl. Diskette, zum Preis von DM 59,– (Abbestellung jederzeit möglich).

Meine Anschrift:

Name

Vorname

Straße, Haus-Nr.

PLZ, Ort

Unterschreiben Sie hier bitte Ihre Bestellung!

Bei Minderjährigen ist die Unterschrift eines gesetzlichen Vertreters erforderlich. Ohne Ihre Unterschrift kann die Bestellung nicht bearbeitet werden.

✘

Datum Unterschrift

Bitte unterschreiben Sie auch Ihre Sicherheitsgarantie,

mit der Sie folgendes zur Kenntnis nehmen: Sie haben das Recht, Ihr angefordertes Werk innerhalb von 10 Tagen ab Lieferung an den INTEREST-VERLAG GmbH, Römerstraße 16, W-8901 Kissing, zurückzusenden, wobei für die Fristwahrung das Datum der Absendung genügt. Sie kommen dadurch von allen Verpflichtungen aus der Bestellung frei.

✘

Datum Unterschrift

DER DIREKTE WE
BESSEREN PROGRA



1. Informieren

DOS-Taschenbücher und DOS EXTRA-Sonderhefte sind eine zu verlässige Informationsquelle fü jeden PC-Besitzer. Nutzen auc Sie dieses Wissen, und vertiefe Sie es mit den dazu angebotene Programmen aus dem Diskette Service. Den PC beherrschen m

- **DOS-EXTRA-Sonderheften**
- **DOS-Taschenbüchern**
- **DOS-Disketten-Service**

Zur Erinnerung:
Meine DOS-Sonderhefte, DOS Taschenbücher, DOS-Disketten wu den bestellt am

PC-Software

Ja, ich mache von Ihrem Angebot Gebrauch und bestelle die umseitig ausgewählten Produkte. Bitte liefern Sie an folgende Anschrift:

Name

Straße/Postfach

PLZ/ORT

☐ Den Betrag bezahle ich mit dem beigefügten Verrechnungsscheck.

☐ Ich bitte um Lieferung per Nachnahme zuzüglich der Nachnahmegebühr (nur innerhalb der BRD).

Datum Unterschrift (bei Minderjährigen des gesetzlichen Vertreters)

Bitte ausreichend frankieren

Antwortkarte

DMV-Verlag
DOS International
Postfach 250

3440 Eschwege

2. Aufrüsten

PC-Software vom DMV-Verlag is unter Kennern bereits ein Begri und auch ein gutes Angebot fü Sie, Ihren PC mit starken und an wendergerechten Applikatione auszustatten. Von der Textve arbeitung bis zum Programmie system, beim DMV-Versand service kaufen Sie günstig ei Den PC konsequent nutzen m PC-Software aus dem DMV-Ve sandservice

- **PC-Anwendungen**
- **PC-Applikationen**
- **PC-Entwicklungssysteme**

Zur Erinnerung:
Meine PC-Software wurde bestellt a

Nutzen Sie unser zusätzliches Angebot:

___ Expl. **„Mehr Erfolg mit dBase II, III, III plus, IV und Weiterentwicklungen"**

Dieses Handbuch liefert Ihnen u. a.:

- kaufmännisches Abrechnungssystem, komplett mit FIBU, Debitoren-/Kreditorenrechnung, Lagerwirtschaftssystem,
- dB-Geschäftsgrafik auf Bildschirm und Drucker,
- dB-Kurse (u. a. Sprachelemente, Benutzeroberflächen, Programmierumgebung),
- Tools und Utilities (z. B. Codeoptimizer dBOPT, dB-Word-Schnittstelle) u.v.m.



Stabiler Ringbuchordner, Format DIN A5, Grundwerk ca. 800 Seiten, inkl. vier 5¼"-Disketten, Bestell-Nr.: 4600, Preis: DM 98,–, wahlweise

☐ zwei 3½"-Disketten (gegen DM 4,– Aufpreis)

Alle 2 – 3 Monate erhalten Sie Erweiterungsausgaben zum Grundwerk mit jeweils ca. 100 Seiten, inkl. Diskette zum Preis von DM 59,– (Abbestellung jederzeit ohne Angabe von Gründen möglich).

60 Pfennig, die sich lohnen!

Postkarte

INTEREST-VERLAG
Fachverlag für Special Interest
Publikationen und Anwendersoftware
z. Hd. Herrn Boos

Römerstraße 16

W-8901 Kissing

IN 110 383

Verlagsgarantie

- Sie erhalten von mir PC-Fachinformationen, von absoluten Spezialisten für Sie geschrieben.
- Ihr Nachschlagewerk können Sie in aller Ruhe 10 Tage zu Hause prüfen und bei Nichtgefallen innerhalb dieser Frist zurücksenden. Der INTEREST-VERLAG gewährt nach Zahlung des Rechnungsbetrages das Recht, die beiliegende Software zu testen und zeitlich unbeschränkt zu nutzen.
- Ihr Nachschlagewerk ist immer aktuell. Dafür sorgt unser Erweiterungsservice.
- Den Erweiterungsservice können Sie jederzeit kündigen. Sie gehen also kein Risiko ein. Darauf gebe ich Ihnen mein Wort.

Ihr INTEREST-VERLAG
Fachverlag für Special Interest
Publikationen und Anwendersoftware

Michael Boos

Michael Boos
Geschäftsführer

DOS INTERNATIONAL

Kompetenz

für

PC-Personalcomputer

im

Abonnement

deutlich

günstiger

DOS INTERNATIONAL

Information

über

PC-Personalcomputer

sammeln Sie mit

- **DOS-Einzelheften**
- **DATABOX PLUS**
  mit mehr als 700 KByte Software und Bonusprogramm

DOS INTERNATIONAL

Vorsprung

für

PC-Personalcomputer

mit

DOS EXTRA

im

Abonnement

Abo-Order DOS International

Hiermit bestelle ich **»DOS International«** für mindestens

☐ **12 Ausgaben** ☐ **6 Ausgaben**

Preise: **BRD**	12 Ausgaben	82,50 DM
	6 Ausgaben	42,– DM
Auslandspreise: **Europa**	12 Ausgaben	110,– DM
	6 Ausgaben	55,– DM
Außereuropäisches Ausland	12 Ausgaben	150,– DM
	6 Ausgaben	75,– DM

Lieferung erfolgt ab der nächsterreichbaren Ausgabe.

Abo-Order DOS DATABOX PLUS

Hiermit bestelle ich **»DOS DATABOX PLUS«** für mindestens

☐ **12 Ausgaben** ☐ **6 Ausgaben**

Diskettenformat: ☐ IBM 5 1/4" ☐ IBM 3 1/2"

Inlandspreise:	12 Ausgaben 300,– DM, 6 Ausgaben 150,– DM
Europa:	12 Ausgaben 320,– DM, 6 Ausgaben 160,– DM
Außereurop. Ausland:	12 Ausg. 360,– DM, 6 Ausg. 180,– DM

Lieferung erfolgt ab der nächsterreichbaren Ausgabe.

Lieferanschrift

☐ Geschäft ☐ Privat

Vorname, Name

Firma (nur wenn Lieferadresse)

Straße, Nr./Postfach

PLZ/Ort *(Bitte genaue Anschrift angeben!)*

Dieses Abonnement verlängert sich automatisch um 6 bzw. 12 Ausgaben, wenn es nicht sechs Wochen vor Ablauf schriftlich gekündigt wird.

Datum 1. Unterschrift (bei Minderjährigen des gesetzlichen Vertreters)

☐ **Bequem und bargeldlos durch Bankabbuchung**

Bankleitzahl

Konto-Nr./-Inhaber

Geldinstitut

☐ **Gegen Rechnung** – zahlbar innerhalb zwei Wochen nach Erhalt. (Bitte keine Vorauszahlung leisten — Rechnung abwarten.)

Bitte unbedingt auf der Vorderseite eine zweite Unterschrift leisten!

DOS-Einzelhefte und DATABOX

Einzelheft Ausgabe 6,50 DM/Stck	DATABOX 5 1/4"-Diskette 24,– DM/Stck	DATABOX 3 1/2"-Diskette 24,– DM/Stck
– 11/87	☐	☐
– 1/88	☐	☐
– 2/88	☐	☐
☐ 3/88	☐	☐
☐ 4/88	☐	☐
☐ 5/88	☐	☐
☐ 6/88	☐	☐
☐ 7/88	☐	☐
☐ 8/88	☐	☐
☐ 9/88	☐	☐
Einzelheft 7,50 DM/Stck		
☐ 10/88	☐	☐
☐ 11/88	☐	☐
☐ 12/88	☐	☐
☐ 1/89	☐	☐
☐ 2/89	☐	☐
☐ 3/89	☐	☐
☐ 4/89	☐	☐
☐ 5/89	☐	☐
☐ 6/89	☐	☐
☐ 7/89	☐	☐
☐ 8/89	☐	☐
☐ 9/89	☐	☐

Einzelheft Ausgabe 7,50 DM/Stck	DATABOX PLUS DATABOX + Bonus 5 1/4" 2 Disketten 24,– DM/Stck	DATABOX PLUS DATABOX + Bonus 3 1/2"-Diskette 24,– DM/Stck
☐ 10/89	☐	☐
☐ 11/89	☐	☐
☐ 12/89	☐	☐
– 1/90	☐	☐
☐ 2/90	☐	☐
☐ 3/90	☐	☐
☐ 4/90	☐	☐
– 5/90	☐	☐
☐ 6/90	☐	☐
☐ 7/90	☐	☐
☐ 8/90	☐	☐
☐ 9/90	☐	☐
☐ 10/90	☐	☐
☐ 11/90	☐	☐
☐ 12/90	☐	☐
☐ 1/91	☐	☐
☐ 2/91	☐	☐
☐ 3/91	☐	☐
☐ 4/91	☐	☐
☐ 5/91	☐	☐
☐ 6/91	☐	☐

☐ Gratis-Sammelordner bei Bestellung von 6 Einzelheften

+ Porto/Verpackung
(Inland 4,– DM, Ausland 6,– DM) _____ DM Gesamtbetrag _____ DM

Abo-Order DOS EXTRA

Hiermit bestelle ich **»DOS EXTRA«** für mindestens

☐ **4 Ausgaben** (vierteljährliche Erscheinungsweise)

Preise: **BRD**	4 Ausgaben 60,– DM
Auslandspreise: **Europa**	4 Ausgaben 70,– DM
Außereuropäisches Ausland	4 Ausgaben 80,– DM

Lieferung erfolgt ab der nächsterreichbaren Ausgabe.

Dauerbezug DOS EXTRA DATABOX

☐ Hiermit bestelle ich **»DOS EXTRA DATABOX«** im Dauerauftrag mit Kündigungsmöglichkeit zum vierteljährlichen Erscheinungstermin.
Achtung: Nur im Bankabbuchungsverfahren möglich!

Preise: **BRD:**	je 39,– DM
Auslandspreise: **Europa:**	je 41,– DM
Außereurop. Ausland:	je 42,– DM

Lieferung erfolgt ab der nächsterreichbaren Ausgabe.
Diskettenformat: ☐ IBM 5 1/4" ☐ IBM 3 1/2"

Lieferanschrift

☐ Geschäft ☐ Privat

Vorname, Name

Firma (nur wenn Lieferadresse)

Straße, Nr./Postfach

PLZ/Ort *(Bitte genaue Anschrift angeben!)*

Dieses Abonnement verlängert sich automatisch um 4 Ausgaben, wenn es nicht sechs Wochen vor Ablauf schriftlich gekündigt wird.

Datum 1. Unterschrift (bei Minderjährigen des gesetzlichen Vertreters)

☐ **Bequem und bargeldlos durch Bankabbuchung**

Bankleitzahl

Konto-Nr./-Inhaber

Geldinstitut

☐ **Gegen Rechnung** – zahlbar innerhalb zwei Wochen nach Erhalt. (Bitte keine Vorauszahlung leisten — Rechnung abwarten.)

Bitte unbedingt auf der Vorderseite eine zweite Unterschrift leisten!

DOS INTERNATIONAL

Abonnement

Rechnungsanschrift

Vorname, Name

Straße, Nr./Postfach

PLZ/Ort

Widerrufsrecht

Ich weiß, daß ich diese Vereinbarung innerhalb einer Woche beim DMV-Verlag, Postfach 250, 3440 Eschwege, schriftlich widerrufen kann, wobei bereits die rechtzeitige Absendung meines Widerrufschreibens zur Fristwahrung ausreicht. Ich bestätige das durch meine zweite Unterschrift.

Datum 2. Unterschrift (bei Minderjährigen des gesetzlichen Vertreters)

Antwortkarte

DMV-Verlag
DOS International
Postfach 250

3440 Eschwege

Bitte ausreichend frankieren

1. Abonnieren

Jeden Monat neu, vermitteln I
nen die DOS International ur
DOS EXTRA wichtige Informati
nen zu Ihrem Personal-Compute
Nutzen Sie dieses Wissen mit e
nem Abonnement zu Ihrem gar
persönlichen

- **Preisvorteil**
- **Informationsvorsprun**
- **Zustellkomfort**

Zur Erinnerung:
Mein persönliches Abonnement wu
de bestellt am

Widerrufsrecht

Ich weiß, daß ich diese Vereinbarung innerhalb ei
ner Woche beim DMV-Verlag, Postfach 250, 344
Eschwege, schriftlich widerrufen kann, wobei bereit
die rechtzeitige Absendung des Widerrufschreiben
zur Fristwahrung ausreicht.

DOS INTERNATIONAL

Einzelhefte

Ja, ich mache von Ihrem Angebot Gebrauch und bestelle die umseitig ausgewählten Produkte. Bitte liefern Sie an folgende Anschrift:

Name

Straße/Postfach

PLZ/ORT

☐ Den Betrag bezahle ich mit dem beigefügten Verrechnungsscheck.
☐ Ich bitte um Lieferung per Nachnahme zuzüglich der Nachnahmegebühr (nur innerhalb der BRD).

Datum Unterschrift (bei Minderjährigen des gesetzlichen Vertreters)

Antwortkarte

DMV-Verlag
DOS International
Postfach 250

3440 Eschwege

Bitte ausreichend frankieren

2. Sammeln

DOS-International-Leser sind in
mer top-informiert. Daher sollte I
nen kein Heft oder Sonderheft
Ihrer Sammlung fehlen. Das Ei
tippen von Listings und Progra
men ersparen Sie sich durc
unseren günstigen Diskette
Service. Immer gut beraten sin
Sie mit DOS International

- **Einzelheften**
- **Sonderheften**
- **Disketten-Service**

Zur Erinnerung:
Meine Hefte und Disketten wurde
bestellt am

DOS INTERNATIONAL

Abonnement

Rechnungsanschrift

Vorname, Name

Straße, Nr./Postfach

PLZ/Ort

Widerrufsrecht

Ich weiß, daß ich diese Vereinbarung innerhalb einer Woche beim DMV-Verlag, Postfach 250, 3440 Eschwege, schriftlich widerrufen kann, wobei bereits die rechtzeitige Absendung meines Widerrufschreibens zur Fristwahrung ausreicht. Ich bestätige das durch meine zweite Unterschrift.

Datum 2. Unterschrift (bei Minderjährigen des gesetzlichen Vertreters)

Antwortkarte

DMV-Verlag
DOS International
Postfach 250

3440 Eschwege

Bitte ausreichend frankieren

3. Schenken

Bei jedem Anlaß ist ein Abonn
ment ein Geschenk, das ankomm
und Ihrem Wunsch nach »gute
Erfolg« Ausdruck verleiht. S
brauchen nur im Feld »Liefera
schrift« die Adresse des B
schenkten einzutragen.

- **DOS International**
- **DOS EXTRA**

Zur Erinnerung:
Mein Geschenk-Abonnement wur
bestellt am

Widerrufsrecht

Ich weiß, daß ich diese Vereinbarung innerhalb ei
ner Woche beim DMV-Verlag, Postfach 250, 344
Eschwege, schriftlich widerrufen kann, wobei bereit
die rechtzeitige Absendung des Widerrufschreiben
zur Fristwahrung ausreicht.

an die spezifizierte Stapeldatei weiter. Wieviele Befehle sich dort dann des Problems annehmen, bleibt dem Anwender überlassen. Sehen Sie sich nochmals das so überarbeitete erste Beispiel (nennen Sie es »test.bat«) mit den zu druckenden Dokumentationsdateien an:

```
for %%a in (*.doc) do call ausdruck.bat
%%a
```

Hierbei wird der externen Stapelverarbeitungsdatei »ausdruck.bat« stets ein neuer Dateiname übergeben, solange ein solcher noch vorhanden ist. Die Batch-Datei »ausdruck.bat« wiederum könnte folgenden Inhalt haben:

```
@echo off
echo.>lpt1:
echo %1 >lpt1:
echo.>lpt1:
type %1 >lpt1:
echo^L >lpt1:
```

In dieser Stapelverarbeitungsdatei finden Sie bekannte Konstruktionen, aber auch eine wichtige Neuerung: »%1« (vom bereits erwähnten Zeichen »@« zur Unterdrückung der Ausgabe der entsprechenden Zeile einmal abgesehen). »%1« steht für den ersten übergebenen Parameter. Da Sie »ausdruck.bat« mit einem Parameter aufrufen wollen (in unserem Beispiel die jeweilige Doc-Datei), können Sie diesen in eine Stapelverarbeitungsdatei übernehmen und weiterverwenden. Mit dieser Technik lassen sich Batch-Dateien so flexibel gestalten, daß sie für beliebige Dateien, beliebige Pfade und Laufwerke gültig sind. Insgesamt können Sie zehn verschiedene Stapelparameter einsetzen, von »%0« bis »%9«. »%0« ist im eigentlichen Sinne kein übergebener Parameter, sondern enthält den Namen der Stapeldatei selbst, in unserem Beispiel also »ausdruck.bat«. Somit stehen Ihnen noch neun Parameter zur Instruktion der Stapeldatei zur Verfügung, für eine normale Anwendung mehr als genug. Unser Beispiel unterdrückt zuerst das Ausgeben der Befehlszeilen der Stapeldatei, dann eine Leerzeile, den Namen der Doc-Datei, eine weitere Leerzeile, schickt daraufhin den Inhalt der Dokumentationsdatei an den Drucker und erzwingt schließlich einen Seitenvorschub. Das Verfahren funktioniert auch bei einem Matrixdrucker in gleicher Form. Der Aufruf von »test« erledigt somit für Sie den Ausdruck aller Doc-Dateien mit Ausgabe des Namens.

»goto« – in Stapeldateien

Dieser Befehl setzt die weitere Bearbeitung an der Position einer Sprungmarke (Label) fort. Sprungmarken sind Namen, denen ein Doppelpunkt vorangestellt ist. Eine Stapeldatei muß demnach nicht »von oben nach unten« (sequentiell) durchlaufen werden, sondern kann Teile überspringen. In Verbindung mit »if ...« gestalten Sie so Batch-Dateien.

```
goto batch_ende
echo Diese Zeile wird nicht verwendet!
:batch_ende
echo Dies ist die erste Zeile der weiteren
echo Stapelausführung
```

Sie können leicht durch Eingabe dieser Zeilen nachvollziehen, wie nur die zweite und dritte Echo-Zeile am Bildschirm auftauchen.

»if«

»if« ist für die bedingte Ausführung von Befehlszeilen verantwortlich und führt einen Befehl nur dann aus, wenn die spezifizierte Bedingung erfüllt ist. Mit den Zusätzen »not«, »errorlevel« und »exist« können Sie auch das Nichtvorhandensein einer Bedingung oder die Existenz einer Datei als Verzweigungskriterium verwenden. »errorlevel« verwenden Sie, um Fehlercodes einiger Programme (beispielsweise »xcopy« oder »replace«) in der nächsten Zeile der Batch-Datei auszuwerten. Übergibt ein Programm Errorlevel-Werte, so geht daraus der Status der Beendigung (»0« für Befehl erfolgreich ausgeführt) bis hin zu Werten, die eine bestimmte Art von Fehlern umschreiben (zum Beispiel »Datei nicht gefunden«), hervor. Diese Errorlevel-Werte sind von Programm zu Programm unterschiedlich, sofern sie überhaupt vorhanden sind. Aus Ihrem MS-DOS-Handbuch sowie den Dokumentationen Ihrer Hilfs- und Zusatzprogramme entnehmen Sie, ob und wie diese Errorlevel-Werte an eine Stapeldatei zurückgeben.

Errorlevel, die Schnittstelle zu den Anwendungen

Eine Eigenart der Errorlevel-Abfrage betrifft die Reihenfolge der Auswertung. Zuerst müssen Sie die höchsten Ergebniswerte abfragen, dann die jeweils niedrigeren. Nach soviel Theorie ein paar praktische Beispiele:

```
if not exist *.bat echo Keine Stapeldateien vorhanden
```

Befindet sich hierbei im aktuellen Verzeichnis und auf dem aktuellen Laufwerk keine Stapeldatei mit der Dateiendung »bat«, erhalten Sie die angegebene Echo-Meldung. »not« und »exist« können Sie – wie gezeigt – auch gemeinsam verwenden, gleiches gilt auch für »not« und »errorlevel«. Bei »copy«, »ren« oder »del« ist es beispielsweise sinnvoll, die Existenz einer Datei zu prüfen, um Fehler auszuschalten oder Daten vor dem ungewollten Überschreiben zu bewahren.
Eine weitere Spezialität des If-Befehls sind Zeichenkettenvergleiche mittels »= =«. Außer der Prüfung von Stapelparametern oder Umgebungsvariablen (siehe hierzu »set«) eignet sich »= =« besonders zur Prüfung, ob bestimmte Parameter an die Stapeldatei übergeben wurden:

```
if "==%1" goto nopara
:nopara
echo Aufrufsyntax: Programm Parameter
```

Die Stapeldatei erkennt also, daß der erste Parameter leer ist (also keine übergeben wurden) und teilt dann die korrekte Art des Aufrufs mit. Nutzen Sie dokumentierende Textausgaben selbst in Stapeldateien nur für Ihre eigenen Zwecke, denn irgendwann vergißt man bei einfachen Batch-Dateien, wieviel Parameter in welcher Reihenfolge Anwendung finden. Kombiniert mit Sicherheitsabfragen (»ist die Datei wirklich korrekt kopiert, bevor sie an anderer Stelle gelöscht wird« und so weiter) schieben Sie so Datenverlusten, die bei manueller Eingabe erfolgt wären, einen Riegel vor.
Ein Beispiel zur Errorlevel-Auswertung (am Beispiel »replace«):

```
if not errorlevel 0
echo Fehler bei der Bearbeitung aufgetreten!
```

Natürlich sollten Sie jeden Fehlerwert – vom größten bis zum kleinsten – abfragen und möglichst mit alternativen Befehlszeilen behandeln. Da »backup«, »xcopy«, »replace« und einige andere Dienstprogramme zahlreiche Errorlevel-Werte zurückgeben, wurde hier darauf verzichtet.

»pause«

»pause« wartet auf Ihren Tastendruck. Sie können »pause« dazu verwenden, eine Meldung auszugeben. Mit diesem Befehl halten Sie beispielsweise eine Stapeldatei an, damit der Anwender die Verarbeitung mit [Strg-C] oder [Strg-Untbr] abbrechen kann. Verwenden Sie eine weitere Meldung nach »pause«, erscheint in der Ausgabe »pause« Meldungstext. Das setzt allerdings voraus, daß Sie »echo« nicht ausgeschaltet haben. Es ist daher weitaus eleganter, »pause« nur seiner Urbestimmung gemäß zu verwenden, und Textausgaben mit »echo« vorzunehmen:

```
echo Abbruch mit [Strg C]
echo jede andere Taste zum fortfahren!
pause >nul
```

Ohne Umleitung nach »nul« würde »pause« die Meldung »Wenn bereit, eine beliebige Taste betätigen!« ausgeben.

»rem« – Remark

Die Bemerkung »rem« verwenden Sie sinnvollerweise nur in Stapeldateien, die dabei den Rest der Befehlszeile als Kommentar interpretieren und dadurch nicht bearbeiten. Wichtige oder komplizierte Befehlszeilen sollten Sie dokumentieren, was nur wenige Sekunden in Anspruch nimmt. Spätere Erweiterungen oder Korrekturen an der Batch-Datei lassen sich dann auch

ohne vollständige Kenntnis der gesamten Stapeldatei ausführen.

»shift« – nur in Stapeldateien
»shift« rückt alle bei Aufruf in der Befehlszeile übergebenen Parameter um eine Position auf. Parameter 2 wird somit Parameter 1 und so weiter. Auf diese Weise können Sie mehr als zehn Parameter übergeben und bearbeiten. In der Praxis finden Sie diesen Befehl höchst selten.

»@« – nur in Stapeldateien
Als erstes Zeichen einer Befehlszeile unterdrückt dies die Ausgabe der Zeile, selbst wenn Sie »echo« auf »on« gesetzt haben (ab MS-DOS 3.3). Dies ist vor allem bei der Zeile »echo off« wichtig; es lassen sich trotz eingeschaltetem »echo«, was für die gesamte Stapeldatei gilt, einzelne Zeilen ausblenden.

Sie schaffen »neue« Befehle

Die bisherigen praxisorientierten Befehlsbeschreibungen sollten zum einen die jeweiligen Beispiele verdeutlichen, zum anderen einige kleinere Standardprobleme lösen. Kombinieren Sie verschiedene Befehle, Parameter und Techniken, die in diesem Beitrag beschrieben sind, miteinander, ergeben sich teilweise überraschende Ergebnisse.
Eine Stapeldatei zum Finden von Programmen in einem beliebigen Verzeichnis eines Laufwerks (hier als »finde.bat« bezeichnet) könnte wie folgt aussehen:

```
@echo off
if "==%1" goto nopara
if "==%2" goto nopara
chkdsk %1/v ¦ find "%2"
goto ende
:nopara
echo Aufruf: Finde Laufwerk: gesuchter Programmname oder Teil davon (Großschrift)
:ende
```

Die Batch-Datei »finde« hat den großen Vorteil, auch Teile von Programmnamen zu erkennen. Suchen Sie demnach »test«, findet die Stapeldatei sowohl »test1« und »test34« als auch »progtest«. Soll »finde« tatsächlich nur Dateien ausweisen, die mit der gesuchten Bezeichnung beginnen, reicht es »\« ([Alt-9-2]) voranzustellen. Anführungszeichen sind nicht erforderlich, da sie schon in der Stapeldatei enthalten sind. »%2« wird dabei innerhalb der Anführungszeichen beim Find-Befehl ausgewertet. Die gesuchte Dateibezeichnung müssen Sie allerdings in Großschrift eingeben, was auf den Ausgabemodalitäten von »chkdsk« beruht. Ein anderer Nachteil dieser Stapeldatei sei nicht verschwiegen: In Netzwerkumgebungen oder bei Laufwerken, die Sie mit »assign« oder »subst« umgeleitet haben, können Sie »find« aufgrund dieser Beschränkungen von »chkdsk« nicht einsetzen.
Das nächste »Programm« heißt »touch.bat«. Es »berührt« die gewünschten Dateien des aktuellen Verzeichnisses und versieht sie mit dem aktuellen Datum sowie Zeiteintrag:

```
@echo off
if "==%1" goto nopara
for %%a in (%1) do copy /b %%a+,,
goto ende
:nopara
echo Aufruf: Touch [Laufwerk:][\Pfad\]Dateiname
:ende
```

Als Wert für »in« finden Sie in diesem Beispiel den Stapelparameter »%1«. Sie könnten auch mehrere Stapelparameter und/oder Umgebungsvariablen (siehe Set-Befehl) aufführen. Eine weitere Verbesserung der Batch-Datei könnte in einer Abfrage bestehen, ob überhaupt die gewünschten Dateien vorhanden sind:

```
if not exist %1 goto nofiles
:nofiles
echo Keine Datei entspricht dem Suchmuster %1!
```

Manchmal wollen Sie nur alle Verzeichnisse, nicht jedoch einzelne Dateien zur Anzeige ausgeben (»findedir.bat«):

```
@echo off
if "==%1" goto nopara
dir %1 ¦ find "<DIR>"
goto ende
:nopara
echo Aufruf: Findedir [Laufwerk:][\Pfad]
:ende
```

Maßgeschneiderter Bedienungskomfort

Ein nach Größe sortiertes Verzeichnis gibt »dirsize.bat« aus:

```
@echo off
if "==%1" goto nopara
dir %1 ¦ find /v "<DIR>" ¦ find "." ¦ find /v " 0" ¦ sort /+14
goto ende
:nopara
echo Aufruf: Dirsize [Laufwerk:][\Pfad]
:ende
```

Für diesen Anwendungsfall enthält die Verzeichnisausgabe zahlreiche Informationen, die lediglich verwirren. Alle Verzeichnisse filtert »dirsize« zuerst aus (»find /v "<DIR>"«), dann alle Teile, die keine Einzeldateiangaben betreffen (die ersten beiden sowie die letzte Ausgabezeile des Dir-Befehls). Letzteres erreicht man dadurch, daß Sie zumindest ein Zeichen finden, wo Sie beide Teile voneinander unterscheiden können. Das Zeichen ».« erfüllt diese Voraussetzung, denn es ist bei allen Einzeldateiangaben im Datum enthalten, jedoch in keiner der »normalen« Textzeilen bei der Darstellung des Inhaltsverzeichnisses. Doch auch damit sondert ein weiterer Filter noch alle Dateien ohne Inhalt heraus (»find /v " 0"). Nach dem Anführungszeichen müssen mindestens vier Leerstellen folgen, da sonst auch Zeiteinträge zwischen Mitternacht und »0:59« entfallen. Die nun noch verbliebene Untermenge sortieren Sie aufsteigend ab der vierzehnten Spalte (Bereich der Dateigröße) (»sort /r ...« für fallend). Prinzipiell die gleiche Technik wenden Sie auch bei »direxT.bat«, »dirname.bat« und »dirdate.bat« an. »direxT.bat« sortiert die Ausgabe nach Programmnamen (ab Spalte 10)

```
@echo off
if "==%1" goto nopara
dir %1 ¦ find /v "<DIR>" ¦ find "." ¦ find /v " 0" ¦ sort /+10
goto ende
:nopara
echo Aufruf: DirexT [Laufwerk:][\Pfad]
:ende
```

»dirname.bat« nach dem Programmnamen selbst (»sort« ohne Zusätze)

```
@echo off
if "==%1" goto nopara
dir %1 ¦ find /v "<DIR>" ¦ find "." ¦ find /v " 0" ¦ sort
goto ende
:nopara
echo Aufruf: Dirname [Laufwerk:][\Pfad]
:ende
```

und schließlich »dirdate« nach dem Dateierstellungs- beziehungsweise -änderungsdatum (ab Position 24 in der Verzeichnisausgabe):

```
@echo off
if "==%1" goto nopara
dir %1 ¦ find /v "<DIR>" ¦ find "." ¦ find /v " 0" ¦ sort /+24
goto ende
:nopara
echo Aufruf: Dirdate [Laufwerk:][\Pfad]
:ende
```

Theorie durch Praxis

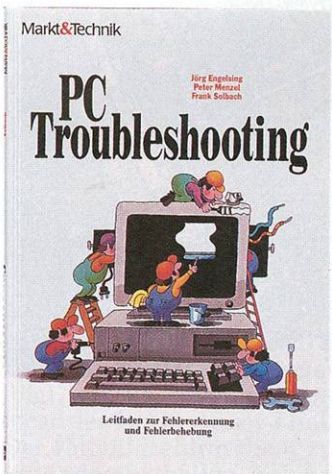
Wenn Sie diese Batch-Beispiele in ein bei »path« aufgeführtes Verzeichnis kopieren, stehen sie Ihnen generell bei der Arbeit zur Verfügung. »edlin« und »debug« (beziehungsweise »sid« bei DR-DOS 5.0) sind weitere lohnende Objekte der Beschäftigung. Mit einer »for ... in (...) ... do«-Konstruktion, »edlin« und einer Eingabeumleitung durchsuchen Sie beispielsweise alle Texte eines Verzeichnisses nach einem Namen und ersetzen ihn gegen einen anderen. Das Feld der Anwendungen ist schier unbegrenzt und die Zeit, die Sie für die Beschäftigung mit dem Betriebssystem verwenden, zahlt sich schon bald aus.
Sicherlich finden Sie auch im Laufe der Zeit zusätzliche Hilfsprogramme, die Ihre Stapeldateien bereichern und Ihnen zu Ihrer Arbeit Nutzen, Freude und Verständnis für die Zusammenhänge bringen.

(Herbert Buckel/et)

Troubleshooting

Natürlich würde man gerne die Schwierigkeiten mit dem PC auf den Mond schießen. Statt einer Rakete haben sich die Autoren die gebräuchlichsten Anwender- und Materialfehler in 14 Kapiteln vorgenommen.

Ein bißchen zäh der Anfang – »das unbekannte PC-Wesen« – und allzu locker: »Sollte ein Chip einmal defekt sein, kann dieser einfach ausgetauscht werden«. Instruktiver ist dann schon die »präventive Wartung« der Gerätschaften wie Vermeidung von Staub, Frequenzstörungen, statischer Aufladung und ein Erste-Hilfe-Köfferchen für den Notfall. Mit diesen Utensilien geht es dann erst mal ans Putzen.



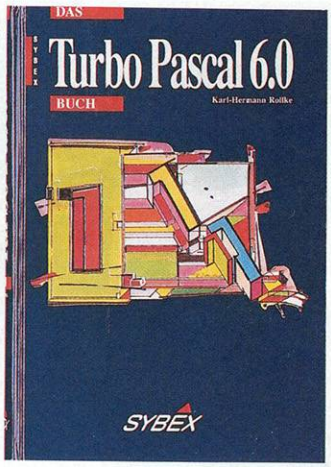
Die Fehlersuche beginnt mit einer Typologie der Fehler anhand jeweiliger Beispiele sowie von Diagnoseprogrammen. Im einzelnen werden untersucht: Netzteil, Bildschirm, Festplatte, Diskettenlaufwerke, Tastatur, Drucker, Modems und serielle Schnittstellen sowie Erweiterungskarten und Chips.

Den Schluß zieren die IBM-Fehlercodes beim Selbsttest, ein unvermeidliches Glossar und ein Stichwortverzeichnis.

Das Buch ist mit den notwendigsten Abbildungen versehen, ansprechend formuliert, kompetent geschrieben, wenn es um Hardware- oder Betriebssystemangelegenheiten geht; doch die häufig auftretenden Softwareprobleme sind nur unzureichend geschildert.

(al)

J. Engelsing, P. Menzel, F. Solbach: PC-Troubleshooting. Leitfaden zur Fehlererkennung und Fehlerbehebung, Markt & Technik 1990, 194 Seiten, ISBN 3-89090-893-4, 49 Mark



Turbo Pascal 6.0

Wie für produktbezogene Bücher üblich, beginnt auch dieses mit der Installation des Systems sowie dessen Handhabung. Anschließend folgt eine Beschreibung des generellen Aufbaus eines Turbo-Pascal-Programms. Hier stellt der Autor die elementaren Bestandteile wie Konstanten, Variablen, Unterprogramme, Ein-/Ausgabefunktionen und reservierte Wörter von Turbo Pascal vor. An dieser Stelle trifft der Leser auch erstmals auf Syntaxdiagramme (auch Backus-Naur-Formalismus oder kurz BNF genannt), einer grafischen Darstellungsform von Sprachelementen, mit der sich Begriffe besser erläutern lassen. In den folgenden Kapiteln geht der Autor ins Detail. Bestandteile der Programmiersprache Pascal wie Datentypen, Schleifen, bedingte Anweisungen, Prozeduren und Funktionen, Felder, Verbunde und Listen beschreibt er auf leicht verständliche Weise. Ein Kapitel widmet sich dem ab der Version 6.0 in Turbo Pascal integrierten Assembler, der es erlaubt, Assembler-Anweisungen direkt in den Pascal-Quelltext einzufügen. Neben dem Assembler ist Turbo Vision die bedeutende Neuerung der Version 6.0. Diese grafische Oberfläche beschreibt der Autor im Vergleich zu anderen Büchern, die dieses Thema meist nur anschneiden, recht ausführlich. Borland bietet ergänzende Toolboxen an, beispielsweise die Database-Toolbox, der der Autor ein eigenes Kapitel widmet.

(rm)

Karl-Hermann Rollke: Turbo Pascal 6.0, Sybex-Verlag 1991, 479 Seiten, ISBN 3-88745-890-7, 49 Mark

Programmierkurs für Turbo Pascal 6.0

Dieses Buch ist vor allem für den Pascal-Einsteiger zu empfehlen, da es schrittweise in die Programmiersprache Turbo Pascal 6.0 einführt. Das erste Kapitel beginnt mit Installationsproblemen. Sie lernen die wichtigsten Pascal-Menüs und deren Befehle sowie die Bedienung der Benutzeroberfläche des integrierten Editors kennen. Wichtiges Turbo-Pascal-Grundwissen bekommen Sie, angefangen von den erlaubten Datentypen bis zu den gängigen Grafikroutinen, in den Kapiteln zwei bis dreizehn vermittelt.



Über Grundlagen der objektorientierten Programmierung informieren Sie sich in dem darauffolgenden Kapitel. Das eigentlich Neue an Turbo Pascal 6.0 stellt die Programmbibliothek Turbo Vision dar. Mit diesem objektorientierten Gerüst entwickeln Sie relativ einfach Anwenderprogramme. Wie diese Programmentwicklung im einzelnen abläuft, erfahren Sie in Kapitel 15. Wesentliches zur effektiven Fehlersuche in Turbo-Pascal-Programmen erläutert Ihnen der Autor in den nachfolgenden Lektionen.

Dieses Buch ist sehr gut gegliedert und übersichtlich gestaltet. Zahlreiche Programmierbeispiele erleichtern Ihnen den Einstieg in diese Programmiersprache. Ein weiterer Vorzug besteht im Nachschlagewerk am Ende des Buches. Dort finden Sie alle wichtigen Grundlagen von Turbo Pascal 6.0 noch einmal zusammengefaßt.

(Kerrin Uhlmann/db)

Manfred Füssinger: Programmierkurs für Turbo Pascal 6.0, Sybex-Verlag 1991, 403 Seiten, ISBN 3-88745-941-5, 39 Mark



EDV-Grundlagen

Dieses Buch stellt eine allgemeine Einführung in die elektronische Datenverarbeitung auf Mikrocomputern dar. Es ist besonders für diejenigen geeignet, die sich erstmals mit einem Rechner auseinandersetzen.

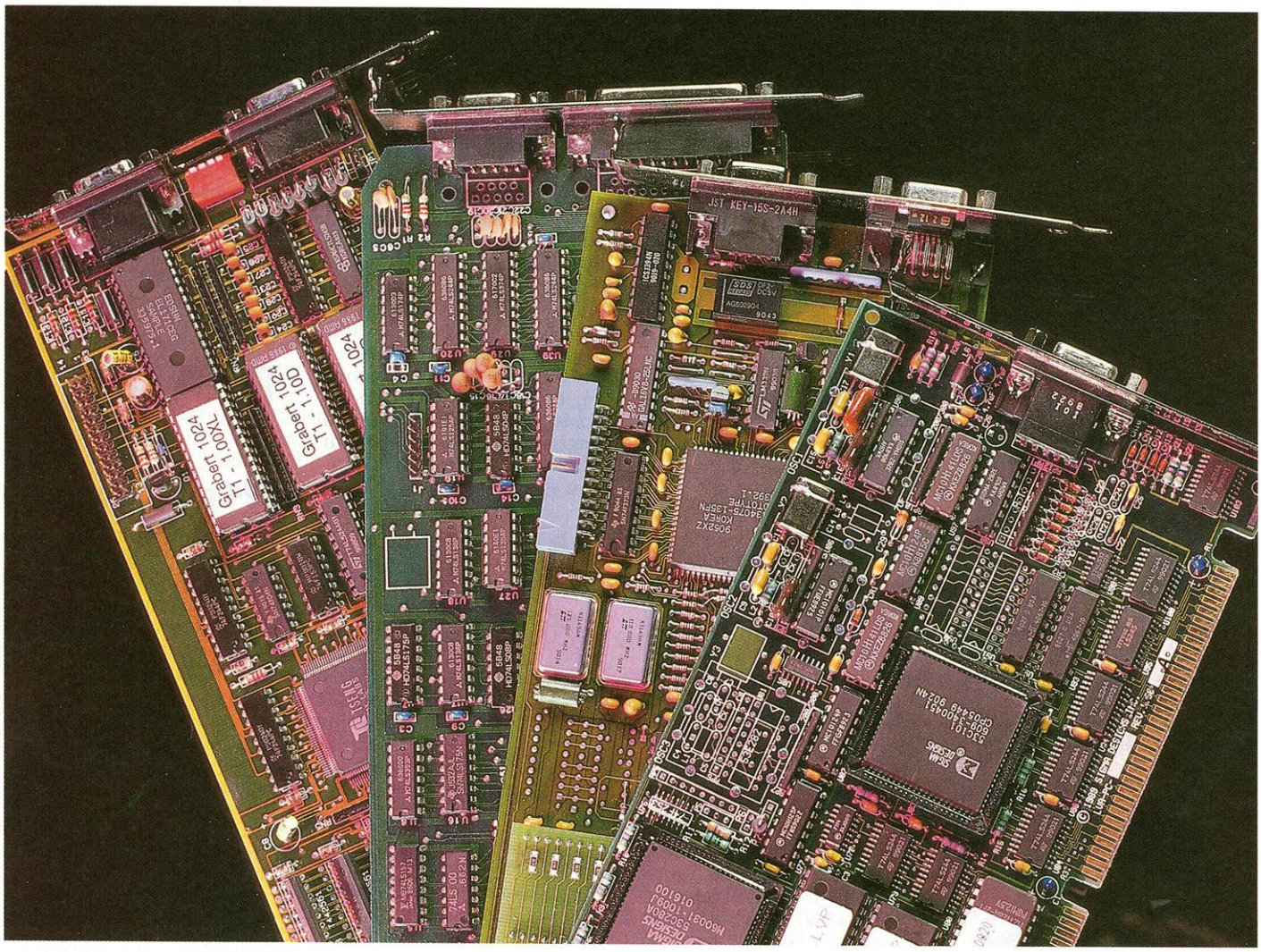
Nach einer kurzen Einführung im ersten Kapitel behandelt der Autor im zweiten ausführlich die Hardware. Dort erfahren Sie alles über die notwendigen und gebräuchlichen Hardwarekomponenten eines Rechners.

In Kapitel 3 informieren Sie sich über Funktion und Arbeitsweise des gängigsten Betriebssystems MS-DOS. Es beinhaltet ebenfalls eine Kurzbeschreibung aller notwendigen und wichtigen MS-DOS-Befehle. Die folgenden Kapitel 4 bis 9 befassen sich mit der Datenorganisation, der Datendarstellung und der Datensicherheit. Sie bekommen Grundkenntnisse der Datenverarbeitung vermittelt, die Sie für die DV-Planung und Programmierung benötigen. Zum Schluß macht Sie der Autor noch mit den Fragen der Datenfernübertragung und des Bildschirmtextes bekannt.

Ein umfangreiches Fachwort- und Stichwortverzeichnis am Ende dieses Buches runden den guten Eindruck ab. Das Werk ist wie ein Lehrbuch aufgebaut. Dieses Konzept wird noch durch die vielen grafischen Darstellungen und zahlreichen Übungsaufgaben unterstützt. Dadurch bietet es Ihnen eine gute Grundlage für das Selbststudium.

(ba)

Richard Pongratz: EDV-Grundlagen, IWT-Verlag 1990, 8011 Vaterstetten, 506 Seiten, ISBN 3-88322-202-X, 68 Mark



Grundlagen Grafikstandards

Bunt oder nicht, hier keine Frage

Wer sich mit Grafikformaten bei Personalcomputern beschäftigt, stößt auf die vier Begriffe »Hercules«, »CGA«, »EGA« und »VGA«. Wir erklären Ihnen die Unterschiede der einzelnen Formate und weisen auf Kriterien hin, die beim praktischen Einsatz zu beachten sind.

Um sich im Dschungel der unterschiedlichen Grafikstandards zurechtzufinden, bedarf es einiger Fachkenntnis. Gerade bei den neueren VGA-Karten fällt es schwer, im Wirrwarr der verschiedenen Bildschirmformate die Übersicht zu behalten.

In Tabelle 1 sehen Sie gängige Grafikstandards mit den üblichen Auflösungen. Dabei kommt es auch auf den passenden Monitor an. Tabelle 2 zeigt eine Aufstellung der Grafikkarten und der jeweils in Frage kommenden Monitortypen. Die bekannteste aller Grafikkarten, wenn auch nicht mehr ganz dem Stand der heutigen Technik entsprechend, ist ohne Zweifel die Hercules-Karte.

Hercules – ein Held kommt in die Jahre

Aufgrund ihres sehr niedrigen Preises wird sie inzwischen bei fast allen Computern als Grundausstattung mitgeliefert. Es handelt sich dabei um eine Monochromkarte, das bedeutet, daß sie keine Farben darstellen kann. Ihre maximale Auflösung beträgt 720x348 Bildpunkte (Bild 1). Dies läßt bereits erkennen, daß damit eine klare Darstellung von sowohl Text als auch Grafik möglich ist. Leider wird ihre Grafikfähigkeit nicht von der Programmiersprache GW-Basic, über die beinahe jeder Anwender verfügt, unterstützt.

Im Textmodus erlaubt sie zwei Helligkeitsstufen (volle und halbe Intensität), was das Hervorheben von Textteilen erlaubt.

Beim Grafikbetrieb ist zu beachten, daß die benutzte Software auch den Hercules-Betrieb zuläßt. Die dabei er-

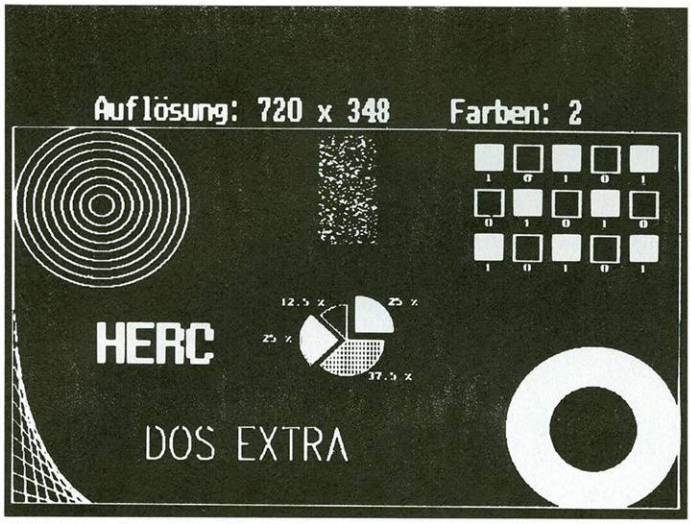


Bild 1. Die Hercules-Grafik kennt zwar keine Farben, dafür überzeugt die Textdarstellung

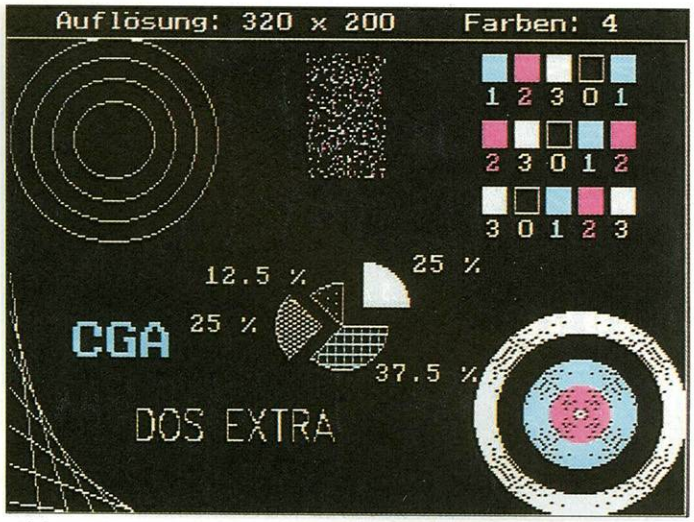


Bild 2. Heute einfach nicht mehr überzeugend, gehört die CGA-Karte bereits zum alten Eisen

zielten Ergebnisse sind für eine »nur« Schwarzweißdarstellung jedoch beachtlich.

Es gibt mittlerweile einige Emulatoren, teilweise auch im PD- oder Shareware-Vertrieb, welche erlauben, auch Programme, die eine CGA-Karte benötigen, mit der Hercules-Grafikkarte laufen zu lassen.

Eine wichtige Stellung nimmt sie auch beim Zwei-Bildschirm-Betrieb ein. Programme wie beispielsweise Turbo Pascal oder Turbo C sind so in der Lage, auf einem Bildschirm das Programm im Schrittmodus (Trace) laufen zu lassen, während auf dem zweiten Monitor die Bildschirmausgabe erfolgt. Auch CAD-Programme wie Autocad laufen in dieser Konfiguration. Der Farbbildschirm dient zur Darstellung der aktuellen Zeichnung. Die Menüoberfläche sieht man auf dem Hercules-Monitor.

CGA – fast schon vergessen und doch noch benutzt

Die Color-Grafikkarte führt heute nur noch ein Schattendasein und kann nicht mehr empfohlen werden. Nennenswerte Gründe dafür sind sehr schlechte Auflösung im Textmodus sowie nur maximal vier Farben im Grafikmodus (Bild 2). Mit ihr können 40 Zeichen bei 320x200 Bildpunkten (Bild 2) oder 80 Zeichen mit 640x200 Bildpunkten pro Zeile (Bild 3) dargestellt werden.

Im Grafikmodus stehen in der niedrigen Auflösung nur vier fest vorgegebene Farben zur Verfügung. Dies sind je nach verwendeter Farbpalette entweder Violett, Türkis, Weiß und Schwarz oder Gelb, Grün und Schwarz. Bei der hohen Auflösung ist sogar nur eine einfarbige Darstellung gegeben.

Die CGA-Farbgrafikkarte findet allenfalls noch in Laptops Verwendung. Es sind in diesem Fall, sofern ein externer Monitor angeschlossen werden kann, meist 16 Farben bei 640x200 Bildpunkten darstellbar. Dies wird durch eine Erweiterung des ursprünglichen Standards realisiert. Auch die 1512er Reihe der Firma Schneider kann diese Auflösung darstellen.

Eine Kombination von Hercules- und CGA-Karte, die sogenannte AGA-Karte (Advanced Graphic Adapter) wurde fast ausschließlich von Commodore vertrieben. Für sie gibt es auch eigene Monitore, die mit beiden Grafikstandards zurechtkommen.

EGA – ein Schritt in die richtige Richtung

Als eine Zwischenstufe in der Entwicklung der Farbgrafikkarten darf man die EGA-Karte bezeichnen. Mit ihr wurden erstmals in einer akzeptablen Auflösung im Text- und vor allem auch im Grafikmodus maximal 16 Farben bei 640x350 Bildschirmpunkten dargestellt. (Bild 4) Aber auch die niedrigeren Farbgrafikauflösungen werden durch sie zusätzlich unterstützt.

Viele Firmen entwickelten die EGA-Karte weiter, so daß damit noch höhere Auflösungen machbar sind; jedoch meist nur in Verbindung mit einem sogenannten Multifrequenz-Monitor, der die verschiedenen Darstellungsformate verarbeiten kann.

Durch den großen Hardwareaufwand läßt sich auch der im Verhältnis hohe Preis der EGA-Kombination »Bildschirm und Karte« erklären. Sie ist deshalb heute als Neuanschaffung nicht mehr zu empfehlen, da es bereits für den annähernd selben Preis VGA-Pakete gibt.

Als Standard ist sie jedoch nach wie vor von Bedeutung, da viele Programme entwickelt wurden, die nur in EGA-Auflösung arbeiten.

Auch tritt dabei wieder das Problem der notwendigen Bildschirmtreiber auf. Nur was die Treibersoftware kennt, kann auch dargestellt werden. Leider beschränkt sich diese meist auf die sogenannten Standardanwendungen wie Autocad, Wordstar, GEM, Windows oder den Ventura Publisher.

Zum Normalfall, bei der Neuanschaffung eines PC, darf man heute die VGA-Karte zählen. Sie hat die Hercules-Karte, obwohl die meisten Firmen diese nach wie vor im Grundpaket anbieten, klar vom Markt verdrängt. Manche VGA-Karten erlauben es sogar, Hercules-Grafik auch auf VGA-Bildschirmen zu emulieren.

Standard	Auflösung	Farben	Bemerkung
Hercules	720x348	monochrom	2 Helligkeiten
CGA	320x200	Farbe	maximal 4 Farben
	640x200	monochrom	teilweise 16 Farben
EGA	640x350	Farbe	maximal 16 Farben
VGA	320x200	Farbe	maximal 256 Farben
	640x480	Farbe	VGA-Standard
Super-VGA	800x600	Farbe	maximal 256 Farben
	1024x768	Farbe	maximal 256 Farben

Tabelle 1. Übersicht gängiger Auflösungen der einzelnen Grafikstandards

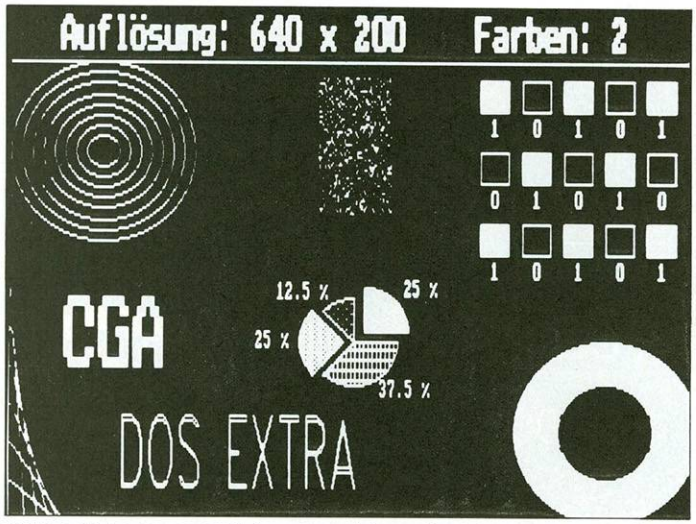


Bild 3. Bei 640x200 Bildpunkten bietet die CGA-Karte zwar fast Hochauflösung, allerdings ohne Farben

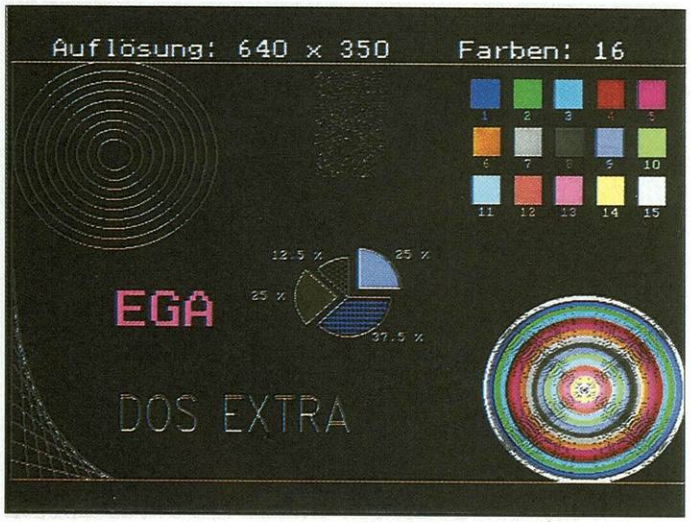


Bild 4. Aufgrund des rasanten technischen Fortschrittes konnte sich die EGA-Karte nur kurze Zeit am Markt behaupten

VGA – heute fast schon Standard

Sie alle besitzen für den Bildschirmspeicher bereits 512 KByte on board, was auch den anfangs teueren Preis verstehen läßt. Mittlerweile sind aber aufgrund der gefallenen Halbleiterpreise bereits VGA-Karten unter 300 Mark erhältlich.

Die Karten der verschiedenen Hersteller unterscheiden sich jedoch stark in ihren Darstellungsmodi und der Geschwindigkeit; dies erklärt die teilweise großen Preisunterschiede.

Waren die ersten Exemplare noch mit einem 8-Bit-Anschluß versehen, so sind heute fast nur noch 16-Bit-Karten erhältlich. Beabsichtigen Sie, eine solche in einen XT-Computer einzubauen, müssen Sie darauf achten, daß die Karte sich auch auf 8-Bit-Betrieb umschalten läßt.

Zu unterscheiden sind bei den VGA-Karten auch die unterschiedlichen Monitoranschlüsse. Es gibt 9poligen TTL- sowie 15poligen Analoganschluß.

Der TTL-Anschluß läßt dabei nur 16 Farben (wie EGA) zu. Der Grund dafür liegt in der Art der Farbsignale, die hierbei nur digtaler Art sind.

Erst mit einem Analogmonitor kann man die 256 Farben der VGA-Karte nutzen. Dies läßt sich dadurch erklären, daß die einzelnen Farbsignale mit unterschiedlichen Spannungswerten einhergehen. Die drei Grundfarben Rot, Grün und Blau lassen sich dabei fast stufenlos einstellen und ergeben insgesamt 262144 Farbtöne. Aus diesen sind wiederum 16 beziehungsweise 256 Farben gleichzeitig darstellbar. Wichtig ist zu wissen, daß nur die beiden VGA-Auflösungen von 320x200 Bildpunkten bei 256 Farben (Bild 5) und 640x480 Bildpunkte mit 16 Farben (Bild 6) bei allen Karten gleich sind.

Sonstige etwa vorhandene Formate können nur genutzt werden, wenn die Software speziell diese Karte kennt. Leider sind es nur wenige Firmen, die auch nachträglich noch zu neuen Softwareprodukten die entsprechenden Bildschirmtreiber liefern. Im Einzelfall sollten Sie sich dabei an Ihren Fachhändler wenden.

Bei einem Neu- oder Zukauf ist also darauf zu achten, daß die Hardware mit Karte und Bildschirm auch durch die vorhandene Software unterstützt wird.

Super-VGA – die High-End-Lösung

Daß man Gutes immer noch verbessern kann, wurde durch die Super-VGA-Karten deutlich. Durch neue Entwicklungen bei den Grafikprozessoren, beispielsweise dem Tseng 4000 oder Trident 8900, sowie einer nochmaligen Verdoppelung des Video-RAM-Speichers auf nunmehr 1 MByte ist jetzt auch die Darstellung von gleichzeitig 256 Farben bei 1024x768 Bildpunkten möglich. Die gleiche Farbenpracht, allerdings nur mit 512 KByte Speicher, kann bei 800x600



Bild 5. Wie Farbe doch das Auge täuschen kann. 320x200 Bildpunkte bei 256 Farben erscheinen deutlicher, als man vermutet.



Bild 6. Mit einer Standardauflösung von 640x480 Bildpunkten wird die VGA-Karte auch höheren Ansprüchen gerecht



Bild 7. Auge, was willst du mehr, könnte man bei dieser Bildqualität der Super-VGA-Karten sagen

Bildpunkten das Auge erheitern (Bild 7).
Leider wird dieser an sich erfreuliche Umstand durch ein kleines Problem getrübt. Da jeder Hersteller sein eigenes Süppchen kocht und kein einheitlicher Standard mehr existiert, benötigt man, um die vorhandenen Fähigkeiten zu nutzen, eigene Bildschirmtreiber. Dies schränkt den Einsatz auf Software ein, für die entsprechende Treiber beim Lieferumfang der Grafikkarte mit dabei sind.

Darum ist bei einer Neuanschaffung äußerste Vorsicht geboten. Oft entpuppt sich ein vermeintlich günstiges Angebot nur als Abverkauf einer alten Kartenversion. Die dabei mitgelieferten Treiber sind dann in aller Regel nicht mehr auf dem neuesten Stand und die VGA-Auflösungen können dann nur bis zu 640x480 Bildpunkten garantiert werden.

Allerdings läßt sich über den praktischen Nutzen dieser »Hochauflösungen« streiten. Der Betrieb von Windows 3.0 auf einem 14-Zoll-Bildschirm bei 1024x768 Bildpunkten wird ausschließlich Ihren Augenoptiker begeistern. Dafür sollte man sich dann zumindest einen 16-Zoll-Monitor anschaffen, was die Sache natürlich erheblich verteuert. Die hohen Auflösungen machen nur dann einen Sinn, wenn nicht, wie es bei Windows der Fall ist, der Informationsinhalt vergrößert, sondern dieser beibehalten und nur die feinere Darstellung durch die größere Zahl von Bildpunkten (Pixel) auf gleicher Fläche genutzt wird. Man erkauft sich jedoch diesen Luxus durch eine geringere Verarbeitungsgeschwindigkeit, was nicht immer sinnvoll sein muß.
Überhaupt ist die Frage des richtigen VGA-Monitors sorgfälltig zu prüfen. Für den vorwiegenden Teil der Anwender dürfte der preisgünstigere Analogmonitor vollkommen ausreichen. Nur wer wirklich alle Features seiner VGA-Karte nutzen will, sollte einen Multifrequenzmonitor mit in die Auswahl einbeziehen. Dabei ist wiederum darauf zu achten, daß Monitor und Grafikkarte zueinanderpassen. Nicht jeder VGA-Bildschirm kann nämlich alle Auflösungsstufen, die eine Karte bietet, auch wirklich synchronisieren.
Letztendlich wird jedoch jeder bei der Anschaffung einer neuen Grafikausrüstung erst seinen Geldbeutel fragen müssen, was dieser zuläßt. Dabei muß nicht immer das Teuerste auch das Beste sein. Für Textverarbeitung allein tut es oftmals auch ein Hercules-Monitor mit Karte.
Sie kennen nun die Unterschiede der einzelnen Grafikstandards und wissen, wo ihre Schwächen und Stärken liegen.
Mit diesem Wissen dürfte es Ihnen nicht schwerfallen, in bezug auf die neue Grafikkarte und den dazu passenden Monitor die richtige Entscheidung zu treffen.

(Helmut Stult/wh)

Grafik-Karte	Bildschirmtyp
Hercules	Monochrom-TTL
CGA	Monochrom-RGB, Farb-RGB
EGA	EGA-Monochrom, EGA-Farbe, Multifrequenz
VGA	Monochrom-Analog, Farb-Analog, Multifreqenz
Super-VGA	Farb-Analog, Multifrequenz (mindestens 1024x768)

Tabelle 2. Der richtige Monitor zur jeweiligen Grafikkarte

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, Druckerspooler 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 (0 81 21) 7 69-0 zur Verfügung.

Schicken Sie Ihre Unterlagen an folgende Adresse:

Redaktion
DOS International
»DOS Software«
Gruberstraße 46c,
8011 Poing

Lexikon

Stichworte zur Computergrafik

Während noch vor gar nicht langer Zeit Heimcomputer wie der Commodore Amiga oder der Atari ST erste Wahl im Bereich Grafik waren, hat sich die Sachlage inzwischen grundlegend geändert. Schlagworte wie VGA, Vektorgrafik und Multiscan-Monitor beherrschen den Markt und verwirren insbesondere den Einsteiger in immer stärkerem Maße. Was liegt da näher, als ein Lexikon mit den wichtigsten Begriffen der PC-Grafik zusammenzustellen, um auch dem Anfänger einen Einblick in das Geschehen zu vermitteln? Im folgenden finden Sie alle relevanten Ausdrücke, die im Zusammenhang mit Grafik auf dem PC auftreten.

8514-Adapter (8514/A): Grafikadapter von IBM, der 1987 für Furore sorgte. Damit wird es auch für PC-Kompatible möglich, Auflösungen von 1024x768 *Bildpunkten* bei Verwendung von 16 aus 262144 Farben zu erreichen. Je nach Ausrüstung des Bildschirmspeichers kann die Anzahl der Farben beträchtlich erhöht werden (bis zu 256 Farben). Zudem handelt es sich beim 8514/A um eine extrem schnelle Karte, die aber momentan nur für die Microchannel-Architektur verfügbar ist.

3D-Darstellung: »Räumliche« Darstellung von Grafiken auf dem Bildschirm. Ein Beispiel für 3D-Grafik sehen Sie in Bild 1.

Animation: Unter Animation versteht man im Zusammenhang mit Grafik, daß einzelne Bilder schnell hintereinander ablaufen, und so den Effekt eines Zeichentrickfilms erzeugen. Einfache Animationen können bereits mit preiswerten Malprogrammen erarbeitet werden.

Auflösung: Maximale Anzahl der *Bildpunkte*, die auf einem Monitor angesprochen werden können. Eine Auflösung von 320x200 Bildpunkten sagt beispielsweise aus, daß 320 Pixel in horizontaler (x-Koordinate) und 200 Pixel in vertikaler (y-Koordinate) Richtung zur Verfügung stehen. Die Auflösung macht also Aussagen über Qualität und Schärfe der Grafik.

Bildpunkt (Pixel): Kleinstes auf einem Monitor ansprechbares und darstellbares Element. Beinahe jedes Malprogramm arbeitet auf der Grundlage von Pixeln.

Bildschirmgröße: Handelsübliche PC-Monitore haben eine Bildschirmdiagonale von 14 Zoll. Im CAD- und DTP-Bereich werden vorwiegend *Ganzseitenmonitore* eingesetzt, die einer Größe von 19 oder 20 Zoll entsprechen.

Bildschirmspeicher: Wie ein PC, so verfügt auch die Grafikkarte über den eigenen Speicher (Bildschirm-, Videospeicher), der zum Aufbau und zur Darstellung einer Grafik genutzt wird. Je größer der Speicher, desto besser ist im allgemeinen auch die Auflösung beziehungsweise die Zahl der Farben, und desto schneller kann die Karte auch Grafiken aufbauen. Eine handelsübliche VGA-Karte besitzt heute 256 KByte Speicher. Besonders leistungsfähige Karten haben bis zu 1 MByte Speicher, was vor allem im professionellen Bereich genutzt wird.

Bildsymbol: Sinnbild in grafischen Applikationen. Eine Schreibmaschine stellt eine Textverarbeitung dar, eine Palette ein Malprogramm und so weiter (Bild 2). Dies erleichtert vor allem Einsteigern die Bedienung der Programme.

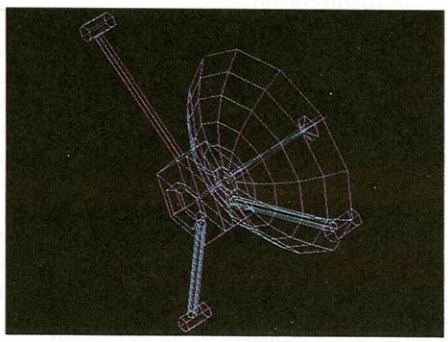


Bild 1. 3D-Grafik vermittelt einen räumlichen Eindruck (hier 3D-Draw)

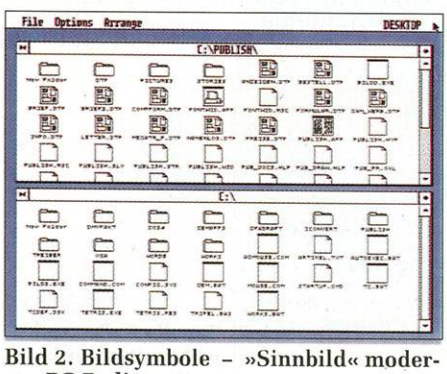


Bild 2. Bildsymbole – »Sinnbild« moderner PC-Bedienung

Businessgrafik: siehe *Geschäftsgrafik*

CAD (Computer Aided Design): Die Abkürzung »CAD« bezeichnet das Konstruieren am Bildschirm (Bild 3). Dies wird vor allem in der Industrie genutzt, beispielsweise zum Zeichnen von Schaltplänen. Auch Architekten, Ingenieure, Designer, Handwerker bedienen sich des PCs, um die Arbeit, die früher mit Tuschestift, Zirkel und Schablone auf dem Zeichenbrett verrichtet wurde, zu bewältigen. Der Vorteil liegt in der enormen Zeitersparnis, der Genauigkeit (Berechnung mit dem Computer) und der Grafikdarstellung, die sich in der Endausgabe auf dem Plotter von der mit der Hand gezeichneten nicht unterscheidet. Außerdem hilft CAD, Papier zu sparen; beispielsweise wenn bei der Erstellung von Konstruktionszeichnungen, die doch mehrere Blatt im Format DIN A0, A1 oder A3 erforderten, eben am Bildschirm gearbeitet wird.

CAD-Monitor: siehe *DTP-Monitor*

CEG (Continous Edge Graphics): Dieses erstmals von Schneider auf der Orgatec '90 vorgestellte Verfahren verhilft VGA-Bildschirmen zu über 700000 Farben. Zudem verhindert es die ungeliebten Zacken und Ecken bei der Darstellung von Kreisen und runden Figuren auf dem Monitor.

CGA-Farbgrafikkarte: Abkürzung für Colour Graphics Adapter. Diese Steckkarte bietet vier Farben (inklusive Schwarz und Weiß) bei einer Auflösung von 320x200 Punkten, wahlweise auch 640x200 Punkte im Monochrommodus

CGA-Emulator: Utility, das Spiele und andere grafische Programme für die CGA-Karte auf einer Hercules-Karte lauffähig macht.

Cursor: Der Cursor ist eine Anzeigemarke, die Ihnen signalisiert, wo Sie sich momentan auf Ihrem Bildschirm befinden. Im Textmodus ist dies meist ein Unterstrich oder ein kleines Rechteck. In Verbindung mit der *Maus* spricht man auch von einem Mauscursor. Im Grafikmodus kann dies ein kleiner Pfeil, ein Fadenkreuz, eine Sanduhr oder ein anderes kleines Sinnbild sein. In Malprogrammen wechselt das Design des Cursors ständig, um dem Anwender zu zeigen, welches Werkzeug momentan aktiv ist.

Digitizer (Digitalisiertablett): siehe *Grafiktablett*

DTP-Monitor: Spezieller Bildschirm für die Verwendung im CAD- und DTP-Bereich mit 19- beziehungsweise 20-Zoll-Bildschirmdiagonale. Alle wichtigen Programme dieses Genres unterstützen diese Bildschirme. Oft werden solche Geräte auch direkt mit einer speziellen Grafikkarte verkauft, die die Eigenschaften dieser Monitore ausnutzen.

EGA (Enhanced Graphics Adapter): Diese Grafikkarte ist der Vorläufer der

VGA-Karte und bietet 640x350 Punkte bei Verwendung von 64 Farben. Damit konnten zum erstenmal qualitativ hochwertige Grafiken auf dem PC erzeugt werden. Heute wird diese Karte immer mehr von der VGA-Karte verdrängt.

EGA-Monitor: Bildschirm, der die hohe Auflösung und die Farbenpracht einer EGA-Karte ausnützt.

Flüssigkristallanzeige: LC-Displays (Liquid Cristall Display) findet man vor allem bei tragbaren Computern (Laptops), da sich dort die aufwendige Technologie eines herkömmlichen Bildschirms nicht realisieren läßt. Der große Vorteil der LCD liegt in der extrem flachen Bauweise, so daß sie schon in einen Laptop-Deckel ausreichend Platz finden. Zudem hält sich der Stromverbrauch sehr in Grenzen. Nachteilig wirkt sich die mitunter sehr schlechte Lesbarkeit aus. Bei bewegten Grafiken wird stets ein »Schatten« hinterhergezogen, der sich sehr störend bemerkbar macht. Momentan existieren fast ausschließlich monochrome LCD-Bildschirme, da farbige Displays zwar technisch machbar, für eine weite Verbreitung aber noch zu teuer sind.

Fraktale: Komplexe Grafiken, denen umfangreiche mathematische Berechnungen zugrundeliegen. Dabei handelt es sich um bestimmte Formeln und Verhältnisse, mit denen sich ausgefallene Gebilde berechnen lassen. Der bekannteste Vertreter dieser Gattung ist das »Apfelmännchen« (Bild 4).

Ganzseitenmonitor: Bildschirme, mit denen jeweils eine ganze DIN-A4-Seite auf einmal betrachtet werden kann. Vorteil: Der Layouter hat eine bessere Übersicht. Ganzseitenmonitore sind wesentlich teurer als herkömmliche 14-Zoll-Bildschirme und werden vor allem im Desktop-Publishing- und CAD-Bereich eingesetzt.

GEM: Grafikformat eines vektororientierten Zeichenprogramms wie GEM-Draw.

Geschäftsgrafik (Businessgrafik): Damit werden alle Grafiken bezeichnet, denen umfangreiches Zahlenmaterial zugrunde liegt und die zur Veranschaulichung komplexer Zusammenhänge dienen (Bild 5). Beispiele sind die Balken- oder Kreisdiagramme zur Darstellung der Prozent- beziehungsweise Sitzverteilung. Geschäftsgrafikmodule sind oft in Tabellenkalkulationen integriert. Programme wie Harvard Graphics, Lotus 1-2-3, Multiplan, Quattro oder MS-Chart erledigen diese Aufgaben.

Grafikeinbindung: Immer mehr Textverarbeitungen werben mit der Aussage, Grafiken aus Mal- und Zeichenprogrammen einbinden zu können. Dies bedeutet, daß Sie Bilder in einem bestimmten *Grafikformat* übernehmen und an einer beliebigen Stelle im Text

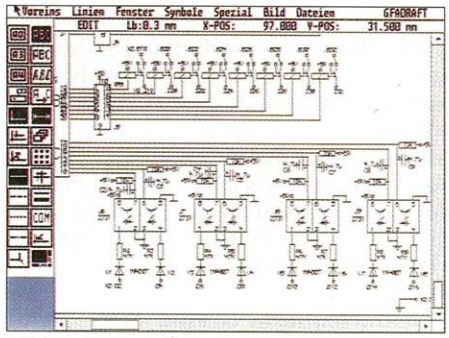


Bild 3. CAD – Konstruieren am PC

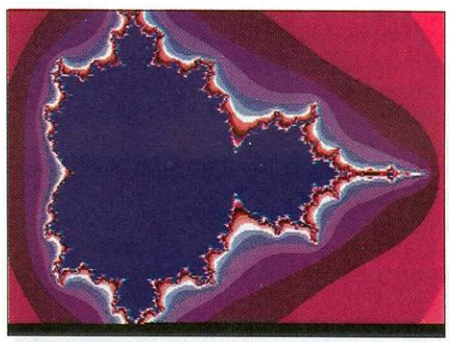


Bild 4. Fraktale Grafik (»Apfelmännchen«)

einfließen lassen. Dies ist eigentlich eine Domäne der DTP-Programme, wird aber in den meisten Textprogrammen angeboten.

Grafikformat: Ein Grafikprogramm legt die Informationen über Größe, Auflösung, Farben etc. in einem bestimmten Format in einer Datei ab. Diese Informationen dienen später zum Aufbau der Grafik auf dem Bildschirm. Die Art und Weise, wie die Daten gespeichert werden, nennt man Grafikformat. Im PC-Bereich existiert eine Vielzahl von verschiedenen Grafikformaten, so zum Beispiel PCX, GEM, IMG.

Grafikmodus: Grafikkarten wie die VGA-Karte bieten mehrere Modi, beispielsweise 256 Farben bei einer Auflösung von 320x200 Bildpunkten. Diese Modi können vom Betriebssystem oder einem Programm genutzt werden, sofern Treiber dafür erhätlich sind.

Grafikstandard: Ähnlich wie im Druckerbereich Epson- oder IBM-Modi Standard sind, so gibt es auch im Grafiksektor Standards, die genau festgelegt sind und Angaben zu Aufbau und Inhalt desselben machen. Hercules, CGA, EGA und VGA sind die bekanntesten Vertreter, daneben existieren aber noch eine Vielzahl weiterer (meist sehr exotischer und leistungsstarker) Standards, die aber noch nicht weit genug verbreitet sind. Die Hersteller von Hard- und Software halten sich weitestgehend an die Vorgaben dieser Standards, um eine höchstmögliche Verbreitung und Kompatibilität des eigenen Produktes zu den bestehenden Standards zu schaffen.

Grafiktablett: Dieses Gerät wird vor allem von Designern und Grafikern eingesetzt. Dabei handelt es sich um ein Plastiktablett, auf dem man mit einem speziellen Stift zeichnen kann. Dieses Eingabeverfahren hat den Vorteil, daß es ziemlich genaues Zeichnen und Konstruieren zuläßt. Auch beim *CAD* werden Grafiktabletts verwendet.

Grafiktreiber: Softwarepakete unterstützen meist alle gängigen Grafikkarten und deren Grafikmodi. Spezielle Eigenschaften einer Karte (zum Beispiel eine hohe Auflösung) werden durch einen Treiber genutzt, der entweder vom Software- oder vom Grafikkartenhersteller mitgeliefert wird.

Grafische Benutzeroberfläche: Windows 3.0, Ensemble oder GEM/3 nutzen die Fähigkeiten der Grafikhardware aus, um zum Beispiel die Bedienung vollständig über *Bildsymbole* abzudecken. So dienen sie vor allem als Grundlage der Benutzeroberfläche.

Hardcopy: Unter einer Hardcopy versteht man die Ausgabe des Bildschirminhalts auf dem angeschlossenen Drucker. Bei Texten stellt dies kein Problem dar, bei Grafiken bedarf es jedoch spezieller Hilfsmittel wie dem MS-DOS-Befehl »graphics«, um die Grauwerte auf den Drucker übertragen zu können. Kommerzielle Produkte wie Pizazz Plus bieten noch mehr Fähigkeiten, um einen optimalen Ausdruck zu erhalten.

Hercules-Karte (HGC): Grafikkarte mit einer maximalen Auflösung von 720x348 in monochromer Darstellung. Die Hercules-Karte bietet aber im Gegensatz zur reinen Monochromkarte noch einen speziellen Modus, der bedingt Grafiken zuläßt. Die meisten Grafikprogramme und Spiele bieten keine Unterstützung für Hercules, hier empfiehlt sich der Einsatz eines CGA-Emulators. Wenn nur Texte und Daten bearbeitet werden, ist die Hercules-Karte eine ernstzunehmende, preiswerte Alternative zu anderen Grafikkarten.

Illustrationsprogramm (Designprogramm): Diese Art von Software wird vor allem von Grafikern verwendet, um professionelle Druckvorlagen von Illustrationen zu erzeugen. Sie verfügen über eine Vielzahl von unterschiedlichen Werkzeugen, um Grafiken und Schriften in jeder nur erdenklichen Weise zu verändern (Zerrungen, Drehungen, Spiegelungen, etc.) (Bild 6). Beispiele für solche Programme sind GEM Artline und Corel Draw.

Interlace: Dies bedeutet, daß Grafiken in zwei Halbbilder zerlegt und nacheinander anzeigt werden (die eine Hälfte umfaßt die geraden, die andere Hälfte die ungeraden Bildzeilen). Die Folge ist, daß die Vertikalfrequenz quasi halbiert wird, da sich das Ge-

samtbild aus zwei halben Bildern zusammensetzt. Dies erweist sich in vielen Fällen als nachteilig, da sich doch ein mehr oder weniger starkes Flimmern bemerkbar macht. Auch der 8514/A von IBM arbeitet auf der Grundlage des Interlace-Verfahrens. Dort begegnet man dem Flimmern aber durch eine relativ hohe Bildfrequenz von 86 Hz, bei der sich derartige Erscheinungen kaum noch bemerkbar machen. Der ganz klare Vorteil des Interlace-Verfahrens besteht im vereinfachten Aufbau der Technik.

IMG: Grafikformat, das vom Malprogramm GEM-Paint (Bild 7) verwendet wird und auch von vielen anderen Grafikprogrammen unterstützt wird.

Konvertierungsprogramm: Mit einem Konvertierungsprogramm wandeln Sie ein Format in ein beliebiges anderes Format um (beispielsweise das Sharewareprogramm Iconvert).

LC-Bildschirm: siehe *Flüssigkristall-Bildschirm*

Malprogramm: Ein Malprogramm arbeitet im allgemeinen bildpunktorientiert und gestattet deshalb pixelgenaues Malen und Editieren. Software dieser Art stellt mehrere Werkzeuge zur Verfügung, so zum Beispiel einen Radiergummi, eine Sprühpistole und verschiedene Arten von Pinseln, Mustern und Farben. Hinzu kommen die beinahe obligatorischen Bestandteile wie Zeichnen von Linien, Kreisen, Ellipsen und Rechtecken.

Maus: Nahezu jedes auf dem Markt befindliche Grafikprogramm benutzt eine Computermaus zur Steuerung. Dabei handelt es sich um ein etwa mausgroßes Plastikgehäuse mit zwei oder drei Tasten, das im Inneren die Elektronik und im Boden eine Rollkugel verbirgt. Wird die Maus bewegt, tastet eine Mechanik die Rollkugel ab und liefert die Informationen an den Computer. Dieser berechnet dann die aktuelle Position des Mauscursors auf dem Bildschirm. Mit der Maus läßt sich sehr schnell und effektiv arbeiten.

Monochromadapter (MDA): Grafikkarte für PC, die eine Auflösung von 720x350 Bildpunkten bietet, jedoch nicht grafikfähig ist; heute meist nicht mehr, und wenn, dann nur noch zur reinen Datenverwaltung eingesetzt, da sie lediglich im Schwarzweißmodus arbeitet.

Monochrombildschirm: Monitor, der je nach Grafikkarte entweder nur Schwarzweißdarstellungen liefert, oder zum Beispiel in Verbindung mit einer VGA-Karte verschiedene Grauabstufungen erlaubt.

MSP: Bildformat des Malprogramms Windows-Paint.

Multiscan-Monitor (Multisync): Monitor, der alle gängigen Grafikkarten und deren Signale ansprechen und verarbeiten kann. Er stellt sich automatisch auf die momentan verwendete Grafikkarte ein und unterstützt alle Modi, die diese verarbeitet. Ein Multiscan-Monitor kann also sowohl für Hercules als auch für VGA benutzt werden.

Muster: Malprogramme bieten Fähigkeiten, Bereiche wie Kreise und Rechtecke mit Mustern zu füllen. Meist können Sie diese Muster nach eigenen Vorstellungen verändern und neue kreieren.

Objektorientierte Grafik: Statt Grafiken pixelweise zu editieren, werden beim objektorientierten Zeichenprogramm Teile einer Zeichnung als Ganzes behandelt. Bei einem Malprogramm können Sie beispielsweise einen gezeichneten Kreis als Figur an sich komplett löschen. Wenn jedes Objekt als eigenständige Figur betrachtet wird, können Sie ohne weiteres Rechtecke aus einer Zeichnung herausnehmen und an einer anderen Stelle positionieren, ohne daß der Hintergrund »mittransportiert« wird. Nahezu alle Illustrationsprogramme arbeiten objektorientiert.

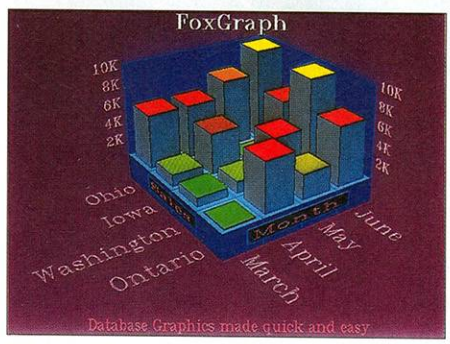


Bild 5. Businessgrafik: »Ein Bild sagt mehr als 1000 Zahlen«



Bild 6. Mit Illustrationsprogrammen generieren Sie Grafiken wie ein Profi

Pattern: siehe *Muster.*

PCC: Grafikformat aus dem Grafikprogramm Print Master (beziehungsweise Lightning Press). Durch die hohe Verbreitung kann dieses Grafikfformat von vielen weiteren Programmen übernommen werden. Außerdem werden mit dieser Endung Dateien mit Ausschnitten aus PCX-Grafiken gekennzeichnet.

PCX: Bildformat des weitverbreiteten Malprogramms PC-Paintbrush. Dieses Format wird praktisch von jedem DTP-Programm gelesen, und auch die meisten Grafikprogramme können mit diesen Grafiken umgehen.

Pixel: siehe *Bildpunkt*

Raytracing: Verfahren zur Berechnung und Darstellung sehr realistischer 3D-Computergrafiken. Dabei werden Lichtquellen, Lichtstrahlen, Schatten, der angestrahlte Körper und deren Eigenschaften (zum Beispiel Form oder Material) berücksichtigt.

Raster: Gewährleistet in CAD- und Zeichenprogrammen eine bessere Orientierung. Dabei wird die eigentliche Zeichnung mit einem Gitternetz unterlegt; so ist beispielsweise gesichert, daß zwei Punkte auf einer Linie liegen.

Scanner: Ein Scanner ist ein Abtastgerät zum Einlesen von Grafiken, Fotografien und Zeichnungen. Dabei tastet eine Optik die Vorlage Punkt für Punkt ab und übermittelt diese an das Scan-Programm. Auf dem Markt existieren sogenannte Flachbett-Scanner zum Bearbeiten ganzer DIN-A4-Seiten, die vor allem im professionellen Bereich eingesetzt werden, sowie Hand-Scanner, die über eine Abtastbreite von etwa 10 cm verfügen. Dabei werden Auflösungen bis zu 400 dpi erreicht. Die besten Ergebnisse erzielen Sie mit Schwarzweißzeichnungen. Farbige Vorlagen werden in Grauabstufungen umgesetzt. Auch zur Bearbeitung von Formularen und zur Texterkennung sind Scanner – die richtige Soft- und Hardware vorausgesetzt – bestens geeignet.

Snapshot-Programm: Hilfsprogramme, um Bilder aus den verschiedensten Programmen festhalten zu können. Sobald die Grafik dargestellt ist, wird eine Tastenkombination betätigt, die das momentane Bild in einer Datei ablegt. Es ist dann zur weiteren Verarbeitung verfügbar.

Symbol: Vor allem CAD-Programme bieten die Option, eine kleine Grafik (zum Beispiel eine Diode oder einen Widerstand), die immer wieder benötigt wird, als Symbol zu speichern, und dann später an beliebiger Stelle wieder einzusetzen.

TIGA (Texas Instruments Graphics Architecture): Ein »Quasi-Grafikstandard«, mit dem man sogenannte »intelligente« Grafikkarten bezeichnet, die den Hauptprozessor von Berechnungen entlasten und so für einen wesentlich schnelleren Bildschirmaufbau sorgen. Dazu gehören mehrere Speicher, welche die umfangreichen Informationen über eine Grafik aufnehmen und weiterverarbeiten. Mit TIGA sind sehr hohe Auflösungen möglich (je nach Karte bis zu 1280x1024 Punkte bei 256 Farben). Ein großer Vorteil ist zudem, daß nur noch ein einziger Grafiktreiber für eine Software verwendet wird. Das Chaos mit

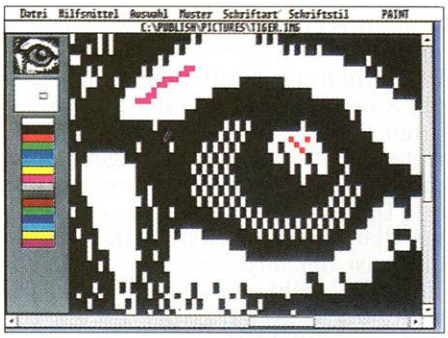


Bild 7. GEM-Paint – ein Beispiel für bildpunktorientierte Malprogramme

den vielen verschiedenen Grafiktreibern würde somit entfallen. Gegen TIGA spricht momentan noch der verhältnismäßig hohe Preis im Vergleich zu EGA- oder VGA-Karten.

Vektororientierte Grafik: Dieser liegt ein ähnliches Verfahren wie der *objektorientierten Grafik* zugrunde. In beiden Fällen werden nicht die einzelnen Bildpunkte, sondern lediglich Informationen über den Aufbau eines Körpers gespeichert, zum Beispiel die Anfangs- und Schlußkoordinaten eines Linienzuges. Deswegen kann man vektororientierte Grafiken auch sehr schnell verändern. Außerdem zeichnen sich solche Grafiken durch eine sehr gute Ausgabequalität auf den Drucker aus.

VGA (Video Graphics Array): Standardmäßig verarbeitet diese Steckkarte eine Auflösung von 640x480 Punkten bei 16 Farben sowie beachtliche 256 Farben aus einer Palette von 262144 Farben bei 320x200 Punkten. Moderne VGA-Karten bieten Spezialmodi, die noch höhere Auflösungen unterstützen (Bild 8). Die VGA-Karte wurde von IBM im Rahmen der Präsentation der PS/2-Modelle der Öffentlichkeit vorgestellt und avancierte zum neuen Grafikstandard.

VGA-Monitor: Analoger Bildschirm mit hoher Auflösung, der mit einer VGA-Karte zusammenarbeitet. VGA-Monitore werden als Farb- oder Monochrommonitore angeboten. Der monochrome Bildschirm stellt dabei die Farben als Graustufen dar.

Video-Controller: Der Video-Controller einer Grafikkarte hat mehrere Funktionen. Zum einen generiert er aus den Informationen des Bildschirmspeichers die Videosignale. Zum anderen sorgt er durch die horizontale und vertikale Synchronisation dafür, daß der Bildschirm korrekt aufgebaut wird.

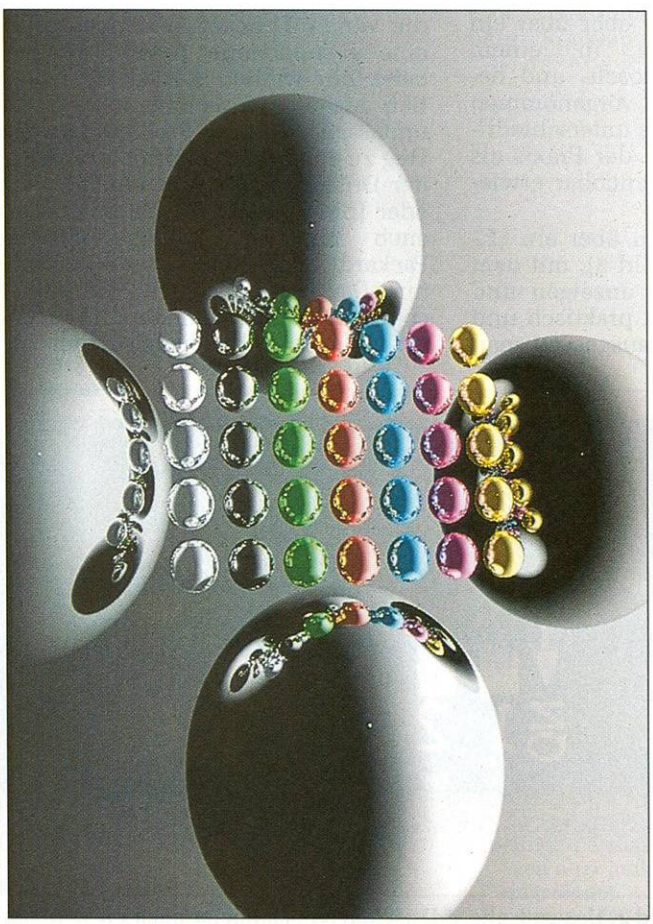


Bild 8. VGA-Grafik bietet hohe Auflösung und viele Farben

Zeichenprogramm: Diese Software arbeitet im Gegensatz zu *Malprogrammen vektororientiert*. Aus diesem Grund lassen sich mit Zeichenprogrammen qualitativ hochwertige Grafiken erzeugen. Sie sind in der Bildbearbeitung jedoch im Vergleich zu den Malprogrammen stark eingeschränkt.

Zoom: Häufig benutzter Begriff in Mal- und Zeichenprogrammen. Damit wird eine Funktion gekennzeichnet, die es erlaubt, Teile einer Grafik in festgelegten oder frei wählbaren Schritten zu vergrößern und zu verkleinern.

(Petra Maueröder/jh)



Träume werden wahr!

Reisen Sie mit LIGHTNING 3D in die faszinierende Welt der Grafik

Lightning 3D ist ein Raytracing-Programm. Sie beschreiben das Bild, Lightning 3D berechnet alle Lichteffekte wie Reflexionen, Brechnungen und Schatten. Mit Lightning 3D brauchen Sie keinen Superrechner mehr, sondern lediglich einen AT, 386er oder 486er, eine Festplatte und eine VGA-Grafikkarte.

Neu in der Version 1.5:

- erheblich kürzere Rechenzeiten
- neue Grundobjekte (beliebige Polygone, Quader)
- farbrichtige Darstellung im Testmodus auch ohne OPTIMIZE
- alle Auflösungen bis 1024x768 Bildpunkte auch ohne entsprechende Grafikkarte berechenbar
- Nutzung von EMS-Speicher
- Unterstützung der Hercules-Graphic-Station-Card in voller Farbauflösung
- Unterstützung des Edsun CEG-Grafikchips für mehr als 700.000 Farben gleichzeitig
- Ausgabe im TARGA-Format mit 16 Mio. Farben

Lightning 3D Version 1.5 nur DM 199,—*

Neu im Programm:

DOS CAD DM 199,—*

Das leistungsfähige Konstruktionsprogramm

* Unabhängig von der Anzahl der bestellten Produkte berechnen wir für das Inland DM 4,— bzw. für das Ausland DM 6,— Porto und Verpackung. Bitte benutzen Sie die Bestellkarte.

DOS SOFTWARE

DMV-Verlag • Postfach 250 • 3440 Eschwege

DMV Daten- und Medienverlag

Grundlagen
Druckerinstallation

Jetzt gibt es Druck

Vor dem Einsatz eines Druckers hat die Technik leider die richtige Installation gesetzt. Dabei haben technisch weniger Versierte oft Probleme, die Anweisungen im Handbuch in die Praxis umzusetzen.

Normalerweise ist es die Aufgabe des Druckers, einen Text genau Zeichen für Zeichen richtig wiederzugeben. Daß in der Praxis jedoch erhebliche Differenzen zwischen »Soll« und »Ist« auftreten, haben Sie sicher auch schon festgestellt. In solchen Fällen ist dann meist guter Rat teuer. Zu groß ist die Zahl der auf dem Markt befindlichen Geräte, als daß man einen generellen Hinweis auf den Fehler geben könnte. Zum Glück sind jedoch gewisse Dinge bei fast allen Typen gleich.

Der Anschluß

Für 99 Prozent der Druckerbesitzer wird nur ein Anschluß an der Parallelschnittstelle (Lpt1) in Frage kommen. Das ist die einfachste Art und bereitet die wenigsten Probleme. Sie wird auch von allen Programmen und vom Betriebssystem direkt unterstützt. Für den Anschluß benötigen Sie ein Druckerkabel, welches normalerweise an einem Ende einen Centronics-Stecker besitzt. Das Gegenstück dazu finden Sie am Drucker (Bild 1). Leider ist dieses Kabel bis auf ganz wenige Ausnahmen bei keinem Drucker im Lieferumfang dabei. Beachten Sie beim Kauf, daß Sie kein Kabel für den Commodore-Amiga erhalten. Es schaut zwar genauso aus, nur fehlen ihm einige Verbindungen. Stecken Sie nie im eingeschalteten Zustand Stecker an oder ab, da dadurch die Schnittstellen zerstört werden.

Die richtige Funktion des Druckers läßt sich am einfachsten durch Drükken der Tasten [Strg-P] testen. Dadurch werden alle Textausgaben auf dem Bildschirm zusätzlich auch noch an die Parallelschnittstelle gegeben. Wenn Sie nun »dir« zur Ausgabe des Laufwerkverzeichnisses eingeben, muß dieses auch auf dem Druckerpapier erscheinen. Sollten dabei Fehler auftreten, muß die Einstellung am Drucker geändert werden.

Die Grundeinstellung

Grundsätzlich gibt es beim Drucker zwei verschiedene Einstellarten.

Dies sind zum einen die Festeinstellungen, welche, je nach Fabrikat, entweder über Dipschalter (Dual-Inline-Package) (Bild 2) oder aber über ein Menü beispielsweise in einem EEPROM (elektrisch lösch- und beschreibbarer Speicher) vorgenommen werden. Hierbei gibt es unterschiedliche Arten, die sich in der Praxis als mehr oder weniger brauchbar erwiesen haben.

Neuere Geräte verfügen über ein LC-Display und Tasten (Bild 3), mit dem sich die Einstellwerte anzeigen und ändern lassen. Dies ist praktisch und geht sehr schnell, vorausgesetzt man kennt die Funktion der einzelnen Tasten.

Wesentlich umständlicher ist dagegen die Einstellung über einen Papierausdruck (Bild 4). Man hat zwar alle Parameter auf einen Blick vor Augen, kann die Einstellung aber nur vor Beginn des Druckens durchführen. Oft muß man dazu auch die Abdeckung öffnen, da sonst der Text nicht zu lesen ist. Nur wer nicht ständig Änderungen vornimmt, wird auf Dauer mit diesem Verfahren glücklich.

Daß all diese Einstellarten ihre Unzulänglichkeiten besitzen, haben selbstverständlich auch die Hersteller erkannt. Was lag also näher, als sich dabei des sowieso vorhandenen Computers zu bedienen. Herausgekommen ist dabei ein Programm, welches übersichtlich die einzelnen Optionen auf dem Bildschirm darstellt (Bild 5). Sie nehmen dann die Auswahl vor und schicken diese an den Drucker. Zwar erlaubt auch diese Art keine Änderungen im Betrieb, besticht aber durch ihren Bedienungskomfort.

Da viele der Einstelloptionen bei fast allen Modellen gleich sind, stellen wir die wichtigsten davon vor.

- Druckeremulation

Da die Anwendersoftware meist einen bestimmten Druckertyp erwartet, können Probleme auftreten, wenn Sie ein anderes Fabrikat besitzen. Zwar verfügen heute bereits fast alle Programme, die einen Drucker benötigen, über eine ausreichende Anzahl an Treibern, nur werden sie selbstverständlich nie alle vorkommenden Geräte umfassen. Um dem Problem aus dem Weg zu gehen, kann normalerweise jeder Drucker zumindest einen Epson- oder IBM-Drucker, bei Laserdruckern auch noch HP-Laserjet (Hewlett Packard) oder Diablo (Xerox) emulieren. Dadurch versteht er die Steuercodes des entsprechenden Typs und verhält sich auch wie ein solcher:

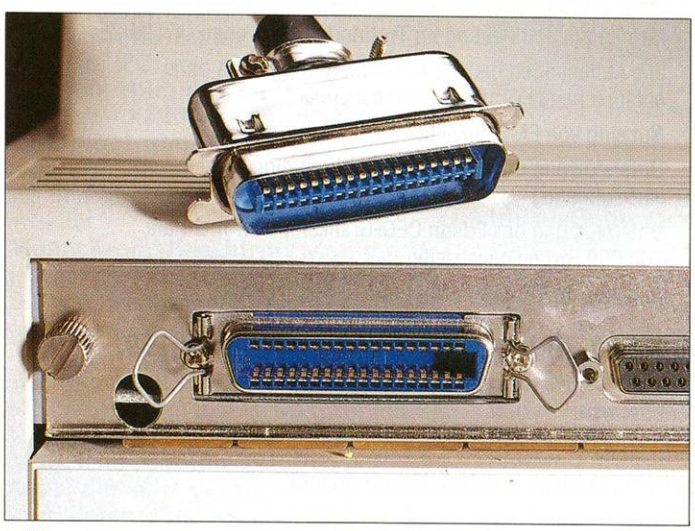


Bild 1. Der sogenannte Centronics-Stecker ist der Standardanschluß für Drucker

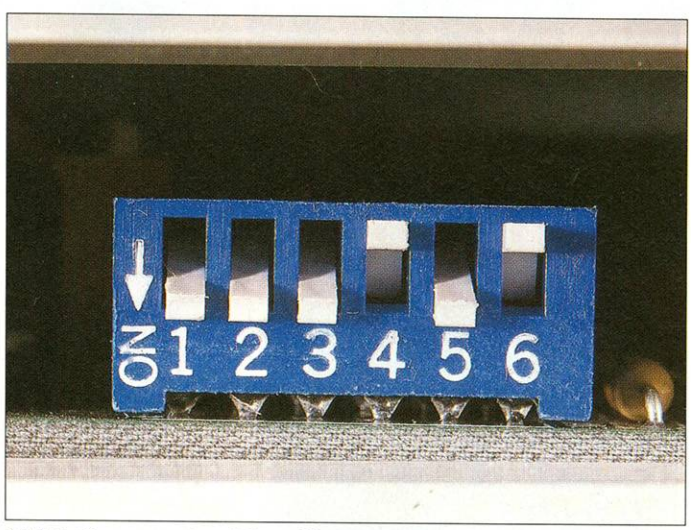


Bild 2. Das gute alte »Mäuseklavier« ist immer noch die einfachste Einstellart, sofern es leicht erreicht werden kann



Bild 3. Mit einem LED-Display lassen sich die Einstellungen abrufen und über Tasten ändern

Deshalb als Tip: Achten Sie beim Neukauf nicht nur allein auf die Schriftarten eines Druckers, sondern auch darauf, daß er zumindest einen der Standardtypen emulieren kann.

● Blattlänge
Die Angabe erfolgt fast immer in Zoll. Endlospapier hat dabei normalerweise 12 Zoll, Einzelblätter der Größe DIN A4 haben 11 2/3 Zoll. Manche Drucker lassen jedoch nur die Wahl zwischen 11 und 12 Zoll zu. In diesem Fall stellen Sie bei Einzelblatt auf 11 Zoll, falls das Handbuch keine andere Angabe dazu macht.

● Zeilenabstand
Der Standardabstand beträgt 1/6 Zoll. Sind andere Vorgaben möglich, können Sie diese im Bedarfsfall ändern. Mit 1/8 Zoll verringert sich der Abstand zum Beispiel, so daß mehr Zeilen pro Seite geschrieben werden können.

● Spaltenbreite
Besitzen Sie einen DIN-A3-Drucker, kommt dieser Einstellung große Bedeutung zu. Da normalerweise nur auf A4-Papier gedruckt wird, muß hier unbedingt auf 80 Zeichen gestellt werden, weil sonst eventuell auf die Walze gedruckt wird. Dies ist beispielsweise beim Testausdruck der Fall.

● Zeilendichte
Die Auswahl hängt hier wieder vom Drucker ab. Standard sind dabei 10 cpi (Zeichen pro Inch). Bei 12 cpi wird die Schrift schmaler, und bei Proportional ändert sich die Breite der einzelnen Zeichen. Ein »l« ist dann beispielsweise schmaler als ein »m«. Dadurch verbessert sich zwar das Schriftbild, jedoch lassen sich keine Tabellen mehr drucken, da die einzelnen Zahlen nicht mehr sauber untereinander stehen.

● Druckqualität
Nachdem heute fast jeder Drucker über NLQ (Near Letter Quality) verfügt, dürfte diese Einstelloption auch vorhanden sein. Damit läßt sich zwischen Normal- (Draft) und Schönschrift (NLQ) umschalten. Sie sollten jedoch den Draft-Druck wählen und nur bei Bedarf per Steuerbefehl auf NLQ umschalten. Dies kommt sowohl dem Druckkopf als auch dem Farbband zugute.

● Druckrichtung
Zwei Einstellungen sind dabei erlaubt: bidirektional (in beiden Richtungen) und unidirektional (nur in einer Richtung). Der bidirektionale Druck ist dabei der schnellere von beiden. Es wird eine Zeile beim Vorlauf, die nächste beim Rücklauf des Druckkopfes gedruckt. Dadurch können sich leichte Verschiebungen der Buchstaben untereinander ergeben. Dies ist gerade beim Ausdruck von Grafiken störend. Viele Drucker lassen deshalb die Wahl zwischen bi- und unidirektional im Text- und Grafikmodus zu.

● Zeilenvorschub nach »Return«
Diese Option ist normalerweise ausgeschaltet, da der Computer automatisch nach jedem Return auch ein Linefeed (Zeilenvorschub) schickt. Sollte auf Ihrem Ausdruck der Zeilenabstand nicht stimmen, überprüfen Sie diese Einstellung.

● »Return« nach Zeilenvorschub
Im Prinzip gilt das gleiche wie zuvor. Die Änderungung der Voreinstellung ist bei PCs normalerweise nicht notwendig.

● Papiersensor
Dieser Schalter erkennt, ob sich noch Papier im Drucker befindet. Soll bei Einzelblättern bis an das untere Ende gedruckt werden, muß der Sensor ausgeschaltet sein. Sonst ist es ratsam, ihn einzuschalten, um zu vermeiden, daß bei Papierende auf die Walze geschrieben wird.

● Automatischer Perforationssprung
Bei der Verwendung von Endlospapier stört es beispielsweise beim Listingausdruck, daß auch auf der Abrißkante gedruckt wird. Durch Einschalten des Perforationssprunges bleibt am Ende und Anfang jeden Blattes ein gewisser Bereich frei. Zur richtigen Funktion muß natürlich auch die Blatt-

```
                    PRINTER MEMORY SETTINGS

CURRENT PRINTER SETTING ( CURRENT SETTING IS [...] )

      PRINT MODE SETTINGS ( 1 thru 7 )
            1. PRINT MODE POWER-UP
               [LQ]   PS    DRAFT
            2. LQ PRINT STYLE DEFAULT
               [COURIER 10]      ITC SOUVENIR 10       PRESTIGE ELITE 12WP
                FAST FOCUS 10    OPTION FONT
            3. PS PRINT STYLE DEFAULT
                BOLD    HELVETTE   [TIMES]   OPTION FONT
            4. 12 CPI DRAFT SETTINGS
               [DRAFT]   HIGH SPEED
            5. LANGUAGE CHAR SET
               [USA]   FRA    GER    ENG    DM1    SWD    ITL    SPA    JPN
                NOR    DM2    NET    TUK    SP2    L.A.
            6. CHAR SET DEFAULT
                ITALIC   [IBM(R)/STANDARD]
            7. SHAPE OF ZERO
               [0]   Ø
      FORMS SETTINGS ( 1 thru 9 )
            1. FORM FEED LENGTH IN INCHES
                3    3.5  4    5.5  6    7    8    8.5  11   11-2/3 [12]  14
            2. SKIP OVER PERFORATION
               [ON]  OFF
            3. LINES PER INCH
               [6]   8
            4. PAPER LOAD/RETURN POSITION
                PLATEN   [TEAR BAR]
            5. CR FUNCTION
               [CR ONLY]    CR+LF
            6. LF FUNCTION
               [CR+LF]      LF ONLY
            7. BUFFER FULL AUTO LINE FEED
               [ON]  OFF
            8. SHEET FEEDER
                INSTALLED   [NOT INSTALLED]
            9. AUTO INSERT
                ON   [OFF]
      INTERFACE SETTINGS ( 1 thru 7 )
            1. MODE
               [PARALLEL]   SERIAL
            2. BUFFER CAPACITY
                8K(NO DL)   [4K]
            3. BAUD RATE SETTING
                150   300   600   1200   2400   4800   [9600]   19200
            4. CHAR LENGTH SETTING
               [8BITS]   7BITS
            5. PARITY CHECK
               [NONE]   ODD    EVEN
            6. DCD,CTS,DSR SIGNALS
               [ENABLE]   DISABLE
            7. COMMUNICATION PROTOCOL SELECT
                ETX/ACK   [X-ON/X-OFF]   X-ON/X-OFF REPETITION

CONTROL PANEL BUTTON FUNCTIONS
 1.PRESS FEED        / YES  = MOVE TO NEXT SELECTION
 2.PRESS PITCH MODE  / NO   = CHOOSE MODE IN MAIN MENU
                            = CHANGE SETTING IN A MODE
 3.PRESS LOAD        / QUIT = EXIT AND SAVE SETTINGS
                            = RETURN TO MAIN MENU
```

Bild 4. Von der Idee her zwar gut, für schnelle Änderungen aber äußerst ungeeignet – das Setup-Menü auf dem Papier

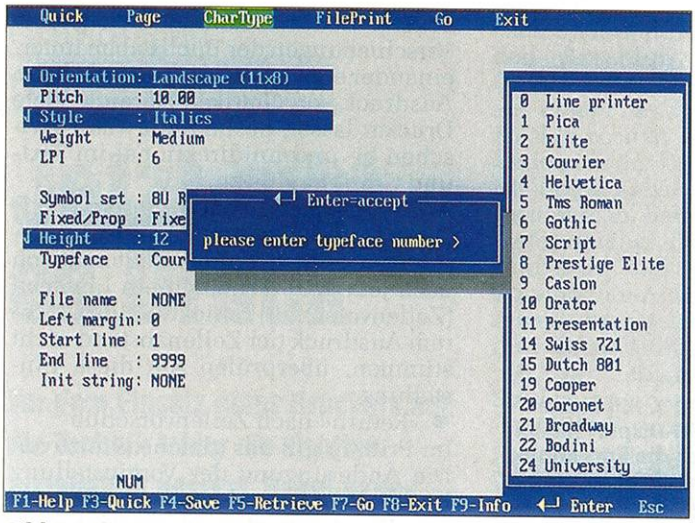


Bild 5. Mit einem Installationsmenü werden die Einstellungen ausgewählt und dann zum Drucker geschickt

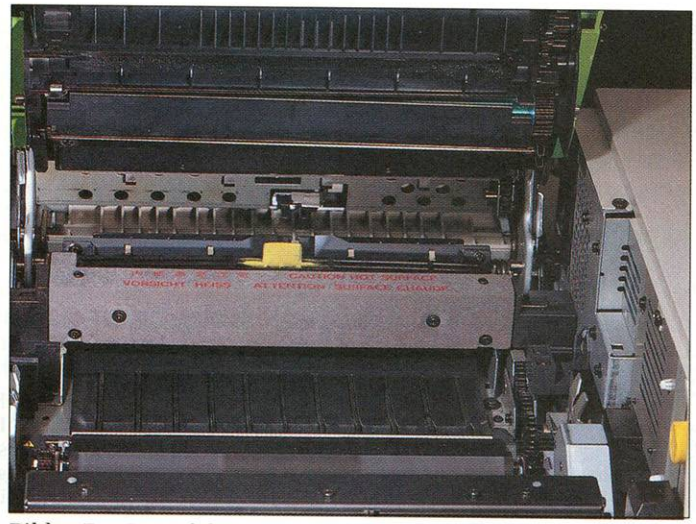


Bild 6. Das Innenleben eines Laserdruckers. Deutlich sind der Tonerbehälter und die Belichtungseinheit zu erkennen.

länge stimmen, da sich der Blattanfang sonst verschiebt.

- Abreißfunktion

Diese Option erleichtert den Papierabriß. Dazu wird das letzte Blatt bis zur Abreißkante geschoben, wo es leicht gelöst werden kann. Bei einem weiteren Ausdruck transportiert es die Walze wieder an die alte Position. Dadurch geht kein Blatt mehr unnötig verloren.

- Null-Schriftzeichen

Normalerweise wird die Null ohne Querstrich geschrieben. Gerade aber beim Ausdruck von Programmlistings empfiehlt es sich, diesen einzuschalten, damit sich die Zahl »0« und der Buchstabe »O« eindeutig voneinander unterscheiden lassen.

- RAM-Wahl/benutzerdefinierte Zeichen

Verfügt Ihr Drucker über einen eigenen RAM-Speicher, läßt sich dieser damit in seiner Funktion umschalten. Dient er als Puffer, können je nach Größe, mehrere Textseiten auf einmal an den Drucker geschickt werden. Als Download-Speicher nimmt er eigene Zeichensätze auf, die von Ihnen mit spezieller Software selbst generierbar sind.

- Automatischer Papiereinzug

Besitzen Sie eine solchen, müssen Sie das auch Ihrem Drucker mitteilen. Dies ist notwendig, da dann der Papiereinzug anders angesteuert wird. Auch dabei ist die richtige Einstellung des Papierformates äußerst wichtig. Nur wenn dieses mit der Vorwahl übereinstimmt, kann der Einzug richtig funktionieren.

- Zeichensätze

Diese Einstellung dürfte die größten Schwierigkeiten bereiten. Um sie richtig vornehmen zu können, müssen Sie die Unterschiede der verschiedenen Zeichensätze kennen. So unterscheidet sich »IBM #1« von »IBM #2« in der Belegung der Zeichen von »ASCII 0 bis 31« und »ASCII 128 bis 159«. Die genaue Belegung entnehmen Sie bitte der Tabelle in Ihrem Druckerhandbuch.

Beim »EPSON-Zeichensatz« sind die Zeichen von »ASCII 128 bis 255« unterschiedlich belegt. Normalerweise befinden sich in diesem Bereich die Kursiv-Buchstaben (Italic). Bei neueren Druckern lassen sich aber auch die IBM-Sonderzeichen einblenden.

Betreiben Sie Ihren Drucker im Epson-Modus, müssen Sie auch noch den richtigen nationalen Zeichensatz wählen. Hierbei kommt es darauf an, inwieweit die benutzte Software die Umwandlung der deutschen Umlaute selbständig vornimmt oder ob Sie auf deutsche Schriftzeichen umschalten müssen.

Nicht alle Einstellungen werden aber allein über das Grund-Setup durchgeführt. Diese lassen sich über Steuerbefehle, die der Computer zum Drucker schickt, vornehmen. Um die Anweisungen vom normalen Text zu unterscheiden, wird ihnen das ASCII-Zeichen 27 (Esc) vorangestellt. Daher resultiert auch der Name »Escape-Sequenzen«. Sie sind bei jedem Modell unterschiedlich, da damit die speziellen Eigenschaften und Fähigheiten gesteuert werden. Die genaue Aufstellung der Befehle für Ihren Drucker finden Sie wiederum im Handbuch.

Zum Schluß noch einige grundsätzliche Hinweise zu den einzelnen Druckertypen.

- Farbdrucker

Verwenden Sie im Normalbetrieb immer nur ein einfaches schwarzes Farbband. Dieses ist wesentlich billiger als das Spezialband. Außerdem wird Schwarz sowieso am häufigsten gebraucht, so daß diese Farbspur meist völlig aufgebraucht ist, während die anderen noch wie neu sind.

- Nadeldrucker

Hierbei sollten Sie schon beim Kauf darauf achten, daß beim Druckerfarbband nur wenige Zentimeter im Freien liegen. Ist dies auf der kompletten Breite der Fall, trocknet die Farbe schneller ein, da sie auf einer viel größeren Fläche mit der Luft in Berührung kommt.

- Tintenstrahldrucker

Für diesen Typ benötigen Sie, um optimale Druckergebnisse zu erzielen, ein besonderes Papier. Hier hilft meist nur Probieren, bis Sie das richtige gefunden haben.

- Laserdrucker

Auch hier kann das Papier eine wichtige Rolle spielen. Achten Sie auf die Empfehlungen des Druckerherstellers bezüglich Gewicht und Oberflächengüte.

Ein Tip aus eigener schlechter Erfahrung: Besorgen Sie sich immer frühzeitig einen Ersatztoner. Selbst in Computer-Hochburgen wie München ist dieser nicht unbedingt auf Anhieb zu beschaffen. Beachten Sie beim Einbau des Tonerbehälters oder Belichtungseinheit (Bild 6) genau die Hinweise im Druckerhandbuch. Wenden Sie dabei keinerlei Gewalt an. Berühren Sie auf keinen Fall die Magnetrolle mit den Fingern, da dies die Oberflächenbeschichtung zerstört.

Als Trost, wenn nicht alles auf Anhieb so funktioniert, wie Sie es sich vorgestellt haben, sei gesagt, daß selbst »Profis« mit Schwierigkeiten zu kämpfen haben, einen Drucker an die verwendeten Softwareprodukte anzupassen. Auch Sie werden nicht daran vorbeikommen, ab und zu bei einem besonderen Programm Ihre Einstellungen zu ändern. Dann sollten Sie jedoch die einzelnen Fähigkeiten kennen und in der Lage sein, sie auch zu nutzen. Hierzu sollen diese Grundlagen beitragen.

(Helmut Stult/wh)

Ein starkes Heft

Im ABO



Es lohnt sich!

Bitte benutzen Sie die Bestellkarte

DMV-Verlag • Postfach 250 • 3440 Eschwege

DMV Daten- und Medienverlag

Grundlagen Netzwerk

Netzarchitektur

Die Begriffe »LAN« und »WAN« etablieren sich in der Computerwelt. Zeitschriften schreiben über »Bridges« und »Accounts«. – Wer den Begriff »PC« wörtlich nimmt, kann mit diesen »Fremdwörtern« wenig anfangen.

Mit dem Aufkommen des PCs, sprich des »Persönlichen Computers«, wurden vielen Anwendern ungeahnte Perspektiven rationellen und kreativen Arbeitens eröffnet. Auch das menschliche Bedürfnis nach Teamarbeit läßt sich im DV-Bereich weiterverfolgen, wenn mehreren Anwendern die Gelegenheit geboten wird, gemeinsam Programme und Peripheriegeräte zu nützen und auf einen einheitlichen und aktuellen Datenbestand zuzugreifen. Mit diesem Konzept war die Entwicklung von leistungsfähigen LANs (Local Area Networks) verbunden; sie erlauben die Verteilung komplexer Aufgaben in einem Netz, um rechenintensive Aufgaben an einer beliebigen Workstation abzuarbeiten.

Netzwerke kann man nach heterogenem oder homogenem Aufbau unterscheiden. »Heterogen« bezeichnet die Verknüpfung von Geräten verschiedenster Hersteller und Rechnerklassen (zum Beispiel, Verbund von PCs, Minicomputer und Mainframe). »Homogen« nennt man dagegen ein Netz, in dem nur Rechner eines Herstellers beziehungsweise eines Rechnertyps ihren Platz finden können.

Die Soft- und Hardwarehersteller erkannten diesen Trend zum Netzwerk frühzeitig, ein breitgefächertes Marktspektrum ist das Resultat (Bild 1). Zu differenzieren ist hier in erster Linie zwischen den unterschiedlichen Topologien der verschiedenen LAN-Produkte.

Netzwerk-Topologie

Ein Weg, mehrere Rechner miteinander zu vernetzen, ist die Verbindung jedes einzelnen Computers mit jedem anderen. Die Stärke dieser Konstellation liegt darin, daß jedes Gerät direkt ohne Vermittlungsaufwand erreicht werden kann. Der Nachteil liegt im hohen Bedarf an Verbindungsleitungen. Aus wirtschaftlichen Gründen und wegen schlechter Erweiterbarkeit wird diese Struktur nur selten eingesetzt.

Das Teilstreckennetz repräsentiert eine besondere Netzarchitektur. Mehrere Brückenrechner sind direkt miteinander verbunden und agieren als Vermittlungsstationen, die die Datenpakete an den nächsten »Vermittlungs«-Computer bis zum Zielpunkt weitergeben. Wegfindung und Zwischenspeicherung erfordern beträchtlichen technischen Aufwand, deshalb diese Struktur vor allem in WANs (Wide Area Network) und in GANs (Global Area Network) Anwendung findet.

Bei der Stern-Topologie sind alle Knotenpunkte mit einem Brückenrechner verbunden, der Verbindung »schaltet«. Bei dieser Variante können mehrere Verbindungen gleichzeitig bestehen, und es können mehrere Empfängerknoten mit einem Senderknoten verbunden sein. Schwierigkeiten treten bei einem Ausfall der Vermittlungsstelle auf, wenn das gesamte Netz lahmliegt. Außerdem wird der Datendurchsatz von der Leistung und Auslastung des Vermittlungsrechners beeinflußt. Hängen sämtliche Knoten an einer Leitung, nennt man diesen Aufbau »Diffusionsnetz«. Ein solches Netz benötigt keinen Vermittlungsaufwand, da immer nur ein Rechner senden kann und diese Sendung alle Knoten gleichzeitig empfangen können. Ein typisches Diffusionsnetz stellt die Bus-Topologie dar, wie zum Beispiel das weitverbreitete Ethernet. Beim Bus ist ein minimaler Leistungsbedarf erforderlich. Das Netz ist auch beim Ausfall einzelner Knoten funktionstüchtig und einfach zu erweitern. Es sind jedoch keinerlei Verzweigungen erlaubt. Auch die Baum-Architektur ist ein Diffusionsnetz mit allen Eigenschaften des Busses, auch mit Verzweigungen. Die Baumstruktur erweist sich durch diese Fähigkeit als günstig für Installationen gesamter Gebäude. Die Wurzel des Baumes, meist eine nachrichtentechnische Einrichtung, empfängt die Daten von einem Knoten und gibt diese an alle anderen Knoten weiter.

Die Bus-Topologie erlaubt eine einfache Erweiterung des Netzes

Eine weit verbreitete Netzstruktur: die Ring-Topologie. Der Datenfluß läuft in einer Richtung. Jeder Knoten interpretiert einen Teil der Daten und gibt sie an seinen Nachbarn weiter. Mit zunehmender Anzahl von Knoten verlangsamt sich die Geschwindigkeit des Rings. Ein Nachteil liegt in geringer Fehlertoleranz, fällt ein Knoten aus, so steht der gesamte Ring. Es gibt jedoch technische Einrichtungen, um passive oder defekte Knoten automatisch aus dem Ring zu nehmen und die dadurch entstehende Lücke im Ring zu schließen.

Verschiedene Netzarchitekturen oder einzelne voneinander unabhängige Netze lassen sich über spezielle Brücken miteinander verbinden. »Gateway« ist hier das Schlagwort, sie öffnen eine Tür zu anderen Netzwerken. Der Anwender soll allerdings von solchen Brücken nichts merken.

Homogene Netze, wie etwa ein PC-LAN, sind durch gleichartige Schnittstellen, Protokolle und Betriebssysteme einfacher zu realisieren als heterogene Netze. Um beispielsweise MS-DOS-Rechner mit Minicomputern

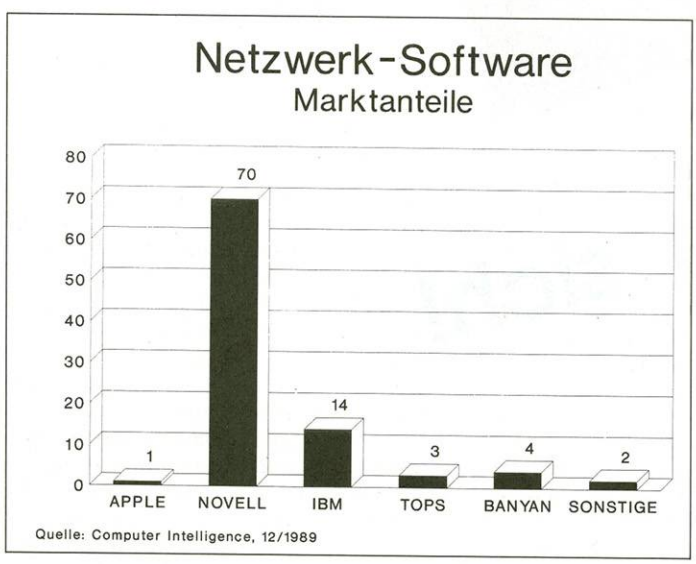


Bild 1. Marktanteile der Netzwerksoftware-Hersteller

oder Großrechnern in Verbund zu setzen (heterogenes Netz), bedarf es eines weit höheren Aufwandes als bei der Vernetzung der PCs untereinander.
Der Schlüssel für diese Vereinfachung ist das Aufteilen der Kommunikation der Rechner eines Netzwerkes in verschiedene Schichten. Jede Schicht stellt dabei eine klar definierte Schnittstelle zur nächsthöheren Schicht zur Verfügung und sollte direkt nur auf die darunterliegende zugreifen können. Neben dem vertikalen Informationsfluß von Schicht zu Schicht existiert ein horizontaler zwischen den einander entsprechenden Ebenen der Kommunikationspartner. Es handelt sich um die virtuelle Kommunikation der Schichtprotokolle. Die ISO (International Standardisation Organisation) hat als internationalen Maßstab zur Gestaltung und Beurteilung von Rechnernetzen ein Referenzmodell entwickelt. Dieses Modell besteht aus sieben Schichten und veranschaulicht die funktionalen Grenzen der zum Netzwerkbetrieb nötigen Module. Bereits vorhandene Standards der Datenübertragung wurden miteingebunden. Die unterste Schicht (»Physical Layer«) erläutert Hardwarekomponenten, die zur Datenübertragung nötig sind. Der darüber liegende »Link Layer« koordiniert die Verbindung zwischen zwei Partnern, »Networklayer« regelt den optimalen Weg der Daten im Netzwerk. Der nächsthöhere »Transport Layer« stellt eine direkte Verbindung zwischen Sender und Empfänger dar. Die Schnittstelle zur Umgebung der verschiedenen Rechner realisiert Ebene fünf: »Session Layer« ist eng mit der Transportschicht und dem Betriebssystem des Brückenrechners verbunden. Die Nummer sechs ist die Präsentationsebene, die im wesentlichen Anpassungen von Datenformaten vornimmt. Die Anwendungsebene, die über allen anderen liegt, unterstützt das Betriebssystem und beinhaltet Komponenten wie »Accounting«, Zugangskontrolle und administrative Funktionen (Bild 2). Dieses 7-Schichten-Modell der ISO liefert die Voraussetzungen dafür, daß jedes Endsystem mit anderen (auf diesem Standard aufgebauten Endsystemen) kommunizieren kann. Derartige Systeme nennt man »offene Systeme«. Aus diesem Grund wird das Referenzmodell auch als ISO-OSI-Modell bezeichnet. Ethernet, in der Norm 802.3 des IEEE (Institute of Electrical and Electronics Engeneers) definiert, ist ein Übertragungsmedium auf der Basis eines Koaxialkabels, auf dem Daten im Busverfahren transportiert werden. Der maximale Datendurchsatz beträgt hier 10 MBit pro Sekunde. Durch die Busarchitektur kann jeweils nur ein Sender Daten an das Medium senden. Es darf also immer nur ein Sender zum

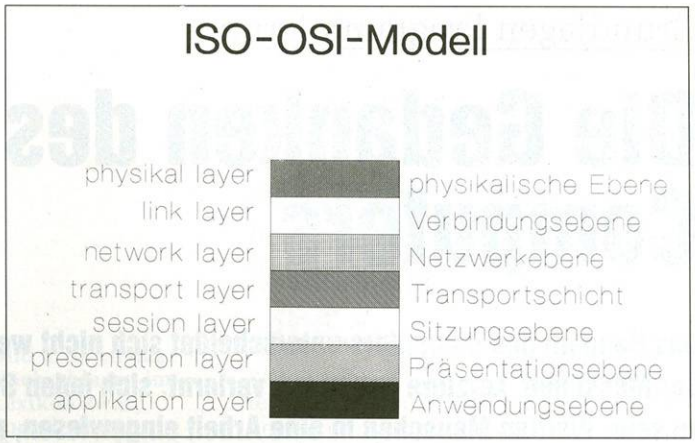


Bild 2. Diese Grafik verdeutlicht den Aufbau des ISO-OSI-Referenzmodells

Zeitpunkt der Datenübertragung aktiviert sein. Dies wird durch das CSMA/CD-Verfahren (»Carrier Sense Multiple Access Collision Detection«) gewährleistet. Will ein Sender Daten verschicken, »hört« er das Übertragungsmedium ab und schickt seine Informationen ins Netz, sowie er keinen Datenverkehr erkennt. Sollte zur gleichen Zeit ein anderer Sender zu arbeiten beginnen, stoßen die Daten aufeinander. Diese Kollision wird registriert (»collision detection«), und Gegenmaßnahmen werden eingeleitet. Je mehr Datenverkehr auf einem Ethernetstrang stattfindet, desto höher ist die Wahrscheinlichkeit solcher Datenkollisionen. Je mehr Kollisionen, desto geringer ist der Durchsatz und in der Folge die Datenübertragungsgeschwindigkeit.

Mit dem ISO-OSI-Modell verschiedene Endsysteme in einem Netz betreiben

Das Token-Ring-System arbeitet nach einem anderem Prinzip. Es muß, wie der Name besagt, ein physikalischer Ring gebildet werden. Diese Architektur wurde mit dem Standard IEEE-802.5 definiert. Verschiedene Token-Ring-Systeme, wie der IBM-Ring mit 4 MBaud, Apollo Domain (12 MBaud) oder auch der IBM-Mikrokanal-Token-Ring (16 MBaud) existieren.

Beim Ring sind die Stationen reihum miteinander verbunden. Ein Bitmuster, das sogenannte »Token«, kreist durch den Ring. Will eine Station Daten senden, so wandelt sie das »Frei-Token« in ein »Belegt-Token« um und hängt die Daten daran. Diese werden von Rechner zu Rechner weitergereicht, bis sie zum Empfänger gelangen. Dieser quittiert den Empfang und schickt die Daten zum Sender zurück, um sicherzustellen, daß alle Informationen korrekt angekommen sind. Danach durchläuft wieder das Frei-Token den Ring, bis das Spiel von neuem beginnt.

Durch neue Medien wie etwa Glasfaserkabel steht einer optimalen Kommunikation nichts mehr im Wege.

Beim Token-Ring läuft alles im Kreis

Die Bereitstellung des Mediums und der daran anzuschließenden Geräte wäre ohne entsprechende Software nutzlos. Ein vielfältiges Angebot der verschiedensten Hersteller eröffnet der Netware-Markt.

Die Firma Novell stellt mit ihrer Netware in den verschiedensten Versionen den Spitzenreiter. Diesen Rang konnte diese Firma vor allem durch das Konzept einer weitgehenden Hardwareunabhängigkeit, eines hohen Bedienerkomforts und der Integrationsmöglichkeit der verschiedensten Betriebssysteme sowie der darauf aufbauenden Anwendungen erreichen. Und diesen Trend könnte sie mit der neuen Netware 386 fortsetzen.

Neue Rechnertypen erfordern angepaßte Software

IBM schickt sich an, seine Marktanteile im Bereich der Netzwerksoftware zu vergrößern. Zusammen mit der Firma Microsoft wird versucht, durch die konsequente Nutzung von OS/2 und dem neuen LAN-Manager 2.0 eine echte Alternative zur Novell-Netware zu etablieren. Zu erwähnen sind jene Softwareanbieter, die unter »andere« in unserer Grafik vereinigt wurden. Es handelt sich in erster Linie um Low-Cost-Anbieter, die ihren Kundenkreis hauptsächlich in Klein- und Familienbetrieben und bei Privatanwendern finden. Das Bemerkenswerte daran ist, daß diese oft unbekannten Netzwerke beachtliche Leistungen zur Verfügung stellen können, und das zu einem Bruchteil des Preises ihrer klangvollen Konkurrenten.

(Hagen Wolfstetter/wh)

Grundlagen Programmierung

Die Gedanken des Computers

Das Handeln des Computers unterscheidet sich nicht wesentlich von dem der Menschen. Letztere haben nur verlernt, sich jeden Schrittes bewußt zu sein. Werden Menschen in eine Arbeit eingewiesen, muß man jeden Schritt erklären. Davor zerlegen Sie den Vorgang, der für Sie bereits Routine ist, in Schritte. Ihr Vorteil ist, daß Sie dem Lernenden zeigen, wie es geht – dem Computer können Sie es nur beschreiben. Programmieren bedeutet also generell: Beschreiben eines geforderten Vorgangs.

Die niedrigste Ebene, auf der Menschen ihre Handlungen kontrollieren, ist die Bewegung. Welche Muskeln in welchem Maße zu spannen oder zu entlasten sind, steuern Sie nicht mehr bewußt. Nach dieser Regel sieht die Beschreibung des Vorgangs »Essen eines Kekses« aus wie in Tabelle 1 dargestellt.

Obwohl dieser Vorgang einleuchtend beschrieben ist, hat er einen Fehler. Vor Schritt 8 haben Sie die Lehne des Stuhls nicht losgelassen. Sollte es Ihnen tatsächlich gelingen, die Hand mit dem Stuhl bis zur Keksdose zu bewegen, würde sich ein häßlicher Unfall ereignen. Natürlich würden Sie niemals einen derartig gravierenden Fehler ausführen, der Computer sehr wohl – wenn er wie ein Mensch gebaut wäre. Im Gegensatz zu Menschen ist der Rechner nicht in der Lage, falsch beschriebene Vorgänge zu erkennen und so nachzuvollziehen, wie er es für richtig hält. Er muß genau das tun, was ihm erklärt wurde. Fehlende oder falsch angeordnete Schritte verursachen immer einen falsch ausgeführten Vorgang.

Die Durchführung eines Vorgangs läßt sich in drei grundlegende Elemente aufspalten

Das Beispiel gibt noch mehr her. Wäre da nicht der Fehler mit dem Stuhl, würden Sie nach dieser Anleitung zum Stuhl »rückwärts« zurückgehen. Das ist kein wirklicher Fehler, maßgebend ist das Ergebnis. Sitzen Sie also am Schluß des Vorgangs wieder am Tisch und haben einen Keks im Magen, ist der Vorgang korrekt abgelaufen. Manchmal ist es eben schneller, unorthodoxe Wege zu gehen. Generell ist der Computer nicht in der Lage, vorwärts beschriebene Vorgänge rückwärts auszuführen, Sie müssen sie dann eben in umgekehrter Reihenfolge beschreiben. Drehungen (wie hier um 90 Grad) müssen Sie natürlich auch in Gegenrichtung ausführen. Sie sehen alle drei grundlegenden Elemente eines Vorgangs.

- Arbeiten
- Entscheidungen
- Verzweigungen

»Arbeit« ist jede Handlung. Die Keksdose vom Schrank zu nehmen, ist eine Folge von Arbeiten. »Entscheidungen« sind abhängig von bestimmten Faktoren. Die sind zu prüfen, und eine entsprechende Handlung ist zu vollziehen:

Wenn die Keksdose nicht leer ist, **dann** nimm einen Keks heraus und iß ihn, **sonst** schließe die Dose und lege sie zurück.

»Verzweigungen« finden Sie häufig gleich hinter Entscheidungen, um Vorgangselemente zu überspringen oder zu wiederholen. Das Kauen des Kekses, bis Sie ihn schlucken können, beinhaltet eigentlich schon eine solche Kombination (Tabelle 2). Sobald von »14 b.« nach »14 a.« verzweigt wurde, wird nochmal gekaut und der Vorgang bei »14 b.« fortgesetzt. Das passiert so lange, bis der Keks weich genug ist. Man nennt diesen Zustand »Abbruchbedingung«, denn mit seinem Eintreten wird der Arbeitskreis (die Schleife) abgebrochen.

Ein Programmablaufplan stellt die einzelnen Schritte übersichtlich dar

Sie sehen, daß selbst so ein vermeintlich einfacher Vorgang, in seine Schritte zerlegt, komplex ist. Und obwohl es sich nicht um einen unbekannten Vorgang handelt, schleichen sich bei der Beschreibung logische Fehler ein. Sie können jedem Vorgangselement ein grafisches Symbol zuordnen, was die Beschreibung erheblich lesbarer gestaltet. Zudem erkennen Sie Fehler schneller.

Arbeiten werden einfach umrahmt, Entscheidungen umschließen Sie mit Rauten. Die Vorgangselemente verbinden Sie mit Linienpfeilen, wodurch der Fluß des Vorgangs deutlich wird. Die Elemente des Programmablaufplans (PAP) sind damit eigentlich komplett. So sind Verzweigungen leicht darzustellen. Lautet die Antwort einer Abfrage »nein«, geht der Ablauf normal nach unten weiter. Bei »ja« führt man die Bearbeitung rechts fort. So stellt sich der Vorgang »Nachschlagen eines Begriffs im Lexikon« grafisch dar (Bild 1).

Zuerst nehmen Sie also das Lexikon zur Hand, schlagen es auf und lesen den ersten Begriff auf der ersten Seite. Ist es der gesuchte, lesen Sie die Ausführung dazu und sind schon fertig. Andernfalls gehen Sie zum nächsten Begriff. Ist er es nicht, dann gehen Sie

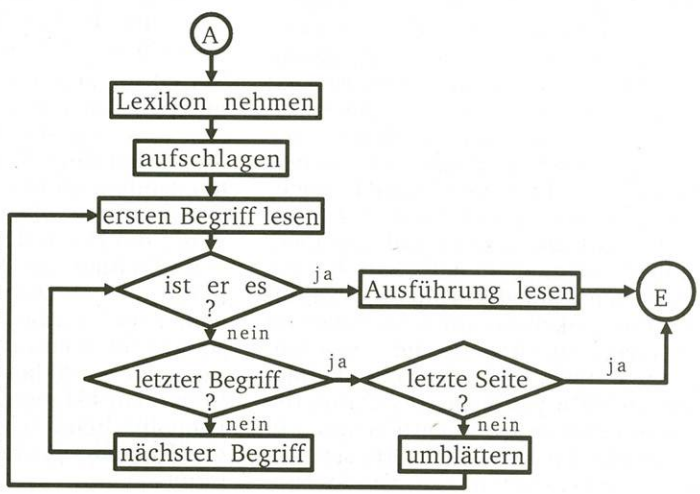


Bild 1. Ein Programmablaufplan zum Nachschlagen eines Begriffs im Lexikon. Anfang und Ende des Vorgangs sind mit dem umkreisten Buchstaben A beziehungsweise E dargestellt

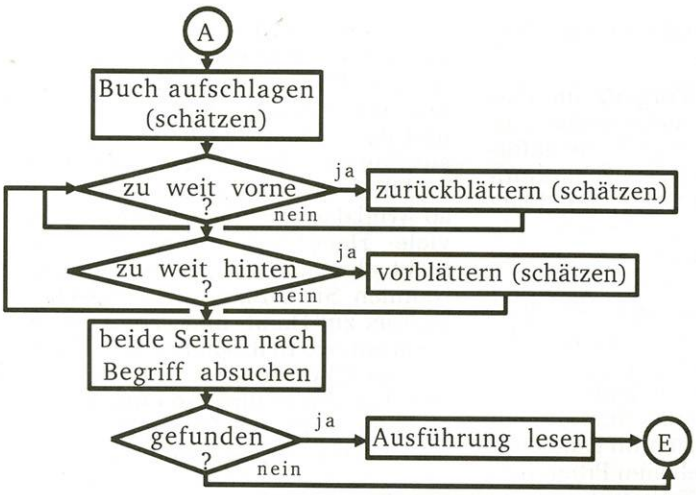


Bild 2. So ist das Nachschlagen im Lexikon zwar komplizierter, aber wesentlich effizienter

weiter zum nächsten, und zwar bis zum Ende der Seite (der zwei Seiten). Dann blättern Sie um und fangen wieder beim ersten Begriff an und so weiter. Sind Sie am Ende der letzten Seite angekommen, ohne den Begriff zu finden, endet die Suche. Kurz: Sie durchsuchen das ganze Lexikon nach einem Wort.

»Moment«, werden Sie sagen: »so mache ich das nicht, das dauert bis Weihnachten!« Nein, so sollten Sie das wirklich nicht angehen, es ist aber sehr einfach zu beschreiben. Würde man dem Computer eine solche Arbeit auftragen, wäre das unter Umständen der richtige Weg. Voraussetzung ist, daß sich die Zeit, die er für diese Arbeit braucht, im vertretbaren Rahmen hält. Der Rechner verarbeitet solche Dinge natürlich viel schneller, als ein Mensch es könnte.

Sie würden das Lexikon schon dort aufschlagen, wo Sie schätzen, daß Begriffe mit dem Anfangsbuchstaben des gesuchten Begriffs behandelt werden. Liegen Sie zu weit hinten, blättern Sie eine Anzahl abgeschätzter Seiten zurück. Haben Sie zu weit vorne aufgeschlagen, blättern Sie vorwärts. Dann beginnen Sie sich nach den Begriffen in der obersten Zeile jeder Seite zu orientieren und blättern entsprechend. Ist der Begriff auf der rechten Seite oben größer, dann suchen Sie rückwärts; ist er kleiner, blättern Sie vorwärts. »Größer« und »kleiner« bedeutet, daß der erste Buchstabe im Wort vor beziehungsweise nach dem ersten Buchstaben des gesuchten Wortes im Alphabet vorkommt. Sind die Buchstaben gleich, gilt jeweils die nächste Stelle. Entsprechend ist »Aal« größer als »Aachen«, aber kleiner als »Aas«. So sieht Ihr Suchsystem im Ablaufdiagramm aus (Bild 2).

Bei der praktischen Arbeit wurde den Programmierern sehr bald klar, daß die Darstellung im PAP zu unübersichtlich ist. Zur »Arbeitserleichterung« fügte man nach und nach immer mehr grafische Symbole hinzu. Wer PAP lesen wollte, mußte sie kennen. Damit war der ursprüngliche Effekt verloren. Als Alternative entstand der Kastenplan, auch Nassi-Schneidermann-Diagramm genannt. Er wurde so arrangiert, daß es kaum eine Chance gab Symbole hinzuzufügen. So hat er sich bis heute in fast unveränderter Form erhalten. Der Kastenplan stellt Arbeiten nach wie vor in Kästen dar. Weil der normale Arbeitslauf von oben nach unten erfolgt, hat man weitgehend auf Pfeile verzichtet.

Noch übersichtlicher als Programmablaufpläne sind Kastenpläne

Nur bei Verzweigungen sind sie noch notwendig. Auf Verzweigungen sollten Sie bei der Beschreibung ohnedies verzichten; sie verliert dadurch an Struktur. Eine Entscheidung besteht auch aus einem Kasten, doch sind zwei Linien eingezogen, die den Fortlauf des Vorgangs abhängig von der Entscheidung trennen. Am Beispiel »abendliches Stellen des Weckers« zeigt Bild 3 einen Kastenplan.
Nach der Abfrage stehen die weiteren Arbeiten nebeneinander. Nur die linken oder rechten Arbeiten werden ausgeführt, je nachdem, ob die Antwort auf die Abfrage »ja« oder »nein«

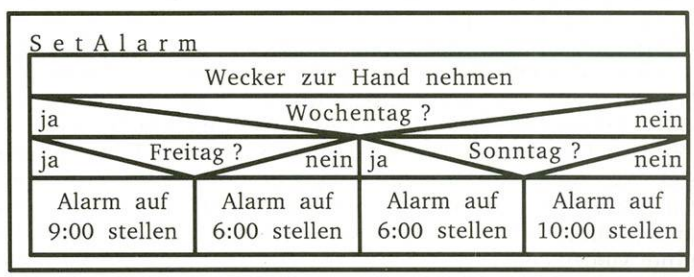


Bild 3. Dieser Kastenplan berücksichtigt die unterschiedlichen Weckzeiten im Laufe einer Woche

Schritt	Teilaufgabe
1.	Steh auf (strecke die Beine und halte Balance)!
2.	Bewege rechte Hand zur Stuhllehne und greife sie!
3.	Gehe um den Tisch herum zum Schrank (90-Grad-Drehung nach rechts, zwei Schritte vorwärts, 90-Grad-Drehung nach links, fünf Schritte vorwärts)!
4.	Bewege den Arm mit dem Stuhl in der Hand vor den Schrank mit der Lehne zum Schrank!
5.	Kopf um 60 Grad nach rechts drehen und prüfen, ob jemand kommt!
6.	Kommt jemand, Vorgang rückgängig machen, sonst:
7.	Rechten Fuß bis zur Sitzfläche des Stuhls heben, Fuß auf die Sitzfläche stellen, Gewicht nach vorne verlagern und Bein strecken (Balance halten), linken Fuß neben den rechten positionieren!
8.	Linken Arm in die Höhe strecken, soweit es geht!
9.	Reicht die Länge nicht, Vorgang rückgängig machen, sonst:
10.	Nach Keksdose tasten und greifen!
11.	Dose öffnen!
12.	Dose leer? Wenn ja, Vorgang rückgängig machen, ärgern, sonst:
13.	Mit rechter Hand Keks herausnehmen, zum Mund führen und einnehmen!
14.	So lange kauen, bis das Produkt zu schlucken ist, schlucken!
15.	Noch Appetit? Wenn ja, dann Vorgang ab Punkt 13 wiederholen, sonst:
16.	Keksdose schließen, zurücklegen!
17.	Linken Fuß anheben und zur Erde herabsetzen, rechten nachziehen!
18.	Stuhl an der Lehne greifen!
19.	Vorgang von Punkt 4 bis 1 rückwärts ausführen!

Tabelle 1. Die einzelnen »Schritte«, die Sie zum Essen eines Kekses durchlaufen müssen

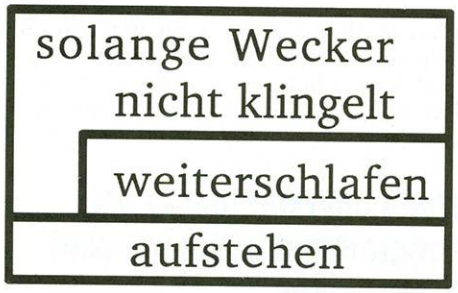


Bild 4. Solange der Wecker nicht klingelt, fährt der Schlafende mit seiner Tätigkeit fort

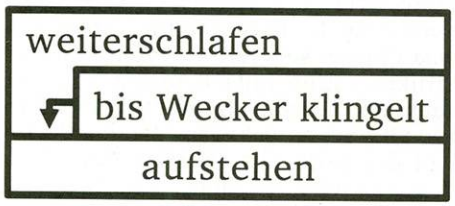


Bild 5. Erst wenn der Wecker klingelt, ist der Teil des Vorgangs beendet

lautet. Über der Teilung steht die Abfrage selbst. Es gibt noch eine Reihe wichtiger Elemente in dieser Darstellungsart, vor allem, um Verzweigungen aus dem Weg zu gehen, zum Beispiel »Ausführen eines Vorgangs, *solange* eine Abfrage mit »ja« zu beantworten ist« (Bild 4). Ein Vorgang könnte auch so lange auszuführen sein, *bis* ein erfragter Faktor zutrifft (Bild 5). Der Pfeil deutet auf das Ende des zu wiederholenden Vorgangs. Darunter stehen Arbeiten, die danach auszuführen sind. Es gibt auch noch eine Mehrfachabfrage. Sie stellt Abfragen dar, deren Antwort nicht nur »ja« oder »nein« ist (Bild 6).

All das müßten Sie im PAP auf dem üblichen Weg darstellen. Deshalb ist PAP lange nicht so übersichtlich wie der Kastenplan. Um eine konkrete Vorstellung davon zu bekommen, sehen Sie Überlegungen zum Lottogewinn noch einmal, nun als Kastenplan dargestellt (Bild 7).

Unterroutinen verringern die Programmlänge ganz entscheidend

Wenn Sie den gleichen Teilvorgang immer wieder anzuordnen hätten, wäre es sehr aufwendig, ihn jedesmal neu zu beschreiben. Es genügt, ihn einmal zu beschreiben. Da, wo er gebraucht wird, befehlen Sie nur noch: »Führe diesen bereits beschriebenen Vorgang aus«. Ist dieser kleine Vorgang abgeschlossen, geht es da weiter, wo er angefordert wurde, genauer gesagt, beim Arbeitsschritt unmittelbar danach.

- Nimm eine Schraube, stecke sie in das Loch und drehe sie mit dem Schraubenzieher hinein,
- und das mache jetzt auch mit allen anderen Löchern.

Man nennt diesen Vorgang im Vorgang »Unterroutine« oder später »Unterprogramm«. Dort, wo Sie ihn anfordern, setzen Sie einen »Unterprogrammaufruf«. Nicht nur, um Platz und Arbeit zu sparen, verwenden Sie Unterprogramme, auch um die Beschreibung lesbarer zu gestalten.

Darüber hinaus vereinfacht die Verwendung von Unterroutinen die »Wartung« von fertigen Projekten. Stellen Sie sich vor, Sie hätten an Ihren Computer einen bestimmten Drucker angeschlossen und geben mit einem selbstgeschriebenen Programm Tabellen aus. Wie Sie beispielsweise waagerechte und senkrechte Striche drucken, ist bei Druckern unterschiedlicher Hersteller oft anders geregelt. Wenn Sie ein Unterprogramm eingerichtet haben, das waagerechte Linien zieht, dann genügt es, nur diese eine Stelle im Programm zu ändern. Für eine Unterroutine entwerfen Sie einen eigenen Ablaufplan und geben ihm einen Namen. Im Programmablaufplan sieht der Aufruf der Routine dann aus wie in Bild 8.

Auch der Kastenplan kennt ein Symbol für Unterprogrammaufrufe. Für den angeforderten Vorgang besteht natürlich ein eigener Kastenplan. Sie sehen hier, daß man einer Unterroutine auch auftragen kann, eines oder mehrere übergebene Objekte miteinzubeziehen (Bild 9). Sie haben jetzt den wahrscheinlich größten Schritt hinter sich; noch können Sie aber keinen Computer programmieren. Die nächste Schwierigkeit besteht darin, die Sprache des Rechners zu erlernen. Zuvor erfahren Sie noch einiges Wissenswerte über mathematische Grundlagen, die beinahe für jede Programmiersprache gelten.

Der Computer ist eine Rechenmaschine, deshalb arbeitet er nur mit Zahlenwerten. Was er bekommt, sind Zahlen, was er damit macht, ist Rechnen, und das Ergebnis besteht wieder nur aus Zahlen. Selbst gespeicherte Texte und Bilder legt der Rechner als Zahlen ab. Weil das so ist, kann der Computer viele Handlungen des Menschen nicht ohne weiteres nachvollziehen. Nehmen Sie einmal einen Taschenrechner zur Hand, und lassen Sie ihn schätzen, wo in Ihrem Lexikon Wörter behandelt werden, die mit »F« beginnen. Das dürfte nicht so einfach sein. Für eine grobe Schätzung genügt es, wenn Sie wissen, an welcher Stelle im Alphabet das »F« vorkommt. Ein Lexikon besteht aus 26 Abschnitten, einen für jeden Buchstaben. Im ersten 26stel liest man Wörter mit dem Anfangsbuchstaben »A«, im zweiten die mit »B« und so weiter. Teilen Sie nun die Gesamtzahl der Seiten im Lexikon durch 26, dann bekommen Sie die Anzahl der Seiten für einen Buchstaben. Vor dem »F« liegen fünf Buchstaben. Multiplizieren Sie die Seitenzahl pro Buchstabe mit 5, so erhalten Sie die Seite, an der »F« beginnen müßte (weniger 1).

Computer kennen nichts anderes als Zahlen

Tatsächlich kommen Sie damit etwa am Anfang von »G« an, als Schätzung ist es nicht schlecht. Bei der Suche nach einem Wort mit »Y« dürften Sie aber gründlich daneben liegen. Das hat seine Ursache in der einfachen Tatsache, daß im Lexikon nicht jedem Buchstaben gleich viele Seiten gewidmet sind. Um etwas genauere Schätzungen zu bekommen, legen Sie sich also eine Tabelle an, die die prozen-

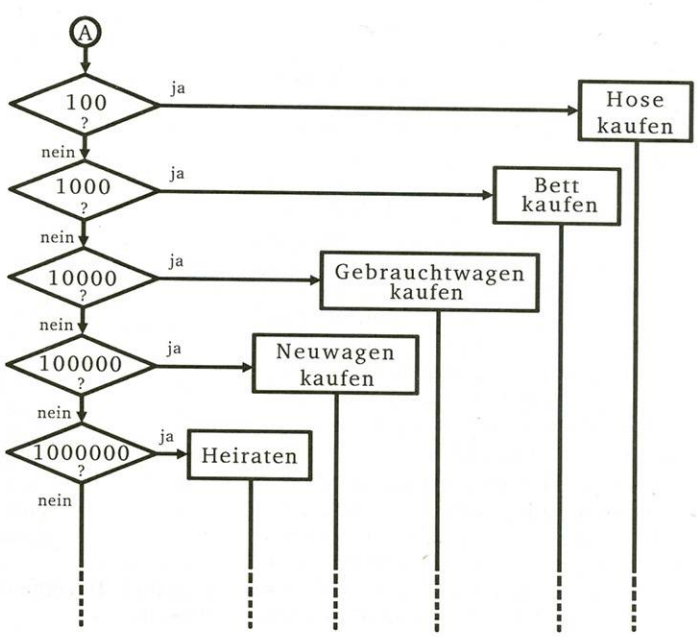


Bild 6. Nach einem Lottogewinn gibt es je nach Summe mehrere Anschaffungsmöglichkeiten

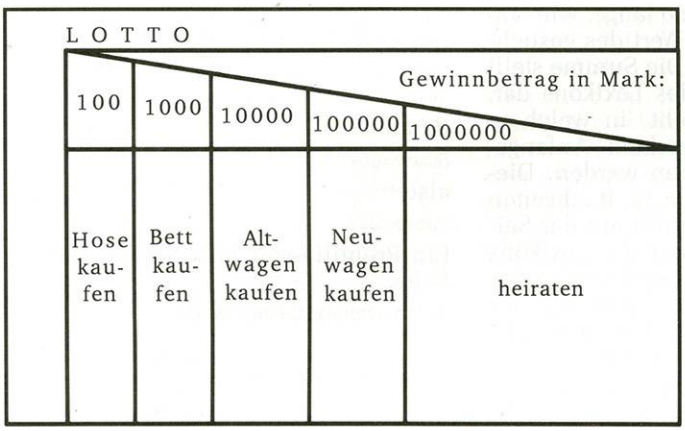


Bild 7. Diese Darstellungsart ist wesentlich übersichtlicher als die aus Bild 6, erfordert aber zunächst mehr Aufwand bei der Ausarbeitung

tualen Anteile jedes Buchstabens am Lexikon aufführt (Tabelle 3).
Die Prozentangaben sind jetzt sogar etwas genauer, als ein Mensch sie im Kopf haben könnte. Würde der Rechner auf Basis dieser Tabelle schätzen, läge er sicher besser als ein Mensch. Probieren Sie es. Suchen Sie sich einen Buchstaben aus dem Alphabet, und schlagen Sie das Lexikon an der Stelle auf, an der dieser Buchstabe schätzungsweise beginnt (nicht unbedingt »A« oder »Z«). Notieren Sie sich die Seitennummer. Schlagen Sie jetzt die letzte Seite auf, und subtrahieren Sie von dieser Seitennummer die Anzahl der Seiten, auf denen keine Begriffserläuterungen stehen (Vorwort, Quellennachweise etc.). Teilen Sie den Wert durch 100. Das Ergebnis ist ein Prozent des Lexikons in Seiten; notieren Sie es. Addieren Sie jetzt alle Werte in der Spalte »%« der Tabelle für jeden Buchstaben, der vor dem gesuchten steht. Multiplizieren Sie das Ergebnis mit der bereits ermittelten Seitenzahl für ein Prozent. Das Ergebnis ist die geschätzte Seitennummer, an dem der gewählte Buchstabe beginnt (weniger 1 und Vorwort etc.). Diese Schätzung liegt sehr wahrscheinlich besser als die Ihrige.

»Felder« können Sie mit dem Computer überaus effektiv beackern

Sie sehen, das Schätzen der Position eines Buchstabens im Lexikon kann durchaus ein mathematischer Vorgang sein. Erst in dieser Form machen Sie dem Rechner klar, was Sie von ihm verlangen. Die Tabelle liegt für das Programm als ein Feld von Daten vor, auf das es mit einem Zeiger zugreift. Sie müssen sich eine Reihe von Schubläden vorstellen. In der ersten Schublade liegt der Prozentsatz für den Buchstaben A, in der zweiten Schublade der für B und so weiter. Die Buchstaben selbst sind nicht mehr Buchstaben, sondern Zahlen von 0 bis 25. So dient der Zahlenschlüssel für die Buchstaben gleichzeitig als Zeiger auf die Schublade, in der der zugehörige Prozentsatz liegt.

Einen Zeiger auf ein Element in einem Zahlenfeld nennt man »Index«. Das Feld selbst ist damit »indiziert«. Schubladen gibt man üblicherweise Namen, in diesem Fall heißt die ganze Kommode zum Beispiel »Prozente« (Bild 10). Die erste Schublade darin (die für »A«) heißt »Prozente(0)«. Die Null ist der Index auf die erste Schublade in der Kommode »Prozente«. Damit Sie den Index selbst ändern können, legen Sie ihn auch in einer Schublade ab. Die hängt dann in einer ganz anderen Kommode, die keinen Namen braucht. Hier bekommt die Schublade selbst gleich den Namen, zum Beispiel »Index«. Manche Buchstaben sind mit so wenigen Seiten vertreten, daß der prozentuale Anteil fast Null ist. Noch genauer liegen Sie dann, wenn Sie mit Promille statt mit Prozent arbeiten (»von Tausend« statt »von Hundert«). Die Werte finden Sie in der letzten Spalte der Tabelle 3. Ein Promille des Lexikons ist die Gesamtseitenzahl durch 1000. Das Feld, das die Promilleanteile jedes Buchstabens im Lexikon beinhaltet, heißt »Promille«. Es hat 26 Elemente (0 bis 25) und enthält die Werte aus der letzten Spalte der Tabelle 3. Wegen der Genauigkeit arbeiten alle Beispiele von nun an mit diesem Feld (Bild 11).
Packt man eine Null in die Schublade »Index«, dann benennt »Promille(Index)« das erste Element in der Kommode »Promille«, es ist das für »A«. Addieren Sie nun eine Eins zu »Index«, dann erreichen Sie mit »Promille(Index)« das zweite Feld – er zeigt dann auf den Promillesatz für »B«, denn Index steht auf 1.

Schleifen ersparen dem Programmierer Tipparbeit

Eigentlich will es auf Anhieb gar nicht einleuchten, warum Sie eine Schublade (»Variable«) verwenden müssen, um mit einem Index auf ein Datenfeld zuzugreifen. Sie könnten ja gleich den Index der Schublade angeben, für die Sie sich gerade interessieren. Dann bräuchten Sie dazu aber immer einen anderen Befehl. Einmal muß »Promille(0)« gelesen werden, danach »Promille(1)« und so weiter. Mit dem Index in einer Variablen verwenden Sie immer nur »Promille(Index)« und ändern dazwischen nur die Variable »Index«. Tabelle 4 enthält die einzelnen Schritte, die dafür notwendig sind.
Die Anweisung in Zeile 2 liest den Wert aus der ersten Schublade in der Kommode »Promille«, weil der Zeiger durch die vorangegangene Anweisung auf 0 gesetzt ist. Der Befehl in Zeile 4 bezieht sich bereits auf die zweite Schublade. Zwischenzeitlich wurde der Zeiger »Index« um den Wert 1 erhöht, womit er auf die zweite

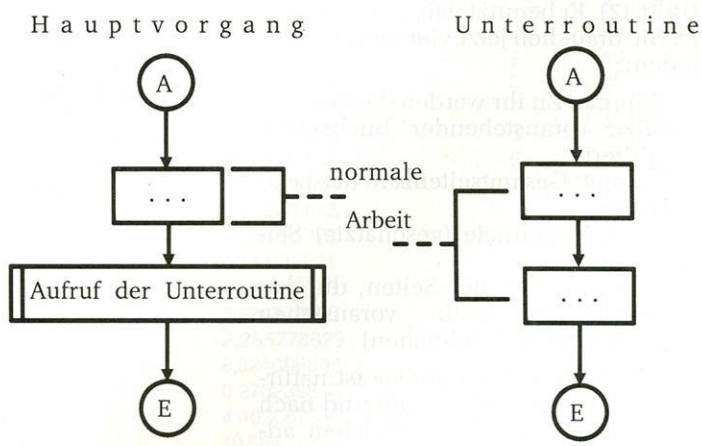


Bild 8. Der Aufruf verzweigt zum Beginn der Unterroutine und kehrt bei Ende des Vorgangs zum Punkt unter dem Aufruf zurück

Schritt	Teilaufgabe
14 a.	kauen
14 b.	Wenn du den Keks noch nicht schlucken kannst, verzweige zu 14 a.
14 c.	Sonst schlucke ihn

Tabelle 2. Teilaufgabe 14 aus Tabelle 1 läßt sich noch aufspalten

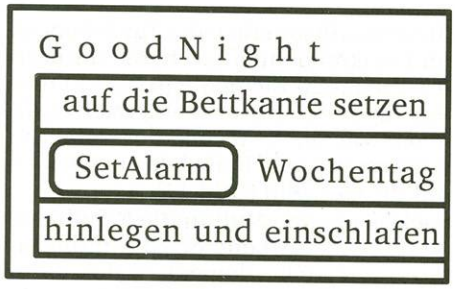


Bild 9. Die Unterprogramme arbeiten jeweils mit dem übergebenen Objekt. Das Unterprogramm »SetAlarm« erhält den Parameter »Wochentag«.

Schublade (Nummer 1) zeigt. Sie könnten nun im gleichen Stil fortfahren, hätten dadurch aber überhaupt nichts eingespart. Für jeden Lesevorgang brauchen Sie jetzt zwei Anweisungen: »Index erhöhen« und »Feld lesen«.

Sie erinnern sich sicher noch an die Verzweigung. In der Anweisungszeile 4 könnte stehen: »Verzweige zu Zeile 2«. Nach dieser Beschreibung würden die Anweisungen in Zeile 2 und 3 in einer Schleife ausgeführt, und zwar bis zum Ausschalten des Computers. Besser verwenden Sie eine Abfrage mit Verzweigung. Dazu ersetzen Sie die Zeilen 4 und 5 durch die folgenden.

```
4. Wenn der Index kleiner als 26 ist, verzweige zu Zeile 2
5. Ende
```

Zeile 2 liest jetzt immer ein Element nach dem anderen, weil in Zeile 3 der Index erhöht wird und Zeile 4 wieder zu Zeile 2 verzweigt, sofern noch nicht alle Felder gelesen worden sind. Nun ist es natürlich nicht sehr sinnvoll, alle Felder zu lesen, ohne mit den gelesenen Werten zu arbeiten. Der folgende Kastenplan addiert alle Felder bis zu dem für den gesuchten Buchstabens (Bild 12). Er benutzt dabei eine Schleife. Sie brauchen jetzt vier neue Schubladen:

- Summe: Zu ihr werden die Prozentsätze voranstehender Buchstaben addiert
- Gesamt: Gesamtseitenzahl des Lexikons
- Seite: Errechnete (geschätzte) Seitennummer
- Voran: Anzahl der Seiten, die den Begriffserklärungen voranstehen (Vorwort und dergleichen)

Der Startwert von »Summe« ist natürlich 0. Zu ihm werden nach und nach die Prozentsätze der Buchstaben addiert, die dem gesuchten im Alphabet voranstehen. »Index« beginnt ebenfalls mit 0. Er zeigt zu Anfang auf die erste Schublade im Feld »Promille«. Nun beginnt das Programm, den Promillesatz des Buchstabens, auf den »Index« zeigt, zur Summe zu addieren. »Index« selbst zählt es um den Wert 1 hoch. Das geschieht so lange, wie »Index« kleiner als der Wert des gesuchten Buchstabens ist. Die Summe stellt den Promilleanteil des Lexikons dar, der vor dem Teil steht, in welchem Worte mit dem gesuchten Anfangsbuchstaben besprochen werden. Diesen Wert rechnen Sie in Buchseiten um, indem Sie »Summe« mit der Seitenzahl für ein Prozent des Lexikons multiplizieren. Jetzt haben Sie die Anzahl der Seiten, die vor den Worten mit dem gesuchten Buchstaben stehen. Es fehlen aber noch die Seiten, die der Autor für ein Vorwort und andere Dinge verwendet hat. Addieren Sie die Anzahl dieser Seiten, ist das Ergebnis die Seitennummer, an der Begriffe mit dem geforderten Anfangsbuchstaben etwa beginnen (weniger 1). Betrachten Sie das Ganze nochmals anhand eines Beispiels:

Ihr Lexikon hat 1020 Seiten. Begriffserklärungen mit dem Anfangsbuchstaben »A« beginnen auf Seite 4. Im hinteren Teil des Lexikons sind 17 Seiten mit Quellen- und Bildnachweisen gefüllt. Dadurch ergibt sich eine (ideale) Gesamtseitenzahl von 1000 relevanten Seiten.

```
Gesamt=1000
Voran=3
```

Gesucht sind Begriffe mit dem Anfangsbuchstaben »F«. Der steht an sechster Stelle im Alphabet, hat also den Indexcode 5 (Zählung ab 0). Der Promilleanteil aller davorliegenden Seiten ist die Summe der Felder »Promille(0)« bis »Promille(4)«.

```
Summe=Promille(0)+Promille(1)+Promille(2)+Promille(3)+Promille(4)
```

daraus folgt:

```
Summe=66+73+29+40+35
```

also

```
Summe=243
```

Ein Promille von 1000 Seiten ist eine Seite.

```
Seite=Summex(Gesamt/1000)
```

oder

```
Seite=243x(1000/1000)
```

somit

```
Seite=243
```

Zu dieser Seitennummer ist die Anzahl der Seiten zu addieren, die den Begriffserklärungen voranstehen.

```
Seite=Seite+Voran
```

also

```
Seite=Seite+3
```

dadurch

```
Seite=246
```

Bis zu dieser Seitennummer stehen die Begriffe vor dem Buchstaben F. Eine Seite weiter beginnt die gesuchte Rubrik.

```
Seite=Seite+1
```

damit

```
Seite=247
```

Dieser Wert ist die geschätzte Seitennummer, an der Begriffe mit dem Anfangsbuchstaben F zu finden sind. Die letzte Addition ist nur der Korrektheit

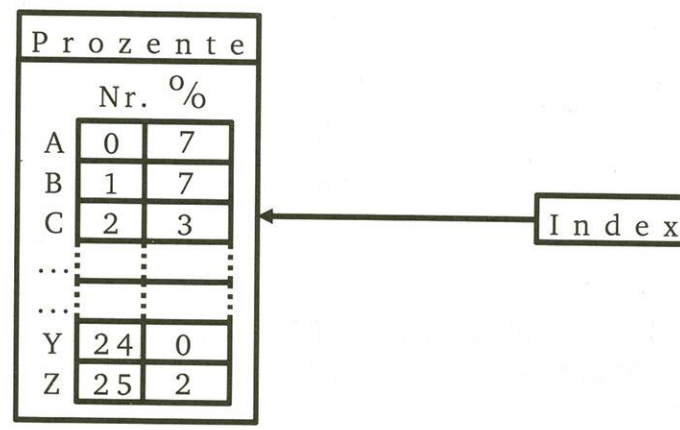


Bild 10. Die Kommode »Prozente« hat 26 (0 bis 25) Schubladen. Der »Index« zeigt jeweils auf eine von ihnen.

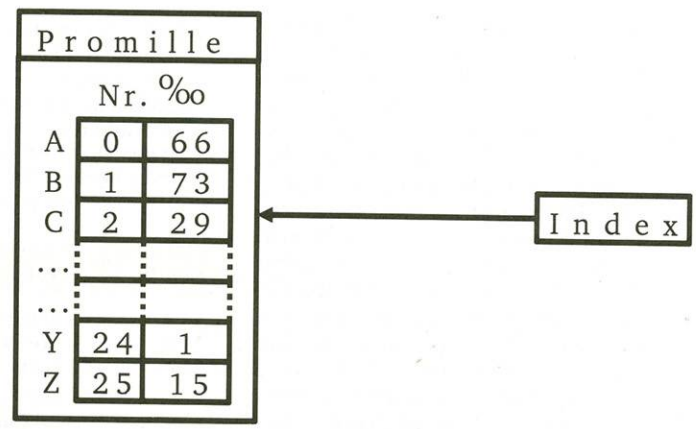


Bild 11. Das Feld »Promille« enthält die Promilleanteile der einzelnen Buchstaben am gesamten Seitenumfang des Lexikons

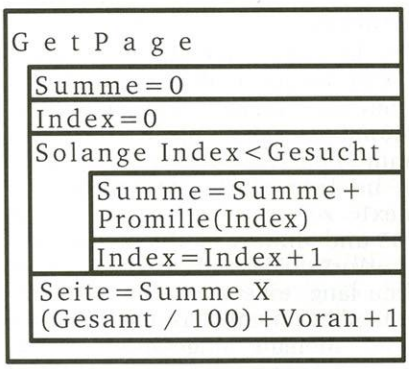


Bild 12. Der computergerechte Kastenplan für den Vorgang »Schätze Seite im Lexikon«

wegen nötig, denn es ist in jedem Fall nur eine Schätzung. Weitgehend genaue Ergebnisse liefert dieser Vorgang nur bei der Arbeit mit einem bestimmten Lexikon, auf das die Promilleanteile der mit den einzelnen Buchstaben beginnenden Begriffe abgestimmt sind.

Die Addition der Promilleanteile zu »Summe« funktioniert mit einer Schleife, einer sogenannten »Zählschleife«. Sie zählt von 0 bis »Gesucht –1« und erledigt dazwischen die Addition. Die klassische Form der Zählschleife lernten Sie schon in Ihrer Kindheit kennen. Sie standen mit verdeckten Augen in der Ecke und zählten bis hundert, während Ihre Freunde sich in den umliegenden Schlupfwinkeln verkrochen. Eine solche Zählschleife ist leer. Das bedeutet, daß innerhalb der Schleife keine Arbeiten vorgenommen werden. Eine leere Zählschleife befehlen Sie hier wie dort nur, um den Ausführenden eine festgelegte Zeit warten zu lassen (Bild 13).

Um die Dauer zu steigern, setzen Sie normalerweise nur den Endwert herauf. Wie würden Sie nun aber vorgehen, wenn Sie bis zehntausend zählen müßten, aber nur bis hundert zählen könnten? Natürlich können Sie bis zehntausend zählen, und der Computer kann es auch. Tatsächlich gibt es für den Rechner aber Grenzen. Die bestehen zwar nicht reell, zuweilen ist es aber günstiger, auf hohe Grenzwerte zu verzichten. Sie würden in einem solchen Fall wahrscheinlich hundertmal bis hundert zählen. Dazu müssen Sie sich zwei Zähler merken. Mit dem einen Zähler zählen Sie immer wieder bis hundert, und mit dem zweiten Zähler zählen Sie, wie oft Sie schon bis hundert gezählt haben. Es sind damit also zwei Schleifen. Die langsamere ist die äußere, und die schnelle ist die innere Schleife. Eine reine Zählschleife ist nur die innere Schleife, die äußere zählt ja nicht nur einen Zähler hoch. Sie enthält selbst eine zweite Schleife. Das Schleifensystem nennt man »verschachtelt«. Für jede Addition der äußeren Schleife zählt die innere komplett durch (Bild 14). Das läßt sich beliebig weiter ausführen. Sie könnten zum Beispiel »zählen, wie häufig Sie schon hundertmal gezählt haben, wie häufig Sie schon bis hundert gezählt haben«.

Es kann natürlich nicht zu den vorrangigen Aufgaben eines Computers gehören, zu zählen und nichts weiter zu tun. Sie verwenden Schleifen grundsätzlich dann, wenn Aufgaben mehrmals zu wiederholen sind. Darin beziehen Sie sich auf ein Feldelement, auf das der Zähler der Schleife als Index zeigt. Eine Schleife beginnt bei einem Startwert und endet beim Endwert. Die Differenz der beiden entspricht der Anzahl der Durchläufe einer Schleife (minus 1). Mit den Anweisungen

```
Variable=Startwert
Solange Variable < Endwert: Addiere 1 zu
Variable
```

läuft die Schleife »Endwert« minus »Startwert« Durchläufe ab. Würden Sie jedesmal den Wert 2 addieren, wäre die Anzahl der Durchläufe aber nur noch halb so groß. Mit jedem Durchlauf nähert sich dann »Variable« dem »Endwert« um 2. Eine Schleife kann auch rückwärts laufen. Der Startwert muß nur größer sein als der Endwert. Die Schleifenvariable wird dann nicht mehr hoch-, sondern herabgezählt. Die Schrittweite ist negativ, kleiner als 0 (Bild 15).

Textverarbeitungsprogramme suchen Wörter im Text wahlweise vorwärts oder rückwärts. Die Schleife, mit der das Programm den Text absucht, läuft entsprechend vorwärts oder rückwärts. Die einzelnen Buchstaben des Textes liegen in einem Datenfeld (Bild 16). Selbst die Räume zwischen den Wörtern sind Leerzeichen und belegen je einen Speicher im Datenfeld.

Suchen ist eine Routineaufgabe, für die Computer wie geschaffen sind

Einen bestimmten Buchstaben im Text zu finden, ist nicht allzu schwer. Sie verwenden eine Schleife, die vom Anfang des Textes bis zum Ende jeden Buchstaben liest und mit dem gesuchten vergleicht (Tabelle 5).

Schwieriger wird es, wenn Sie mehrbuchstabige Anteile eines Textes suchen, zum Beispiel ein Wort. Für das gesuchte Wort reicht dazu keine Variable mehr, man braucht ein weiteres Datenfeld. Dieses Datenfeld soll beispielsweise das Wort »ist« enthalten (Bild 17). Um ein Wort im Satz zu finden, suchen Sie zunächst den ersten Buchstaben des Wortes im Text. Ist er gefunden, vergleichen Sie den folgenden Buchstaben im Text mit dem nächsten im Wort. Stimmen sie nicht überein, suchen Sie wieder nach dem ersten Buchstaben des Wortes im Text weiter.

Bei der Suche nach dem Wort »IST« gehen Sie also folgenderweise vor. Sie beschränken sich zunächst auf den Buchstaben »I«. Ihn finden Sie schon im Element 2 des Feldes »Text«. Wenn der folgende Buchstabe im Text ein »S« ist, könnte die Suche schon bald zu

Buchstabe	Seitenzahl durch 1%	%	°/₀₀
A	6,620876027	7	66
B	7,296475622	7	73
C	2,916338250	3	29
D	3,974777615	4	40
E	3,490597906	3	35
F	4,109897534	4	41
G	5,269676838	5	53
H	4,672897196	5	47
I	2,330818602	2	23
J	1,553879068	2	16
K	7,037495778	7	70
L	4,132417521	4	41
M	6,710955973	7	67
N	3,524377885	4	35
O	2,285778629	2	23
P	6,024096386	6	60
Q	0,270239837	0	3
R	4,402657358	4	44
S	10,58439365	11	106
T	3,850917689	4	39
U	1,035919378	1	10
V	2,499718500	3	25
W	3,727057764	4	37
X	0,045039972	0	1
Y	0,135119918	0	1
Z	1,497579101	1	15

Tabelle 3. In der zweiten Spalte sehen Sie die Prozentanteile, in der dritten Spalte die Promilleanteile jedes Buchstabens an der Gesamtseitenzahl des Lexikons

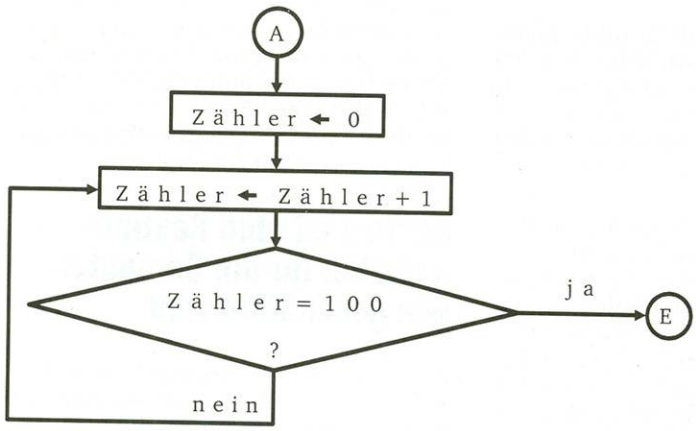


Bild 13. Die leere Zählschleife hat nur eine Aufgabe: Zählen

Ende sein. Das ist aber leider nicht der Fall. In »Text(3)« steht ein »E«. So suchen Sie weiter im Text nach dem »I« und finden es bei Element 17. Das ist wieder ein Fehlschlag, der folgende Buchstabe heißt wieder »E«. Weiter geht es bei Element 27. Dem folgt ein »H«, wieder daneben. An Position 33 steht das nächste »I«. Ihm folgt ein »S« und dem ein »T«. Das Suchwort ist gefunden. Element 33 im Feld »Text« stimmt mit »Gesucht(0)« überein, genauso »Text(34)« mit »Gesucht(1)« und »Text(35)« mit »Gesucht(2)«. Das Programm setzt voraus, daß in der Variablen »WortLänge« die Länge des gesuchten Wortes steht, hier 3. Außerdem muß »TextLänge« die Länge des Felds »Text« enthalten, für diesen Text 72.

Beim Studium des Kastenplans werden Sie auf die ersten Unschönheiten der Programmierung stoßen (Bild 19). Er gibt nicht exakt wieder, was die Beschreibung vorsieht. Eigentlich sollte eine zweite Schleife in Aktion treten, sobald der erste Buchstabe des gesuchten Wortes im Text erkannt wurde.

Das gesuchte Wort darf nicht Teil eines längeren Wortes sein

Statt dessen laufen dann aber zwei Zähler parallel. Zuerst setzen Sie »Index2« auf 0. Er bezieht sich auf das Feld »Wort«, worin das Suchwort steht. Im Text suchen Sie nun den Buchstaben, auf den »Index2« im Feld »Wort« zeigt. Das ist vorerst der erste, hier ein »I«. Ist dieser Buchstabe erkannt, erhöhen Sie parallel zu »Index1« auch »Index2«. Fortan suchen Sie den folgenden Buchstaben. Stimmt er nicht mehr mit dem im Text überein, muß »Index2« wieder zurückgestellt werden. Das geschieht deshalb, weil wieder der erste Buchstabe von »Wort« zu suchen ist. Darum wird »Index2« auch wieder auf Feld 0 zurückpositioniert, dem ersten Buchstaben des Suchwortes. Nach dieser Beschreibung endet das Programm in beiden Fällen in gleicher Weise: bei erfolgreicher und bei erfolgloser Suche. Die Abfrage »wenn Index2 = WortLänge« müßte vor dem Ausstieg eine Erfolgsmeldung ausgeben.

Nun können Sie aber immer noch nicht ganz sicher sein, daß Sie auch wirklich das gesuchte Wort gefunden haben. Das gefundene Wort »ist« könnte auch der Bestandteil des Wortes »Kapitalist« sein. Deshalb müssen Sie prüfen, ob die gefundene Textstelle ein einzelnes Wort ist. Ein Wort ist meistens von Trennzeichen umgeben. Also vor und unmittelbar nach einem ganzen Wort findet man ein Leerzeichen, ein Komma, einen Punkt, ein Semikolon, ein Fragezeichen oder ein Ausrufezeichen. Die Elemente 32 und 36 im Feld »Text« sind Leerzeichen, damit steht fest: Bei der gefundenen Textstelle handelt es sich um ein einzelnes Wort. Würden Sie nach dem Wort »KRIEG« gesucht haben, so hätte sich auf diese Weise nicht einwandfrei feststellen lassen, ob der gefundene Ausdruck (ab Index 0) wirklich ein ganzes Wort ist. Ihm steht nämlich kein Trennzeichen voran. Befindet sich das Wort am Textbeginn, können Sie davor kein Trennzeichen finden. Das muß also gesondert geprüft werden. Theoretisch kann das Wort auch am Textende stehen. Es folgt ihm dann kein Trennzeichen. Für diesen speziellen Fall gilt das nicht, denn das letzte Element im Feld »Text« ist ein Punkt. Trotzdem muß ein korrektes Programm diesen Faktor berücksichtigen (Bild 19).

Das sieht jetzt alles nicht mehr schön aus. Um ihn zu verstehen, vollziehen Sie den Kastenplan am besten für einige Kombinationen nach. Das Programm »FindWord« erkennt das Wort »ist« im Feld »Text« an Position 33. »Index1« zeigt zu diesem Zeitpunkt auf 35 und »Index2« auf Position 3 im Feld »Wort«. Dieses Feld ist nur drei Stellen lang, eine vierte Stelle (Nummer 3) gibt es nicht. Das macht nichts, da sie niemals angefordert wird. »Index2« ist nur noch als Wert wichtig. Entspricht er der Wortlänge (3), ist das Wort komplett im Text vertreten. Jetzt geht es darum, festzustellen, ob das Wort am Textende steht. Dann entspricht »Index1« nämlich der Textlänge minus 1. In einem solchen Fall erübrigt sich die Frage nach dem Trennzeichen hinter dem Wort. Nun steht es aber ziemlich in der Mitte, wodurch das Programm bei der Abfrage »Text(Index1 + 1) = Trennzeichen?« weiterarbeitet. An Position 36 (35 + 1) steht ein Leerzeichen. Bei der nächsten Abfrage geht es um den Wortbeginn. Befindet sich das Wort am Textanfang, entspricht »Index1« der Länge des Wortes minus 1. Das trifft hier nicht zu. Also muß vor dem Wort ein Trennzeichen stehen. Die Position dieses Zeichens errechnet sich durch »Index1« minus »Index2«, also 35 weniger 3. Dort steht ein Leerzeichen, womit die Textstelle als ganzes Wort erkannt ist. Die Meldung wird ausgegeben, und das Programm endet. Übrigens müssen Sie immer zuerst prüfen, ob das Wort an einer Textgrenze liegt. Nur wenn das nicht so ist, hat die Abfrage nach dem Zeichen vor beziehungsweise nach dem Wort Sinn.

Rückwärts ist das auch nicht mehr weiter schwierig. »Index1« und »Index2« beginnen bei »TextLänge« beziehungsweise »WortLänge« minus 1. Jede Addition im Programm wird zur Subtraktion und umgekehrt. Lediglich die Vorzeichen werden umgekehrt. Text- und Wortende sind jeweils erreicht, wenn die Indizes bei -1 ankommen. Die Problemstellungen werden jetzt

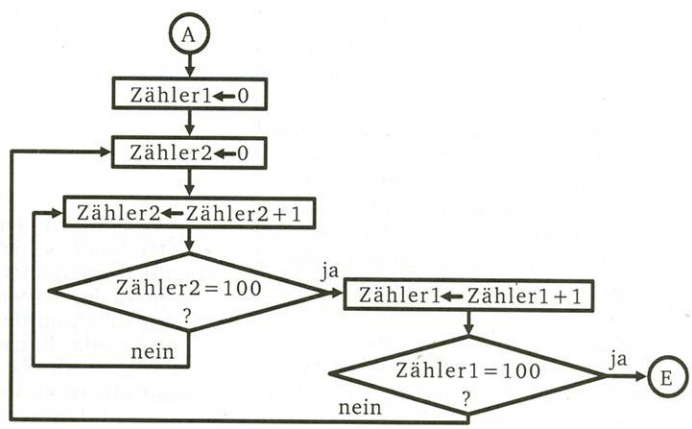


Bild 14. Verschachtelte Schleifen können beim Zählen sehr große Grenzwerte erreichen

langsam so verzwickt, daß es mehrere Lösungswege und damit mehrere Beschreibungen geben kann. Spielen Sie ruhig ein bißchen herum. Durchdenken Sie einen anderen Lösungsweg, und schreiben Sie einen Kastenplan dazu. Der muß nicht unbedingt simpler oder effizienter sein als der vorliegende.
Wie Sie im letzten Kapitel schon gemerkt haben, sollen Problemlösungen meistens auf unterschiedliche Daten anwendbar sein. Das Nachschlagen im Lexikon darf nicht nur für ein bestimmtes Lexikon gelten, die Wortsuche muß in unterschiedlichen Texten anzuwenden sein. Das ist deshalb so, weil Sie zur Zeit der Programmierung oft gar nicht wissen, mit welchem Text Sie es zu tun bekommen, welche Daten Sie zu verarbeiten haben.

Beim Sortieren spielt der Computer seine überlegene Arbeitsgeschwindigkeit aus

Daneben gibt es Probleme, die immer wieder auftauchen. Die Lösungen dazu müssen dann entsprechend allgemein gehalten sein. Man spricht von sogenannten »Standardlösungen«. »Sortieren eines Feldes« ist eines dieser Standardprobleme. Viele Menschen haben sich damit beschäftigt, und entsprechend viele Lösungen gibt es auch dazu. Stellen Sie sich also ein Feld mit zehn zweistelligen Zahlen in gemischter Folge vor (Bild 20). An erster Stelle steht eine 20. Der niedrigste Wert im Feld ist eine 15 an Position 5. Sie könnten die zwei Werte einfach vertauschen. Dann steht immerhin schon der niedrigste Wert zu Anfang. Selbst das ist aber nicht so einfach. Sie brauchen noch eine zusätzliche Variable als Zwischenspeicher. Sie heißt »Speicher«. Wenn Sie nämlich einer Variablen im Feld den Inhalt einer anderen geben, dann geht ihr vorheriger Inhalt verloren. Deshalb speichern Sie diesen Wert zuvor zwischen. Diesen Tausch nennt man »Dreiecktausch«, er geht in drei Schritten vor sich (Bild 21). Im Befehlstext schreiben Sie:

```
1. Speicher=Zahlen(5)
2. Zahlen(5)=Zahlen(0)
3. Zahlen(0)=Speicher
```

Tatsächlich funktioniert der Tausch auch ohne Zwischenspeicher. Dieser Weg ist dann aber rechnerisch aufwendig und in der Ausführung langsamer.

```
1. Zahlen(5)=Zahlen(5)+Zahlen(0)
2. Zahlen(0)=Zahlen(5)-Zahlen(0)
3. Zahlen(5)=Zahlen(5)-Zahlen(0)
```

Nehmen Sie ruhig einmal zwei Werte kleiner als 10, und überprüfen Sie diese erstaunlich simple Formel. Nun aber genug zum Thema »Tauschen«. Bis jetzt steht ja nur der niedrigste Wert des Datenfeldes zu Anfang. Alles andere ist noch ungeordnet (Bild 23, Zeile 1).
Indem Sie die erste Stelle (0) jetzt aber sicher unberücksichtigt lassen können, dürfen Sie aus den restlichen neun Elementen (1 bis 9) wieder das niedrigste heraussuchen. Es ist das Element Nummer 8, wie Zeile 2 von Bild 23 zeigt. Dessen Inhalt tauschen Sie nun gegebenenfalls mit dem Inhalt der zweiten Stelle des Feldes, das Ergebnis sehen Sie in Zeile 3. So gehen Sie dann weiter für jede Stelle bis zum Ende des Feldes vor. Das Ergebnis ist eine geordnete Liste wie in Zeile 4.
Die niederwertigste Stelle aus einer Liste zu suchen, ist nicht so einfach, wie es sich anhört. Um diesen Wert an den Beginn der Liste zu stellen, gehen Sie beispielsweise so vor: Sie vergleichen die erste Stelle mit der zweiten. Ist der Wert der zweiten Stelle kleiner, tauschen Sie die Werte. Nun vergleichen Sie den Wert der ersten Stelle mit dem der dritten. Wieder tauschen Sie, sollte dieses Mal die dritte Stelle kleiner sein. So geht das weiter bis zum Ende des Feldes. Logischerweise steht hiernach der niedrigste Wert der Liste zu Anfang. Jetzt beginnen Sie den gleichen Vorgang mit dem zweiten Element, dann mit dem dritten und so weiter (Bild 23). Von der Schleife wissen Sie schon, wie Sie funktioniert. So genügt es sicher, wenn über einem mehrfach auszuführenden Programmteil steht, wo die Schleife beginnt und endet. Außerdem ist die verwendete Zählvariable aufgeführt.

Die erste Schleife bewegt sich vom ersten bis zum vorletzten Element des Feldes. Die zweite Schleife beginnt beim Element nach der Variablen im Feld, auf das der Zähler der ersten Schleife als Index zeigt (»Zähler1«). Diese Schleife läuft bis zum Ende des Feldes. Ist das Element, auf das »Zähler2« zeigt, kleiner als »Zahlen(Zähler1)«, muß der Inhalt der beiden Positionen vertauscht werden, damit in »Zahlen(Zähler1) der kleinere Wert steht. Lief die zweite Schleife komplett durch, enthält »Zahlen(Zähler1)« den kleinsten Wert aus dem Rest des Felds. Hiernach addieren Sie eine 1 zu »Zähler1«. Die zweite Schleife vergleicht und tauscht gegebenenfalls mit allen Elementen des Feldes von Index 2 bis 9 und so weiter.

Schritt	Teilaufgabe
1.	Lege den Wert »0« in die Schublade »Index«
2.	Lies den Wert in der Schublade, auf die der Zeiger »Index« zeigt aus der Kommode »Promille«
3.	Erhöhe den Wert in der Schublade Index um eins
4.	Lies den Wert in der Schublade, auf die der Zeiger »Index« zeigt aus der Kommode »Promille«
5.	...

Tabelle 4. Um auf sämtliche Schubladen hintereinander zuzugreifen, sind diese Schritte notwendig

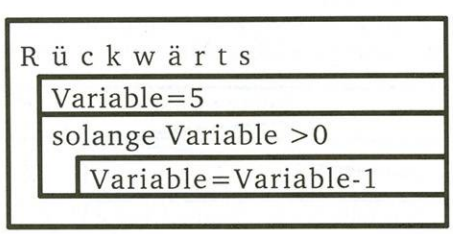


Bild 15. Durch Subtraktion der Zählvariablen läuft die Schleife rückwärts

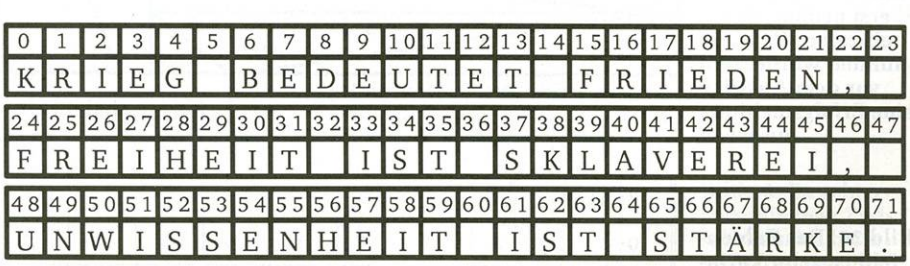


Bild 16. Die einzelnen Buchstaben und Satzzeichen liegen in je einem Element des Datenfelds



Bild 17. Das Feld »Wort« besteht hier aus drei Elementen und hat den Inhalt »IST«

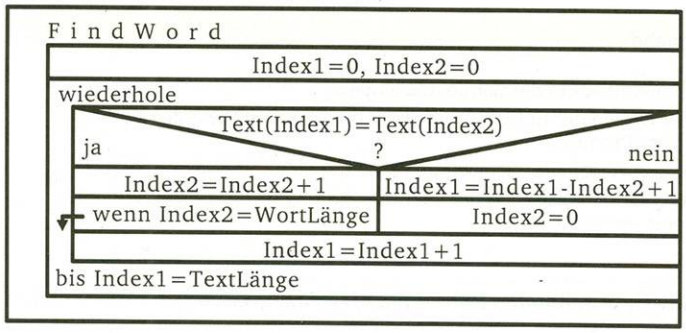


Bild 18. Mit diesem Programm finden Sie ein Wort im Feld »Text«

Sie würden beim Sortieren möglicherweise anders vorgehen. Sie würden wahrscheinlich den kleinsten Wert aus der Liste nehmen und ihn an den Kopf einer zweiten schreiben. In der ersten Liste streichen Sie ihn aus.

Das einfache Austauschen benachbarter Elemente ist nicht die beste Lösung

Dann suchen Sie den kleinsten Wert aus dem Rest der Liste und schreiben den in die zweite Liste genau unter den ersten. Dann streichen Sie auch diesen Wert aus der ersten Zahlenreihe. So geht das weiter, bis keine Werte mehr in der ursprünglichen Liste stehen. Die zweite entspricht dann der ersten Liste, liegt aber sortiert vor.

Dieser Weg ist übrigens auch wesentlich schneller, denn das Tauschen kostet Zeit. Es gibt aber einen wirtschaftlichen Grund dafür, warum man dem Rechner diese Arbeit in der Form des Minsort befiehlt: die Knappheit der Ressourcen. Stellen Sie sich vor, Sie müßten eine Liste »im Kopf« sortieren. Es gelingt Ihnen vielleicht gerade eben, eine Liste mit zehn Zahlen zu behalten. Beim Anlegen der zweiten (sortierten) Liste würden Sie aber bestimmt Probleme bekommen. Sie müßten nochmals zehn Speicher dafür reservieren. Für den Computer spielt das bei einer Liste dieser Größe noch keine Rolle. Meist hat er aber wesentlich größere Datenmengen zu verarbeiten. Überdies müssen Sie hier noch einen ganz anderen Faktor berücksichtigen: Wie streichen Sie eine Stelle durch? Sie können dem Rechner nicht einfach sagen: »Nun, diese Stelle ist jetzt einfach nicht mehr da«. Der Computer ist mit einer bestimmten Menge Speicher ausgerüstet. Diese existiert so lange, bis Sie sie manuell entfernen.

Trotzdem gibt es drei Lösungen hierfür. Sie überprüfen zum Beispiel die Liste und suchen sich einen Wert, der darin sicher nicht vorkommt. Den verwenden Sie dann quasi als Marke: »Immer, wenn Du in der Liste diesen Wert findest, behandle das Element als durchgestrichen!« Bei großen Listen ist das aber nicht so einfach. Sie müßten schon beinahe mit Sortieren beginnen, um einen entsprechenden Wert zu erarbeiten. Üblicherweise verwendet man den Wert 0 oder einen sehr hohen Wert.

Praktischer ist es nun aber, ein drittes Feld anzulegen, das für jeden Eintrag in der Liste 1 den Zustand jeden Elements festhält. Null steht für »gestrichen« und Eins für »vorhanden«. Ist beispielsweise das Element »Zahlen(5)« gestrichen, steht an Stelle 5 in der Liste »Gestrichen« eine 0, ansonsten eine 1. Wichtig ist, daß alle Werte im Feld »Gestrichen« vor dem Sortieren eine 1 beinhalten. Zu diesem Zeitpunkt ist nämlich noch kein Wert aus der Liste »Zahlen« gestrichen worden.

»Neu-Sort« vermeidet das häufige Austauschen von Listenelementen

Nach dem Sortieren stehen alle Stellen auf »0«, denn alle Elemente aus »Zahlen« wurden dann ausgestrichen. Wie gesagt, Speicher ist knapp. Ein drittes Zahlenfeld zu belegen, wäre eine grobe Verschwendung. Vor allem bei größeren Listen ist dieser Weg nicht akzeptabel. Angenommen, »Zahlen(5)« müßte gestrichen werden. Statt dessen könnte man ja »Zahlen(Ende)« auf »Zahlen(5)« schreiben und die Variable »Ende« um den Wert 1 verringern. So hat man die Liste gekürzt, die ursprüngliche fünfte Stelle ist mit der ursprünglich letzten überschrieben (gestrichen) worden.

Das zeitaufwendigste am Minsort ist das häufige Tauschen. Kombinieren Sie ihn mit »menschlichem« Sortieren, erhalten Sie eine schnellere Routine. Sie müßten für jeden Durchlauf der Schleife mit »Zähler1« eigentlich nur höchstens einmal tauschen. Es genügt, festzuhalten, in welchem Feld die kleinste Zahl liegt. Deshalb brauchen Sie eine neue Indexvariable. »kleiner« enthält von nun an den Index für die kleinste Stelle des Listenteils, den »Zähler2« gerade durchläuft. Vor dem Beginn der Schleife mit »Zähler2« gehen Sie davon aus, daß in »Zahlen(Zähler1)« die kleinere Zahl liegt. Deshalb bekommt »kleiner« zunächst den Wert von »Zähler1«. Innerhalb der Schleife wird auf »Zahlen(Zähler2)<Zahlen(kleiner)« geprüft. Ist die Abfrage mit »ja« zu beantworten, erhält »kleiner« den Index auf das kleinere Element. Der steht in »Zähler2«. Endet die zweite Schleife, hält »kleiner« den kleinsten Wert aus dem durchlaufenen Bereich. Es ist davon auszugehen, daß sich »kleiner« geändert hat. Deshalb müssen Sie den Wert »Zahlen(kleiner)« mit »Zahlen(Zähler1)« vertauschen. Sie könnten sehr wohl zuerst prüfen, ob »kleiner« ungleich »Zähler2« ist und nur dann tauschen. Weil das aber meistens zutrifft, ist es weniger zeitaufwendig, generell zu tauschen. Ein Feld mit seinem Inhalt zu tauschen, hat keinen Effekt (Bild 24). Der Kastenplan und später auch das Programm sind jetzt geringfügig länger. Tatsächlich braucht es in der Ausführung aber nur etwas mehr als drei Viertel der Zeit.

Diese Zeitmessungen sind nicht immer aussagekräftig. Manchmal kommt

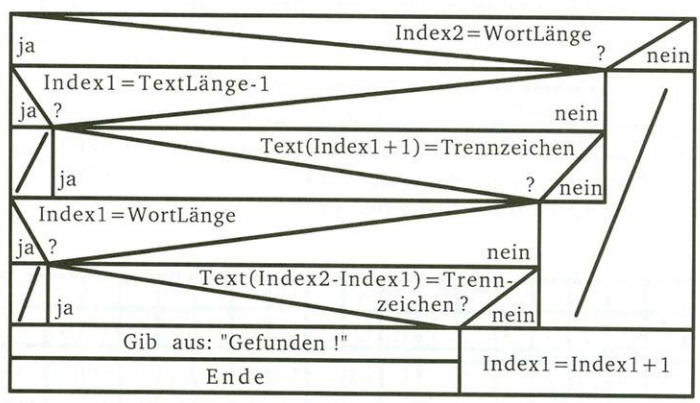


Bild 19. Ist die Textstelle gefunden, muß das Programm erst prüfen, ob ein einzelnes Wort gefunden wurde, und vor dem Ausstieg eine Erfolgsmeldung ausgeben

Zahlen

0	1	2	3	4	5	6	7	8	9
20	68	88	75	81	15	53	50	18	95

Bild 20. Das Zahlenfeld ist zehn Variablen groß und von 0 bis 9 indiziert

es auch auf den Zustand der Daten an. Sind die Daten schon mehr oder weniger sortiert, kann eine sonst sehr schnelle Routine viel Zeit aufwenden, zu erkennen, daß das Ziel erreicht ist. Dasselbe Programm ist bei absolut gemischten Daten im Verhältnis schneller. Sie müssen auch noch andere Probleme berücksichtigen. Verarbeiten Sie Daten von einem Datenträger, müssen Sie vermeiden, abwechselnd vom Anfang und Ende der Datenkette zu lesen oder dorthin zu schreiben. Der Schreib-/Lesekopf, der diese Arbeit übernimmt, müßte sonst jedesmal umpositioniert werden. Das kostet viel Zeit.

»Bubble-Sort« kommt mit vorsortierten Daten bestens zurecht

Hierfür hat man einen Sortiervorgang namens »Bubble-Sort« entwickelt. Damit werden benachbarte Datenelemente verglichen und gegebenenfalls vertauscht. Die Routine arbeitet so lange, bis kein Tausch mehr nötig ist. Auf diese Weise steigen die niedrigeren Werte wie Blasen (englisch »Bubbles«) zu ihrer korrekten Position auf (Bild 25). Sie brauchen eine Variable »Marke«, die kennzeichnet, ob innerhalb der Schleife mit »Zähler« zwei Elemente des Felds »Zahlen« vertauscht werden mußten. Vor dem Start der Schleife wird »Marke« auf 0 herabgesetzt. Für den Fall, daß nicht getauscht werden muß, bleibt dieser Wert unverändert und die Wiederhole-bis-Bedingung ist nicht mehr erfüllt. Die Schleife beginnt beim ersten Element. Ist das Element, auf das »Zähler« zeigt, nicht kleiner als das folgende (»Zahlen(Zähler + 1)«), muß getauscht werden. Das niedrigere Element steigt eine Stufe höher, das höhere eine Stufe herab. So bewegt sich die Schleife bis zum vorletzten Element. Das vergleicht der Programmteil mit dem letzten Element und vertauscht beide, sollte es nötig sein. Hat das Programm innerhalb der Schleife getauscht, besitzt »Marke« den Wert 1. Die Schleife startet neu. Mußte das Programm nicht tauschen, blieb »Marke« 0. Das Feld ist sortiert, das Programm endet.

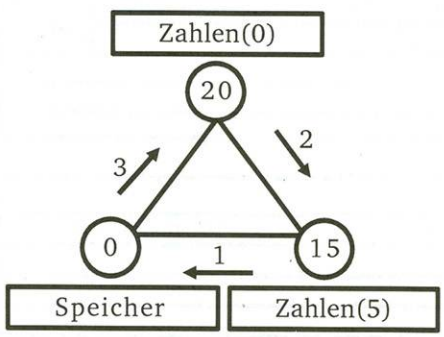


Bild 21. Den Austausch der Inhalte zweier Variablen bewerkstelligt der Dreiecktausch

Bubble-Sort eignet sich auch besonders gut, teilsortierte Felder zu ordnen. Liegt ein Feld in der Wertereihenfolge »7,8,9,4,5,6,0,1,2« vor, nähert sich jeweils die erste falsch positionierte Zahl ihrem richtigen Platz. Richtig gesetzte Zahlen verändern ihren Standort nur, um die »wandernde Zahl vorbeizulassen«. Zum Ende der Schleife stehen sie wieder korrekt (Tabelle 6).

In einer Abwandlung des Bubble-Sort fängt die Schleife nicht immer wieder von vorne an. Statt dessen wechselt der »Shake-Sort« die Richtung. Begann die Schleife zuerst bei 0 und endete bei 8, so geschieht das beim nächsten Durchlauf umgekehrt. Die Schleife beginnt bei 9 und endet bei 1. Beim nächsten Durchgang geht es wieder richtig herum, wie in einem Mixbecher (»to shake«, englisch, zu deutsch »schütteln«) (Bild 26). Eine Spielerei ist das natürlich nicht. Erinnern Sie sich an die Arbeit mit Datenträgern? Sie müssen tunlichst vermeiden, mit dem Schreib-/Lesekopf tatenlos weite Strecken zu überqueren. Mit Bubble-Sort vergleichen und tauschen Sie zwar nur benachbarte Felder, nach jedem Schleifendurchlauf beginnen Sie aber wieder beim Anfang der Datenkette. Der Shake-Sort legt diese Strecke innerhalb einer Sortierrunde zurück.

	0	1	2	3	4	5	6	7	8	9
1	15	68	88	75	81	20	53	50	18	95
2	15	68	88	75	81	20	53	50	18	95
3	15	18	88	75	81	20	53	50	68	95
4	15	18	20	50	53	68	75	81	88	95

Bild 22. So sehen verschiedene Zwischenstufen beim Sortieren mit dem Minsort-Algorithmus aus

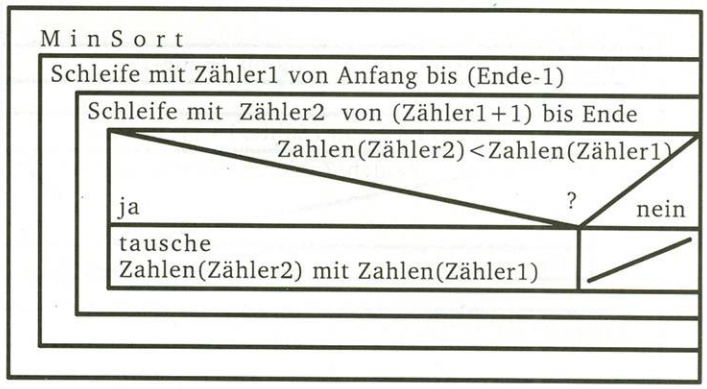


Bild 23. Die einfache Form der Sortierung mit dem klassischen »Minsort«

Schritt	Teilaufgabe
1.	Zähler = 0
2.	Wenn Text (Zähler) nicht den gesuchten Buchstaben enthält, dann erhöhe den Zähler um eins, sonst rufe: »Gefunden!« und beende
3.	Wenn der Zähler kleiner als 72 ist, denn verzweige zu Zeile 2

Tabelle 5. Die Vorgänge beim Suchen eines Buchstabens in einem Text

Der Shake-Sort-Algorithmus nimmt auf die Arbeitsweise eines Computers Rücksicht

Die Schwierigkeit besteht darin, daß Anfangs-, Endwert und Schrittweite sich für jeden Durchgang ändern. Zuerst arbeitet die Schleife von Element 0 bis 8, danach aber von 9 bis 1. Der Zähler wird im ersten Fall um den Wert 1 erhöht, beim zweiten Durchlauf aber um 1 verringert. Läuft die Schleife rückwärts durch das Feld, müssen Sie außerdem das Feld »Zahlen(Zähler-1)« mit »Zahlen(Zähler)« vergleichen. In diesem Fall tauschen Sie, wenn »Zahlen(Zähler)« kleiner ist. Man ist also beinahe geneigt, für jede Richtung einen eigenen Programmteil zu verfassen. So wäre es zwar am schnellsten, der Beschreibungsaufwand aber sehr hoch.

Zuerst brauchen Sie eine Marke, die darstellt, in welcher Richtung das Feld gerade abgearbeitet wird. Mit ein und derselben Anweisung könnten Sie beispielsweise einer Marke abwechselnd den Wert 1 oder 0 geben.

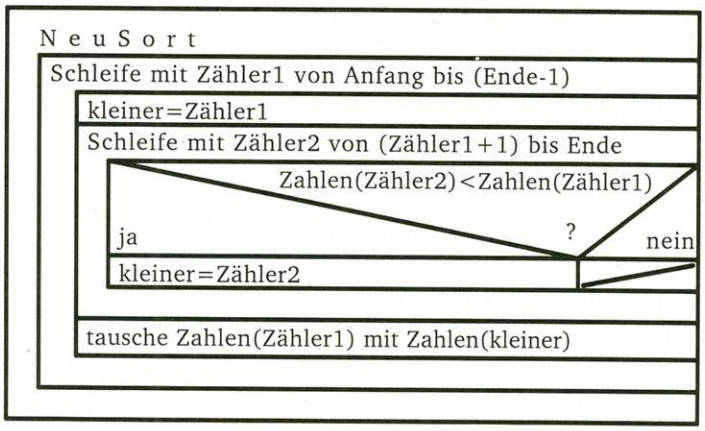


Bild 24. »NeuSort« tauscht nur noch, wenn das höchste Element des Listenrests feststeht

Gehen Sie davon aus, daß die Variable »Richtung« den Wert 1 hat. Nach der Anweisung

```
Richtung=1-Richtung
```

hat »Richtung« den Wert 0. Führen Sie diese Rechnung noch einmal durch (jetzt mit »Richtung« gleich 0), wird »Richtung« wieder zu Eins. Also jedesmal, wenn diese Anweisung abgearbeitet wird, ändert die Variable ihren Wert von 1 auf 0 oder umgekehrt. Jetzt brauchen Sie nur noch drei Felder mit je zwei Elementen. Wenn die Marke den Wert 0 hat, beginnt die Schleife bei 0, sonst bei 9. Die Schleife endet entsprechend bei 8 beziehungsweise bei 1. Für den ersten Durchgang müßte der Befehl lauten:

```
Schleife mit Zähler von 0 bis 8
```

Der zweite Durchgang beginnt mit

```
Schleife mit Zähler von 9 bis 1
```

Nun verwenden Sie statt »Anfang« und »Ende« je ein Datenfeld. »Anfang(0)« hat den Wert 0, »Anfang(1)« 9. Entsprechend besitzt »Ende(0)« den Wert 8, und »Ende(1)« ist 1. So lautet der Befehl generell

```
Schleife mit Zähler von Anfang(Richtung)
bis Ende(Richtung)
```

Je nachdem, welchen Wert »Richtung« hat, läuft die Schleife vorwärts oder rückwärts. Wichtig ist jetzt noch die Schrittweite. Der Zähler muß für die vorwärtsablaufende Schleife um 1 erhöht werden. Rückwärts subtrahieren Sie vom Zähler 1 oder addieren den negativen Wert -1. Verwenden Sie ein Feld »Schritt« mit den Werten

```
Schritt(0)=1
Schritt(1)=-1
```

würde die Schleife im ausführlichen Satz lauten:

```
Schleife mit Zähler von Anfang(Richtung)
bis Ende(Richtung), Schrittweite:
Schritt(Richtung)
```

Das Feld Schritt findet noch zwei weitere Verwendungen. Wie bereits erklärt, muß die Abfrage im ersten Durchlauf lauten:

```
Zahlen(Zähler+1)<Zahlen(Zähler) ?
```

Der zweite Durchgang läuft rückwärts. Dementsprechend muß »Zahlen(Zähler-1)« mit »Zahlen(Zähler)« verglichen werden. Theoretisch könnte man schreiben:

```
Zahlen(Zähler+Schritt(Richtung))<Zahlen(Zähler) ?
```

»Schritt(Richtung)« besitzt den Wert 1 bei vorwärts ablaufender Schleife und -1, wenn die Schleife rückwärts läuft. Entsprechend bezieht sich der Vergleich auf das Element nach beziehungsweise vor »Zahlen(Zähler)«. Wenn die Schleife rückwärts läuft, dürfen Sie nur tauschen, wenn das Element »Zahlen(Zähler-1)« größer ist als »Zahlen(Zähler)«. Der Vergleich prüft aber auf »kleiner«, denn der Befehl bezieht sich ja eigentlich auf einen vorwärts ablaufenden Durchgang. Nach dem ersten Durchgang sieht das einmal durchlaufene Feld wie in Bild 27 aus.

Wenn die Schleife jetzt rückwärts von 9 bis 1 läuft, prüft die Schleife zunächst ob »Zahlen(8)« kleiner ist als »Zahlen(9)«. Und weil 88 kleiner ist als 95, tauscht das Programm, was ein grober Fehler wäre. Würde man die beiden Zahlen in negative Zahlen umwandeln, dann würde der Vergleich das Ergebnis »NEIN« erbringen, denn -88 ist nicht kleiner als -95. Eine Zahl wandelt man in eine negative, indem man sie mit -1 multipliziert. So müßte die korrekte Abfrage lauten:

```
(Schritt(Rich-
tung)*Zahlen(Zähler+Schritt(Rich-
tung))) < (Schritt(Rich-
tung)*Zahlen(Zähler)) ?
```

Für den Fall, daß die Schleife gerade rückwärts läuft, besitzt »Schritt(Richtung)« den Wert -1. Damit sind die zwei Vergleichsargumente kleiner als 0, und der Vergleich »kleiner als« ergibt das für den Fortlauf richtige (eigentlich falsche) Ergebnis. Das Programm führt noch zwei neue Variablen ein. »Erster« ist der Index auf das erste Element im Feld Zahlen, also 0. »Letzter« zeigt auf das letzte Element (9). Das sind die Werte, die zuvor in den

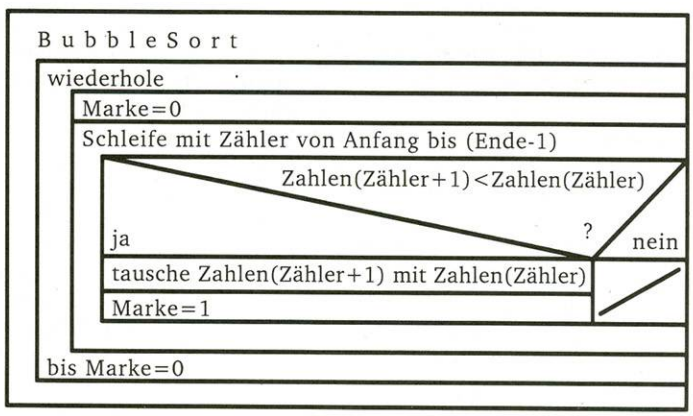


Bild 25. »BubbleSort« sortiert ein Zahlenfeld, indem das Programm benachbarte Felder solange vertauscht, wie es nötig ist

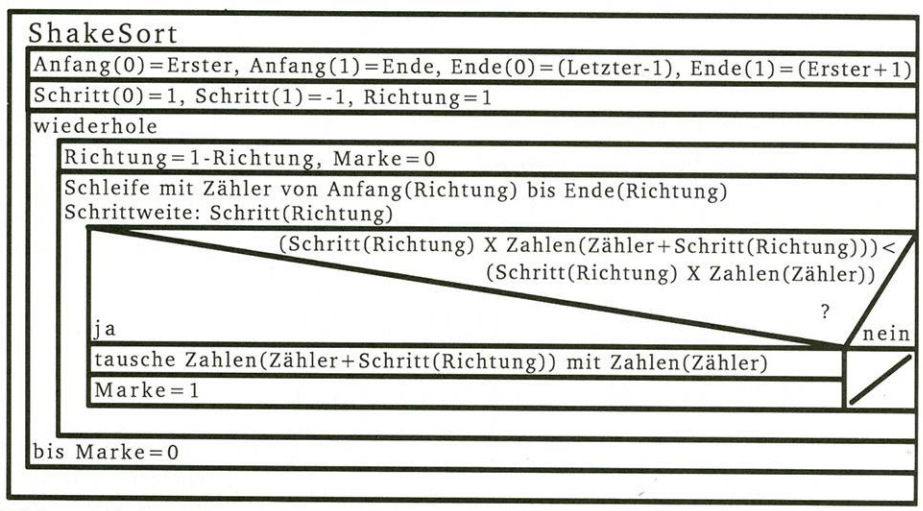


Bild 26. »ShakeSort« sortiert ein Zahlenfeld wie »Bubble-Sort«, ändert aber nach jedem kompletten Schleifendurchlauf die Richtung

Variablen »Anfang« und »Ende« steckten, die ja nun Felder sind. Die jeweiligen Endwerte sind bereits korrigiert, das heißt um den Wert 1 verringert beziehungsweise erhöht.

Computer verstehen nur künstliche Sprachen

Diese Einführung soll Ihnen helfen, die »Denkweise« von Computern zu verstehen. Diese ist unabhängig von konkreten Computern oder bestimmten Programmiersprachen.
Egal in welcher Sprache Sie Ihre Anwendungen formulieren, letztendlich entsteht daraus immer Maschinencode. Das ist nichts anderes als reine Zahlen, die im Arbeitsspeicher Ihres Computers stehen, und die der Prozessor, das Gehirn der Maschine, versteht – ganz im Gegensatz zu den allermeisten Programmierern. Die einzelnen Anweisungen gehen nicht weit über einfache Lade- und Speicheroperationen, Unterprogrammaufrufe und bedingte Sprünge hinaus. Um die Arbeit damit zu erleichtern, erfand man die Assemblersprache, die anstelle der Maschinencodes wenige Zeichen lange Kürzel akzeptiert. Deshalb wurden

Durchgang	Reihenfolge der Zahlen									
0	1	2	3	4	5	6	7	8	9	
1	7	8	4	5	6	0	1	2	3	9
2	7	4	5	6	0	1	2	3	8	9
3	4	5	6	0	1	2	3	7	8	9
4	4	5	0	1	2	3	6	7	8	9
5	4	0	1	2	3	5	6	7	8	9
6	0	1	2	3	4	5	6	7	8	9

Tabelle 6. Die erste Spalte bezeichnet den Schleifendurchgang, und die restlichen zehn sind Indizes auf Elemente ein einem Feld, mit dem Inhalt »7,8,9,4,5,6,0,1,2«

0	1	2	3	4	5	6	7	8	9
20	68	75	81	15	53	50	18	88	95

Bild 27. So sieht das unsortierte Datenfeld aus Bild 20 nach dem ersten Schleifendurchlauf von »ShakeSort« aus

Hochsprachen wie Basic, Pascal oder C entwickelt, die einfache Befehle, beispielsweise zur Ausgabe eines Textes auf dem Bildschirm, enthalten.
Letztendlich sind diese Sprachen nur Sammlungen von komplexen Maschinencode-Routinen, die zur leichteren Handhabung einprägsame Namen haben und in eine komfortable Benutzeroberfläche, die Programmiersprache als solche, eingebunden sind.
Mit dem Rüstzeug, daß Sie sich beim Durchlesen dieses Artikels angeeignet haben, sind Sie in der Lage, eine Aufgabenstellung zu analysieren und computergerechte Ablaufpläne zu generieren. Ob Sie Ihre Kastenpläne anschließend in einer höheren Programmiersprache wie Basic oder Pascal oder auf niedrigerer Ebene, aber mit größerer Ablaufgeschwindigkeit in Assembler umsetzen, ist davon unabhängig. Und denken Sie daran: Programmieren lernen Sie am besten, indem Sie es tun.

(Dieter Bäumler/wn)

DOS TASCHENBÜCHER

Ihre PC-Informationen im Westentaschenformat - platzsparend und immer griffbereit!

Band 1:
Von Assembler bis MS-DOS

Die Neuauflage des legendären, seit langem vergriffenen Sonderheftes DOS EXTRA 1: Die wichtigsten Interna des PC als Beiträge, Tabellen und Listings. Wissen, das jeder PC-Besitzer haben muß!

Band 2:
Word 3.0 und 4.0

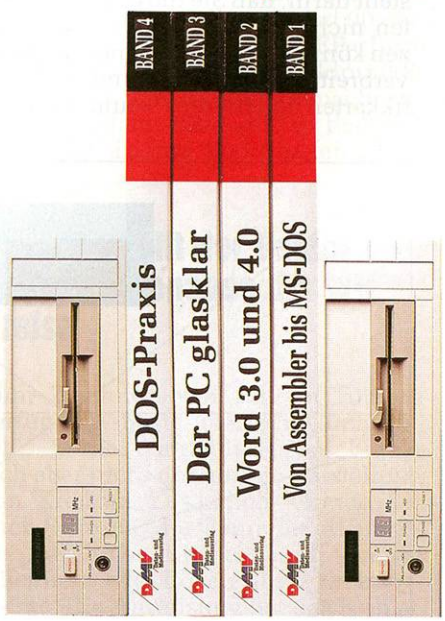
Mit Tips, Tricks und gezieltem Know-how über das Textverarbeitungsprogramm Word sparen Sie viel Zeit und Ärger. Sie lernen den Umgang mit Druckformatvorlagen, Textbausteinen, Makros etc. Ein Muß für jeden Word-Anwender.

Band 3:
Der PC glasklar

Sie erhalten einen Einblick in das geheimnisvolle Innere der PC-Hardware und die Funktionen des PC-Betriebssystems. Wichtige Begriffe wie BIOS, Interrupt und EMS-Speicher werden leicht verständlich erklärt. Das Buch für den interessierten PC-Einsteiger.

Band 4:
DOS-Praxis

Wenn es um die Modifikation der »Config.sys«, das Schreiben von Batch-Dateien, Datensicherung bei Festplatten, Gerätetreiber oder die Beherrschung der DOS-Kommandos geht – dann ist dieses Taschenbuch genau das Richtige für Sie!



PC-Wissen kompakt: DOS-Taschenbücher

DOS-Taschenbücher
Alle Bände ca. 270 Seiten, mit vielen Abbildungen, Hardcover broschiert — jeweils DM **29,80***

DATABOX
Alle Listings, Beispiele und Programme zu den DOS-Taschenbüchern Band 1, 3 und 4 erhalten Sie auch auf Datenträger zum günstigen DATABOX-Preis — jeweils DM **24,–***

* Unabhängig von der Anzahl der bestellten Produkte berechnen wir für das Inland DM 4,– bzw. für das Ausland DM 6,– Porto und Verpackung. Bitte benutzen Sie die Bestellkarte.

DMV-Verlag · Postfach 250 · 3440 Eschwege

DMV Daten- und Medienverlag

Basic-Werkstatt

Grafik mit GW-Basic

Hinter der Abkürzung GW von GW-Basic verbergen sich die beiden Begriffe Grafik und Window. GW-Basic hält was es verspricht. Mit vielen komfortablen Befehlen unterstützt es sowohl die Programmierung farbiger Grafiken als auch die Verwaltung von Fenstern. Erfahren Sie in dieser Werkstatt, wie Sie diese Fähigkeiten des Interpreters nutzen.

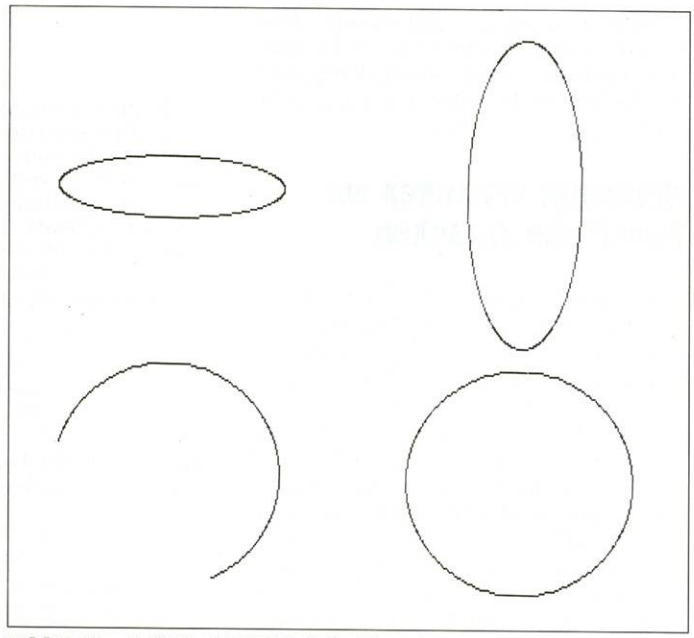


Bild 2. Der Befehl »CIRCLE« läßt die Erzeugung von unterschiedlichen Kreisformen zu

Ein großer Nachteil der mit den ersten Versionen von MS-DOS zusätzlich gelieferten Programmiersprache Basic war die fehlende Grafikfähigkeit. Seit GW-Basic gehört dies der Vergangenheit an. Selbstverständlich kann man diese Sprache mit den Weiterentwicklungen Super Basic oder Quick Basic vergleichen; sie bietet aber gerade dem Einsteiger die Gelegenheit, sich ohne zusätzlichen Kostenaufwand mit der Programmierung eines Computers vertraut zu machen.

Die damit angefertigten Programme benötigen allerdings zur Ausführung immer GW-Basic, was jedoch kein großes Problem darstellt, weil fast jedem Computerbesitzer die Sprache zur Verfügung steht. Der einzige Nachteil besteht darin, daß Sie die Grafikfähigkeiten nicht mit der Hercules-Karte nutzen können. Dank der immer stärkeren Verbreitung von EGA- und VGA-Grafikkarten dürfte dies kaum mehr ins Gewicht fallen. Lernen Sie also mit dieser Einsteigerwerkstatt die Fähigkeiten von GW-Basic gerade im Grafikbereich kennen.

Der Cls-Befehl dient zum Löschen des Bildschirms. Im Textmodus löscht er den Bildschirm und setzt den Cursor in die oberste linke Ecke. Im Grafikmodus werden der Bildschirm gelöscht und zusätzlich die aktuelle Hintergrundfarbe eingestellt. Außerdem setzt der Befehl den Ausgangspunkt für Grafikanweisungen, beispielsweise »DRAW«, in die Bildschirmmitte. Die Koordinaten der Mitte sind von der jeweiligen Grafikkarte und dem gewählten Modus abhängig.

Um GW-Basic mitzuteilen, daß Sie im Grafikmodus arbeiten wollen, benutzen Sie den Befehl »SCREEN«, über den Sie zusätzlich die Auflösung des Bildschirms angeben. Die maximale Auflösung ist dabei normalerweise 640x200 Bildpunkte bei zwei Farben (schwarz/weiß). Einige GW-Basic-Versionen unterstützen auch die höheren Auflösungen der EGA-Karten mit zusätzlichen Farben (Tabelle 1). Um beispielsweise auf die EGA-Auflösung zu schalten, geben Sie den Befehl

```
SCREEN 9
```

ein. Dazu benötigen Sie natürlich ebenfalls eine EGA- oder VGA-Karte. Es lassen sich aber auch noch weitere Parameter übergeben.

```
SCREEN <Modus>,<Farbzustand>,<aktive Seite>,<sichtbare Seite>
```

Mit »Farbzustand« kann man laut Handbuch die Farbdarstellung mit »0« beziehungsweise »1« aktivieren oder deaktivieren, sofern die gegebene Konfiguration überhaupt Farbdarstellung unterstützt. In der Praxis kann jedoch nicht bei jeder VGA-Karte eine Änderung festgestellt werden. Die Ausgabe erfolgt dann immer in Farbe.

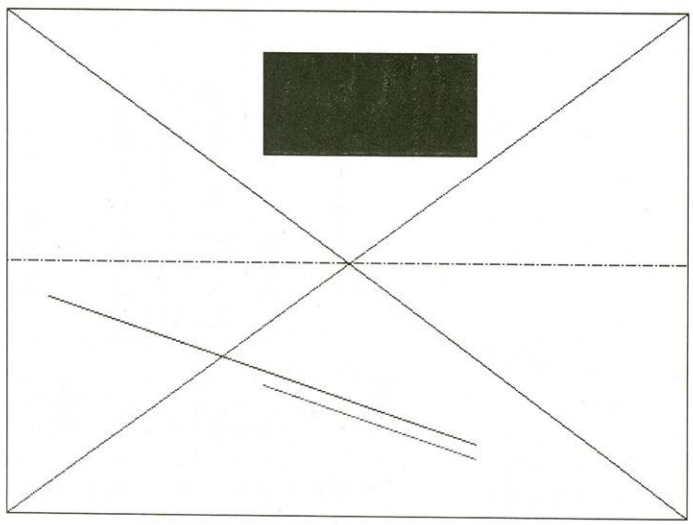


Bild 1. Die Bildschirmausgabe des Programms »line.bas« zeigt die Auswirkung der unterschiedlichen Parameter beim Line-Befehl

Auf EGA- oder VGA-Karten haben Sie bis zu acht Grafikseiten

Die beiden folgenden Parameter erlauben das Umschalten der Bildschirmseiten im Grafikmodus. Das Programm »screen.bas« verdeutlicht die Anwendung (Listing 1).

Betrachten wir nun die Funktion der einzelnen Zeilen. Zeile 10 löscht den aktuellen Bildschirm. Danach wird mit »SCREEN 0,0,0,0« die Seite 0 aktiviert und auch dargestellt, so daß die Ausgabe der Print-Anweisung in der nächsten Zeile auf dieser Seite erfolgt. Die Zeilen 40 und 50 erfüllen die gleiche Aufgabe, nur erscheint dabei der Text auf Bildschirmseite 1. In Zeile 60 schalten Sie wieder auf Seite 0 um,

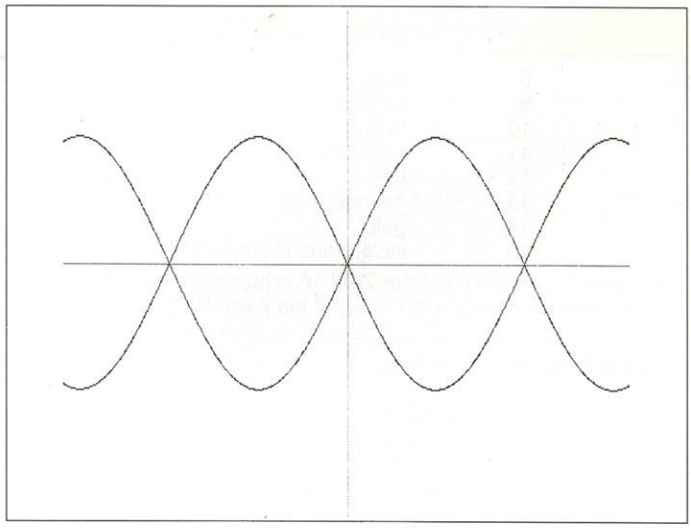


Bild 3. Durch »WINDOW« läßt sich der Bildschirm neu skalieren, so daß aufwendige Umrechnungen entfallen können

wodurch der Text »Bildschirm 0« sichtbar wird. Nach einem Tastendruck wird wiederum die Seite 1 angezeigt und zusätzlich aktiviert, damit der Input-Befehl seine Eingabeaufforderung auch auf dieser Seite ausgibt. Nach dem Drücken einer beliebigen Taste schaltet das Programm wieder auf die Grundeinstellung zurück.

Als praktische Anwendung wäre beispielsweise an eine Hilfefunktion zu denken: Während auf der ersten Bildschirmseite das Programm läuft, kann per Tastendruck auf der zweiten Seite unabhängig davon jederzeit eine passende Hilfestellung gegeben werden. Oder Sie lassen den Aufbau einer komplexen Grafikdarstellung im Hintergrund ablaufen und schalten erst um, wenn die Grafik komplett ist.

Den Befehl »SCREEN« verwenden Sie auch als Funktion, er liefert dann einen Wert zurück und hat keine spezifische Wirkung. Wird die Funktion im Textmodus angewendet, läßt sich mit

```
A=SCREEN(<Zeile>,<Spalte>),
<Schalter>
```

der Variablen »A« der ASCII-Code des jeweiligen Zeichens zuweisen. Dazu muß <Schalter> den Wert 0 besitzen oder gänzlich weggelassen werden. Hat <Schalter> einen Wert ungleich 0 (»True« entspricht »wahr«), enthält »A« anstelle des Codes anschließend das Farbattribut des betreffenden Zeichens. Die Rechnung

```
(Attribut MOD 16)
```

gewinnt aus dem Farbattribut die Vordergrundfarbe,

```
(((Attribut-Vordergrundfarbe)/16) MOD
128)
```

ergibt die Hintergrundfarbe. Ist das Farbattribut größer als 127, dann handelt es sich um ein blinkendes Zeichen.

Mit dem Befehl »COLOR« lassen sich die Bildschirmfarben individuell einstellen. Im Textmodus können Sie aus einer Farbpalette mit 16 unterschiedlichen Tönen auswählen (Tabelle 2). Damit eine hellgrüne Schrift auf rotem Hintergrund mit einem gelben Bildschirmrand erscheint, müssen Sie die Zeile

```
COLOR 10,4,14
```

eingeben. Die Reihenfolge der Parameter ist dabei »Vordergrund«, »Hintergrund« und »Rahmen«. Im CGA-Modus bei 320x200 Bildpunkten mit vier Farben hat der Color-Befehl eine andere Wirkung: Sie können eine von zwei vordefinierten Farbpaletten auswählen (Tabelle 3). Andere Farbzusammenstellungen sind mit einer CGA-Karte nicht realisierbar.

Der Line-Befehl zeichnet beliebig gestrichelte Linien

Der Line-Befehl dient zum Zeichnen von Linien. Er kann aber auch Rechtecke zeichnen und sie sogar ausfüllen. Die praktische Anwendung demonstriert das kurze Programm »line.bas« (Listing 2). Bild 1 zeigt die Bildschirmausgabe, die das Programm erzeugt. Durch die Eingabe von »SCREEN 9« schalten Sie den EGA-Modus mit einer maximalen Auflösung von 640x350 Bildpunkten ein. Da die obere linke Ecke des Bildschirms die Koordinaten »0,0« hat, ergeben sich zwangsläufig für die rechte untere Ecke die Werte »639,349«. Es werden also immer zuerst der horizontale Wert (x-Richtung) und dann der vertikale Wert (y-Richtung) angegeben. Besitzen Sie eine Grafikkarte mit geringerer Auflösung, so müssen Sie diese Daten für das Beispiel anpassen.

Modus	Auflösung	Farben
0	Text	
1	320 x 200	4 Farben aus Palette
2	640 x 200	2 Farben (schwarz/weiß)
7	320 x 200	16 Farben
8	640 x 200	16 Farben
9	640 x 350	16 Farben

Tabelle 1. Übersicht der Grafikmodi und Auflösungen

Die zusätzlichen Angaben, die Sie beim Befehl »LINE« machen können, sind sehr umfangreich und lauten in voller Länge:

```
LINE(<x1>,<y1>)-<STEP>(<x2>,<-
y2>),<Farbe>,<BF>,<Muster>
```

Den Bildschirmrahmen erzeugen Sie, wie in Zeile 40 demonstriert. Durch den Parameter »B« (»Block«) wird statt einer einfachen Linie ein Rechteck gezeichnet. Gibt man, wie in Zeile 100, noch zusätzlich ein »F« nach dem »B« an, wird das Rechteck in der angegebenen Farbe ausgefüllt. In den Zeilen 50 und 60 sehen Sie die Auswirkung der unterschiedlichen Koordinatenangaben. Um verschieden gemusterte Linien zu zeichnen, müssen Sie erst das gewünschte Muster entwerfen, welches aus 16 Punkten (Bits) besteht. Diese 16 Punkte wiederholt der Line-Befehl, bis er den Endpunkt der Linie erreicht hat. Mit Hilfe von Tabelle 4 können Sie den Zahlenwert für ein bestimmtes Bitmuster errechnen. Für alle gesetzten Bits addieren Sie die in der Tabelle nebenstehenden Zahlen und wandeln das Ergebnis in eine Hexadezimalzahl um. Mit dem Befehl

```
PRINT HEX$(<Zahl>)
```

führt Basic im Direktmodus für Sie diese Aufgabe aus. Ein Beispiel dafür ist Zeile 70. Das Ergebnis sehen Sie an der waagerechten Linie, die über den ganzen Bildschirm läuft.

Soll die Angabe des Endpunktes einer Linie nicht auf den Bildschirm, sondern auf den Anfangspunkt bezogen werden, so ist dem zweiten Wertepaar der Parameter »STEP« voranzustellen. Die unterschiedlichen Auswirkungen sehen Sie in den Zeilen 80 und 90 des Beispielprogramms. Bei »STEP« liegt der Endpunkt somit 400 Bildpunkte nach rechts und 100 Punkte nach unten verschoben, ohne »STEP« liegt der Endpunkt an den Koordinaten »400, 100«.

Ellipsen stellt der Befehl »CIRCLE« genauso dar wie Kreise

Zum Zeichnen von runden Formen verfügt GW-Basic über den Circle-Befehl. Seine Anwendung beschränkt sich aber nicht nur auf die Darstellung von Kreisen. Zusätzlich können Sie auch Ellipsen, die ja nur der allgemeine Fall eines Kreises sind, ebenso zeichnen wie Bögen und Kreissegmente. Die Anweisung hat das Format

```
CIRCLE(<x>,<y>),<r>,<Farbe>,
<Anfang>,<Ende>,<Verzerrung>
```

Das kleine Beispiel »circle.bas« (Listing 3) mit vier verschiedenen Aufrufen zeigt die Auswirkung der einzelnen Parameter (Bild 2). Beachten Sie dabei wieder den Grafikmodus, den Sie unter Umständen an Ihre Konfiguration anpassen müssen.
Zeile 40 zeichnet einen normalen Kreis. Dieser sollte auf Ihrem Bildschirm vollkommen rund erscheinen. Ist dies nicht der Fall, müssen Sie die Bildhöhe oder -breite am Monitor korrigieren. Daneben sehen Sie ein Kreissegment. Der Anfangs- und Endwinkel kann optional im Bogenmaß angegeben werden. Die weiteren Zeilen stellen Ellipsen dar, die durch unterschiedliche Verzerrungsparameter erzeugt werden. Bei Werten größer 1 erfolgt die Ausdehnung in vertikaler, bei Werten kleiner 1 in horizontaler Richtung.
Mit der Anweisung »PSET« setzen Sie einen einzelnen Punkt an eine beliebige Stelle auf dem Grafikbildschirm. Beim Aufruf mit

```
PSET (<x>,<y>),<Farbe>
```

gibt »x« die waagerechte und »y« die senkrechte Position an. Mit »Farbe« können Sie je nach Grafikmodus aus unterschiedlich vielen Farben auswählen. Ergänzend dazu ist der Preset-Befehl verfügbar, der mit »PSET« identisch ist, solange Sie eine Farbe angeben. Geben Sie keinen Parameter an, oder hat er den Wert 0 (entspricht der Hintergrundfarbe), dann wird der Punkt an der entsprechenden Stelle gelöscht.

Parameter	Farbe	Parameter	Farbe
0	schwarz	8	grau
1	blau	9	hellblau
2	grün	10	hellgrün
3	cyan	11	hellcyan
4	rot	12	hellrot
5	magenta	13	hellmagenta
6	braun	14	gelb
7	weiß	15	weiß, hohe Helligkeit

Durch Addieren des jeweiligen Parameters mit der Zahl 16 erhalten Sie eine blinkende Darstellung. Als Hintergrund können Sie nur die Farben 0 bis 7 wählen.

Bei Monochrom-Adaptern gilt folgende Zuordnung:

Vordergrund	0	schwarz
	1	unterstrichen
	2 bis 7	weiß
	8 bis 15	weiß, hohe Helligkeit
	16 bis 31	entsprechend, jedoch blinkend
Hintergrund	0 bis 6	schwarz
	7	weiß

Schwarz als Vordergrund ist nur darstellbar, wenn Sie zugleich als Hintergrund 7 gewählt haben. Bei Hintergrund 0 sind die Zeichen unsichtbar.

Tabelle 2. Die unterstützten Farbeinstellungen mit Parameter und entsprechender Farbe

»POINT« ermittelt die Farbe jedes einzelnen Bildschirmpunktes

Die Farbe eines Punktes lesen Sie mit dem Aufruf

```
A=POINT(<x>,<y>)
```

aus, sie wird der Variablen »A« zugewiesen. Der Wert kann dann mit

```
PRINT A
```

angezeigt werden. Er läßt sich anhand der Farbtabelle (Tabelle 2) der jeweiligen Farbe zuordnen. Liegt der Punkt außerhalb des angezeigten Bildschirmbereichs, beträgt der Wert –1.
Der Befehl »WINDOW« führt oft zu irrtümlichen Deutungen. Sie können damit nicht, wie es das Wort vermuten läßt, ein Fenster aktivieren. Die Anweisung dient vielmehr der Neufestlegung der Bildschirm-Koordinaten, was beispielsweise bei der grafischen Ausgabe mathematischer Funktionen notwendig ist. Den Window-Befehl rufen Sie mit den folgenden Parametern

```
WINDOW <SCREEN> (<x1>,<y1>)-
(<x2>,<y2>)
```

auf. Das Beispiel »window.bas« (Listing 4) einer Sinusfunktion demonstriert die Anwendung. Da der Maximalwert der Sinusfunktion 1 ist, wäre die Kurve normalerweise nur zwei Bildschirmpunkte hoch. Durch die Zeile 50 im Programm »window.bas« werden die Werte der Bildschirm-Eckpunkte neu gesetzt. Die linke obere Ecke hat nun den Wert »6,2«, die linke untere den Wert »– 6, – 2«. Da die Sinusfunktion von »– 5« bis »+ 5« in Schritten von »0,01« berechnet wird, bleibt links und rechts jeweils eine Bildschirmeinheit frei. Auf Grund der Höchstwerte ± 1 füllt die Kurve nur den halben Bildschirm. Durch die zusätzliche Angabe »SCREEN« werden die Eckpunkte getauscht. Das Paar »x1,y1« benennt nun die linke untere, das Paar »x2,y2« die rechte obere Ecke des Bildschirms. Die danach berechnete und gezeichnete Sinuskurve steht deshalb auf dem Kopf (Bild 3).

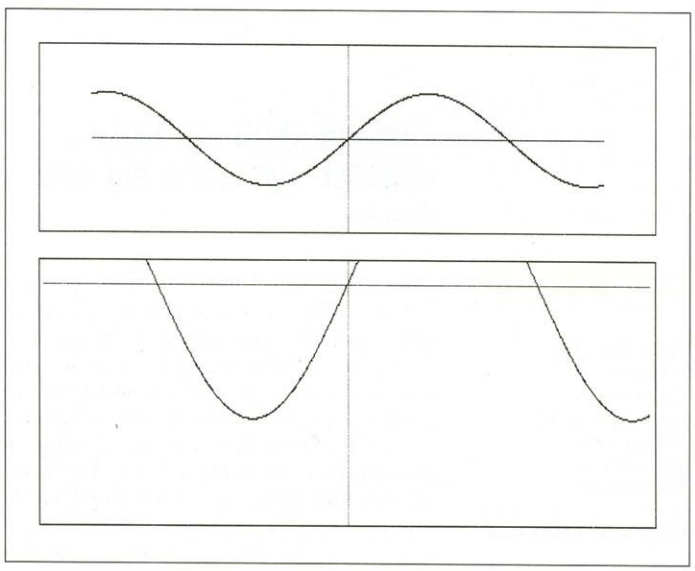


Bild 4. Mit »VIEW« lassen sich auch unter GW-Basic im Grafikmodus »Fenster« verwirklichen

»VIEW« legt einen bestimmten Ausschnitt als »Fenster« fest

Die Anweisung »VIEW« erlaubt Ihnen, einen bestimmten Ausschnitt des Bildschirms im Grafikmodus als »Fenster« zu definieren. Diesem können Sie mit dem bereits bekannten Befehl »WINDOW« zusätzlich neue Eckkoordinaten zuweisen. Die richtige Syntax dazu lautet:

```
VIEW <SCREEN> (<x1>,<y1>)-(<x2>,
<y2>),<Farbe>,<Rahmen>
```

Wie Sie aus dem Beispiel im Programm »view.bas« (Listing 5) ersehen, fehlt in der ersten View-Anweisung der Parameter »SCREEN«. Dadurch bezieht sich die nachfolgende Skalierung nur auf den mit »VIEW« festgeleg-

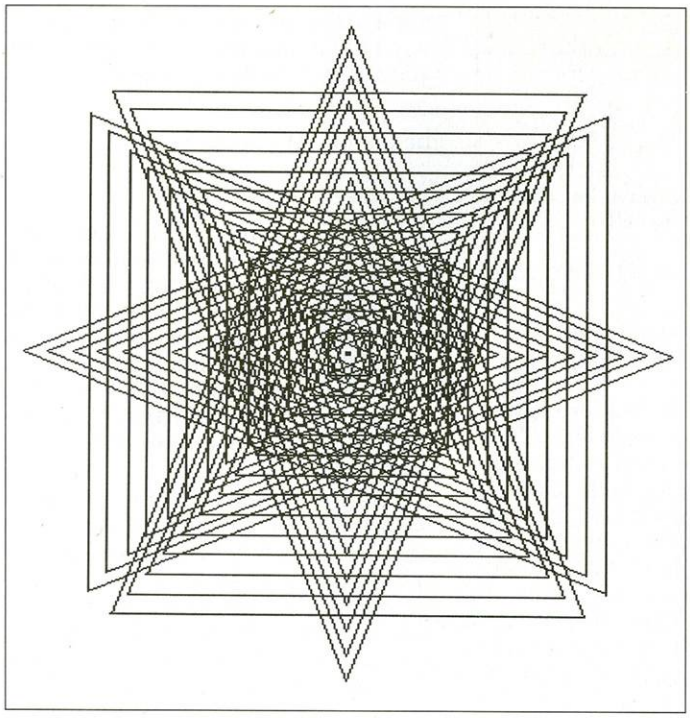


Bild 5. Diese Ausgabe ist mit dem Befehl »DRAW« sehr einfach zu realisieren. Eine einzige Draw-Anweisung, die mit unterschiedlichen Parametern aufgerufen wird, reicht aus.

ten Bereich. Die Sinusfunktion paßt also genau in das Bildschirmfenster (Bild 4). Beim zweiten View-Befehl wurde zusätzlich »SCREEN« angegeben. Damit gilt die folgende Window-Anweisung für den gesamten Bildschirm. Das Fenster hat aus diesem Grund den Effekt eines »Gucklochs«: Die Sinusfunktion zieht sich über den ganzen Bildschirm, jedoch nur der mit »VIEW« gekennzeichnete Bereich ist sichtbar.

Die Form des Cursors ist für Basic-Programmierer nicht tabu

Mit »LOCATE« können Sie den Cursor auf eine bestimmte Position setzen, unabhängig davon, ob Sie gerade im Text- oder im Grafikmodus arbeiten. Mit zusätzlichen Angaben schalten Sie den Cursor ein oder aus und ändern seine Form. Dazu muß die Befehlszeile folgenden Aufbau haben:

```
LOCATE <Zeile>,<Spalte>,<Cursor>,
<Anfang>,<Ende>
```

»Zeile« ist ein numerischer Wert von 1 bis 25, der die Zeile angibt, in welche der Cursor gesetzt werden soll. »Spalte« gibt die Position innerhalb der Zeile an. Normalerweise sind Werte im Bereich zwischen 1 und 80 zulässig. Im CGA-Modus bei 320x200 Bildpunkten ist maximal 40 erlaubt. Die Grafikkarte stellt in diesem Modus auch nur 40 Zeichen pro Zeile dar. Wenn der Parameter »Cursor« 0 ist, schaltet GW-Basic den Cursor ab, mit dem Wert 1 wird er wieder sichtbar. In der Praxis funktioniert dies nicht mit allen GW-Basic-Versionen.

Die beiden Werte »Anfang« und »Ende« bestimmen das Aussehen des Cursors. Der Cursor besteht im Regelfall aus 14 untereinanderliegenden Strichen. Mit »Anfang« geben Sie an, welcher Strich, von oben aus gezählt, als erster dargestellt werden soll. »Ende« gibt entsprechend den letzten Strich an, der noch angezeigt wird. Normalerweise sind dabei Zahlenwerte zwischen 0 und 13 erlaubt. Mit

```
LOCATE ,,,0,13
```

erzeugen Sie einen Cursor in voller Zeichenhöhe. Im Grafikmodus können Sie die Cursorgröße mit dem Locate-Befehl nicht ändern.

Farbcode	Palette 0	Palette 1
0	schwarz	schwarz
1	grün	cyan
2	rot	magenta
3	braun	weiß

Tabelle 3. In der Auflösung 320x200 Bildpunkte können Sie im CGA-Modus nur Farben aus einer vorgegebenen Palette wählen

Bit	Wert
0	1
1	2
2	4
3	8
4	16
5	32
6	64
7	128
8	256
9	512
10	1024
11	2048
12	4096
13	8192
14	16384
15	32768

Tabelle 4. Mit Hilfe dieser Tabelle lassen sich die Linienmuster berechnen

Der komplizierteste und zugleich sehr leistungsfähige Grafikbefehl ist »DRAW«. Diese Anweisung erlaubt Ihnen, auch umfangreiche und komplexe Objekte darzustellen. Der Aufruf hat die Form

```
DRAW <String-Ausdruck>
```

Sie können sich die Funktion von »DRAW» so vorstellen, als würden Sie einen Zeichenstift über den Bildschirm führen. Je nach gewählter Zeichenfarbe ist die Linie sichtbar oder nicht. Die Kommandos, die steuern, ob der Stift aufgesetzt ist oder nicht, welche Farbe er hat und in welche Richtung er sich bewegt, geben Sie in einer Zeichenkette an.

Der Draw-Befehl läßt Ihnen alle Freiheiten beim Linienziehen

Das Programm »draw.bas« (Listing 6) demonstriert Ihnen exemplarisch die Fähigkeiten des Befehls. Es malt eine Figur aus Dreiecken auf Ihren Bildschirm (Bild 5). Im einzelnen sind folgende Kommandos erlaubt:

U<Schritte>
Bewegt den »Zeichenstift« um die angegebenen Schritte senkrecht nach oben (Up).

D<Schritte>
Arbeitet wie »U«, nur diesmal nach unten (Down).

L<Schritte>
Geht die angegebene Schrittzahl nach links.

R<Schritte>
Bewegt den »Zeichenstift« nach rechts.

E<Schritte>
Fährt diagonal mit einem Winkel von 45 Grad zur Waagerechten um »Schritte« nach rechts oben.

F<Schritte>
Arbeitet wie »E«, nur nach rechts unten.

G<Schritte>
Bewegt den »Stift« in Richtung unten links.

H<Schritte>
Diesmal geht es nach oben links.

M± <X>,± <Y>
Damit lassen sich, im Gegensatz zu den vorherigen Kommandos, die nur Winkel von 45 Grad zulassen, auch beliebige andere Winkel zeichnen. Dabei bewegt GW-Basic den »Zeichenstift« vom Ausgangspunkt in x- (horizontal) und y-Richtung (vertikal) um die angegebene Schrittweite. Wenn Sie ein Plus oder ein Minus angeben, so werden der Wert von »x« oder »y« zur aktuellen Cursorposition addiert und eine Linie zwischen der alten Position und der neu berechneten gezogen. Be-

achten Sie, daß links oben am Bildschirm die Position »0,0« ist und daß größere Koordinatenwerte nach links beziehungsweise nach unten verschoben sind. Wenn Sie das Vorzeichen weglassen, fährt der »Zeichenstift« auf die entsprechende absolute Bildschirm-Koordinate. Mit

```
DRAW "M+20,-40"
```

erhalten Sie eine Linie, die von der momentanen Cursorposition aus zu einem Punkt führt, der zwanzig Punkte weiter rechts und vierzig Punkte höher liegt. Neben den Bewegungs- gibt es noch zusätzliche Sonderkommandos.

B
Beim nachfolgenden Bewegungskommando wird keine Linie gezeichnet.

N
Nach der folgenden Bewegung kehrt der »Stift« an seinen Ausgangspunkt zurück.

A<n>
Legt den Winkel fest, um den die nachfolgende Figur gegenüber der Normallage gedreht wird. Die für »n« erlaubten Einstellungen können Sie Tabelle 5 entnehmen. Bei einer Drehung von 90 oder 270 Grad werden die anschließenden Bewegungen automatisch so skaliert, daß ein normales Höhen-/Seitenverhältnis von 4/3 eingehalten wird.

Wert	Drehungswinkel
0	0 Grad
1	90 Grad
2	180 Grad
3	270 Grad

Tabelle 5. Diese Drehungswinkel können Sie beim Draw-Befehl einstellen

C<n>
Hiermit können Sie die Farbe für die nachfolgenden Linien bestimmen.

S<n>
Mit »S« legen Sie die Skalierung der folgenden Bewegungen neu fest. Dabei muß <n> zwischen 1 und 255 liegen. Teilt man diesen Wert durch 4, erhält man den tatsächlichen Faktor. Die Vorgabe von GW-Basic ist 4, was einem Faktor von 1 entspricht. Dies bedeutet, daß die Schrittweite 1 auch einem Bildschirmpunkt (Pixel) entspricht. Bei einem Wert von 20 zeichnet der Befehl »DRAW "M+10,+0"« eine Linie, die 50 Bildschirmpunkte lang ist.

X<Variablenname>;
Kommandos können auch in einer Zeichenkette mit dem angegebenen Variablennamen abgelegt werden. Dies ist besonders dann nützlich, wenn Sie eine komplexe Figur mehrmals zeichnen lassen möchten. So erzielt beispielsweise die Befehlsfolge

```
DRAW "M+40,-20M-60,+30"
```

die gleiche Wirkung wie

```
befehle$="M+40,-20"
DRAW "Xbefehle$;M-60,+30"
```

Teile eines zusammengehörigen Kommandos dürfen Sie aber nicht trennen. So führt die Anweisung »MXbefehle$;,-20« zu einer Fehlermeldung. Den Strichpunkt als Abschluß für den Variablennamen dürfen Sie keinesfalls vergessen, sonst kann GW-Basic den Namen der Variablen, die die auszuführende Zeichenkette enthält, nicht richtig identifizieren.

Mit den Befehlen, die Sie in dieser Werkstatt kennengelernt haben, lassen sich ansprechende Anwendungen programmieren. Je mehr Sie experimentieren, besonders mit dem Draw-Befehl, desto sicherer können Sie schon im voraus den Effekt der einzelnen Anweisungen abschätzen. Wenn Sie erst ein Gefühl dafür entwickelt haben, verwirklichen Sie Ihre Ideen mit wenigen Anweisungen.

Wie immer bei der Programmierung gilt auch hier: durch die Praxis zum Erfolg.

(Helmut Stult/wn)

```
[6UØ] 1Ø CLS
[M52] 2Ø SCREEN Ø,Ø,Ø,Ø
[LG3] 3Ø PRINT "Bildschirm Ø"
[W52] 4Ø SCREEN Ø,Ø,1,Ø
[UG3] 5Ø PRINT "Bildschirm 1"
[W52] 6Ø SCREEN Ø,Ø,Ø,Ø
[TN4] 7Ø INPUT "Taste drücken";A$
[662] 8Ø SCREEN Ø,Ø,1,1
[XN4] 9Ø INPUT "Taste drücken";A$
[2D2] 1ØØ SCREEN Ø,Ø,Ø,Ø
```

Listing 1. Mit EGA-Karten kann GW-Basic mehrere Bildschirme verwalten, wie »screen.bas« zeigt

```
[ZB1]    1: 1Ø KEY OFF
[WL1]    2: 2Ø SCREEN 9
[AUØ]    3: 3Ø CLS
[AD3]    4: 4Ø LINE (Ø,Ø)-(639,349),14,B
[H53]    5: 5Ø LINE (Ø,Ø)-(639,349),11
[663]    6: 6Ø LINE (Ø,349)-(639,Ø),12
[8K4]    7: 7Ø LINE (Ø,175)-(639,175),13,,&HCFFC
[LD3]    8: 8Ø LINE (2ØØ,4Ø)-(4ØØ,9Ø),9
[894]    9: 9Ø LINE (2ØØ,5Ø)-STEP(4ØØ,1ØØ),1Ø
[M84]   1Ø: 1ØØ LINE (2ØØ,25Ø)-(4ØØ,32Ø),15,BF
```

Listing 2. »line.bas« veranschaulicht die Anwendung des Line-Befehls

```
[UL1]    1: 1Ø SCREEN 9
[8UØ]    2: 2Ø CLS
[3C1]    3: 3Ø KEY OFF
[K13]    4: 4Ø CIRCLE (15Ø,1ØØ),8Ø,1Ø
[LG3]    5: 5Ø CIRCLE (4ØØ,1ØØ),8Ø,11,2,6
[9G3]    6: 6Ø CIRCLE (15Ø,25Ø),8Ø,12,,,2
[MM3]    7: 7Ø CIRCLE (4ØØ,25Ø),8Ø,13,,,.2
```

Listing 3. Das Programm »circle.bas« zeigt die Fähigkeiten der Circle-Anweisung

```
[UL1]    1: 1Ø SCREEN 9
[8UØ]    2: 2Ø CLS
[3C1]    3: 3Ø KEY OFF
[883]    4: 4Ø LINE(Ø,Ø)-(639,349),1,B
[YX2]    5: 5Ø WINDOW (6,2)-(-6,-2)
[VH2]    6: 6Ø LINE(Ø,2)-(Ø,-2),4
[PH2]    7: 7Ø LINE(5,Ø)-(-5,Ø),5
[SL3]    8: 8Ø FOR X = -5 TO 5 STEP .Ø1
```

```
[MY1]    9: 9Ø  Y = SIN(X)
[5C2]   1Ø: 1ØØ PSET (X,Y),1Ø
[WH1]   11: 11Ø NEXT X
[894]   12: 12Ø WINDOW SCREEN (6,2)-(-6,-2)
[ØV3]   13: 13Ø FOR X = -5 TO 5 STEP .Ø1
[GØ2]   14: 14Ø Y = SIN(X)
[PC2]   15: 15Ø PSET (X,Y),11
[BK1]   16: 16Ø NEXT X
```

Listing 4. »window.bas« macht die Vorteile der Koordinaten-Neudefinition deutlich

```
[UL1]    1: 1Ø SCREEN 9
[8UØ]    2: 2Ø CLS
[3C1]    3: 3Ø KEY OFF
[883]    4: 4Ø LINE(Ø,Ø)-(639,349),1,B
[3L3]    5: 5Ø VIEW (5Ø;2Ø)-(59Ø,14Ø),Ø,3
[ØY2]    6: 6Ø WINDOW (6,2)-(-6,-2)
[XH2]    7: 7Ø LINE(Ø,2)-(Ø,-2),4
[SH2]    8: 8Ø LINE(5,Ø)-(-5,Ø),5
[UL3]    9: 9Ø FOR X = -5 TO 5 STEP .Ø1
[142]   1Ø: 1ØØ  Y = SIN(X)
[722]   11: 11Ø PSET (X,Y)
[ZH1]   12: 12Ø NEXT X
[N85]   13: 13Ø VIEW SCREEN (5Ø,16Ø)-(59Ø,33Ø),Ø,14
[V43]   14: 14Ø WINDOW (6,2)-(-6,-2)
[ØS2]   15: 15Ø LINE(Ø,2)-(Ø,-2),4
[RS2]   16: 16Ø LINE(5,Ø)-(-5,Ø),5
[CV3]   17: 17Ø FOR X = -5 TO 5 STEP .Ø1
[T42]   18: 18Ø  Y = SIN(X)
[Z22]   19: 19Ø PSET (X,Y)
[VH1]   2Ø: 2ØØ NEXT X
```

Listing 5. Die View-Anweisung erlaubt die Definition von Bildschirmfenstern. »view.pas« verdeutlicht die Anwendung.

```
[UL1]    1: 1Ø SCREEN 9
[8UØ]    2: 2Ø CLS
[3C1]    3: 3Ø KEY OFF
[S55]    4: 4Ø A$="BM-5,+4M+5,-9M+5,+9M-1Ø,+ØBM+5,-4"
[A73]    5: 5Ø FOR S=1 TO 131 STEP 1Ø
[BG5]    6: 6Ø DRAW "S=S;AØXA$;A1XA$;A2XA$;A3XA$;"
[MB1]    7: 7Ø NEXT S
```

Listing 6. Mit »DRAW« zeichnen Sie auch komplizierte Körper. Das kurze Programm »draw.bas« ist ein Beispiel dafür.

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 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

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«

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 für Binärdateien
- dBase-Datenbanken in Basic
- Superhirn für Quick Basic 4.0

DOS EXTRA Nr. 7

- MS-DOS-Benutzeroberfläche Easydos
- Benutzeroberfläche für SBasic
- Datengenerator für Basic, C und Pascal
- Sprite-Editor für Farbgrafikkarten
- DOS-Packer
- Virenschutzprogramm
- Weltkarte im Computer
- Einfacher Prozessor-Emulator

DOS EXTRA Nr. 8

- Makroassembler SAM III
- Exe2bin-Ersatz durch "exe2com"
- Programm zur Verbesserung der Speicherverwaltung unter MS-DOS
- Routine zum Speichern von Hardcopies
- Alle Listings zum Einsteigerschwerpunkt und zu den Werkstätten

DOS EXTRA Nr. 10

- Units-Bibliotheken für Turbo Pascal: Datum, Sound, Bildschirmausgabe
- Snapshot-Utility, Version 2.7
- TSR-Datei-Manager
- DOS-Assembler SAM III, Version 1.1
- Schnelles Kopierprogramm
- Komfortable Dateiumleitung
- Hercules-Grafik in Basic programmiert
- Patch für BGI-Treiber von Turbo Pascal

DOS EXTRA Nr. 11

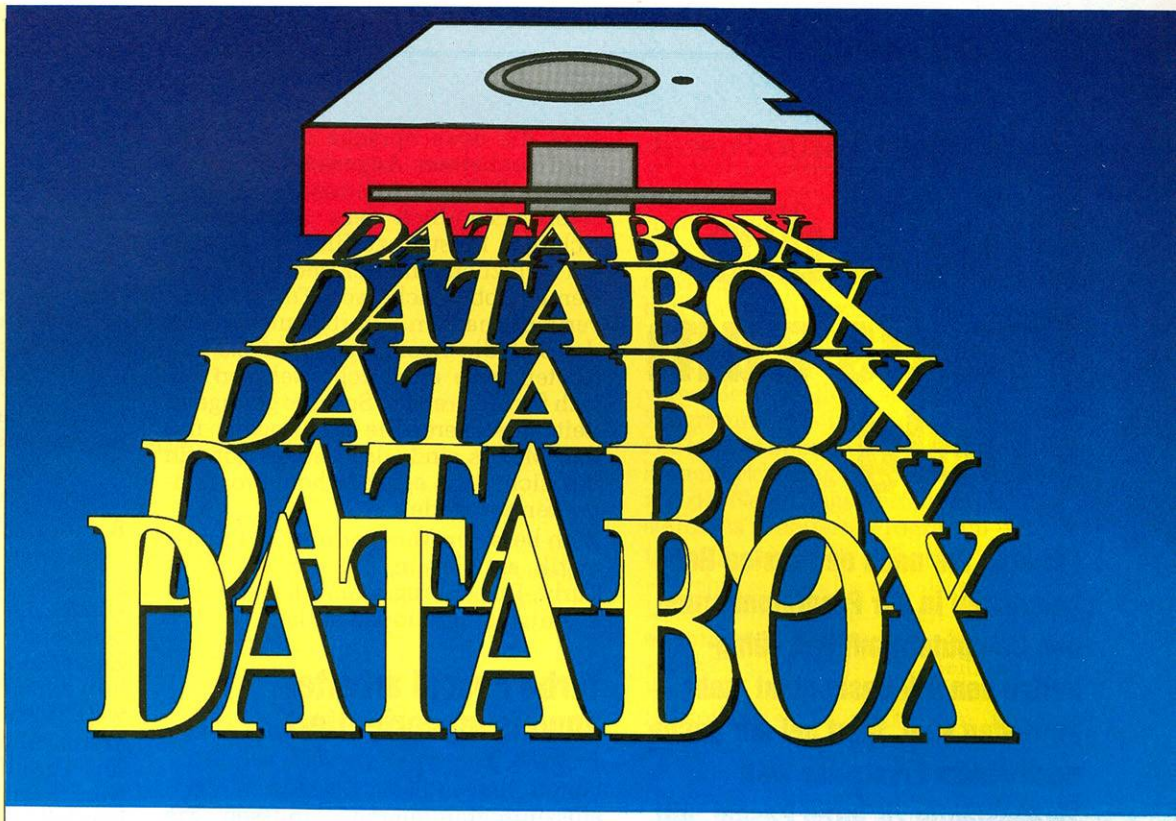
- Preiswerte HPGL-Erweiterung, der HPGL-Extender
- Leistungsstarker Diskettneditor
- Unit für die komfortable Speicherverwaltung unter MS-DOS
- Blockbefehle für den Turbo-Pascal-Editor nach dem SAA-Standard
- Sound-Erzeugung unter Turbo Pascal
- Speicherresidente Uhr

DOS EXTRA Nr. 12

- Kreative Grafik in Basic zum Mitmachen
- Subroutinen zur Anpassung von Basic-Programmen an alle PC-Grafikkarten
- Außergewöhnliche Fraktalgrafiken in Turbo Pascal
- Das ultimative Hardcopy-Utility für fast alle Drucker
- Testbild für Monitore in Basic programmiert

DOS EXTRA Nr. 14

- komfortables Datensicherungsprogramm »SBackup«
- Zeichensatzeditor für 24-Nadel-Drucker
- Pascal-Quelltexte in Schreibweise und Wirkung optimieren
- Utilities zur Bearbeitung von Dateien und Verzeichnissen



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. 15

- der Nachbrenner für GW-Basic: SBasic Version 2.0
- der Tastaturtreiber »SKeybgr« in der Version 4.0
- »Schiffe versenken« mit einem Quick-Basic-Programm
- Simulation von Tastatureingaben mit »Prokey«

DMV-Verlag · Postfach 250 · 3440 Eschwege

DMV Daten- und Medienverlag

Grundlagen Turbo Pascal

Durch die Praxis zum Erfolg

Für alle, die nach den ersten Gehversuchen in der Programmierung von Computern mit den Fähigkeiten von GW-Basic nicht mehr zufrieden sind, führt bei der Suche nach einem Ersatz der Weg zwangsläufig zu Turbo Pascal. Mit dieser Einsteiger-Werkstatt eignen Sie sich die Grundlagen dieser interessanten Sprache an und sind danach in der Lage, kleine Programme selbst zu entwickeln.

Wer bereits in GW-Basic programmiert hat, kennt zumindest die Grundlagen, wie sich ein Problem mit Hilfe des Computers lösen läßt. An der Struktur von Basic liegt es, daß die Programme meist keinen systematischen Aufbau besitzen. Sie bestehen aus einer Aneinanderreihung von Einzelabfragen oder Berechnungen (Algorithmen), zwischen denen mit dem Befehl »GOTO« (gehe zur Zeile) hin und her gesprungen wird. Sie widmen sich ausschließlich der Problemlösung und legen oft keinerlei Wert auf eine saubere Strukturierung oder einen systematischen Aufbau. Da jede Programmzeile eine Nummer besitzt, auf die mit den sogenannten »Sprungbefehlen« ein Zugriff stattfinden kann, ist die Arbeitsweise des Programms ab einer gewissen Größe für Sie gar nicht mehr oder nur sehr schwer nachvollziehbar.

Dieser gravierende Nachteil wurde bereits frühzeitig erkannt, was dazu führte, daß man an Universitäten »Pascal« als Lehrsprache einführte. Die Firma Borland entwickelte diese Grundsprache in mehreren Stufen bis zur aktuellen Version Turbo Pascal 6.0 weiter.

Dem Einsteiger, der dieses Softwareprodukt das erste Mal in Händen hält und die Handbücher durchblättert, wird die neue Sprache sehr kompliziert erscheinen. Angesichts der ungeheuren Fülle an Informationen ist es selbstverständlich sehr schwer, den richtigen Einstieg zu finden. Sie sollten sich zuerst mit der Bedienung der Benutzeroberfläche von Turbo Pascal vertraut machen. Hierbei unterscheiden sich die Versionen 4.0, 5.0/5.5 und 6.0 teilweise erheblich. Nehmen Sie sich für diese ersten Schritte genügend Zeit. Je sicherer Sie im Umgang mit den Menüs sind, desto mehr können Sie sich dem eigentlichen Programmieren zuwenden.

Zum besseren Verständnis der Problematik entwickeln Sie ein einfaches Turbo-Pascal-Programm, das alle notwendigen Funktionen enthält.

Turbo Pascal arbeitet anweisungsorientiert

Waren Sie von Basic gewohnt, einfach »loszuprogrammieren«, müssen Sie sich bei Turbo Pascal über den grundsätzlichen Aufbau Ihres Programms im klaren sein, noch bevor eine Zeile auf dem Bildschirm steht. Deshalb sind vor Ihrem ersten Programm einige Punkte zu klären und festzulegen.

Ein Pascal-Programm besteht grundsätzlich aus zwei Hauptblöcken, dem »Deklarationsteil« und dem »Anweisungsteil«. Vom Deklarationsteil sind für den Anfang nur die Variablen-, Konstanten- und Prozedurdeklarationen wichtig. Beim Anweisungsteil handelt es sich um das eigentliche »Programm« selbst.

Turbo Pascal arbeitet, da es keine Zeilennummern kennt, anweisungsorientiert, das heißt, eine Anweisung kann mehr als eine Zeile sein. Um die Übersichtlichkeit zu wahren und die Strukturierung der einzelnen Programmblöcke zu verdeutlichen, ist es außerdem empfehlenswert, jeden neu beginnenden Programmblock zwei Leerschritte einzurücken. Alle dazugehörenden Zeilen beginnen dann in dieser neuen Spalte. In den folgenden Beispielen erkennen Sie dieses Prinzip deutlich.

Obwohl Groß- und Kleinschreibung zulässig ist, sollten Sie sich doch an die Schreibvereinbarungen von Turbo Pascal halten. Dabei werden alle Anweisungen klein, Funktionen und Prozeduren in wechselnder Groß-/Kleinschreibung geschrieben.

Jedes Turbo-Pascal-Programm beginnt in der ersten Zeile mit der Vereinbarung des Programmnamens. Jede Vereinbarung muß mit einem Semikolon abgeschlossen werden. Damit weiß Turbo Pascal, daß alle vor dem Semikolon stehenden Wörter zur Anweisung gehören. Die erste Zeile kann zum Beispiel lauten:

```
program Erste_Uebung;
```

Bei dieser Angabe bezeichnet »program« den Namen des folgenden Programms. Es heißt in diesem Fall »Erste_Uebung«. Hierbei fallen Ihnen bestimmt zwei Besonderheiten auf. Zuerst einmal sind die beiden Worte mit einem »Unterstrich« verbunden, weil Turbo Pascal ein Leerzeichen als Trennstelle verstehen würde. So wird für Turbo Pascal der Anschein erweckt, daß es sich nur um ein Wort handelt. Weil die deutschen Umlaute nicht zugelassen sind, wird außerdem anstelle von »Ü« das »Ue« verwendet.

Die Vereinbarungen im Deklarationsteil sind die am schwierigsten zu verstehenden. Haben Sie diese Hürde genommen, ist der Rest »fast« ein Kinderspiel.

Von Basic wissen Sie sicher noch, daß Programme wie

```
10 a=5
20 b=3
30 c=a+b
40 print c
```

durchaus erlaubt sind und nach der Eingabe von »run« zur Bildschirmausgabe von »8« führen.

Die Vereinbarungen im Deklarationsteil sind ziemlich kompliziert

In dem Basic-Beispiel sind die beiden Festwerte (Konstanten) »a=5« und »b=3« sowie die zu berechnende und damit veränderliche Größe (Variable) »c« verwendet. In Zeile 30 wird ihr mit der Formel »c=a+b« der Wert zugewiesen. Zeile 40 gibt diesen auf dem Bildschirm aus.

Bei Pascal ist dies so nicht erlaubt. Hier müssen Sie jede Konstante oder

```
program Uebung2;
var     Ergebnis,
        Wert1,
        Wert2: Real;
        Zeichen,
        Anzahl: Byte;
        Operation: Char;
```

Listing 1. Deklarationsvereinbarungen für das zweite Übungsprogramm

```
procedure Eingabe;
begin
  Write('          Wert 1: ');
  ReadLn(Wert1);
  Write('Rechenoperation: ');
  ReadLn(Operation);
  Write('          Wert 2: ');
  ReadLn(Wert2);
end;
```

Listing 2. Die Prozedur »Eingabe«

```
procedure Berechnung;
begin
  if Operation<>'' then
  begin
    case Operation of
      '+': Ergebnis:=Wert1+Wert2;
      '-': Ergebnis:=Wert1-Wert2;
      '*': Ergebnis:=Wert1*Wert2;
      '/': Ergebnis:=Wert1/Wert2;
    else
      begin
        WriteLn('Operation unbekannt !');
        Ergebnis:=0;
      end;
    end;
  end;
end;
```

Bild 3. Die Prozedur »Berechnung«

Variable vor ihrer ersten Benutzung im Programm angeben (deklarieren). Dies hat gegenüber Basic den enormen Vorteil, daß kein unnötiger Speicherplatz für die Ablage freigehalten werden muß. Es besteht ebenfalls keine Begrenzung des Speicherplatzes.

Je nach Art (Datentyp) sind unterschiedliche Wertebereiche zulässig (Tabelle 1). Deshalb ist es sinnvoll, sich schon vor dem Entwickeln des Programms Gedanken zu machen, welchen Wert beispielsweise eine Variable maximal erreichen kann. Danach wählen Sie den in Frage kommenden Typ aus.

Ergänzen Sie Ihr Beispiel also um den Deklarationsteil. Da Turbo Pascal auch längere Bezeichnungen zuläßt, sollten Sie dies nutzen und den Variablen und Konstanten Namen geben, die ihre Funktion verdeutlichen.

```
program Erste_Uebung;
const Wert1: Byte=5;
      Wert2: Byte=3;
var Summe: Byte;
```

Mit »const« vereinbaren Sie die Konstanten (feste Werte). Dem »Wert1« wird »5«, dem »Wert2« dagegen »3« zugewiesen.

Mit »var« beginnt, wie Sie sicher richtig vermuten, die Festlegung der Variablen. Die »Summe« ist vom Typ »Byte« und kann maximal die Zahl 255 erreichen. Da alle drei Deklarationen Byte-Typen sind, benötigen sie nur insgesamt 3 Byte an Speicher.

Jetzt fehlt nur noch die Berechnung der »Summe«. Dies geschieht im Anweisungsteil. Um ihn deutlich hervorzuheben, fügen Sie am besten eine Kommentarzeile ein. Damit Turbo Pascal eine solche erkennt und bei der Ausführung des Programms überspringt, muß sie entweder zwischen zwei geschweiften Klammern stehen »{ Hinweise }« oder durch die Ersatzzeichen »(* « und » *)« gekennzeichnet sein.

Erweitern Sie Ihr Programm deshalb wie folgt:

```
{ Hauptprogramm }
begin
  Summe:=Wert1+Wert2;
  WriteLn(Summe);
end.
```

Mit »begin« kennzeichnen Sie den Anfang des Hauptprogramms. Danach darf kein Semikolon stehen! In der nächsten Zeile wird »Summe« der Wert aus der Addition zugewiesen (Zuweisung). Beachten Sie die Schreibweise »:=«, die sich hier von Basic unterscheidet. Das Gegenstück zum bekannten »print« ist in Turbo Pascal die Prozedur »WriteLn«. Sie gibt das Berechnungsergebnis »Summe« auf den Bildschirm aus, und der Cursor springt danach an den Anfang der nächsten Bildschirmzeile. Dies ist der Unterschied zwischen »Write« (schreibe) und »WriteLn« (schreibe und beginne dann eine neue Zeile). Vergessen Sie nicht das Semikolon am Ende jeder Zeile. Das Ende Ihres ersten Programms legen Sie mit »end.« fest. Der Punkt ist dabei sehr wichtig, da er das Programmende kennzeichnet. Er zeigt an, daß nun keine Anweisungen mehr kommen.

In das Programm »Uebung2« fügen Sie vier neue Berechnungsformeln ein

Wenn Sie jetzt das Programm kompilieren (in Maschinensprache übersetzen), und nach dem Start die Zahl »8« auf dem Bildschirm erscheint, haben Sie Ihr erstes Pascal-Programm erfolgreich abgeschlossen. Selbstverständlich können Sie zum Probieren die Zahlen verändern oder andere Rechenoperationen durchführen, solange nicht die Zahl 255 überschritten wird. Nur werden Sie wohl auf Dauer mit der Leistung Ihres Programms nicht zufrieden sein.

```
procedure Ausgabe;
begin
  Write('       Ergebnis: ');
  WriteLn(Ergebnis);
  WriteLn;
end;
```

Listing 4. Die Prozedur »Ausgabe«

Was liegt näher, als Ihr Programm zu erweitern, damit Sie Zahlen frei eingeben können und zusätzlich die Art der Berechnung wählbar ist.

Da Sie sicher auch den Rechenbereich vergrößern wollen, schreiben Sie Ihr Programm am besten teilweise neu. Die bereits bekannten Punkte bedürfen keiner Erklärung mehr, so daß Sie jetzt nur die Neuerungen erfahren. Listing 1 stellt das erneuerte Programm dar.

Wie Sie sehen, wurde der Variablen »Ergebnis« anstelle von »Byte« der Typ »Real« zugewiesen. Auch die beiden ehemaligen Konstanten »Wert1« und »Wert2« sind jetzt vom gleichen Typ. Beachten Sie die dabei verwendete Schreibweise. Zuweisungen für den gleichen Typ lassen sich, durch ein »Komma« getrennt, zusammenfassen. Die neu hinzugekommene Variable »Operation« nimmt das Rechenzeichen auf. Als Hilfsvariablen dienen »Zeichen« und »Anzahl«. Deren Sinn sehen Sie noch im weiteren Programm.

Um die zusätzlichen Erweiterungen aufzunehmen, könnten Sie den Hauptteil des ersten Programms ausbauen. Doch es gibt eine wesentlich bessere Lösung: die Verwendung von Unterprogrammen. Nach außen verhalten sie sich wie eine eigene Anweisung. Die Turbo-Pascal-Bezeichnung dafür ist »procedure«. Am besten sehen Sie die Anwendung in einem praktischen Beispiel (Listing 2).

Gehen Sie das Programm zeilenweise durch und betrachten Sie die Funktion. Mit »procedure« wird der Anfang des Unterprogramms angegeben. Unter dem Namen »Eingabe« läßt es sich später aufrufen. »begin« kennen Sie bereits. Damit Sie wissen, welche Eingabe zu erfolgen hat, zeigt Ihnen »Write« dies am Bildschirm an. Daß es sich dabei um eine direkte Textausgabe handelt, erkennt Turbo Pascal an den zwei Hochkommas, die den Text einschließen. Die Eingabe geschieht mit »ReadLn«, wobei der eingegebene Wert im ersten Fall der Variablen »Wert1« zugewiesen wird. Er darf dabei die zulässigen Grenzen (Tabelle 2) nicht überschreiten. Der Unterschied zwischen »ReadLn« und »Read« besteht (wie bei »Write« und »WriteLn«) darin, daß nach »ReadLn« der Cursor zusätzlich an den nächsten Zeilenanfang springt. Das Ende der Prozedur markiert wieder »end«, diesmal jedoch von einem Semikolon gefolgt, da es danach noch weitergeht. Jetzt fehlt nur mehr die Berechnung anhand von »Wert1«, »Operation« und »Wert2«. Diese Aufgabe überlassen Sie erneut einem Unterprogramm (Listing 3).

```
procedure Titel;
begin
  WriteLn('  Rechenprogramm');
  for Zeichen:=1 to 70 do
    Write('*');
  WriteLn;
end;
```

Listing 5. Die Prozedur »Titel«

Dieser Teil dürfte der schwierigste sein, doch auch er läßt sich leicht verstehen. Dabei kommen zwei neue Anweisungsarten ins Spiel.
Um die If-Anweisung zu verstehen, müssen Sie ihren Aufbau kennen.

```
if <Bedingung> then <Anweisung>
else
<Anweisung>
```

Die Anweisung, die nach »then« folgt, wird nur ausgeführt, wenn die »Bedingung« erfüllt ist. Soll das Programm mehrere Anweisungen in Abhängigkeit von einer Bedingung ausführen, ist dieser Anweisungsblock in »begin« und »end« einzuschließen. Im Beispiel muß bei der Abfrage der Rechenoperation eine Eingabe erfolgt sein, da sonst die Angabe

```
Operation <> ''
```

nicht zutrifft, was die Ausführung der Anweisungen nach »else« zur Folge hat. Beachten Sie, daß »''« zwei Hochkommas sind, und keine Anführungszeichen. Sie bezeichnen eine Leereingabe.
Da nach der Eingabe mehr als eine einzige Anweisung aufzuführen ist, befinden sich diese zwischen »begin« und »end«. Die Anweisungen sollten untereinander in der gleichen Zeilenspalte stehen; dies ist zwar für die Funktion nicht wichtig, verdeutlicht aber den Programmaufbau erheblich. Sie sollten sich deshalb von Anfang an diese Schreibweise angewöhnen.

Durch eine Prozedur tätigen Sie Ausgaben auf den Bildschirm

In unserem Beispiel sollen die vier Grundrechenarten durchführbar sein. Dazu sind die vier Berechnungsformeln in das Programm einzufügen. Die Wahl der richtigen Berechnung geschieht durch »case«. Das in »Operation« eingegebene Rechenzeichen wird mit den aufgeführten Werten überprüft, bei Übereinstimmung erfolgt die Ausführung der nachstehenden Anweisungen. In diesem Fall ist das nur die Berechnung. Sollte der Wert nicht in den Vorgaben zu finden sein, gibt das Programm die nach »else« stehende Anweisung »Operation unbekannt!« aus und setzt die Variable »Ergebnis« auf 0.
Die schönste Berechnung hilft nichts, wenn das Programm das Ergebnis

```
{ Hauptprogramm }
begin
  ClrScr;
  Titel;
  Anzahl:=1;
  while Anzahl<11 do
  begin
    Eingabe;
    Berechnung;
    Ausgabe;
    Inc(Anzahl);
  end;
end.
```

Listing 6. Die Gestaltung des Hauptprogramms

nicht anzeigt. Versuchen Sie, die dafür notwendige Prozedur einmal allein zu schreiben. Sieht Ihre Lösung so ähnlich wie in Listing 4 aus?
Die erste und dritte Zeile nach »begin« dienen nur einer schöneren Ausgabe und sind somit unwichtig. Der Rest sollte jedoch übereinstimmen. Dabei können Sie anstatt »Ausgabe« auch einen anderen Namen verwendet haben.
Was nun noch fehlt, ist das Hauptprogramm, das die einzelnen Prozeduren in der richtigen Reihenfolge aufruft. Vielleicht vermuten Sie bereits, wie es aussehen könnte:

```
{Hauptprogramm}
begin
Eingabe;
Berechnung;
Ausgabe;
end.
```

Alle Unterprogramme werden nur mit ihrem Namen aufgerufen. Kompilieren Sie nun das fertige Programm und starten Sie es. Wenn Sie beim Abtippen keinen Fehler gemacht haben, müßte die Aufforderung zur Eingabe des ersten Wertes auf dem Bildschirm stehen.
Geben Sie nun zum Beispiel »1+1« ein. Wenn die Turbo-Pascal-Oberfläche wieder erscheint, schalten Sie mit [Alt-F5] auf den MS-DOS-Bildschirm um. Steht nun

```
Ergebnis: 2.0000000000E+00
```

auf dem Bildschirm, stimmt das Programm.
Speichern Sie zuvor das Programm ab, da Sie es noch brauchen. Sie wissen bestimmt, daß das Betriebssystem MS-DOS schon viele Funktionen für den Umgang mit Zusatzgeräten eingebaut hat. So können Sie auf MS-DOS-Ebene mit »cls« den Bildschirm löschen. Auch Funktionen wie das Lesen und Beschreiben von Disketten sind bereits vorhanden. Diese Programme sind aber in Turbo Pascal nicht verwendbar. Wenn Sie trotzdem zum Beispiel den Bildschirm nach jedem Programmdurchlauf löschen wollen, funktioniert das nur, wenn Sie eine entsprechende Funktion dafür entwickeln. Um Ihnen die Arbeit des Programmierens abzunehmen, haben die Entwickler von Turbo Pascal die »units« erfunden.
Sie benötigen die Unit »Crt«. Damit das Beispielprogramm weiß, daß diese Unit benutzt werden soll, ändern Sie den Programmanfang folgendermaßen:

```
program Uebung2;
uses Crt;
```

Nun stehen Ihnen die zusätzlichen Befehle des Unit zur Verfügung. Den Befehl zum Löschen des Bildschirms benutzen Sie in Ihrem Programm. Tragen Sie dafür im Hauptteil nach »begin« noch die Zeile

```
ClrScr;
```

ein. »ClrScr« löscht nun den Bildschirm nach jedem Programmstart.

Datentyp	Wertebereich von/bis	Platzbedarf
Byte	0 bis 255	8 Bit
Shortint	-128 bis 127	8 Bit+Vorzeichen
Word	0 bis 65535	16 Bit
Integer	-32768 bis 32767	16 Bit+Vorzeichen
Longint	-2147483648 bis 2147483647	32 Bit+Vorzeichen
Char	ein Zeichen aus dem Zeichensatz	8 Bit
string[n]	n Zeichen aus dem Zeichensatz (maximal 255 Zeichen)	8 Bit*n+8 Bit
Boolean	»True« oder »False« (0 oder 1)	1 Bit

Tabelle 1. Übersicht der Typen für Variablen und Konstanten mit den Kenndaten

Typ	Bereich	Genauigkeit
Real	$2.9*10^{-39}$ bis $1.7*10^{38}$	11 -12 Stellen
Single	$1.5*10^{-45}$ bis $3.4*1.0^{38}$.	7 - 8 Stellen
Double	$5.0*10^{-324}$ bis $1.7*10^{308}$	15 -16 Stellen
Extended	$1.9*10^{-4951}$ bis $1.1*10^{4932}$	19 -20 Stellen

Tabelle 2. Übersicht des Wertebereichs und der Genauigkeit für Real-Typen (Fließkommazahlen)

Turbo Pascal erlaubt auch, etwas mehrmals zu wiederholen. Zur Veranschaulichung fügen Sie dazu eine neue Prozedur vor die »procedure Eingabe« ein (Listing 5).

Die Anweisung »for-to-do« ist die Pascal-Version der Ihnen von Basic bekannten For-Next-Schleife. Jetzt bekommt auch die bisher ungenutzte Variable »Zeichen« ihre Aufgabe. Sie dient in diesem Fall nur als »Dummy«, das heißt, daß sie nicht direkt ausgegeben wird. Die Schleife zählt so lange um den Wert 1 weiter, bis die Zahl 70 erreicht ist und schreibt jedesmal das Zeichen »*« auf den Bildschirm. Insgesamt somit 70mal.

Mit Schleifen wiederholen Sie die Abarbeitung von Anweisungen

Möchten Sie noch weitere Anweisungen innerhalb der Schleife ausführen lassen, so müssen diese wieder zwischen »begin« und »end« stehen. Um rückwärts zu zählen, ändern Sie »to« in »downto«.

Die neue Prozedur können Sie nur aufrufen, wenn Sie im Hauptprogramm nach »ClrScr« folgendes hinzufügen:

```
Titel;
```

Damit sieht das Demo-Programm schon etwas ansprechender aus.

Bis jetzt können Sie mit dem Programmbeispiel nur immer eine einzige Berechnung durchführen. Damit das Programm nicht endlos läuft, setzen Sie die Zahl der Berechnungen auf zehn fest. Hierfür sind im Hauptprogramm fünf Zeilen einzufügen. Der Übersichtlichkeit wegen ist es in Listing 6 noch einmal komplett dargestellt. Die gewünschten zehn Berechnungen werden durch die bedingte While-Do-Schleife erreicht. Da jede numerische Variable nach deren Vereinbarung einen zufälligen Wert annimmt, ist diese zu Beginn des Programms immer zu initialisieren.

```
Anzahl:=1
```

Die Variable »Anzahl« wird in dem Beispiel mit 1 initialisiert. Die Schleife

```
while Anzahl<11 do
```

wird nun solange durchlaufen, wie die Bedingung »Anzahl ist kleiner als 11« erfüllt ist. Um den Wert »Anzahl« zu erhöhen, zählt die Prozedur »Inc(Anzahl)« bei jedem Schleifendurchlauf um die Zahl 1 hoch. Da bei »while« die Abbruchbedingung am Anfang steht, erfolgt das Verlassen der Schleife, wenn »Anzahl« den Wert 11 erreicht hat und damit die Bedingung stimmt. Nun wissen Sie endlich auch, wozu die Variable »Anzahl« dient, die bisher nicht gebraucht wurde. Das komplette Programm finden sie noch einmal übersichtlich in »uebung2 .pas« (Listing 7). Das gleiche läßt sich mit der Struktur »repeat-until« erreichen. Dabei steht die Abbruchbedingung am Ende der Schleife.

Tauschen Sie die While-Do-Schleife gegen folgendes aus:

```
repeat
  Eingabe;
  Berechnung;
  Ausgabe;
  Inc(Anzahl);
until Anzahl=10
```

Ändern Sie außerdem die Zuweisung im Hauptprogramm für »Anzahl« auf den Wert 0.

```
Anzahl:=0
```

Dies ist nötig, da bei der ersten Abfrage »Anzahl« bereits um 1 hochgezählt wurde. Eine neue Berechnung kann dadurch ebenfalls so lange erfolgen, bis Anzahl gleich 10 ist.

Beim nächsten Beispiel handelt es sich um ein neues Übungsprogramm. Die Vorgehensweise wird Ihnen jetzt schon geläufig sein. Geben Sie dazu das Programm »spiel.pas« (Listing 8) ein. Es ist das bekannte »Zahlenraten«. Schauen Sie nun, was Sie davon noch nicht kennen. Alle Neuerungen finden Sie in der Prozedur »Erzeuge_Zufallszahl«.

Für das Spiel benötigen Sie eine Zahl, die nur der Computer kennt. Diese stellt die Funktion »Random« bereit.

Die Funktion erzeugt eine Zahl, die im Bereich zwischen 0 und 255 liegt, und weist ihren Wert der Variablen »Suchzahl« zu. Da die Reihenfolge dieser Zahlen nach jedem Start des Computers gleich ist, wird vor der Random-Funktion erst der »Zufallszahlengenerator« gestartet. Dieser verwendet für seinen Startwert das Datum und die Uhrzeit. Nicht immer will man warten, bis beispielsweise eine Programmschleife hundertmal durchlaufen wurde. Um dies abzubrechen, könnten Sie zum Beispiel abfragen, ob ein Tastendruck erfolgt ist. Ist dies der Fall, wird die Schleife verlassen. Betrachten Sie dazu folgenden Listing-Auszug:

```
repeat
  Suchzahl:=Random(256);
until KeyPressed;
```

Die Variable »Suchzahl« nimmt so lange neue Werte an, bis eine Taste gedrückt wird. Die Funktion »KeyPressed« benötigt ebenfalls die Unit »Crt«. Deshalb steht auch am Programmanfang die Zeile:

```
uses Crt;
```

Dieses Programm prüft, ob die Zahl schon gefunden ist. Wenn dies nicht der Fall ist, wiederholt sich die Schleife. Um die Anzahl der benötigten Versuche zu ermitteln, können Sie diese mit

```
Inc(Versuche);
```

zählen. Die Prozedur »Ausgabe« zeigt die Anzahl der Versuche an.

(Helmut Stult/Kerrin Uhlmann/db)

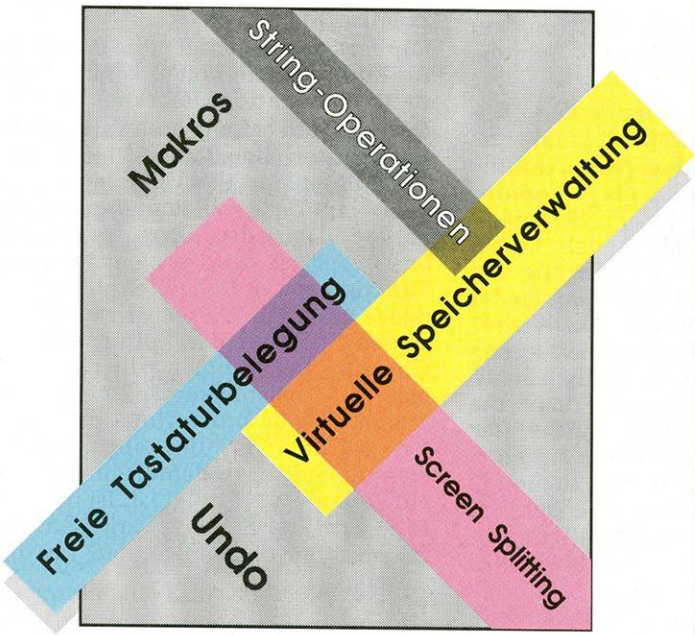
```
[8S3]    1: {Programm: uebung2.pas
[4G5]    2:  Funktion: Durchführung der vier
[NY2]    3:            Grundrechenarten
[ET3]    4:  Sprache:  Turbo Pascal
[GN3]    5:  Autor:    Helmut Stult
[TX4]    6:  (c)1991 DMV Widuch GmbH & Co. KG)
[ØØØ]    7:
[YS2]    8: program Uebung2;
[5H1]    9: uses Crt;
[N72]   1Ø: var Ergebnis,
[ZZØ]   11:     Wert1,
[NX1]   12:     Wert2: Real;
[DD1]   13:     Zeichen,
[B92]   14:     Anzahl: Byte;
[8P2]   15:     Operation: Char;
[ØØØ]   16:
[PV2]   17: procedure Titel;
[MUØ]   18: begin
[8G4]   19:   WriteLn('  Rechenprogramm');
[FP3]   2Ø:   for Zeichen:=1 to 7Ø do
[TM1]   21:     Write('*');
[XF1]   22:   Writeln;
[ZNØ]   23: end;
[ØØØ]   24:
[A23]   25: procedure Eingabe;
[MUØ]   26: begin
[3G3]   27:   Write('         Wert 1: ');
[S92]   28:   ReadLn(Wert1);
[RG4]   29:   Write('Rechenoperation: ');
[M13]   3Ø:   ReadLn(Operation);
```

```
[6G3]   31:   Write('         Wert 2: ');
[T92]   32:   ReadLn(Wert2);
[ZNØ]   33: end;
[ØØØ]   34:
[ZM3]   35: procedure Berechnung;
[MUØ]   36: begin
[AB3]   37:   if Operation<>'' then
[MUØ]   38:   begin
[FV2]   39:     case Operation of
[ZC4]   4Ø:       '+': Ergebnis:=Wert1+Wert2;
[9D4]   41:       '-': Ergebnis:=Wert1-Wert2;
[UC4]   42:       '*': Ergebnis:=Wert1*Wert2;
[LD4]   43:       '/': Ergebnis:=Wert1/Wert2;
[ZRØ]   44:     else
[MUØ]   45:       begin
[YC5]   46:         WriteLn('Operation unbekannt !');
[CZ1]   47:         Ergebnis:=Ø;
[ZNØ]   48:       end;
[ZNØ]   49:     end;
[ZNØ]   5Ø:   end;
[ZNØ]   51: end;
[ØØØ]   52:
[G23]   53: procedure Ausgabe;
[MUØ]   54: begin
[VS3]   55:   Write('       Ergebnis: ');
[L53]   56:   WriteLn(Ergebnis);
[XF1]   57:   WriteLn;
```

Listing 7. Das komplette Programm »uebung2.pas«

Prof Ed

Der Editor für Programmierer



Ob Sie programmieren, einfache Texte schreiben oder Ihre Config.sys- oder Autoexec.bat- Datei umändern wollen:
ProfEd ist das ideale Werkzeug für Sie.

Und das bietet ProfEd:

- Bis zu fünf Dateien lassen sich gleichzeitig bearbeiten.
- Mehrere MByte lange Dateien sind kein Problem für ProfEd.
- Jede Taste der Tastatur kann mit einem oder mehreren Kommandos versehen werden.
- Makros gestatten beliebig viele Editorkommandos abzuarbeiten.
- Mehrere Undo-Funktionen lassen den letzten Arbeitsschritt rückgängig machen.
- Teile einer oder verschiedener Dateien lassen sich gleichzeitig am Bildschirm zur Bearbeitung bereitstellen.
- Suchen und Ersetzen von Zeichenfolgen ist überhaupt kein Problem.
- Zeilen- und spaltenweises Löschen, Kopieren, Verschieben und Füllen von markierten Textbereichen gestatten ein schnelles Bearbeiten Ihrer Dateien. Diese Funktion arbeitet dateiübergreifend und ermöglicht jede nur denkbare Textmanipulation.
- Eine spezielle Netzwerkversion ist erhältlich.

Systemvoraussetzung:
PC, XT, AT, 386er oder 486er Rechner mit mindestens 256 KByte RAM, DOS ab Version 3.0

ProfEd DM **99,–***

Fontline DM **299,–***

Softfontmanipulation und Druckertreibergenerierung

* Unabhängig von der Anzahl der bestellten Programme berechnen wir für das Inland DM 4,– bzw. für das Ausland DM 6,– Porto und Verpackung.
Bitte benutzen Sie die Bestellkarte.

DMV-Verlag • Postfach 250 • 3440 Eschwege DMV Daten- und Medienverlag


```
[ZNØ]    58: end;
[ØØØ]    59:
[753]    6Ø: { Hauptprogramm }
[ØØØ]    61:
[MUØ]    62: begin
[DB1]    63:   ClrScr;
[131]    64:   Titel;
[1P1]    65:   Anzahl:=1;
[TV2]    66:   while Anzahl<11 do
[MUØ]    67:   begin
[VB1]    68:     Eingabe;
[812]    69:     Berechnung;
[BD1]    7Ø:     Ausgabe;
[SØ2]    71:     Inc(Anzahl);
[ZNØ]    72:   end;
[KNØ]    73: end.
```

Listing 7. Das komplette Programm »uebung2.pas«

```
[BF3]     1: {Programm: spiel.pas
[7A5]     2:  Funktion: Beispielprogramm zur
[MW3]     3:            Erzeugung von Zufalls-
[R21]     4:            zahlen
[ET3]     5:  Sprache:  Turbo Pascal
[GN3]     6:  Autor:    Helmut Stult
[TX4]     7: (c)1991 DMV Widuch GmbH & Co. KG}
[ØØØ]     8:
[WG2]     9: program Spiel;
[5H1]    1Ø: uses Crt;
[6C2]    11: var  Versuche,
[MH1]    12:      Suchzahl,
[PN2]    13:      Vermutet: byte;
[ØØØ]    14:
[PV2]    15: procedure Titel;
[MUØ]    16: begin
[2U5]    17:   WriteLn('Zahlenraten von Ø bis 255');
[XF1]    18:   WriteLn;
[ZNØ]    19: end;
[ØØØ]    2Ø:
[RF5]    21: procedure Erzeuge_Zufallszahl;
[MUØ]    22: begin
[1T1]    23:   Randomize;
[X95]    24:   WriteLn('Bitte Taste drücken');
[P31]    25:   repeat
[ML3]    26:     Suchzahl:=Random(256);
[K13]    27:   until Keypressed;
[ZNØ]    28: end;
[ØØØ]    29:
[A23]    3Ø: procedure Eingabe;
[MUØ]    31: begin
[V74]    32:   Write('Ihre Vermutung: ');
[2X2]    33:   ReadLn(Vermutet);
[ZNØ]    34: end;
[ØØØ]    35:
[GE3]    36: procedure Pruefung;
[MUØ]    37: begin
[ZA4]    38:   if Vermutet<Suchzahl then
[VH3]    39:     WriteLn('Zu klein !');
[1B4]    4Ø:   if Vermutet>Suchzahl then
[1M3]    41:     WriteLn('Zu gross !');
[ZNØ]    42: end;
[ØØØ]    43:
[G23]    44: procedure Ausgabe;
[MUØ]    45: begin
[XF1]    46:   WriteLn;
[294]    47:   Write('Sie haben die Zahl ');
[TS2]    48:   Write(Suchzahl);
[ØØ2]    49:   Write(' in ');
[DT2]    5Ø:   Write(Versuche);
[9B5]    51:   WriteLn(' Versuchen gefunden');
[ZNØ]    52: end;
[ØØØ]    53:
[753]    54: { Hauptprogramm }
[MUØ]    55: begin
[DB1]    56:   ClrScr;
[MØ2]    57:   Versuche:=Ø;
[131]    58:   Titel;
[KS3]    59:   Erzeuge_Zufallszahl;
[P31]    6Ø:   repeat
[VB1]    61:     Eingabe;
[FN1]    62:     Pruefung;
[1D2]    63:     Inc(Versuche);
[MB4]    64:   until Vermutet=Suchzahl;
[BD1]    65:   Ausgabe;
[KNØ]    66: end.
```

Listing 8. Zwar ist »spiel.pas« kein Meisterwerk, jedoch zeigt es, daß auch mit wenigen Befehlen schon etwas möglich ist

DOS EXTRA



DOS EXTRA 2
3D-CAD-Programm
DOS-Commander
DOS-Utilities
SForth-Compiler

DOS EXTRA 3
Der PC – glasklar
DFÜ-Programm
Hercules-Grafik
Sprache digitalisiert

DOS EXTRA 4
Speicherresidente Programme
DTP-Programm
EGA-Grafik
MS-DOS Intern

DOS EXTRA 5
PC-Grafikkarten
Das BIOS des PC
GW-Basic
Gerätetreiber programmieren

DOS EXTRA 6
Basic im Detail
GW-Basic intern
SBasic-Compiler
Basic-Utilities

DOS EXTRA 7
PC-Komponenten
Sprite-Editor
SBasic Plus
C-Grundlagen

DOS EXTRA 8
Assembler SAM III
Assembler und Hochsprachen
Assembler-Werkstatt
Assembler-Grundlagen

DOS EXTRA 10
MS-DOS 4.01
Hardware-Praxis
Snapshot-Programm
Turbo-Pascal-Units

DOS EXTRA 11
Turbo Pascal total
Turbo Pascal intern
SAA-Blockbefehle
MS-DOS-Speicherverwaltung

DOS EXTRA 12
Grafik im Basic
Hardcopy-Utility
Fraktalgrafik in Turbo Pascal
Basic für PC-Grafikkarten

DOS EXTRA 13
Spielekategorien
Die besten Spiele
Abenteuerspiele
Flugsimulator
Sharewarespiele



NEU

DOS EXTRA Nr. 14

Aus dem Inhalt der DATABOX:

- Komfortables Datensicherungsprogramm mit Benutzeroberfläche
- Professioneller Zeichensatzeditor für 24-Nadeldrucker
- Turbo Pascal
- Unit zur Generierung von 3D
- Animation
- Pascal-Quelltexte in Schreibweise und Wirkung optimieren
- Utilities zur Bearbeitung von Dateien und Verzeichnissen
- Schnelle Grafikroutinen und mathematische Funktionen

Vorsprung für PC-Benutzer mit DOS EXTRA im Abonnement!

4 Ausgaben bei vierteljährlicher Erscheinungsweise mit dem bequemen Zustellerservice per Post:

BRD	DM 60,–
Europa	DM 70,–
Außereuropäisches Ausland	DM 80,–

Keine Bestellprobleme mit dem Dauerbezug der DOS-EXTRA-DATABOX!

Bequemes Dauerbezugsverfahren mit Bankabbuchung und vierteljährlicher Kündigungsmöglichkeit:

BRD	DM 39,–
Europa	DM 41,–
Außereuropäisches Ausland	DM 42,–

Bitte benutzen Sie nur die Bestellkarte!

DATABOX

Die Software zu den DOS-EXTRA-Heften

Sagenhaft günstig, extra viel gute Software fürs Geld! Bringt alle Programme der DOS EXTRA auf Datenträger

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 kommen, sofern diese nicht vollkompatibel sind.

jeweils **2 Disketten 5 1/4"**
1 Diskette 3 1/2"

35,– DM
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 nur die Bestellkarte!

DMV-Verlag • Postfach 250 • 3440 Eschwege

DMV Daten- und Medienverlag

Turbo Pascal auf einen Blick

PC-Programmierer antworten auf die Frage, welche Programmiersprache sie denn verwenden, recht häufig mit Turbo Pascal. Nun hat dieses System (inzwischen Version 6.0) im Laufe seiner Entwicklungsgeschichte doch einige Veränderungen erfahren. In früheren Versionen enthaltene Befehle sind nicht mehr vorhanden, neue dazugekommen. Daher möchten wir Ihnen mit folgender Tabelle einen Überblick über die in den Versionen 3.0, 4.0, 5.0/5.5 und 6.0 enthaltenen Befehle geben.

(Christian Q. Spitzner/rm)

Befehl/Funktion	Turbo Pascal 3.0 Syntax ( ): Ergebnistyp	Turbo Pascal 4.0 Syntax ( ): Ergebnistyp	Turbo Pascal 5.0/5.5 Syntax ( ): Ergebnistyp	Turbo Pascal 6.0 Syntax ( ): Ergebnistyp	Beschreibung
Abs	Abs (*z*: Integer/Real):Integer/Real	Abs (*z*: Integer/Real): Integer/Real	Abs (*z*: Integer/Real): Integer/Real	Abs (*z*: Integer/Real): Integer/Real	liefert den Absolutwert von *z*
Addr	Addr (*b*): Integer	Addr (*b*): Pointer	Addr (*b*): Pointer	Addr (*b*): Pointer	übergibt Speicheradresse der Variablen *b*
Append	–	Append (d: Text)	Append (d: Text)	Append (d: Text)	öffnet und erweitert die Datei *d*
Arc	Arc (x,y,winkel,radius,farbe)	Arc (x: Integer,y: Integer,start: Word, ende: Word,radius: Word)	Arc (x: Integer,y: Integer,start: Word, ende: Word,radius: Word)	Arc (x: Integer,y: Integer,start: Word, ende: Word,radius: Word)	zeichnet einen Kreis beziehungsweise Teilkreis (*start, ende* in Grad)
ArcTan	ArcTan (z: Real/Integer): Real	ArcTan (z: Real): Real	ArcTan (z: Real): Real	ArcTan (z: Real): Real	liefert den Arcustangens von *z*
Assign	Assign (d,n: String)	Assign (d,n: String)	Assign (d,n: String)	Assign (d,n: String)	ordnet der Dateivariablen *d* die externe Datei *n* zu
AssignCrt	AssignCrt (dvar: Text)	AssignCrt (dvar: Text)	AssignCrt (dvar: Text)	AssignCrt (dvar: Text)	ordnet dem Bildschirm eine Textdatei zu
Back	Back (weg: Integer)	–	–	–	bewegt Grafikzeiger um *weg* rückwärts
Bar	–	Bar (x1: Integer,y1: Integer,x2: Integer, y2: Integer)	Bar (x1: Integer,y1: Integer,x2: Integer, y2: Integer)	Bar (x1: Integer,y1: Integer,x2: Integer, y2: Integer)	zeichnet ein ausgefülltes Rechteck
Bar3D	–	Bar3D (x1: Integer,y1: Integer,x2: Integer, y2: Integer,tiefe: Word,dach: Boolean)	Bar3D (x1: Integer,y1: Integer,x2: Integer, y2: Integer,tiefe: Word,dach: Boolean)	Bar3D (x1: Integer,y1: Integer,x2: Integer, y2: Integer,tiefe: Word,dach: Boolean)	zeichnet ein dreidimensionales Rechteck
Beep	–	–	–	–	erzeugt einen kurzen Piepston
BlockRead	BlockRead (bd: File,ziel: Word,s: Word)	BlockRead (bd: File,ziel,a: Word,s: Word)	BlockRead (bd: File,ziel,a: Word,s: Word)	BlockRead (bd: File,ziel,a: Word,s: Word)	liest *a* Blöcke einer Datei *bd* in eine Puffervariable *ziel*. *s* ist die Summe der gelesenen Blöcke
BlockWrite	BlockWrite (bd: File,quel,a: Word,s: Word)	BlockWrite (bd: File,quel,a: Word,s: Word)	BlockWrite (bd: File,quel,a: Word,s: Word)	BlockWrite (bd: File,quel,a: Word,s: Word)	schreibt a Blöcke aus Puffervariable quel in Datei bd. s ist die Anzahl geschriebener Blöcke
Chain	Chain (programm: String)	–	–	–	ruft ein Nachfolgeprogramm auf und löscht das aktuelle Programm
ChDir	ChDir (dir: String)	ChDir (dir: String)	ChDir (dir: String)	ChDir (dir: String)	Verzeichniswechsel: *dir* muß ein gültiger DOS-Pfad sein
Chr	Chr (asciicode: Ordinaltyp): Char	Chr (asciicode: Byte): Char	Chr (asciicode: Byte): Char	Chr (asciicode: Byte): Char	übergibt dem Wert entsprechendes ASCII-Zeichen
Circle	Circle (x,y,rad,farbe: Integer)	Circle (x: Integer,y: Integer,radius: Word)	Circle (x: Integer,y: Integer,radius: Word)	Circle (x: Integer,y: Integer,radius: Word)	zeichnet einen Kreis um die Koordinate *x,y* mit dem Radius *radius*
ClearDevice	ClearDevice	ClearDevice	ClearDevice	ClearDevice	löscht den Bildschirm und setzt alle Grafikparameter auf ihre Standardwerte zurück
ClearScreen	ClearScreen	–	–	–	löscht den gesamten Bildschirm und setzt den Cursor in die linke obere Bildschirmecke
ClearViewPort	ClearViewPort	ClearViewPort	ClearViewPort	ClearViewPort	löscht den Inhalt des aktuellen Zeichenfensters
Close	Close (d)	Close (d)	Close (d)	Close (d)	schließt Datei *d*
CloseGraph	–	CloseGraph	CloseGraph	CloseGraph	schaltet den Grafikmodus ab
ClrEol	ClrEol	ClrEol	ClrEol	ClrEol	löscht alle Zeichen von der Cursorposition bis zum Zeilenende
ClrScr	ClrScr	ClrScr	ClrScr	ClrScr	löscht den gesamten Bildschirm und setzt den Cursor in die linke obere Bildschirmecke
ColorTable	ColorTable (f1,f2,f3,f4)	–	–	–	ändert die Farben einer Farbpalette

Befehl/Funktion	Turbo Pascal 3.0 Syntax ( ): Ergebnistyp	Turbo Pascal 4.0 Syntax ( ): Ergebnistyp	Turbo Pascal 5.0/5.5 Syntax ( ): Ergebnistyp	Turbo Pascal 6.0 Syntax ( ): Ergebnistyp	Beschreibung
Concat	Concat (s1: String,...,sn: String): String	Concat (s1: String,...,sn: String): String	Concat (s1: String,...,sn: String): String	Concat (s1: String,...,sn: String): String	verknüpft einzelne Strings (*s1* bis *sn*) zu einem String
Copy	Copy (s: String,ab: Integer,l: Integer): String	Copy (s: String,ab: Integer,l: Integer): String	Copy (s: String,ab: Integer,l: Integer): String	Copy (s: String,ab: Integer,l: Integer): String	kopiert *l* Zeichen ab Zeichen *ab* von String *s*
Cos	Cos (z: Real): Real	Cos (z: Real): Real	Cos (z: Real): Real	Cos (z: Real): Real	liefert den Cosinus von *z*
CrtExit	CrtExit	–	–	–	schickt den Terminal-Reset-String zum Bildschirm
CrtInit	CrtInit	–	–	–	schickt den Terminal-Initialisierungs-String zum Bildschirm
Cseg	CSeg: Integer	CSeg: Word	CSeg: Word	CSeg: Word	liefert die Basisadresse des momentanen Codesegments
Dec	Dec (z: Integer)	Dec (z: Integer)	Dec (z: Integer)	Dec (z: Integer)	liefert das Ergebnis von (z - 1)
Dec	Dec (z: Integer,zaehler: LongInt)	Dec (z: Integer,zaehler: LongInt)	Dec (z: Integer,zaehler: LongInt)	Dec (z: Integer,zaehler: LongInt)	Ergebnis von (z - zaehler)
Delay	Delay (delta__t: Word)	Delay (delta__t: Word)	Delay (delta__t: Word)	Delay (delta__t: Word)	hält Programmablauf für Zeitspanne delta__t in Millisekunden auf
Delete	Delete (s: String,ab: Integer,l: Integer)	Delete (s: String,ab: Integer,l: Integer)	Delete (s: String,ab: Integer,l: Integer)	Delete (s: String,ab: Integer,l: Integer)	löscht *l* Zeichen ab Zeichen *ab* von String *s*
DelLine	DelLine	DelLine	DelLine	DelLine	löscht die gesamte Zeile in der sich der Cursor befindet
DetectGraph	–	DetectGraph (grafiktreiber: Integer, grafikmodus: Integer)	DetectGraph (grafiktreiber: Integer, grafikmodus: Integer)	DetectGraph (grafiktreiber: Integer, grafikmodus: Integer)	bestimmt den Grafiktreiber und dazugehörigen Grafikmodus
DiskFree	–	DiskFree (laufwerk: Byte): LongInt	DiskFree (laufwerk: Byte): LongInt	DiskFree (laufwerk: Byte): LongInt	liefert den noch freien Speicher des Laufwerks in Byte
DiskSize	–	DiskSize (laufwerk: Byte): LongInt	DiskSize (laufwerk: Byte): LongInt	DiskSize (laufwerk: Byte): LongInt	liefert den Gesamtspeicherbereich des Laufwerks in Byte
Dispose	–	–	Dispose (p:Pointer, Destrukt)	Dispose (p:Pointer, Destrukt)	speziell für OOP
Dispose	Dispose (p: Pointer)	Dispose (p: Pointer)	Dispose (p: Pointer)	Dispose (p: Pointer)	gibt Speicherbereich der Bezugsvariablen *p* wieder frei
DosExitCode	–	DosExitCode: Word	DosExitCode: Word	DosExitCode: Word	liefert den Fehlercode nach einem Ein-/Ausgabeprozeß
DosVersion	–	–	DosVersion: Word	DosVersion: Word	liefert die Versionsnummer des Betriebssystems
Draw	Draw (x,1,y1,x2,y2: Integer)	–	–	–	zeichnet eine Linie zwischen den Koordinaten *x1,y1* und *x2,y2* in der Farbe *farbe*
DrawPoly	–	DrawPoly (anzahl: Word,koordinatenfeld: PointType)	DrawPoly (anzahl: Word,koordinatenfeld: PointType)	DrawPoly (anzahl: Word,koordinatenfeld: PointType)	zeichnet einen Polygonzug beziehungsweise ein Vieleck
Dseg	DSeg: Integer	DSeg: Word	DSeg: Word	DSeg: Word	liefert Segmentadresse des Datensegments
Ellipse	–	Ellipse (x: Integery: Integer,start: Word, ende: Word,a: Word,b: Word)	Ellipse (x: Integery: Integer,start: Word, ende: Word,a: Word,b: Word)	Ellipse (x: Integery: Integer,start: Word, ende: Word,a: Word,b: Word)	zeichnet eine Ellipse beziehungsweise einen Ellipsenbogenausschnitt
EnvCount	–	–	EnvCount: Integer	EnvCount: Integer	liefert die Anzahl der von DOS gesetzten Variablen
EnvStr	–	–	EnvStr (nummer: Integer): String	EnvStr (nummer: Integer): String	liefert die von DOS gesetzten Variablen
Eof	Eof (d): Boolean	Eof (d): Boolean	Eof (d): Boolean	Eof (d): Boolean	überprüft, ob das Ende der Datei *d* erreicht ist
Eoln	Eoln (d): Boolean	Eoln (d): Boolean	Eoln (d): Boolean	Eoln (d): Boolean	überprüft, ob das Ende einer Zeile erreicht ist
Erase	Erase (d)	Erase (d)	Erase (d)	Erase (d)	löscht die Datei *d*
Exec	–	Exec (programm: String,kommando: String)	Exec (programm: String,kommando: String)	Exec (programm: String,kommando: String)	startet aus einem Programm heraus ein anderes
Execute	Execute (f: File)	–	–	–	Datei ausführen
Exit	Exit	Exit	Exit	Exit	Blockabbruch und Rücksprung in die aufrufende Ebene
Exp	Exp (z: Integer/Real): Real	Exp (z: Real): Real	Exp (z: Real): Real	Exp (z: Real): Real	Exponentialfunktion (e hoch *z*)
Fail	–	–	–	Fail	gibt die dynamische Instanz eines Objekts wieder frei und verläßt Konstrukt
FExpand	–	–	FExpand (suchweg: PathStr): PathStr [String[79]]	FExpand (suchweg: PathStr): PathStr [String[79]]	erweitert einen unvollständigen Dateinamen um den dazugehörigen Suchweg
FilePos	FilePos (d): Integer	FilePos (d): LongInt	FilePos (d): LongInt	FilePos (d): LongInt	liefert Position des Zeigers innerhalb der Datei d
FileSize	FileSize (d): Integer	FileSize (d): LongInt	FileSize (d): LongInt	FileSize (d): LongInt	gibt Größe der Datei d an
FillChar	–	FillChar (v, a: Word, z: Char/Byte)	FillChar (v, a: Word, z: Char/Byte)	FillChar (v, a: Word, z: Char/Byte)	füllt *a* Speicherzellen der beliebigen Variablen *v* mit dem Zeichen *z*

Befehl/Funktion	Turbo Pascal 3.0 Syntax ( ): Ergebnistyp	Turbo Pascal 4.0 Syntax ( ): Ergebnistyp	Turbo Pascal 5.0/5.5 Syntax ( ): Ergebnistyp	Turbo Pascal 6.0 Syntax ( ): Ergebnistyp	Beschreibung
FillEllipse	–	FillEllipse (x: Integery: Integer, a: Word,b: Word)	FillEllipse (x: Integery: Integer, a: Word,b: Word)	FillEllipse (x: Integery: Integer, a: Word,b: Word)	erzeugt eine mit dem aktuellen Muster gefüllte Ellipse
FillPattern	FillPattern (x1,y1,x2,y2,farbe: Integer)	–	–	–	füllt ein Rechteck auf dem Grafikbildschirm mit einem Muster
FillPoly	–	FillPoly (anzahl: Word,koordinatenfeld PointType)	FillPoly (anzahl: Word,koordinatenfeld PointType)	FillPoly (anzahl: Word,koordinatenfeld PointType)	zeichnet ein ausgefülltes Vieleck
FillScreen	FillScreen (farbe: Integer)	–	–	–	füllt das aktuelle Grafikfenster mit Farbe
FillShape	FillShape (x,y,füllfarbe, rahmen)	–	–	–	füllt einen Bereich ab der Koordinate *x,y*, der mit der Farbe *rahmen* umrandet ist, mit der Farbe *füllfarbe*
FindFirst	–	FindFirst (files: String,attribute: Byte, fileinfo: SearchRec)	FindFirst (files: String,attribute: Byte, fileinfo: SearchRec)	FindFirst (files: String,attribute: Byte, fileinfo: SearchRec)	sucht Verzeichniseinträge des anzugebenden Formats und speichert deren Informationen
FindNext	–	FindNext (fileinfos: SearchRec)	FindNext (fileinfos: SearchRec)	FindNext (fileinfos: SearchRec)	setzt eine durch FindFirst begonnene Suche fort
FloodFill	–	FloodFill (x: Integery: Integer, grenzfarbe: Word)	FloodFill (x: Integery: Integer, grenzfarbe: Word)	FloodFill (x: Integery: Integer, grenzfarbe: Word)	füllt abgegrenzte Bereiche mit dem aktuellen Muster aus
Flush	Flush (d: Text)	Flush (d: Text)	Flush (d: Text)	Flush (d: Text)	schreibt Textpuffer in Datei
Forwd	Forwd (weg: Integer)	–	–	–	bewegt Grafikzeiger um *weg* vorwärts
Frac	Frac (z: Integer/Real): Real	Frac (z: Real): Real	Frac (z: Real): Real	Frac (z: Real): Real	Nachkommaanteil der Zahl z
FreeMem	FreeMem (b: Pointer,l: Integer)	FreeMem (b: Pointer,l: Word)	FreeMem (b: Pointer,l: Word)	FreeMem (b: Pointer,l: Word)	gibt Speicherbereich der Variablen b mit Größe l auf Heap wieder frei
FSearch	–	–	FSearch (datei: PathStr,liste: String): PathStr	FSearch (datei: PathStr,liste: String): PathStr	liefert den Suchweg einer Datei aus einem Inhaltsverzeichnis
FSplit	–	–	FSplit (suchweg: PathStr,weg: DirStr, name: NameStr,suffix: ExtStr)	FSplit (suchweg: PathStr,weg: DirStr, name: NameStr,suffix: ExtStr)	teilt einen Dateinamen mit vollständigem Suchpfad auf in Suchweg, Dateiname und Suffix
GetArcCoords	–	GetArcCoords (kreisbogendaten: ArcCoordsType)	GetArcCoords (kreisbogendaten: ArcCoordsType)	GetArcCoords (kreisbogendaten: ArcCoordsType)	gibt die Daten der letzten Arc-Prozedur wieder
GetAspectRatio		GetAspectRatio (xwert: Word,ywert: Word)	GetAspectRatio (xwert: Word,ywert: Word)	GetAspectRatio (xwert: Word,ywert: Word)	liefert die Werte für das Höhen- und Seitenverhältnis des Bildschirms
GetBkColor	–	GetBkColor: Word	GetBkColor: Word	GetBkColor: Word	gibt die Kennziffer der aktuellen Hintergrundfarbe aus
GetCBreak	–	–	GetCBreak: Boolean	GetCBreak: Boolean	zeigt die Reaktion von DOS auf [Untbr]
GetColor	–	GetColor: Word	GetColor: Word	GetColor: Word	gibt die Kennziffer der aktuellen Zeichengrundfarbe aus
GetDate	–	GetDate (jahr: Word,monat: Word, tag: Word,wochentag: Word)	GetDate (jahr: Word,monat: Word, tag: Word,wochentag: Word)	GetDate (jahr: Word,monat: Word, tag: Word,wochentag: Word)	gibt das aktuelle Kalenderdatum des Rechners an
GetDefaultPalette	–	–	GetDefaultPalette (pal: Palettetype)	GetDefaultPalette (pal: Palettetype)	gibt die Daten der ursprünglichen Farbpalette wieder
GetDir	GetDir (laufwerk: Integer,dir: String)	GetDir (laufwerk: Byte,dir: String)	GetDir (laufwerk: Byte,dir: String)	GetDir (laufwerk: Byte,dir: String)	liefert Verzeichnis und Pfad des gewünschten Laufwerks
GetDriverName	–	–	GetDriverName: String	GetDriverName: String	gibt den Namen des aktuellen Grafiktreibers als String aus
GetEnv	–	–	GetEnv (umgebung: String): String	GetEnv (umgebung: String): String	liest eine Umgebungsvariable
GetFAttr	–	GetFAttr (datei,attribut: Word)	GetFAttr (datei,attribut: Word)	GetFAttr (datei,attribut: Word)	liefert die Attribute einer angegebenen Datei zurück
GetFillPattern	–	–	GetFillPattern (fuellmuster: FillPatternType)	GetFillPattern (fuellmuster: FillPatternType)	liefert das aktuelle Flächenfüllmuster als Bitmap
GetFillSettings	–	GetFillSettings (fuellmusterdaten: FillSettingsType)	GetFillSettings (fuellmusterdaten: FillSettingsType)	GetFillSettings (fuellmusterdaten: FillSettingsType)	liefert die Daten des aktuellen Füllmusters und die Farbe
GetFTime	–	GetFTime (datei,datumzeit: LongInt)	GetFTime (datei,datumzeit: LongInt)	GetFTime (datei,datumzeit: LongInt)	gibt Datum und Zeit der letzten Veränderung einer Datei an
GetGraphMode	–	GetGraphMode: Integer	GetGraphMode: Integer	GetGraphMode: Integer	gibt den Wert des momentanen Grafikmodus aus
GetImage	–	GetImage (x1: Integery1: Integer,x2: Integer, y2: Integer,puffer)	GetImage (x1: Integery1: Integer,x2: Integer, y2: Integer,puffer)	GetImage (x1: Integery1: Integer,x2: Integer, y2: Integer,puffer)	speichert einen rechteckigen Bildausschnitt in Puffervariable
GetIntVec	–	GetIntVec (intnummer: Byte,zeiger: Pointer)	GetIntVec (intnummer: Byte,zeiger: Pointer)	GetIntVec (intnummer: Byte,zeiger: Pointer)	holt die Adresse eines Interrupts in eine Zeigervariable
GetLineSettings	–	GetLineSettings (liniendaten: LineSettingsType)	GetLineSettings (liniendaten: LineSettingsType)	GetLineSettings (liniendaten: LineSettingsType)	ermittelt aktuelle Linienart, -muster und -breite
GetMaxColor	–	–	GetMaxColor: Word	GetMaxColor: Word	liefert aktuelle Farbpalettengröße

Befehl/Funktion	Turbo Pascal 3.0 Syntax ( ): Ergebnistyp	Turbo Pascal 4.0 Syntax ( ): Ergebnistyp	Turbo Pascal 5.0/5.5 Syntax ( ): Ergebnistyp	Turbo Pascal 6.0 Syntax ( ): Ergebnistyp	Beschreibung
GetMaxMode	–	–	GetMaxMode: Word	GetMaxMode: Word	ermittelt den Grafikmodus mit der höchstmöglichen Nummer, der vom Grafiktreiber noch darstellbar ist
GetMaxX	–	GetMaxX: Integer	GetMaxX: Integer	GetMaxX: Integer	liefert die maximale x-Koordinate des Bildschirms
GetMaxY	–	GetMaxY: Word	GetMaxY: Word	GetMaxY: Word	liefert die maximale y-Koordinate des Bildschirms
GetMem	GetMem(b: Pointer,I: Integer)	GetMem (b: Pointer,I: Word)	GetMem (b: Pointer,I: Word)	GetMem (b: Pointer,I: Word)	reserviert der Zeigervariablen b einen Speicherbereich der Größe I auf dem Heap
GetModeName	–	–	GetModeName (modenr: Integer): String	GetModeName (modenr: Integer): String	gibt den Namen eines Grafikmodus als String wieder
GetModeRange	–	–	GetModeRange (grafiktreiber: Integer, modusmin: Integer,modusmax: Integer)	GetModeRange (grafiktreiber: Integer, modusmin: Integer,modusmax: Integer)	ermittelt die mit einem bestimmten Grafiktreiber darzustellenden Grafikmodi
GetPalette	–	GetPalette (pal:PaletteType)	GetPalette (pal:PaletteType)	GetPalette (pal:PaletteType)	gibt die Daten der aktuellen Farbpalette wieder
GetPaletteSize	–	–	GetPaletteSize: Integer	GetPaletteSize: Integer	gibt die Anzahl der möglichen Farbpaletteneinträge an
GetPic	GetPic (puffer: Bel, x1y1,x2y2: Integer)	–	–	–	kopiert einen Grafikbereich in die Variable *puffer*
GetPixel	–	GetPixel (x: Integerx: Integer): Word	GetPixel (x: Integerx: Integer): Word	GetPixel (x: Integerx: Integer): Word	gibt Farbe eines Pixels an
GetPoint	GetPoint (x,y:Integer): Integer	–	–	–	liefert die Farbe der Bildschirmkoordinate *x,y*
GetTextSettings	–	GetTextSettings (textdaten: TextSettingsType)	GetTextSettings (textdaten: TextSettingsType)	GetTextSettings (textdaten: TextSettingsType)	liefert die Daten über das Schriftausgabeformat
GetTime	–	GetTime (stunde: Word,minute: Word, sekunde: Word,sek100: Word)	GetTime (stunde: Word,minute: Word, sekunde: Word,sek100: Word)	GetTime (stunde: Word,minute: Word, sekunde: Word, sek100: Word)	liefert die interne Uhrzeit des Computers
GetVerify	–	–	GetVerify: Boolean	GetVerify: Boolean	liefert die Information, ob bei Disketten-Schreiboperationen automatisch ihre Richtigkeit überprüft wird oder nicht
GetViewSettings	–	GetViewSettings (fensterdaten: ViewPortType)	GetViewSettings (fensterdaten: ViewPortType)	GetViewSettings (fensterdaten: ViewPortType)	holt Daten über das aktuelle Zeichenfenster
GetX	–	GetX: Integer	GetX: Integer	GetX: Integer	liefert den Wert der X-Koordinate der aktuellen Grafikcursor-Position
GetY	–	GetY: Integer	GetY: Integer	GetY: Integer	liefert den Wert der Y-Koordinate der aktuellen Grafikcursor-Position
GotoXY	GotoXY (x: Bytey: Byte)	GotoXY (x: Bytey: Byte)	GotoXY (x: Bytey: Byte)	GotoXY (x: Bytey: Byte)	setzt im Textbildschirm den Cursor an die Koordinate *x,y*
GraphBackground	GraphBackground (farbe: Integer)	–	–	–	verändert die Hintergrundfarbe des Bildschirms
GraphColorMode	GraphColorMode	–	–	–	aktiviert den 320*200-Punkte-Farbgrafikmodus
GraphDefaults	–	–	GraphDefaults	GraphDefaults	setzt alle Grafikparameter auf ihre Standardvorgabe zurück
GraphErrorMsg	GraphErrorMsg (fehlercode: Integer): String	GraphErrorMsg (fehlercode: Integer): String	GraphErrorMsg (fehlercode: Integer): String	GraphErrorMsg (fehlercode: Integer): String	liefert den zu einem Fehlercode des Grafikpakets zugehörigen Text
GraphMode	GraphMode	–	–	–	aktiviert den 320*200-Punkte-Schwarzweiß-Modus
GraphResult	–	GraphResult: Integer	GraphResult: Integer	GraphResult: Integer	liefert die Fehlernummer des zuletzt ausgeführten Grafikbefehls
GraphWindow	GraphWindow (x1y1,x2y2: Integer)	–	–	–	definiert ein Grafikfenster von den Koordinaten *x1y1* zu *x2y2*
Halt	Halt	Halt (beenden: Word) {beenden = Exitcode}	Halt (beenden: Word) {beenden = Exitcode}	Halt (beenden: Word) {beenden = Exitcode}	beendet ein Programm
Heading	Heading: Integer	–	–	–	liefert die Ausrichtung des Grafikcursors (0 bis 360 Grad)
Hi	Hi (z: Integer): Integer	Hi (z: Word/Integer): Byte	Hi (z: Word/Integer): Byte	Hi (z: Word/Integer): Byte	liefert höherwertiges Byte des Ausdrucks z
HideTurtle	HideTurtle	–	–	–	versteckt den Grafikcursor
HighVideo	–	HighVideo	HighVideo	HighVideo	setzt die Zeichenausgabe auf hohe Intensität
HiRes	HiRes	–	–	–	aktiviert den 640*200-Punkte-2-Farben-Modus
HiResColor	HiResColor (farbe: Integer)	–	–	–	ändert die Zeichenfarbe für den Hires-Modus
Home	Home	–	–	–	setzt den Grafikcursor an seine Ausgangsposition zurück

Befehl/Funktion	Turbo Pascal 3.0 Syntax ( ): Ergebnistyp	Turbo Pascal 4.0 Syntax ( ): Ergebnistyp	Turbo Pascal 5.0/5.5 Syntax ( ): Ergebnistyp	Turbo Pascal 6.0 Syntax ( ): Ergebnistyp	Beschreibung
ImageSize	–	ImageSize (x1: Integery1: Integer,x2: Integer, y2: Integer): Word	ImageSize (x1: Integery1: Integer,x2: Integer, y2: Integer): Word	ImageSize (x1: Integery1: Integer,x2: Integer, y2: Integer): Word	berechnet Speicherbedarf für Bildausschnitte
Inc	–	Inc (z: Integer)	Inc (z: Integer)	Inc (z: Integer)	liefert das Ergebnis von (*z* + 1)
Inc	–	Inc (z: Integer,zaehler: LongInt)	Inc (z: Integer,zaehler: LongInt)	Inc (z: Integer,zaehler: LongInt)	liefert das Ergebnis von (*z* + *zaehler*)
InitGraph	–	InitGraph (grafiktreiber: Integer,grafikmodus: Integer,treiberdir: String)	InitGraph (grafiktreiber: Integer,grafikmodus: Integer,treiberdir: String)	InitGraph (grafiktreiber: Integer,grafikmodus: Integer,treiberdir: String)	initialisiert die Grafikkarte und setzt einen Grafikmodus
Insert	Insert (s1: String,s2: String,ab: Integer)	Insert (s1: String,s2: String,ab: Integer)	Insert (s1: String,s2: String,ab: Integer)	Insert (s1: String,s2: String,ab: Integer)	fügt Zeichenkette *s1* ab der Stelle *ab* in den String *s2* ein
InsLine	InsLine	InsLine	InsLine	InsLine	fügt an der aktuellen Cursorposition eine Leerzeile ein
InstallUserDriver	–	–	InstallUserDriver (treibername: String, option: Pointer): Integer	InstallUserDriver (treibername: String, option: Pointer): Integer	installiert Grafiktreiber, die nicht von Turbo Pascal angeboten werden
InstallUserFont	–	–	InstallUserFont (zeichensatzname: String): Integer	InstallUserFont (zeichensatzname: String): Integer	installiert Vektorzeichensätze, die nicht von Turbo Pascal angeboten werden
Int	Int (z: Integer/Real): Real	Int (z: Real): Real	Int (z: Real): Real	Int (z: Real): Real	liefert die Vorkommastellen der Zahl z
Intr	Intr (intrnummer: Byte,regists: Registers)	Intr (intrnummer: Byte,regists: Registers)	Intr (intrnummer: Byte,regists: Registers)	Intr (intrnummer: Byte,regists: Registers)	bewirkt Softwareinterrupt
IOResult	IOResult: Integer	IOResult: Word	IOResult: Word	IOResult: Word	gibt den Fehlercode der letzten Ein-/Ausgabeoperation wieder
Keep	–	Keep (exitcode: Word)	Keep (exitcode: Word)	Keep (exitcode: Word)	erzeugt ein speicherresidentes Programm
KeyPressed	KeyPressed: Boolean	KeyPressed: Boolean	KeyPressed: Boolean	KeyPressed: Boolean	überprüft, ob eine Taste gedrückt wurde
Length	Length (w: String): Integer	Length (w: String): Integer	Length (w: String): Integer	Length (w: String): Integer	liefert die Anzahl der Zeichen einer Zeichenkette *w*
Line	–	Line (x1: Integery1: Integer,x2: Integer, y2: Integer)	Line (x1: Integery1: Integer,x2: Integer, y2: Integer)	Line (x1: Integery1: Integer,x2: Integer, y2: Integer)	zieht zwischen zwei Punkten eine Linie
LineRel	–	LineRel (delta__x: Integer,delta__y: Integer)	LineRel (delta__x: Integer,delta__y: Integer)	LineRel (delta__x: Integer,delta__y: Integer)	zeichnet Linie relativ zur momentanen Position der Cursorposition
LineTo	–	LineTo (x: Integery: Integer)	LineTo (x: Integery: Integer)	LineTo (x: Integery: Integer)	zieht von der aktuellen Cursorposition aus zu einem anzugebenden Punkt eine Linie
Ln	Ln (z: Integer/Real): Real	Ln (z: Real): Real	Ln (z: Real): Real	Ln (z: Real): Real	natürlicher Logarithmus von *z*
Lo	Lo (z: Integer): Integer	Lo (z: Word/Integer): Byte	Lo (z: Word/Integer): Byte	Lo (z: Word/Integer): Byte	liefert niederwertiges Byte des Ausdrucks z
LowVideo	LowVideo	LowVideo	LowVideo	LowVideo	setzt die Zeichenausgabe auf niedrige Intensität
Mark	Mark (b: Pointer)	Mark (b: Pointer)	Mark (b: Pointer)	Mark (b: Pointer)	weist der Zeigervariablen b die Adresse der Heapspitze zu
MaxAvail	–	MaxAvail: LongInt	MaxAvail: LongInt	MaxAvail: LongInt	liefert die Speicherkapazität des größten freien Blocks auf dem Heap
MemAvail	MemAvail: Integer	MemAvail: LongInt	MemAvail: LongInt	MemAvail: LongInt	liefert den gesamten freien Speicherplatz auf dem Heap
MkDir	MkDir (name: String)	MkDir (name: String)	MkDir (name: String)	MkDir (name: String)	richtet ein neues Unterverzeichnis mit dem Namen *name* ein
Move	Move (quel,ziel,anz: Integer)	Move (quel,ziel,anz: Word)	Move (quel,ziel,anz: Word)	Move (quel,ziel,anz: Word)	kopiert *anz* Byte eines Speicherbereichs ab *quel* in einen anderen ab *ziel*
MoveRel	–	MoveRel (delta__x: Integer,delta__y: Integer)	MoveRel (delta__x: Integer,delta__y: Integer)	MoveRel (delta__x: Integer,delta__y: Integer)	verschiebt den Grafikcursor relativ zur momentanen Position in x- und y-Richtung
MoveTo	–	MoveTo (x: Integery: Integer)	MoveTo (x: Integery: Integer)	MoveTo (x: Integery: Integer)	setzt den Grafikcursor auf einen festzulegenden Punkt
MsDos	–	MsDos (regists:Registers)	MsDos (regists:Registers)	MsDos (regists:Registers)	führt eine DOS-Funktion aus
New	–	–	New (p: Pointer, Konstrukt)	New (p: Pointer, Konstrukt)	belegt Bereich auf Heap
New	New (p: Pointer)	New (p: Pointer)	New (p: Pointer)	New (p: Pointer)	legt neue Zeigervariable *p* an
NormVideo	NormVideo	NormVideo	NormVideo	NormVideo	setzt das Textattribut auf seinen Ausgangswert
NoSound	NoSound	NoSound	NoSound	NoSound	schaltet den Tongenerator ab
NoWrap	NoWrap	–	–	–	verhindert, daß der Grafikcursor den Grafikbereich verläßt und auf der gegenüberliegenden Seite erscheint
Odd	Odd (z: Integer): Boolean	Odd (z: LongInt): Boolean	Odd (z: LongInt): Boolean	Odd (z: LongInt): Boolean	prüft, ob Zahl *z* ungerade ist
Ofs	Ofs (b): Integer	Ofs (b): Word	Ofs (b): Word	Ofs (b): Word	gibt den Offset-Teil der Adresse von b zurück
Ord	Ord (w): Integer	Ord (w): LongInt	Ord (w): LongInt	Ord (w): LongInt	liefert Ordinalzahl der Variablen *w*
OutText	–	OutText (texte: String)	OutText (texte: String)	OutText (texte: String)	gibt einen Text an der Position des Grafikcursors aus

Befehl/Funktion	Turbo Pascal 3.0 Syntax ( ): Ergebnistyp	Turbo Pascal 4.0 Syntax ( ): Ergebnistyp	Turbo Pascal 5.0/5.5 Syntax ( ): Ergebnistyp	Turbo Pascal 6.0 Syntax ( ): Ergebnistyp	Beschreibung
OutTextXY	–	OutTextXY (x: Integery: Integer,texte: String)	OutTextXY (x: Integery: Integer,texte: String)	OutTextXY (x: Integery: Integer,texte: String)	texte bei x,y ausgeben
OvrClearBuf	OvrClearBuf	OvrClearBuf	–	–	löscht den gesamten Inhalt des Overlay-Puffers
OvrDrive	OvrDrive (laufwerk: Integer)	–	–	–	definiert das Laufwerk für die Overlays
OvrGetBuf	–	–	OvrGetBuf: LongInt	OvrGetBuf: LongInt	liest die momentan gesetzte Puffergröße des Overlay-Puffers
OvrGetRetry	–	–	OvrGetRetry: LongInt {ab Version 5.5}	OvrGetRetry: LongInt	liefert die Größe des Overlay-Bewährungsbereichs
OvrInit	–	–	OvrInit (dateiname: String)	OvrInit (dateiname: String)	initialisiert Overlay-System
OvrInitEMS	–	–	OvrInitEMS	OvrInitEMS	initialisiert das Extended Memory-System (EMS)
OvrPath	OvrPath (pfad: String)	–	–	–	definiert den Dateipfad für die Overlays
OvrSetBuf	–	–	OvrSetBuf (groesse: LongInt)	OvrSetBuf (groesse: LongInt)	setzt die Größe des Overlay-Puffers fest
OvrSetRetry	–	–	OvrSetRetry (groesse: LongInt) {ab Version 5.5}	OvrSetRetry (groesse: LongInt)	setzt die Größe des Overlay-Bewährungsbereichs fest
PackTime	–	PackTime (datumzeit: Datetime,zeit: LongInt)	PackTime (datumzeit: Datetime,zeit: LongInt)	PackTime (datumzeit: Datetime,zeit: LongInt)	packt Einzelkomponenten von datumzeit in LongIntTyp zeit
Palette	Palette (palette: Integer)	–	–	–	ändert die Farbpalette für den Farbgrafikmodus
ParamCount	–	ParamCount: Word	ParamCount: Word	ParamCount: Word	liefert die Anzahl der Kommandozeilen-Parameter
ParamStr	–	ParamStr (num: Word): String	ParamStr (num: Word): String	ParamStr (num: Word): String	liefert den Kommandozeilen-Parameter Nummer num
Pattern	Pattern (muster: Array [0..7] of Byte)	–	–	–	definiert das Füllmuster für die Prozedur FillPattern
PenDown	PenDown	–	–	–	PenDown
PenUp	PenUp	–	–	–	deaktiviert den Grafikcursor
Pi	–	Pi: Real	Pi: Real	Pi: Real	Zahl Pi (3,14...)
PieSlice	–	PieSlice (x: Integery: Integer,start: Word, ende: Word,radius: Word)	PieSlice (x: Integery: Integer,start: Word, ende: Word,radius: Word)	PieSlice (x: Integery: Integer,start: Word, ende: Word,radius: Word)	zeichnet ein ausgefülltes »Tortenstück«
Plot	Plot (x,y,farbe: Integer)	–	–	–	zeichnet einen Punkt in der gewünschten Farbe an der Koordinate *x,y*
Pos	Pos (stueck: String,gesamt: String): Integer	Pos (stueck: String,gesamt: String): Byte	Pos (stueck: String,gesamt: String): Byte	Pos (stueck: String,gesamt: String): Byte	liefert Position im String *gesamt*, an der eine Zeichenkette *stueck* beginnt
Pred	Pred (Ordinaltyp): Ordinaltyp	Pred (Ordinaltyp): Ordinaltyp	Pred (Ordinaltyp): Ordinaltyp	Pred (Ordinaltyp): Ordinaltyp	zeigt Vorgänger des Arguments
Ptr	Ptr (Seg:Integer,Ofs: Integer): Pointer	Ptr (Seg: Word,Ofs: Word): Pointer	Ptr (Seg: Word,Ofs: Word): Pointer	Ptr (Seg: Word,Ofs: Word): Pointer	faßt Seg und Ofs zu einem Zeiger zusammen
PutImage	–	PutImage (x: Word,y: Word,puffer,mode: Word)	PutImage (x: Word,y: Word,puffer,mode: Word)	PutImage (x: Word,y: Word,puffer,mode: Word)	holt aus Puffervariable bitweise Daten und kopiert sie in rechteckigen Bildausschnitt
PutPic	PutPic (puffer: Bel,x,y: Integer)	–	–	–	kopiert einen Grafikbereich aus *puffer* auf den Grafikbildschirm
PutPixel	–	PutPixel (x: Integery: Integer,farbe: Word)	PutPixel (x: Integery: Integer,farbe: Word)	PutPixel (x: Integery: Integer,farbe: Word)	setzt einen Punkt auf den Bildschirm
Random	Random (z: Integer): Integer	Random (z: Word): Word	Random (z: Word): Word	Random (z: Word): Word	erzeugt Zufallszahl zw.[0..z[
Random	Random: Real	Random: Real	Random: Real	Random: Real	erzeugt Zufallszahl zw.[0..1[
Randomize	Randomize	Randomize	Randomize	Randomize	initialisiert den Zufallsgenerator neu
Read	Read (d,var1 [,var2,...,varn])	Read (d,var1 [,var2,...,varn])	Read (d,var1 [,var2,...,varn])	Read (d,var1 [,var2,...,varn])	liest Komponente der Variable(n) *var1,...* aus der Datei *d*
Read	Read (var1 [,var2,...,varn])	Read (var1 [,var2,...,varn])	Read (var1 [,var2,...,varn])	Read (var1 [,var2,...,varn])	liest Wert(e) der Variable(n) *var1,...*
ReadKey	–	ReadKey: Char	ReadKey: Char	ReadKey: Char	liest ein Zeichen von der Tastatur ohne Bildschirmausgabe
ReadLn	ReadLn (d,var1 [,var2,...,varn]){ ausgenommen typisierte Dateien}	ReadLn (d,var1 [,var2,...,varn]){ ausgenommen typisierte Dateien}	ReadLn (d,var1 [,var2,...,varn]){ ausgenommen typisierte Dateien}	ReadLn (d,var1 [,var2,...,varn]){ ausgenommen typisierte Dateien}	Springt zum Anfang der nächsten Zeile, bei gleichem Lesevorgang wie bei Read
Rectangle	–	Rectangle (x1: Integery1: Integer, x2: Integery2: Integer)	Rectangle (x1: Integery1: Integer, x2: Integery2: Integer)	Rectangle (x1: Integery1: Integer, x2: Integery2: Integer)	zeichnet ein Rechteck um die angegebenen Koordinaten
RegisterBGIDriver	–	–	RegisterBGIDriver (treiber: Pointer): Integer	RegisterBGIDriver (treiber: Pointer): Integer	registriert einen als Obj-Datei eingebundenen Grafiktreiber
RegisterBGIFont	–	–	RegisterBGIFont (font: Pointer): Integer	RegisterBGIFont (font: Pointer): Integer	registriert einen als Obj-Datei eingebundenen Vektorzeichensatz

Befehl/Funktion	Turbo Pascal 3.0 Syntax ( ): Ergebnistyp	Turbo Pascal 4.0 Syntax ( ): Ergebnistyp	Turbo Pascal 5.0/5.5 Syntax ( ): Ergebnistyp	Turbo Pascal 6.0 Syntax ( ): Ergebnistyp	Beschreibung
Release	Release (b: Pointer)	Release (b: Pointer)	Release (b: Pointer)	Release (b: Pointer)	setzt Heap-Zeiger auf in b enthaltene alte Adresse und Größe zurück
Rename	Rename (d,neuname: String)	Rename (d,neuname: String)	Rename (d,neuname: String)	Rename (d,neuname: String)	benennt externe Datei *d* um
Reset	–	Reset (d,recgroeße: Word)	Reset (d,recgroeße: Word)	Reset (d,recgroeße: Word)	öffnet die Datei *d*, und setzt den Positionszeiger auf Anfang, und legt bei untypisierten Dateien über den Parameter recgroeße die Record-Größe fest
Reset	Reset (d)	Reset (d)	Reset (d)	Reset (d)	öffnet die Datei *d*, und setzt den Positionszeiger für den Lesezugriff auf den Dateianfang
RestoreCrt	–	RestoreCrt	RestoreCrt	RestoreCrt	aktiviert den ursprünglichen Videomodus
RestoreCrtMode	–	RestoreCrtMode	RestoreCrtMode	RestoreCrtMode	reaktiviert den vor dem Start des Grafikpakets aktiven Videomodus
Retype	Retype (x,Text): Text	–	–	–	wandelt *x* in Text; *x* kann jeden beliebigen Typ annehmen
Rewrite	–	Rewrite (d,recgroeße: Word)	Rewrite (d,recgroeße: Word)	Rewrite (d,recgroeße: Word)	erzeugt und öffnet eine Datei *d*, und setzt den Positionszeiger auf Anfang, und legt bei untypisierten Dateien über den Parameter *recgroeße* die Record-Größe fest
Rewrite	Rewrite (d)	Rewrite (d)	Rewrite (d)	Rewrite (d)	erzeugt und öffnet neue Dateien
RmDir	RmDir (name: String)	RmDir (name: String)	RmDir (name: String)	RmDir (name: String)	löscht ein Unterverzeichnis
Round	Round (z: Real): Integer	Round (z: Real): LongInt	Round (z: Real): LongInt	Round (z: Real): LongInt	Rundet die Realzahl *z* auf einen ganzzahligen Wert
RunError	–	–	RunError (nummer: Word)	RunError (nummer: Word)	erzeugt einen Laufzeitfehler, der das Programm abbricht
RunError	–	–	RunError	RunError	erzeugt einen Laufzeitfehler, der das Programm abbricht
Sector	–	–	Sector (x,y: Integer,start,ende,a,b: Word)	Sector(x,y:Integer,start,ende,a,b: Word)	zeichnet ein elliptisches ausgefülltes Bogenstück
Seek	Seek (bd,element: Integer)	Seek (bd,element: LongInt)	Seek (bd,element: LongInt)	Seek (bd,element: LongInt)	positioniert Dateizeiger in der Binärdatei *bd* auf ein bestimmtes Element, wobei *element* die Nummer des Elements angibt
SeekEof	SeekEof (d: Text): Boolean	SeekEof (d: Text): Boolean	SeekEof (d: Text): Boolean	SeekEof (d: Text): Boolean	kontrolliert, ob zwischen aktueller Dateizeigerposition und Eof-Marke lesbare Daten stehen
SeekEoln	SeekEoln (d: Text): Boolean	SeekEoln (d: Text): Boolean	SeekEoln (d: Text): Boolean	SeekEoln (d: Text): Boolean	kontrolliert, ob zwischen aktueller Dateizeigerposition und Eoln-Marke lesbare Daten sind
Seg	Seg (b): Integer	Seg (b): Word	Seg (b): Word	Seg (b): Word	gibt den Segment-Teil der Adresse von b zurück
SetActivePage	–	SetActivePage (seitennr: Word)	SetActivePage (seitennr: Word)	SetActivePage (seitennr: Word)	legt die Grafikseite fest, auf der Zeichenbefehle ausgeführt werden
SetAllPalette	–	SetAllPalette (farbdaten: PaletteType)	SetAllPalette (farbdaten: PaletteType)	SetAllPalette (farbdaten: PaletteType)	setzt alle Palettenfarben neu
SetAspectRatio	–	–	SetAspectRatio (x__wert: Word,y__wert: Word)	SetAspectRatio (x__wert: Word,y__wert: Word)	legt das Höhen-/Seitenverhältnis des Bildschirms fest
SetBkColor	–	SetBkColor (farbe: Word)	SetBkColor (farbe: Word)	SetBkColor (farbe: Word)	stellt die Hintergrundfarbe des Bildschirms ein
SetCBreak	–	–	SetCBreak (flag: Boolean)	SetCBreak (flag: Boolean)	legt fest, ob DOS immer oder nur bei Ein-/Ausgabeoperationen auf [Ctrl-Break] reagiert
SetColor	–	SetColor (farbe: Word)	SetColor (farbe: Word)	SetColor (farbe: Word)	stellt neue Zeichenfarbe ein
SetDate	–	SetDate (jahr: Word,monat: Word,tag: Word, wochentag: Word)	SetDate (jahr: Word,monat: Word,tag: Word, wochentag: Word)	SetDate (jahr: Word,monat: Word,tag: Word, wochentag: Word)	stellt das Systemdatum des Computers ein
SetFAttr	–	SetFAttr (datei,attribut: Byte)	SetFAttr (datei,attribut: Byte)	SetFAttr (datei,attribut: Byte)	setzt Attribute einer Datei
SetFillPattern	–	SetFillPattern (muster: Fillpatterntype, farbe: Word)	SetFillPattern (muster: Fillpatterntype, farbe: Word)	SetFillPattern (muster: Fillpatterntype, farbe: Word)	definiert Muster für aufzufüllende Flächen
SetFillStyle	–	SetFillStyle (muster: Word,farbe: Word)	SetFillStyle (muster: Word,farbe: Word)	SetFillStyle (muster: Word,farbe: Word)	setzt ein vordefiniertes Flächenmuster
SetFTime	–	SetFTime (datei,zeit: LongInt)	SetFTime (datei,zeit: LongInt)	SetFTime (datei,zeit: LongInt)	setzt Datums- und Uhrzeitangabe direkt auf eine Datei
SetGraphBufSize	–	–	SetGraphBufSize (groesse: Word)	SetGraphBufSize (groesse: Word)	bestimmt die Größe des Puffers für Polygone und Flächenfüllungen
SetGraphMode	–	SetGraphMode (modus: Integer)	SetGraphMode (modus: Integer)	SetGraphMode (modus: Integer)	schaltet in einen Grafikmodus um und löscht den Bildschirm
SetHeading	SetHeading (richtung: Integer)	–	–	–	setzt den Grafikcursor auf eine Ausrichtung zwischen 0 und 360 Grad

Befehl/Funktion	Turbo Pascal 3.0 Syntax ( ): Ergebnistyp	Turbo Pascal 4.0 Syntax ( ): Ergebnistyp	Turbo Pascal 5.0/5.5 Syntax ( ): Ergebnistyp	Turbo Pascal 6.0 Syntax ( ): Ergebnistyp	Beschreibung
SetIntVec	–	SetIntVec (intnummer: Byte,zeiger: Pointer)	SetIntVec (intnummer: Byte,zeiger: Pointer)	SetIntVec (intnummer: Byte,zeiger: Pointer)	stellt einen Interruptvektor des Systems neu ein
SetLineStyle	–	SetLineStyle (art: Word,muster: Word, breite: Word)	SetLineStyle (art: Word,muster: Word, breite: Word)	SetLineStyle (art: Word,muster: Word, breite: Word)	legt die aktuelle Linienart und -breite fest
SetPalette	–	SetPalette (palnr: Word,farbe: Shortint)	SetPalette (palnr: Word,farbe: Shortint)	SetPalette (palnr: Word,farbe: Shortint)	verändert bestimmten Eintrag in der aktuellen Farbpalette
SetPenColor	SetPenColor (farbe: Integer)	–	–	–	setzt die aktuelle Farbe des Grafikcursors
SetPosition	SetPosition (x,y: Integer)	–	–	–	bewegt den Grafikcursor an eine neue Position
SetRGBPalette	–	–	SetRGBPalette (farbnummer: Integer,rot: Integer, gruen: Integer,blau: Integer)	SetRGBPalette (farbnummer: Integer,rot: Integer, gruen: Integer,blau: Integer)	setzt neuen Farbpaletteneintrag für IBM-Adapter 8514 und für VGA-Karten im Modus mit 256 Farben
SetTextBuf	–	SetTextBuf (d,puffer,groesse: Word)	SetTextBuf (d,puffer,groesse: Word)	SetTextBuf (d,puffer,groesse: Word)	ändert die Puffergröße der Textdateivariablen d auf den Wert des Parameters groesse
SetTextBuf	–	SetTextBuf (d,puffer)	SetTextBuf (d,puffer)	SetTextBuf (d,puffer)	ändert die Puffergröße der Textdateivariablen d auf die Größe der Variablen puffer
SetTextJustify	–	SetTextJustify (horizontal: Word, vertikal: Word)	SetTextJustify (horizontal: Word, vertikal: Word)	SetTextJustify (horizontal: Word, vertikal: Word)	definiert die Schriftlage bezüglich der Grafikcursorposition bei Textausgaben durch OutText(XY)
SetTextStyle	–	SetTextStyle (zeichensatz: Word, richtung: Word, groesse: Word)	SetTextStyle (zeichensatz: Word, richtung: Word,groesse: Word)	SetTextStyle (zeichensatz: Word, richtung: Word,groesse: Word)	legt Zeichensatz, Richtung und Größe der Zeichen bei Textausgaben durch OutText(XY) fest
SetTime	–	SetTime (stunde: Word,minute: Word, sekunde: Word,sek100: Word)	SetTime (stunde: Word,minute: Word, sekunde: Word,sek100: Word)	SetTime (stunde: Word,minute: Word, sekunde: Word,sek100: Word)	stellt die Systemzeit Ihres Computers ein
SetUserCharSize	–	–	SetUserCharSize (breiter: Word,schmaeler: Word, hoeher: Word,tiefer: Word)	SetUserCharSize (breiter: Word,schmaeler: Word, hoeher: Word,tiefer: Word)	verändert Zeichen aus Grafikzeichensätzen nach Höhe und Breite
SetVerify	–	SetVerify (flag: Boolean)	SetVerify (flag: Boolean)	SetVerify (flag: Boolean)	legt fest, ob Diskettenschreiboperationen auf Fehler überprüft werden
SetViewPort	–	SetViewPort (x1: Integery1: Integer,x2: Integer, y2: Integer,clip: Boolean)	SetViewPort (x1: Integery1: Integer,x2: Integer, y2: Integer,clip: Boolean)	SetViewPort (x1: Integery1: Integer,x2: Integer, y2: Integer,clip: Boolean)	setzt die Koordinaten eines Zeichenfensters
SetVisualPage	–	SetVisualPage (seitennr: Word)	SetVisualPage (seitennr: Word)	SetVisualPage (seitennr: Word)	bestimmt Grafikspeicherseite
SetWriteMode	–	–	SetWriteMode (modus: Integer)	SetWriteMode (modus: Integer)	bestimmt, ob Linienzeichenoperationen überschreibend oder verknüpfend ausgeführt werden
ShowTurtle	ShowTurtle	–	–	–	zeigt den Grafikcursor
Sin	Sin (z: Integer/Real): Real	Sin (z: Real): Real	Sin (z: Real): Real	Sin (z: Real): Real	liefert den Sinus von *z*
SizeOf	SizeOf (v): Word	SizeOf (v): Word	SizeOf (v): Word	SizeOf (v): Word	gibt an wieviel Bytes *v* an Speicherplatz belegt; v ist ist ein Typenbezeichner oder ein beliebiger Variablentyp
Sound	Sound (frequenz: Integer)	Sound (frequenz: Word)	Sound (frequenz: Word)	Sound (frequenz: Word)	dient zur Tonerzeugung mit dem internen Tongenerator
SoundOff	siehe NoSound	–	–	–	deaktiviert den Tongenerator
SPtr	–	SPtr: Word	SPtr: Word	SPtr: Word	liefert den aktuellen Wert des Stackzeigers
Sqr	Sqr (z: Integer/Real): Integer/Real	Sqr (z: Integer/Real): Integer/Real	Sqr (z: Integer/Real): Integer/Real	Sqr (z: Integer/Real): Integer/Real	liefert die Quadratzahl von *z*
Sqrt	Sqrt (z: Integer/Real): Real	Sqrt (z: Real): Real	Sqrt (z: Real): Real	Sqrt (z: Real): Real	liefert die Quadratwurzel von *z*
SSeg	SSeg: Integer	SSeg: Word	SSeg: Word	SSeg: Word	liefert die Adresse des Stacksegments
Str	Str (z: Integer/Real,w: String)	Str (z: Integer/Real,w: String)	Str (z: Integer/Real,w: String)	Str (z: Integer/Real,w: String)	wandelt den Zahlenwert *z* in eine Zeichenkette *w* um
Succ	Succ (Ordinaltyp): Ordinaltyp	Succ (Ordinaltyp): Ordinaltyp	Succ (Ordinaltyp): Ordinaltyp	Succ (Ordinaltyp): Ordinaltyp	gibt Nachfolger des Arguments zurück
Swap	–	Swap (z: Word): Word	Swap (z: Word): Word	Swap (z: Word): Word	tauscht höherwertiges Byte von z mit dem niederwertigen
Swap	Swap (z: Integer): Integer	Swap (z: Integer): Integer	Swap (z: Integer): Integer	Swap (z: Integer): Integer	tauscht höherwertiges Byte von z mit dem niederwertigen
SwapVectors	–	–	SwapVectors	SwapVectors	vertauscht die Interruptvektoren des Betriebssystems mit denen des Unitsystems und umgekehrt
TextBackground	TextBackground (farbe: Integer)	TextBackground (farbe: Byte)	TextBackground (farbe: Byte)	TextBackground (farbe: Byte)	bestimmt die Hintergrundfarbe zu Textausgaben
TextColor	TextColor (farbe: Integeryte)	TextColor (farbe: Byte)	TextColor (farbe: Byte)	TextColor (farbe: Byte)	stellt die Farbe der Schriftzeichen für Textausgaben ein
TextHeight	–	TextHeight (textstring: String): Word	TextHeight (textstring: String): Word	TextHeight (textstring: String): Word	gibt die Höhe eines Textstrings in Pixel an
TextMode	TextMode (modus: Integer)	TextMode (modus: Word)	TextMode (modus: Word)	TextMode (modus: Word)	setzt bestimmten Textmodus

Befehl/Funktion	Turbo Pascal 3.0 Syntax ( ): Ergebnistyp	Turbo Pascal 4.0 Syntax ( ): Ergebnistyp	Turbo Pascal 5.0/5.5 Syntax ( ): Ergebnistyp	Turbo Pascal 6.0 Syntax ( ): Ergebnistyp	Beschreibung
TextWidth	–	TextWidth (textstring: String): Word	TextWidth (textstring: String): Word	TextWidth (textstring: String): Word	gibt die Breite eines Textstrings in Pixel an
Trunc	–	Trunc (z: Real): LongInt	Trunc (z: Real): LongInt	Trunc (z: Real): LongInt	gibt Vorkommateil der Zahl *z* als Integer-Wert aus
Truncate (d)	Truncate (d)	Truncate (d)	Truncate (d)	Truncate (d)	schneidet die offene Datei d an der aktuellen Position des Dateizeigers ab
TurnLeft	TurnLeft (winkel: Integer)	–	–	–	dreht den Grafikcursor um *winkel* Grad nach links
TurnRight	TurnRight (winkel: Integer)	–	–	–	dreht den Grafikcursor um *winkel* Grad nach rechts
TurtleDelay	TurtleDelay (delta__t)	–	–	–	verzögert die Turtle-Bewegung um *delta__t* Millisekunden
TurtleHome	siehe Home	–	–	–	bringt den Grafikcursor in die Ausgangsposition (Koordinate 0,0)
TurtleThere	TurtleThere: Boolean	–		–	überprüft, ob Grafikcursor sichtbar ist
TurtleWindow	Turtle Window (x,y,breite,höhe: Integer)	–	–	–	definiert ein neues Grafikfenster für die Turtle-Grafik
TypeOf	–	–	–	TypeOf (objekt)	liefert den Typ des Operanden; KOOP liefert einen Zeiger auf die VMT eines Objekts
UnPackTime	–	UnPackTime (zeit: LongInt,datumzeit: Datetime)	UnPackTime (zeit: LongInt,datumzeit: Datetime)	UnPackTime (zeit: LongInt,datumzeit: Datetime)	entpackt Variable zeit in ihre Komponenten
UpCase	UpCase (c: Char): Char	UpCase (c: Char): Char	UpCase (c: Char): Char	UpCase (c: Char): Char	wandelt einen Kleinbuchstaben c in einen Großbuchstaben um
Val	Val (w: String,z: Integer/real,k: Integer)	Val (w: String,z: Integer/real,k: Integer)	Val (w: String,z: Integer/real,k: Integer)	Val (w: String,z: Integer/real,k: Integer)	wandelt den aus Ziffern bestehenden String *w* in einen Zahlenwert *z* um. Variable *k* enthält String-Position eines auftretenden Fehlers
WhereX	WhereX: Integer	WhereX: Byte	WhereX: Byte	WhereX: Byte	gibt die Spalte der aktuellen Cursorposition an
WhereY	WhereY: Integer	WhereY: Byte	WhereY: Byte	WhereY: Byte	gibt die Zeile der aktuellen Cursorposition an
Window	Window (x1,y1,x2,y2: Integer)	Window (x1: Byte,y1: Byte,x2: Byte,y2: Byte)	Window (x1: Byte,y1: Byte,x2: Byte,y2: Byte)	Window (x1: Byte,y1: Byte,x2: Byte,y2: Byte)	legt Bereich des aktuellen Textfensters am Bildschirm fest
Wrap	Wrap	–	–	–	gestattet, daß der Grafikcursor den Grafikbereich verläßt und auf der gegenüberliegenden Seite erscheint
Write	Write (d: Text,var1 [,var2,...,varn])	Write (d: Text,var1 [,var2,...,varn])	Write (d: Text,var1 [,var2,...,varn])	Write (d: Text,var1 [,var2,...,varn])	schreibt die Komponente(n) der Variable(n) *var1,...* in die Datei *d*
Write	Write (var1 [,var2,...,varn])	Write (var1 [,var2,...,varn])	Write (var1 [,var2,...,varn])	Write (var1 [,var2,...,varn])	gibt den Inhalt der Variable(n) *var1,...* am Bildschirm aus
WriteLn	WriteLn ([d: Text,]var1 [,var2,...,varn]) K{ausgenommen typisierte Dateien}	WriteLn ([d: Text,]var1 [,var2,...,varn]) K{ausgenommen typisierte Dateien}	WriteLn ([d: Text,]var1 [,var2,...,varn]) K{ausgenommen typisierte Dateien}	WriteLn ([d: Text,]var1 [,var2,...,varn]) K{ausgenommen typisierte Dateien}	führt nach dem Schreibvorgang (siehe *Write*) zusätzlich einen Zeilenvorschub aus
XCor	XCor: Integer	–	–	–	gibt die x-Koordinate des Grafikcursors zurück
YCor	YCor: Integer	–	–	–	gibt die y-Koordinate des Grafikcursors zurück

GREENPEACE



18,-DM 145 öS 18,-sfr 55,-Fmk

WinDOS

Nr. 1
1991
1. Jahrg

ANZEIGE NEU

DMV-Verlag

1´91 Erstausgabe

Das Windows-Magazin der DOS International

Lieber Windows-Anwender,

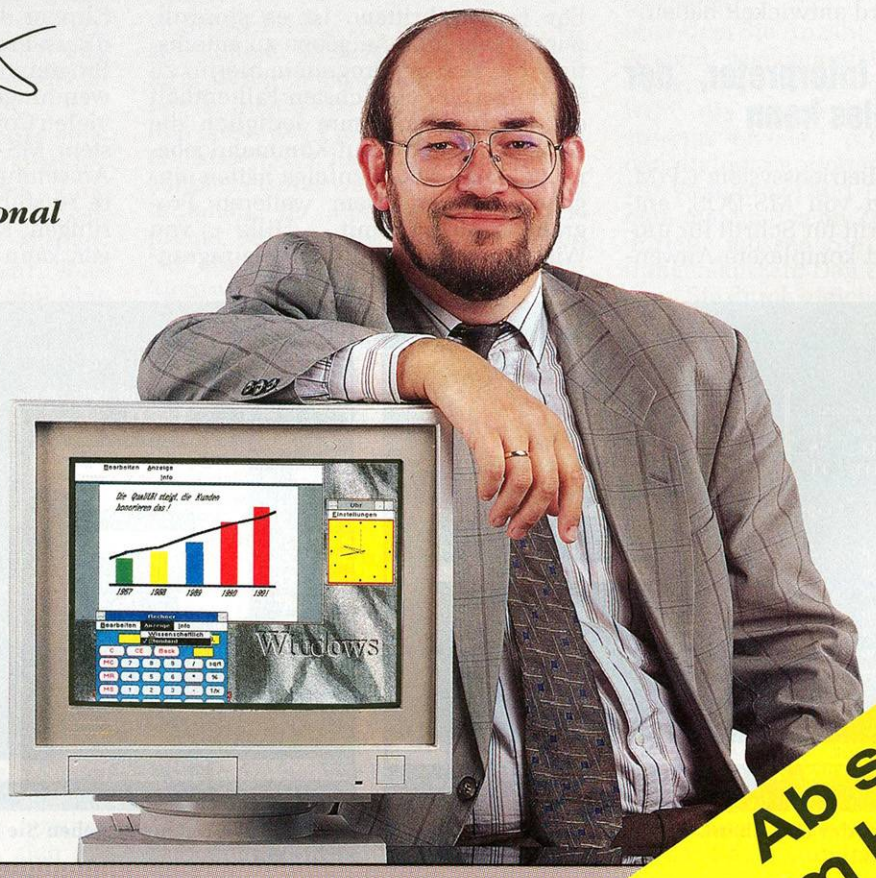
in unserem neuen Sonderheft bieten wir Ihnen qualifizierte und umfassende Informationen zum brandaktuellen Windows-Thema.

In der WinDOS finden Sie Werkstätten, Tabellen und natürlich jede Menge Tips & Tricks. Wir zeigen Ihnen, wie Sie mit der richtigen Software unter Windows professionell aussehende Applikationen programmieren, mit Makros zaubern und warum Sie kein teures Software-Entwicklungssystem für Windows brauchen.

Holen Sie sich also jetzt das Windows-Magazin, um Windows besser zu verstehen, zu beherrschen und zu programmieren.

Viel Erfolg mit Windows und natürlich WinDOS wünscht Ihnen
Ihr

Volker Everts
Chefredakteur
DOS International



Ab sofort im Handel

dBase und Clipper

Kontrollierte Datenflut

Ohne Hilfsmittel wachsen täglich neu eintreffende Informationen zum unüberschaubaren Chaos. Karteikarten und Notizzettel versagen, wenn sie eine kritische Menge überschreiten, oder wenn Sie nach zahlreichen Kriterien suchen müssen. Dieser Beitrag soll Ordnung in dem Handwerkszeug schaffen, mit dem Sie Überblick im Datenstrom behalten. Dabei greifen Sie auf grundlegende Bearbeitungstechniken zurück.

Das Informationszeitalter bringt es mit sich, daß eine immer größere Datenmenge Sie bedrängt. Ob in Wissenschaft und Forschung, bei Behörden, in der Industrie, in großen Firmen und Unternehmen oder auch im privaten Bereich: Überall gilt es, möglichst schnell und gezielt auf Informationen zugreifen und diese nach den jeweiligen Bedingungen miteinander verknüpfen zu können. Zudem können Sie als Anwender viele dieser Daten mehrfach verwenden und durch geringfügige Veränderungen leicht aktualisieren.

In diesem Artikel erfahren Sie die Besonderheiten der Datenbanksprache »dBase« und des Compilers »Clipper«, da sich beide Programme gewissermaßen zum Standard entwickelt haben.

dBase, ein Interpreter, der schon fast alles kann

Schon auf dem Betriebssystem CP/M, einem Vorfahren von MS-DOS, entwickelt und Schritt für Schritt für umfangreichere und komplexere Anwendungen ausgeweitet, besticht dBase durch mehrere Vorteile. Neueinsteigern erlaubt das Programm, auf der Kommandozeile einzelne Befehle einzugeben und diese nacheinander ausführen zu lassen. Der mit einer kleinen Benutzeroberfläche versehene Befehl »assist« vereinfacht zusätzlich das Anfreunden mit den Leistungsmerkmalen von dBase, da Sie damit bereits über zahlreiche wichtige Funktionen verfügen, für die Sie nicht einmal das entsprechende Kommando mit allen Zusatzinformationen kennen müssen. Vielmehr wählen Sie aus den vorhandenen Menüpunkten mit Hilfe der Richtungstasten beispielsweise eine Datenbank aus, legen sie neu an oder stellen ihren Inhalt dar.

Für Fortgeschrittene ist es sinnvoll, wiederkehrende Aufgaben zu automatisieren, also ein Programm hierfür zu schreiben. Im einfachsten Fall enthält ein solches Programm lediglich die Eingaben, die Sie auf Kommandoebene in gleicher Reihenfolge hätten eingeben müssen. Dem weiteren Programmausbau mit Hilfe von Wiederholungs- und Verzweigungsanweisungen steht bei Bedarf nichts im Wege.

Hinter dieser Flexiblität des Einsatzes steckt das Konzept eines Interpreters, der Befehle, Funktionen, Anweisungen und Zusatzinformationen (»Parameter«) direkt auswertet. dBase prüft also bei Befehlseingaben (oder Programmen) ständig, ob die Schreibweise korrekt ist, die Rahmenbedingungen für die Anweisung – zum Beispiel eine geöffnete Datenbank – erfüllt sind und die richtige Anzahl, Reihenfolge und Typen von Parametern oder Variablen vorliegen. Nach erfolgreichem Abschluß dieser Kontrollen veranlaßt dBase die gewünschten Schritte. Da Prüfung und Auswertung Bestandteile des Programms selbst sind, liegt der damit verbundene Nachteil eines Interpreters nahe: Die Befehle und Programme sind nur innerhalb dBase lauffähig, auf einem Computer ohne das »Hauptprogramm« können Sie dBase-Programme nicht verwenden. Doch selbst mit dBase ergibt sich durch das wiederholte Prüfen der Anweisungsfolgen, selbst wenn das Programm schon bearbeitet und ausgewertet wurde, eine wesentlich niedrigere Bearbeitungsgeschwindigkeit, als dies bei einem auf MS-DOS-Ebene ausführbaren Programm der Fall wäre. Zudem liegen die interpretierten dBase-Programme in direkt lesbarer Form vor, wodurch auch Außenstehende das Programm ändern können. Eine Alternative oder Ergänzung zum Interpreter dBase stellt Clipper dar. Als Compiler wandelt er dBase-Programme in autark lauffähige Programme um. Die so erzeugten Anwendungen können Sie auf beliebig vielen Computern mit dem Betriebssystem MS-DOS einsetzen, wobei ihre Anwendungen eine wesentlich kürzere Ausführungszeit benötigen. Änderungen, Korrekturen oder Erweiterungen kann nur der Programmierer vor-

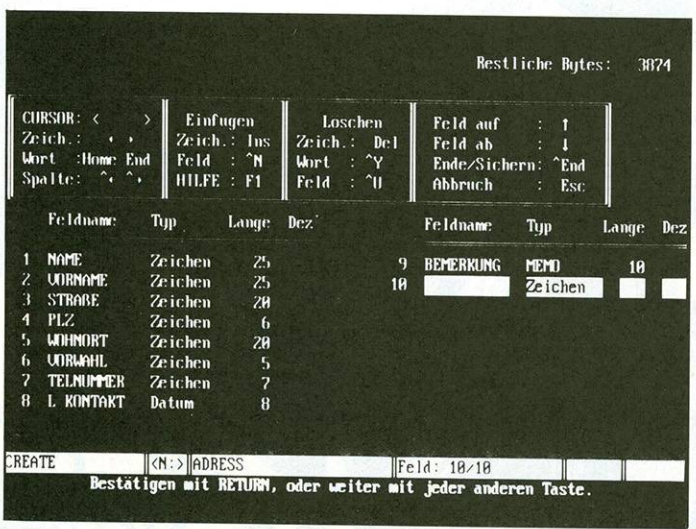


Bild 1. Die Struktur der Datenbank »adress.dbf« gestalten Sie mit »dBase« interaktiv

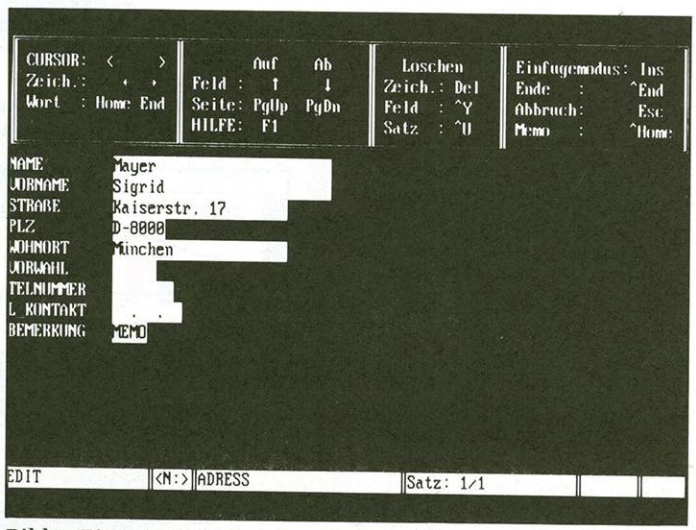


Bild 2. Einen Datensatz geben Sie bei »dBase« mit dem Befehl »edit« ein

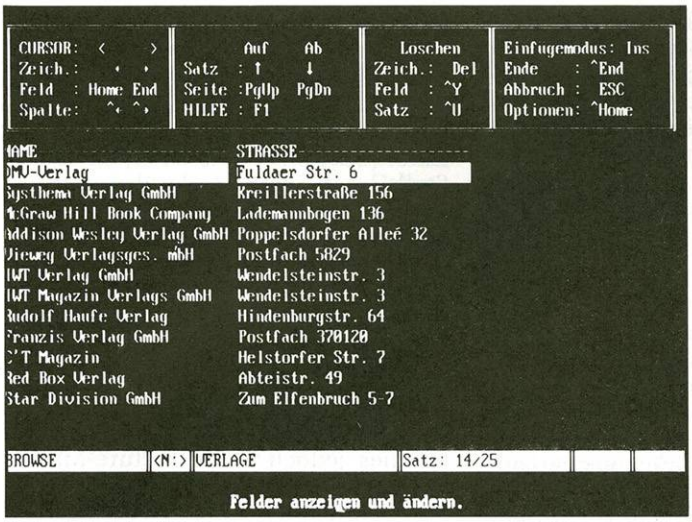


Bild 3. Mit dem dBase-Befehl »browse« stellen Sie Datensätze mit horizontalen und vertikalen Bewegungen der Richtungstasten dar

nehmen, der auch die Quellprogramme besitzt.

Clipper, der Datenbank-Compiler, läßt kaum mehr Wünsche offen

Speziell für den Einstieg in die Datenbankwelt ohne Programmierkenntnisse bietet sich dBase an.

Zusätzlich zu den bereits geschilderten Unterschieden gibt es bei beiden Programmen spezifische Befehle, Funktionen und Kontrollstrukturen, die beim jeweils anderen Programm nicht, oder nicht in dieser Form verfügbar sind. Wesentliche Befehle sowie der Aufbau und Einsatz von Datenbanken sind jedoch identisch und bilden die Grundlage ihrer gemeinsamen Darstellung.

Vor der ersten Programmierzeile müssen Sie die Struktur der künfig zu bearbeitenden Daten kennen. Eine Adreßverwaltung enthält beispielsweise Name, Vorname, Straße, Postleitzahl, Wohnort, Telefonvorwahl, Telefonnummer, letzter Kontakt und eine Kurzbemerkung. Name, Vorname und dergleichen stellen in der zu erzeugenden Datenbank die Feldnamen dar. Nachdem Anzahl und Bezeichnung der Felder somit bekannt sind, stellt sich die Frage nach dem Feldtyp, der sich auf den Inhalt der Felder bezieht. Als Klassen sind dabei Zeichen (»z«), numerisch (»n«), logisch (»l«), Datum (»d«) und Memo (»m«) denkbar. In unserem Beispiel sind Name, Vorname, Straße und Wohnort typische Zeichenfelder. Erlaubte Einträge können hierbei aus Groß- und Kleinbuchstaben, Zahlen und Sonderzeichen bestehen (zum Beispiel: Kaiserstr. 17). Postleitzahl, Telefonvorwahl und -nummer enthalten nur Zahlen, weshalb Sie sie als »numerisch« oder als »Zeichen« definieren. Die Festlegung »numerisch« hat den Vorzug, daß sich mit dem Inhalt des Feldes sofort Berechnungen durchführen lassen – beispielsweise können Sie alle oder bestimmte Feldinhalte addieren. Bei Postleitzahl, Telefonvorwahl und -nummer ist dies jedoch nicht sinnvoll.

»Letzter Kontakt« ist ein Datumfeld, in das Sie einen Kalendertermin eintragen. Die Kurzbemerkung kann wiederum ein Zeichenfeld darstellen, flexibler ist jedoch ein Memo-Feld. Dieses hat die Eigenschaft, beliebig lange Texte mit Zahlen und Sonderzeichen aufnehmen zu können und bietet sich dann an, wenn bei einem Eintrag wenig oder keine Kommentare, bei einem anderen sehr viel Begleittext zu erwarten ist.

Logische Felder, die in unserem Beispiel nicht enthalten sind, können zwei Zustände ausdrücken: wahr und falsch. Sie dienen nahezu ausschließlich dazu, den Programmablauf zu beeinflussen. Ein logisches Feld »mwst« könnte beispielsweise der Anwendung mitteilen, ob eine Mehrwertsteuerpflicht besteht oder nicht. Im ersten Fall könnten Sie den entsprechenden Mehrwertsteuersatz dem numerischen Feld »mwstsatz« entnehmen, andernfalls entfiele die Zwischenberechnung.

Die Festlegung der Feldlänge ist nur bei numerischen und Zeichenfeldern relevant, da logische Felder immer die Länge eins und Datumfelder die Länge acht besitzen. Memo-Felder enthalten einen zehn Zeichen langen Zeiger auf extern abgelegte Informationen, die also nicht in der Datenbank selbst zu finden sind. Die beliebig langen Texte des Memo-Feldes finden Sie in einer Dbt-Datei mit dem Namen der Datenbank. So sind beispielsweise bei der Datenbank »adress.dbf« Memo-Feldinhalte in »adress.dbt« enthalten.

Achten Sie bei der Definition numerischer Felder darauf, die Anzahl der Stellen auch für künftige Zwecke ausreichend zu dimensionieren. Nachkommastellen – die letzte noch ausstehende Felddefinition – und Dezimalpunkt müssen Sie nämlich von der Länge des numerischen Felds abziehen. Weisen Sie beispielsweise dem numerischen Feld »ausgaben« die Länge sieben mit zwei Nachkommastellen zu, liegt der Wertebereich des Feldes nicht im Bereich der Millionen (»9,999,999«), sondern lediglich in dem der Tausender (»9,999.99«). Größere Wertzuweisungen würden an dieser Stelle einen Fehler im Programm verursachen.

Bei Zeichenfeldern spielt die Länge nur insofern eine Rolle, als ausreichend Platz vorhanden sein sollte, um auch längere Einträge ohne Abkürzungen vorzunehmen. Auf der anderen Seite liegt es in Ihrem Interesse, die vorhandene Speicherkapazität sparsam zu verwenden, indem Sie die Zeichenfelder nicht übermäßig dimensionieren. Bei einer Datenbank mit 5000 Datensätzen und unter anderem drei Zeichenfeldern, die Sie jeweils nur zehn Zeichen zu umfangreich festgelegt haben, ergibt sich ein zusätzlich belegter, aber effektiv nicht benötigter Speicherplatz von 150000 Zeichen.

Die Datenbank selbst enthält, bis auf die Daten der Memo-Felder, alle Informationen. Ein Feld stellt eine Kategorie dar, beispielsweise »Name«. Ein Datensatz enthält alle Einträge, die einem Element angehören. Demnach finden Sie beispielsweise unter dem Namen »Mayer« alle dazugehörigen Daten wie »Sigrid«, »Kaiserstr. 17« und so weiter.

Nachdem Sie Anzahl, Name, Typ, Länge und gegebenenfalls Nachkommastellen der Felder definiert haben, wird die physikalische Datenbankstruktur erzeugt. dBase stellt dafür den dialogorientierten Befehl »create« zur Verfügung (Bild 1). Da die Feldnamen maximal zehn Zeichen lang sein dürfen, müssen Sie einige Bezeichnungen kürzen. Den Unterstrich erreichen Sie durch gleichzeitiges Drücken von [Umschalt] und [-], wie Sie es bei der Feldnummer acht »L_KONTAKT« vorgeführt sehen. Dieser Unterstrich dient der optischen Trennung, denn Leerzeichen sind innerhalb der Feldnamen nicht erlaubt.

Clippers »create« stellt nur eine Vorstufe zur Erzeugung der gewünschten Datenbank dar, das heißt, es wird nur eine leere Strukturdatei angelegt, welche die künftigen Feldnamen, -typen und dergleichen aufnimmt. Jeder Datensatz der Strukturdatei steht für ein Feld der eigentlichen Datenbank. Erst durch die Anweisung »create Dateiname from Strukturdatei« legen Sie die gewünschte Datenbank (Listing) an. Die Strukturdatenbank enthält die in der Tabelle dargestellten Felder.

Nach ihrer Generierung bleibt die aktuelle Datenbank geöffnet. Das ge-

währleistet, daß Clipper die Datenbank nur einmal anlegt und dann auch ohne Neuanlage korrekt aktiviert. Wie Sie schon am Programmende von »erst_db.prg« (Listing) sehen, schließen Sie die aktive Datenbank durch »close«; das Öffnen erfolgt durch »use Datenbankname«. Dabei sind zwei Fälle zu unterscheiden: Wenn die Datenbank gefunden wird, also bereits existiert, soll sie aktiviert werden; andernfalls muß Clipper sie erst anlegen, wobei ein erstes Öffnen automatisch enthalten sein muß.
Die Funktion »file()« prüft, ob eine bestimmte Datei im aktuellen Verzeichnis steht. Die gesuchte Datei setzen Sie dabei in Anführungszeichen zwischen die Klammern. Das Ergebnis der Funktion ist entweder wahr oder nicht wahr. In unserem Fall bietet sich die Verwendung der »if ... else ... endif«-Konstruktion an. »if« prüft, ob die Bedingung erfüllt ist: hier stellen Sie im If-Pfad erst einmal fest, ob die Datei vorhanden ist. Alle weiteren Befehlszeilen bis »else« behandeln somit den Fall, daß der Dateiname gefunden wurde. Die Anweisungen nach »else« bis »endif« gehen vom Gegenteil aus. »endif« schließlich beendet die Fallunterscheidung. Die hieraus resultierenden Programmanweisungen könnten also folgendermaßen lauten:

```
if file("adress.dbf")
  use adress.dbf
else
  create adress.dbf
endif
```

Die Einrückungen dienen der besseren Lesbarkeit des Programms, um Fallunterscheidungen oder Programmschleifen optisch besser erkennen zu können.
Nachdem Sie eine Datenbank öffnen, schließen und erzeugen können, richten wir unseren Blick auf die Eingabe von Daten. Zuerst müssen Sie einen leeren Datensatz anlegen. Der Befehl »append blank« fügt den »Rohdatensatz« an das Ende der Datenbank an (Listing). dBase besitzt den Befehl »edit«, um Datensätze einzugeben oder zu überarbeiten (Bild 2). In dieser Form bietet Clipper keinen direkten Ersatz für »edit«, durch Programmierung läßt sich jedoch über »dbedit()« eine noch leistungsfähigere Eingabefunktion schaffen.

Datensätze gehen ein und aus

Zur Darstellung und Abfrage von Informationen stehen formatierte und unformatierte Anweisungen zur Verfügung. Unformatiert bedeutet in diesem Zusammenhang, daß Sie keinerlei Vorgaben, Muster oder Schablonen einsetzen. Dies ist unter Umständen von Vorteil, wenn Sie Eingaben beliebigen – erst bei der Programmausführung bekannten – Typs tätigen oder die Ausgaben ohne direkte Positionierung einfach nacheinander darstellen wollen. Häufiger verwenden Sie jedoch die formatierten Ein- und Ausgaben. Der Befehl »@ ... say ...« gibt an definierter Stelle des Bildschirms Daten aus:

```
@ 10,0 say "Dieser Text steht in Zeile 10, Spalte 0"
```

Analog dazu verwenden Sie »@ ... get ...«, um Informationen an spezifizierter Stelle einzulesen. Die Eingabe selbst veranlassen Sie erst durch den Befehl »read«, der alle bislang mit »get« definierten Einträge nacheinander abarbeitet. Die nach »get« verwendete Variable müssen Sie zuvor definieren, da das Programm sonst nicht feststellen kann, welchen Variablentyp (Zeichen, numerisch, logisch und so weiter) es berücksichtigen soll. Die Länge des Eingabefeldes richtet sich nach der bereits definierten Länge der Variablen oder des Datenbankfeldes. Durch vorherige Wertzuweisungen lassen sich Vorgaben erzielen:

```
stadt="München"
@ 3,2 say "Bitte Stadt eingeben: "
@ 3,25 get stadt
read
```

Die Variable »stadt« ist damit als Zeichenvariable festgelegt und besitzt die Länge sieben (sofern kein gleichnamiges Feld der aktiven Datenbank andere Werte vorgibt). »say« gibt einen Meldetext aus, und »get« definiert danach in der gleichen Zeile ein sieben Zeichen langes Eingabefeld mit der Vorgabe »München«. »read« veranlaßt die eigentliche Eingabe. Lediglich [Enter] anstelle eines neuen Wertes beläßt die Vorgabe. Die Zeichenvariable »stadt« können Sie nun beliebig weiterverwenden. »get« gestattet auch den Einsatz von Schablonen, wenn mit »picture« eine Festlegung erfolgt:

```
stadt="8000 München"
@ 3,2 say "Bitte Stadt eingeben: "
@ 3,25 get stadt picture "9999 # AAAAAAA"
read
```

»picture« beeinflußt die Gültigkeit von Zeichen an festgelegten Positionen. Beispielsweise erzwingt »9« eine Ziffer (oder ein Vorzeichen) an dieser Stelle, »#« erlaubt darüber hinaus auch Leerzeichen, und »A« erwartet ausschließlich Zeichen des Alphabeths. Alle Symbole ohne festgelegte Bedeutung werden als Festwerte behandelt und bei der Eingabe übersprungen:

```
@ 3,25 get stadt picture "9999 AAAAAAA"
```

Anstelle von »#« finden Sie in der letzten Listingzeile ein Leerzeichen. Da dies jedoch keine Picture-Maske darstellt, wird bei der Eingabe nach der Postleitzahl direkt zur Ortsbezeichnung gesprungen. Dennoch ist das Leerzeichen normaler Bestandteil des Variablen- beziehungsweise Feldinhaltes. Schablonen vermindern Fehleingaben und reduzieren Routinebestandteile in Abfragen. Trotzdem sollten Sie diese »Masken« nur mit Bedacht ins Programm einbauen, da Sie damit den Anwender zwingen, bestimmte Eingaben zu tätigen.
Außer den formatierten und unformatierten Ausgaben stellen dBase und Clipper jeweils Anweisungen zur Verfügung, mit deren Hilfe Sie Datenbanken in tabellarischer Form durchblättern können. Mit dem dBase-Befehl »browse« durcheilen Sie ausgewählte Felder horizontal und vertikal mit den Richtungstasten (Bild 3). Clippers Funktion »dbedit()« (Bild 4) bietet zahlreiche weitere Fähigkeiten, unter anderem können Sie diese Funktion fast beliebig erweitern, um beispielsweise das aktuell angezeigte Feld bei Eingabe von [Enter] zu editieren, und vieles mehr.

Variationen von Variablen

»say«, »get« und »read« sind die Grundfunktionen, um den Inhalt von

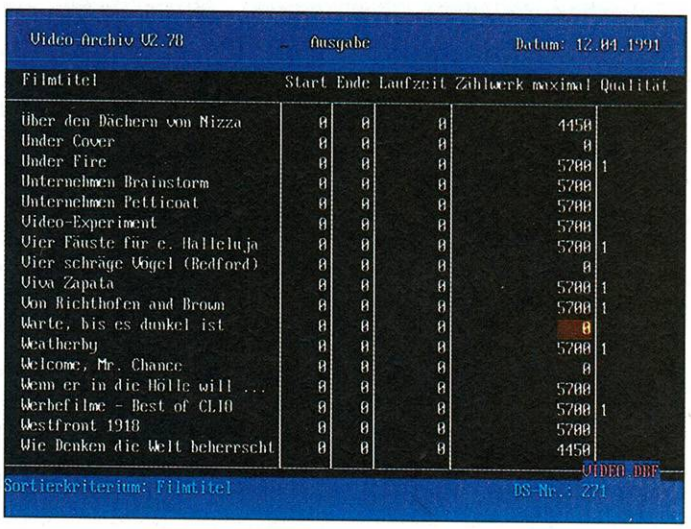


Bild 4. Die Clipper-Funktion »dbedit()« ist ein stark erweitertes »browse«

Datensätzen mit beliebigen Kommentaren an frei wählbaren Bildschirmpositionen auszugeben und gegebenenfalls neue Werte entgegenzunehmen. In unserem vorherigen Beispiel würde die Variable »stadt« die eingegebenen Zeichen enthalten, ohne das Datenbankfeld »stadt« zu beeinflussen. Betrachten Sie also die verschiedenen Arten von Variablen genauer, um eine Lösung für dieses Problem zu finden.
Nach dem Öffnen einer Datenbank sind alle Feldnamen als Datenbankvariablen frei einsetzbar. Mit »?«, »??« oder »say« können Sie den Inhalt des benannten Feldes darstellen, wobei die Informationen des aktuellen Datensatzes verwendet werden. Im Gegensatz zu diesen Variablen müssen Sie Speichervariablen erst festlegen. Die Zuweisung erfolgt entweder durch » = « oder »store«. Das Gleichheitszeichen weist einer Variablen einen (neuen) Wert zu, durch »store« lassen sich zahlreiche Variablen in einem Schritt definieren. Die Wertzuweisungen

```
ort="München"
mwst1=7
mwst2=14
```

legen also einzelne Werte in den Speichervariablen ab und klassifizieren bei der ersten Zuweisung den Typ als numerisch, Zeichen oder dergleichen. Prinzipiell identisch ist der Einsatz von »store«:

```
store 0 to a,b,c,d
```

Die Speichervariablen »a«, »b«, »c« und »d« sind also alle numerischer Art und erhalten den Wert »0«. Die Zuweisung von Werten an Datenbankfelder erfolgt mit »replace«:

```
ort="8000 München"
replace wohnort with ort
```

»Wohnort« sei in diesem Fall ein Feldname einer geöffneten Datenbank. Den Inhalt des Feldes im aktuellen Datensatz ersetzen Sie somit durch den Wert der Speichervariablen »ort«.
Fußangeln bestehen bei der Vorbelegung mit Anfangswerten in den Variablenarten »logisch« und »Datum«. Eine logische Variable erhält einen Wert mittels ».t.« (wahr) oder ».f.« (falsch):

```
ismwst=.t.
if ismwst
...Anweisungen
else
...Anweisungen
endif
```

»if« prüft, ob die Bedingung erfüllt ist (»ismwst = .t.«), » = .t.« brauchen Sie daher nicht mehr explizit erwähnen.
Datumfelder sehen zwar äußerlich wie Zeichenfelder aus, die »get-Anweisung« wertet den Typ »Datumfeld« jedoch automatisch richtig aus. Eine Zuweisung der Form

```
termin="12.04.91"
```

ergäbe daher eine Zeichen- und keine Datumvariable. Das Ersetzen der Datenbankvariable »l_kontakt« (Datum) ergäbe eine Fehlermeldung, die auf ein ungültiges Datenformat schließen ließe. Die korrekte Zuweisung lautet:

```
termin=ctod("12.04.91")
```

Die Umformung von Zeichen- in Datumvariable übernimmt die Funktion »ctod()«. Die Umkehrfunktion »dtoc()« wandelt entsprechend ein Datum wieder in eine Zeichenkette um.
Nach »say«, »get« und »read« bildet »replace« das letzte noch ausstehende Element, um Feldinhalte von Datensätzen zu verändern.

Bewegung in der Datenbank

Bisher war lediglich vom aktuellen Datensatz die Rede. dBase und Clipper bedienen sich eines sogenannten Datensatzzeigers, den Sie entweder direkt oder auch indirekt einsetzen. »append blank« zum Anfügen eines leeren Datensatzes ans Ende einer Datenbank beeinflußt indirekt diesen Satzzeiger, der leere Satz wird »aktuell«. Zahlreiche Befehle verschieben den Satzzeiger gezielt auf gewünschte Positionen: »go top« springt auf den ersten, »go bottom« auf den letzten Datensatz. »skip« oder »skip Anzahl« bewegt den Zeiger um eine oder in Anzahl definierte Satzposition weiter. Negative Werte (beispielsweise »skip -1«) verschieben entsprechend rückwärts. »go Zahl« springt direkt auf den gewünschten Datensatz. Zur Prüfung oder auch zur Zwischenspeicherung der Satzposition stehen die Funktionen »eof()« (Dateiende, nach dem Ende des letzten Satzes), »bof()« (Dateianfang, vor dem Beginn des ersten Satzes), und »recno()« (aktuelle Satznummer) zur Verfügung.
Die einzelnen Datensätze liegen physikalisch auf Ihrem Datenträger in der Reihenfolge vor, in der Sie diese eingegeben haben. »go top« mit folgendem »skip« bis zum Ende der Datenbank behält demnach auch diese physikalische Reihenfolge bei. Mit dem Befehl »sort« verändern Sie diese Reihenfolge, doch es gibt eine wesentlich leistungsfähigere Alternative.

Den indizierten Datensatz im schnellen Zugriff

Der Einsatz von Indexdateien erlaubt das logische Sortieren von Informationen. Auf dem Datenträger bleibt die physikalische Reihenfolge erhalten, aber durch Wahl und Aktivieren eines Indizes beachten Sie die darin vorgegebene Rangfolge. Ein Index kann alphabethischer, numerischer oder chronologischer Natur sein. Beim Erzeugen einer Indexdatei geben Sie einen Schlüsselausdruck an, der entweder aus einem Feld, einer Kombination mehrerer Felder oder auch aus Teilen davon besteht. Bei einzelnen numerischen, Zeichen- oder Datumfeldern vergibt das System je nach Feldtyp diesen Schlüsselausdruck automatisch. Verschiedene Feldtypen zusammengesetzter Schlüsselausdrücke müssen Sie mit Zahlen-, Zeichen- oder Datumumwandlungsfunktionen (»dtoc()«, »ctod()«, »str()« und so weiter) einander angleichen. Die Anweisung

```
index on Schlüsselausdruck to Dateiname
```

durchläuft alle Datensätze des Feldes oder der Felder des Schlüsselausdrucks und legt eine aufsteigend sortierte »Rangfolge« in der durch »Dateiname« angegebenen Indexdatei ab. Die automatisch vorgegebene Dateiendung ist bei dBase »ndx«, bei Clipper »ntx«. Der Inhalt der Indexdateien ist lediglich vom Programm selbst verwendbar. Sie aktivieren einen Index durch den Befehl:

```
set index to Indexdatei
```

Sie können auch mehrere Indexdateien, die Sie voneinander durch Kommata trennen, eingeben. Die erste Datei stellt den Hauptindex dar. Verwenden Sie beispielsweise mit

```
index on name to i_name
index on vorname to i_vorname
index on wohnort to i_wohnort
```

die Datenbankfelder »name«, »vorname« und »wohnort« zur Bildung einfacher Schlüsselausdrucke in die jeweils zugehörigen Indexdateien und aktivieren Sie durch

```
set index to i_name,i_vorname,i_wohnort
```

die Indizes, so ist die logische Reihenfolge die der Namen. Die physikalische Reihenfolge Ihrer ehemaligen Eingaben haben Sie damit ausgeschaltet. Die beiden weiteren Indizes sind nicht direkt aktiviert. Solange die Indizes eingeschaltet bleiben, werden Änderungen, Löschungen oder Erweiterungen sofort an der richtigen Stelle eingeordnet. Die Reihenfolge der Bearbeitung richtet sich jedoch nach dem Hauptindex. Mit dem Befehl

```
set order to Indexnummer
```

aktivieren Sie einen Nebenindex als

Feldname	Feldtyp	Feldlänge
FIELD_NAME	C	10
FIELD_TYP	C	1
FIELD_LEN	N	3
FIELD_DEC	N	3

Clippers »create« legt eine Strukturdatenbank mit festgelegten Feldinformationen an. Erst über den Inhalt der Datensätze und Umwandlung mit »create ... from ...« entsteht die eigentliche Datenbank.

neuen Hauptindex. In unserem letzten Beispiel wäre »i_name« der Index 1 (automatischer Hauptindex), »i_vorname« Nummer 2 und »i_wohnort« schließlich die Nummer 3. Die Indexnummer 0 (»set order to 0«) wählt die physikalische Reihenfolge, wobei allerdings alle laufenden Indizes – sofern erforderlich – korrigiert werden.

Wer Daten sucht, der findet

Mit aktiven Indizes verhalten sich auch zahlreiche Befehle zum Positionieren des Datensatzzeigers unterschiedlich. Nur die beiden »festen« Anweisungen behalten ihre Funktion bei, alle relativen Befehle und Funktionen passen sich der geänderten logischen Reihenfolge an. »skip« verschiebt zwar immer noch den Satzzeiger um eins weiter, aber beispielsweise anstelle von Satznummer acht auf neun nun von »Mayer« zu »Meyer«.
Aktive Indizes bringen auch den schnellen Zugriff auf gesuchte Informationen mit sich. Die Befehle »seek« und »find« dienen ausschließlich der Suche von Daten in Datenbanken mit aktiviertem Index. Die Suche erfolgt dabei im gültigen Hauptindex. Prinzipiell arbeiten beide Befehle identisch; »find« ist etwas problemloser bei der Übergabe von Suchdaten, da Zeichenketten nicht unbedingt in Leerzeichen eingeschlossen sein müssen. »seek« wiederum erlaubt auch die Verwendung von Variablen. Beide Befehle finden nur dann den gesuchten Ausdruck, wenn dieser mit dem oder den ersten Zeichen eines durch Index sortierten Feldinhalts übereinstimmt.
Wird ein Ausdruck gefunden, ist es denkbar, daß die folgenden Elemente des Index das Suchkriterium ebenfalls erfüllen. Aus diesem Grund sollten Sie mit »eof()« zuerst feststellen, ob das Ende der Datenbank schon erreicht ist, und danach mit »skip« den folgenden, im Sortiermuster vorliegenden Datensatz anwählen. Das neue Datenbankfeld müssen Sie dann nochmals mit dem Suchmuster vergleichen, da »found()« nur die erfolgreiche Durchführung der Suchbefehle testet, also gefunden oder nicht gefunden. »seek« und »find« suchen allerdings nur bis zum ersten gefundenen Eintrag.
Entscheidend beeinflußt das Suchverhalten der Schalter »set exact«. Steht er auf »on«, finden die Befehle nur völlig mit dem Suchmuster identische Einträge, »off« gibt sich schon mit gleichem Anfang zufrieden und findet beispielsweise bei der Suche nach »Mayer« auch »Mayerhofer«.
Wesentlich flexibler, aber sehr viel langsamer arbeitet der Locate-Befehl. Er funktioniert auch ohne Indexdateien und sucht in den angegebenen oder auch allen Feldern festgelegter oder aller Datensätze nach dem gewünschten Ausdruck. Interessant ist dabei die Fähigkeit, auch Teile von Zeichenketten aufzufinden.

```
locate for "yer"$name
```

Dieser Befehl begrenzt die Suche auf alle Elemente, die im Feld »name« die Zeichenkette »yer« enthalten (also »Mayer«, »Meyer«, »Bayer«). Anstelle des Operators »$« können Sie auch die Funktion »substr()« einsetzen. Bei Verwendung von »locate« sollten Sie unbedingt genaue Bedingungen vorgeben, damit nicht zufällig aus anderen, unwichtigen Feldern Fundstellen auftreten, die Sie gar nicht wünschen. Wie bei »find« und »seek« positioniert auch »locate« den Satzzeiger automatisch auf den Datensatz mit der gefundenen Information. Aufgrund der logisch anderen Arbeitsweise von »locate« können Sie mit der Anweisung »continue« die Suche fortführen. »eof()« und »found()« sind auch hier – wie bereits beschrieben – benutzbar.

Die Arbeit mit Zeichenketten ordnet die Datenflut

Beim Locate-Befehl tauchten bereits zwei Funtionen zur Zeichenkettenmanipulation auf: die Teilkettenoperatoren »$« und »substr()«. Die Praxis besteht häufig aus Verarbeitung von Zeichenketten (»Strings«), beispielsweise verbinden Sie diese miteinander zu einer neuen Zeichenkette:

```
Variable=Zeichenkette+Zeichenkette
```

Die Funktionen »left()« und »right()« geben die bestimmte Anzahl von Zeichen einer Zeichenkette ab dem Anfang zurück. Genauso erhalten Sie gewünschte Zeichen auch vom Ende in Richtung Anfang Ihrer Zeichenketten. Zur Ermittlung der Länge eines Strings dient die Funktion »len()«. »lower()« wandelt die Großbuchstaben einer Zeichenkette in Kleinbuchstaben, »upper()« setzt analog die Kleinbuchstaben in Großbuchstaben um. Häufige Wiederholungen von Zeichen lassen sich durch »replicate()« vereinfachen. »space()« füllt Variablen mit der gewünschten Anzahl von Leerzeichen auf. Das Entfernen von überflüssigen Leerzeichen am Anfang übernimmt »ltrim()«; am Ende stehende Leerzeichen beseitigen Sie durch »rtrim()« oder »trim()«. »at« sucht eine Zeichenkette in einer anderen, und mit »stuff()« können Sie gezielt Teile von Strings ersetzen.
Das Spektrum der Funktionen zur Zeichenkettenbearbeitung und Veränderung ist enorm groß. Neben weiteren Anweisungen zur Zeichenkettenmanipulation erschließen auch die Funktionen zur Datentypenumwandlung (»ctod()«, »dtoc()«, »val()« und »str()«) zusätzliche Einsatzgebiete. Verwenden Sie ruhig ausreichend Zeit damit, sich mit allen Bearbeitungsfunktionen vertraut zu machen.

Logische und relationale Verknüpfungen

Ähnlich elementar wie die Stringbearbeitung sind auch logische und relationale Operatoren. Die drei logischen Verknüpfungen ».and.«, ».or.« und ».not.« decken einzeln oder in Kombination das weite Feld von Verknüpfungszusammenhängen ab. ».and.« liefert als Ergebnis »wahr«, wenn die durch ».and.« verbundenen Ausdrücke beide wahr sind, sonst bleibt als Rückgabewert »falsch«. Bei ».or.« reicht es aus, wenn einer der beiden Ausdrücke wahr ist, damit das Ergebnis von ».or.« ebenfalls »wahr« lautet. ».not.« dreht gewissermaßen einen logischen Wert um: Aus wahr wird falsch und umgekehrt. Zahlreiche Funktionen wie etwa »file()« liefern logische Werte, die sich mittels dieser Operatoren prüfen oder verändern lassen. Haupteinsatzgebiete sind bedingte Verzweigungen durch »if ... endif«, »if ... else ... endif« oder Programmschleifen mit »do while ... enddo«, wie Sie dem folgenden Beispiel entnehmen können:

```
do while .not. eof()
...Anweisungen
enddo
```

Diese Syntax wiederholt eine Anweisung oder Anweisungsfolge solange, bis das Ende der Datenbank erreicht ist.
Die relationalen Operatoren prüfen auf kleiner (»<«), größer (»>«), gleich (»=«), ungleich (»<>«), kleiner oder gleich (»<=«) und größer oder gleich (»>=«). Die Prüfung kann numerische, Zeichen- und Datumfelder behandeln. Bei Zeichenfeldern, -variablen oder -konstanten ziehen Sie den Wert des Zeichens laut Zeichensatz heran – »b« ist demnach größer als »a« –, bei datumorientierten Typen erfolgt der Vergleich chronologisch:

```
a=5
b=7
geb_tag1=ctod("15.01.63")
geb_tag2=ctod("17.10.61")
l_var=.t.
if geb_tag2<geb_tag1 .and. b>=a .or.
l_var=.f.
...Anweisungen
else
...alternative Anweisungen
endif
```

Die Anweisungen nach »if« treffen zu, da der zweite Geburtstag früheren Datums als der erste ist, und die Zusatzbedingung ».and. b>=a« auch zutrifft. »l_var« ist zwar wahr, obwohl

die Prüfung auf »falsch« testet, da aber ».or.« nur einen wahren Ausdruck benötigt (»b> = a«), bleibt auch das Gesamtergebnis der If-Abfrage schließlich wahr; »net« eignet sich in diesem Zusammenhang ausgezeichnet, um bedingte Verzweigungen dem von Ihnen gewünschten logischen Aufbau anzupassen. Üblicherweise behandeln Sie zuerst die Situation, die am häufigsten auftritt, sowohl aus Gründen der Übersichtlichkeit als auch wegen der Geschwindigkeit bei der Programmausführung. Sofern Sie jedoch eine andere Auswertungsreihenfolge bevorzugen, bietet sich »if .not.« an.

Zeit- und Datumfunktionen sowie mathematische Funktionen und Berechnungen bieten sich geradezu an, komplexe Auswertungen aus oder in Datenfeldern vorzunehmen. So steht beispielsweise »sum« zur Verfügung, um die Gesamtsumme beliebiger Felder zu ermitteln. Dabei lassen sich auch Felder durch Formeln miteinander verbinden.

```
sum netto*1.14 to s_var
```

Das Ergebnis der Berechnung legen Sie hierbei in der Speichervariable »s__var« ab. Bei der Definition der Datenbank können Sie durchaus Felder anlegen, die nicht zur direkten Eingabe durch den Anwender bestimmt sind. Beispielsweise könnte lediglich die Eingabe eines Nettowerts erforderlich sein. Der Bruttowert ließe sich dann durch

```
replace all brutto with netto*1.07
```

bilden. Entsprechend könnten Sie verschiedene, änderbare Mehrwertsteuersätze festlegen und eine Zuweisung zum jeweiligen Datensatz erfolgen lassen. Das Programm könnte so feststellen, welcher Steuersatz (7 oder 14 Prozent) im einzelnen Fall gültig ist und Anpassungen an neue Rahmenbedingungen automatisch vornehmen.

Gleiches gilt für mathematische sowie Zeit- und Datumfunktionen. Häufig, wenn auch nicht immer, können Sie sich und den Anwender von Eingaben befreien, die zwar in der Datenbank enthalten sein, aber möglichst autonom dorthin gelangen sollen. Ein typisches Beispiel hierfür wäre eine Video-Datenbank, bei der aus dem Zählwerkstand am Anfang und Ende des Films die Gesamtspielzeit und der auf Kassette noch verfügbare Zeitraum hervorginge.

Ergebnisse der Berechnungen füllen zusätzliche Felder

Beide errechneten Werte bieten sich an, diese in der Datenbank selbst für weitere Verwendungen zu speichern. Um bei der Programmierung zahlreiche, mehrfach auftretende Programmteile nicht immer erneut schreiben zu müssen, stehen Ihnen Prozeduren zur Verfügung. Sie stellen Unterprogramme dar, die in sich abgeschlossene Tätigkeiten erfüllen, beispielsweise den Bildschirmaufbau bewältigen, Bildschirmmasken für »say«, »get« und »read« aufrufen und so weiter. Durch den Einsatz von Prozeduren halten Sie Ihr Programm kürzer und übersichtlicher, wobei der Prozedurname schon auf die Arbeitsweise des Unterprogramms hinweisen sollte. Der größte Vorteil dieser Technik betrifft das Verhalten bei Änderungen am Programm. Arbeiten Sie mit ständig in gleicher Form wiederkehrenden Programmblöcken, in welchen eine Korrektur erforderlich ist, müssen sie alle diesbezüglichen Teile suchen und anpassen. Eine Prozedur macht nur eine Änderung notwendig.

Prozeduren oder die Technik der Unterprogramme

Das Schlüsselwort »procedure«, gefolgt vom gewünschten Prozedurnamen, leitet ein solches Unterprogramm ein, die Anweisung »return« beendet es. Legen Sie die Prozedur nicht im Hauptprogramm selbst ab, müssen Sie bei dBase mit »set procedure to« den Namen der Prozedurdatei angeben. Aus dem Programm rufen Sie die Prozedur mit »do Prozedurname« oder »do Prozedurname with Parameterliste« auf. Das Unterprogramm läßt sich also ohne oder mit Übergabewerten einsetzen. Wenn Sie Parameter verwenden, muß in der Prozedur an erster Stelle die Anweisung »parameters« mit einer Liste der Übergabewerte stehen, die in Anzahl und Typen mit denen beim Aufruf konform sind. »parameters« richtet dabei nach Bedarf lokale Variablen ein, wenn Sie bei der Übergabe feste Ausdrücke verwendet haben. Sofern Sie beim Aufruf Variablen als Parameter einsetzen, können Sie ihren Inhalt im Unterprogramm ändern und sie dem Hauptprogramm nach Abschluß der Prozedur mit geändertem Inhalt zur Verfügung stellen. Hierin unterscheiden sich Prozeduren bei der Datenbanksprache dBase von denen in anderen Hochsprachen, wo eine klare Trennung von Prozedur und Funktion besteht: Prozeduren können üblicherweise keine Werte zurückgeben, dies ist dort einzig den Funktionen vorbehalten.

Sofern »parameters« lokale Variablen einrichtet, sind sie nur dem Unterprogramm bekannt. Diese Gesamtübersicht über den Aufbau von Prozeduren soll Sie nicht erschrecken, denn für den Anfang muß schließlich nicht jeder Aspekt zum Einsatz kommen. Als Vorgehensweise empfiehlt es sich, bereits fertig getestete Programmteile in Prozeduren umzuwandeln. Mit der Zeit oder bei steigenden Anforderungen erfolgt dann die schrittweise Parametrisierung von ehemaligen Konstanten im Unterprogramm. Dann gestalten Sie Ihre Prozedur flexibler und können sie in weitere Anwendungen aufnehmen. Manchmal lohnt es sich auch, kleine Programmteile in solche Unterprogramme umzusetzen, beispielsweise bei »set color to ...«. Die entsprechenden Programmzeilen sind weniger verständlich, als wenn Sie etwa »do gruen__schrift«, »do rot__schrift« und dergleichen programmieren.

Dieser Beitrag konnte nur die wichtigsten Konzepte, Techniken und Anweisungen behandeln und soll Ihnen viel Freude beim weiteren Experimentieren machen und einen kleinen Ausblick auf die in den Anweisungen schlummernden Fähigkeiten geben.

(Herbert Buckel/et)

```
[DY1]    1: create temp
[AM6]    2: && Temporäre Strukturdatenbank anlegen
[C97]    3: && »temp« ist dadurch automatisch geöffnet
[ØØØ]    4:
[ZZ1]    5: append blank
[SL4]    6: && leeren Datensatz anfügen
[ØØØ]    7:
[ØZ4]    8: replace field_name with "name"
[WH4]    9: replace field_type with "c"
[W44]   1Ø: replace field_len with 25
[ØØØ]   11:
[XP6]   12: && ersetzen der leeren Datensätze durch
[F56]   13: && die gewünschten Feldinformationen
[MP6]   14: && »replace field_dec with zahl« ist nur
[7A6]   15: && bei numerischen Feldern mit Nachkomma-
[7T3]   16: && stellen erforderlich
[ØØØ]   17:
[ZZ1]   18: append blank
[Y16]   19: && weiteren leeren Datensatz anfügen
[1L5]   2Ø: replace field_name with "vorname"
```

```
[WH4]   21: replace field_type with "c"
[W44]   22: replace field_len with 25
[ØØØ]   23:
[T56]   24: && ... hier werden alle weiteren Felder
[KS5]   25: && wie bereits gezeigt mit »replace«
[DU1]   26: && definiert
[ØØØ]   27:
[8V3]   28: create adress from temp
[VM6]   29: && Aus den Feldinformationen in »temp«
[8X6]   3Ø: && entsteht die Datenbank »adress.dbf«
[Z94]   31: && »adress.dbf« ist nach
[7H5]   32: && »create ... from ...« automatisch
[112]   33: && geöffnet.
[ØØØ]   34:
[3WØ]   35: close
[VC6]   36: && Schließen der aktiven Datenbank, da
[264]   37: && das "Programm" hier endet
```

Listing. Mit »erst__db.prg« schaffen Sie über eine Strukturdatei die gewünschte Datenbank

Tips & Tricks für Einsteiger

Hier finden Sie Hinweise, wie Sie die verschiedenen Packer komfortabel handhaben, wie Sie eine Printer-Karte installieren oder das Environment aus einem Turbo-Pascal-Programm heraus verändern. Ein Programm erlaubt Ihnen das zeitabhängige Ausführen Ihrer Anwendungen; Sie installieren Ihre Maus nur wenn nötig und benennen in Zukunft auch Unterverzeichnisse um.

Tips zur Datenkomprimierung

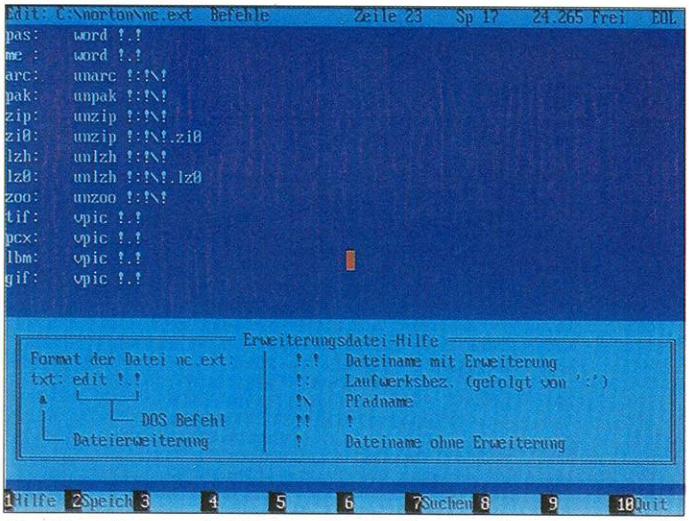
Sowohl beim Übertragen von Daten und Programmen zwischen Computern als auch beim Archivieren leisten Kompressionsprogramme wertvolle Hilfe. Ein Nadelöhr in der Datenübertragung bleibt das analoge Telefonnetz. Jedes KByte an Daten kostet je nach Entfernung und Übertragungsgeschwindigkeit Telefongebühren.
Komprimieren Sie die Daten vor dem Versenden, sparen Sie Geld. Die neuen Generationen dieser sogenannten Packer – auch Arcer genannt – enthalten zahlreiche wichtige Funktionen:

- Sie reduzieren die Datenmenge häufig um mehr als 50 Prozent.
- Sie fassen zahlreiche Einzeldateien, einschließlich verschiedener Unterverzeichnisse, in eine einzige komprimierte Datei zusammen.
- Optional erreichen Sie eine Verschlüsselung Ihrer Daten.
- Die komprimierten Daten lassen sich in selbstentpackende Exe- oder Com-Dateien auf Ihren Datenträgern speichern.

Die derzeit geläufigsten Komprimierungsprogramme sind »pkzip« und »lharc«. »pkzip« erhalten Sie als Shareware, wobei Sie für den kommerziellen Einsatz die kostenpflichtige Nutzungslizenz brauchen. »lharc« beziehen Sie »Public Domain«, also in allen Einsatzgebieten unentgeltlich. Beide Programme glänzen mit besten Komprimierungsraten. Seltener kommen »pkxarc«, »pak«, »arc« oder »zoo« zum Einsatz. Da alle erwähnten Programme entweder Public-Domain-Programme oder Shareware sind, lassen Sie sich über Mailboxen, Computerclubs oder Shareware-Händler beziehen. Sofern Sie zum Ortstarif eine Mailbox erreichen können, die diese Packer zur Verfügung stellt, sollten Sie davon Gebrauch machen – die Ersparnis an Telefonübertragungskosten amortisieren sich meist in wenigen Tagen.
Ein weiterer Zeitvorteil ergibt sich durch die Zusammenfassung beliebig vieler Einzeldateien zu einer Zieldatei. Viele Übertragungsprotokolle sind beispielsweise nicht in der Lage, Dateigruppen zu versenden, hierzu gehört beispielsweise das Protokoll »XModem«. Bieten nun entweder Sender oder Empfänger der Daten den Einsatz der komfortablen Protokolle nicht an, stünden Sie vor einer Sisyphusarbeit: Für jede Datei müßten Sie das Zielsystem informieren, ob Sie senden oder empfangen wollen, wie die Datei heißt und mit welchem Protokoll Sie übertragen wollen. In jedem Fall schneller, bequemer und rationeller ist also auch hierbei der Einsatz eines Packers.
Beim Archivieren von Quellprogrammen, Bildern und so weiter können Sie ganze Unterverzeichnisse in den »Arc«, wie das Resultat der Komprimierung heißt, einbeziehen. Einen praktischen Nutzen haben Sie besonders dann, wenn Sie kurzfristig Platz auf der Festplatte schaffen müssen, ohne hierzu ein Programmsystem löschen und hinterher neu installieren zu müssen.
Beide Vorteile nutzen Sie besonders beim Konvertieren von Daten zwischen verschiedenen Computern. Der verwendete Packer muß allerdings auf beiden Geräten verfügbar sein. Wenn Sie Daten zwischen PC und Amiga austauschen wollen, muß auf beiden Rechnern beispielsweise »zoo« installiert sein. Bei einer Direktverbindung entstehen normalerweise keine Kosten; allerdings sind die Datenmengen, etwa beim Umsteigen auf ein leistungsfähigeres Modell einer anderen Computerkategorie, üblicherweise enorm. Zudem sind es oft Hunderte von Einzeldateien, die Sie weiterverwenden möchten.
In bestimmten Fällen lohnt sich auch die Verschlüsselung der Daten, unter Umständen beim Datenaustausch zwischen Filialen und einer Hauptstelle via Telefonnetz oder bei mehreren Personen, die sich einen Account in einer kommerziellen Mailbox teilen. Private Post ist dadurch für Unbefugte nicht einsehbar.
Selbstentpackende Dateien herzustellen ist ebenfalls in zahlreichen Fällen von Vorteil. Der Packer versieht dabei die Daten mit einem ausführbaren »Kopf« und speichert sie als Com- oder Exe-Datei auf Ihrem Datenträger ab. In diesem Kopf befindet sich die Information, die zum Entpacken erforderlich ist. Der Empfänger startet den Prozeß des Entpackens, indem er auf MS-DOS-Ebene den Namen der Datei eingibt und mit der Eingabe-Taste bestätigt. Alles weitere geschieht automatisch, und vor allem ohne daß der Empfänger den jeweiligen Packer selbst besitzen oder aufrufen muß. Sofern der Packer diese Option explizit als Public-Domain herausstellt, können sogar kommerziell vertriebene Programme mit hohem Datenträgerverbrauch (wie beispielsweise Bilder, Laserfonts und dergleichen) davon sinnvoll und kostenlos Gebrauch machen. Geringere Herstellungs- und Konfektionierungskosten und einfachere sowie schnellere Installation auf der Festplatte sprechen dabei für sich. Die selbstentpackenden Dateien sind allerdings computer- oder betriebssystemspezifisch, womit Sie sie also nicht für die Konvertierung zwischen verschiedenen Geräteklassen verwenden können.
Den aufgeführten Vorteilen stehen einige Einschränkungen gegenüber: Packer benötigen Zeit zum Verdichten und zudem Platz auf Datenträgern. Beides ist natürlich stark von der verwendeten technischen Ausstattung abhängig. »lharc« und »pkzip« arbeiten auch bei 4,77-MHz-PCs mit mäßig schneller Festplatte zufriedenstellend, ihre Vorzüge lassen sich jedoch erst richtig ab der AT-Klasse mit RAM-Disk ausreizen. Vorhandene Coprozessoren bringen bei einigen Komprimierungsprogrammen zusätzliche Geschwindigkeitsvorteile. Schwerwiegender ist das Problem des externen Speicherplatzbedarfs, wenn Sie ausschließlich mit Diskettenlaufwerken arbeiten. In einem solchen Fall ist es empfehlenswert, für die Zieldatei und eine temporär verwendete Arbeitsdatei jeweils ein anderes Laufwerk anzugeben, da Sie sonst schnell die Meldung »Diskette voll« erhalten.
Nicht unerwähnt sei auch die von Packer zu Packer verschiedene Syntax beim Aufruf mit verschiedenartigen Parametern. Die Aufrufsyntax der beiden wichtigsten Operationen ist bei nahezu allen Packern identisch:

- Komprimieren:

```
<Name des Packers> a <Name der Zieldatei> <Dateispezifikation>
```



Die »Erweiterungsdatei« des »Norton Commander« ergänzen Sie um die Dateiendungen der gepackten Programme

Zum besseren Verständnis ein Beispiel:

```
lharc a test *.txt
```

»lharc« verdichtet alle Dateien des aktiven Verzeichnisses mit Endung »txt« zur Datei »test.lzh«.

- Dekomprimieren:

```
<Name des Packers> e <Name der Zieldatei>
```

Also ergibt beispielsweise die Syntax:

```
lharc e test
```

»lharc« entpackt den Inhalt der Datei »test.lzh« in das aktive Verzeichnis.

Die Option »e« zum Entpacken sollten Sie dem Schalter »x« immer vorziehen, da einige Packer (beispielsweise »pak«) bei »x« die Arc-Datei löschen, was nicht immer wünschenswert ist, da es zu Datenverlusten führen kann. Die obigen Beispiele des Aufrufs sind nur einfache Grundversionen, die Sie mit Hilfe zahlreicher weiterer Parameter verändern oder erweitern können. Genaueres entnehmen Sie den jeweiligen Hilfsdateien der einzelnen Programme.

Einige Packer sind etwas flexibler, was die Reihenfolge der Argumente betrifft, andere weniger.

Nachdem Sie die für den eigenen Bedarf wichtigsten Parameter aus den schon erwähnten Dokumentationsdateien entnommen haben, sollten Sie sich Stapeldateien schreiben, in denen Sie die erforderlichen Parameter unter leicht einprägsamen Aufrufnamen festlegen. Dazu lesen Sie in diesem Sonderheft den Beitrag »Befehle, die vom Stapel laufen«. Selbst bei Verwendung zahlreicher Packer mit verschiedener Aufrufsyntax ergeben sich bei Aufruf und Bedienung keine Schwierigkeiten mehr. Ebenfalls von Vorteil ist der Einsatz eines Unterverzeichnisses für zeitweilige Arbeiten, welches somit häufig den Namen »C:\tmp« trägt. »unzip.bat« (Listing 1) ist ein Beispiel für ein kleines Stapelverarbeitungsprogramm, das ».zip«-Dateien in das Verzeichnis »tmp« auf Laufwerk D entpackt, wobei die Stapeldatei »unzip.bat« auch komprimierte Unterverzeichnisse in gleicher Weise anlegt. Listing 2, »unlzh.bat«, bewältigt dieselbe Aufgabe für das Komprimierungsprogramm »lharc«. Passen Sie die Stapeldateien Ihren Verzeichnis- und Laufwerksgegebenheiten an. Zum Entpacken müssen Sie dann nur noch die Namen der Stapeldatei und der zu dekomprimierenden Datei eingeben, ohne dabei Gedanken an die jeweilige Aufrufsyntax verschwenden zu müssen.

»pkarc« und »pkzip« arbeiten beim Entpacken mit eigens dafür bestehenden Programmen. So verwirklichen beispielsweise »pkunzip«, »pak« oder »lharc« das Entpacken als internen Übergabeparameter. Mit »Norton Commander« geht es sogar noch komfortabler. Die Datei zur Festlegung, was mit welcher Programmendung (beispielsweise »*.txt«) geschehen soll, ergänzen Sie um die Endungen der verschiedenen Arc-Programme und rufen dann beispielsweise obige Stapeldatei auf (Bild). Die Stapeldatei sollte sich in einem durch »path« festgelegten Verzeichnis befinden. Zum Entpacken einer Datei reicht die Eingabe ihres Namens mit einem abschließenden [Enter]. Durch den Einsatz einer Maus erhalten Sie noch mehr Bedienungskomfort.

In Listing 3 ist ein beispielhafter Inhalt der Datei »nc.ext« (Steuerdatei für den »Norton Commander«) dargestellt. Diese »Erweiterungsdatei« rufen Sie in der deutschen Version des Programms durch [F9] [B] [W] auf.

Erwähnenswert ist bei diesem Beispiel, daß Endungen wie »zi0«, »zi1« oder »lz1« aufgeführt sind und der Batch-Datei diese Endungen mit übergeben werden. Dateien dieser Art entstehen, wenn Sie mit bestimmten Übertragungsprotokollen arbeiten, die beim Empfangen von Daten feststellen, ob die gewünschte Datei schon vorhanden und identisch ist (beispielsweise »ZModem«). Ist der Name identisch, nicht aber der Inhalt und soll auch keine Fehlerkorrektur durchgeführt werden, benennt das Übertragungsprotokoll jeden weiteren Versuch mit einer fortlaufenden Zahl am Ende (statt »lzh« also »lz0« etc.). Der Packer würde diese Dateien normalerweise nicht bearbeiten, da er die eigene Kennung voraussetzt (etwa »arc«), außer Sie teilen ihm etwas anderes (zum Beispiel durch den Befehl »lharc e bats.lz1«) mit. Bei den vergleichsweise geringen Kosten für einen Packer lohnt die Anschaffung sicherlich.

(Herbert Buckel/et)

```
[DA1]    1: echo off
[P9Ø]    2: d:
[Ø51]    3: cd\tmp
[412]    4: pkunzip %1 -d
```

Listing 1. Mit »unzip.bat« dekomprimieren Sie automatisch die mit »pkzip« gepackten Dateien im Verzeichnis »d:\tmp«

```
[DA1]    1: echo off
[AAØ]    2: n:
[Ø51]    3: cd\tmp
[1U1]    4: lharc e %1 *.*
```

Listing 2. Die Stapeldatei »unlzh.bat« ist dem Packer und Entpacker »lharc« angepaßt

```
[X82]    1: txt:    word !.!
[G42]    2: bak:    word !.!
[A62]    3: sik:    word !.!
[Y42]    4: doc:    word !.!
[WM2]    5: wks:    123view !.!
[YD2]    6: dbf:    dbview !.!
[F62]    7: prg:    word !.!
[C62]    8: asm:    word !.!
[3W1]    9: c  :    word !.!
[DW1]   1Ø: h  :    word !.!
[C62]   11: pas:    word !.!
[6Ø2]   12: me :    word !.!
[6L2]   13: arc:    unarc !:!\!
[CN2]   14: pak:    unpak !:!\!
[6T2]   15: zip:    unzip !:!\!
[393]   16: ziØ:    unzip !:!\!.ziØ
[6R2]   17: lzh:    unlzh !:!\!
[U83]   18: lzØ:    unlzh !:!\!.lzØ
[ST2]   19: zoo:    unzoo !:!\!
[M42]   2Ø: tif:    vpic !.!
[262]   21: pcx:    vpic !.!
[B42]   22: lbm:    vpic !.!
[U32]   23: gif:    vpic !.!
```

Listing 3. So könnte der Inhalt der häufig genutzten Erweiterungsdatei »nc.ext« aussehen

Printer-Karte installieren

Sowohl bei der Installation als auch bei der direkten Programmierung von parallelen Schnittstellen entstehen oft Rätsel, auf welchen Port und welches Lpt-Kürzel die Karte anspricht. Die Beschreibungen von Printer-Karten geben in dieser Hinsicht meist sehr verwirrende Auskunft. Grundsätzlich werden alle Peripherie-Geräte (Tastatur, Bildschirm sowie Drucker) über sogenannte Ports angesprochen. Ein Port hat nichts mit den Steckplätzen auf der Hauptplatine zu tun, sondern ist eine Adresse auf der Ebene des Betriebssystems.

Für parallele Drucker-Schnittstellen stehen drei Adressen zur Verfügung, nämlich 03BChex, 0378hex und 0278hex. Beim Starten des Computers wird zunächst jeder Port in der genannten Reihenfolge überprüft, ob er von einer Drucker-Karte belegt ist. Die erste benutzte Portadresse wird in der 2 Byte großen BIOS-Variablen »lpt1« an der Adresse 0:0408hex gespeichert, die zweite in der Variablen »lpt2« an der Adresse 0:040Ahex und die dritte als »lpt3« an 0:040Chex. Nicht benötigte Variablen enthalten den Wert 0.

Ist später zum Beispiel von »lpt1« die Rede, so wird immer anhand dieser Variablen die Portadresse ermittelt. An der Adresse 0:040Ehex wäre zwar noch Platz für die Portadresse von »lpt4« vorhanden; für eine vierte parallele Schnittstelle ist aber keine Portadresse vorgesehen, weshalb dieser Speicherplatz frei bleibt.

Für die Installation von Printer-Karten bedeutet dies, daß eine einzelne Karte auf jeden Fall immer mit »lpt1« addressiert wird, gleichgültig auf welchen Port sie konfiguriert wurde. Bei mehreren parallelen Schnittstellen muß man in erster Linie darauf achten, keinen Port zweimal zu belegen. Sehen Sie dazu in der Beschreibung nach, welche der genannten Portadressen die Karten unterstützen, und stimmen Sie dementsprechend die Karten anhand der Jumper oder Dipschalter aufeinander ab.

Der Programmierer sieht daraus, daß er stets erst über die BIOS-Variablen den Port abfragen muß, wenn er eine bestimmte parallele Schnittstelle direkt programmieren will. Außerdem läßt sich leicht feststellen, wieviele parallele Druckeranschlüsse im Computer eingebaut sind. Dazu prüfen Sie, wieviele der Lpt-Variablen ungleich Null sind. Auch das Wechseln beispielsweise von »lpt1« und »lpt2« ist sehr einfach: Sie tauschen nur die Werte der Variablen miteinander aus.

Für die seriellen Schnittstellen gilt weitgehend das Gleiche. Die Variablen »com1« bis »com4« befinden sich ab der Speicheradresse 0:0400hex. Die Portadressen sind hier jedoch fest verteilt: com1 = 03F8hex und com2 = 02F8hex. Für die seltener vorhandenen, weiteren zwei Schnittstellen gilt com3 = 03E8hex und com4 = 02E8hex.

Anbei finden Sie das Beispielprogramm »lpt_com.c« (Listing 4), das alle vorhandenen Schnittstellen ermittelt. Sie wandeln es mit MS-C 5.0 oder Turbo-C 2.0 in eine Exe-Datei um.

Rufen Sie das Programm auf, erhalten Sie beispielsweise an einem 386-PC mit einer seriellen Schnittstelle und einer Modemkarte, die auf der zweiten seriellen Schnittstelle installiert ist, die Meldung:

```
serielle Schnittstellen (RS 232):
 COM1 am Port 03F8hex
 COM2 am Port 02F8hex
2 gefunden
```

für die parallele Schnittstelle erhalten Sie die entsprechende Meldung:

```
parallele Schnittstellen (Centronics):
 LPT1 am Port 0378hex
1 gefunden
```

(Hartmut Lemmel/Helmut Stult/et)

```
[5M3]    1: /*Programm: lpt_com.c
[166]    2:   Funktion: ermittelt Schnittstellen
[HW1]    3:   Sprache : C
[3K3]    4:   Autor: Hartmut Lemmel
[T15]    5:   (c)1991 DMV Widuch GmbH & Co. KG */
[ØØØ]    6:
[UWØ]    7: main()
[CX3]    8: { unsigned int far *sst;
[613]    9:   /* Pointer to WORD */
[7Z5]   1Ø:   unsigned char ix; /* Zähler-BYTE */
[BC5]   11:   printf("\nSchnittstellen Ihres
[YY2]   12:            Computers:\n\n");
[PL5]   13:   printf("serielle Schnittstellen
[7V1]   14:                    (RS232):\n");
[YU3]   15:   sst = (int far *) ØxØØØØØ4ØØ;
[CT4]   16:   /* Adresse der BIOS-Variable */
[8E5]   17:   for(ix=1; ix<=4 && *sst; ix++,sst++)
[UM3]   18:   printf("   COM%1i am Port
[TX2]   19:                Ø%3Xhex\n",ix,*sst);
[P35]   2Ø:   printf("%1i gefunden \n\n",ix-1);
[BM5]   21:   printf("parallele Schnittstellen
[S23]   22:                    (Centronics):\n");
[6V3]   23:   sst = (int far *) ØxØØØØØ4Ø8;
[CT4]   24:   /* Adresse der BIOS-Variable */
[8E5]   25:   for(ix=1; ix<=4 && *sst; ix++,sst++)
[UN3]   26:   printf("   LPT%1i am Port
[TX2]   27:                Ø%3Xhex\n",ix,*sst);
[P35]   28:   printf("%1i gefunden \n\n",ix-1);
[U7Ø]   29: }
```

Listing 4. »lpt_com.c« gibt Auskunft über die Schnittstellen Ihres Rechners

Speicherplatzermittlung komplexer Unterverzeichnisse

Wer seine Festplatte mit Hilfe von Unterverzeichnissen sauber und strukturiert verwaltet, hat Probleme, wenn er nachträglich wissen will, wieviel Speicherplatz diese belegen. MS-DOS stellt hier leider noch kein Programm zur Verfügung, das diese Aufgabe bewältigt. Mit dem kleinen Turbo-Pascal-Programm »pl.pas« (Listing 5) können Sie dieses Problem lösen. Es ermittelt die Summe aller Dateien in einem Unterverzeichnis einschließlich aller darunter liegenden Verzeichnisse sowie den davon benötigten Speicherbedarf. Der Aufruf kann auf verschiedene Arten erfolgen. Starten Sie »Pl« ohne Parameter, so nimmt es das aktuelle Verzeichnis als Ausgang und berücksichtigt auch die eventuell vorhandenen Unterverzeichnisse. Mit

```
pl <Pfad>
```

wird von dem als Parameter übergebenen Unterverzeichnis (im Extremfall c:\) ausgegangen.
Es ist selbstverständlich auch möglich, die Ausgaben zum Beispiel auf den Drucker umzuleiten. Dies erreichen Sie durch

```
pl > prn
```

Durch Eingabe eines Dateinamens anstelle von »prn« werden die Informationen in eine Datei geschrieben.

(Jörg Sellmann/Helmut Stult/rm)

```
[M33]    1: { Programm: pl.pas
[U16]    2:   Funktion: Speicherplatzermittlung
[YY5]    3:   Sprache:  Turbo Pascal ab Version 5.Ø
[EU3]    4:   Autor:    Jörg Sellmann
[P35]    5:   (C)1991 DMV Widuch GmbH & Co. KG }
[ØØØ]    6:
[X82]    7: USES Crt, Dos;
[ØØØ]    8:
[KY4]    9: VAR aktdir, suchdir : STRING;
[RS3]   1Ø:     dateizahl       : INTEGER;
[9F3]   11:     disk_laeng      : real;
[EY2]   12:     out             : TEXT;
[ØØØ]   13:
[8V4]   14: PROCEDURE suchen(p : STRING);
[ØØØ]   15:
[PW3]   16: VAR datbuf : searchrec;
[ØØØ]   17:
[MUØ]   18: BEGIN
[ØD5]   19:   FindFirst(p+'*.*',anyfile,datbuf);
[EA3]   2Ø:   WHILE doserror=Ø DO
[MUØ]   21:     BEGIN
```

```
[9L2]   22:        IF ((datbuf.attr
[FN3]   23:          AND directory)>Ø) THEN
[MUØ]   24:          BEGIN
[623]   25:            IF (datbuf.name<>'.')
[494]   26:              AND (datbuf.name<>'..') THEN
[X64]   27:              suchen(p+datbuf.name+'\');
[4MØ]   28:          END
[ZRØ]   29:        ELSE
[MUØ]   3Ø:          BEGIN
[894]   31:            disk_laeng:= disk_laeng +
[L22]   32:                         datbuf.size;
[AØ4]   33:             dateizahl:= dateizahl+1;
[ZNØ]   34:          END;
[GX2]   35:        FindNext(datbuf);
[ZNØ]   36:      END;
[ZNØ]   37: END;
[ØØØ]   38:
[MUØ]   39: BEGIN
[9D2]   4Ø:   Assign(out,'');
[RB2]   41:   Rewrite(out);
[R92]   42:   dateizahl:= Ø;
[RH2]   43:   disk_laeng:= Ø;
[8K3]   44:   IF ParamStr(1)<>'' THEN
[MUØ]   45:     BEGIN
[ET2]   46:       GetDir(Ø,aktdir);
[DXØ]   47:       {$I-}
[A33]   48:       ChDir(ParamStr(1));
[BXØ]   49:       {$I+}
[T53]   5Ø:       IF IOResult<>Ø THEN
[MUØ]   51:         BEGIN
[124]   52:           WriteLn('Falscher Pfad');
[T41]   53:           Halt(1);
[ZNØ]   54:         END;
[ZNØ]   55:     END;
[6Ø3]   56:   GetDir(Ø,suchdir);
[ZR4]   57:   Write(out,'Unter ''',suchdir);
[V64]   58:   Write(out,''' belegen die ');
[236]   59:   IF suchdir[Length(suchdir)]='\' THEN
[SE5]   6Ø:     Delete(suchdir,Length(suchdir),1);
[Y43]   61:   suchen(suchdir+'\');
[8K3]   62:   IF ParamStr(1)<>'' THEN
[MUØ]   63:     BEGIN
[ØM1]   64:       ChDir('\');
[F82]   65:       ChDir(aktdir);
[ZNØ]   66:     END;
[D75]   67:   Write(out,dateizahl,' Dateien ');
[E86]   68:   WriteLn(out,disk_laeng/1Ø24:1:Ø,' kB.');
[VW1]   69:   Close(out);
[KNØ]   7Ø: END.
```

Listing 5. »pl.pas« bringt Licht ins Dunkel der Unterverzeichnisse. Es zeigt die Anzahl der Dateien und den benötigten Speicherplatz.

»Tips & Tricks für Programmierer« ist eine Rubrik zum Mitmachen. Sie müssen kein Schriftsteller sein, damit ein Beitrag aus Ihrer Feder (beziehungsweise aus Ihrer Tastatur) auf diesen Seiten veröffentlicht wird. Einschicken können Sie alles, was irgendwie mit der Programmierung des PC zusammenhängt. Das muß keine seitenlange Superroutine sein, auch eine nützliche Assembler-Routine, eine genial programmierte C-Funktion oder ein Patch für MS-DOS haben gute Chancen, in »Tips & Tricks für Programmierer« abgedruckt zu werden.

Und ein angemessenes Honorar gibt es für Ihre Bemühungen natürlich auch. Es richtet sich nach der Qualität der Tricks und der Dokumentation. Pauschalhonorare gibt es bei uns nicht, schließlich möchten wir gute Leistungen entsprechend würdigen.

Bitte schicken Sie eine Diskette mit einer kurzen Programmbeschreibung und dem Quelltext an:

DMV-Redaktionsbüro, DOS International,
Stichwort »Tips & Tricks«,
Gruberstraße 46c,
8011 Poing bei München

Wir freuen uns über jede Zuschrift.

(et)

BATH BEACH
SOFTWARE-SERVICE

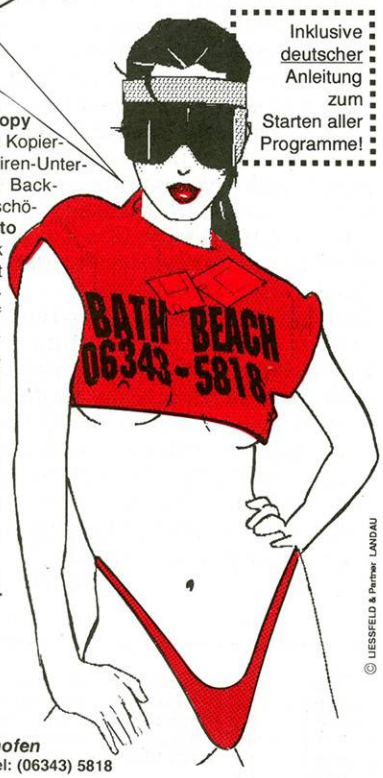
PUBLIC DOMAIN 150

Translator übersetzt englische Programm-Dokumentationen in Deutsch, **Lightning-Press** DTP-Desktop-Publishing, Fantastic **As Easy As** sehr gutes Tabellenkalkulationsprogramm, Bestseller **Prophet-Lotto** Statistikprogramm für Lotterieziehungen, **3D-Chess** dreidimensionales Schachprogramm, **PC-Write** sehr gute Textverarbeitung mit vielen Möglichkeiten, **Free-File** sehr gute Datenverwaltung mit vielen Möglichkeiten, **Composer** Musik komponieren, Noten schreiben und ausspielen, **PC-Akt** erzeugt digitalisierte Aktbilder auf ihrem PC, **Dancad-3D** dreidimensionales CAD-Zeichenprogramm mit Animation, **PC-KWIK** beschleunigt ihr Diskettenlaufwerk, **Mahjongg** sehr gutes chinesisches Spiel mit guter Grafik **Megagopoly** Spiel, wie erreiche ich eine Million, **Nuke New York** zeigt einen nuklearen Angriff und die Auswirkungen auf New York **Designer** ein Zeichenprogramm, das bewegliche Bilder erstellt **Black Jack** spielen wie im Casino **Horoskop** erstellt Ihr persönliches Horoskop **Kniffel** Kartenspiel **Mastermind** populäres Strategiespiel **Star-Traders** Händlerspiel, Aktien kaufen im Weltraum **Etikett** erstellt Etiketten in verschiedenen Formaten und druckt sie aus **Flugsimulator** simuliert Jet-Flüge **PC-Calc** populäre Tabellenkalkulation **Copy-All** nun können Sie fast alle geschützten Programme kopieren **PC-Window** PC-Fenstertechnik **Kalender** erstellt einen Kalender mit Bildschirm/Druck **Printum** schaltet den Drucker in andere Schriftarten um **Bastools** Basic-Tools **Your Menu** erstellt Ihr eigenes Menü zum Selbst-Ausführen **Form-Master** umfangreicher Formular-Generator **DOS-Help** zeigt und erklärt alle DOS-Befehle**StockCharting** Aktienverwaltung,Kauf-/V erkaufssignal(graf. Darstellung) **Biorythm** Biorythmen **Finance** Finanzen und Berechnungen**Taskplan** Zeitplanungsprogramm **Cover** druckt Diskettenhüllen mit Inhaltsverzeichnis **VCR-D-Base** verwaltet Video-Casetten und druckt Aufkleber **Bookinventory** Inventurprogramm für bis zu 2000 Bücher

150 Programme für nur 130.- DM!

P C -Lock schützt ihre Festplatte mit Passwort **Best Utilities** jede Menge Hilfsprogramme **Cataloging** erstellt einen Diskettenkatalog **FB-Copy** ein sehr schnelles und komfortabeles Kopierprogramm **Virus Check** Computerviren-Untersuchungsprogramm **Backgammon** Backgammonspiel **Space-Invaders** sehr schönes, bekanntes Arcade-Spiel **PC-Foto** Fotoverwaltungsprogramm incl. Druck von Dia-Aufklebern **Packdisk** macht mehr Platz auf Ihrer Festplatte **Compress** Datenkompakter, der Platz auf der Diskette spart **Chasm** Super Assembler-Programmiersprache **plus weitere 101 Public Domain- und Shareware-Programme.** *Hilfs-, Anwendungs- und Spielprogramme (BAS-, COM- und EXE-Startdateien) für IBM PC/XT/AT mit VGA, EGA-, CGA- und Herculeskarte.Alle Programme wurden auf Viren geprüft.*

Inklusive deutscher Anleitung zum Starten aller Programme!



© LIESSFELD & Partner LANDAU

Schnellversand am Tag des Bestelleingangs. Großer INFO-Katalog mit Programm-Beschreibungen und zusätzlichen Angeboten liegt jeder Bestellung bei.

Bath Beach
Software-Service GmbH
Speckstraße 23, D-6749 Schweighofen
BTX: * BATH # • FAX:(06342)6216 • Tel: (06343) 5818

BESTELL-COUPON

Ja, bitte senden Sie mir umgehend das PC-Paket "Public Domain 150" zum Discount-Preis von 130.-DM. **Bei Vorkasse keine Versandkosten.**
Bei Nachnahme: DM 10.- Versandkosten. Den Gesamtbetrag (Endpreis) bezahle ich wie angekreuzt. (Ausland nur Vorkasse: Scheck oder Bar).
Zusätzlich erhalte ich **kostenlos** den ausführlichen INFO-Katalog.

Endpreis: ☐ 5 1/4"Disketten ☐ Scheck/Bar DM 130.- ☐ Per Nachnahme DM 140.-
Endpreis: ☐ 3,5"Disketten ☐ Scheck/Bar DM 150.- ☐ Per Nachnahme DM 160.-
(3,5"Zoll-Disketten Material-Mehrpreis : 20.-)

Name:
Straße: Computer-Typ:
PLZ/ORT. Grafik-Karte:
Telefon:
Datum /Unterschrift
(Bei Minderjährigen der Erziehungsberechtigte) Geburtsdatum

DOEX 15/91

Environment nach Bedarf verändern

Beim Schreiben eines Menüprogramms in Turbo Pascal taucht die Frage auf, wie man im MS-DOS-Environment die Variable »path« ändert. Turbo Pascal bietet hier nur Routinen an, die eine Kopie des Environments oder einzelner Variablen daraus liefern.
Auf die übliche Weise können Sie Umgebungsvariablen unter Turbo Pascal nicht ändern. Auch mit Hilfe der Prozedur »exec ('\command.com', '/C PATH ...')« läßt sich kein Erfolg erzielen.
MS-DOS legt für alle Programme eine Kopie des Environments an. Verändert man diese, so bleibt das Original davon unbeeinflußt. Man muß also das Original verändern. Jedoch besitzt das MS-DOS-Environment keine festgelegte Speicheradresse, auf die man zugreifen kann. Daher muß man nach dem Eintrag suchen.
Dies sowie die Manipulation der Variablen erlaubt die Unit »env.pas« (Listing 6).
Zum Ermitteln der Startadresse des Environments dient dabei die Funktion »get_env_seg«. Sie durchsucht den freien Hauptspeicher nach dem Wort »COMSPEC«, welches den Beginn des Environments anzeigt. Dabei wird ausgenutzt, daß die Startadresse den Offset 0 besitzt. Die so ermittelte Adresse wird automatisch bei der Initialisierung der Unit in der globalen Variablen »env_seg« gespeichert.
Die Funktionen »get_env_str« und »get_env_var« dienen dazu, das komplette Environment oder einzelne Variablen als String zu liefern. Hierbei hilft, daß die Einträge durch das Zeichen »#0« getrennt sind und das Ende durch »#0#0« gekennzeichnet ist.
Zum Verändern oder Neueintragen von Variablen dient die Prozedur »put_env_var«. Diese prüft zuerst, ob bereits eine Variable vorhanden ist. Wenn ja, wird der zugeordnete Eintrag verändert, andernfalls wird die Variable neu angelegt. Bei einer leeren Stringvariablen erfolgt eine Löschung. Zur Vereinfachung des Programmablaufs erfolgen alle Einträge in Großschrift.
Eine kleine Demonstration der Arbeitsweise bietet das Programm »envtest.pas« (Listing 7).
Ein Wermutstropfen sei allerdings nicht verschwiegen: Läßt man ein Programm in der integrierten Entwicklungsumgebung von Turbo Pascal laufen, welches Variablen im Environment verändert, so kann man diese Veränderungen nicht mit dem Befehl »SET« in der Kommandoshell oder mit der Pascal-Routine »EnvStr« sehen, da beide auf eine Kopie des Environments zugreifen, die bereits vor der Manipulation generiert wurde.

(Rainer Satzinger/Helmut Stult/rm)

```
[PG3]    1: { Programm: env.pas
[Ø46]    2:   Funktion: Environment-Manipulation
[8Z5]    3:   Sprache:  Turbo Pascal ab Version 5.5
[W94]    4:   Autor:    Rainer Satzinger
[P35]    5:   (C)1991 DMV Widuch GmbH & Co. KG }
[ØØØ]    6:
[ZG1]    7: UNIT env;
[ØØØ]    8:
[SK1]    9: INTERFACE
[ØØØ]   1Ø:
[EC3]   11: VAR env_seg : WORD;
[ØØØ]   12:
[675]   13: FUNCTION get_env_str:STRING;
[ØØØ]   14:
[8P6]   15: FUNCTION get_env_var(env_var:STRING):
[GC1]   16:          STRING;
[ØØØ]   17:
[D37]   18: PROCEDURE put_env_var(env_var,var_str:
[3C1]   19:          STRING);
[ØØØ]   2Ø:
[4L2]   21: IMPLEMENTATION
[ØØØ]   22:
[FV5]   23: FUNCTION  upstr(s:STRING):STRING;
[ØØØ]   24:
[BL2]   25: VAR l   : INTEGER;
[BØ2]   26:     us : STRING;
[ØØØ]   27:
```

```
[MUØ]   28: BEGIN
[M21]   29:   us:= '';
[RV3]   3Ø:   FOR l:= 1 TO Length(s) DO
[UL3]   31:     us:= us+UpCase(s[l]);
[F12]   32:   upstr:= us;
[ZNØ]   33: END;
[ØØØ]   34:
[UV3]   35: FUNCTION get_env_str;
[ØØØ]   36:
[MR3]   37: VAR env_str : STRING;
[ZM2]   38:     env_ofs : BYTE;
[ØØØ]   39:
[MUØ]   4Ø: BEGIN
[PZ1]   41:   env_ofs:= Ø;
[232]   42:   env_str:= '';
[RØ6]   43:   WHILE NOT ((Mem[env_seg:env_ofs]=Ø)
[ER5]   44:         AND (Mem[env_seg:env_ofs+1]=Ø)) DO
[MUØ]   45:     BEGIN
[4D3]   46:       env_str:= env_str +
[3S4]   47:                 Chr(Mem[env_seg:env_ofs]);
[EE3]   48:       env_ofs:= env_ofs+1;
[ZNØ]   49:     END;
[3N4]   5Ø:   get_env_str:= env_str+#Ø#Ø;
[ZNØ]   51: END;
[ØØØ]   52:
[RT3]   53: FUNCTION get_env_var;
[ØØØ]   54:
[WS3]   55: VAR env_str  : STRING;
[5A3]   56:     position : INTEGER;
[ØØØ]   57:
[MUØ]   58: BEGIN
[F34]   59:   env_str:= get_env_str;
[ØV6]   6Ø:   position:= Pos(upstr(env_var),env_str);
[S53]   61:   IF position > Ø THEN
[MUØ]   62:     BEGIN
[MM4]   63:       Delete(env_str,1,position +
[PV2]   64:              Length(env_var));
[YH4]   65:       position:= Pos(#Ø,env_str);
[H15]   66:       get_env_var:= Copy(env_str,1,
[VZ1]   67:                          position-1);
[4MØ]   68:     END
[ZRØ]   69:   ELSE
[VT2]   7Ø:     get_env_var:= '';
[ZNØ]   71: END;
[ØØØ]   72:
[E34]   73: PROCEDURE put_env_var;
[ØØØ]   74:
[163]   75: VAR i               : BYTE;
[9H4]   76:     env_str,mod_str : STRING;
[ØØØ]   77:
[MUØ]   78: BEGIN
[F34]   79:   env_str:= get_env_str;
[DL5]   8Ø:   i:= Pos(upstr(env_var),env_str);
[2N1]   81:   IF i=Ø THEN
[MUØ]   82:     BEGIN
[V33]   83:       IF var_str<>'' THEN
[394]   84:         mod_str:= Copy(env_str,1,
[H53]   85:                        Length(env_str)-2) +
[L13]   86:                        #Ø+upstr(env_var) +
[673]   87:                        '='+upstr(var_str) +
[KCØ]   88:                        #Ø#Ø
[ZRØ]   89:       ELSE
[M83]   9Ø:         mod_str:= env_str;
[4MØ]   91:     END
[ZRØ]   92:   ELSE
[MUØ]   93:     BEGIN
[V33]   94:       IF var_str<>'' THEN
[MUØ]   95:         BEGIN
[UV3]   96:           i:= i+Length(env_var)+1;
[6Z4]   97:           mod_str:= Copy(env_str,1,i-1);
[FH3]   98:           Delete(env_str,1,i-1);
[P16]   99:           mod_str:= mod_str+upstr(var_str);
[A83]  1ØØ:           i:= Pos(#Ø,env_str);
[FH3]  1Ø1:           Delete(env_str,1,i-1);
[1S4]  1Ø2:           mod_str:= mod_str+env_str;
[4MØ]  1Ø3:         END
[ZRØ]  1Ø4:       ELSE
[MUØ]  1Ø5:         BEGIN
[6Z4]  1Ø6:           mod_str:= Copy(env_str,1,i-1);
[FH3]  1Ø7:           Delete(env_str,1,i-1);
[A83]  1Ø8:           i:= Pos(#Ø,env_str);
[693]  1Ø9:           Delete(env_str,1,i);
[1S4]  11Ø:           mod_str:= mod_str+env_str;
[4MØ]  111:         END
[ZNØ]  112:     END;
[Z95]  113:   FOR i:= 1 TO Length(mod_str) DO
[C16]  114:     Mem[env_seg:i-1]:= Ord(mod_str[i]);
[ZNØ]  115: END;
[ØØØ]  116:
[1V4]  117: FUNCTION get_env_seg : WORD;
[ØØØ]  118:
[G64]  119: VAR start_str : STRING;
[N92]  12Ø:     Addr      : WORD;
[9U2]  121:     env_ofs   : BYTE;
[MUØ]  122: BEGIN
[FU3]  123:   start_str:= 'COMSPEC';
[7L1]  124:   Addr:= $6Ø;
[4Ø3]  125:   WHILE Addr<$AØØØ DO
[MUØ]  126:     BEGIN
[5E6]  127:       IF Mem[Addr:Ø]=Ord(start_str[1]) THEN
```

```
[DM5]  128:          IF Mem[Addr:1]=Ord(start_str[2])
[2RØ]  129:            THEN
[KM5]  13Ø:            IF Mem[Addr:2]=Ord(start_str[3])
[2RØ]  131:              THEN
[PM2]  132:              IF Mem[Addr:3]=
[9Y3]  133:                Ord(start_str[4]) THEN
[SM2]  134:                IF Mem[Addr:4]=
[DY3]  135:                  Ord(start_str[5]) THEN
[UM2]  136:                  IF Mem[Addr:5]=
[HY3]  137:                    Ord(start_str[6]) THEN
[WM2]  138:                    IF Mem[Addr:6]=
[NY3]  139:                      Ord(start_str[7]) THEN
[MUØ]  14Ø:                      BEGIN
[EC3]  141:                        get_env_seg:= Addr;
[ØWØ]  142:                        Exit;
[ZNØ]  143:                      END;
[P32]  144:      Addr:= Addr+1;
[ZNØ]  145:      END;
[RP2]  146:    get_env_seg:= Ø;
[RH6]  147:    WriteLn('Environment nicht gefunden');
[DUØ]  148:    Halt;
[ZNØ]  149: END;
[ØØØ]  15Ø:
[MUØ]  151: BEGIN
[HZ3]  152:    env_seg:= get_env_seg;
[KNØ]  153: END.
```

Listing 6. »env.pas« erweitert Turbo Pascal um die Möglichkeit der Environment-Manipulation

```
[X94]    1: { Programm: envtest.pas
[H67]    2:   Funktion: Umgebungsspeicher modifizieren
[8Z5]    3:   Sprache:  Turbo Pascal ab Version 5.5
[W94]    4:   Autor:    Rainer Satzinger
[P35]    5:   (C)1991 DMV Widuch GmbH & Co. KG }
[ØØØ]    6:
[U62]    7: USES Crt,env;
[ØØØ]    8:
[WØ4]    9: VAR position : INTEGER;
[GX2]   1Ø:     oldpath  : STRING;
[ØØØ]   11:
[ØT4]   12: PROCEDURE ausgabe(s:STRING);
[MUØ]   13: BEGIN
[RZ1]   14:   WriteLn(s);
[XF1]   15:   WriteLn;
[AX3]   16:   WriteLn(get_env_str);
[991]   17:   ReadLn;
[ZNØ]   18: END;
[ØØØ]   19:
[MUØ]   2Ø: BEGIN
[DB1]   21:   ClrScr;
[A42]   22:   GotoXY(1Ø,25);
[LM4]   23:   Write('Weiter mit <RETURN>');
[GV1]   24:   GotoXY(1,1);
[EB5]   25:   ausgabe('MSDOS - Environment :');
[FR3]   26:   Window(1,WhereY,8Ø,24);
[U85]   27:   put_env_var('test','envtest');
[BC6]   28:   ausgabe('Variable "TEST" eingefügt ...');
[KX3]   29:   put_env_var('test','');
[T76]   3Ø:   ausgabe('... und wieder entfernt ...');
[3X4]   31:   oldpath:=get_env_var('path');
[HC5]   32:   put_env_var('path','pathtest');
[9K4]   33:   ausgabe('PATH geändert ...');
[PT4]   34:   put_env_var('path',oldpath);
[PR6]   35:   ausgabe('... und wieder zurückgesetzt');
[GH2]   36:   Window(1,1,8Ø,25);
[DB1]   37:   ClrScr;
[FH3]   38:   WriteLn('Das war''s');
[KNØ]   39: END.
```

Listing 7. »envtest.pas« zeigt die Änderung des Environments in einem Beispiel

Die Zeit entscheidet!

»timer.pas« (Listing 8) ist ein Hilfsprogramm zum zeitabhängigen Ausführen von Programmen oder Befehlen nach dem Einschalten des Computers. Eingefügt in die Datei »autoexec.bat« führt das Programm jeden gewünschten Befehl aus, vorausgesetzt das Einschalten des Programms fällt in den angegebenen Zeitraum.

Mit »timer.pas« können Sie also Ihren PC über eine Schaltuhr aktivieren, so daß er beispielsweise über das Modem zu einer festgelegten Zeit Nachrichten verschickt oder empfängt. Oder wenn die Hausfrau zwischen 8 und 16 Uhr ihren Haushaltsplan bearbeiten möchte, wird beim Einschalten das entsprechende Programm direkt gestartet.

Doch nun zum Programm selbst. Der Zeitraum und das auszuführende Programm werden in der Kommandozeile an »timer.pas« übergeben. Eine vollständige Kommandozeile zum Aufruf von Timer sieht folgendermaßen aus:

`timer <first> <last> <Programm> [Parameter]`

Bei fehlerhafter Eingabe bleibt der Aufruf wirkungslos. Sie erhalten nur die Anzeige der Kommandozeilen-Syntax. Durch die Parameter »first« und »lasttime« können Sie den Zeitraum, in dem »timer.pas« ein Programm aufrufen soll, in Stunden und Minuten (hh:mm) festlegen. Der Parameter »Programm« darf neben dem Dateinamen auch die Angabe von Laufwerk und Pfad enthalten. »Parameter« ist optional zu verwenden, falls Ihr Programm zusätzliche Angaben zum Aufruf benötigt.

Mit dem Timer können Sie auch MS-DOS-Befehle oder Batch-Dateien aufrufen, die mehrere Programme in Folge ausführen.

Nach dem Aufruf von Timer werden in der Prozedur »param_test« die Kommandozeilenparameter überprüft, um im Fehlerfall die korrekte Syntax anzuzeigen und das Programm abzubrechen. »times« liest dann die Grenzzeiten »ParamStr(1)« und »ParamStr(2)« ein und wandelt sie in numerische Werte um, was später einen Vergleich mit der Systemzeit erlaubt. Ungültige Zeitangaben (größer als 24 Uhr) erkennt das Programm in »times«, was ebenfalls zum Abbruch führt.

Wenn keine Fehler festzustellen sind, erfolgt die Rückkehr zum Hauptprogramm, das »compare« aufruft, um die Systemzeit einzulesen und diese mit den Grenzzeiten zu vergleichen. Bei Übereinstimmung der Zeiten bereitet das Hauptprogramm den eigentlichen Befehls- oder Programmaufruf vor. Hierzu werden »ParamStr(3)« und »ParamStr(4)« (Programm und Parameter) eingelesen und in der Variablen »command« zusammengefaßt. In der folgenden Schleife hat der Anwender ungefähr fünf Sekunden Zeit, den Start des angezeigten Befehls mit der Escape-Taste abzubrechen. Andernfalls startet in der Prozedur »execute« das gewünschte Programm. Dazu wird, nach dem Vertauschen der vom System belegten Interrupt-Vektoren mit den Variablen im Unit-System von Pascal, ein neuer Kommandoprozessor gestartet, dem der auszuführende Befehl als Parameter übergeben wurde. Nach Programmende meldet sich Timer mit »...zurück vom Aufruf«.

Bei den vielen Steuer- und Kontrollaufgaben, die der Rechner in zunehmendem Maße auch für den Privatmann erledigen kann, setzen Sie Timer in Zusammenarbeit mit einer Schaltuhr für ein reiches Aufgabenfeld ein.

(Herbert Palm/Helmut Stult/et)

```
[YN3]    1: { Programm: timer.pas
[N47]    2:   Funktion: Programmaufruf in Abhängigkeit
[S42]    3:             der Uhrzeit
[YY5]    4:   Sprache:  Turbo Pascal ab Version 5.Ø
[FF3]    5:   Autor:    Herbert Palm
[W25]    6:   (c)1991 DMV Widuch GmbH & Co. KG    }
[882]    7: {$M $2ØØØ,Ø,Ø }
[5H2]    8: Program Timer;
[S62]    9: uses Crt,Dos;
[RNØ]   1Ø: Var
[TØ5]   11: Stunde, Min, sek, sek1ØØ: Word;
[RY1]   12: time: Word;
[R44]   13: prg, kpara: String[5Ø];
[XF3]   14: command: String[1ØØ];
[BZ3]   15: time1, time2: String[5];
[M65]   16: time1i, time2i, code, I: Integer;
[GC1]   17: Ch: Char;
[L71]   18: C: Byte;
[AØ3]   19: procedure syntax;
[MUØ]   2Ø: begin
[XF1]   21:   WriteLn;
[XZ5]   22:   WriteLn(' Syntax : TIMER firsttime ',
[KK5]   23:       'lasttime programm [Parameter]');
[XF1]   24:   WriteLn;
[FF5]   25:   WriteLn(' firsttime: frueheste ',
[RD3]   26:           'Startzeit (hh:mm) ');
[4C5]   27:   WriteLn(' lasttime : spaeteste ',
[RD3]   28:           'Startzeit (hh:mm) ');
[R56]   29:   WriteLn(' programm : Laufwerk:\Pfad',
[H64]   3Ø:           '\Programm.Extension   ');
```

```
[WC6]   31:   WriteLn(' Parameter: Kommandozeilen',
[KT3]   32:           'parameter (optional)');
[DUØ]   33:   Halt;
[ZNØ]   34: end;
[3S3]   35: procedure param_test;
[7B7]   36: { Kommandozeilenparameter überprüfen }
[MUØ]   37: begin
[XX2]   38:   if (ParamCount < 3)
[KZ2]   39:   or (ParamCount > 4)
[7A4]   4Ø:   or (Length(ParamStr(1)) <> 5)
[N55]   41:   or (Length(ParamStr(2)) <> 5) then
[MUØ]   42:   begin
[XF1]   43:     WriteLn;
[8H4]   44:     WriteLn('******* Fehler beim ',
[T34]   45:             'Programmaufruf ! *******');
[LD1]   46:     syntax;
[ZNØ]   47:   end;
[ZNØ]   48: end;
[4W2]   49: procedure times;
[MUØ]   5Ø: begin
[554]   51:   { Grenzzeiten einlesen }
[383]   52:   time1:= ParamStr(1);
[883]   53:   time2:= ParamStr(2);
[CM4]   54:   { Trennzeichen entfernen }
[AP2]   55:   Delete(time1,3,1);
[BP2]   56:   Delete(time2,3,1);
[CH6]   57:   { Zeitstring nach Integer umwandeln }
[AU3]   58:   val(time1,time1i,code);
[GU3]   59:   val(time2,time2i,code);
[7T2]   6Ø:   if (time1i > 24ØØ) or
[FW2]   61:      (time2i > 24ØØ) then
[MUØ]   62:      begin
[XF1]   63:        WriteLn;
[4K4]   64:        WriteLn('***** Fehler in der ',
[XY2]   65:                'Zeiteingabe *****');
[ZZ4]   66:        WriteLn('Eingaben > 24:ØØ sind ',
[YZ2]   67:                'nicht moeglich !');
[DUØ]   68:        Halt;
[ZNØ]   69:      End;
[ZNØ]   7Ø: End;
[H33]   71: procedure compare;
[MUØ]   72: begin
[2B4]   73:   { Systemzeit einlesen }
[135]   74:   GetTime(stunde,min,sek,sek1ØØ);
[GX3]   75:   time  :=(stunde*1ØØ)+ min;
[1Z3]   76:   { Zeiten vergleichen }
[UU3]   77:   if (time1i > time2i) then
[MUØ]   78:   begin
[7D3]   79:     if (time >= time1i) OR
[B93]   8Ø:        (time <= time2i) then
[ZE1]   81:        C:= 1 else
[DUØ]   82:        Halt;
[4MØ]   83:   end
[ZRØ]   84:   else
[MUØ]   85:   begin
[FM3]   86:     if (time >= time1i) AND
[B93]   87:        (time <= time2i) then
[ZE1]   88:        C:= 1 else
[DUØ]   89:        Halt;
[ZNØ]   9Ø:   end;
[ZNØ]   91: end;
[X43]   92: procedure execute;
[MUØ]   93: begin
[F16]   94:   { Interupt-Vektoren vertauschen }
[N72]   95:   SwapVectors;
[GD3]   96:   { Programm starten }
[LB4]   97:   Exec (GetEnv('COMSPEC'), '/C '
[YT1]   98:         + (command));
[N72]   99:   SwapVectors;
[T55]  1ØØ:   WriteLn('...zurück vom Aufruf');
[XF1]  1Ø1:   WriteLn;
[ZNØ]  1Ø2: end;
[753]  1Ø3: { Hauptprogramm }
[MUØ]  1Ø4: begin
[662]  1Ø5:   param_test;
[L21]  1Ø6:   times;
[YD1]  1Ø7:   compare;
[CS1]  1Ø8:   if c = 1 then
[MUØ]  1Ø9:   begin
[S35]  11Ø:     { Pfad und Programm einlesen }
[WT2]  111:     prg:=paramstr(3);
[WD6]  112:     { Parameter einlesen wenn angegeben }
[2G3]  113:     if ParamCount = 4 then
[LX3]  114:       kpara :=ParamStr(4) else
[3R1]  115:       kpara :='';
[KW3]  116:     command :=(prg+' '+kpara);
[DB1]  117:     ClrScr;
[XF1]  118:     WriteLn;
[7E5]  119:     WriteLn(' TIMER  (c) Februar 1991',
[Y42]  12Ø:             ' by H.Palm ');
[XF1]  121:     WriteLn;
[NN4]  122:     WriteLn('Befehl : ',command);
[XF1]  123:     WriteLn;
[1X5]  124:     WriteLn('Sie haben 5 Sekunden Zeit',
[YZ3]  125:             ' den Aufruf des Befehls',
[Z94]  126:             ' mit <ESC> abzubrechen.');
[XF1]  127:     WriteLn;
[RV6]  128:     { Warteschleife für Programmabbruch }
[TP2]  129:     for I := 1 TO 1Ø Do
[MUØ]  13Ø:     begin
```

```
[Y23]  131:       if KeyPressed then
[MUØ]  132:       begin
[152]  133:         Ch := ReadKey;
[572]  134:         if Ch = #27 then
[DUØ]  135:         Halt;
[ZNØ]  136:       end;
[YL1]  137:       Delay (5ØØ);
[ZNØ]  138:     end;
[YE1]  139:     execute;
[X93]  14Ø:     if DosError <> Ø then
[U35]  141:     WriteLn('DOS-Fehlermeldung :
[HT1]  142:             DosError);
[ZNØ]  143:   end;
[KNØ]  144: end.
```

Listing 8. »timer.pas« schaltet Ihre Befehle im abgesteckten Zeitrahmen

Potenzieren in Turbo Pascal

Die Versionen 4.0, 5.0 und 5.5 von Turbo Pascal stellen keine Routine für höhere Rechenarten wie das Potenzieren zur Verfügung. Da Turbo Pascal jedoch sowohl den Logarithmus als auch die Exponentialfunktion kennt, kann man die Eigenschaften dieser beiden Funktionen zur Berechnung von Potenzen ausnutzen (Listing 9).
Der erste Schritt besteht darin, zu wissen, daß der Logarithmus und die Exponentialfunktion zueinander Umkehroperationen sind. Ein Beispiel hierfür ist etwa

`exp(ln(10))=10`

oder allgemein

`exp(ln(a)) = a`

Abhilfe schafft hier eines der leider so selten beachteten, vermutlich weil durch den Computer verdrängten, Logarithmusgesetze. Mit diesen lassen sich nämlich die etwas komplizierteren Rechenarten wie Mulitplizieren, Dividieren, Potenzieren und Radizieren um eine Stufe vereinfachen. So wird aus dem Potenzieren zum Beispiel ein Multiplizieren. Die Rechnung »a hoch b« läßt sich auf

`b*ln(a)`

vereinfachen. Die Herleitung ist folgende:

$a^b = \exp(\ln(a^b)) = \exp(b*\ln(a))$

Ein Zahlenbeispiel ist

$8 = 2^3 = \exp(b*\ln(3))$

Im kleinen Beispielprogramm ist noch zu beachten, daß die Basis nie Null beziehungsweise negativ werden darf, da der Logarithmus von negativen Zahlen nicht definiert ist. Des weiteren müssen Sie bei der Übertragung in eine andere Sprache beachten, daß die Funktion »exp« eventuell zur Basis 10 rechnet und nicht wie hier, zur Basis 2,7183 ...

(Hubert Ilbertz/Herbert Buckel/rm)

```
[WX3]    1: { Programm: potenz.pas
[ZS4]    2:   Funktion: Potenzrechnung }
[ØØØ]    3:
[5H1]    4: uses crt;
[ØØØ]    5:
[RNØ]    6: VAR
[CH3]    7: basis, exponent:real;
[ØØØ]    8:
[LE7]    9: Function Potenz(basis,exponent:Real):Real;
[ØØØ]   1Ø:
[MUØ]   11: Begin
[RL5]   12:   Potenz:=exp(exponent*Ln(basis));
[ZNØ]   13: end;
[ØØØ]   14:
[MUØ]   15: Begin
[DB1]   16:   clrscr;
[DØ4]   17:   write('Basis eingeben: ');
[M82]   18:   readln(basis);
[KN4]   19:   write('Exponent eingeben: ');
[FX2]   2Ø:   readln(exponent);
[77B]   21:   writeln( basis:Ø:9, '^ ',exponent:Ø:9,' =
              ',potenz(basis,exponent):Ø:9);
[KNØ]   22: end.
```

Listing 9. Das Beispielprogramm »potenz.pas« verwendet mathematische Umformungen bei der Potenzberechnung

Mäuse bitte nur, wenn nötig!

Jeder Mausbesitzer hat das Problem schon einmal gehabt: Das Programm, beispielweise »Dr. Genius« ist gerade gestartet, da fällt einem ein, daß der Maustreiber noch gar nicht installiert ist. Dies läßt sich zwar durch das Einbinden in die Datei »config.sys« umgehen, jedoch gehen dabei immer einige KByte Hauptspeicher unnötig verloren.

Es geht aber auch anders. Das Programm »maustest.pas« (Listing 10) erlaubt es, in einer Stapeldatei festzustellen, ob der Maustreiber schon installiert ist, damit er gegebenenfalls nachgeladen werden kann. Dadurch wird der Treiber nur dann geladen, wenn er benötigt wird.

Maustest prüft, ob ein Maustreiber installiert ist und gibt das Ergebnis als »Errorlevel« zurück. Dieser kann in Stapeldateien über die Abfrage »if Errorlevel ...« erkannt werden.

Ein »Errorlevel« von 0 bedeutet, daß kein Maustreiber installiert ist, ein »Errorlevel« von 1 zeigt an, daß der Maustreiber bereits geladen wurde.

Um die Funktionsweise verständlich zu machen, hier als Beispiel der Auszug aus einer Stapeldatei:

```
@echo off
MausTest
if ErrorLevel 1 goto MausDa
echo Noch kein Maustreiber installiert!
REM Maustreiber installieren...
Mouse
goto Weiter
:MausDa
echo Der Maustreiber schon installiert!
:Weiter
REM Hier kann das mausgesteuerte
REM Programm aufgerufen werden...
```

Dazu noch ein Tip: Fertigen Sie eine Stapeldatei nach gezeigtem Muster an, und benennen Sie diese zum Beispiel mit »mausda.bat«. Steht diese nun in einem in der Umgebungsvariablen »path« angegebenen Verzeichnis, kann sie beispielweise aus der Startdatei für »Dr. Genius« mit »call mausda.bat« aufgerufen werden. Auf diese Weise ersparen Sie sich, in jeder Stapeldatei die komplette Testabfrage eingeben zu müssen.

Den Errorlevel setzt in einem Turbo-Pascal-Programm die Funktion »Halt«. Dieser können Sie einen Wert übergeben, den »Halt« dann in die Variable »Errorlevel« schreibt. Diesen Wert können Sie zusätzlich zu der oben beschriebenen Methode, mit der Funktion »DosExitCode« aus der Unit »Dos« in einem als Unterprozeß gestarteten Programm abfragen.

(Heiner Steven/Helmut Stult/rm)

```
[533]    1: PROGRAM maustest;
[ØØØ]    2:
[VG1]    3: USES Dos;
[ØØØ]    4:
[6XØ]    5: CONST
[8E6]    6:   maus_int  = $33;     { Maus-Interrupt }
[DR6]    7:   maus_init = Ø;  { Maus initialisieren }
[XW6]    8:   maus_da = $FFFF;{ Rückgabe Maustreiber}
[ØØØ]    9:
[DU3]   1Ø: VAR regs    : REGISTERS;
[EH6]   11:     intadr : ^LONGINT; { Adresse Mausint.}
[ØØØ]   12:
[MUØ]   13: BEGIN
[LY6]   14:   intadr:= Ptr(Ø,maus_int*SizeOf(LONGINT));
[513]   15:   IF intadr^ <> Ø THEN
[MUØ]   16:     BEGIN
[FG3]   17:       regs.ax := maus_init;
[ØS3]   18:       Intr (maus_int, regs);
[714]   19:       IF regs.ax = maus_da THEN
[K71]   2Ø:         Halt (1);
[ZNØ]   21:     END;
[E71]   22:     Halt (Ø);
[KNØ]   23: END.
```

Listing 10. Mit »maustest.pas« den Maustreiber nur bei Bedarf laden

Umbenennen von Verzeichnissen

Das Betriebssystem MS-DOS erlaubt zwar das Umbenennen von Dateien, nicht jedoch von Verzeichnissen. Dieser Mangel läßt sich mit dem Programm »rendir.com« beseitigen.

Zur Generierung der Com-Datei wird ein kurzer Basic-Lader benötigt. Das Programm »rendir.bas« (Listing 11) läßt sich mit jedem beliebigen Editor generieren. Nach dem Speichern kann es in GW-Basic oder Quick Basic gestartet werden. Der Basic-Lader enthält zwei Prüfalgorithmen, die sicherstellen, daß das Programm die Datei »rendir.com« korrekt erzeugt.

Wer den Aufbau des Programms erforschen will, kann dazu den Debugger aufrufen. Mit einem Disassembler läßt sich dann die Struktur ergründen. Die Benutzung des Programms ist dem MS-DOS-Befehl »rename« für Dateien ähnlich. Auch bei Rendir geht man immer auf die Ebene, in der sich der zu ändernde Name (hier das Verzeichnis) befindet.

Nach dem Aufruf mit

`rendir <altdir> <neudir>`

wird das Verzeichnis »altdir« in »neudir« umbenannt. Falls nur »rendir« eingegeben wurde oder ein Fehler das Umbenennen verhindert, wird der richtige Befehlsaufbau erklärt, andernfalls wird die Umbenennung bestätigt.

Das Programm »rendir.com« benutzt die Funktion 17hex des Interrupt 21hex (Datei umbenennen). MS-DOS legt die Verzeichnisnamen auf die gleiche Art wie bei Dateien ab, jedoch mit einem anderen Attribut. Das heißt, das Umbenennen von Verzeichnissen funktioniert durch diesen neuen MS-DOS-Befehl ebenso wie »rename« für Dateien.

(Ulrich Borgdorf/Helmut Stult/rm)

```
[KU3] 1Ø 'Programm: rendir.bas
[BH6] 2Ø 'Funktion: Verzeichnisse umbenennen
[DA3] 3Ø 'Sprache:  GW-Basic
[S84] 4Ø 'Autor:    Ulrich Berghof
[Ø35] 5Ø '(C)1991 DMV Widuch GmbH & Co. KG
[W7Ø] 6Ø
[Y7Ø] 7Ø
[SY5] 8Ø OPEN "rendir.com" FOR OUTPUT AS #1
[PUØ] 9Ø CLS
[4P6] 1ØØ PRINT "Datei rendir.com wird erzeugt"
[9KØ] 11Ø :
[8P2] 12Ø FOR I = 1 TO 8Ø
[YY1] 13Ø READ BYTE
[AG3] 14Ø PRINT #1, CHR$(BYTE);
[G45] 15Ø P1 = P1 + BYTE: P2 = P2 + (I * BYTE)
[K46] 16Ø LOCATE 1, 6Ø: PRINT I; "von 8Ø Bytes";
[1H1] 17Ø NEXT I
[GF5] 18Ø IF P1 = 8392 AND P2 = 297821! THEN 26Ø
[1LØ] 19Ø :
[A41] 2ØØ CLS
[4U4] 21Ø PRINT "Prüfsummen-Fehler"
[ØN6] 22Ø PRINT "Fehler in Programm oder DATA- ";
[GT2] 23Ø PRINT "Zeilen!"
[FA2] 24Ø CLOSE : END
[LM3] 25Ø PRINT #1, "(C)1991 DMV ";
[KZ4] 26Ø PRINT #1, "Widuch GmbH & Co KG"
[2L4] 27Ø PRINT #1, "Befehlsformat: ";
[TZ5] 28Ø PRINT #1, "rendir <altdir> <neudir>"
[6M6] 29Ø PRINT #1, "<altdir> = Verzeichnis das";
[ØS5] 3ØØ PRINT #1, " umbenannt werden soll"
[UE6] 31Ø PRINT #1, "<neudir> = neuer Name des ";
[XK4] 32Ø PRINT #1, "Verzeichnisses"
[582] 33Ø PRINT #1, "$";
[5B1] 34Ø CLOSE
[TKØ] 35Ø :
[1Z4] 36Ø DATA 252,191,85,Ø,137,25Ø,184,255,Ø
[TP4] 37Ø DATA 171,254,192,171,171,198,5,16
[7Ø5] 38Ø DATA 18Ø,23,2Ø5,33,1Ø,192,116,9,186
[5Ø5] 39Ø DATA 8Ø,1,18Ø,9,2Ø5,33,2Ø5,32,186,43
[6Z4] 4ØØ DATA 1,18Ø,9,2Ø5,33,2Ø5,32,68,97,115
[LØ5] 41Ø DATA 32,86,1Ø1,114,122,1Ø1,1Ø5,99,1Ø4
[FY4] 42Ø DATA 11Ø,1Ø5,115,32,119,117,114,1ØØ
[P25] 43Ø DATA 1Ø1,32,117,1Ø9,98,1Ø1,11Ø,97,11Ø
[P84] 44Ø DATA 11Ø,116,46,32,32,32,32,36
```

Listing 11. Auch im alten GW-Basic lassen sich noch gute Programme entwickeln. »rendir.bas« zum Umbenennen von Verzeichnissen ist ein Beispiel dafür.

Quick Basic 4.5

Der programmierte Untergang

Jeder von Ihnen kennt das bei Alt und Jung beliebte Spiel »Schiffe versenken«, häufig verbotenerweise während des Schulunterrichts gespielt. Benutzten Sie bisher Papier und Bleistift als Spielutensilien, so werden Sie jetzt in die Lage versetzt, Ihren Computer »sinnvoll« einzusetzen.

Altbekannte Spiele sind oft die liebsten, weswegen sie auch Tradition erlangten. »Schiffe versenken« bietet ein Maximum an Spaß. Wie viele andere auch, müssen Sie dieses Spiel zu zweit spielen; die PC-Tastatur bietet jedoch nur Platz für einen Bediener. Unser Programm löst das Problem auf zwei unterschiedliche Arten:

1. Sie spielen gegen Ihren PC.

2. Sie spielen gegen eine zweite Person, deren PC über eine serielle Schnittstelle mit Ihrem Computer verbunden ist.

Zur Erinnerung noch einmal die Regeln des Spiels »Schiffe versenken«. Sie spielen auf einem Spielfeld, das in 10x10 Zellen aufgeteilt ist. Jeder Spieler hat sein eigenes Spielfeld, das er vor seinem Gegner versteckt hält. In der Aufbauphase des Spiels plaziert jeder Spieler seine fünf Schiffe senkrecht oder waagerecht auf dem Spielfeld. Dazu drücken Sie zunächst [w] oder [s] für waagerecht oder senkrecht. Zur Plazierung geben Sie das Feld an, welches die linke beziehungsweise obere Grenze des Schiffs bildet, in den Bildern »B2« für den Flugzeugträger, Sie tippen also [w] [b] [2]. Die Schiffe, ihren Namen und Größe entnehmen Sie der Tabelle. Ein zweites Spielfeld benötigen Sie, um eine Kontrolle über die von Ihnen gemachten »Schüsse« und Treffer zu haben. Wenn beide Spieler alle Schiffe plaziert haben, beginnt die Spielphase. Die Spieler zielen abwechselnd auf die Schiffe des Gegners, indem Sie eine Zelle benennen und Auskunft vom Gegner bekommen, ob der »Schuß« ein Treffer oder ein Fehlschlag gewesen ist. Zusätzlich ist ein Spieler dazu verpflichtet, das Versenken eines Schiffs bekanntzugeben (Bild 1). Ein Schiff ist versenkt, wenn alle Zellen, die das Schiff belegt, von Schüssen getroffen sind. Wer als erstes alle Schiffe des Gegners versenkt, gewinnt das Spiel.

Im ersten Bild sehen Sie den Bildschirm, wie er vom Programm aufbereitet wird. Links ist das eigene Spielfeld, rechts das des Gegners. Im eigenen Spielfeld sehen Sie Ihre Schiffe, zum Beispiel von B2 bis B6, und auch die Stellen, die vom Gegner angeschossen wurden. Treffer werden als rote Vierecke, Fehlschläge als gelbe Kreise dargestellt. Im gegnerischen Feld sehen Sie natürlich keine Schiffe – damit wäre alles ein bißchen zu einfach! Aber Ihre Schüsse werden wie im eigenen Feld durch rote Vierecke (Treffer) und gelbe Kreise (Fehlschläge) angezeigt. Das Programm führt Buch und meldet seine Schüsse, die versenkten Schiffe und natürlich das Ende des Spiels mit entsprechenden Meldungen (Bild 2).

Wenn der PC als Gegner keine Herausforderung mehr darstellt, oder wenn eine zweite Person teilnehmen will, müssen Sie zunächst einen zweiten PC und ein sogenanntes Null-Modem-Kabel beschaffen. Verbinden Sie die beiden PCs mit dem Kabel und geben Sie »COM1« beziehungsweise »COM2« an, wenn Sie das Programm starten. (Von der Quick-Basic-Entwicklungsumgebung benutzen Sie dazu das Menü »Run« und die Option »Command$«.)

Zu zweit spielen Sie verkabelt

Als erstes nach dem Programmstart und nur für sehr kurze Zeit lesen Sie die Meldung »Verbindung zum zweiten PC wird aufgebaut«. Danach läuft das Spiel so ab, wie oben bereits beschrieben. Lediglich die Pausen, in denen Ihr Gegner seinen nächsten Zug überlegt, werden länger – Ihr menschlicher Gegner wird etwas mehr Zeit in Anspruch nehmen, seinen Zug vorzubereiten, als der PC benötigt. Die beiden Computer kommunizieren miteinander über die serielle Schnittstelle und sorgen für einen korrekten Ablauf.

Wenn die Kommunikation nicht klappen sollte, gibt es eigentlich nur zwei mögliche Gründe. Zuerst prüfen Sie, ob die serielle Schnittstelle, die Sie beim Programmstart angegeben haben, auch wirklich die ist, an der das Kabel angeschlossen ist. Wenn diese Angabe stimmt, starten Sie das Programm noch einmal, diesmal auf beiden PCs möglichst gleichzeitig. Wenn die Kommunikation immer noch nicht funktioniert, sollten Sie das Kabel prüfen. Die vorhergehen-

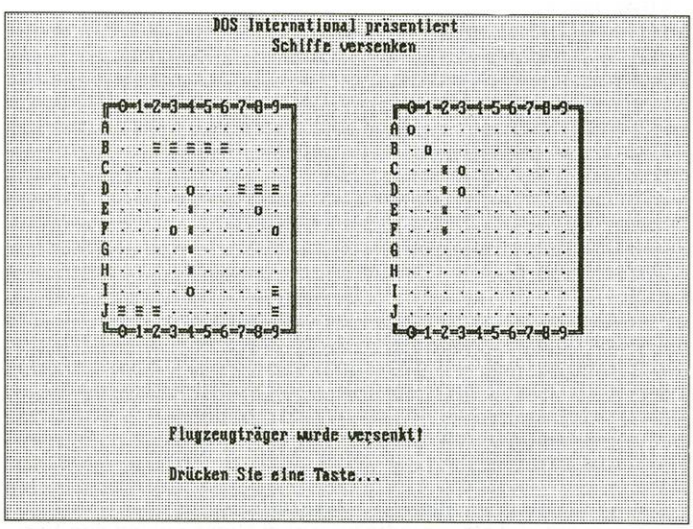


Bild 1. Das Programm meldet Ihnen, wenn ein gegnerisches Schiff versenkt wurde

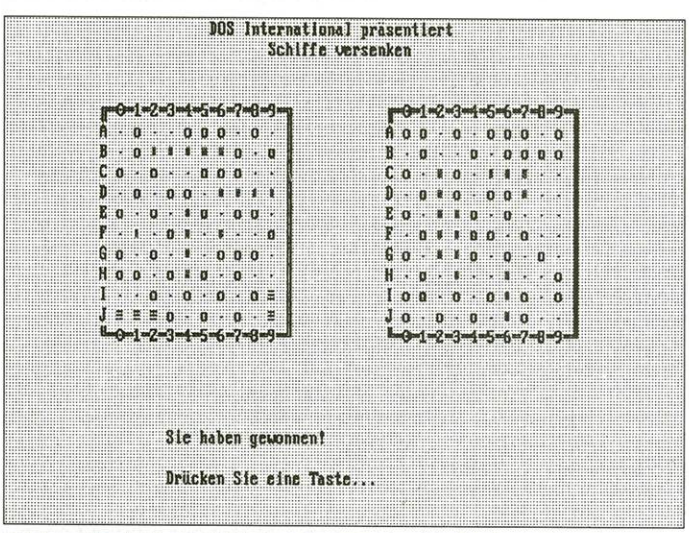


Bild 2. Sind alle Schiffe des Gegners versenkt, kürt das Programm den Sieger

de Anleitung genügt. Probieren Sie es ruhig aus und lesen Sie dann weiter, um etwas über den Aufbau des Programms zu erfahren.

Das Programm besteht aus einigen Gruppen von Prozeduren. Jede Gruppe erledigt eine bestimmte Aufgabe. Eine wichtige Aufgabe ist es, Züge zu lesen und zu bearbeiten. Diese Aufgabe wird nach der Quelle der Züge aufgeteilt: Es gibt eine Gruppe von Prozeduren für Züge von der Tastatur, eine zweite für die Züge des PC und eine dritte Gruppe für Züge von der seriellen Schnittstelle. Weitere Gruppen von Prozeduren zeigen die Spielfelder auf dem Bildschirm und überwachen den Zustand der Schiffe und des Spiels.

Die serielle Schnittstelle wird in diesem Programm zu zwei verschiedenen Zwecken eingesetzt. Sie wird natürlich benutzt, um Informationen wie einen Zug zu übertragen, aber ebenso für die Synchronisation der beiden Programme. Dies beginnt gleich nach dem Programmstart, wenn die beiden PCs »aushandeln«, wer zuerst spielen wird. Jedes Programm schickt eine zufällig gewählte Zahl, entweder eine »1« oder eine »2«, und liest anschließend die Zahl, die vom anderen PC gesendet wurde. Sind die Zahlen unterschiedlich, so ist die Entscheidung getroffen: Der PC, der die »1« gesendet hat, spielt zuerst. Falls die beiden Zahlen gleich sind, wird das Ganze wiederholt, bis irgendwann zwei unterschiedliche Zahlen gewählt werden.

Danach müssen die beiden Spieler ihre Schiffe plazieren. Da das eigentliche Spiel erst dann beginnen darf, wenn beide Spieler alle Schiffe gesetzt haben, sendet das Programm am Ende dieser Aufbauphase eine Bestätigung und wartet anschließend auf die Bestätigung vom Gegner. Erst wenn diese eintrifft, gehts in die Spielphase über.

Ein Zug wird zunächst über die Tastatur eingegeben. Da der PC in dieser Spielart die Positionen der Schiffe des Gegners nicht kennt, wird der Zug zum anderen Computer geschickt. Dieser antwortet entweder, daß an dieser Zelle bereits geschossen wurde, oder daß die Zelle noch frei ist. Im ersten Fall wird die Eingabe wiederholt. Im zweiten Fall ist der Zug gültig und wird wieder zum Gerät des Gegners gesendet. Diesmal kommt eine Antwort, die besagt, was der Schuß bewirkt hat: Ein Treffer, ein Fehlschlag oder ein Treffer, mit dem ein Schiff versenkt wurde. Das gegnerische Feld auf dem Bildschirm wird entsprechend aktualisiert, gegebenenfalls wird eine Meldung ausgegeben, und die beiden PCs tauschen die Rollen, um den Gegner schießen zu lassen.

Der PC entwickelt eine spezielle Strategie

Der Computer als Spieler fordert ganz andere Prozeduren, um die gegnerischen Züge zu generieren. Diese Züge werden vom Rechner nach einer Strategie festgelegt, um dem PC zumindest eine Chance zu geben, seinen intelligenteren, menschlichen Gegner zu besiegen.

Die Strategie ist in fünf Stufen aufgeteilt, die jeweils eine Kennzahl haben. Diese Kennzahlen finden Sie in der Prozedur »getmove« wieder, die die Hauptprozedur dieser Gruppe darstellt. Stufe 0 wird nur beim ersten Zug benutzt und wählt eine Zelle ganz zufällig aus. Stufe 1 ist eine Suchstrategie, die von der Zelle, die in Stufe 0 ausgewählt wurde, ausgeht. In dieser Stufe wird das Spielfeld durch zwei schräge Schießlinien aufgeteilt, so daß es nie mehr als vier Zellen in einem zusammenhängenden Bereich gibt. Dadurch wird gewährleistet, daß mindestens das größte Schiff angeschossen wird; die Chancen sind ebenfalls gut, daß zwei oder drei andere Schiffe entdeckt werden. Durch den zufälligen Anfangspunkt dieser Suchstrategie ist es nicht möglich, daß Sie als Gegner des Computers Ihre Schiffe »möglichst aus dem Weg« plazieren – Sie können unmöglich wissen, wo »aus dem Weg« sein wird.

Ein weiterer Zweig der Strategie versenkt ein Schiff, nachdem ein Treffer erzielt wurde. Dieser Zweig ist relativ einfach. In Stufe 2 wird der Treffer von weiteren Schüssen umgangen, bis ein zweiter Treffer gelandet ist. Damit ist die Orientierung (senkrecht oder waagerecht) des Schiffs eindeutig. Stufe 3 schießt links beziehungsweise oben von den ersten zwei Treffern, bis entweder ein Fehlschlag berichtet oder die Grenze des Spielfelds erreicht wird. Stufe 4 vervollständigt das Versenken durch Schießen nach rechts beziehungsweise nach unten.

Eine weitere Stufe wird eingesetzt, wenn die Suchstrategie von Stufe 1 komplett durchgeführt wurde, ohne daß alle Schiffe versenkt wurden. Diese Stufe sucht nach Bereichen, die groß genug sind, um ein Schiff zu enthalten, und schießt in der Mitte des gefundenen Bereichs. Die Suche beginnt jedesmal von einem neuen, zufällig gewählten Anfangspunkt, um zu vermeiden, daß der Gegner die Strategie kennenlernt.

Noch ein Wort zum Schluß:

Verzweifeln Sie nicht, wenn Sie beispielsweise Ihren Flugzeugträger waagerecht auf der Position »B6« unterbringen wollen, das Programm jedoch immer wieder die Eingabe der Position verlangt. Das Programm »weiß« selbstverständlich, wie groß die Schiffe sind (Tabelle). Bei der Position dieses Beispiels wäre ein Feld des Flugzeugträgers außerhalb des Spielfelds plaziert, was natürlich in keiner Weise den Regeln entspricht.

Wir wünschen Ihnen viel Spaß beim Spielen!

(Thomas Little/hw)

Die Schiffe und ihre Größen

5 Felder: Flugzeugträger	■■■■■
4 Felder: Kreuzer	■■■■
3 Felder: U-Boot	■■■
3 Felder: Schlachtschiff	■■■
2 Felder: Zerstörer	■■

Die Tabelle gibt Ihnen eine Übersicht über den Platzbedarf Ihrer Schiffe

```
Name..... SCHIFFE.BAS

Sprache..... Quick Basic ab Version 4.5

Grafikkarte..... VGA

Besonderheiten.. Sie spielen entweder gegen den Computer
                 oder eine Person an einem anderen PC

[XK3]    1: 'Programm: schiffe.bas
[G17]    2: 'Schiffe versenken gegen den Computer oder
[847]    3: 'per Datenübertragung gegen eine Person
[SL5]    4: 'Sprache:  Quick Basic ab Version 4.5
[LU3]    5: 'Autor:    Thomas Little
[EP4]    6: '(c)1991 DMV Widuch GmbH & Co. KG
[ØØØ]    7:
[V69]    8: DECLARE SUB recordhit (row!, col!, n, shipp
            os() AS INTEGER)
[EAB]    9: DECLARE SUB getmoveremote (board() AS INTEG
            ER, n, shippos() AS INTEGER)
```

Listing. »Schiffe.bas« bringt ein beliebtes Papierspiel in den PC

```
[RH4]   1Ø: DECLARE FUNCTION connect! ()
[3M9]   11: DECLARE FUNCTION CheckMove! (row!, col!, bo
            ard() AS INTEGER)
[3C5]   12: DECLARE SUB sendmove (row!, col!)
[TR4]   13: DECLARE FUNCTION getreply! ()
[H99]   14: DECLARE FUNCTION CheckHit! (row!, col!, boa
            rd() AS INTEGER)
[5E5]   15: DECLARE SUB sendquery (row!, col!)
[YW3]   16: DECLARE SUB shipsunk ()
[7PA]   17: DECLARE SUB getmove (board() AS INTEGER, n,
             shippos() AS INTEGER)
[RW5]   18: DECLARE SUB getmoveinit (row!, col!)
[HHB]   19: DECLARE SUB getmovesearch (row!, col!, rbas
            e!, cbase!, board() AS INTEGER, n)
[ENC]   2Ø: DECLARE SUB getmovesurround (row!, col!, rb
            ase!, cbase!, board() AS INTEGER, n)
[RTB]   21: DECLARE SUB getmovedestroy1 (row!, col!, sh
            prow!, shpcol!, row2!, col2!, n)
[VTB]   22: DECLARE SUB getmovedestroy2 (row!, col!, sh
            prow!, shpcol!, row2!, col2!, n)
[PAC]   23: DECLARE SUB setships (board() AS INTEGER, s
            hipsize!(), shippos() AS INTEGER)
[KEC]   24: DECLARE SUB getmoveinteractive (board() AS
            INTEGER, n, shippos() AS INTEGER)
[47H]   25: DECLARE SUB setshipsinteractive (board() AS
             INTEGER, ship() AS STRING, shipsize!(), sh
            ippos() AS INTEGER)
[D46]   26: DECLARE SUB init (ship$(), shipsize!())
[WW5]   27: DECLARE SUB getposition (row!, col!)
[T8E]   28: DECLARE FUNCTION checkposition! (board() AS
             INTEGER, row!, col!, length!, vh AS STRING
            )
[NM6]   29: DECLARE SUB initboard (board() AS INTEGER)
[Z2B]   3Ø: DECLARE SUB ShowBoard (board() AS INTEGER,
            x AS INTEGER, y AS INTEGER)
[ØØØ]   31:
[M89]   32: DIM MyBoard(12, 12) AS INTEGER, HisBoard(12
            , 12) AS INTEGER
[UZ4]   33: DIM SHARED ship$(5), shipsize(5)
[MM9]   34: DIM myshippos(5, 5) AS INTEGER, hisshippos(
            5, 5) AS INTEGER
[ØØØ]   35:
[EX3]   36: CONST FALSE = Ø, TRUE = -1
[NVC]   37: CONST empty = Ø, hg = 1, vg = 2, lo = 3, ro
             = 4, lu = 5, ru = 6, shp = 7, hit = 8, mis
            s = 9
[ØLA]   38: CONST boardtop = 5, myleft = 12, hisleft =
            46, msgline1 = 21, msgcol = 2Ø
[ØØØ]   39:
[4D5]   4Ø: COMMON SHARED GameOver AS INTEGER
[Z65]   41: COMMON SHARED ComGame AS INTEGER
[ØØØ]   42:
[HE1]   43: COLOR 7, 1
[PNØ]   44: CLS
[B75]   45: i = INSTR(UCASE$(COMMAND$), "COM")
[252]   46: IF (i <> Ø) THEN
[ØC2]   47:   ComGame = TRUE
[9SA]   48:   LOCATE 1Ø, 2Ø: PRINT "Verbindung mit zwei
            tem PC wird aufgebaut...";
[Y69]   49:   OPEN MID$(COMMAND$, i, 4) + ":12ØØ,N,8,1"
             FOR RANDOM AS #1 LEN = 256
[XM2]   5Ø:   order = connect
[ZRØ]   51: ELSE
[3L2]   52:   ComGame = FALSE
[8Ø1]   53: END IF
[ZM4]   54: CALL init(ship$(), shipsize())
[CP2]   55: PlayAgain = TRUE
[PC4]   56: DO WHILE (PlayAgain = TRUE)
[PNØ]   57:   CLS
[HY9]   58:   COLOR 15: LOCATE 1, 25: PRINT "DOS Intern
            ational präsentiert";
[KM7]   59:   COLOR 14: LOCATE 2, 32: PRINT "Schiffe ve
            rsenken";
[K61]   6Ø:   COLOR 7
[US2]   61:   RANDOMIZE TIMER
[914]   62:   CALL initboard(MyBoard())
[384]   63:   CALL initboard(HisBoard())
[YØ7]   64:   CALL ShowBoard(MyBoard(), myleft, boardto
            p)
[7B7]   65:   CALL ShowBoard(HisBoard(), hisleft, board
            top)
[9AB]   66:   CALL setshipsinteractive(MyBoard(), ship$
            (), shipsize(), myshippos())
[KT3]   67:   IF (ComGame = TRUE) THEN
[B54]   68:     DO UNTIL (getreply = TRUE)
[HSØ]   69:     LOOP
[ZRØ]   7Ø:   ELSE
[7B8]   71:     CALL setships(HisBoard(), shipsize(), h
            isshippos())
[8Ø1]   72:   END IF
[MØ4]   73:   myships = 5: hisships = 5
[4M2]   74:   GameOver = FALSE
[X34]   75:   DO UNTIL (GameOver = TRUE)
[NU2]   76:     IF (order = 1) THEN
[E2A]   77:       CALL getmoveinteractive(HisBoard(), h
            isships, hisshippos())
[TB3]   78:       IF (hisships = Ø) THEN
[UK8]   79:         LOCATE msgline1, msgcol: PRINT "Sie
             haben gewonnen!";
[GDA]   8Ø:         LOCATE msgline1 + 2, msgcol: PRINT
            "Drücken Sie eine Taste...";
[6H3]   81:         WHILE (INKEY$ = ""): WEND
[8Ø1]   82:       END IF
[7B7]   83:       CALL ShowBoard(HisBoard(), hisleft, b
            oardtop)
[8Ø1]   84:     END IF
[KT3]   85:     IF (ComGame = TRUE) THEN
[YD8]   86:       CALL getmoveremote(MyBoard(), myships
            , myshippos())
[ZRØ]   87:     ELSE
[9M7]   88:       CALL getmove(MyBoard(), myships, mysh
            ippos())
[8Ø1]   89:     END IF
[YØ7]   9Ø:     CALL ShowBoard(MyBoard(), myleft, board
            top)
[L93]   91:     IF (myships = Ø) THEN
[V78]   92:       LOCATE msgline, msgcol: PRINT "Sie ha
            ben verloren!";
[1DA]   93:       LOCATE msgline + 2, msgline: PRINT "D
            rücken Sie eine Taste...";
[6H3]   94:       WHILE (INKEY$ = ""): WEND
[8Ø1]   95:     END IF
[3B1]   96:     order = 1
[HSØ]   97:   LOOP
[YC7]   98:   CLS : LOCATE 1Ø, 25: PRINT "Nochmal spiel
            en? ";
[9RØ]   99:   i$ = ""
[K74]   1ØØ:  DO UNTIL (i$ = "J") OR (i$ = "N")
[3T2]   1Ø1:    i$ = UCASE$(INKEY$)
[HSØ]   1Ø2:  LOOP
[C65]   1Ø3:  IF (i$ = "N") THEN PlayAgain = FALSE
[HSØ]   1Ø4: LOOP
[4MØ]   1Ø5: END
[ØØØ]   1Ø6:
[RW7]   1Ø7: FUNCTION CheckHit (row, col, board() AS INT
            EGER)
[KT3]   1Ø8:   IF (ComGame = TRUE) THEN
[M24]   1Ø9:     CALL sendquery(row, col)
[1U5]   11Ø:     CheckHit = getreply: EXIT FUNCTION
[ZRØ]   111:   ELSE
[GP4]   112:     IF (board(row, col) <> shp) THEN
[E45]   113:       CheckHit = FALSE: EXIT FUNCTION
[ZRØ]   114:     ELSE
[AØ5]   115:       CheckHit = TRUE: EXIT FUNCTION
[8Ø1]   116:     END IF
[8Ø1]   117:   END IF
[R52]   118: END FUNCTION
[ØØØ]   119:
[YX7]   12Ø: FUNCTION CheckMove (row, col, board() AS IN
            TEGER)
[KT3]   121:   IF (ComGame = TRUE) THEN
[NS3]   122:     CALL sendmove(row, col)
[3Ø6]   123:     CheckMove = getreply: EXIT FUNCTION
[ZRØ]   124:   ELSE
[V99]   125:     IF (board(row, col) <> empty) AND (boar
            d(row, col) <> shp) THEN
[ZC5]   126:       CheckMove = FALSE: EXIT FUNCTION
[ZRØ]   127:     ELSE
[S95]   128:       CheckMove = TRUE: EXIT FUNCTION
[8Ø1]   129:     END IF
[8Ø1]   13Ø:   END IF
[R52]   131: END FUNCTION
[ØØØ]   132:
[2EC]   133: FUNCTION checkposition (board() AS INTEGER,
             row, col, length, vh AS STRING)
[MA2]   134:   IF vh$ = "W" THEN
[KN3]   135:     IF col + length > 11 THEN
[RH3]   136:       checkposition = FALSE
[FA2]   137:       EXIT FUNCTION
[8Ø1]   138:     END IF
[1S4]   139:     FOR k = col TO col + length - 1
[8T4]   14Ø:       IF board(row, k) <> empty THEN
[RH3]   141:         checkposition = FALSE
[FA2]   142:         EXIT FUNCTION
[8Ø1]   143:       END IF
[ZSØ]   144:     NEXT
[ZRØ]   145:   ELSE
[2R3]   146:     IF row + length > 11 THEN
[RH3]   147:       checkposition = FALSE
[FA2]   148:       EXIT FUNCTION
[8Ø1]   149:     END IF
[VV4]   15Ø:     FOR k = row TO row + length - 1
[FS4]   151:       IF board(k, col) <> empty THEN
[RH3]   152:         checkposition = FALSE
[FA2]   153:         EXIT FUNCTION
[8Ø1]   154:       END IF
[ZSØ]   155:     NEXT
[8Ø1]   156:   END IF
[KG3]   157:   checkposition = TRUE
[R52]   158: END FUNCTION
[ØØØ]   159:
[HV2]   16Ø: FUNCTION connect
[US2]   161:   RANDOMIZE TIMER
[Y22]   162:   me = 1: him = 1
```

```
[R23]  163:    DO UNTIL (me <> him)
[YY4]  164:      IF (RND < .5) THEN me = 1 ELSE me = 2
[V33]  165:      PRINT #1, CHR$(me + 48);
[XR2]  166:      him$ = INPUT$(1, #1)
[154]  167:      him = ASC(LEFT$(him$, 1)) - 48
[HSØ]  168:    LOOP
[FØ2]  169:    connect = me
[R52]  17Ø: END FUNCTION
[ØØØ]  171:
[HKA]  172: SUB getmove (board() AS INTEGER, n, shippos
       () AS INTEGER) STATIC
[N32]  173: DIM nmoves(4)
[RT2]  174:   SELECT CASE phase
[AXØ]  175:     CASE Ø
[TA4]  176:       CALL getmoveinit(row, col)
[FP2]  177:       searchrow = row
[LH2]  178:       searchcol = col
[FW1]  179:       nmoves(1) = Ø
[GP4]  18Ø:       IF (board(row, col) <> shp) THEN
[F91]  181:         phase = 1
[ZRØ]  182:       ELSE
[T12]  183:         shprow = row
[HY1]  184:         shpcol = col
[GW1]  185:         nmoves(2) = Ø
[H91]  186:         phase = 2
[8Ø1]  187:       END IF
[DXØ]  188:     CASE 1
[M61]  189: phase1:
[36B]  19Ø:       CALL getmovesearch(row, col, searchro
       w, searchcol, board(), nmoves(1))
[TM4]  191:       IF (board(row, col) = shp) THEN
[T12]  192:         shprow = row
[HY1]  193:         shpcol = col
[GW1]  194:         nmoves(2) = Ø
[H91]  195:         phase = 2
[8Ø1]  196:       END IF
[GXØ]  197:     CASE 2
[DHA]  198:       CALL getmovesurround(row, col, shprow
       , shpcol, board(), nmoves(2))
[TM4]  199:       IF (board(row, col) = shp) THEN
[BM1]  2ØØ:         row2 = row
[HG1]  2Ø1:         col2 = col
[HW1]  2Ø2:         nmoves(3) = Ø
[L91]  2Ø3:         phase = 3
[8Ø1]  2Ø4:       END IF
[LXØ]  2Ø5:     CASE 3
[AXA]  2Ø6:       CALL getmovedestroy1(row, col, shprow
       , shpcol, row2, col2, nmoves(3))
[NC8]  2Ø7:       IF (board(row, col) <> shp) OR (row =
        Ø) OR (col = Ø) THEN
[KW1]  2Ø8:         nmoves(4) = Ø
[N91]  2Ø9:         phase = 4
[8Ø1]  21Ø:       END IF
[996]  211:       IF (row = Ø) OR (col = Ø) THEN GOTO p
       hase4
[PXØ]  212:     CASE 4
[X61]  213: phase4:
[FXA]  214:       CALL getmovedestroy2(row, col, shprow
       , shpcol, row2, col2, nmoves(4))
[2G8]  215:       IF (board(row, col) <> shp) OR (row >
        1Ø) OR (col > 1Ø) THEN
[F91]  216:         phase = 1
[S92]  217:         CALL shipsunk
[8Ø1]  218:       END IF
[FE6]  219:       IF (row > 1Ø) OR (col > 1Ø) THEN GOTO
        phase1
[GS1]  22Ø:   END SELECT
[X1A]  221:   LOCATE msgline1, msgcol: PRINT "Ich schie
       sse auf Position: ";
[G15]  222:   PRINT CHR$(row + 64); CHR$(col + 47);
[681]  223:   SLEEP (1)
[CN6]  224:   LOCATE msgline1, msgcol: PRINT SPACE$(59)
       ;
[625]  225:   IF (board(row, col) = empty) THEN
[FC3]  226:     board(row, col) = miss
[ZRØ]  227:   ELSE
[443]  228:     board(row, col) = hit
[U56]  229:     CALL recordhit(row, col, n, shippos())
[8Ø1]  23Ø:   END IF
[HA1]  231: END SUB
[ØØØ]  232:
[6UA]  233: SUB getmovedestroy1 (row, col, shprow, shpc
       ol, row2, col2, n) STATIC
[232]  234:   IF (n = Ø) THEN
[FM3]  235:     IF (shprow = row2) THEN
[6H3]  236:       IF (shpcol < col2) THEN
[HP2]  237:         lowpos = shpcol
[ZRØ]  238:       ELSE
[U32]  239:         lowpos = col2
[8Ø1]  24Ø:       END IF
[ZRØ]  241:     ELSE
[EM3]  242:       IF (shprow < row2) THEN
[VS2]  243:         lowpos = shprow
[ZRØ]  244:       ELSE
[K52]  245:         lowpos = row2
[8Ø1]  246:       END IF
[8Ø1]  247:     END IF
[8Ø1]  248:   END IF
```

```
[U11]  249:   n = n + 1
[FM3]  25Ø:   IF (shprow = row2) THEN
[YE2]  251:     col = lowpos - n
[ZRØ]  252:   ELSE
[8H2]  253:     row = lowpos - n
[8Ø1]  254:   END IF
[SDA]  255:   IF (board(row, col) <> empty) AND (board(
       row, col) <> shp) THEN col = Ø
[HA1]  256: END SUB
[ØØØ]  257:
[AUA]  258: SUB getmovedestroy2 (row, col, shprow, shpc
       ol, row2, col2, n) STATIC
[232]  259:   IF (n = Ø) THEN
[FM3]  26Ø:     IF (shprow = row2) THEN
[6H3]  261:       IF (shpcol < col2) THEN
[B92]  262:         highpos = col2
[ZRØ]  263:       ELSE
[7T2]  264:         highpos = shpcol
[8Ø1]  265:       END IF
[ZRØ]  266:     ELSE
[EM3]  267:       IF (shprow < row2) THEN
[XB2]  268:         highpos = row2
[ZRØ]  269:       ELSE
[ZU2]  27Ø:         highpos = shprow
[8Ø1]  271:       END IF
[8Ø1]  272:     END IF
[8Ø1]  273:   END IF
[U11]  274:   n = n + 1
[FM3]  275:   IF (shprow = row2) THEN
[9L2]  276:     col = highpos + n
[ZRØ]  277:   ELSE
[LN2]  278:     row = highpos + n
[8Ø1]  279:   END IF
[CFA]  28Ø:   IF (board(row, col) <> empty) AND (board(
       row, col) <> shp) THEN col = 1Ø
[HA1]  281: END SUB
[ØØØ]  282:
[AD4]  283: SUB getmoveinit (row, col)
[743]  284:   row = INT(1Ø * RND + 1)
[X13]  285:   col = INT(1Ø * RND + 1)
[HA1]  286: END SUB
[ØØØ]  287:
[R3B]  288: SUB getmoveinteractive (board() AS INTEGER,
        n, shippos() AS INTEGER)
[LP2]  289:   GoodMove = FALSE
[S34]  29Ø:   DO UNTIL (GoodMove = TRUE)
[KK9]  291:     LOCATE msgline1, msgcol: PRINT "Schiess
       en auf Position: ";
[GB4]  292:     CALL getposition(row, col)
[PX6]  293:     IF (CheckMove(row, col, board()) = FALS
       E) THEN
[1WE]  294:       LOCATE msgline1 + 1, msgcol: PRINT CH
       R$(7); "Sie haben bereits auf diese Positio
       n geschossen";
[681]  295:       SLEEP (1)
[627]  296:       LOCATE msgline1 + 1, msgcol: PRINT SP
       ACE$(59);
[ZRØ]  297:     ELSE
[1N2]  298:       GoodMove = TRUE
[8Ø1]  299:     END IF
[HSØ]  3ØØ:   LOOP
[NR6]  3Ø1:   IF CheckHit(row, col, board()) = FALSE TH
       EN
[FC3]  3Ø2:     board(row, col) = miss
[ZRØ]  3Ø3:   ELSE
[443]  3Ø4:     board(row, col) = hit
[U56]  3Ø5:     CALL recordhit(row, col, n, shippos())
[8Ø1]  3Ø6:   END IF
[HA1]  3Ø7: END SUB
[ØØØ]  3Ø8:
[LZ9]  3Ø9: SUB getmoveremote (board() AS INTEGER, n, s
       hippos() AS INTEGER)
[CN6]  31Ø:   LOCATE msgline1, msgcol: PRINT SPACE$(59)
       ;
[T49]  311:   LOCATE msgline1, msgcol: PRINT "Warten au
       f den Gegner...";
[T82]  312:   row = Ø: col = Ø
[C89]  313:   DO UNTIL (board(row, col) = empty) OR (bo
       ard(row, col) = shp)
[9RØ]  314:     i$ = ""
[FM2]  315:     DO UNTIL (i$ = "M")
[292]  316:       i$ = INPUT$(1, #1)
[HSØ]  317:     LOOP
[5Ø4]  318:     row = ASC(INPUT$(1, #1)) - 48
[VX3]  319:     col = ASC(INPUT$(1, #1)) - 48
[269]  32Ø:     IF (board(row, col) = empty) OR (board(
       row, col) = shp) THEN
[PU1]  321:       PRINT #1, "T";
[ZRØ]  322:     ELSE
[8U1]  323:       PRINT #1, "F";
[8Ø1]  324:     END IF
[HSØ]  325:   LOOP
[5M2]  326:   DO UNTIL (i$ = "H")
[292]  327:     i$ = INPUT$(1, #1)
[HSØ]  328:   LOOP
[CN6]  329:   LOCATE msgline1, msgcol: PRINT SPACE$(59) ;
```

Listing. »Schiffe.bas« bringt ein beliebtes Papierspiel in den PC

```
[5Ø4]  33Ø:    row = ASC(INPUT$(1, #1)) - 48
[VX3]  331:    col = ASC(INPUT$(1, #1)) - 48
[2SD]  332:    LOCATE msgline1, msgcol: PRINT "Gegener s
           chießt auf "; CHR$(row + 64); CHR$(col + 47
           );
[681]  333:    SLEEP (1)
[CN6]  334:    LOCATE msgline1, msgcol: PRINT SPACE$(59)
           ;
[625]  335:    IF (board(row, col) = empty) THEN
[8U1]  336:      PRINT #1, "F";
[FC3]  337:      board(row, col) = miss
[ZRØ]  338:    ELSE
[PU1]  339:      PRINT #1, "T";
[443]  34Ø:      board(row, col) = hit
[U56]  341:      CALL recordhit(row, col, n, shippos())
[8Ø1]  342:    END IF
[HA1]  343: END SUB
[ØØØ]  344:
[KØA]  345: SUB getmovesearch (row, col, rbase, cbase,
           board() AS INTEGER, n)
[K52]  346:    IF (n < 2Ø) THEN
[GTA]  347:      DO UNTIL (board(row, col) = empty) OR (
           board(row, col) = shp) OR (n > 2Ø)
[U11]  348:        n = n + 1
[NP3]  349:        col = cbase + INT(n * 5.5)
[1U2]  35Ø:        row = INT(col / 11)
[DB3]  351:        col = (col - 1) MOD 1Ø + 1
[6M4]  352:        row = (row + rbase - 1) MOD 1Ø + 1
[HSØ]  353:      LOOP
[8Ø1]  354:    END IF
[U52]  355:    IF (n > 2Ø) THEN
[7D3]  356:      cbase = INT(1Ø * RND + 1)
[5E3]  357:      rbase = INT(1Ø * RND + 1)
[VR2]  358:      IF (RND >= .5) THEN
[7D3]  359:        FOR row = rbase TO 1Ø
[6LØ]  36Ø:          i = Ø
[193]  361:          FOR col = cbase TO 1Ø
[9U7]  362:            IF (board(row, col) = empty) THEN
           i = i + 1 ELSE i = Ø
[ØR5]  363:            IF (i >= 3) THEN col = col - 1: E
           XIT SUB
[ZSØ]  364:          NEXT
[6LØ]  365:          i = Ø
[AG3]  366:          FOR col = 1 TO cbase - 1
[9U7]  367:            IF (board(row, col) = empty) THEN
           i = i + 1 ELSE i = Ø
[ØR5]  368:            IF (i >= 3) THEN col = col - 1: E
           XIT SUB
[ZSØ]  369:          NEXT
[ZSØ]  37Ø:        NEXT
[3L3]  371:        FOR row = 1 TO rbase - 1
[6LØ]  372:          i = Ø
[193]  373:          FOR col = cbase TO 1Ø
[9U7]  374:            IF (board(row, col) = empty) THEN
           i = i + 1 ELSE i = Ø
[ØR5]  375:            IF (i >= 3) THEN col = col - 1: E
           XIT SUB
[ZSØ]  376:          NEXT
[6LØ]  377:          i = Ø
[AG3]  378:          FOR col = 1 TO cbase - 1
[9U7]  379:            IF (board(row, col) = empty) THEN
           i = i + 1 ELSE i = Ø
[ØR5]  38Ø:            IF (i >= 3) THEN col = col - 1: E
           XIT SUB
[ZSØ]  381:          NEXT
[ZSØ]  382:        NEXT
[ZRØ]  383:      ELSE
[193]  384:        FOR col = cbase TO 1Ø
[6LØ]  385:          i = Ø
[7D3]  386:          FOR row = rbase TO 1Ø
[9U7]  387:            IF (board(row, col) = empty) THEN
           i = i + 1 ELSE i = Ø
[8U5]  388:            IF (i >= 3) THEN row = row - 1: E
           XIT SUB
[ZSØ]  389:          NEXT
[6LØ]  39Ø:          i = Ø
[3L3]  391:          FOR row = 1 TO rbase - 1
[9U7]  392:            IF (board(row, col) = empty) THEN
           i = i + 1 ELSE i = Ø
[8U5]  393:            IF (i >= 3) THEN row = row - 1: E
           XIT SUB
[ZSØ]  394:          NEXT
[ZSØ]  395:        NEXT
[AG3]  396:        FOR col = 1 TO cbase - 1
[6LØ]  397:          i = Ø
[7D3]  398:          FOR row = rbase TO 1Ø
[9U7]  399:            IF (board(row, col) = empty) THEN
           i = i + 1 ELSE i = Ø
[8U5]  4ØØ:            IF (i >= 3) THEN row = row - 1: E
           XIT SUB
[ZSØ]  4Ø1:          NEXT
[6LØ]  4Ø2:          i = Ø
[3L3]  4Ø3:          FOR row = 1 TO rbase - 1
[9U7]  4Ø4:            IF (board(row, col) = empty) THEN
           i = i + 1 ELSE i = Ø
[8U5]  4Ø5:            IF (i >= 3) THEN row = row - 1: E
           XIT SUB
[ZSØ]  4Ø6:          NEXT
[ZSØ]  4Ø7:        NEXT
[8Ø1]  4Ø8:      END IF
[8Ø1]  4Ø9:    END IF
[HA1]  41Ø: END SUB
[ØØØ]  411:
[RØB]  412: SUB getmovesurround (row, col, rbase, cbase
           , board() AS INTEGER, n)
[U32]  413:    SELECT CASE n
[AXØ]  414:      CASE Ø
[RS2]  415:        IF (rbase > 1) THEN
[YC2]  416:          row = rbase - 1
[UV1]  417:          col = cbase
[ZRØ]  418:        ELSE
[KLØ]  419:          n = 1
[6Ø9]  42Ø:          CALL getmovesurround(row, col, rbas
           e, cbase, board(), n)
[8Ø1]  421:        END IF
[DXØ]  422:      CASE 1
[ZU2]  423:        IF (rbase < 1Ø) THEN
[UC2]  424:          row = rbase + 1
[UV1]  425:          col = cbase
[ZRØ]  426:        ELSE
[MLØ]  427:          n = 2
[6Ø9]  428:          CALL getmovesurround(row, col, rbas
           e, cbase, board(), n)
[8Ø1]  429:        END IF
[GXØ]  43Ø:      CASE 2
[TR2]  431:        IF (cbase > 1) THEN
[S82]  432:          col = cbase - 1
[ØØ2]  433:          row = rbase
[ZRØ]  434:        ELSE
[PLØ]  435:          n = 3
[6Ø9]  436:          CALL getmovesurround(row, col, rbas
           e, cbase, board(), n)
[8Ø1]  437:        END IF
[LXØ]  438:      CASE 3
[1U2]  439:        IF (cbase < 1Ø) THEN
[M82]  44Ø:          col = cbase + 1
[ØØ2]  441:          row = rbase
[8Ø1]  442:        END IF
[GS1]  443:    END SELECT
[U11]  444:    n = n + 1
[V99]  445:    IF (board(row, col) <> empty) AND (board(
           row, col) <> shp) THEN
[6Ø9]  446:      CALL getmovesurround(row, col, rbase, c
           base, board(), n)
[8Ø1]  447:    END IF
[HA1]  448: END SUB
[ØØØ]  449:
[FD4]  45Ø: SUB getposition (row, col)
[BRØ]  451:    j$ = ""
[N44]  452:    DO UNTIL j$ >= "A" AND j$ <= "J"
[5T2]  453:      j$ = UCASE$(INKEY$)
[HSØ]  454:    LOOP
[KE1]  455:    PRINT j$;
[8T3]  456:    row = ASC(LEFT$(j$, 1)) - 64
[F14]  457:    DO UNTIL j$ >= "Ø" AND j$ <= "9"
[ØM1]  458:      j$ = INKEY$
[HSØ]  459:    LOOP
[KE1]  46Ø:    PRINT j$;
[681]  461:    SLEEP (1)
[HE2]  462:    col = ASC(j$) - 47
[HA1]  463: END SUB
[ØØØ]  464:
[123]  465: FUNCTION getreply
[EE3]  466:    answer$ = INPUT$(1, #1)
[X33]  467:    IF answer$ = "T" THEN
[4M2]  468:      getreply = TRUE
[ZRØ]  469:    ELSE
[PN2]  47Ø:      getreply = FALSE
[8Ø1]  471:    END IF
[R52]  472: END FUNCTION
[ØØØ]  473:
[RM4]  474: SUB init (ship$(), shipsize())
[RX6]  475:    shipsize(Ø) = 5: ship$(Ø) = "Flugzeugträg
           er"
[4G5]  476:    shipsize(1) = 4: ship$(1) = "Kreuzer"
[3P6]  477:    shipsize(2) = 3: ship$(2) = "Schlachtschi
           ff"
[A95]  478:    shipsize(3) = 3: ship$(3) = "U-Boot"
[C86]  479:    shipsize(4) = 2: ship$(4) = "Zerstörer"
[HA1]  48Ø: END SUB
[ØØØ]  481:
[U95]  482: SUB initboard (board() AS INTEGER)
[V62]  483:    FOR i = 1 TO 1Ø
[L62]  484:      board(Ø, i) = hg
[E82]  485:      board(11, i) = hg
[492]  486:      board(i, Ø) = vg
[K92]  487:      board(i, 11) = vg
[X62]  488:      FOR j = 1 TO 1Ø
[M13]  489:        board(i, j) = empty
[ZSØ]  49Ø:      NEXT
[ZSØ]  491:    NEXT
[Ø52]  492:    board(Ø, Ø) = lo
[U62]  493:    board(Ø, 11) = ro
[A62]  494:    board(11, Ø) = lu
[KE2]  495:    board(11, 11) = ru
[HA1]  496: END SUB
```

```
[ØØØ]  497:
[4V7]  498: SUB recordhit (row, col, n, shippos() AS IN
            TEGER)
[212]  499:   FOR i = Ø TO 4
[XAG]  5ØØ:     IF (row >= shippos(i, Ø)) AND (row <= s
            hippos(i, 2)) AND (col >= shippos(i, 1)) AN
            D (col <= shippos(i, 3)) THEN
[8T4]  5Ø1:       shippos(i, 4) = shippos(i, 4) - 1
[NY3]  5Ø2:       IF (shippos(i, 4) = Ø) THEN
[CN6]  5Ø3:         LOCATE msgline1, msgcol: PRINT SPAC
            E$(59);
[RMA]  5Ø4:         LOCATE msgline1, msgcol: PRINT ship
            $(i); " wurde versenkt!"; CHR$(7);
[GDA]  5Ø5:         LOCATE msgline1 + 2, msgcol: PRINT
            "Drücken Sie eine Taste...";
[6H3]  5Ø6:         WHILE (INKEY$ = ""): WEND
[CN6]  5Ø7:         LOCATE msgline1, msgcol: PRINT SPAC
            E$(59);
[A27]  5Ø8:         LOCATE msgline1 + 2, msgcol: PRINT
            SPACE$(59);
[221]  5Ø9:         n = n - 1
[MP4]  51Ø:         IF (n = Ø) THEN GameOver = TRUE
[8Ø1]  511:       END IF
[8Ø1]  512:     END IF
[ZSØ]  513:   NEXT
[HA1]  514: END SUB
[ØØØ]  515:
[CU3]  516: SUB sendmove (row, col)
[356]  517:   PRINT #1, "M"; CHR$(row + 48); CHR$(col +
            48);
[HA1]  518: END SUB
[ØØØ]  519:
[8Ø4]  52Ø: SUB sendquery (row, col)
[Y46]  521:   PRINT #1, "H"; CHR$(row + 48); CHR$(col +
            48);
[HA1]  522: END SUB
[ØØØ]  523:
[ØVA]  524: SUB setships (board() AS INTEGER, shipsize(
            ), shippos() AS INTEGER)
[212]  525:   FOR i = Ø TO 4
[F73]  526:     IF INT(2 * RND) = Ø THEN
[ZYØ]  527:       i$ = "W"
[ZRØ]  528:     ELSE
[FYØ]  529:       i$ = "S"
[8Ø1]  53Ø:     END IF
[E42]  531:     PosOK = FALSE
[HS3]  532:     DO UNTIL (PosOK = TRUE)
[743]  533:       row = INT(1Ø * RND + 1)
[X13]  534:       col = INT(1Ø * RND + 1)
[W29]  535:       PosOK = checkposition(board(), row, c
            ol, shipsize(i), i$)
[HSØ]  536:     LOOP
[9D2]  537:     IF (i$ = "W") THEN
[864]  538:       FOR j = Ø TO shipsize(i) - 1
[GL3]  539:         board(row, col + j) = shp
[ZSØ]  54Ø:       NEXT
[D46]  541:       shippos(i, Ø) = row: shippos(i, 1) =
            col
[EM8]  542:       shippos(i, 2) = row: shippos(i, 3) =
            col + shipsize(i) - 1
[KB4]  543:       shippos(i, 4) = shipsize(i)
[ZRØ]  544:     ELSE
[864]  545:       FOR j = Ø TO shipsize(i) - 1
[7L3]  546:         board(row + j, col) = shp
[ZSØ]  547:       NEXT
[D46]  548:       shippos(i, Ø) = row: shippos(i, 1) =
            col
[8H8]  549:       shippos(i, 2) = row + shipsize(i) - 1
            : shippos(i, 3) = col
[KB4]  55Ø:       shippos(i, 4) = shipsize(i)
[8Ø1]  551:     END IF
[ØY6]  552:     CALL ShowBoard(board(), hisleft, boardt
            op)
[ZSØ]  553:   NEXT
[HA1]  554: END SUB
[ØØØ]  555:
[DWF]  556: SUB setshipsinteractive (board() AS INTEGER
            , ship() AS STRING, shipsize(), shippos() A
            S INTEGER)
[212]  557:   FOR i = Ø TO 4
[CN6]  558:     LOCATE msgline1, msgcol: PRINT SPACE$(5
            9);
[627]  559:     LOCATE msgline1 + 1, msgcol: PRINT SPAC
            E$(59);
[KKA]  56Ø:     LOCATE msgline1, msgcol: PRINT ship$(i)
            ; ": Waag- oder senkrecht? ";
[9RØ]  561:     i$ = ""
[Ø94]  562:     DO UNTIL (i$ = "W") OR (i$ = "S")
[YL1]  563:       i$ = INKEY$
[PØ2]  564:       i$ = UCASE$(i$)
[HSØ]  565:     LOOP
[EE1]  566:     PRINT i$;
[E42]  567:     PosOK = FALSE
[HS3]  568:     DO UNTIL (PosOK = TRUE)
[627]  569:       LOCATE msgline1 + 1, msgcol: PRINT SP
            ACE$(59);
[SY6]  57Ø:       LOCATE msgline1 + 1, msgcol + 2 + LEN
            (ship$(i))
[9D2]  571:       IF (i$ = "W") THEN
[9E2]  572:         PRINT "Linkes";
[ZRØ]  573:       ELSE
[ZD2]  574:         PRINT "Oberes";
[8Ø1]  575:       END IF
[2D4]  576:       PRINT " Ende an Position: ";
[GB4]  577:       CALL getposition(row, col)
[W29]  578:       PosOK = checkposition(board(), row, c
            ol, shipsize(i), i$)
[HSØ]  579:     LOOP
[9D2]  58Ø:     IF (i$ = "W") THEN
[864]  581:       FOR j = Ø TO shipsize(i) - 1
[GL3]  582:         board(row, col + j) = shp
[ZSØ]  583:       NEXT
[D46]  584:       shippos(i, Ø) = row: shippos(i, 1) =
            col
[EM8]  585:       shippos(i, 2) = row: shippos(i, 3) =
            col + shipsize(i) - 1
[KB4]  586:       shippos(i, 4) = shipsize(i)
[ZRØ]  587:     ELSE
[864]  588:       FOR j = Ø TO shipsize(i) - 1
[7L3]  589:         board(row + j, col) = shp
[ZSØ]  59Ø:       NEXT
[D46]  591:       shippos(i, Ø) = row: shippos(i, 1) =
            col
[8H8]  592:       shippos(i, 2) = row + shipsize(i) - 1
            : shippos(i, 3) = col
[KB4]  593:       shippos(i, 4) = shipsize(i)
[8Ø1]  594:     END IF
[LP6]  595:     CALL ShowBoard(board(), myleft, boardto
            p)
[ZSØ]  596:   NEXT
[CN6]  597:   LOCATE msgline1, msgcol: PRINT SPACE$(59)
            ;
[627]  598:   LOCATE msgline1 + 1, msgcol: PRINT SPACE$
            (59);
[KT3]  599:   IF (ComGame = TRUE) THEN
[PU1]  6ØØ:     PRINT #1, "T";
[8Ø1]  6Ø1:   END IF
[HA1]  6Ø2: END SUB
[ØØØ]  6Ø3:
[772]  6Ø4: SUB shipsunk
[WD3]  6Ø5:   myships = myships - 1
[L93]  6Ø6:   IF (myships = Ø) THEN
[CN6]  6Ø7:     LOCATE msgline1, msgcol: PRINT SPACE$(5
            9);
[BM8]  6Ø8:     LOCATE msgline1, msgcol: PRINT "Sie hab
            en verloren!";
[GDA]  6Ø9:     LOCATE msgline1 + 2, msgcol: PRINT "Drü
            cken Sie eine Taste...";
[6H3]  61Ø:     WHILE (INKEY$ = ""): WEND
[KK2]  611:     GameOver = TRUE
[8Ø1]  612:   END IF
[HA1]  613: END SUB
[ØØØ]  614:
[4S9]  615: SUB ShowBoard (board() AS INTEGER, x AS INT
            EGER, y AS INTEGER)
[X62]  616:   FOR i = Ø TO 11
[462]  617:     LOCATE y + i, x
[Z62]  618:     FOR j = Ø TO 11
[RN3]  619:       SELECT CASE board(i, j)
[ØT1]  62Ø:         CASE empty
[4T1]  621:           PRINT "ф ";
[E81]  622:         CASE hg
[BL3]  623:           PRINT CHR$(47 + j); "═";
[S91]  624:         CASE vg
[U22]  625:           IF (j = Ø) THEN
[Y23]  626:             PRINT CHR$(64 + i); " ";
[ZRØ]  627:           ELSE
[SF1]  628:             PRINT "╟ "
[8Ø1]  629:           END IF
[U91]  63Ø:         CASE lo
[D22]  631:           PRINT "╔═";
[CA1]  632:         CASE ro
[5R1]  633:           PRINT "╗ ";
[KA1]  634:         CASE lu
[C22]  635:           PRINT "╚═";
[4B1]  636:         CASE ru
[6R1]  637:           PRINT "╝ ";
[3C1]  638:         CASE shp
[A62]  639:           IF (x < 4Ø) THEN
[F71]  64Ø:             COLOR 15
[US1]  641:             PRINT "≡ ";
[K61]  642:             COLOR 7
[ZRØ]  643:           ELSE
[4T1]  644:             PRINT "ф ";
[8Ø1]  645:           END IF
[AB1]  646:         CASE hit
[661]  647:            COLOR 4
[8T1]  648:           PRINT "▪ ";
[K61]  649:           COLOR 7
[YG1]  65Ø:         CASE miss
[E71]  651:           COLOR 14
[TL1]  652:           PRINT "o ";
[K61]  653:           COLOR 7
[GS1]  654:       END SELECT
[ZSØ]  655:     NEXT
[ZSØ]  656:   NEXT
[HA1]  657: END SUB
```

Listing. »Schiffe.bas« bringt ein beliebtes Papierspiel in den PC

SCheck 2.02

Checksummer für die Listing-Eingabe

Die Programme in der DOS International und der Sonderheftreihe DOS Extra werden nach einem speziellen Verfahren mit Prüfsummen versehen und abgedruckt. Beinahe jedes Listing, das Sie in der DOS International abgedruckt sehen, beinhaltet auch die Prüfsummen für SCheck.

Jede Programmzeile in den Listings der DOS 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 Checksummenprogramm »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 1.0« 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.

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.

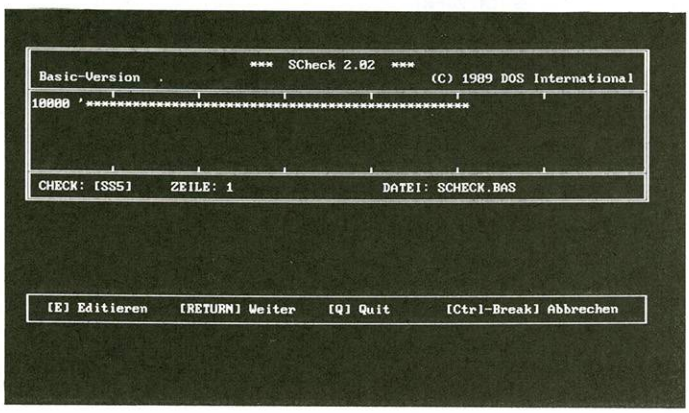
Wie Sie im Bild sehen, zeigt SCheck die aktuelle Zeile stets in einem besonderen Bildschirmfenster an. Die Prüfsumme erscheint links unter dem Fenster. Beim Überprüfen Ihres Listings vergleichen Sie die angezeigte Prüfsumme mit der in der DOS-Ausgabe in eckigen Klammern abgedruckten Prüfsumme. Rechts 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. 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 Löschtaste 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.

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 Programmzeile berechnet SCheck die Prüfsumme und vergleicht 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 zwei weitere Optionen. 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



Der Arbeitsbildschirm von SCheck in Aktion

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 Namenserweiterung »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
```

Verwenden Sie nach dem Abtippen des Listings aus der DOS stets SCheck. Nur dann können Sie sicher sein, daß Ihre Eingaben korrekt sind.

(db)

Name	SCheck.bas
Sprache	GW-Basic und SBasic
Grafikkarte	jede
Besonderheiten	Checksummer für Listings in DOS International und DOS Extra

```
[SS5] 1ØØØØ '*****************************************
      *******
[F29] 1ØØ1Ø '* SCheck - Prüfsummengenerator für DOS-Lis
      tings *
[ML4] 1ØØ2Ø '*
            *
[8Ø7] 1ØØ3Ø '*            Konzeption: Volker Everts
            *
[AC7] 1ØØ4Ø '*         Basic-Version: Markus Zietlow
            *
[XS5] 1ØØ5Ø '*****************************************
      *******
[4UØ] 1ØØ6Ø '
[FL7] 1ØØ7Ø KEY OFF: DEFINT A-O,Q-Z: DEFSNG P: COLOR 7,
      Ø,1
[5G8] 1ØØ8Ø T = 1: TEMP = 2: F = 3'Ein-/Ausgabekanäle f
      estlegen
[332] 1ØØ9Ø FALSE = Ø
[9Ø2] 1Ø1ØØ TRUE = -1
[EY7] 1Ø11Ø ZEICHEN$ = "Ø123456789ABCDEFGHKLMNPRSTUVWXY
      Z"
[9H2] 1Ø12Ø LAUF = FALSE
[6B2] 1Ø13Ø BIN = FALSE
[851] 1Ø14Ø CLS
[9S1] 1Ø15Ø PRINT
[KØ7] 1Ø16Ø PRINT CHR$(2Ø1); STRING$(78, 2Ø5); CHR$(187
      );
[Z5C] 1Ø17Ø PRINT CHR$(186); STRING$(28, 32); "***  SCh
      eck 2.Ø2  ***"; STRING$(29, 32); CHR$(186);
[ØKE] 1Ø18Ø PRINT CHR$(186); " Basic-Version
                                (C) 1989 DOS Internatio
      nal "; CHR$(186);
[C5C] 1Ø19Ø PRINT CHR$(2Ø4); "═══════════════════════
      ═══════════════════════════════════════
      ═══"; CHR$(185);
[DR3] 1Ø2ØØ FOR ZAEHLER = 1 TO 5
[ZY6] 1Ø21Ø   PRINT CHR$(186); STRING$(78, 32); CHR$(18
      6);
[CR2] 1Ø22Ø NEXT ZAEHLER
[84C] 1Ø23Ø PRINT CHR$(2Ø4); "═══════════════════════
      ═══════════════════════════════════════
      ═══"; CHR$(185);
[ØØD] 1Ø24Ø PRINT CHR$(186); " CHECK:           ZEILE:
                          DATEI:
          "; CHR$(186);
[PØ7] 1Ø25Ø PRINT CHR$(2ØØ); STRING$(78, 2Ø5); CHR$(188
      );
[2D2] 1Ø26Ø LOCATE 2Ø, 1
[B17] 1Ø27Ø PRINT CHR$(218); STRING$(78, 196); CHR$(191
      );
[ØØ7] 1Ø28Ø PRINT CHR$(179); STRING$(78, 32); CHR$(179)
      ;
[C27] 1Ø29Ø PRINT CHR$(192); STRING$(78, 196); CHR$(217
      );
[4V1] 1Ø3ØØ COLOR 7
[9F1] 1Ø31Ø Y = 7
[9V6] 1Ø32Ø LOCATE 7, 31: PRINT "SCheck  Version 1.x"
[PR6] 1Ø33Ø LOCATE 8, 31: PRINT "SCheck  Version 2:Ø"
[M7H] 1Ø34Ø LOCATE 21, 4: PRINT "Bitte wählen Sie mit d
      en Pfeiltasten aus und bestätigen Sie durch [Ente
      r] ";
[2PD] 1Ø35Ø COLOR 14, 1: FOR Q = 3Ø TO 5Ø: LOCATE Y, Q:
       PRINT CHR$(SCREEN(Y, Q)): NEXT Q: COLOR 7, Ø
[UP6] 1Ø36Ø TASTE$ = INKEY$: IF TASTE$ = "" THEN 1Ø36Ø
[P96] 1Ø37Ø IF TASTE$ <> CHR$(Ø) + CHR$(72) THEN 1Ø4ØØ
[4ND] 1Ø38Ø IF Y <> 7 THEN FOR Q = 3Ø TO 5Ø: LOCATE Y,
      Q: PRINT CHR$(SCREEN(Y, Q)): NEXT Q: Y = Y - 1
[532] 1Ø39Ø GOTO 1Ø35Ø
[1A6] 1Ø4ØØ IF TASTE$ <> CHR$(Ø) + CHR$(8Ø) THEN 1Ø43Ø
[1ND] 1Ø41Ø IF Y <> 8 THEN FOR Q = 3Ø TO 5Ø: LOCATE Y,
      Q: PRINT CHR$(SCREEN(Y, Q)): NEXT Q: Y = Y + 1
[232] 1Ø42Ø GOTO 1Ø35Ø
[F35] 1Ø43Ø IF TASTE$ <> CHR$(13) THEN 1Ø36Ø
[8N2] 1Ø44Ø FOR Q = 7 TO 8
[U16] 1Ø45Ø   LOCATE Q, 3Ø: PRINT STRING$(21, 32)
[KN1] 1Ø46Ø NEXT Q
[8Y2] 1Ø47Ø VERSION = Y - 6
[GV1] 1Ø48Ø COLOR 7
[NF1] 1Ø49Ø Y = 7
[MA5] 1Ø5ØØ LOCATE 7, 36: PRINT "Standard"
[4A6] 1Ø51Ø LOCATE 8, 34: PRINT "Mit Prüfsumme"
[C65] 1Ø52Ø LOCATE 9, 37: PRINT "Hexdump"
[V7H] 1Ø53Ø LOCATE 21, 4: PRINT "Bitte wählen Sie mit d
      en Pfeiltasten aus und bestätigen Sie durch [Ente
      r] ";
[9PD] 1Ø54Ø COLOR 14, 1: FOR Q = 3Ø TO 5Ø: LOCATE Y, Q:
       PRINT CHR$(SCREEN(Y, Q)): NEXT Q: COLOR 7, Ø
[3R6] 1Ø55Ø TASTE$ = INKEY$: IF TASTE$ = "" THEN 1Ø55Ø
[4B6] 1Ø56Ø IF TASTE$ <> CHR$(Ø) + CHR$(72) THEN 1Ø59Ø
[BND] 1Ø57Ø IF Y <> 7 THEN FOR Q = 3Ø TO 5Ø: LOCATE Y,
      Q: PRINT CHR$(SCREEN(Y, Q)): NEXT Q: Y = Y - 1
[E32] 1Ø58Ø GOTO 1Ø54Ø
[GA6] 1Ø59Ø IF TASTE$ <> CHR$(Ø) + CHR$(8Ø) THEN 1Ø62Ø
[CND] 1Ø6ØØ IF Y <> 9 THEN FOR Q = 3Ø TO 5Ø: LOCATE Y,
      Q: PRINT CHR$(SCREEN(Y, Q)): NEXT Q: Y = Y + 1
[B32] 1Ø61Ø GOTO 1Ø54Ø
[P35] 1Ø62Ø IF TASTE$ <> CHR$(13) THEN 1Ø55Ø
[GN2] 1Ø63Ø FOR Q = 7 TO 9
[126] 1Ø64Ø   LOCATE Q, 3Ø: PRINT STRING$(21, 32)
[TN1] 1Ø65Ø NEXT Q
[WG4] 1Ø66Ø IF Y = 8 THEN LAUF = TRUE
[NA4] 1Ø67Ø IF Y = 9 THEN BIN = TRUE
[RU5] 1Ø68Ø LOCATE 12, 54: PRINT STRING$(25, 32)
[8DF] 1Ø69Ø GOSUB 1315Ø: LOCATE 21, 12: PRINT "Bitte ge
      ben Sie den Dateinamen der gewünschten Datei ein
      ";
[9P8] 1Ø7ØØ Y = 12: X = 54: LAENGE = 24: MENGE$ = "alle
      ": GOSUB 1Ø89Ø
[N6D] 1Ø71Ø IF (TASTE$ = CHR$(27)) OR (Q$ = "") THEN LO
      CATE 12, 54: PRINT STRING$(25, 32): GOTO 1Ø68Ø
[LM3] 1Ø72Ø ON ERROR GOTO 1116Ø
[C74] 1Ø73Ø OPEN Q$ FOR INPUT AS #T
[732] 1Ø74Ø GOTO 1123Ø
[3Ø2] 1Ø75Ø COLOR 12
[2A6] 1Ø76Ø LOCATE 8, 3Ø: PRINT "D A T E I F E H L E R"
[VV1] 1Ø77Ø COLOR 7
[5GC] 1Ø78Ø GOSUB 1315Ø: LOCATE 21, 21: PRINT "Drücken
      Sie bitte eine beliebige Taste ";
[B83] 1Ø79Ø WHILE INKEY$ = ""
[GX2] 1Ø8ØØ   PRINT CHR$(7);
[CE1] 1Ø81Ø WEND
[6U5] 1Ø82Ø LOCATE 8, 3Ø: PRINT STRING$(21, 32)
[842] 1Ø83Ø GOTO 1Ø68Ø
[2VØ] 1Ø84Ø '
[184] 1Ø85Ø '**********************************
[UD6] 1Ø86Ø '* Unterprogramm zur Dateneingabe *
[384] 1Ø87Ø '**********************************
[6VØ] 1Ø88Ø '
[GG1] 1Ø89Ø Q$ = ""
[EF2] 1Ø9ØØ LOCATE Y, X
[FS3] 1Ø91Ø PRINT Q$; CHR$(219);
[GR6] 1Ø92Ø TASTE$ = INKEY$: IF TASTE$ = "" THEN 1Ø92Ø
[ZK5] 1Ø93Ø IF (TASTE$ <> CHR$(8)) THEN 1Ø99Ø
[224] 1Ø94Ø   IF LEN(Q$) = Ø THEN 1Ø92Ø
[X44] 1Ø95Ø   Q$ = LEFT$(Q$, LEN(Q$) - 1)
[6N2] 1Ø96Ø   LOCATE Y, X
[V44] 1Ø97Ø   PRINT Q$; CHR$(219); " ";
[8A2] 1Ø98Ø   GOTO 1Ø92Ø
[T98] 1Ø99Ø IF TASTE$ = CHR$(13) THEN LOCATE Y, X: PRIN
      T Q$; " ";
[639] 11ØØØ IF (TASTE$ = CHR$(13)) OR (TASTE$ = CHR$(27
      )) THEN RETURN
[C55] 11Ø1Ø IF (ASC(TASTE$) < 32) THEN 1Ø92Ø
[NX4] 11Ø2Ø IF MENGE$ <> "zahl" THEN 11Ø5Ø
[ZC8] 11Ø3Ø   IF (ASC(TASTE$) < 48) OR (ASC(TASTE$) > 5
      7) THEN 1Ø92Ø
```

Listing. »SCheck.bas« für GW-Basic

```
[892] 11Ø4Ø   GOTO 11Ø7Ø
[PU4] 11Ø5Ø IF MENGE$ = "alle" THEN 11Ø7Ø
[E86] 11Ø6Ø IF INSTR(MENGE$, TASTE$) = Ø THEN 1Ø92Ø
[YF6] 11Ø7Ø IF LEN(Q$) < LAENGE THEN Q$ = Q$ + TASTE$
[T22] 11Ø8Ø GOTO 1Ø9ØØ
[6H8] 11Ø9Ø Y = 1Ø: X = 65: LAENGE = 5: MENGE$ = "zahl"
      : GOSUB 1Ø89Ø
[BV5] 111ØØ IF (TASTE$ = CHR$(27)) THEN RETURN
[6UØ] 1111Ø '
[NK2] 1112Ø '******************
[TR3] 1113Ø '* Fehler-Routine *
[RK2] 1114Ø '******************
[AUØ] 1115Ø '
[KN1] 1116Ø CLOSE
[ZG2] 1117Ø RESUME 1Ø75Ø
[DUØ] 1118Ø '
[8G2] 1119Ø '******************
[TH3] 112ØØ '* Hauptprogramm *
[4G2] 1121Ø '******************
[BUØ] 1122Ø '
[KP3] 1123Ø COLOR 7: GOSUB 1315Ø
[N43] 1124Ø ON ERROR GOTO Ø
[K62] 1125Ø DATEI$ = Q$
[DF7] 1126Ø GROSS$ = DATEI$: GOSUB 13Ø4Ø: DATEI$ = GROS
      S$
[3W3] 1127Ø I = INSTR(DATEI$, ".")
[6K7] 1128Ø IF I = Ø THEN DATEI$ = DATEI$ + ".": I = LE
      N(DATEI$)
[U64] 1129Ø EXT$ = MID$(DATEI$, I, 4)
[8T4] 113ØØ DATEI$ = LEFT$(DATEI$, I - 1)
[HN5] 1131Ø LOCATE 12, 54: PRINT DATEI$; EXT$
[HB4] 1132Ø TEMPNAME$ = DATEI$ + ".$$$"
[5E6] 1133Ø OPEN TEMPNAME$ FOR OUTPUT AS #TEMP
[CZ7] 1134Ø IF BIN THEN OPEN DATEI$ + ".BIN" FOR OUTPUT
       AS #F
[CP7] 1135Ø IF VERSION = 2 THEN NUMMER = 1 ELSE NUMMER
      = Ø
[KYC] 1136Ø IF BIN THEN GOSUB 1315Ø: LOCATE 21, 25: PRI
      NT ">>> Binärdatei wird erzeugt <<<"
[TE3] 1137Ø WHILE NOT EOF(T)
[X24] 1138Ø   LINE INPUT #T, ZEILE$
[LDB] 11385   WHILE RIGHT$(ZEILE$,1)=" " : ZEILE$=LEFT$
      (ZEILE$,LEN(ZEILE$)-1) : WEND
[333] 1139Ø   WEITER = FALSE
[WT3] 114ØØ   IF WEITER THEN 115ØØ
[WN6] 1141Ø   GOSUB 1182Ø 'Gelesene Zeile anzeigen
[TS4] 1142Ø   IF BIN THEN WEITER = TRUE
[ZT3] 1143Ø   IF WEITER THEN 115ØØ
[AM6] 1144Ø   CH$ = INKEY$: IF CH$ = "" THEN GOTO 1144Ø
[3G3] 1145Ø   ERLAUBT = FALSE
[ZH8] 1146Ø   IF CH$ = CHR$(13) THEN ERLAUBT = TRUE: WE
      ITER = TRUE
[N1C] 1147Ø   IF (CH$ = "e") OR (CH$ = "E") THEN ERLAUB
      T = TRUE: GOSUB 12Ø6Ø'Zeile bearbeiten
[8ME] 1148Ø   IF (CH$ = "q") OR (CH$ = "Q") THEN IF NOT
       LAUF THEN GOSUB 1242Ø: IF SCHLUSS THEN GOTO 1163
      Ø'Quit
[F57] 1149Ø   IF ERLAUBT THEN GOTO 114ØØ ELSE GOTO 1144
      Ø
[8Y3] 115ØØ   PRINT #TEMP, ZEILE$
[DU3] 1151Ø   IF NOT BIN THEN 1161Ø
[D99] 1152Ø   WHILE LEFT$(ZEILE$, 1) = " ": ZEILE$ = MI
      D$(ZEILE$, 2): WEND
[364] 1153Ø   IF ZEILE$ = "" THEN 1161Ø
[K75] 1154Ø   B = VAL("&H" + LEFT$(ZEILE$, 2))
[TSH] 1155Ø   IF (B < Ø) OR (B > 255) THEN GOSUB 1315Ø:
       LOCATE 21, 22: PRINT "+++++ ABBRUCH WEGEN DATENF
      EHLER +++++": LOCATE 25, 1: END
[HN1] 1156Ø   BY = B
[TK3] 1157Ø   PRINT #F, CHR$(BY);
[YA4] 1158Ø   ZEILE$ = MID$(ZEILE$, 3)
[M99] 1159Ø   WHILE LEFT$(ZEILE$, 1) = " ": ZEILE$ = MI
      D$(ZEILE$, 2): WEND
[T92] 116ØØ   GOTO 1153Ø
[DY3] 1161Ø   NUMMER = NUMMER + 1
[8E1] 1162Ø WEND
[S7B] 1163Ø WHILE NOT EOF(T): LINE INPUT #T, ZEILE$: PR
      INT #TEMP, ZEILE$: WEND
[4P1] 1164Ø CLOSE
[8U5] 1165Ø OPEN DATEI$ + EXT$ FOR INPUT AS #1
[XØ6] 1166Ø OPEN DATEI$ + ".BAK" FOR OUTPUT AS #2
[XGB] 1167Ø WHILE NOT EOF(1): LINE INPUT #1, ZEILE$: PR
      INT #2, ZEILE$: WEND: CLOSE
[6D5] 1168Ø OPEN TEMPNAME$ FOR INPUT AS #1
[576] 1169Ø OPEN DATEI$ + EXT$ FOR OUTPUT AS #2
[UGB] 117ØØ WHILE NOT EOF(1): LINE INPUT #1, ZEILE$: PR
      INT #2, ZEILE$: WEND: CLOSE
[713] 1171Ø KILL TEMPNAME$
[GTN] 1172Ø GOSUB 1256Ø: LOCATE 12, 9: PRINT STRING$(11
      , 32): LOCATE 12, 25: PRINT STRING$(21, 32): LOCA
      TE 12, 53: PRINT STRING$(27, 32)'Statuszeile lösc
      hen
[494] 1173Ø IF BIN THEN EXT$ = ".BIN"
[N4H] 1174Ø GOSUB 1315Ø: LOCATE 21, (8Ø - (31 + LEN(DAT
      EI$ + EXT$))) / 2: PRINT "Die Datei '"; DATEI$ +
      EXT$; "' wurde gespeichert."
[CE2] 1175Ø LOCATE 25, 1
[H51] 1176Ø END
[4VØ] 1177Ø '
[TH3] 1178Ø '****************************
[B65] 1179Ø '* Gelesene Zeile anzeigen *
[NH3] 118ØØ '****************************
[2VØ] 1181Ø '
[BBB] 1182Ø IF LAUF THEN CHECK$ = "[" + LEFT$(ZEILE$, 3
      ) + "]": ZEILE$ = MID$(ZEILE$, 5)
[MDN] 1183Ø 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] 1184Ø NEXT Q
[B57] 1185Ø LINKS = 2: OBEN = 6: RECHTS = 79: UNTEN = 1
      Ø
[CX4] 1186Ø LOCATE 12, 26: PRINT NUMMER
[Z45] 1187Ø GOSUB 1256Ø'Fenster löschen
[WX8] 1188Ø GOSUB 1266Ø'Zeile innerhalb des Fensters an
      zeigen
[366] 1189Ø GOSUB 128ØØ'Check-Summe errechnen
[ØM2] 119ØØ LOCATE 12, 1Ø
[HB2] 1191Ø PRINT PS$
[HM7] 1192Ø IF BIN THEN GOTO 12ØØØ'Binärdatei erzeugen
[N8C] 1193Ø IF LAUF THEN GOTO 1197Ø'Übereinstimmung der
       Prüfsummen feststellen
[X74] 1194Ø GOSUB 1315Ø: LOCATE 21, 4
[LKD] 1195Ø PRINT "[E] Editieren    [RETURN] Weiter
      [Q] Quit      [Ctrl-Break] Abbrechen"
[152] 1196Ø GOTO 1199Ø
[ATP] 1197Ø GOSUB 1315Ø: IF PS$ = CHECK$ THEN LOCATE 21
      , (8Ø - (22 + LEN(STR$(NUMMER)))) / 2: PRINT "Prü
      fsumme OK für Zeile"; NUMMER: WEITER = TRUE: GOTO
       12ØØØ
[1NH] 1198Ø LOCATE 21, 4: PRINT CHR$(7); "Prüfsummenfeh
      ler!  [E] Editieren  [RETURN] Weiter  [Ctrl-Break
      ] Abbrechen"
[XM2] 1199Ø LOCATE 12, 15
[KV1] 12ØØØ RETURN
[5UØ] 1201Ø '
[MK2] 1202Ø '******************
[4H3] 1203Ø '* Zeile eingeben *
[PK2] 1204Ø '******************
[9UØ] 1205Ø '
[HRD] 1206Ø GOSUB 1315Ø: LOCATE 21, 17: PRINT "Ändern S
      ie die Programmzeile        [Enter] Ende"
[245] 1207Ø GOSUB 1256Ø'Fenster löschen
[7B5] 1208Ø GOSUB 1266Ø'Zeile darstellen
[FL3] 1209Ø LOCATE OBEN, LINKS
[7G3] 121ØØ X = 1: PRINT "■";
[8E2] 1211Ø CH$ = INKEY$
[FE6] 1212Ø IF LEFT$(CH$, 1) <> CHR$(Ø) THEN GOTO 1222Ø
[4N3] 1213Ø   CH$ = RIGHT$(CH$, 1)
[4NE] 1214Ø   IF CH$ = "M" THEN IF X > LEN(ZEILE$) THEN
       PRINT CHR$(7); : GOTO 1211Ø ELSE GOSUB 1234Ø: X
      = X + 1
[KZC] 1215Ø   IF CH$ = "K" THEN IF X = 1 THEN PRINT CHR
      $(7); : GOTO 1211Ø ELSE GOSUB 1234Ø: X = X - 1
[YZC] 1216Ø   IF CH$ = "H" THEN IF X < 79 THEN PRINT CH
      R$(7); : GOTO 1211Ø ELSE GOSUB 1234Ø: X = X - 78
[YCF] 1217Ø   IF CH$ = "P" THEN IF X + 77 > LEN(ZEILE$)
       THEN PRINT CHR$(7); : GOTO 1211Ø ELSE GOSUB 1234
      Ø: X = X + 78
[TV7] 1218Ø   IF CH$ = "G" THEN IF X <> 1 THEN GOSUB 12
      34Ø: X = 1
[41C] 1219Ø   IF CH$ = "O" THEN IF X <> LEN(ZEILE$) + 1
       THEN GOSUB 1234Ø: X = LEN(ZEILE$) + 1
[8PM] 122ØØ   IF CH$ = "S" THEN IF X <= LEN(ZEILE$) THE
      N ZEILE$ = LEFT$(ZEILE$, X - 1) + MID$(ZEILE$, X
      + 1): GOSUB 1266Ø ELSE PRINT CHR$(7); : GOTO 1211
      Ø
[G92] 1221Ø   GOTO 1224Ø
[XXN] 1222Ø IF CH$ = CHR$(8) THEN IF X > 1 THEN ZEILE$
      = LEFT$(ZEILE$, X - 2) + MID$(ZEILE$, X): GOSUB 1
      266Ø: GOSUB 1234Ø: X = X - 1 ELSE PRINT CHR$(7);
      : GOTO 1211Ø
[LZN] 1223Ø IF CH$ >= " " THEN IF LEN(ZEILE$) = 255 THE
      N PRINT CHR$(7); : GOTO 1211Ø ELSE ZEILE$ = LEFT$
      (ZEILE$, X - 1) + CH$ + MID$(ZEILE$, X): GOSUB 12
      66Ø: X = X + 1
[7YB] 1224Ø LOCATE OBEN + INT((X - 1) / 78), LINKS - 1
      + X - INT((X - 1) / 78) * 78: PRINT "■";
[LK5] 1225Ø IF CH$ <> CHR$(13) THEN GOTO 1211Ø
[648] 1226Ø IF LAUF THEN ZEILE$ = MID$(CHECK$, 2, 3) +
      " " + ZEILE$
[XL6] 1227Ø GOSUB 1234Ø: GOSUB 1266Ø'Zeile anzeigen
[2W1] 1228Ø RETURN
[NUØ] 1229Ø '
[CP3] 123ØØ '****************************
[2H5] 1231Ø '* Zeichen wiederherstellen *
[EP3] 1232Ø '****************************
```

Listing. »SCheck.bas« für GW-Basic

```
[LUØ] 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
[RUØ] 12370 '
[ØA2] 12380 '***************
[3X2] 12390 '* Quit (Ende) *
[W92] 12400 '***************
[NUØ] 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
      HER$
[BW1] 12490 RETURN
[651] 12500 END
[TUØ] 12510 '
[6R2] 12520 '******************
[U24] 12530 '* Fenster löschen *
[8R2] 12540 '******************
[XUØ] 12550 '
[Ø44] 12560 FOR Q = OBEN TO UNTEN
[Z53] 12570 LOCATE Q, LINKS
[8B6] 12580   PRINT STRING$(RECHTS - LINKS + 1, 32)
[ZN1] 12590 NEXT Q
[AW1] 12600 RETURN
[XUØ] 12610 '
[265] 12620 '****************************************
      *
[D98] 12630 '* Inhalt von »zeile$« im Fenster anzeigen
      *
[465] 12640 '****************************************
      *
[1VØ] 12650 '
[SR2] 12660 F.X = Ø: F.Y = Ø
[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 = Ø: F.Y =
      F.Y + 1
[6S2] 12720 NEXT ZAEHLER
[RX6] 12730 LOCATE OBEN + F.Y, LINKS + F.X: PRINT " ";
[KW1] 12740 RETURN
[5VØ] 12750 '
[AH4] 12760 '************************************
[ET6] 12770 '* Zeile auf Check-Summe überprüfen *
[CH4] 12780 '************************************
```

```
[9VØ] 12790 '
[F42] 12800 P = Ø: 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"
[Ø22] 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)
[ØP1] 12920 NEXT I
[KF2] 12930 PS$ = "[    ]"
[AM2] 12940 FOR I = 2 TO 4
[K25] 12945   IF P > 32767 THEN P = P - 32768
[5T7] 12950   MID$(PS$, I, 1) = MID$(ZEICHEN$, (P AND 3
      1) + 1, 1)
[HZ2] 12960   P = FIX(P / 32)
[5P1] 12970 NEXT I
[YW1] 12980 RETURN
[HVØ] 12990 '
[ZX3] 13000 '********************************
[KØ6] 13010 '* In Großbuchstaben umwandeln *
[934] 13020 '********************************
[AUØ] 13030 '
[9ZN] 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
      $, Q, 1) = "Ü"
[DN1] 13080 NEXT Q
[YV1] 13090 RETURN
[BUØ] 13100 '
[YZ2] 13110 '**********************
[9N4] 13120 '* Meldezeile löschen *
[ØØ3] 13130 '**********************
[FUØ] 13140 '
[ZT5] 13150 LOCATE 21, 2: PRINT STRING$(78, 32)
[ZV1] 13160 RETURN
```

Listing. »SCheck.bas« für GW-Basic

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 (0 81 21) 7 69-0 zur Verfügung.

Schicken Sie Ihre Unterlagen an folgende Adresse:

Redaktion DOS International, »DOS Software«,
Gruberstraße 46c, 8011 Poing

C-Listing

Skeybgr in neuer Version 4.0

Schon mancher Anwender saß in seinen Anfängertagen vor dem Rechner und wunderte sich darüber, daß dieser statt dem »y« ein »z« auf den Bildschirm schrieb. Auch so manche andere Taste brachte nicht das gewünschte Zeichen. Hilfsbereite Freunde erzählten beflissen von amerikanischer und deutscher Tastaturbelegung und einem Tastaturtreiber, der hier Abhilfe schafft.

```
AX = 7900hex: Ermittelt den Status des Tastaturtreibers
Eingabe: CX = 0: Der Skeybgr in Version 3.10 verändert
                 dieses Register nicht
Ausgabe: AX = 7777hex: installierter Treiber, sonst xxxx
         BX = Statusflags und gesetzte Parameter:
            Bit 0  : älteres BIOS
            Bit 1  : erweiterter Umschalt-Modus aktiv
            Bit 2  : erweiterter Tastaturpuffer
            Bit 3  : Dezimalpunkt in Ziffernblock
            Bit 4  : Tastatur auf größte Geschwindigkeit
            Bit 5  : kein Speichertest bei Reset
            Bit 6  : Fremdtastatur (nicht IBM-kompatibel)
            Bit 7  : erweiterte Tastaturfunktionen
                     (Interrupt 16hex)
            Bit 8  : deutsche Tastenbelegung aktiv
            Bit 9  : erweitertes Setzen von Akzenten
            Bit 10 : Rollen aktiv
            Bit 11 : Num aktiv
            Bit 12 : Umschalten aktiv
            Bit 13 : Taste [M] gedrückt
            Bit 14 : AT- oder MF2-Tastatur installiert
            Bit 15 : MF2-Tastatur installiert bei Version
                     bis 3.41, Bits 10-15 = 0.
         CX = Versionsnummer in BCD-Format,
            bei Version 3.10, wie Eingabe (CX=0),
         CF = 0.

AX = 7901hex: Setzt die Parameter für den Tastaturtreiber
Eingabe: BX = Flags für gesetzte Parameter:
            Bit 0  : Anpassung an ältere BIOS-Versionen
            Bit 1  : erweiterten Umschalt-Modus setzen
            Bit 2  : Tastaturpuffer auf 127 Zeichen erweitern
            Bit 3  : Punkt statt Komma im Ziffernblock
            Bit 4  : Tastatur auf größte Geschwindigkeit
            Bit 5  : kein Speichertest bei Reset
            Bit 6  : Fremdtastatur (nicht IBM-kompatibel)
            Bit 7  : Funktionen des Interrupt 16hex ersetzen
            Bit 8  : deutsche Tastenbelegung aktiviert
            Bit 9  : erweiterte Akzentsetzung
            Bit 10 : Rollen aktivieren
            Bit 11 : Num aktivieren
            Bit 12 : erweitertes oder normales Umschalten

Ausgabe: CF = 0
Ein jeweils gesetztes Bit schaltet den Tastaturtreiber auf die
gewünschte Funktion. Ist das Bit Null, setzt das diese Funktion
zurück. Eine einmal gesetzte größte Geschwindigkeit läßt sich
allerdings nicht wieder löschen.

AX = 7903hex: belegt Taste mit neuem Zeichen
Eingabe: BL = Tastenebene von neu zu belegender Taste:
            0 : Normal (amerikanische Belegung)
            1 : Umschalt (amerikanische Belegung)
            2 : Normal (deutsche Belegung)
            3 : Umschalt (deutsche Belegung)
            4 : AltGr
            5 : Strg
            6 : Alt-Strg
            7 : Umschalt-Alt-Strg
         CH = Scancode der neu zu belegenden Taste,
            erlaubte Werte sind 1 bis 53 und 86
         CL = neuer ASCII-Code der angegebenen Taste
Ausgabe: AL = Fehlercode:
            0 : angegebene Taste neu belegt
            1 : unbekannte Tastenebene
            2 : ungültiger Scancode
         DL = bisheriger ASCII-Code der geänderten Taste
         CF = 0 bei neu belegter  Taste, sonst 1

AX = 7904hex: liefert den Zeiger auf das aktuelle Passwort
Ausgabe: BX = Offset von Passwort
         DI = Segment von Passwort
         CF = 1,bei gesperrter Tastatur gesperrt, sonst 0

AX = 79xxhex: unbekannte Funktion
Ausgabe: CF = 1, wenn xx ungleich 0, 1, 3 oder 4 ist
                 sonst siehe oben
```

Bild 1. Die zusätzlichen Aufrufe des Interrupts 16hex in Skeybgr

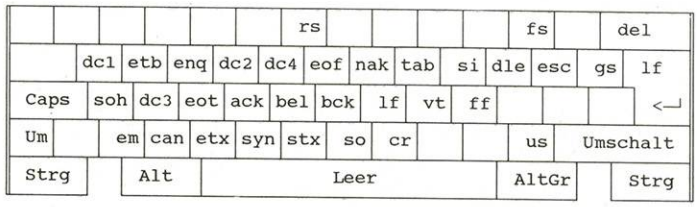


Bild 2. Die Tastenbelegung bei gedrückter Strg-Taste

Mit »Skeybgr« erhalten Sie einen vollwertigen Ersatz für den Tastaturtreiber »keyb.com«, der im Lieferumfang des Betriebssytems enthalten ist. Neben den grundsätzlichen Tastenbelegungen, die Keyb bietet, ist Skeybgr um einige Leistungsmerkmale umfangreicher, die Sie über Parameter steuern können.

Diesen Treiber können Sie, genauso wie seine Vorgängerversion »Skeybgr Version 3.10«, von der Kommandozeile oder aus Stapeldateien heraus aufrufen. Die Parameter müssen Sie immer in Kleinbuchstaben angeben. Zur Trennung der Parameter genügt ein Leerzeichen oder der Schrägstrich »/«. Weiterhin hat sich die Parameterliste im Vergleich zur Vorgängerversion des Treibers geändert (Tabelle 1). Einen Parameter können Sie nur dann löschen, wenn Sie ihn bei einem erneuten Aufruf des Tastaturtreibers mit einem unmittelbar folgenden Minuszeichen angeben. Den Parameter einfach wegzulassen genügt nicht mehr. Bisher konnten Sie alle gesetzten Parameter löschen, indem Sie den Treiber ohne Parameter aufriefen. Jetzt müssen Sie entweder alle Parameter mit unmittelbar folgenden Minuszeichen oder »/l« angeben. Beim nochmaligen Laden des Treibers bringen Sie nur die Parameter auf den angegebenen Stand, ohne daß er sich dabei resident installiert. Um ihn erneut zu laden, müssen Sie den Rechner erst zurücksetzen. Folgende Meldungen gibt der Tastaturtreiber nach seinem Aufruf aus:

`Skeybgr ist installiert !`

Der Tastaturtreiber wurde geladen und speicherresident installiert. Damit übernimmt er die Abfrage der Tastatur und stellt die Tastendaten in gleicher Weise bereit, wie es sonst Keyb macht.

`Interner Fehler !`

Der binäre Aufbau des Tastaturtreibers wurde verändert. Diese Meldung tritt auf, wenn ein fremdes Programm (Virus?!) auf die Programmdatei »skeybgr.exe« des Treibers zugegriffen und sie verändert hat.

`keine Parameter geändert !`

Falls der Tastaturtreiber bereits resident ist und Sie ihn noch einmal ohne beziehungsweise mit den Parametern »/h-« oder »/l-« aufrufen, erscheint diese Meldung, weil sich an seiner Konfiguration nichts geändert hat.

`neue Parameter sind übergeben !`

Hiermit meldet der Tastaturtreiber die erfolgreiche Übergabe der neuen Parameter an den bereits residenten Treiber. Damit ist dessen Konfiguration auf dem angegebenen Stand.

`Fehler in Parameter, Hilfe mit /h !`

Weiter durch Tastendruck. Enthält die Parameterliste des Tastaturtreibers eine unbekannte Größe, wird diese Meldung ausgegeben. Dabei ist es empfehlenswert, den Treiber mit dem Parameter »/h« aufzurufen, um den Fehler zu fin-

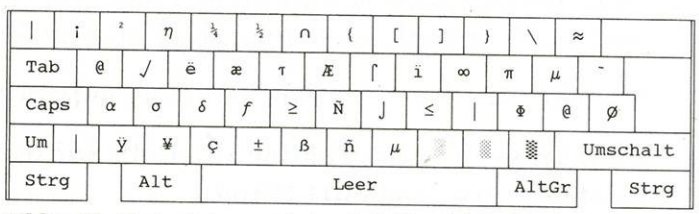


Bild 3. Die Tastenbelegung bei gedrückter AltGr-Taste



Bild 4. Die Tastenbelegung, wenn Alt- und Strg-Taste gedrückt sind



Bild 5. Die Scancodes der Tasten ohne zusätzliche Sondertasten

den. Eventuell richtige Parameter bleiben unberücksichtigt. Vorher ist natürlich eine Taste zu drücken.

`frühere Version ist installiert !`

Der Tastaturtreiber hat eine ältere Version von sich selbst gefunden, ohne sie zu ersetzen. Es empfiehlt sich, den Tastaturtreiber ohne den Parameter »/h« aufzurufen, um sie abzulösen.

`Skeybgr ist nicht installiert !`

Weder der Tastaturtreiber noch ein Skeybgr wurden installiert. Erst durch den Aufruf des Treibers ohne den Parameter »/h« erfolgt seine Installation, so daß er die Abfrage der Tastatur sowie das Bereitstellen von Tastendaten übernehmen kann.

Alle folgenden Parameter können Sie beim Aufruf des Tastaturtreibers durch Angabe des aufgeführten Kleinbuchstabens aktivieren. Die einzelnen Eingaben müssen Sie durch einen Schrägstrich »/« oder durch ein Leerzeichen trennen. Großbuchstaben sind für den Treiber keine gültigen Parameter.

Falls sich in der von Ihnen angegebenen Liste ein ungültiger Parameter befindet, gibt das Programm eine entsprechende Fehlermeldung aus. Eventuell gültige Parameter berücksichtigt der Tastaturtreiber nicht. Durch die Angabe eines Minuszeichens unmittelbar hinter einem Parameter löschen Sie ihn wieder. So läßt sich beispielsweise mit dem Parameter »/a« die amerikanische Tastenbelegung aktivieren und mit »/a-« wieder deaktivieren.

`/h`

zeigt eine Kurzbeschreibung der Parameter an. Falls der Tastaturtreiber schon geladen ist, verarbeitet er vor der Listenausgabe weitere gültige Parameter, die Sie in Verbindung mit diesem Parameter angegeben haben. Andernfalls ignoriert er die zusätzlich angegebenen. Bei geladenem Skeybgr erscheint in der Liste unmittelbar hinter dem Parameterbuchstaben ein Plus- beziehungsweise Minuszeichen. Es gibt an, ob der Parameter zum Zeitpunkt der Ausgabe der Kurzbeschreibung aktiv ist oder nicht. Die Angabe von »/h-« hat keine Auswirkungen auf den Treiber.

`/a`

aktiviert die amerikanische Tastaturbelegung. Mit [Alt-Strg-F2] oder durch Aufruf des Tastaturtreibers mit »/a-« schalten Sie wieder auf die deutsche Tastenbelegung um.

`/b`

paßt den Treiber an ältere BIOS-Versionen an. Diesen Parameter müssen Sie in Zukunft setzen, wenn beim Drücken der abgesetzten Cursortasten α-Zeichen erscheinen. Er ist jedoch nur in Verbindung mit dem Parameter »/i« wirksam, weil Skeybgr sonst sowieso alle BIOS-Funktionen für die Tastatur ersetzt. Eine Anpassung an ältere BIOS-Versionen ist deshalb nicht erforderlich.

gewünschte Funktion	Version 3.10	Version 4.0
amerikanische Tastaturbelegung	/a	/a
Anpassung an älteres BIOS	/b	/b
normales Umschalt fest	kein /s	/c
erweiterte Funktionen setzen	/e	/i-
Fremdtastatur (nicht kompatibel)	/f	/f
Tastatur auf größte Geschwindigkeit	/g	/g
Anzeige der Parameterliste	/h	/h
erweiterte Funktionen ignorieren	kein /e	/i
kein Speichertest bei Reset	/r	/k
alle gesetzten Parameter löschen	siehe oben	/l
Num einschalten	/n	/n
Punkt statt Komma im Ziffernblock	/p	/p
Rollen einschalten	unbekannt	/r
Umschalt fest einschalten	unbekannt	/s
erweitertes festes Umschalten	/s	/c-
Tastaturpuffer erweitern	/t	/t
erweiterte Akzentsetzung	/z	/z

Mit »kein /s« beziehungsweise »kein /e« ist bei Skeybgr 3.10 gemeint, daß die gewünschte Funktion bei einem Aufruf des Treibers ohne den Parameter »/s« beziehungsweise »/e« aktiviert wird. Die Parameter »/i-« beziehungsweise »/c-« bei der aktuellen Version sind nur dann erforderlich, wenn der Treiber vorher mit den Parametern »/i« beziehungsweise »/c« aufgerufen wurde.

Tabelle 1. Vergleich der Parameterlisten

`/c`

ersetzt das »erweiterte feste Umschalten« durch das »normale feste Umschalten«. Während Sie beim normalen festen Umschalten nur die Größe der Buchstaben umkehren, bewirken Sie mit dem erweiterten Modus ein Umschalten sämtlicher Zeichentasten im Haupttastenfeld. Der Ziffernblock ist davon nicht betroffen. Den normalen Modus schalten Sie durch Drücken von [Umschalt fest] wieder aus, das erweiterte Umschalten mit Hilfe einer der beiden Umschalt-Tasten.

`/f`

unterstützt Fremdtastaturen, die nicht hundertprozentig IBM-kompatibel sind. Im Regelfall sollten Sie den Tastaturtreiber ohne diesen Parameter laden. Falls es aber nach dem Betätigen der Feststell-Tasten beziehungsweise bei der Benutzung der Parameter »/n«, »/r« oder »/s« zu Problemen mit der Tastatur kommt, sollten Sie diesen Parameter bei jedem Laden des Treibers angeben. Nach bisherigen Erfahrungen können diese Probleme nur in Verbindung mit dem Parameter »/i« auftreten.

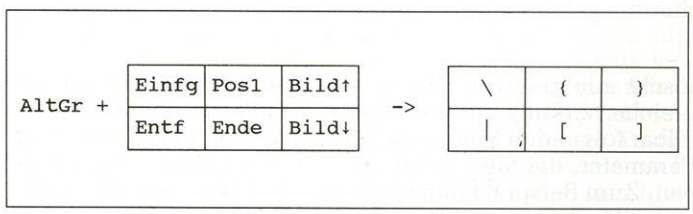


Bild 6.Die Tastenbelegung in Kombination mit <Umschalt-Strg-Alt>

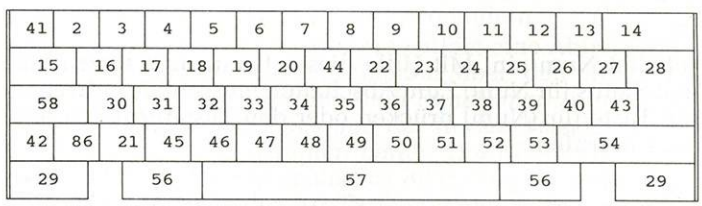


Bild 7. Die zusätzliche Belegung einiger Tasten in Kombination mit <AltGr>

Bit	Bedeutung
0	rechte Umschalt-Taste gedrückt
1	linke Umschalt-Taste gedrückt
2	eine Strg-Taste gedrückt
3	eine Alt-Taste gedrückt
4	Rollen aktiv
5	Num aktiv
6	erweitertes Umschalt fest
7	Einfügemodus aktiv

Tabelle 2. Belegung der Adresse 0040:0017hex

Bit	Bedeutung
0	linke Strg-Taste gedrückt
1	Alt-Taste gedrückt
2	Taste Systemabfrage gedrückt
3	Pausenmodus aktiv
4	Taste [Rollen] gedrückt
5	Taste [Num] gedrückt
6	Taste [Umschalten fest] gedrückt
7	Taste [Einf] gedrückt

Tabelle 3. Belegung der Adresse 0040:0018hex

`/g`

schaltet die Tastatur auf größte Geschwindigkeit. Bei einer MF2-Tastatur stellen Sie die Wiederholrate für die Zeichen wiedergabe bei gedrückter Taste auf den größten Wert ein. Diesen Parameter können Sie jedoch mit »/g-« nicht wieder zurücksetzen.

`/i`

ignoriert die erweiterten Tastaturfunktionen. Mit dem Laden des Treibers ersetzen Sie die BIOS-Funktionen für den Zugriff auf Tastaturpuffer und -flags durch neue, zum PS/2-BIOS kompatible Funktionen. Dadurch gibt der Interrupt 16hex mit Sicherheit alle unten aufgeführten, erweiterten Scancodes zurück. Der Parameter »/b« wird unterdrückt, weil alte BIOS-Funktionen nicht mehr aktiv sind. Mit dem Parameter »/i« unterdrücken Sie die BIOS-Erweiterung und erlauben die Angabe von »/b«. Im allgemeinen ist es jedoch nicht erforderlich, diesen Parameter anzugeben.

`/k`

schaltet den Speichertest bei Reset aus. Der Speichertest, den das BIOS bei jedem Reset durchführt, wird nun unterdrückt, sofern es im Rechner diese Unterscheidung beim Reset zuläßt. Bei einigen Computersystemen führt das BIOS immer einen beziehungsweise keinen Speichertest durch, egal, ob man es dazu auffordert oder nicht.

`/l`

löscht alle gesetzten Parameter. Dieser Parameter hat die gleiche Wirkung wie die Angabe aller Parameter mit unmittelbar folgenden Minuszeichen. Davon nicht betroffen sind Parameter, die Sie zusätzlich in Verbindung mit »/l« angeben. Zum Beispiel können Sie mit der Liste »/n/l« alle Parameter löschen, aber die Num-Funktion aktivieren. Die Angabe von »/l-« hat keine Auswirkungen auf die Parameterflags im Tastaturtreiber.

`/n`

schaltet Num ein. Mit Hilfe dieses Parameters setzen Sie den Status für Num. Zum Abschalten müssen Sie entweder die Taste für [Num] drücken oder den Tastaturtreiber mit »/n-« aufrufen.

`/p`

liefert bei Betätigen der Taste [,] im Ziffernblock einen Dezimalpunkt statt eines Dezimalkommas.

Bit	Belegung
0	letzter Code war E1hex
1	letzter Code war E0hex
2	rechte Strg-Taste gedrückt
3	AltGr-Taste gedrückt
4	Tastatur mit 101/102 Tasten installiert
5	Num erzwingen, wenn Bit 4 gesetzt
6	letztes Zeichen war erstes ID-Zeichen
7	ID wird gelesen

Tabelle 4. Belegung der Adresse 0040:0096hex

Bit	Belegung
0	Led für Scrollen ein
1	Led für Num ein
2	Led für Umschalt fest ein
3	reserviert (muß 0 sein)
4	Empfangsbestätigung erhalten
5	letztes Kommando wiederholen
6	Modusanzeige, Aktualisierung läuft
7	Fehlerflag Tastaturübertragung

Tabelle 5. Belegung der Adresse 0040:0097hex

`/r`

schaltet Rollen ein. Um es wieder abzuschalten, müssen Sie entweder die Taste [Rollen] drücken oder den Tastaturtreiber mit dem Parameter »/r-« aufrufen.

`/s`

setzt den Status für erweitertes oder normales Umschalten. Diesen Status schalten Sie durch Drücken der entsprechenden Taste (siehe oben) oder Aufruf des Tastaturtreibers mit dem Parameter »/s-« wieder aus.

`/t`

erweitert den Tastaturpuffer von 15 auf 127 Zeichen. Dieser Parameter arbeitet nur in Verbindung mit den erweiterten Tastaturfunktionen, die Sie deshalb nicht mit Hilfe des Parameters »/i« unterdrücken dürfen. Skeybgr plaziert den Tastaturpuffer an eine neue Adresse im Hauptspeicher, so daß er ihn auf eine Länge von 256 Byte erweitern kann. Dieser Parameter ist nicht unproblematisch, weil es immer noch alte Programme gibt, die direkt auf den Tastaturpuffer zugreifen. Bisher ist jedoch kein Fall bekannt, wo es in Verbindung mit diesem Parameter Schwierigkeiten gab.

`/v`

bestimmt das Paßwort für die Verriegelung. Hinter dem Parameter »/v« folgt nach einem Doppelpunkt ein Paßwort mit

Adresse	Belegung
0040:0019hex	ASCII-Code vom letzten Zeichen, das über die alternative Tastatureingabe eingegeben wurde [Alt-Ziffernblock]
0040:001Ahex	zwei Byte Integer zeigt auf das nächste aus dem Tastaturpuffer zu lesende Zeichen; Basissegment ist 40hex
40:001Chex	zwei Byte Integer zeigt auf die nächste freie Position im Tastaturpuffer; Basissegment ist 40hex
0040:0072hex	zwei Byte Reset-Flag. Falls das Flag auf 1234hex gesetzt wurde, dann wird beim Reset der Speichertest umgangen.
0040:0080hex	zwei Byte Integer zeigt auf den Anfang des Tastaturpuffers; Basissegment ist 40hex
0040:0082hex	zwei Byte Integer zeigt auf das erste Zeichen hinter dem Tastaturpuffer; Basissegment ist 40hex

Tabelle 6. Weitere von Skeybgr belegte Adressen

Code	Taste	Code	Taste
1	[Alt-Esc]	120-131	[Alt-1] bis [Alt-']
3	[Strg-2] (ASCII NULL)	132	[Strg-Bild oben]
15	[Rückschritt-Tab]	133,134	[F11], [F12]
16- 28	[Alt-Q] bis [Alt-Eingabe]	135,136	[Umschalt-F11], [Umschalt-F12]
30- 41	[Alt-A] bis [Alt-#]	137,138	[Strg-F11], [Strg-F12]
43- 53	[Alt-<] bis [Alt-]	139,140	[Alt-F11], [Alt-F12]
55	[Alt-*] im Ziffernblock	141	[Strg-Cursor hoch]
59- 68	[F1] bis [F10]	142	[Strg-] im Ziffernblock
71	[Pos1]	143	[Strg-5] im Ziffernblock
72	[Cursor hoch]	144	[Strg-+] im Ziffernblock
73	[Bild hoch]	145	[Strg-Cursor unten]
74	[Alt-] im Ziffernblock	146	[Strg-Einfg]
75	[Cursor links]	147	[Strg-Entf]
76	[Center-Cursor]	148	[Strg-Tab]
77	[Cursor rechts]	149	[Strg-/] im Ziffernblock
78	[Alt-+] im Ziffernblock	150	[Strg-*] im Ziffernblock
79	[Ende]	151	[Alt-Pos1]
80	[Cursor unten]	152	[Alt-Cursor hoch]
81	[Bild hoch]	153	[Alt-Bild hoch]
82	[Einfg]	155	[Alt-Cursor links]
83	[Entf]	157	[Alt-Cursor rechts]
84- 93	[Umschalt-F1] bis [Umschalt-F10]	159	[Alt-Ende]
94-103	[Strg-F1] bis [Strg-F10]	160	[Alt-Cursor unten]
104-113	[Alt-F1] bis [Alt-F10]	161	[Alt-Bild unten]
114	[Strg-Druck]	162	[Alt-Einfg]
115	[Strg-Cursor links]	163	[Alt-Entf]
116	[Strg-Cursor rechts]	164	[Alt-/] im Ziffernblock
117	[Strg-Ende]	165	[Alt-Tab]
118	[Strg-Bild unten]	166	[Alt-Eingabe]
119	[Strg-Pos1]		

Tabelle 7. Die erweiterten Scancodes

höchstens 17 Zeichen. Das Leerzeichen und den Schrägstrich zur Parametertrennung dürfen Sie dabei nicht verwenden. Mit der Tastenkombination [Umschalt-Pause] sperren Sie dann im folgenden die Tastatur. Erst nach dem Eintippen des Paßwortes und dem Betätigen der Eingabe-Taste können Sie wieder Eingaben vornehmen.

/z

erlaubt es, Buchstaben mit einem Akzent zu versehen. Dazu tippen Sie zuerst das gewünschte Akzentzeichen und anschließend den Buchstaben ein, auf dem der Akzent erscheinen soll. Dieser Parameter ist besonders für fremdsprachliche Einsätze des Tastaturtreibers gedacht, wenn im System als Zeichensatz die Codepage 850 installiert ist. Bei anderen Codepages können bestimmte Kombinationen aus Akzent- und Buchstabentaste andere Zeichen zur Folge haben, als aus der Eingabe zu schließen wäre.

Neben den üblichen Tasten zur Zeicheneingabe können Sie mit bestimmten Tastenkombinationen Kommandos an den Treiber senden. So hält [Pause] das laufende Programm an. Durch nochmaliges Betätigen lassen Sie das Programm anschließend weiterlaufen. Mit [Umschalt-Pause] sperren Sie die Tastatur und Skeybgr wartet auf das Paßwort, das mit dem Parameter »/v« vorgegeben wurde. Falls Sie kein Paßwort angegeben haben, bewirkt diese Tastenkombination nichts. Durch Betätigen von [Alt-Pause] löschen Sie den Inhalt des Tastaturpuffers, eventuelle Eingaben mit [Alt-Zifferntaste], bevor Sie die Alt-Taste lösen. Diese Funktion ist wichtig, wenn Sie im Eifer des Gefechts bereits Tastatureingaben machen, bevor das Programm diese auswerten kann. Falls Sie sich dabei vertippen, löschen Sie mit [Alt-Pause] alle noch nicht verarbeiteten Eingaben. Wie Sie es bereits von Keyb her kennen, aktiviert [Alt-Strg-F1] die amerikanische Tastaturbelegung, [Alt-Strg-F2] die deutsche Tastaturbelegung.

Fehlercode	Bedeutung
0	Tastaturtreiber installiert
1	interner Fehler
2	neue Parameter an Treiber übergeben
3	unbekannter Parameter
4	Parameterliste ausgegeben

Tabelle 8: Zurückgegebene Errorlevel

Tastaturtreiber hängen sich in den Interrupt 16hex ein. Informationen zu den üblichen Funktionen dieses Interrupts finden Sie in der gängigen Literatur, daher beschreibt Bild 1 nur die zusätzlichen Funktionen des Skeybgr. Von ihm belegte Adressen und deren Bedeutung erklären die Tabellen 2 bis 6. Welchen Scancode eine Taste hat, zeigt Tabelle 7, die Fehlercodes Tabelle 8. Die Belegung der Tastatur bei Betätigen einer Taste zusammen mit einer Sondertaste ([Alt], [Strg] etc.) demonstrieren die Bilder 2 bis 6. Auf Bild 7 sehen Sie die normalen Scancodes.

Der Tastaturtreiber besteht aus zwei voneinander abhängigen Interruptroutinen, die über BIOS-Variablen kommunizieren und sich aufeinander abstimmen. Bei der ersten Routine handelt es sich um den Service-Interrupt 9, den der Rechner immer dann auslöst, wenn Sie eine Taste betätigen. Danach liest die Serviceroutine aus Port 60hex die Nummer der Taste oder ein Steuerbyte, das die betätigte Taste näher beschreibt. Anschließend löst sie den Interrupt 15hex aus. Das Register »AH« hat dabei den Wert 4Fhex, und »AL« enthält die Nummer der Taste beziehungsweise das Steuerbyte. Diese Funktion erlaubt es anderen Programmen, die Nummer der ausgelösten Taste zu verändern oder die Taste zu unterdrücken, indem es ein gelöschtes Carry-Flag an die Serviceroutine zurückgibt.

Handelt es sich bei der betätigten Taste um [Alt], [Strg] oder [Umschalt], dann protokolliert der Treiber diese Taste in den BIOS-Variablen 40:17hex, 40:18hex und 40:96hex (Tabellen 2 bis 4). Genauso verhält es sich mit den drei Feststell-Tasten, nur daß er deren Status lediglich bei einem erneuten Drücken der entsprechenden Taste verändert.

Einige Tastenkombinationen rufen von sich aus weitere BIOS-Funktionen auf, beispielsweise [Druck] den Interrupt 5. Den Bildschirm können Sie jedoch nur ausdrucken, wenn das »Print-Screen-Status-Byte« an Adresse 50:00hex den Wert 0 hat. Die Tastenkombination [Strg-Untbr] (beziehungsweise [Strg-Rollen] bei älteren Tastaturen) löst einen Interrupt 1Bhex aus, nachdem die Serviceroutine den Tastaturpuffer gelöscht und das Break-Flag 40:71hex auf 80hex gesetzt hat. Bei [Alt-Strg-Entf] bewirkt die Routine einen Reset.

Eine besondere Bedeutung hat die Taste [S-Abf] der AT- und MF2-Tastatur. Diese Taste löst einen Interrupt 15hex aus (»AH« gleich 85; »AL« gleich 0 bei gedrückter oder 1 bei gelöster Taste [S-Abf]).

Das ASCII-Zeichen und den Scancode, die der Treiber aus einer gedrückten Taste sowie aus den Informationen in den BIOS-Variablen herleitet, schreibt er in den Tastaturpuffer, der normalerweise 32 Byte lang ist und an der Adresse 40:1Ehex beginnt. Die aktuelle Schreibposition im Tastaturpuffer befindet sich an der Adresse 40:1Chex, die nach dem Schreibvorgang um 2 erhöht wird. Wenn sie sich hinter dem Ende des Tastaturpuffers befindet, wird sie auf den Pufferanfang gesetzt. Die Position des Tastaturpuffers läßt sich bei Rechnern ab der AT-Klasse über die Variablen 40:80hex und 40:82hex (Tabelle 6) verändern.

Das Gegenstück zur Serviceroutine bildet eine Abfrageroutine, die jedes beliebige Programm über den Interrupt 16hex erreicht. Diese Routine stellt Funktionen zum Abfragen der Tastaturstatusflags und der in den Tastaturpuffer geschriebenen Informationen bereit. Die Routinen 10hex und 12hex wurden erst später ins BIOS aufgenommen, um Tastaturen mit 101 oder 102 Tasten zu unterstützen.

(Jan-Jürgen Eden/rm)

```
[WH3]    1: /*Programm: skeybgr.c
[KG4]    2:   Funktion: Tastaturtreiber
[BU2]    3:   Sprache : Quick C
[G65]    4:    Speichermodell Small verwenden
[YH4]    5:    alle Debugflags auschalten
[P34]    6:    Stack-Check ausschalten
[UL4]    7:    Typ für char ist unsigned
[5P3]    8:   Autor: Jan Jürgen Eden
[T15]    9:   (c)1991 DMV Widuch GmbH & Co. KG */
[ØØØ]   1Ø:
[2Ø3]   11: #include <conio.h>
[XM2]   12: #if __TURBOC__
[1L2]   13: #include <dir.h>
[NZØ]   14: #endif
[9L2]   15: #include <dos.h>
[L13]   16: #include <stdio.h>
[U83]   17: #include <stdlib.h>
[ØØØ]   18:
[KR4]   19: typedef unsigned char byte;
[YX4]   2Ø: typedef unsigned short word;
[2X4]   21: typedef unsigned long dword;
[2K4]   22: #define _INTERVALL(a,x,y) \
[PF2]   23: ((a)>=(x)&&(a)<=(y))
[VX5]   24: #define _NEXT(a,x,y) ((a)>=(y)?(x):(a)+2)
[G45]   25: #define _CPEEK(a) (*(byte far*)(a))
[T75]   26: #define _SPEEK(a) (*(word far *)(a))
[LB5]   27: #define _LPEEK(a) (*(dword far *)(a))
[HR3]   28: #define _FARPTR(s,o) \
[1Ø5]   29: ((char far*)(((dword)(s)<<16)|(o)))
[T55]   3Ø: #define BIOSPAGE (byte far*)ØxØ4ØØ
[6T4]   31: #define FLAG_Ø _CPEEK(ØxØØ4ØØØ17)
[AT4]   32: #define FLAG_1 _CPEEK(ØxØØ4ØØØ18)
[BU4]   33: #define FLAG_2 _CPEEK(ØxØØ4ØØØ96)
[FU4]   34: #define FLAG_3 _CPEEK(ØxØØ4ØØØ97)
[L36]   35: #define BIOSMEM(a)   _CPEEK(BIOSPAGE+a)
[726]   36: #define BUFFER_HEAD  _SPEEK(ØxØØ4ØØØ1a)
[746]   37: #define BUFFER_TAIL  _SPEEK(ØxØØ4ØØØ1c)
[WR5]   38: #define KB_START     _SPEEK(ØxØØ4ØØØ8Ø)
[LF5]   39: #define KB_END       _SPEEK(ØxØØ4ØØØ82)
[WP5]   4Ø: #define ALT_ASCII    _CPEEK(ØxØØ4ØØØ19)
[ST5]   41: #define BREAK_FLAG   _CPEEK(ØxØØ4ØØØ71)
[AS5]   42: #define RESETFLAG    _SPEEK(ØxØØ4ØØØ72)
[BØ6]   43: #define PRINTSCREEN _CPEEK(ØxØØ5ØØØØØ)
[XM2]   44: #if __TURBOC__
[DV2]   45: #pragma warn -sus
[BU2]   46: #pragma warn -par
[X33]   47: #define INT_REGS \
[G46]   48: bp,di,si,ds,es,dx,cx,bx,ax,ip,cs,flags
[PS5]   49: #define _dos_getvect(v) getvect(v)
[6B6]   5Ø: #define _dos_setvect(v,f) setvect(v,f)
[AB4]   51: #define _disable() disable()
[234]   52: #define _enable() enable()
[B45]   53: #define _dos_keep(e,l) keep(e,l)
[346]   54: #define GO_OLD_IRQ16() old_ask_key()
[8H1]   55: word end;
[2TØ]   56: #else
[EC3]   57: #if !defined(M_I86SM)
[TW5]   58: #error requires small memory model
[NZØ]   59: #endif
[NK4]   6Ø: #pragma check_pointer (off)
[X33]   61: #define INT_REGS \
[7L6]   62: es,ds,di,si,bp,sp,bx,dx,cx,ax,ip,cs,flags
[3Y3]   63: #define GO_OLD_IRQ16() \
[UG4]   64: _chain_intr(old_ask_key)
[FS2]   65: extern word end;
[NZØ]   66: #endif
[6V5]   67: void interrupt far int_keyboard();
[ZP5]   68: void interrupt far int_ask_key();
[Z36]   69: void (interrupt far *old_ask_key)();
[H56]   7Ø: void (far *reset)()=(void far*)Øxffff0000;
[RW3]   71: #define VERSION      Øx41Ø
[KX6]   72: char *programname = "\r\nSkeybgr Version"
[RUØ]   73:                     " 4.1Ø";
[VR5]   74: char *copyright = "\r\n(c)1991 by DMV"
[4T4]   75:          " Widuch GmbH & Co. KG\r\n\n";
[X83]   76: word paraflag[26] =
[VT4]   77: { ØxØ1ØØ,ØxØØØ1,ØxØØØ2,Øx0000,Øx0000,
[HN4]   78:   ØxØØ4Ø,ØxØØ1Ø,Øx8ØØØ,ØxØØ8Ø,Øx0000,
[ØN4]   79:   ØxØØ2Ø,Øx4ØØØ,Øx0000,Øx0000,Øx0000,
[HN4]   8Ø:   Øx0008,Øx0000,Øx04ØØ,Øx1ØØØ,Øx0004,
[CM4]   81:   Øx0000,Øx4ØØØ,Øx0000,Øx0000,Øx0000,
[AUØ]   82:   Øx02ØØ
[BDØ]   83: };
[NL2]   84: byte *code[18] =
[722]   85: { NULL,NULL,
[ØVC]   86:   "\Ø≡¡²ηńℓ∩{[]}\\≈\Ø\Ø@⁄ëæτÆ⌠ï∞πμ˜\Ø\Ø"
[HNC]   87:   "ασδl≥Ñ⌡≤|÷@||·ÿØç±βΣμ▒\Ø\Ø\Ø ",
[SA6]   88:   "\Ø\33\Ø\Ø\Ø\Ø\36\Ø\Ø\Ø\34\Ø\177\Ø"
[XD6]   89:   "\21\27\5\22\24\31\25\t\17\2Ø\33\35\n\Ø"
[AH6]   9Ø:   "\1\23\4\6\7\b\n\13\14\Ø\Ø\Ø\Ø\32\3Ø"
[WG4]   91:   "\3\26\2\16\r\Ø\Ø\37\Ø\Ø\Ø ",
[74D]   92:   "\Ø≡¡²ηńℓ∩{[]}\\≈\Ø\Ø┌┐┌┐ΘLμ˜\Ø\Ø"
[EFC]   93:   "├┼┤┤╟╫╢|Ω@||φ└┘└┘ãÅ\Ø\Ø\Ø ",
[GMD]   94:   "\Ø≡¡öOfÇø±³¤Σ¿\Ø\Ø\Ø┌┐┌┐▀ϵ\Ø\Ø"
[DTC]   95:   "╟╫╢╢╞┼┤Γ▌▐\Ø·└┘└┘åÅ\Ø\Ø\Ø ",
[676]   96:   "\Øêúíôâçñ","\ØéúíóáΣå",
[837]   97:   "\Øèùìòà╞∞","\ØëüïöäøΦæ",
[ØM7]   98:   "\Ø╥Ω╫Γ╢ÇÑ","\ØÉΘ╓α┤σÅ",
[K78]   99:   "\Ø└δ▐π╖╟φ","\Ø╙Ü╪ÖÄØτÆ",
[5G6]  1ØØ:   "\Ø\8123456789Ø-=\27\tqwertyuiop[]\r\Ø"
```

```
[DW5]  1Ø1:   "asdfghjkl;\'\\\\\zxcvbnm,./\Ø*\Ø ",
[UV6]  1Ø2:   "\Ø\8!@#$%\94&*()_+\27\ØQWERTYUIOP{}\r\Ø"
[XA6]  1Ø3:   "ASDFGHJKL:\"~||ZXCVBNM<>?\Ø\Ø\Ø ",
[LW6]  1Ø4:   "\Ø\8234567890ß'\27\tqwertzuiopü+\r\Ø"
[EZ5]  1Ø5:   "asdfghjklöä\94<#yxcvbnm,.-\Ø*\Ø ",
[SZ6]  1Ø6:   "\Ø\8\"\21$%&/()=?`\27\ØQWERTZUIOPÜ*\r\Ø"
[7P6]  1Ø7:   "ASDFGHJKLÖÄ°>\'YXCVBNM;:_\Ø\Ø\Ø "
[BDØ]  1Ø8: };
[KL2]  1Ø9: byte *numblock =
[586]  11Ø: "789-456+123Ø,{\Ø}\Ø³\Ø¤\Ø[\Ø]\\|";
[ØØØ]  111:
[6N3]  112: byte *strg_scancode =
[984]  113: "wìäÄsÅtÉuævÆô";
[ØØØ]  114:
[LF3]  115: char *paraliste[15] =
[9E6]  116: { "/a : amerikanische Tastaturbelegung",
[F75]  117:   "/b : Anpassung an älteres BIOS",
[5N6]  118:   "/c : normales statt erweitertes festes"
[L62]  119:        " Umschalten",
[FD6]  12Ø:   "/f : Fremdtastatur, nicht kompatibel",
[4L6]  121:   "/g : Tastatur auf max. Geschwindigkeit",
[GP6]  122:   "/i : erweiterte Funktionen ignorieren",
[EP5]  123:   "/k : kein Speichertest bei Reset",
[UD6]  124:   "/l : alle gesetzten Parameter löschen",
[RL3]  125:   "/n : Num einschalten",
[CN6]  126:   "/p : Punkt statt Komma im Ziffernblock",
[H44]  127:   "/r : Rollen einschalten",
[YV5]  128:   "/s : ",/* /s : Shift-Lock einschalten */
[3C5]  129:   "/t : Tastaturpuffer erweitern",
[8K5]  13Ø:   "/v : Passwort für Verriegelung",
[LR5]  131:   "/z : erweiterte Akzentsetzung\n"
[BDØ]  132: };
[7N3]  133: char *pass_chr = NULL;
[H93]  134: char far *passwort;
[SD1]  135: char key;
[FP2]  136: union REGS regs;
[B72]  137: int scancode;
[184]  138: word scancode_gefunden;
[E43]  139: word hochkomma = Ø;
[KØ2]  14Ø: word ascii;
[422]  141: word stack;
[K23]  142: word paras = ØxØ182;
[7T2]  143: word new_paras;
[7P4]  144: word paras_change = Øxffff;
[2H4]  145: word word paras_newbit = Øx0000;
[H52]  146: int keycode;
[KE2]  147: int asciicode;
[ØØØ]  148:
[566]  149: void amerikanische_Tastatur(void);
[WK2]  15Ø: void beep(void);
[V25]  151: void deutsche_Tastatur(void);
[UF2]  152: int  Ende(int);
[BØ7]  153: void Interruptvektoren_anpassen(void);
[4G5]  154: void Lockstatus_pruefen(void);
[1F3]  155: int  Parameter(void);
[X66]  156: int  Parameterliste_ausgeben(void);
[W96]  157: int  Parameter_holen(char far*,word*);
[1G5]  158: void Parameter_uebergeben(void);
[X76]  159: int  Passwort_holen(char far*,word*);
[5Ø3]  16Ø: void print(char*);
[DR3]  161: void printpara(char*);
[294]  162: int  Programmtitel(void);
[S26]  163: int  Treiberparameter_holen(void);
[F74]  164: int  Treiber_laden(void);
[FL5]  165: void Tastatur_entriegeln(void);
[UC6]  166: void Tastatur_identifizieren(void);
[6A6]  167: void Tastaturpuffer_loeschen(void);
[Z86]  168: void Taste_aus_Puffer_lesen(void);
[SK6]  169: void Taste_in_Puffer_schreiben(void);
[814]  17Ø: dword TSR_Laenge(char*);
[A86]  171: void Zeichentabellen_fuer_AT(void);
[ØØØ]  172:
[5G2]  173: main(argc,argv)
[TH1]  174: int argc;
[7V1]  175: char **argv;
[C86]  176: { if(!Programmtitel()) return(Ende(1));
[CB4]  177:   Treiberparameter_holen();
[B45]  178:   if(!Parameter()) return(Ende(3));
[BV4]  179:   if((new_paras & ØxaØ00) == ØxaØ00)
[HF2]  18Ø:   return(Ende(4));
[VT5]  181:   if(!Treiber_laden()) return(Ende(2));
[U95]  182:   Interruptvektoren_anpassen();
[YR4]  183:   Tastatur_identifizieren();
[F44]  184:   Parameter_uebergeben();
[8L4]  185:   Zeichentabellen_fuer_AT();
[475]  186:   _dos_keep(Ø,TSR_Laenge(*argv));
[U7Ø]  187: }
[ØØØ]  188:
[TD3]  189: int Programmtitel()
[2V1]  19Ø: { stack = Ø;
[Ø25]  191:   for(ascii = Ø; ascii < 22; ascii++)
[GØ6]  192:   stack += programname[ascii] * ascii;
[A14]  193:   if(stack != 18Ø86) return(Ø);
[WB3]  194:   print(programname);
[4G1]  195:   stack = Ø;
[B25]  196:   for(ascii = Ø; ascii < 37; ascii++)
[UL5]  197:   stack += copyright[ascii] * ascii;
[B14]  198:   if(stack != 53542) return(Ø);
[MØ3]  199:   print(copyright);
[UL1]  2ØØ:   return(1);
```

```
[U7Ø]  2Ø1: }
[ØØØ]  2Ø2:
[2Z4]  2Ø3: int Treiberparameter_holen()
[TK5]  2Ø4: { passwort = (char far*)copyright;
[Z33]  2Ø5:   new_paras = Øx2182;
[WX2]  2Ø6:   regs.x.ax = Øx79ØØ;
[P32]  2Ø7:   regs.x.cx = Ø;
[TL3]  2Ø8:   int86(Øx16,&regs,&regs);
[725]  2Ø9:   if(regs.x.ax != Øx7777) return(Ø);
[HY3]  21Ø:   if(regs.x.cx >= VERSION)
[YA5]  211:   { new_paras = regs.x.bx & Øx1fff;
[8Y2]  212:     regs.x.ax = Øx79Ø4;
[TL3]  213:     int86(Øx16,&regs,&regs);
[DS6]  214:     passwort =_FARPTR(regs.x.di,regs.x.bx);
[TL1]  215:     return(Ø);
[U7Ø]  216:   }
[UL1]  217:   return(1);
[U7Ø]  218: }
[ØØØ]  219:
[PM2]  22Ø: int Parameter()
[9X1]  221: { int lauf;
[A16]  222:   char far *argptr = _FARPTR(_psp,Øx8Ø);
[MR3]  223:   int arglen = *argptr++;
[U93]  224:   new_paras ^= ØxØ182;
[GB2]  225:   for(lauf = Ø; ;)
[7U4]  226:   { while(argptr[lauf] == Øx2Ø
[HH3]  227:     && lauf < arglen) lauf++;
[LH3]  228:     if(argptr[lauf] == '/'
[BY3]  229:     && lauf + 1 < arglen) lauf++;
[WR3]  23Ø:     if(lauf >= arglen) break;
[LH3]  231:     if(argptr[lauf] == '/'
[5Z5]  232:     || argptr[lauf] == Øx2Ø) return(Ø);
[ØP5]  233:     if(!Parameter_holen(argptr + lauf,
[CP3]  234:     &(word)lauf)) return(Ø);
[BG2]  235:     if(lauf < arglen
[N93]  236:     && argptr[lauf] != '/'
[ZF5]  237:     && argptr[lauf] != Øx2Ø) return(Ø);
[U7Ø]  238:   }
[3S3]  239:   if(paras_newbit & Øx4ØØØ)
[U13]  24Ø:   new_paras &= Øx2ØØØ;
[UD4]  241:   new_paras &= paras_change;
[HP4]  242:   new_paras ^= paras_newbit;
[U93]  243:   new_paras ^= ØxØ182;
[UL1]  244:   return(1);
[U7Ø]  245: }
[ØØØ]  246:
[4H6]  247: int Parameter_holen(argptr,nextpara)
[6P2]  248: char far *argptr;
[2R2]  249: word *nextpara;
[3D2]  25Ø: { word flag = Ø;
[N52]  251:   (*nextpara)++;
[6K4]  252:   if(*argptr == '?') *argptr = 'h';
[AN4]  253:   if(_INTERVALL(*argptr,'a','z'))
[8X4]  254:     flag = paraflag[*argptr - 'a'];
[VØ3]  255:   if(!flag) return(Ø);
[MS3]  256:     paras_change &= ~flag;
[7X2]  257:   if(*argptr++ == 'v')
[BZ6]  258:   return(Passwort_holen(argptr,nextpara));
[XD2]  259:   if(*argptr != '-')
[HS3]  26Ø:     paras_newbit |= flag;
[MG3]  261:   else if(flag == ØxØ8ØØ)
[D72]  262:     FLAG_2 &= Øxdf;
[185]  263:   if(*argptr == '+' || *argptr == '-')
[N52]  264:     (*nextpara)++;
[UL1]  265:   return(1);
[U7Ø]  266: }
[ØØØ]  267:
[2C6]  268: int Passwort_holen(argptr,nextpara)
[6P2]  269: char far *argptr;
[2R2]  27Ø: word *nextpara;
[9X2]  271: { if(*argptr == '-')
[U33]  272:   { *passwort = ØxØd;
[N52]  273:     (*nextpara)++;
[UL1]  274:     return(1);
[U7Ø]  275:   }
[V14]  276:   if(*argptr != ':') return(Ø);
[S25]  277:   for(stack = Ø; stack < 18; stack++)
[M45]  278:   { passwort[stack] = *++argptr;
[BM2]  279:     if(*argptr == ØxØd
[5S5]  28Ø:     || *argptr == '/' || *argptr == Øx2Ø)
[KA4]  281:     { passwort[stack] = ØxØd;
[YF3]  282:       *nextpara += stack + 1;
[PØ1]  283:       break;
[U7Ø]  284:     }
[CW2]  285:     else *argptr = '-';
[U7Ø]  286:   }
[RY2]  287:   return(stack < 18);
[U7Ø]  288: }
[ØØØ]  289:
[9A3]  29Ø: int Treiber_laden()
[Ø14]  291: { if(~new_paras & Øx2ØØØ)
[X84]  292:   { Parameter_uebergeben();
[TL1]  293:     return(Ø);
[U7Ø]  294:   }
[RN4]  295:   else if(paras_change & ØxØ8ØØ)
[X62]  296:   FLAG_2 |= Øx2Ø;
[UL1]  297:   return(1);
[U7Ø]  298: }
[ØØØ]  299:
[T26]  3ØØ: void Interruptvektoren_anpassen()
```

```
[D42]  3Ø1: { _disable();
[2T2]  3Ø2:   KB_START = Øx1Ø1E;
[LC2]  3Ø3:   KB_END = Øx1Ø3C;
[KT4]  3Ø4:   Tastaturpuffer_loeschen();
[5L5]  3Ø5:   _dos_setvect(Øx9,int_keyboard);
[3S5]  3Ø6:   old_ask_key = _dos_getvect(Øx16);
[ZH5]  3Ø7:   _dos_setvect(Øx16,int_ask_key);
[DN1]  3Ø8:   _enable();
[K83]  3Ø9:   new_paras &= Øxdfff;
[7U6]  31Ø:   print("Skeybgr ist installiert !\r\n\n");
[U7Ø]  311: }
[ØØØ]  312:
[ZC5]  313: void Tastatur_identifizieren()
[UV2]  314: { byte starttime;
[162]  315:   FLAG_1 &= Øxf7;
[SE5]  316:   if(new_paras & Øx1Ø8Ø) FLAG_2 &= Øxef;
[972]  317:   FLAG_2 |= Øx8Ø;
[GD2]  318:   outp(Øx6Ø,Øxf2);
[WF2]  319:   regs.h.ah = Øx2c;
[CL3]  32Ø:   int86(Øx21,&regs,&regs);
[Y94]  321:   starttime = 1ØØ - regs.h.dl;
[5W5]  322:   while((regs.h.dl+starttime) % 1ØØ < 2Ø)
[MY2]  323:   { regs.h.ah = Øx2c;
[CL3]  324:     int86(Øx21,&regs,&regs);
[U7Ø]  325:   }
[B62]  326:   FLAG_2 &= Øx3f;
[U7Ø]  327: }
[ØØØ]  328:
[1P4]  329: void Parameter_uebergeben()
[624]  33Ø: { if((FLAG_2 & Øx3Ø) == Øx3Ø)
[ØD3]  331:   new_paras |= Øx8ØØ;
[ZX2]  332:   regs.x.ax = Øx79Ø1;
[DP3]  333:   regs.x.bx = new_paras;
[TL3]  334:   int86(Øx16,&regs,&regs);
[D72]  335:   FLAG_2 &= Øxdf;
[183]  336:   if(new_paras & Øx8ØØØ)
[RP4]  337:   Parameterliste_ausgeben();
[U7Ø]  338: }
[ØØØ]  339:
[LØ6]  34Ø: int Parameterliste_ausgeben(void)
[DB3]  341: { register int lauf;
[636]  342:   int old = Treiberparameter_holen();
[U93]  343:   new_paras ^= ØxØ182;
[W63]  344:   if(*passwort != ØxØd)
[MC3]  345:   new_paras |= Øx4ØØØ;
[5K4]  346:   for(lauf = Ø; lauf < 15; lauf++)
[2A5]  347:   { printpara(paraliste[lauf]);
[8Y1]  348:     if(lauf == 11)
[TD4]  349:     { print(new_paras & ØxØØØ2 ?
[UH2]  35Ø:       "Caps" : "Shift");
[BD4]  351:       print("-Lock einschalten");
[U7Ø]  352:     }
[192]  353:     print("\r\n");
[U7Ø]  354:   }
[U93]  355:   new_paras ^= ØxØ182;
[SY1]  356:   return(old);
[U7Ø]  357: }
[ØØØ]  358:
[EU3]  359: void printpara(string)
[ØØØ]  36Ø:
[642]  361: char *string;
[Ø14]  362: { if(~new_paras & Øx2ØØØ)
[9T3]  363:   string[3] = new_paras
[U65]  364:   & paraflag[string[1]-'a']?'+':'-';
[9E2]  365:   print(string);
[U7Ø]  366: }
[ØØØ]  367:
[F85]  368: void Zeichentabellen_fuer_AT()
[6E4]  369: { if((paras & ØxcØ0Ø) == Øx4ØØØ)
[ØA3]  37Ø:   { code[16][41] = '<';
[ST2]  371:     code[17][41] = '>';
[ØW2]  372:     code[17][43] = '^';
[9T2]  373:     code[Ø4][Ø3] = '@';
[HV2]  374:     code[Ø4][26] = '[';
[TV2]  375:     code[Ø4][27] = ']';
[FZ2]  376:     code[Ø4][41] = '\\';
[U7Ø]  377:   }
[U7Ø]  378: }
[ØØØ]  379:
[KK4]  38Ø: dword TSR_Laenge(progname)
[5K2]  381: char *progname;
[D43]  382: { word prg_laenge;
[ØØØ]  383:
[XM2]  384:   #if __TURBOC__
[KG3]  385:   struct ffblk ffblock;
[LT4]  386:   findfirst(progname,&ffblock,
[184]  387:   FA_RDONLY | FA_HIDDEN);
[LØ5]  388:   prg_laenge = ffblock.ff_fsize;
[2TØ]  389:   #else
[6E3]  39Ø:   char huge *progstart;
[K73]  391:   char huge *progende;
[794]  392:   FP_SEG(progstart) = _psp;
[LL3]  393:   FP_OFF(progstart) = Ø;
[TB4]  394:   progende = (char huge*)&end;
[GN5]  395:   prg_laenge = progende - progstart;
[NZØ]  396:   #endif
[ØØØ]  397:
[6X2]  398:   prg_laenge += Øx1Ø;
[2W3]  399:   return(prg_laenge >> 4);
```

Listing: »skeybgr.c« ersetzt den normalen »keyb«

```
[U7Ø]  4ØØ: }
[ØØØ]  4Ø1:
[BK3]  4Ø2: int Ende(errorlevel)
[ØØØ]  4Ø3:
[FU2]  4Ø4: int errorlevel;
[HL3]  4Ø5: { switch(errorlevel)
[X91]  4Ø6:   { case 1:
[VY5]  4Ø7:     print("\a\r\n\nInterner Fehler !");
[PØ1]  4Ø8:     break;
[S41]  4Ø9:     case 2:
[M46]  41Ø:     print((paras_change&Øx5fff) == Øx5fff ?
[LM4]  411:     "keine Parameter geändert !" :
[8X5]  412:     "neue Parameter sind übergeben !");
[PØ1]  413:     break;
[V41]  414:     case 3:
[AU4]  415:     print("\aFehler in Parameter,"
[YL3]  416:           " Hilfe mit /h !\r\n");
[AU5]  417:     print("Weiter durch Tastendruck. ");
[1F2]  418:     regs.h.ah = ØxØØ;
[TL3]  419:     int86(Øx16,&regs,&regs);
[PØ1]  42Ø:     break;
[Y41]  421:     case 4:
[WV4]  422:     if(Parameterliste_ausgeben())
[6N4]  423:     print("frühere Version ist"
[KM2]  424:           " installiert !");
[ZRØ]  425:     else
[YY3]  426:     print("Skeybgr ist nicht"
[KM2]  427:           " installiert !");
[U7Ø]  428:   }
[TH2]  429:   print("\r\n\n");
[PD3]  43Ø:   return(errorlevel);
[U7Ø]  431: }
[ØØØ]  432:
[633]  433: void print(string)
[642]  434: char *string;
[FW4]  435: { regs.x.dx = FP_OFF(string);
[AF2]  436:   regs.h.ah = Øx9;
[LR2]  437:   while(*++string);
[Ø22]  438:   *string = '$';
[CL3]  439:   int86(Øx21,&regs,&regs);
[U7Ø]  44Ø: }
[ØØØ]  441:
[NN5]  442: void interrupt far int_keyboard()
[ØN3]  443: { scancode = inp(Øx6Ø);
[BU2]  444:   stack = inp(Øx61);
[ZS3]  445:   outp(Øx61,stack | Øx8Ø);
[FU2]  446:   outp(Øx61,stack);
[WY4]  447:   if(paras & Øx4ØØØ && ~FLAG_2 & Øx8Ø
[Ø33]  448:   && pass_chr == NULL)
[CW3]  449:   { regs.x.ax = scancode;
[7G2]  45Ø:     regs.h.ah = Øx4f;
[PL3]  451:     int86(Øx15,&regs,&regs);
[LL2]  452:     if(!regs.x.cflag)
[4R2]  453:     { outp(Øx2Ø,Øx2Ø);
[KC1]  454:       return;
[U7Ø]  455:     }
[B83]  456:     scancode = regs.h.al;
[U7Ø]  457:   }
[7F1]  458:   ascii = Ø;
[6B2]  459:   if(FLAG_2 & Øx8Ø)
[9E3]  46Ø:   { if(scancode == Øxfa)
[PU2]  461:     { paras |= Øx4ØØØ;
[972]  462:       FLAG_2 &= Øxbf;
[U7Ø]  463:     }
[5Ø4]  464:     else if(scancode == Øxfe)
[GD2]  465:     outp(Øx6Ø,Øxf2);
[CM3]  466:     else if(~FLAG_2 & Øx4Ø)
[BH5]  467:     { if(scancode != Øxab) FLAG_2 &= Øx3f;
[V53]  468:       else FLAG_2 |= Øx4Ø;
[U7Ø]  469:     }
[ZRØ]  47Ø:     else
[Ø85]  471:     if(scancode != Øx41) FLAG_2 &= Øx3f;
[ZRØ]  472:     else
[NL2]  473:     { FLAG_2 &= Øx3f;
[KL2]  474:       paras |= Øx8ØØØ;
[V62]  475:       FLAG_2 |= Øx1Ø;
[U7Ø]  476:     }
[U7Ø]  477:   }
[1Ø4]  478:   else if(scancode == Øxfa)
[UU2]  479:   { FLAG_3 |= Øx1Ø;
[SA2]  48Ø:     if(FLAG_3 & Øx4Ø)
[C73]  481:     { if(~paras & Øx8ØØ)
[MN2]  482:       { FLAG_3 &= Øxf8;
[ØA4]  483:         FLAG_3 |= (FLAG_Ø & Øx7Ø) >> 4;
[YS3]  484:         outp(Øx6Ø,FLAG_3 & ØxØ7);
[UL2]  485:         paras |= Øx8ØØ;
[U7Ø]  486:       }
[ZRØ]  487:       else
[KN2]  488:       { FLAG_3 &= Øxbf;
[GE2]  489:         paras &= Øxf7ff;
[U7Ø]  49Ø:       }
[U7Ø]  491:     }
[SF3]  492:     else if(paras & Øx8ØØ)
[AR4]  493:     { for(stack = Øx2ØØØ; stack--;);
[YB2]  494:       outp(Øx6Ø,ØxØØ);
[GE2]  495:       paras &= Øxf7ff;
[U7Ø]  496:     }
[U7Ø]  497:   }
[5Ø4]  498:   else if(scancode == Øxfe)
[YU2]  499:   { FLAG_3 |= Øx2Ø;
```

```
[SA2]  5ØØ:     if(FLAG_3 & Øx4Ø)
[C73]  5Ø1:     { if(~paras & Øx8ØØ)
[2M2]  5Ø2:       { FLAG_3 &= Øx6f;
[4F2]  5Ø3:         outp(Øx6Ø,Øxed);
[U7Ø]  5Ø4:       }
[CD4]  5Ø5:       else outp(Øx6Ø,FLAG_3 & ØxØ7);
[U7Ø]  5Ø6:     }
[SF3]  5Ø7:     else if(paras & Øx8ØØ)
[LD2]  5Ø8:     outp(Øx6Ø,Øxf3);
[U7Ø]  5Ø9:   }
[3U5]  51Ø:   else if(FLAG_Ø & Øx4 && scancode == 7Ø)
[ØM3]  511:   { if(pass_chr == NULL)
[X13]  512:     { BREAK_FLAG = Øx8Ø;
[XM3]  513:       int86(Øx1b,&regs,&regs);
[KT4]  514:       Tastaturpuffer_loeschen();
[U7Ø]  515:     }
[U7Ø]  516:   }
[W63]  517:   else if(FLAG_1 & Øx8
[DD3]  518:   && (scancode & Øx7f) != 29
[BØ5]  519:   && scancode != 197 && scancode != 224)
[525]  52Ø:   { if(FLAG_Ø & Øx4 && scancode == 69
[855]  521:     || _INTERVALL(scancode,128,223))
[162]  522:     FLAG_1 &= Øxf7;
[U7Ø]  523:   }
[5U5]  524:   else if(paras & Øx1ØØØ && scancode == 59)
[GT4]  525:   { amerikanische_Tastatur();
[CF2]  526:     paras &= Øxfeff;
[U7Ø]  527:   }
[ØU5]  528:   else if(paras & Øx1ØØØ && scancode == 6Ø)
[N14]  529:   { deutsche_Tastatur();
[CL2]  53Ø:     paras |= Øx1ØØ;
[U7Ø]  531:   }
[BU5]  532:   else if(paras & Øx1ØØØ && scancode == 83)
[ØS1]  533:   (*reset)();
[ZRØ]  534:   else
[V35]  535:   /* Statusflags aktualisieren */
[C24]  536:   { scancode_gefunden = 1;
[ZP2]  537:     switch(scancode)
[3F1]  538:     { case 54:
[PL2]  539:       if(~FLAG_2 & Øx2)
[MU2]  54Ø:       { FLAG_Ø |= Øx1;
[ML4]  541:         if(paras & Øx2)FLAG_Ø &= Øxbf;
[U7Ø]  542:       }
[PØ1]  543:       break;
[L91]  544:       case 182:
[SU4]  545:       if(~FLAG_2 & Øx2) FLAG_Ø &= Øxfe;
[PØ1]  546:       break;
[S51]  547:       case 42:
[PL2]  548:       if(~FLAG_2 & Øx2)
[NU2]  549:       { FLAG_Ø |= Øx2;
[ML4]  55Ø:         if(paras & Øx2)FLAG_Ø &= Øxbf;
[U7Ø]  551:       }
[PØ1]  552:       break;
[D91]  553:       case 17Ø:
[PU4]  554:       if(~FLAG_2 & Øx2) FLAG_Ø &= Øxfd;
[PØ1]  555:       break;
[E61]  556:       case 29:
[BX4]  557:       if(FLAG_2 & Øx2) FLAG_2 |= Øx4;
[F53]  558:       else FLAG_1 |= Øx1;
[PØ1]  559:       break;
[C91]  56Ø:       case 157:
[FS4]  561:       if(FLAG_2 & Øx2) FLAG_2 &= Øxfb;
[N43]  562:       else FLAG_1 &= Øxfe;
[PØ1]  563:       break;
[B61]  564:       case 56:
[FX4]  565:       if(FLAG_2 & Øx2) FLAG_2 |= Øx8;
[L53]  566:       else FLAG_1 |= Øx2;
[PØ1]  567:       break;
[N91]  568:       case 184:
[4S4]  569:       if(FLAG_2 & Øx2) FLAG_2 &= Øxf7;
[H43]  57Ø:       else FLAG_1 &= Øxfd;
[PØ1]  571:       break;
[C61]  572:       case 84:
[272]  573:       FLAG_1 |= Øx4;
[PØ1]  574:       break;
[U81]  575:       case 212:
[272]  576:       FLAG_1 &= Øxfb;
[PØ1]  577:       break;
[T51]  578:       case 7Ø:
[F65]  579:       if(~FLAG_1 & Øx1Ø && ~FLAG_Ø & Øx4)
[R52]  58Ø:       FLAG_Ø ^= Øx1Ø;
[S62]  581:       FLAG_1 |= Øx1Ø;
[PØ1]  582:       break;
[X91]  583:       case 198:
[C72]  584:       FLAG_1 &= Øxef;
[PØ1]  585:       break;
[U61]  586:       case 69:
[UD3]  587:       if(FLAG_1 & Øx2Ø) break;
[6A2]  588:       if(FLAG_Ø & Øx4)
[ZT2]  589:       { if(FLAG_Ø & Øx3)
[MR3]  59Ø:         { if(*copyright != Øxd
[Ø33]  591:           && pass_chr == NULL)
[3P3]  592:           { key = _CPEEK(Øx46cL);
[SY3]  593:             pass_chr = copyright+18;
[9Ø4]  594:             for(end = Ø; end < 18; end++)
[823]  595:             { pass_chr[end]
[A44]  596:               = key ^ copyright[end];
[7L3]  597:               copyright[end] = end;
[U7Ø]  598:             }
[E31]  599:             beep();
```

```
[U7Ø]  6ØØ:            }
[U7Ø]  6Ø1:          }
[MA3]  6Ø2:          else if(FLAG_Ø & Øx Ø8)
[4Y4]  6Ø3:          { Tastaturpuffer_loeschen();
[W72]  6Ø4:            ALT_ASCII = Ø;
[U7Ø]  6Ø5:          }
[964]  6Ø6:          else if(pass_chr == NULL)
[PU2]  6Ø7:          { FLAG_2 |= Øx Ø1;
[E72]  6Ø8:            FLAG_1 |= Øx Ø8;
[U7Ø]  6Ø9:          }
[U7Ø]  61Ø:        }
[H33]  611:        else FLAG_Ø ^= Øx2Ø;
[U62]  612:        FLAG_1 |= Øx2Ø;
[PØ1]  613:        break;
[W91]  614:        case 197:
[A72]  615:        FLAG_1 &= Øxdf;
[PØ1]  616:        break;
[L61]  617:        case 58:
[LL2]  618:        if(~FLAG_1 & Øx4Ø)
[CB5]  619:        { if(paras & Øx ØØØ2) FLAG_Ø |= Øx4Ø;
[R33]  62Ø:          else FLAG_Ø ^= Øx4Ø;
[U7Ø]  621:        }
[YG2]  622:        FLAG_1 |= Øx4Ø;
[PØ1]  623:        break;
[R91]  624:        case 186:
[672]  625:        FLAG_1 &= Øxbf;
[PØ1]  626:        break;
[461]  627:        case 82:
[XP2]  628:        if(!(FLAG_Ø & Øx Øf)
[HD2]  629:        && ~FLAG_1 & Øx8Ø)
[TS2]  63Ø:        { FLAG_Ø ^= Øx8Ø;
[672]  631:          FLAG_1 |= Øx8Ø;
[U7Ø]  632:        }
[ØM3]  633:        scancode_gefunden = Ø;
[PØ1]  634:        break;
[S81]  635:        case 21Ø:
[G62]  636:        FLAG_1 &= Øx7f;
[PØ1]  637:        break;
[Ø91]  638:        case 224:
[Z62]  639:        FLAG_2 |= Øx Ø2;
[PØ1]  64Ø:        break;
[191]  641:        case 225:
[W62]  642:        FLAG_2 |= Øx Ø1;
[PØ1]  643:        break;
[8E1]  644:        default:
[7Y3]  645:        if(scancode > Øx7f) break;
[ØM3]  646:        scancode_gefunden = Ø;
[U7Ø]  647:      }
[MD5]  648:      if(scancode ==  5Ø) paras |= Øx2ØØØ;
[1B5]  649:      if(scancode == 178) paras &= Øxdfff;
[F55]  65Ø:      if(FLAG_1 & Øx Ø1 || FLAG_2 & Øx Ø4)
[Z62]  651:      FLAG_Ø |= Øx Ø4;
[ZRØ]  652:      else
[Z62]  653:      FLAG_Ø &= Øxfb;
[S55]  654:      if(FLAG_1 & Øx Ø2 || FLAG_2 & Øx Ø8)
[B72]  655:      FLAG_Ø |= Øx Ø8;
[ZRØ]  656:      else
[GN2]  657:      { FLAG_Ø &= Øxf7;
[LS4]  658:        if(scancode == 184 && ALT_ASCII)
[WB2]  659:        { scancode = Ø;
[YZ2]  66Ø:          ascii = ALT_ASCII;
[W72]  661:          ALT_ASCII = Ø;
[ØM3]  662:          scancode_gefunden = Ø;
[U7Ø]  663:        }
[U7Ø]  664:      }
[BL3]  665:      if((FLAG_1 & Øx Ø3) == Øx Ø3)
[BL2]  666:      paras |= Øx1ØØØ;
[ZRØ]  667:      else
[DF2]  668:      paras &= Øxefff;
[NR3]  669:      Lockstatus_pruefen();
[HW2]  67Ø:      if(scancode > Øx7f)
[2M3]  671:      scancode_gefunden = 1;
[3U5]  672:      /* auswerten von Alt-Zifferblock */
[KW4]  673:      if(FLAG_Ø & Øx Ø8 && ~FLAG_2 & Øx Ø2)
[LP4]  674:      if(_INTERVALL(scancode,71,82))
[DM5]  675:      { stack = numblock[scancode - 71];
[Z94]  676:        if(_INTERVALL(stack,48,57))
[7M4]  677:        { stack += ALT_ASCII * 1Ø - 48;
[BZ3]  678:          ALT_ASCII = stack % Øx1ØØ;
[2M3]  679:          scancode_gefunden = 1;
[U7Ø]  68Ø:        }
[U7Ø]  681:      }
[VM3]  682:      if(!scancode_gefunden)
[WH2]  683:      { ALT_ASCII = Ø;
[2L4]  684:        /* aktuelles Ascii Zeichen, */
[D45]  685:        /* Codepage für Steuertaste */
[5N4]  686:        stack = "\Ø\1\1\1\3\3\3\3\2\2"
[Y42]  687:                "\2\2\4\5\5\5"
[9H2]  688:        [FLAG_Ø & Øx Øf];
[ML4]  689:        if(stack == 2 && ~FLAG_2 & Øx Ø8)
[GG1]  69Ø:        stack = 6;
[6S4]  691:        if(FLAG_Ø & Øx4Ø && paras & Øx ØØØ2)
[UN3]  692:        if(stack % 4 == Ø) stack++;
[KL4]  693:        else if(stack % 4 == 1) stack--;
[GS4]  694:        if(scancode == 86) scancode = 42;
[PK4]  695:        if(_INTERVALL(scancode,1,57))
[3A2]  696:        { if(stack < 6)
[885]  697:          ascii = code[stack][scancode];
[TU4]  698:          if(ascii == 'α') scancode = Ø;
[SS4]  699:          if(scancode == 42) scancode = 86;
```

```
[U7Ø]  7ØØ:        }
[MP4]  7Ø1:        if(_INTERVALL(scancode,71,83))
[9P3]  7Ø2:        { stack = FLAG_Ø & Øx2f;
[FV4]  7Ø3:          if(_INTERVALL(stack,Øx Ø1,Øx Ø3)
[YN2]  7Ø4:          || stack == Øx2Ø)
[K85]  7Ø5:          ascii = numblock[scancode - 71];
[XR2]  7Ø6:          if(!(FLAG_Ø & Øx Øc))
[V23]  7Ø7:          { switch(scancode)
[H84]  7Ø8:            { case 74: ascii = '-'; break;
[1Ø4]  7Ø9:              case 78: ascii = '+'; break;
[F73]  71Ø:              case 76: if(ascii == Ø
[NK2]  711:              && ~paras & Øx ØØØ1)
[WZ1]  712:              ascii = Øxfø;
[U7Ø]  713:            }
[U7Ø]  714:          }
[S72]  715:          if(FLAG_Ø & Øx Ø8
[SZ2]  716:          && (scancode == 74 ||
[EA2]  717:              scancode == 78)
[HL4]  718:          && ~paras & Øx ØØØ1) ascii = Øxfø;
[U7Ø]  719:        }
[8A2]  72Ø:        if(FLAG_2 & Øx Ø2)
[Y55]  721:        { if(_INTERVALL(scancode,71,83))
[W72]  722:          if(FLAG_2 & Øx Ø8
[VL2]  723:          || paras & Øx1ØØØ)
[C15]  724:          ascii = numblock[scancode-58];
[ZRØ]  725:          else
[YC4]  726:          ascii = paras & Øx Ø1 ? Ø : Øxeø;
[XR2]  727:          if(!(FLAG_Ø & Øx Øc))
[MR4]  728:          { if(scancode == 53) ascii = '/';
[Y14]  729:            if(scancode == 55) ascii = Ø;
[U7Ø]  73Ø:          }
[U7Ø]  731:        }
[YK3]  732:        else if(paras & Øx8ØØØ &&
[1A2]  733:                scancode == 55)
[CB4]  734:        if(~FLAG_Ø & Øx Ø8) ascii = '*';
[UR3]  735:        else if(~paras & Øx ØØØ1)
[WZ1]  736:                ascii = Øxfø;
[4G4]  737:        if(_INTERVALL(scancode,1,57)
[8U4]  738:        && FLAG_Ø & Øx4Ø && ~paras & Øx ØØØ2)
[6P4]  739:        { if(_INTERVALL(ascii,'a','z')
[PN4]  74Ø:          || _INTERVALL(ascii,'A','Z'))
[A72]  741:          ascii ^= Øx2Ø;
[ZRØ]  742:          else
[W25]  743:          for(stack = 6; stack < 14; stack++)
[MX3]  744:          for(end = 1; end < 9; end++)
[XW4]  745:          if(ascii == code[stack][end])
[2B4]  746:          { stack += stack < 1Ø ? 4 : -4;
[BB4]  747:            ascii = code[stack][end];
[XP1]  748:            stack = 14;
[PØ1]  749:            break;
[U7Ø]  75Ø:          }
[U7Ø]  751:        }
[A32]  752:        if(hochkomma)
[G42]  753:        switch(ascii)
[SP1]  754:        { case '^':
[EØ2]  755:          case '·':
[AØ2]  756:          case '°':
[2H1]  757:          case '~':
[6Ø2]  758:          case '≈':
[GU3]  759:          if(~paras & Øx Ø2ØØ) break;
[AD1]  76Ø:          case '`':
[W62]  761:          hochkomma = Ø;
[2H1]  762:          case '\'':
[WL2]  763:          if(scancode == 13)
[W62]  764:          hochkomma = Ø;
[U7Ø]  765:        }
[XU3]  766:        if(scancode && hochkomma)
[C45]  767:        { if(_INTERVALL(scancode,18,3Ø)
[884]  768:          && _INTERVALL(ascii,'A','z'))
[PV1]  769:          { stack = 5;
[CS4]  77Ø:            if(hochkomma == '^') stack = 1Ø;
[PV4]  771:            if(hochkomma == '\'') stack = 11;
[US4]  772:            if(hochkomma == '`') stack = 12;
[975]  773:            if(hochkomma == '·') stack = 13;
[NV3]  774:            if(ascii & Øx2Ø) stack -= 4;
[UR3]  775:            else if(~paras & Øx Ø2ØØ)
[EG1]  776:                    stack = 5;
[2M3]  777:            end = "\1\Ø\Ø\Ø\2\3\4\Ø"
[EV1]  778:                  "\Ø\Ø\Ø\Ø\5"
[SK2]  779:            [scancode - 18];
[934]  78Ø:            end = code[stack][end];
[Ø33]  781:            if(end != Ø && stack > 5)
[762]  782:            { ascii = end;
[6X1]  783:              scancode = Ø;
[W62]  784:              hochkomma = Ø;
[U7Ø]  785:            }
[U7Ø]  786:          }
[VZ2]  787:          else if(ascii == ' ')
[M93]  788:          { ascii = hochkomma;
[6X1]  789:            scancode = Ø;
[W62]  79Ø:            hochkomma = Ø;
[U7Ø]  791:          }
[U7Ø]  792:        }
[N24]  793:        if(scancode && hochkomma &&
[LP1]  794:           ascii < 128)
[S93]  795:        { if(hochkomma == '~')
[G42]  796:          switch(ascii)
[DZ4]  797:          { case 'a': ascii = 'ã'; break;
[Ø64]  798:            case 'A': ascii = 'Ã'; break;
```

Listing: »skeybgr.c« ersetzt den normalen »keyb«

```
[H64]  799:            case 'n': ascii = 'ñ'; break;
[K64]  8ØØ:            case 'N': ascii = 'Ñ'; break;
[N84]  8Ø1:            case 'o': ascii = 'Σ'; break;
[P84]  8Ø2:            case 'O': ascii = 'σ'; break;
[U7Ø]  8Ø3:          }
[YG3]  8Ø4:          if(hochkomma == '°')
[G42]  8Ø5:          switch(ascii)
[DT4]  8Ø6:          { case 'a': ascii = 'å'; break;
[844]  8Ø7:            case 'A': ascii = 'Å'; break;
[U7Ø]  8Ø8:          }
[VG3]  8Ø9:          if(hochkomma == '≈')
[G42]  81Ø:          switch(ascii)
[MT4]  811:          { case 'c': ascii = 'ç'; break;
[144]  812:            case 'C': ascii = 'Ç'; break;
[U7Ø]  813:          }
[KC3]  814:          if(hochkomma == '\'' &&
[S42]  815:             paras & Øx Ø2ØØ)
[G42]  816:          switch(ascii)
[F45]  817:          { case 'y': ascii = '∞'; break;
[6A4]  818:            case 'Y': ascii = 'φ'; break;
[U7Ø]  819:          }
[KC3]  82Ø:          if(hochkomma == '\'' &&
[NT1]  821:             ascii == 'E')
[BX1]  822:            ascii = 'É';
[PP3]  823:          if(hochkomma == '·' &&
[5W1]  824:             ascii == 'y')
[3Y1]  825:            ascii = 'ÿ';
[NB4]  826:          if(ascii > 127) hochkomma = Ø;
[U7Ø]  827:        }
[A32]  828:        if(hochkomma)
[DR5]  829:        { /* Hochkomma in Tastaturpuffer */
[HN3]  83Ø:          asciicode = hochkomma;
[ZV1]  831:          keycode = Ø;
[E73]  832:          if(pass_chr != NULL)
[LZ3]  833:          Tastatur_entriegeln();
[ZRØ]  834:          else
[MZ4]  835:          Taste_in_Puffer_schreiben();
[E31]  836:          beep();
[W62]  837:          hochkomma = Ø;
[U7Ø]  838:        }
[HZ4]  839:        /* erweiterte Scancodes,           */
[F86]  84Ø:        /* zusätzliche spezielle Scancodes */
[6R4]  841:        if(_INTERVALL(scancode,87,88))
[N33]  842:        switch(FLAG_Ø & ØxØf)
[EU1]  843:        { case ØxØc:
[2P3]  844:          case ØxØ8: scancode += 2;
[UN3]  845:          case Øx Ø4: scancode += 2;
[AK1]  846:          case Øx Ø1:
[CK1]  847:          case Øx Ø2:
[SN3]  848:          case Øx Ø3: scancode += 2;
[VV3]  849:          default:    scancode += 46;
[U7Ø]  85Ø:        }
[ZP4]  851:        if(_INTERVALL(scancode,59,68))
[N33]  852:        switch(FLAG_Ø & ØxØf)
[EU1]  853:        { case Øx Øc:
[WU3]  854:          case Øx Ø8: scancode += 1Ø;
[MU3]  855:          case Øx Ø4: scancode += 1Ø;
[AK1]  856:          case Øx Ø1:
[CK1]  857:          case Øx Ø2:
[XU3]  858:          case Øx Ø3: scancode += 25;
[U7Ø]  859:        }
[6A2]  86Ø:        if(FLAG_1 & Øx Ø2)
[4U2]  861:        { if(FLAG_2 & Øx Ø2)
[N45]  862:          { if(_INTERVALL(scancode,71,83)
[PK2]  863:              && ~paras & Øx1ØØØ)
[EG2]  864:              { scancode += 8Ø;
[7F1]  865:                ascii = Ø;
[U7Ø]  866:              }
[ØM2]  867:              if(scancode == 53)
[NG2]  868:              { scancode = 164;
[7F1]  869:                ascii = Ø;
[U7Ø]  87Ø:              }
[7M2]  871:              if(scancode == 28)
[B92]  872:                 scancode = 166;
[U7Ø]  873:          }
[VH4]  874:          if(_INTERVALL(scancode,2,13))
[EB2]  875:          scancode += 118;
[MF4]  876:          if(_INTERVALL(scancode,1,53)
[YG5]  877:          &&!_INTERVALL(code[Ø][scancode],
[UZØ]  878:                        'a','z')
[Z14]  879:          && ~FLAG_Ø & Øx Ø4 && ~paras
[RZØ]  88Ø:          & ØxØØØ1)
[WZ1]  881:          ascii = Øxfø;
[ØM2]  882:          if(scancode == 15)
[PG2]  883:          { scancode = 165;
[7F1]  884:            ascii = Ø;
[U7Ø]  885:          }
[DG4]  886:          if(scancode == 57) ascii = Øx2Ø;
[U7Ø]  887:        }
[BM3]  888:        if((FLAG_Ø & Øx Øc) == Øx Ø4)
[Y55]  889:        { if(_INTERVALL(scancode,71,83))
[584]  89Ø:          scancode =strg_scancode[
[ZØ2]  891:                    scancode-71];
[8X4]  892:          if(scancode == 15) scancode = 148;
[RW4]  893:          if(scancode == 53 && FLAG_2 & Øx Ø2)
[KG2]  894:          { scancode = 149;
[7F1]  895:            ascii = Ø;
[U7Ø]  896:          }
[4M2]  897:          if(scancode == 55)
[LB1]  898:          if(ascii)
```

```
[DG2]  899:            { scancode = 15Ø;
[7F1]  9ØØ:              ascii = Ø;
[U7Ø]  9Ø1:            }
[833]  9Ø2:            else scancode = 114;
[U7Ø]  9Ø3:        }
[B44]  9Ø4:        if(scancode && paras & Øx Ø2ØØ)
[YY2]  9Ø5:        hochkomma = ascii;
[G42]  9Ø6:        switch(ascii)
[2P1]  9Ø7:        { case '\'':
[ZL2]  9Ø8:            if(scancode == 43)
[W62]  9Ø9:            hochkomma = Ø;
[AD1]  91Ø:            case '`':
[WL2]  911:            if(scancode == 13)
[YY2]  912:            hochkomma = ascii;
[2D1]  913:            case '^':
[EØ2]  914:            case '·':
[AØ2]  915:            case '°':
[2H1]  916:            case '~':
[M33]  917:            case '≈': break;
[FT3]  918:            default: hochkomma = Ø;
[U7Ø]  919:        }
[W32]  92Ø:        if(!hochkomma
[ZY3]  921:        && (scancode != 55 || ascii
[SN2]  922:        || FLAG_Ø & Øx Øc))
[YT5]  923:        { /* Zeichen in Tastaturpuffer,     */
[695]  924:          /* Ascii- und Scancode von Taste */
[MC2]  925:          if(paras & Øx Ø1ØØ)
[2S4]  926:          if(scancode == 21) scancode = 44;
[PM3]  927:          else if(scancode == 44)
[522]  928:                  scancode = 21;
[WX2]  929:          asciicode = ascii;
[X83]  93Ø:          keycode = scancode;
[E73]  931:          if(pass_chr != NULL)
[LZ3]  932:          Tastatur_entriegeln();
[MA3]  933:          else if(paras & Øx2ØØØ
[X84]  934:          && FLAG_1 & Øx Ø2 && FLAG_2 & Øx Ø8
[DB3]  935:          && (FLAG_Ø & Øx7f) >= Øx7d
[ZM4]  936:          && BUFFER_HEAD == BUFFER_TAIL)
[NV4]  937:          { char *text = programname + 2;
[3U3]  938:            for(stack = Ø; stack < 25;
[2K1]  939:                stack += 2)
[314]  94Ø:            { if(stack == 18) text += 7;
[GC3]  941:              asciicode = *(text++);
[ZV1]  942:              keycode = Ø;
[MZ4]  943:              Taste_in_Puffer_schreiben();
[U7Ø]  944:            }
[U7Ø]  945:          }
[RV5]  946:          else Taste_in_Puffer_schreiben();
[5K3]  947:          if(asciicode < Ø) beep();
[U7Ø]  948:        }
[U7Ø]  949:      }
[U7Ø]  95Ø:    }
[LØ5]  951:    if(scancode != 224) FLAG_2 &= Øxfd;
[DN1]  952:    _enable();
[YB2]  953:    outp(Øx2Ø,Øx2Ø);
[6X3]  954:    if(scancode == 55 && !ascii
[8U2]  955:    && PRINTSCREEN != 1
[RH2]  956:    && !(FLAG_Ø & Øx Øc))
[ML3]  957:    int86(Øx Ø5,&regs,&regs);
[WM4]  958:    else if((scancode & Øx7f) == 84
[P62]  959:    && paras & Øx4ØØØ)
[Ø43]  96Ø:    { regs.x.ax = Øx85ØØ;
[SM4]  961:      if(scancode & Øx8Ø) regs.x.ax++;
[PL3]  962:      int86(Øx15,&regs,&regs);
[U7Ø]  963:    }
[PK4]  964:    if(FLAG_1 & Øx Ø8 && FLAG_2 & Øx Ø1)
[ØP2]  965:    { FLAG_2 &= Øxfe;
[143]  966:      while(FLAG_1 & Øx Ø8);
[U7Ø]  967:    }
[KC1]  968:    return;
[U7Ø]  969: }
[ØØØ]  97Ø:
[367]  971: void interrupt far int_ask_key(INT_REGS)
[683]  972: unsigned INT_REGS;
[NV2]  973: { word ah = ax >> 8;
[8D4]  974:   if(~paras & ØxØØ8Ø && ah != Øx79)
[AE1]  975:   ah = Øxff;
[DN1]  976:   _enable();
[224]  977:   if(ah == ØxØØ || ah == Øx1Ø)
[NNØ]  978:   { do
[BX1]  979:     { _enable();
[NR3]  98Ø:       Lockstatus_pruefen();
[NV1]  981:       _disable();
[566]  982:     } while(BUFFER_HEAD == BUFFER_TAIL);
[3F4]  983:     Taste_aus_Puffer_lesen();
[242]  984:     BUFFER_HEAD
[AX6]  985:     = _NEXT(BUFFER_HEAD,KB_START,KB_END-2);
[DN1]  986:     _enable();
[L75]  987:     if(keycode > 7Ø && asciicode == Øxeø
[ØL1]  988:     && ah == ØxØØ
[TY4]  989:     || keycode && asciicode == Øxfø)
[U52]  99Ø:     asciicode = Ø;
[UP4]  991:     ax = keycode * Øx1ØØ + asciicode;
[KC1]  992:     return;
[U7Ø]  993:   }
[GN4]  994:   else if(ah == Øx Ø1 || ah == Øx11)
[164]  995:   { Lockstatus_pruefen();
[NV1]  996:     _disable();
[5V4]  997:     if(BUFFER_HEAD == BUFFER_TAIL)
```

```
[1T2]  998:      { flags |= 0x0004;
[KC1]  999:        return;
[U70] 1000:      }
[HD2] 1001:      flags &= 0xffbf;
[3F4] 1002:      Taste_aus_Puffer_lesen();
[DN1] 1003:      _enable();
[L75] 1004:      if(keycode > 70 && asciicode == 0xe0
[2L1] 1005:      && ah == 0x01
[TY4] 1006:      || keycode && asciicode == 0xf0)
[U52] 1007:      asciicode = 0;
[UP4] 1008:      ax = keycode * 0x100 + asciicode;
[KC1] 1009:      return;
[U70] 1010:    }
[MN4] 1011:    else if(ah == 0x02 || ah == 0x12)
[CX2] 1012:    { ax = FLAG_1 & 0x73;
[KX2] 1013:      ax |= FLAG_2 & 0x0c;
[VM3] 1014:      ax |= (FLAG_1 & 0x04) << 5;
[351] 1015:      ax <<= 8;
[X12] 1016:      ax |= FLAG_0;
[KC1] 1017:      return;
[U70] 1018:    }
[6S2] 1019:    else if(ah == 0x05)
[D42] 1020:    { _disable();
[2P3] 1021:      asciicode = cx & 0x00ff;
[VS2] 1022:      keycode = cx >> 8;
[MZ4] 1023:      Taste_in_Puffer_schreiben();
[ZE2] 1024:      if(asciicode < 0)
[473] 1025:      { ax = (ax & 0xff00) + 1;
[XH2] 1026:        flags |= 0x0001;
[U70] 1027:      }
[ZR0] 1028:      else
[U82] 1029:      { ax &= 0xff00;
[TD2] 1030:        flags &= 0xfffe;
[U70] 1031:      }
[DN1] 1032:      _enable();
[KC1] 1033:      return;
[U70] 1034:    }
[EZ2] 1035:    else if(ax == 0x7900)
[0Z1] 1036:    { ax = 0x7777;
[GZ2] 1037:      bx = paras & 0xe3ff;
[PN3] 1038:      bx |= (FLAG_0 & 0x70) << 6;
[M32] 1039:      cx = VERSION;
[RN3] 1040:      dx = paras & 0x0100 ? 1 : 0;
[TD2] 1041:      flags &= 0xfffe;
[KC1] 1042:      return;
[U70] 1043:    }
[FZ2] 1044:    else if(ax == 0x7901)
[DT2] 1045:    { paras &= 0xfc10;
[P93] 1046:      paras |= bx & 0x03ff;
[F62] 1047:      FLAG_0 &= 0x8f;
[VK3] 1048:      FLAG_0 |= (bx >> 6) & 0x70;
[NR3] 1049:      Lockstatus_pruefen();
[F33] 1050:      while(FLAG_3 & 0x40);
[8T2] 1051:      if(~paras & 0x0080)
[KV2] 1052:      { paras &= 0xfffb;
[FM4] 1053:        if(paras & 0x8000) FLAG_2 &= 0xef;
[U70] 1054:      }
[ZR0] 1055:      else
[SV2] 1056:      { paras &= 0xfffe;
[7X4] 1057:        if(paras & 0x8000) FLAG_2 |= 0x10;
[U70] 1058:      }
[NV1] 1059:      _disable();
[SC2] 1060:      if(paras & 0x0004)
[S53] 1061:      { KB_START = 0x0200;
[LC2] 1062:        KB_END = 0x0300;
[U70] 1063:      }
[ZR0] 1064:      else
[B83] 1065:      { KB_START = 0x001e;
[AF2] 1066:        KB_END = 0x003e;
[U70] 1067:      }
[Y54] 1068:      BUFFER_HEAD = KB_START;
[U74] 1069:      BUFFER_TAIL = KB_START;
[DN1] 1070:      _enable();
[H85] 1071:      if(paras & 0x0008) numblock[12] ='.';
[5T3] 1072:      else numblock[12] = ',';
[885] 1073:      if(paras & 0x0020) RESETFLAG = 0x1234;
[5R3] 1074:      else RESETFLAG = 0x0000;
[V16] 1075:      if(paras & 0x0100) deutsche_Tastatur();
[2E5] 1076:      else amerikanische_Tastatur();
[904] 1077:      if((paras & 0x8010) == 0x8010)
[AV2] 1078:      { paras |= 0x0800;
[LD2] 1079:        outp(0x60,0xf3);
[B83] 1080:        while(paras & 0x0800);
[U70] 1081:      }
[TD2] 1082:      flags &= 0xfffe;
[KC1] 1083:      return;
[U70] 1084:    }
[HZ2] 1085:    else if(ax == 0x7903)
[U82] 1086:    { ax &= 0xff00;
[XH2] 1087:      flags |= 0x0001;
[592] 1088:      ah = bx & 0x00ff;
[0D2] 1089:      if(ah < 4) ah += 14;
[083] 1090:      else if(ah < 8) ah -= 2;
[ZR0] 1091:      else
[7H2] 1092:      { ax++; return;
[U70] 1093: }
[VS2] 1094:      keycode = cx >> 8;
[9K4] 1095:      if(keycode == 86) keycode = 42;
[GH4] 1096:      if(!_INTERVALL(keycode,1,53))
[7R2] 1097:      { ax += 2; return;
[U70] 1098: }
[144] 1099:      dx = code[ah][keycode];
[H65] 1100:      code[ah][keycode] = cx & 0x00ff;
[TD2] 1101:      flags &= 0xfffe;
[KC1] 1102:      return;
[U70] 1103:    }
[KZ2] 1104:    else if(ax == 0x7904)
[0004] 1105:   { bx = FP_OFF(passwort);
[6P3] 1106:      di = FP_SEG(passwort);
[TR5] 1107:      if(pass_chr == NULL) flags &= 0xfffe;
[E23] 1108:      else flags |= 0x0001;
[KC1] 1109:      return;
[U70] 1110:    }
[0T2] 1111:    else if(ah == 0x79)
[ZS2] 1112:    { flags |= 0x0001;
[KC1] 1113:      return;
[U70] 1114:    }
[XM2] 1115:    #if __TURBOC__
[VD1] 1116:    _CX = cx;
[DD1] 1117:    _AX = ax;
[NZ0] 1118:    #endif
[RN2] 1119:    GO_OLD_IRQ16();
[XM2] 1120:    #if __TURBOC__
[PB1] 1121:    ax = _AX;
[HS2] 1122:    __emit__(0x9f);
[392] 1123:    flags &= 0xff00;
[D82] 1124:    flags |= _AH;
[NZ0] 1125:    #endif
[U70] 1126: }
[000] 1127:
[BA5] 1128: void Tastaturpuffer_loeschen()
[SU6] 1129: { BUFFER_HEAD = BUFFER_TAIL = KB_START;
[U70] 1130: }
[000] 1131:
[ML4] 1132: void Tastatur_entriegeln()
[8B4] 1133: { stack = key ^ asciicode;
[EH3] 1134:   if(*pass_chr == stack)
[4X4] 1135:   { pass_chr[-18] = asciicode;
[K53] 1136:     if(asciicode == 0x0d)
[B43] 1137:     { pass_chr = NULL;
[2C2] 1138:       asciicode = -1;
[U70] 1139:     }
[9N2] 1140:     else pass_chr++;
[U70] 1141:   }
[ZR0] 1142:   else
[734] 1143:   pass_chr = copyright + 18;
[U70] 1144: }
[000] 1145:
[3S5] 1146: void Taste_in_Puffer_schreiben()
[3V5] 1147: { end = _NEXT(BUFFER_TAIL,KB_START,
[EL1] 1148:                KB_END-2);
[LB3] 1149:   if(BUFFER_HEAD != end)
[YU5] 1150:   { BIOSMEM(BUFFER_TAIL) = asciicode;
[9T5] 1151:     BIOSMEM(BUFFER_TAIL + 1) = keycode;
[L23] 1152:     BUFFER_TAIL = end;
[U70] 1153:   }
[L13] 1154:   else asciicode = -1;
[U70] 1155: }
[000] 1156:
[DB5] 1157: void Taste_aus_Puffer_lesen()
[E16] 1158: { keycode = BIOSMEM(BUFFER_HEAD + 1);
[XL5] 1159:   asciicode = BIOSMEM(BUFFER_HEAD);
[U70] 1160: }
[000] 1161:
[WS1] 1162: void beep()
[7Z2] 1163: { stack = inp(0x61);
[SS3] 1164:   outp(0x61,stack | 0x03);
[WM3] 1165:   for(end = 0x2000; end--;);
[FU2] 1166:   outp(0x61,stack);
[U70] 1167: }
[000] 1168:
[KK4] 1169: void Lockstatus_pruefen()
[AT3] 1170: { if((FLAG_0 & 0x70) >> 4 !=
[062] 1171:       (FLAG_3 & 0x07))
[UL2] 1172:   { FLAG_3 &= 0x4f;
[LW3] 1173:     if((paras & 0x8004) == 0x8000
[424] 1174:     && ~FLAG_3 & 0x40 && ~paras
[E01] 1175:     & 0x0800)
[6V2] 1176:     { FLAG_3 |= 0x40;
[4F2] 1177:       outp(0x60,0xed);
[U70] 1178:     }
[U70] 1179:   }
[U70] 1180: }
[000] 1181:
[L85] 1182: void amerikanische_Tastatur()
[6C3] 1183: { code[0] = code[14];
[V93] 1184:   code[1] = code[15];
[YN2] 1185:   code[3][21] = 25;
[CP2] 1186:   code[3][44] = 26;
[U70] 1187: }
[000] 1188:
[684] 1189: void deutsche_Tastatur()
[EC3] 1190: { code[0] = code[16];
[Z93] 1191:   code[1] = code[17];
[ZN2] 1192:   code[3][21] = 26;
[BP2] 1193:   code[3][44] = 25;
[U70] 1194: }
```

Listing: »skeybgr.c« ersetzt den normalen »keyb«

Turbo Pascal

Eingaben wie von Geisterhand

Es ist mitunter recht lästig, immer wiederkehrende Eingaben erneut einzutippen. Viel angenehmer wäre es, die Daten einmal einzugeben und bei neuerlichem Bedarf zu übernehmen. Somit müßte nicht ständig jemand an der Tastatur sitzen und Eingaben machen, damit der Computer agiert.

»Prokey« ist in Turbo Pascal geschrieben – zur Simulation von Tastatureingaben auf IBM-kompatiblen PCs unter MS-DOS. Im Gegensatz zu anderen Utilities dieser Art ist es programmierbar, das heißt, in Abhängigkeit von der Bildschirmausgabe aufgerufener Programme können beliebige Eingabesequenzen vorprogrammiert werden. Dies geschieht in einer Sprache, die sich an die Pascal-Notation anlehnt, daher für Pascal-Programmierer leicht erlernbar ist. Zum Einstieg ein kurzes Beispiel zur Prokey-Programmierung:

```
clrscr
type('VER', chr(13))
if test('3.3')
write('Sie benutzen DOS 3.3')
endif
```

Starten Sie dieses Programm mit Hilfe des Prokey-Interpreters von der Betriebssystemebene aus, so geschieht folgendes: Zunächst wird der Bildschirm gelöscht. Dann simuliert Prokey die Eingabe des MS-DOS-Kommandos »ver [Enter]«. Dieses Kommando bewirkt, vorausgesetzt Sie benutzen die richtige MS-DOS-Version, folgende Bildschirmausgabe:

```
Sie benutzen DOS 3.3
```

In diesem Falle führt die nachfolgende Zeile, die mittels des Prokey-Kommandos »test« den Bildschirm auf das Vorhandensein des Strings »3.3« überprüft, zum Erfolg und zeigt über »write« die entsprechende Meldung an.

Die Anwendungsgebiete von Prokey sind vielfältig. Prokey-Programme können Sie einerseits analog zu Stapeldateien verwenden, es stellt Ihnen aber auch eine komfortablere Programmierschnittstelle zur Verfügung. Im Gegensatz zu MS-DOS-Stapelprogrammen erlauben Prokey-Programme es auch, Anwendungen aufzurufen und durch Simulation manueller Eingaben (auch Funktionstasten) zu bedienen, die nicht für Stapelbetrieb vorgesehen sind und sich daher nicht über Kommandozeilenparameter steuern lassen. Weiterhin hilft Prokey Ihnen auch dann weiter, wenn in Anwendungsprogrammen Makro- oder Alias-Funktionen fehlen. In der täglichen Praxis des Autors hat sich Prokey bislang insbesondere beim Herunterladen großer Datenmengen von Textdatenbanken in der Telekommunikation zum Aufbau störanfälliger Verbindungen, die ein oftmaliges Wiederholen bestimmter Einlogprozeduren erfordern, sowie zur Erzeugung von Demos für Anwenderprogramme bewährt. Das Programm »prokey.pas« (Listing 1) können Sie mit Turbo Pascal Version 4.0, 5.0 oder 5.5 kompilieren. Zu beachten ist, daß Sie das Programm nicht aus der Entwicklungsumgebung von Turbo Pascal starten. Sie müssen es auf Disk kompilieren und anschließend die Exe-Datei vom Betriebssystem aus hochfahren.

Prokey-Programme können Sie mit jedem Editor, der ASCII-Dateien erzeugt, eingeben, insbesondere auch mit dem Turbo-Pascal-Editor. Zur Ausführung des Prokey-Programms geben Sie

```
prokey dateiname
```

ein, wobei »dateiname« das von Prokey auszuführende Programm bezeichnet (Voreinstellung: »prokey.txt«). Dadurch wird das Programm in einer aufbereiteten und komprimierten Form (beispielsweise alle Kommandos als 1 Byte große Tokens) in den Speicher geladen. Prokey installiert sich resident und beginnt sofort mit der Programmausführung. Mit jedem Timer-Interrupt (etwa 18,2mal in der Sekunde) führt es genau ein Kommando aus. Werden Tastatureingaben simuliert (Type-Kommando), so erfolgt mit jedem Timer-Tick die Eingabe eines Zeichens in den BIOS-Tastaturpuffer, was einer realistischen Tippgeschwindigkeit entspricht.

Durch Betätigen der Taste [Druck] (bei XTs [Umschalt-Druck]), also derjenigen Taste, die Sie normalerweise zur Erzeugung eines Bildschirmausdrucks verwenden, halten Sie die Programmausführung an und setzen sie durch nochmaliges Betätigen von [Druck] wieder fort. Intermediär ist normaler interaktiver Betrieb gegeben. Nach Beendigung des Programms bewirkt [Druck] einen Neustart. Prokey, das resident im Speicher liegt, gibt bei erneutem Aufruf (den es am Vorhandensein der Datei »provect.dat« erkennt) den belegten Speicherplatz frei, und dieser steht MS-DOS wieder zur Verfügung. Mehrfach residentes Installieren läßt sich so vermeiden.

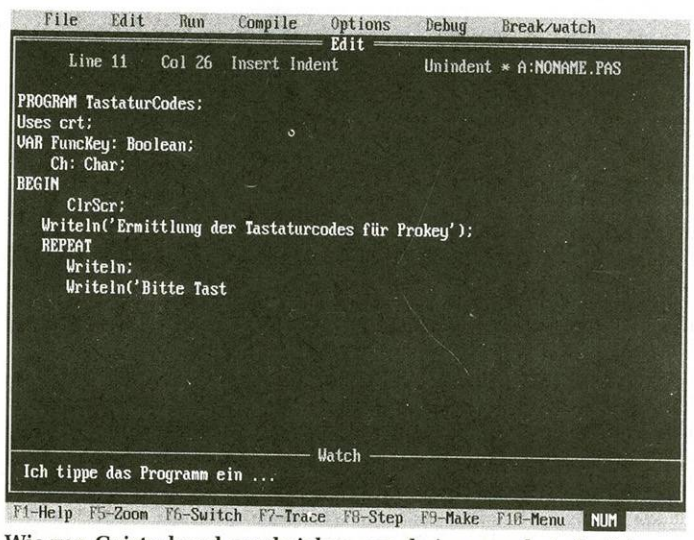
Die Kommandos liegen als Tokens vor

Die Datei »demo« (Listing 2) zeigt Ihnen die Arbeitsweise von Prokey. Diese muß sich mit »prokey.exe« im Pascal-Hauptverzeichnis befinden. Nach dem Start mit

```
prokey demo
```

wird automatisch Turbo Pascal geladen, das Programm »tastcode.pas« eingetippt (Bild), kompiliert und gestartet. Mit [Esc] brechen Sie es ab.

Nach diesem Überblick folgt eine Auflistung des für die Prokey-Programmierung vorgesehenen Befehlsumfangs. In den angegebenen Beispielen, bei denen es sich durchweg um vollständige, separat ablauffähige Programme handelt, finden ausschließlich MS-DOS-Kommandos Verwendung. Für reale Anwendungen ist es wesentlich interessanter,



Wie von Geisterhand geschrieben, erscheinen nach Aufruf des Demo-Programms die Buchstaben im Turbo-Pascal-Editor

daß Sie fast beliebige Anwendungsprogramme aufrufen können. Zur Notation: In spitze Klammern (»< >«) eingeschlossene Begriffe sind separat erläutert.

```
type <Zeichenfolge>
```

schreibt eine Folge von Zeichen in den Tastaturpuffer und simuliert damit für alle Programme, die diesen verwenden, eine Tastatureingabe (vergleichen Sie mit dem Chr-Kommando für die Eingabe von Funktionstasten und Spezialzeichen).
Zum Beispiel:

```
type ('dir',chr(13))
```

bewirkt auf MS-DOS-Ebene die Ausgabe des Directorys. »chr(13)« entspricht hierbei der Eingabetaste.

```
write <Zeichenfolge>
```

schreibt die spezifizierte Folge von Zeichen, beginnend bei der oberen linken Ecke, auf den Bildschirm. Nach Erreichen der rechten unteren Ecke wird wiederum oben links fortgefahren. Mit »gotoxy« können Sie die Startkoordinaten für die Bildschirmausgabe angeben. Da die Routine die Zeichen direkt in den Bildschirmspeicher schreibt, werden einzelnen Zeichen zugeordnete Funktionen (zum Beispiel »cr« und »lf«) nicht ausgeführt. Beispiel:

```
write('ABC')
```

schreibt den String »ABC« auf den Bildschirm.

```
gotoxy(<Integer>,<Integer>)
```

positioniert die Ausgabe einer nachfolgenden Write-Operation. Bei den anzugebenden Koordinaten handelt es sich um Spalte und Zeile auf dem Bildschirm, wobei »Spalte« Werte von 1 bis 80 und »Zeile« Werte von 1 bis 25 annehmen kann.

```
gotoxy(40,1)
write('ABC')
```

gibt »ABC« in Spalte 40 von Zeile 1 aus.

```
test(<Zeichenfolge>)
```

liefert einen Wahrheitswert, der davon abhängt, ob »Zeichenfolge« in einem durch das Window-Kommando definierten Bereich des Bildschirms aufgefunden werden kann oder nicht. Je kleiner dieser Bereich ist, um so schneller die Abfrage und um so weniger wird ein gerade laufendes Programm verzögert. »test« ist in der Regel nicht als eigenständiger Befehl, sondern nur in Verbindung mit If-Endif und Repeat-Until sinnvoll einzusetzen.

```
lasttest
```

ergibt den Wert des zuletzt ausgeführten Test-Kommandos, ohne diesen Test zu wiederholen (Zeitersparnis).

```
test('ABC')
if lasttest write('Suchstring gefunden') endif
```

dient dazu, falls »ABC« auf dem Bildschirm gefunden wurde, eine entsprechende Meldung auszugeben.

```
clrscr
```

löscht den Bildschirm, ohne die Cursorposition zu verändern. Dieses Kommando empfiehlt sich immer dann vor Testoperationen, wenn der vorherige Inhalt des Bildschirms nicht genau bekannt ist.

```
Window(<Integer>,<Integer>,<Integer>,<Integer>)
```

Das Window-Kommando begrenzt den Bereich des Bildschirms, den »test« auf das Vorhandensein eines Strings überprüft. Den vier Parametern entsprechen hierbei die Koordinaten zweier Punkte auf dem Textbildschirm in folgender Reihenfolge: »Spalte1«, »Zeile1«, »Spalte2«, »Zeile2«. Die Werte für die Spalten bewegen sich hierbei im Bereich von 1 bis 80, diejenigen für die Zeilen zwischen 1 und 25. Zu beachten ist, daß diese Koordinaten in der Regel nicht die Eckpunkte eines Rechtecks spezifizieren, sondern daß das Kommando dazwischenliegende Zeilen vollständig erfaßt (entsprechend der Blockmarkierungsfunktion im Turbo-Pascal-Editor).

```
gotoxy(1,1)
write('ABC')
window(1,3,80,1)
if test('ABC')
write('ABC in Zeile 1 gefunden')
endif
```

Dieses Programm schreibt in Spalte 1 von Zeile 1 den String »ABC«. Der folgende Test bleibt ohne Erfolg, da der Text nicht in dem durch »window« bestimmten Bereich liegt, der Text »ABC in Zeile 1 gefunden« wird also nicht ausgegeben.

```
wait(<Integer>)
```

wartet eine in Sekunden angegebene Zeit, die sich aus der Multiplikation des Parameters mit 0.0553562 ergibt. Der Kehrwert dieses Multiplikators ergibt die Frequenz des Timer-Interrupts, die zirka 18,2 Hertz beträgt.

```
wait(182)
write('Ende der Wartezeit')
```

Erst nach ungefähr 10 Sekunden Wartezeit erfolgt die Ausführung des Write-Kommandos.

```
pause
```

Mit der Ausführung des nächsten Kommandos wird bis zur Betätigung der Taste [Druck] (auf manchen Tastaturen [Umschalt-Druck]) gewartet. Dieses Kommando gestattet es, die Tastatursimulation vorübergehend abzuschalten und in den normalen, interaktiven Modus zu wechseln.

```
pause
write('ABC')
```

Nach dem Starten von Prokey wird die Programmausführung unterbrochen, bis Sie die Taste [Druck] betätigen.

```
if <Bedingung> <Kommandos> endif
```

bedeutet: Ist die Bedingung erfüllt, führt Prokey die Kommandos aus.

```
if test('ABC') write('Directory: ')
type('dir',chr(13))
endif
```

Falls auf dem Bildschirm der String »ABC« vorhanden ist, erfolgt die Ausgabe des Wortes »Directory« sowie die Ausführung des Dir-Kommandos.

```
repeat <Kommandos> until <Bedingung>
```

Prokey wiederholt die Kommandos so lange, bis die Bedingung erfüllt ist.

```
repeat until test('ABC')
```

initiiert Warten, bis der String »ABC« auf dem Bildschirm erscheint.

```
for <Integer> <Kommandos> endfor
```

wiederholt die Kommandos innerhalb der Schleife Integer-mal.

```
for 5
write('Schleife ')
endfor
```

schreibt das Wort »Schleife « fünfmal auf den Bildschirm.

```
exit
```

bewirkt sofortiges Verlassen der aktiven Repeat-Until- oder For-Endfor-Schleife.
Zum Beispiel:

```
for 100
write('Loop ')
if test('ABC') exit
endif
endfor
```

führt die For-Schleife so lange aus, bis die Zeichenkette »ABC« auf dem Bildschirm erscheint, maximal jedoch 100mal.

`chr(<Integer>)`

ist nur innerhalb von Zeichenfolgen erlaubt und wandelt ganze Zahlen im Bereich von 1 bis 254 in die zugehörigen ASCII-Zeichen um. Wollen Sie mit Hilfe des Type-Kommandos Eingaben von Tasten, die den Extended-ASCII-Code verwenden, simulieren, so ist mittels zweier Chr-Kommandos der entsprechende Doppelcode anzugeben. Welchen Tasten welche Codes zugeordnet sind, ermitteln Sie mit Hilfe des kleinen Turbo-Pascal-Programms »tastcode.pas« (Listing 3).
Nur ganz wenige Tastenkombinationen können Sie mittels Prokey nicht simulieren. Diese sind: [Strg-Untbr], [Alt-Strg-Entf], [Druck], [Strg-Num] sowie [SysAnfr] (nur auf ATs vorhanden). Das BIOS behandelt diese Tastenkombinationen direkt, sie laufen nicht über den von Prokey verwendeten Tastaturpuffer.
Zum Beispiel:

```
type('A',chr(66),chr(67))
type(chr(0), chr(59))
```

simuliert die Tastendrücke [A], [B], [C] und [F1].

`time`

ist nur innerhalb von Zeichenfolgen erlaubt und liefert Datum und Zeit beim Aufruf von Prokey im Format »Tag Monat Stunde Minute«. Dieses etwas ungewöhnliche Format erlaubt die Generierung eindeutiger Dateinamen.
Zum Beispiel:

```
type('dir >',time,'.txt')
```

Angenommen, Sie rufen Prokey am 7. November um 8.15 Uhr auf, dann wird das aktuelle Directory in eine Datei mit Namen »07110815.txt« kopiert.

`{beliebiger Kommentar}`

überliest der Interpreter, da er den in die geschweiften Klammern eingeschlossenen Text als Kommentar betrachtet. Das Verschachteln von Kommentaren ist nicht gestattet.

`<Kommandos>`

ist eine Folge beliebig vieler der oben beschriebenen Befehle (auch Null), getrennt durch Leerzeichen oder Strichpunkt. Groß-/Kleinschreibung wird bei Kommandos ignoriert. Repeat-Until, For-Endfor und If-Endif können Sie bis zu 50mal verschachteln.

`<Bedingung>`

Eine Bedingung hat eine der folgenden Formen:
- TRUE: immer wahr
- FALSE: immer falsch
- test(Zeichenfolge): optional, »Zeichenfolge« wird in einem durch das Window-Kommando definierten Bereich des Bildschirmes gefunden oder nicht
- lasttest: der Wert des zuletzt ausgeführten Test-Kommandos gilt

Jeder dieser vier Bedingungen können Sie ein »not« voranstellen.

`<Integer>`

ist eine ganze Zahl im Bereich 1 bis 32767.

`<Zeichenfolge>`

ist eine beliebige Aneinanderreihung von in Apostroph oder Anführungszeichen eingeschlossenen Zeichen, Chr- und Time-Kommandos.

```
'A',chr(66),'CDEF'
```

wird interpretiert als »ABCDEF«, da 66 dem ASCII-Wert von B entspricht. Soll die Zeichenfolge das Apostroph »'« enthalten, können Sie als Begrenzer auch »"« verwenden. Ferner ist die Systemvariable »time« erlaubt.
Die folgende Kurzbeschreibung gibt einen Überblick über die grundsätzliche Arbeitsweise von Prokey. Das Programm hat gegenüber anderen Interrupt-Programmen den Vorteil, daß es ausschließlich in Pascal programmiert ist und auf mitunter kryptische Inline-Passagen (eingebauten Maschinencode) weitgehendst verzichtet.

Nach dem Starten liest Prokey zunächst die in der Kommandozeile spezifizierte Programmdatei ein und wandelt alle Schlüsselwörter in 1-Byte-Tokens um. Das komprimierte Programm legt Prokey im Character-Array »Prog« ab.

Prokey ist 18mal pro Sekunde aktiv

Danach erfolgt die Umlegung des Timer-Interrupt-Vektors auf die Interrupt-Prozedur »IntProc«, die in Folge bei jedem Timer-Interrupt also etwa 18mal in der Sekunde zum Aufruf kommt. Entsprechend wird Interrupt 5, der normalerweise für den Ausdruck einer Hardcopy des Bildschirms zuständig ist und mit der Taste [Druck] zu aktivieren ist, auf die Prozedur »PrIntProc« gelegt. Die alten Werte dieser Vektoren sichert Prokey in eine Datei. Danach installiert sich Prokey durch Aufruf einer MS-DOS-Routine resident im Speicher.

In jeder Interrupt-Routine arbeitet Prokey einen Befehl ab

Während nun der Benutzer beliebige DOS-Kommandos absetzen kann, aktiviert jeder Timer-Tick die Interrupt-Routine. Diese arbeitet normalerweise mit jedem Tick genau einen Befehl des im Array »Prog« gespeicherten Programms ab. Tokens werden in der Routine »CommandInterpreter« entschlüsselt und die zugehörigen Kommandos gemäß der Sprachdefinition ausgeführt. Bei der Programmierung aller Routinen, die »IntProc« direkt oder indirekt aufruft, dürfen keine Turbo-Pascal-Befehle verwendet werden, die auf MS-DOS-Routinen zurückgreifen. Das ist insbesondere bei allen Ein-/Ausgabe-Kommandos der Fall. Dies hängt damit zusammen, daß MS-DOS nicht reentrant ist und es bei gleichzeitiger Verwendung einer MS-DOS-Funktion durch die Interrupt-Routine und durch das unterbrochene Programm zu Komplikationen kommen kann, die in der Regel zu einem Systemabsturz führen. Für die Bildschirmausgabe müssen deshalb eigene Routinen, die direkt auf den Bildschirmspeicher zugreifen, geschrieben werden. Programmänderungen sind also im Hinblick darauf mit Bedacht vorzunehmen. Wenn Prokey bei einem erneuten Aufruf feststellt, daß es sich bereits im Speicher befindet, so entfernt es sich zunächst mittels der Routine »DisableProkey«. Hierbei werden auch alle Interruptvektoren wieder auf die ursprünglichen Werte zurückgesetzt.

Der Tastaturpuffer ist die Schnittstelle zu anderen Programmen

Bezüglich der Anwendbarkeit von Prokey in Verbindung mit anderen Programmen gibt es ein paar Einschränkungen. Prokey läuft nur mit Programmen, die den normalen Tastaturpuffer verwenden, was fast immer der Fall ist. Obwohl es die Art des im Rechner verwendeten Bildschirms (monochrom, CGA, EGA, VGA) erkennt, funktionieren die bildschirmorientierten Befehle »clrscr« und insbesondere »test« nur im Textmodus. Deshalb müssen Sie auf die Verwendung der Bildschirmabfrage immer dann verzichten, wenn Anwendungsprogramme im Grafikmodus arbeiten.
Eine weitere Einschränkung betrifft die maximale Länge eines Prokey-Programms. Es gilt, daß die komprimierte Version des Programms, die für jeden Befehl nur ein Byte benötigt, nicht länger als 20000 Byte sein darf. Diesen Wert legt die Konstante »SourceLength« im Quellcode von Prokey fest, er kann Werte von 1 bis zirka 64000 Byte annehmen. Nach oben hin stellt die maximale Größe des Datensegmentes (64 KByte einschließlich sonstiger globaler Variablen) unter Turbo Pascal eine Schranke dar. Ein kleiner Wert hat hingegen den Vorteil, daß das residente Programm weniger

Speicherplatz benötigt. Der Speicherplatzbedarf ergibt sich hierbei aus der Summe der Codelänge (zirka 20 KByte) und des Wertes von »SourceLength.« Sie können sich auch verschiedene Versionen des Programms (mit einem passenden Wert von »SourceLength«) anlegen und in einer Autoexec-Datei über Abfragen die gerade benötigte Version starten. Wenn Sie besser mit dem Programmaufbau vertraut sind, können Sie auch nicht benötigte Funktionen entfernen und sich somit ein auf Ihre Bedürfnisse zugeschnittenes Prokey entwerfen. (Reinhard Rapp/Helmut Stult/rm)

```
[2X3]    1: { Programm: prokey.pas
[7N4]    2:   Funktion: programmierbarer
[R63]    3:             Tastatursimulator
[B65]    4:   Sprache:  Turbo Pascal 4.Ø bis 5.5
[SB6]    5:   Autor:    Dipl.-Inf. wiss. Reinhard Rapp
[P35]    6:   (c)1991 DMV Widuch GmbH & Co. KG }
[ØØØ]    7:
[G41]    8: {$S+ }
[C41]    9: {$R+ }
[T12]   1Ø: {$M 4Ø96, Ø, Ø }
[ØØØ]   11:
[5T4]   12: PROGRAM keyboardsimulator;
[ØØØ]   13:
[ASØ]   14: USES
[ØØØ]   15:
[WRØ]   16:    Dos;
[ØØØ]   17:
[6XØ]   18: CONST
[ØØØ]   19:
[RW2]   2Ø:   tastbuf       = $ØØ4Ø;
[DZ2]   21:   bufstart      = $ØØ1E;
[213]   22:   startptr      = $ØØ1A;
[6S2]   23:   endptr        = $ØØ1C;
[ND3]   24:   sourcelength  = 2ØØØØ;
[WØ5]   25:   defaultinfile = 'PROKEY.TXT';
[UN4]   26:   vectorfile    = 'PROVECT.DAT';
[EW3]   27:   stackdepth    = 5Ø; { max.
[RU2]   28:                         Verschachtelung
[ØP2]   29:                         fuer Schleifen }
[ØØØ]   3Ø:
[HM2]   31:   tokenif       = 'A';
[GW2]   32:   tokenendif    = 'B';
[V23]   33:   tokenrepeat   = 'C';
[8Z2]   34:   tokenuntil    = 'D';
[AR2]   35:   tokenfor      = 'E';
[T33]   36:   tokenendfor   = 'F';
[AW2]   37:   tokentype     = 'G';
[YU2]   38:   tokentest     = 'H';
[TV2]   39:   tokentrue     = 'I';
[GW2]   4Ø:   tokenfalse    = 'J';
[463]   41:   tokenwindow   = 'K';
[E73]   42:   tokengotoxy   = 'L';
[FØ3]   43:   tokenwrite    = 'M';
[ZS2]   44:   tokennot      = 'N';
[P83]   45:   tokenlasttest = 'O';
[BW2]   46:   tokenexit     = 'P';
[8V2]   47:   tokenwait     = 'Q';
[L43]   48:   tokenclrscr   = 'R';
[KY2]   49:   tokenpause    = 'S';
[ØØØ]   5Ø:
[STØ]   51: TYPE
[824]   52:   wrkstring = STRING [15Ø];
[T44]   53:   lstring   = STRING[255];
[B33]   54:   stacktype = RECORD
[VR2]   55:                 token          : CHAR;
[2R3]   56:                 position       : LONGINT;
[ØB4]   57:                 forloopcounter : INTEGER;
[ZNØ]   58:               END;
[ØØØ]   59:
[Y13]   6Ø:   inttochar = RECORD
[BM2]   61:                 CASE INTEGER OF
[1K3]   62:                   1: (intrepr: INTEGER);
[3B3]   63:                   2: (char1, char2: CHAR);
[ZNØ]   64:               END;
[RNØ]   65: VAR
[ØØØ]   66:
[F12]   67:   proglength,
[GP3]   68:   progptr       : LONGINT;
[KT3]   69:   progtoggle    : BOOLEAN;
[6Ø6]   7Ø:   prog          : ARRAY[1..sourcelength]
[981]   71:                       OF CHAR;
[ØØØ]   72:
[PR3]   73:   regs          : REGISTERS;
[LT3]   74:   screenstartadd : WORD;
[DF5]   75:   stack         : ARRAY[1..stackdepth]
[L52]   76:                       OF stacktype;
[ØØØ]   77:
[E14]   78:   stackpointer  : INTEGER;
[1S3]   79:   currentcommand : CHAR;
[974]   8Ø:   teststatusflag : BOOLEAN;
[FW2]   81:   writexposition,
[KW2]   82:   writeyposition,
[8D2]   83:   testwindowx1,
[CD2]   84:   testwindowy1,
[9D2]   85:   testwindowx2,
[DD2]   86:   testwindowy2,
[SØ4]   87:   tickstowait   : INTEGER;
[XH3]   88:   success       : BOOLEAN;
[6F3]   89:   disable       : BOOLEAN;
[UG3]   9Ø:   strdelimiter  : CHAR;
```

```
[ØØØ]   91:
[W54]   92: PROCEDURE makeresident;
[ØØØ]   93:
[RNØ]   94: VAR
[MZ2]   95:   codelength : WORD;
[ØØØ]   96:
[MUØ]   97: BEGIN
[Ø25]   98:   Write('Prokey made resident ');
[V45]   99:   codelength:= Seg(freeptr^)+$1ØØØ
[GZ1]  1ØØ:                 - prefixseg;
[RØ2]  1Ø1:   WITH regs DO
[MUØ]  1Ø2:   BEGIN
[Y41]  1Ø3:     ah:= $31;
[721]  1Ø4:     al:= Ø;
[LR2]  1Ø5:     dx:= codelength;
[3Ø2]  1Ø6:     MsDos(regs);
[ZNØ]  1Ø7:   END;
[ZNØ]  1Ø8: END;
[ØØØ]  1Ø9:
[LT6]  11Ø: PROCEDURE writedirectchar(zeich: CHAR);
[ØØØ]  111:
[RNØ]  112: VAR
[YS2]  113:   actpos : INTEGER;
[ØØØ]  114:
[MUØ]  115: BEGIN
[DF5]  116:   actpos:= (writeyposition - 1) * 16Ø
[AØ4]  117:            + (writexposition - 1) * 2;
[K17]  118:   Mem[screenstartadd:actpos]:= Ord(zeich);
[WC6]  119:   writexposition:= writexposition + 1;
[YF4]  12Ø:   IF writexposition > 8Ø THEN
[MUØ]  121:   BEGIN
[3E3]  122:     writexposition:= 1;
[3D6]  123:     writeyposition:= writeyposition + 1;
[3G4]  124:     IF writeyposition > 25 THEN
[MUØ]  125:     BEGIN
[3E3]  126:       writexposition:= 1;
[6E3]  127:       writeyposition:= 1;
[ZNØ]  128:     END;
[ZNØ]  129:   END;
[ZNØ]  13Ø: END;
[ØØØ]  131:
[9U6]  132: PROCEDURE writedirectstr(st: lstring);
[ØØØ]  133:
[RNØ]  134: VAR
[BZ1]  135:   i : INTEGER;
[ØØØ]  136:
[MUØ]  137: BEGIN
[964]  138:   FOR i:= 1 TO Length(st) DO
[984]  139:   writedirectchar(st[i]);
[ZNØ]  14Ø: END;
[ØØØ]  141:
[YZ6]  142: PROCEDURE writedirectcharxy(x: INTEGER;
[DY3]  143:                 y: INTEGER; zeich: CHAR);
[ØØØ]  144:
[RNØ]  145: VAR
[VV2]  146:   actpos: INTEGER;
[ØØØ]  147:
[MUØ]  148: BEGIN
[GF4]  149:   IF (x > Ø) AND (x < 81) AND (y > Ø)
[7E2]  15Ø:     AND (y < 26) THEN
[MUØ]  151:   BEGIN
[5L4]  152:     actpos:= (y - 1) * 16Ø + (x - 1) * 2;
[145]  153:     Mem[screenstartadd:actpos]:=
[DX1]  154:                               Ord(zeich);
[4MØ]  155:   END
[ZRØ]  156:   ELSE
[ØC6]  157:     writedirectstr('Screen-coordinates ' +
[LK2]  158:                    'out of range ');
[ZNØ]  159: END;
[ØØØ]  16Ø:
[XU3]  161: PROCEDURE clearscreen;
[ØØØ]  162:
[RNØ]  163: VAR
[8V1]  164:   i: INTEGER;
[ØØØ]  165:
[MUØ]  166: BEGIN
[NX2]  167:   FOR i:= Ø TO 1999 DO
[MUØ]  168:   BEGIN
[AX4]  169:     Mem[screenstartadd:2*i]:= 32;
[ZNØ]  17Ø:   END;
[ZNØ]  171: END;
[ØØØ]  172:
[KU6]  173: PROCEDURE writekbdbuffer (func: BOOLEAN;
[XL1]  174:                           ch   : CHAR);
[ØØØ]  175:
[MUØ]  176: BEGIN
[CY1]  177:   IF func THEN
[MUØ]  178:   BEGIN
[PL4]  179:     Mem[tastbuf:bufstart]   := Ø;
[3T5]  18Ø:     Mem[tastbuf:bufstart+1]:= Ord(ch);
```

Listing 1. »prokey.pas«

```
[4MØ]  181:    END
[ZRØ]  182:    ELSE
[KK5]  183:      Mem[tastbuf:bufstart]:= Ord(ch);
[886]  184:    MemW[tastbuf:startptr]:= bufstart;
[BA6]  185:    MemW[tastbuf:endptr]  := bufstart + 2;
[ZNØ]  186: END;
[ØØØ]  187:
[YU6]  188: FUNCTION checkscreen(datastartpointer:
[823]  189:                          LONGINT): BOOLEAN;
[ØØØ]  19Ø:
[RNØ]  191: VAR
[AD3]  192:    cursor        : INTEGER;
[582]  193:    ch            : CHAR;
[H23]  194:    ende          : BOOLEAN;
[L34]  195:    currentstpos  : INTEGER;
[3Ø4]  196:    startcharpos  : INTEGER;
[ZR3]  197:    endcharpos    : INTEGER;
[ØØØ]  198:
[MUØ]  199: BEGIN
[H56]  2ØØ:    startcharpos:= (testwindowy1 - 1) * 8Ø +
[U23]  2Ø1:                   (testwindowx1 - 1);
[EX5]  2Ø2:    endcharpos  := (testwindowy2 - 1) * 8Ø +
[W23]  2Ø3:                   (testwindowx2 - 1);
[HD3]  2Ø4:    checkscreen := FALSE;
[7N2]  2Ø5:    ende        := FALSE;
[1B4]  2Ø6:    cursor      := startcharpos;
[P31]  2Ø7:    REPEAT
[6T4]  2Ø8:      ch:= Chr(Mem[screenstartadd:
[722]  2Ø9:               cursor * 2]);
[G46]  21Ø:      IF ch = prog[datastartpointer] THEN
[MUØ]  211:        BEGIN
[EX2]  212:          currentstpos:= Ø;
[P31]  213:          REPEAT
[YK5]  214:            currentstpos:= currentstpos + 1;
[6T4]  215:            ch:= Chr(Mem[screenstartadd:
[5U4]  216:               (cursor + currentstpos) * 2]);
[B94]  217:            IF prog[datastartpointer
[U95]  218:               + currentstpos] = strdelimiter
[2RØ]  219:              THEN
[MUØ]  22Ø:            BEGIN
[4K2]  221:              ende       := TRUE;
[EA3]  222:              checkscreen:= TRUE;
[ZNØ]  223:            END;
[LF3]  224:          UNTIL (ende = TRUE) OR
[KU4]  225:                (ch <> prog[datastartpointer
[HV4]  226:                 + currentstpos]) OR ((cursor
[KK3]  227:                 + currentstpos) > 2ØØØ);
[ZNØ]  228:        END;
[S93]  229:      cursor:= cursor + 1;
[FN4]  23Ø:      IF cursor >=  endcharpos THEN
[UX1]  231:      ende:= TRUE;
[FX2]  232:    UNTIL ende = TRUE;
[ZNØ]  233: END;
[ØØØ]  234:
[W15]  235: PROCEDURE push(token       : CHAR;
[ØM3]  236:                position    :  LONGINT;
[E14]  237:                loopcounter : INTEGER);
[ØØØ]  238:
[MUØ]  239: BEGIN
[5A5]  24Ø:    stackpointer:= stackpointer + 1;
[XK5]  241:    IF stackpointer > stackdepth THEN
[LU6]  242:      writedirectstr('ERROR Stack Overflow');
[P86]  243:    stack[stackpointer].token    := token;
[9A7]  244:    stack[stackpointer].position:= position;
[7S6]  245:    stack[stackpointer].forloopcounter:=
[Ø92]  246:                                  loopcounter;
[ZNØ]  247: END;
[ØØØ]  248:
[2L5]  249: PROCEDURE pop(VAR token       : CHAR;
[T34]  25Ø:               VAR position    : LONGINT;
[5L4]  251:               VAR loopcounter: INTEGER);
[ØØØ]  252:
[MUØ]  253: BEGIN
[766]  254:    token    := stack[stackpointer].token;
[U77]  255:    position:= stack[stackpointer].position;
[TC2]  256:    loopcounter:=
[EP6]  257:          stack[stackpointer].forloopcounter;
[9A5]  258:    stackpointer:= stackpointer - 1;
[HV3]  259:    IF stackpointer < Ø THEN
[435]  26Ø:      writedirectstr('ERROR Stack '+
[B42]  261:                     'Underflow');
[ZNØ]  262: END;
[ØØØ]  263:
[SF3]  264: PROCEDURE maketest;
[ØØØ]  265:
[RNØ]  266: VAR
[AD3]  267:    notstatus: BOOLEAN;
[ØØØ]  268:
[MUØ]  269: BEGIN
[H53]  27Ø:    notstatus:= FALSE;
[GT5]  271:    WHILE prog[progptr] = tokennot DO
[MUØ]  272:    BEGIN
[GT4]  273:      notstatus:= NOT notstatus;
[HZ3]  274:      progptr  := progptr + 1;
[ZNØ]  275:    END;
[FR3]  276:    CASE prog[progptr] OF
[356]  277:      tokentrue    : teststatusflag:= TRUE;
[R86]  278:      tokenfalse   : teststatusflag:= FALSE;
[AZ6]  279:      tokenlasttest: {Leave TestStatusFlag as
[F91]  28Ø:                     it is} ;
[E13]  281:      tokentest    : BEGIN
[DM3]  282:                       progptr:= progptr + 1;
[MW2]  283:                       teststatusflag:=
[RR3]  284:                        checkscreen(progptr);
[GN3]  285:                       WHILE prog[progptr]
[W13]  286:                         <> strdelimiter DO
[HN1]  287:                         progptr:=
[KY1]  288:                                 progptr + 1;
[ZNØ]  289:                     END;
[ZNØ]  29Ø:    END;
[EY2]  291:    IF notstatus THEN
[EG6]  292:      teststatusflag:= NOT teststatusflag;
[H74]  293:    progptr       := progptr + 1;
[LU5]  294:    currentcommand:= prog[progptr];
[ZNØ]  295: END;
[ØØØ]  296:
[YF4]  297: PROCEDURE skipexpression;
[ØØØ]  298:
[MUØ]  299: BEGIN
[GT5]  3ØØ:    WHILE prog[progptr] = tokennot DO
[MUØ]  3Ø1:    BEGIN
[DM3]  3Ø2:      progptr:= progptr + 1;
[ZNØ]  3Ø3:    END;
[PM5]  3Ø4:    IF prog[progptr] = tokentest THEN
[MUØ]  3Ø5:    BEGIN
[DM3]  3Ø6:      progptr:= progptr + 1;
[CS6]  3Ø7:      WHILE prog[progptr] <> strdelimiter DO
[DM3]  3Ø8:        progptr:= progptr + 1;
[ZNØ]  3Ø9:    END;
[H74]  31Ø:    progptr       := progptr + 1;
[LU5]  311:    currentcommand:= prog[progptr];
[ZNØ]  312: END;
[ØØØ]  313:
[DE7]  314: FUNCTION getinteger(VAR position: LONGINT)
[LL1]  315:                                    : INTEGER;
[ØØØ]  316:
[RNØ]  317: VAR
[DG3]  318:    transfer: inttochar;
[ØØØ]  319:
[MUØ]  32Ø: BEGIN
[FE4]  321:    position      := position + 1;
[PR5]  322:    transfer.char1:= prog[position];
[FE4]  323:    position      := position + 1;
[TR5]  324:    transfer.char2:= prog[position];
[WL5]  325:    getinteger    := transfer.intrepr;
[ZNØ]  326: END;
[ØØØ]  327:
[2Z5]  328: PROCEDURE moveforward(searchtoken,
[8Z2]  329:                       complement : CHAR;
[1T3]  33Ø:                       VAR success: BOOLEAN);
[RNØ]  331: VAR
[H23]  332:    ende         : BOOLEAN;
[2L3]  333:    instring     : BOOLEAN;
[HZ3]  334:    nestinglevel : INTEGER;
[ØØØ]  335:
[MUØ]  336: BEGIN
[7N2]  337:    ende         := FALSE;
[S53]  338:    instring     := FALSE;
[143]  339:    success      := FALSE;
[CT2]  34Ø:    nestinglevel:= Ø;
[P31]  341:    REPEAT
[YR2]  342:      IF instring THEN
[MUØ]  343:      BEGIN
[486]  344:        IF prog[progptr] = strdelimiter THEN
[SV2]  345:          instring:= FALSE;
[4MØ]  346:      END
[ZRØ]  347:      ELSE
[MUØ]  348:      BEGIN
[FR3]  349:        CASE prog[progptr] OF
[XY4]  35Ø:          tokentype:   instring:= TRUE;
[EX4]  351:          tokentest:   instring:= TRUE;
[A35]  352:          tokenwrite:  instring:= TRUE;
[3L5]  353:          tokenfor:    progptr:= progptr + 3;
[RP5]  354:          tokenwait:   progptr:= progptr + 3;
[Y16]  355:          tokengotoxy: progptr:= progptr + 5;
[316]  356:          tokenwindow: progptr:= progptr + 9;
[ZNØ]  357:        END;
[566]  358:        IF (prog[progptr] = complement) THEN
[MUØ]  359:        BEGIN
[PA5]  36Ø:          nestinglevel:= nestinglevel + 1;
[ZNØ]  361:        END;
[VB6]  362:        IF (prog[progptr] = searchtoken) THEN
[MUØ]  363:        BEGIN
[VW3]  364:          IF nestinglevel = Ø THEN
[MUØ]  365:          BEGIN
[482]  366:            ende    := TRUE;
[YP2]  367:            success:= TRUE;
[4MØ]  368:          END
[ZRØ]  369:          ELSE
[MUØ]  37Ø:          BEGIN
[UA5]  371:            nestinglevel:= nestinglevel - 1;
[TW3]  372:            IF nestinglevel < Ø THEN
[G34]  373:              writedirectstr('ERROR ' +
[R22]  374:                             'Erroneous' +
[8V1]  375:                             ' nesting ');
[ZNØ]  376:          END;
[ZNØ]  377:        END;
```

```
[ZNØ]  378:      END;
[DM3]  379:      progptr:= progptr + 1;
[MP4]  38Ø:      IF progptr > proglength THEN
[UX1]  381:        ende:= TRUE;
[AY1]  382:    UNTIL ende;
[5G2]  383:    IF success THEN
[MUØ]  384:    BEGIN
[R74]  385:      progptr       := progptr - 1;
[LU5]  386:      currentcommand:= prog[progptr];
[ZNØ]  387:    END;
[ZNØ]  388: END;
[ØØØ]  389:
[C45]  39Ø: PROCEDURE commandinterpreter;
[ØØØ]  391:
[RNØ]  392: VAR
[VD2]  393:   token      : CHAR;
[2D3]  394:   position   : LONGINT;
[3T2]  395:   ende       : BOOLEAN;
[9K3]  396:   forcounter : INTEGER;
[PZ2]  397:   dummy      : INTEGER;
[743]  398:   intnum     : INTEGER;
[X73]  399:   success    : BOOLEAN;
[8Z2]  4ØØ:   complement : CHAR;
[ØØØ]  4Ø1:
[MUØ]  4Ø2: BEGIN
[UX1]  4Ø3:   ende:= TRUE;
[P31]  4Ø4:   REPEAT
[8W3]  4Ø5:     CASE currentcommand OF
[G53]  4Ø6:       tokenrepeat: BEGIN
[613]  4Ø7:                      push(tokenrepeat,
[SY1]  4Ø8:                           progptr, Ø);
[DM3]  4Ø9:                    progptr:= progptr + 1;
[DX2]  41Ø:                    currentcommand:=
[1R2]  411:                          prog[progptr];
[ZNØ]  412:                  END;
[843]  413:       tokenuntil:  BEGIN
[DM3]  414:                    progptr:= progptr + 1;
[VK1]  415:                    maketest;
[HU3]  416:                    IF teststatusflag THEN
[XE3]  417:                      pop(token, position,
[N11]  418:                          dummy)
[ZRØ]  419:                    ELSE
[MUØ]  42Ø:                    BEGIN
[XE3]  421:                      pop(token, position,
[191]  422:                          dummy);
[EH3]  423:                      progptr:= position;
[DX2]  424:                      currentcommand:=
[1R2]  425:                            prog[progptr];
[ZNØ]  426:                    END;
[ZNØ]  427:                  END;
[DZ2]  428:       tokenexit:   BEGIN
[XE3]  429:                    pop(token, position,
[191]  43Ø:                        dummy);
[C32]  431:                    CASE token OF
[L62]  432:                      tokenrepeat:
[M42]  433:                                 complement:=
[F42]  434:                                  tokenuntil;
[PL1]  435:                      tokenfor:
[M42]  436:                                 complement:=
[972]  437:                                 tokenendfor;
[ZRØ]  438:                    ELSE
[WS2]  439:                      writedirectstr(
[DF2]  44Ø:                          'ERROR Illegal'
[3M2]  441:                        + ' use of exit');
[ZNØ]  442:                    END;
[C72]  443:                    moveforward(
[533]  444:                        complement, token,
[2L1]  445:                                  success);
[D33]  446:                    IF NOT success THEN
[WS2]  447:                      writedirectstr(
[F62]  448:                          'ERROR End of '
[K33]  449:                        +' loop not found');
[DM3]  45Ø:                    progptr:= progptr + 1;
[DX2]  451:                    currentcommand:=
[1R2]  452:                          prog[progptr];
[VH3]  453:                    IF token = tokenrepeat
[LN3]  454:                      THEN skipexpression;
[ZNØ]  455:                  END;
[CR2]  456:       tokenif:     BEGIN
[DM3]  457:                    progptr:= progptr + 1;
[VK1]  458:                    maketest;
[HU3]  459:                    IF teststatusflag THEN
[MUØ]  46Ø:                    BEGIN
[DX2]  461:                      currentcommand:=
[1R2]  462:                            prog[progptr];
[4MØ]  463:                    END
[ZRØ]  464:                    ELSE
[MUØ]  465:                    BEGIN
[C72]  466:                      moveforward(
[2F3]  467:                      tokenendif, tokenif,
[2L1]  468:                                  success);
[D33]  469:                      IF NOT success THEN
[WS2]  47Ø:                        writedirectstr(
[R42]  471:                            'ERROR EndIf '
[2B2]  472:                          + ' not found');
[ZNØ]  473:                    END;
[ZNØ]  474:                  END;
[P13]  475:       tokenendif:  BEGIN
[1G3]  476:                     progptr:= progptr +1;
[DX2]  477:                     currentcommand:=
[1R2]  478:                           prog[progptr];
[ZNØ]  479:                  END;
[LU2]  48Ø:       tokenfor:    BEGIN
[K62]  481:                    forcounter:=
[LE3]  482:                      getinteger(progptr);
[AY3]  483:                    push(tokenfor,progptr,
[C42]  484:                         forcounter);
[DM3]  485:                    progptr:= progptr + 1;
[DX2]  486:                    currentcommand:=
[1R2]  487:                          prog[progptr];
[ZNØ]  488:                  END;
[563]  489:       tokenendfor: BEGIN
[XE3]  49Ø:                    pop(token, position,
[C42]  491:                        forcounter);
[K62]  492:                    forcounter:=
[TG2]  493:                          forcounter - 1;
[AH3]  494:                    IF forcounter > Ø THEN
[MUØ]  495:                    BEGIN
[CL3]  496:                      push(token,position,
[C42]  497:                           forcounter);
[EH3]  498:                      progptr:= position;
[ZNØ]  499:                    END;
[DM3]  5ØØ:                    progptr:= progptr + 1;
[DX2]  5Ø1:                    currentcommand:=
[1R2]  5Ø2:                          prog[progptr];
[ZNØ]  5Ø3:                  END;
[7Y2]  5Ø4:       tokenwait:   BEGIN
[KN3]  5Ø5:                    IF tickstowait= Ø THEN
[MUØ]  5Ø6:                    BEGIN
[AS2]  5Ø7:                      tickstowait:= 1 +
[LE3]  5Ø8:                      getinteger(progptr);
[DX2]  5Ø9:                      currentcommand:=
[DU1]  51Ø:                            tokenwait;
[ZNØ]  511:                    END;
[MN3]  512:                    IF tickstowait= 1 THEN
[MUØ]  513:                    BEGIN
[EE3]  514:                      progptr:= progptr+1;
[DX2]  515:                      currentcommand:=
[1R2]  516:                            prog[progptr];
[ZNØ]  517:                    END;
[ZNØ]  518:                  END;
[M43]  519:       tokenwrite:  BEGIN
[DM3]  52Ø:                    progptr:= progptr + 1;
[GN3]  521:                    WHILE prog[progptr]
[W13]  522:                        <> strdelimiter DO
[MUØ]  523:                      BEGIN
[HW2]  524:                        writedirectchar(
[RW2]  525:                          prog[progptr]);
[E23]  526:                        progptr:= progptr
[NDØ]  527:                                  + 1;
[ZNØ]  528:                      END;
[DM3]  529:                    progptr:= progptr + 1;
[DX2]  53Ø:                    currentcommand:=
[1R2]  531:                          prog[progptr];
[ZNØ]  532:                  END;
[893]  533:       tokengotoxy: BEGIN
[4Ø3]  534:                    writexposition:=
[LE3]  535:                      getinteger(progptr);
[7Ø3]  536:                    writeyposition:=
[LE3]  537:                      getinteger(progptr);
[DM3]  538:                    progptr:= progptr + 1;
[DX2]  539:                    currentcommand:=
[1R2]  54Ø:                          prog[progptr];
[ZNØ]  541:                  END;
[V53]  542:       tokenclrscr: BEGIN
[642]  543:                    clearscreen;
[DM3]  544:                    progptr:= progptr + 1;
[DX2]  545:                    currentcommand:=
[1R2]  546:                          prog[progptr];
[ZNØ]  547:                  END;
[183]  548:       tokenwindow: BEGIN
[VL2]  549:                    testwindowx1:=
[LE3]  55Ø:                      getinteger(progptr);
[ZL2]  551:                    testwindowy1:=
[LE3]  552:                      getinteger(progptr);
[WL2]  553:                    testwindowx2:=
[LE3]  554:                      getinteger(progptr);
[ØM2]  555:                    testwindowy2:=
[LE3]  556:                      getinteger(progptr);
[DM3]  557:                    progptr:= progptr + 1;
[DX2]  558:                    currentcommand:=
[1R2]  559:                          prog[progptr];
[ZNØ]  56Ø:                  END;
[523]  561:       tokenpause:  BEGIN
[ØB3]  562:                    progtoggle:= FALSE;
[UX1]  563:                    ende:= TRUE;
[DM3]  564:                    progptr:= progptr + 1;
[DX2]  565:                    currentcommand:=
[1R2]  566:                          prog[progptr];
[ZNØ]  567:                  END;
[SY2]  568:       tokentest:   BEGIN
[VK1]  569:                    maketest;
[ZNØ]  57Ø:                  END;
[7Ø3]  571:       tokentype:   BEGIN
[DX2]  572:                    currentcommand:=
[DW1]  573:                          tokentype;
[DM3]  574:                    progptr:= progptr + 1;
[ZNØ]  575:                  END;
```

Listing 1. »prokey.pas«

```
[ZRØ]  576:        ELSE
[MUØ]  577:        BEGIN
[BD5]  578:          writedirectstr('ERROR Unknown '
[VX2]  579:                         + 'command token: ');
[SR5]  58Ø:          writedirectchar(currentcommand);
[DC3]  581:          writedirectstr('    ');
[UX1]  582:          ende:= TRUE;
[ZNØ]  583:        END;
[ZNØ]  584:      END;
[YU5]  585:      IF currentcommand = tokentype THEN
[UX1]  586:        ende:= TRUE;
[MP4]  587:      IF progptr > proglength THEN
[UX1]  588:        ende:= TRUE;
[AY1]  589:    UNTIL ende;
[ZNØ]  59Ø: END;
[ØØØ]  591:
[L63]  592: PROCEDURE intproc;
[ØØØ]  593:
[8X1]  594: INTERRUPT;
[ØØØ]  595:
[RNØ]  596: VAR
[4M3]  597:   st              : wrkstring;
[964]  598:   olddatapointer : INTEGER;
[GV3]  599:   startp, endp    : INTEGER;
[UY3]  6ØØ:   datasegment     : INTEGER;
[C73]  6Ø1:   stor            : INTEGER;
[154]  6Ø2:   waitarpointer   : INTEGER;
[Ø53]  6Ø3:   func            : BOOLEAN;
[ØØØ]  6Ø4:
[MUØ]  6Ø5: BEGIN
[SØ2]  6Ø6:   INLINE  ($FA);
[P64]  6Ø7:   IF (progptr < proglength)
[ZB3]  6Ø8:      AND progtoggle THEN
[MUØ]  6Ø9:   BEGIN
[BP3]  61Ø:     IF tickstowait > Ø THEN
[F25]  611:       tickstowait:= tickstowait - 1;
[YU5]  612:     IF currentcommand = tokentype THEN
[MUØ]  613:     BEGIN
[BY5]  614:       startp:= MemW[tastbuf:startptr];
[H45]  615:       endp  := MemW[tastbuf:endptr];
[8F3]  616:       IF  startp = endp THEN
[MUØ]  617:       BEGIN
[DM3]  618:         progptr:= progptr + 1;
[D43]  619:         IF prog[progptr] =
[Y13]  62Ø:            strdelimiter THEN
[MUØ]  621:         BEGIN
[H74]  622:           progptr       := progptr + 1;
[LU5]  623:           currentcommand:= prog[progptr];
[4MØ]  624:         END
[ZRØ]  625:         ELSE
[MUØ]  626:         BEGIN
[4V4]  627:           IF prog[progptr] = Chr(Ø) THEN
[MUØ]  628:           BEGIN
[1A2]  629:             func    := TRUE;
[DM3]  63Ø:             progptr:= progptr + 1;
[4MØ]  631:           END
[ZRØ]  632:           ELSE
[432]  633:             func:= FALSE;
[6F3]  634:           writekbdbuffer(func,
[BN2]  635:                          prog[progptr])
[ZNØ]  636:         END;
[ZNØ]  637:       END;
[4MØ]  638:     END
[ZRØ]  639:     ELSE
[1H3]  64Ø:       commandinterpreter;
[ZNØ]  641:   END;
[Z35]  642:   IF (progptr >= proglength) THEN
[ØB3]  643:     progtoggle:= FALSE;
[9Y1]  644:   INLINE ($FB);
[ZNØ]  645: END;
[ØØØ]  646:
[2H6]  647: PROCEDURE initializeglobalvariables;
[ØØØ]  648:
[MUØ]  649: BEGIN
[1R2]  65Ø:   progptr       := 1;
[L33]  651:   stackpointer  := Ø;
[LU5]  652:   currentcommand:= prog[progptr];
[HF3]  653:   progtoggle    := TRUE;
[3E3]  654:   writexposition:= 1;
[6E3]  655:   writeyposition:= 1;
[M63]  656:   testwindowx1  := 1;
[S63]  657:   testwindowy1  := 1;
[T73]  658:   testwindowx2  := 8Ø;
[Z73]  659:   testwindowy2  := 25;
[X23]  66Ø:   tickstowait   := Ø;
[ZNØ]  661: END;
[ØØØ]  662:
[1M3]  663: PROCEDURE printproc;
[ØØØ]  664:
[8X1]  665: INTERRUPT;
[ØØØ]  666:
[MUØ]  667: BEGIN
[KU1]  668:   INLINE($FA);
[CZ4]  669:   progtoggle:= NOT progtoggle;
[Z35]  67Ø:   IF (progptr >= proglength) THEN
[NL4]  671:     initializeglobalvariables;
[NU1]  672:   INLINE($FB);
[ZNØ]  673: END;
[ØØØ]  674:
```

```
[PS6]  675: FUNCTION fileexist(fn: STRING): BOOLEAN;
[ØØØ]  676:
[RNØ]  677: VAR
[RH1]  678:   fv: TEXT;
[ØØØ]  679:
[MUØ]  68Ø: BEGIN
[782]  681:   Assign(fv,fn);
[DXØ]  682: ($I-)
[NM1]  683:   Reset(fv);
[BXØ]  684: ($I+)
[5A3]  685:   IF IOResult <> Ø THEN
[BX2]  686:     fileexist:= FALSE
[ZRØ]  687:   ELSE
[MUØ]  688:   BEGIN
[WZ2]  689:     fileexist:= TRUE;
[CL1]  69Ø:     Close(fv);
[ZNØ]  691:   END;
[ZNØ]  692: END;
[ØØØ]  693:
[M86]  694: FUNCTION buchstabe(ch: CHAR): BOOLEAN;
[ØØØ]  695:
[MUØ]  696: BEGIN
[RR4]  697:  IF UpCase(ch) IN ['A'..'Z'] THEN
[2U2]  698:    buchstabe:= TRUE
[ZRØ]  699:  ELSE
[SØ3]  7ØØ:    buchstabe:= FALSE;
[ZNØ]  7Ø1: END;
[ØØØ]  7Ø2:
[AY5]  7Ø3: FUNCTION ziffer(ch: CHAR): BOOLEAN;
[ØØØ]  7Ø4:
[MUØ]  7Ø5: BEGIN
[UD3]  7Ø6:   IF ch IN ['Ø'..'9'] THEN
[N72]  7Ø7:     ziffer:= TRUE
[ZRØ]  7Ø8:   ELSE
[NK2]  7Ø9:     ziffer:= FALSE;
[ZNØ]  71Ø: END;
[ØØØ]  711:
[XP5]  712: PROCEDURE displaycompressedcode;
[ØØØ]  713:
[RNØ]  714: VAR
[8V1]  715:   i: INTEGER;
[ØØØ]  716:
[MUØ]  717: BEGIN
[XF1]  718:   WriteLn;
[XF1]  719:   WriteLn;
[3S4]  72Ø:   WriteLn('Compressed Code: ');
[XF1]  721:   WriteLn;
[GB4]  722:   FOR i:= 1 TO proglength DO
[MUØ]  723:   BEGIN
[7V3]  724:     IF prog[i] =  Chr(13) THEN
[L22]  725:       Write(Chr(2Ø))
[ZRØ]  726:     ELSE
[TR2]  727:       Write(prog[i]);
[ZNØ]  728:   END;
[XF1]  729:   WriteLn;
[XF1]  73Ø:   WriteLn;
[ZNØ]  731: END;
[ØØØ]  732:
[VD4]  733: FUNCTION getdata: BOOLEAN;
[ØØØ]  734:
[RNØ]  735: VAR
[MY2]  736:   indat            : TEXT;
[5K2]  737:   ch               : CHAR;
[U24]  738:   inputfilename     : STRING;
[594]  739:   datapointer       : LONGINT;
[7V3]  74Ø:   readexpr          : BOOLEAN;
[NC1]  741:   readno,i,
[GV4]  742:   sourcelinecounter: INTEGER;
[ØØØ]  743:
[EØ6]  744: PROCEDURE readandcount(VAR ch:`CHAR);
[ØØØ]  745:
[MUØ]  746: BEGIN
[DF2]  747:   Read(indat, ch);
[HW2]  748:   IF ch = Chr(1Ø) THEN
[WH3]  749:     sourcelinecounter:=
[TP3]  75Ø:                     sourcelinecounter + 1;
[ZNØ]  751: END;
[ØØØ]  752:
[B16]  753: PROCEDURE displayerror(st: STRING);
[ØØØ]  754:
[MUØ]  755: BEGIN
[3D5]  756:   WriteLn('SYNTAX ERROR in line ',
[TV4]  757:           sourcelinecounter:1, '. ', st);
[ZNØ]  758: END;
[ØØØ]  759:
[9M6]  76Ø: PROCEDURE readcommand(VAR token : CHAR;
[RØ5]  761:                       VAR readexpression: BOOLEAN;
[CW4]  762:                       VAR readnumbers   : INTEGER);
[RNØ]  763: VAR
[GC1]  764:   ch: CHAR;
[U12]  765:   st: STRING;
[MUØ]  766: BEGIN
[P31]  767:   REPEAT
[VV2]  768:     readandcount(ch);
[8C2]  769:     IF ch = '{' THEN
[P31]  77Ø:     REPEAT
[VV2]  771:       readandcount(ch);
[YX4]  772:     UNTIL (ch = '}') OR EoF(indat);
[VC5]  773:   UNTIL buchstabe(ch) OR EoF(indat);
```

```
[L21]  774:    st:= '';
[P31]  775:    REPEAT
[WB3]  776:      st:= st + UpCase(ch);
[VV2]  777:      readandcount(ch);
[A16]  778:    UNTIL NOT buchstabe(ch) OR EoF(indat);
[1S5]  779:    IF buchstabe(ch) AND EoF(indat) THEN
[WB3]  78Ø:      st:= st + UpCase(ch);
[TF2]  781:    token:= Chr(Ø);
[734]  782:    readexpression:= FALSE;
[T13]  783:    readnumbers    := Ø;
[SF2]  784:    IF st = 'IF' THEN
[MT2]  785:      token:= tokenif;
[EZ2]  786:    IF st = 'ENDIF' THEN
[1D3]  787:      token:= tokenendif;
[W73]  788:    IF st = 'REPEAT' THEN
[SH3]  789:      token:= tokenrepeat;
[Ø23]  79Ø:    IF st = 'UNTIL' THEN
[LF3]  791:      token:= tokenuntil;
[VN2]  792:    IF st = 'FOR' THEN
[WY2]  793:      token:= tokenfor;
[H83]  794:    IF st = 'ENDFOR' THEN
[DK3]  795:      token:= tokenendfor;
[4X2]  796:    IF st = 'TRUE' THEN
[U63]  797:      token:= tokentrue;
[NP2]  798:    IF st = 'NOT' THEN
[PZ2]  799:     token:= tokennot;
[PY2]  8ØØ:    IF st = 'FALSE' THEN
[9C3]  8Ø1:      token:= tokenfalse;
[1X2]  8Ø2:    IF st = 'EXIT' THEN
[R63]  8Ø3:      token:= tokenexit;
[783]  8Ø4:    IF st = 'CLRSCR' THEN
[3K3]  8Ø5:      token:= tokenclrscr;
[XZ2]  8Ø6:    IF st = 'PAUSE' THEN
[GD3]  8Ø7:      token:= tokenpause;
[FL3]  8Ø8:    IF st = 'LASTTEST' THEN
[5X3]  8Ø9:      token:= tokenlasttest;
[DA3]  81Ø:    IF st = 'WINDOW' THEN
[MUØ]  811:    BEGIN
[9M3]  812:      token:= tokenwindow;
[RN2]  813:      readnumbers:= 4;
[ZNØ]  814:    END;
[VN2]  815:    IF st = 'FOR' THEN
[MUØ]  816:    BEGIN
[SE3]  817:      token      := tokenfor;
[BN2]  818:      readnumbers:= 1;
[ZNØ]  819:    END;
[VV2]  82Ø:    IF st = 'WAIT' THEN
[MUØ]  821:    BEGIN
[HK3]  822:      token      := tokenwait;
[BN2]  823:      readnumbers:= 1;
[ZNØ]  824:    END;
[VX2]  825:    IF st = 'TYPE' THEN
[MUØ]  826:    BEGIN
[2Z3]  827:      token         := tokentype;
[SX3]  828:      readexpression:= TRUE;
[ZNØ]  829:    END;
[MB3]  83Ø:    IF st = 'GOTOXY' THEN
[MUØ]  831:    BEGIN
[AZ3]  832:      token      := tokengotoxy;
[FN2]  833:      readnumbers:= 2;
[ZNØ]  834:    END;
[C23]  835:    IF st = 'WRITE' THEN
[MUØ]  836:    BEGIN
[934]  837:      token         := tokenwrite;
[SX3]  838:      readexpression:= TRUE;
[ZNØ]  839:    END;
[CW2]  84Ø:    IF st = 'TEST' THEN
[MUØ]  841:    BEGIN
[1Ø4]  842:      token         := tokentest;
[SX3]  843:      readexpression:= TRUE;
[ZNØ]  844:    END;
[FV2]  845:    IF (token = Chr(Ø))
[7X3]  846:      AND NOT EoF(indat) THEN
[6C5]  847:      displayerror('Invalid Command: "'
[PA1]  848:                   + st + '"');
[ZNØ]  849: END;
[ØØØ]  85Ø:
[EW3]  851: PROCEDURE puttimedate;
[ØØØ]  852:
[RNØ]  853: VAR
[MY4]  854:   hour, min, second, sec1ØØ  : WORD;
[1K5]  855:   year, month, day, dayofweek: WORD;
[DE5]  856:   timestring                 : STRING[3Ø];
[F25]  857:   s1, s2, s3, s4             : STRING[1Ø];
[ES3]  858:   i                          : INTEGER;
[ØØØ]  859:
[MUØ]  86Ø: BEGIN
[BE5]  861:   GetTime(hour, min, second, sec1ØØ);
[526]  862:   GetDate(year, month, day, dayofweek);
[ZB2]  863:   Str(day:2, s1);
[9M2]  864:   IF s1[1] = ' ' THEN
[7S1]  865:     s1[1]:= 'Ø';
[RR2]  866:   Str(month:2, s2);
[BM2]  867:   IF s2[1] = ' ' THEN
[AS1]  868:     s2[1]:= 'Ø';
[DH2]  869:   Str(hour:2, s3);
[DM2]  87Ø:   IF s3[1] = ' ' THEN
[DS1]  871:     s3[1]:= 'Ø';
[3C2]  872:   Str(min:2, s4);
[FM2]  873:   IF s4[1] = ' ' THEN
[GS1]  874:     s4[1]:= 'Ø';
[AL4]  875:   timestring:= s1 + s2 + s3 + s4;
[ØF2]  876:   FOR i:= 1 TO 8 DO
[MUØ]  877:   BEGIN
[BL5]  878:     datapointer      := datapointer + 1;
[496]  879:     prog[datapointer]:= timestring[i];
[ZNØ]  88Ø:   END;
[ZNØ]  881: END;
[ØØØ]  882:
[ØE4]  883: PROCEDURE readexpression;
[ØØØ]  884:
[RNØ]  885: VAR
[582]  886:   ch           : CHAR;
[H23]  887:   ende         : BOOLEAN;
[5Ø3]  888:   delimiter    : CHAR;
[ZT3]  889:   numberstring : STRING;
[BØ4]  89Ø:   numbervalue  : INTEGER;
[H83]  891:   error        : INTEGER;
[ØØØ]  892:
[MUØ]  893: BEGIN
[712]  894:   ende:= FALSE;
[P31]  895:   REPEAT
[P31]  896:     REPEAT
[VV2]  897:       readandcount(ch);
[TH2]  898:       ch:= UpCase(ch);
[FM4]  899:     UNTIL (ch = '"') OR (ch = Chr(39))
[AF3]  9ØØ:           OR (ch = 'C') OR (ch = 'T')
[CX3]  9Ø1:           OR (ch = ')') OR EoF(indat);
[TC4]  9Ø2:     IF EoF(indat) OR (ch = ')') THEN
[ZV1]  9Ø3:       ende:= TRUE
[ZRØ]  9Ø4:     ELSE
[MUØ]  9Ø5:     BEGIN
[XL4]  9Ø6:       IF (ch = '"') OR (ch = Chr(39)) THEN
[MUØ]  9Ø7:       BEGIN
[AM2]  9Ø8:         delimiter:= ch;
[P31]  9Ø9:         REPEAT
[VV2]  91Ø:           readandcount(ch);
[DZ4]  911:           datapointer:= datapointer + 1;
[E54]  912:           prog[datapointer]:= ch;
[5G3]  913:         UNTIL (ch = delimiter)
[N32]  914:               OR EoF(indat);
[NZ4]  915:         datapointer:= datapointer - 1;
[ZNØ]  916:       END;
[DE2]  917:       IF (ch = 'C') THEN
[MUØ]  918:       BEGIN
[P31]  919:         REPEAT
[VV2]  92Ø:           readandcount(ch);
[495]  921:         UNTIL ziffer(ch) OR EoF(indat);
[Y23]  922:             numberstring:= ch;
[P31]  923:         REPEAT
[VV2]  924:           readandcount(ch);
[XL5]  925:           numberstring:= numberstring + ch;
[5C3]  926:         UNTIL (NOT ziffer(ch))
[N32]  927:               OR EoF(indat);
[X43]  928:         numberstring[Ø]:=
[VV4]  929:              Chr(Length(numberstring) - 1);
[385]  93Ø:         Val(numberstring, numbervalue,
[291]  931:             error);
[9T2]  932:         IF error <> Ø THEN
[BB5]  933:           displayerror('Unvalid numeric '
[832]  934:                        + 'constant: '
[CM2]  935:                        + numberstring);
[DZ4]  936:         datapointer:= datapointer + 1;
[4K3]  937:         prog[datapointer]:=
[223]  938:                           Chr(numbervalue);
[ZNØ]  939:       END;
[ZNØ]  94Ø:     END;
[HG2]  941:     IF (ch = 'T') THEN
[172]  942:       puttimedate;
[AY1]  943:   UNTIL ende;
[BL5]  944:   datapointer      := datapointer + 1;
[VY5]  945:   prog[datapointer]:= strdelimiter;
[ZNØ]  946: END;
[ØØØ]  947:
[RZ5]  948: PROCEDURE readnumbers(no: INTEGER);
[ØØØ]  949:
[RNØ]  95Ø: VAR
[5M2]  951:   i          : INTEGER;
[1L3]  952:   nostr      : STRING[3Ø];
[4W3]  953:   numbervalue: LONGINT;
[E33]  954:   error      : INTEGER;
[7M3]  955:   transf     : inttochar;
[ØØØ]  956:
[MUØ]  957: BEGIN
[HW2]  958:   FOR i:= 1 TO no DO
[MUØ]  959:   BEGIN
[YT1]  96Ø:     nostr:= '';
[P31]  961:     REPEAT
[VV2]  962:       readandcount(ch);
[KV2]  963:     UNTIL ziffer(ch);
[Z73]  964:     WHILE ziffer(ch) DO
[MUØ]  965:       BEGIN
[K63]  966:         nostr:= nostr + ch;
[VV2]  967:         readandcount(ch);
[ZNØ]  968:       END;
[595]  969:     Val(nostr, numbervalue, error);
[9T2]  97Ø:     IF error <> Ø THEN
[BB5]  971:     displayerror('Unvalid numeric '
[9K3]  972:                  + 'constant: ' + nostr);
```

Listing 1. »prokey.pas«

```
[H44]  973:      IF numbervalue > 32767 THEN
[K46]  974:        displayerror('Numeric constant out '
[BB3]  975:                     + 'of Range: ' + nostr);
[PE5]  976:      transf.intrepr    := numbervalue;
[BL5]  977:      datapointer       := datapointer + 1;
[9S5]  978:      prog[datapointer]:= transf.char1;
[BL5]  979:      datapointer       := datapointer + 1;
[AS5]  98Ø:      prog[datapointer]:= transf.char2;
[ZNØ]  981:    END;
[ZNØ]  982: END;
[ØØØ]  983:
[MUØ]  984: BEGIN
[FV3]  985:   sourcelinecounter:= 1;
[F15]  986:   inputfilename     := ParamStr(1);
[D34]  987:   IF inputfilename = '' THEN
[W75]  988:     inputfilename:= defaultinfile;
[DE5]  989:   IF fileexist(inputfilename) THEN
[MUØ]  99Ø:   BEGIN
[EM2]  991:     getdata:= TRUE;
[4W4]  992:     Assign(indat, inputfilename);
[G62]  993:     Reset(indat);
[9N2]  994:     datapointer:= Ø;
[P31]  995:     REPEAT
[XL5]  996:       readcommand(ch, readexpr, readno);
[DZ4]  997:       datapointer:= datapointer + 1;
[MN5]  998:       IF datapointer > sourcelength THEN
[MUØ]  999:       BEGIN
[W95] 1ØØØ:         Write('ERROR Source too long');
[DUØ] 1ØØ1:         Halt;
[ZNØ] 1ØØ2:       END;
[E54] 1ØØ3:       prog[datapointer]:= ch;
[2P2] 1ØØ4:       IF readexpr THEN
[BV2] 1ØØ5:         readexpression;
[3P2] 1ØØ6:       IF readno > Ø THEN
[3F3] 1ØØ7:         readnumbers(readno);
[ST2] 1ØØ8:     UNTIL EoF(indat);
[752] 1ØØ9:     Close(indat);
[LB4] 1Ø1Ø:     proglength:= datapointer;
[NL4] 1Ø11:     initializeglobalvariables;
[4MØ] 1Ø12:   END
[ZRØ] 1Ø13:   ELSE
[MUØ] 1Ø14:   BEGIN
[ØP2] 1Ø15:     getdata:= FALSE;
[UZ4] 1Ø16:     WriteLn('File ',inputfilename,
[5Y2] 1Ø17:             ' does not exist! ');
[ZNØ] 1Ø18:   END;
[ZNØ] 1Ø19: END;
[ØØØ] 1Ø2Ø:
[C45] 1Ø21: PROCEDURE initscreenstartadd;
[ØØØ] 1Ø22:
[RNØ] 1Ø23: VAR
[LU2] 1Ø24:    regs: REGISTERS;
[ØØØ] 1Ø25:
[MUØ] 1Ø26: BEGIN
[K12] 1Ø27:   regs.ah:= $ØF;
[TB2] 1Ø28:   Intr($1Ø, regs);
[2W2] 1Ø29:   IF regs.al = 7 THEN
[6L3] 1Ø3Ø:     screenstartadd:= $BØØØ
[ZRØ] 1Ø31:   ELSE
[UU3] 1Ø32:     screenstartadd:= $B8ØØ;
[ZNØ] 1Ø33: END;
[ØØØ] 1Ø34:
[CR3] 1Ø35: PROCEDURE saveintvec;
[ØØØ] 1Ø36:
[RNØ] 1Ø37: VAR
[V23] 1Ø38:   f: FILE OF POINTER;
[5Y3] 1Ø39:   timerinterruptvector,
[X34] 1Ø4Ø:   timerinterruptroutine,
[K44] 1Ø41:   prtscrinterruptvector,
[SZ5] 1Ø42:   prtscrinterruptroutine : POINTER;
[Ø14] 1Ø43:   pspaddress             : WORD;
[HK4] 1Ø44:   psppointer             : POINTER;
[ØØØ] 1Ø45:
[MUØ] 1Ø46: BEGIN
[LB6] 1Ø47:   GetIntVec($1C, timerinterruptvector);
[D16] 1Ø48:   timerinterruptroutine:= @intproc;
[3G6] 1Ø49:   GetIntVec($5, prtscrinterruptvector);
[ØP6] 1Ø5Ø:   prtscrinterruptroutine:= @printproc;
[BS3] 1Ø51:   pspaddress:= CSeg - 16;
[ØW5] 1Ø52:   psppointer:= POINTER(pspaddress);
[KN3] 1Ø53:   Assign(f, vectorfile);
[YY1] 1Ø54:   Rewrite(f);
[CB5] 1Ø55:   Write(f, timerinterruptvector,
[X34] 1Ø56:         timerinterruptroutine,
[K44] 1Ø57:         prtscrinterruptvector,
[PK6] 1Ø58:         prtscrinterruptroutine,psppointer);
[KD1] 1Ø59:   Close(f);
[ZNØ] 1Ø6Ø: END;
[ØØØ] 1Ø61:
[8W5] 1Ø62: FUNCTION alreadyresident: BOOLEAN;
[ØØØ] 1Ø63:
[RNØ] 1Ø64: VAR
[V23] 1Ø65:   f: FILE OF POINTER;
[ØØØ] 1Ø66:
[MUØ] 1Ø67: BEGIN
[KN3] 1Ø68:  'Assign(f, vectorfile);
[DXØ] 1Ø69:   {$I-}
[VE1] 1Ø7Ø:   Reset(f);
[BXØ] 1Ø71:   {$I+}
[263] 1Ø72:   IF IOResult = Ø THEN
```

```
[MUØ] 1Ø73:   BEGIN
[KD1] 1Ø74:     Close(f);
[164] 1Ø75:     alreadyresident:= TRUE;
[4MØ] 1Ø76:   END
[ZRØ] 1Ø77:   ELSE
[L74] 1Ø78:     alreadyresident:= FALSE;
[ZNØ] 1Ø79: END;
[ØØØ] 1Ø8Ø:
[S54] 1Ø81: PROCEDURE enableprokey;
[ØØØ] 1Ø82:
[MUØ] 1Ø83: BEGIN
[732] 1Ø84:   saveintvec;
[S33] 1Ø85:   tickstowait   := 9;
[VZ4] 1Ø86:   currentcommand:= tokenwait;
[YP2] 1Ø87:   progptr       := Ø;
[H94] 1Ø88:   SetIntVec($5, @printproc);
[414] 1Ø89:   SetIntVec($1C, @intproc);
[ZNØ] 1Ø9Ø: END;
[ØØØ] 1Ø91:
[694] 1Ø92: PROCEDURE disableprokey;
[ØØØ] 1Ø93:
[RNØ] 1Ø94: VAR
[GG6] 1Ø95:   vector, timervec, timerproc : POINTER;
[FW5] 1Ø96:    prtscrvec, prtscrproc      : POINTER;
[V23] 1Ø97:    f: FILE OF POINTER;
[HW4] 1Ø98:    psppointer                 : POINTER;
[A22] 1Ø99:    pspaddress,
[3Ø5] 11ØØ:    segadrenvironment          : WORD;
[PN4] 11Ø1:    regs                       : REGISTERS;
[ØØØ] 11Ø2:
[MUØ] 11Ø3: BEGIN
[KN3] 11Ø4:   Assign(f, vectorfile);
[DXØ] 11Ø5:   {$I-}
[VE1] 11Ø6:   Reset(f);
[BXØ] 11Ø7:   {$I+}
[5A3] 11Ø8:   IF IOResult <> Ø THEN
[6G5] 11Ø9:     WriteLn('ERROR: Prokey has to be
[TN2] 111Ø:             + 'started first')
[ZRØ] 1111:   ELSE
[MUØ] 1112:   BEGIN
[XM6] 1113:     Read(f, timervec, timerproc, prtscrvec,
[B84] 1114:          prtscrproc, psppointer);
[KD1] 1115:     Close(f);
[5M4] 1116:     WriteLn('Disabling Prokey');
[GC1] 1117:     Erase(f);
[5Z3] 1118:     SetIntVec($1C, timervec);
[N34] 1119:     SetIntVec($5, prtscrvec);
[V95] 112Ø:     pspaddress:= WORD(psppointer);
[ZF3] 1121:     segadrenvironment:=
[XT3] 1122:                       MemW[pspaddress:$Ø2C];
[VY4] 1123:     regs.es:= segadrenvironment;
[WØ2] 1124:     regs.ah:= $49;
[3Ø2] 1125:     MsDos(regs);
[ZH3] 1126:     regs.es:= pspaddress;
[WØ2] 1127:     regs.ah:= $49;
[3Ø2] 1128:     MsDos(regs);
[ZNØ] 1129:   END;
[ZNØ] 113Ø: END;
[ØØØ] 1131:
[MUØ] 1132: BEGIN
[FT3] 1133:   strdelimiter:= Chr(255);
[1Ø4] 1134:   IF alreadyresident THEN
[GB2] 1135:     disableprokey
[ZRØ] 1136:   ELSE
[MUØ] 1137:   BEGIN
[C63] 1138:     success:=  getdata;
[5G2] 1139:     IF success THEN
[MUØ] 114Ø:     BEGIN
[DK3] 1141:       initscreenstartadd;
[N82] 1142:       enableprokey;
[F92] 1143:       makeresident;
[ZNØ] 1144:     END;
[ZNØ] 1145:   END;
[KNØ] 1146: END.
```

Listing 1. »prokey.pas«

```
[M17]    1: { Demonstrationsprogramm für Prokey }
[ØØØ]    2:
[C93]    3: type('turbo',chr(13))
[NF2]    4: type('e',chr(13))
[1Ø2]    5: Gotoxy(3,23);
[NZ5]    6: Write('Ich tippe das Programm ein ... ')
[ØØØ]    7:
[VD6]    8: { Wegen der Auto-Indent-Funktion des
[3N6]    9:   Turbo-Pascal Editors wird am Ende jeder
[A26]   1Ø:   Zeile <Pos1> gedrückt (chr(Ø),chr(71)) }
[ØØØ]   11:
[Z66]   12: type("PROGRAM TastaturCodes;",chr(13),
[U32]   13:      chr(Ø),chr(71))
[LY5]   14: type("Uses crt;",chr(13),chr(Ø),chr(71))
[DT5]   15: type("VAR FuncKey: Boolean;",chr(13),
[U32]   16:      chr(Ø),chr(71))
[YY4]   17: type("    Ch: Char;",chr(13),chr(Ø),
[W41]   18:      chr(71))
```

Listing 2. »demo« zeigt Ihnen die Fähigkeiten von »Prokey«

```
[8A5]   19: type("BEGIN",chr(13),chr(Ø),chr(71))
[GT5]   2Ø: type("   ClrScr;",chr(13),chr(Ø),chr(71))
[9B5]   21: type("   Writeln('Ermittlung der"
[7X4]   22:      "Tastaturcodes für Prokey');",
[L83]   23:       chr(13),chr(Ø),chr(71))
[HT5]   24: type("   REPEAT",chr(13),chr(Ø),chr(71))
[145]   25: type("      Writeln;",chr(13),chr(Ø),
[W41]   26:      chr(71))
[UC6]   27: type("      Writeln('Bitte Taste drücken"
[UF3]   28:      " (<ESC> für Ende)');",
[L83]   29:       chr(13),chr(Ø),chr(71))
[WK5]   3Ø: type("      FuncKey := False;",chr(13),
[U32]   31:      chr(Ø),chr(71))
[WX4]   32: type("      Ch := ReadKey;",chr(13),
[U32]   33:      chr(Ø),chr(71))
[KF5]   34: type("      IF Ch = Chr(Ø) THEN",chr(13),
[U32]   35:      chr(Ø),chr(71))
[WG5]   36: type("      BEGIN",chr(13),chr(Ø),chr(71))
[Y25]   37: type("         Ch := ReadKey;",chr(13),
[U32]   38:      chr(Ø),chr(71))
[MH5]   39: type("         FuncKey := True;",chr(13),
[U32]   4Ø:      chr(Ø),chr(71))
[KD5]   41: type("      END;",chr(13),chr(Ø),chr(71))
[865]   42: type("      IF FuncKey THEN",chr(13),
[U32]   43:      chr(Ø),chr(71))
[NV5]   44: type("         Writeln('Tastaturcodes:"
[XL2]   45:      " Ø und ',Ord(Ch))",
[L83]   46:       chr(13),chr(Ø),chr(71))
[KF5]   47: type("      ELSE",chr(13),chr(Ø),chr(71))
[NT5]   48: type("         Writeln('Tastaturcode: ',
[3M1]   49:      Ord(Ch));",
[L83]   5Ø:       chr(13),chr(Ø),chr(71))
[HA5]   51: type("   UNTIL Ch = Chr(27);",chr(13),
[U32]   52:      chr(Ø),chr(71))
[915]   53: type("END.",chr(13),chr(Ø),chr(71))
[ØØØ]   54:
[BW1]   55: Gotoxy(3,23)
[UB5]   56: Write('Ich compiliere und starte'
[DG2]   57:       ' das Programm ')
[RYØ]   58: for 3Ø
[HD1]   59:    write('.')
[G31]   6Ø:    wait(1)
[Y21]   61: endfor
[AT3]   62: type(chr(Ø),chr(68),'rr')
[N51]   63: wait(1Ø)
[VZ3]   64: repeat until test('File')
[N51]   65: wait(1Ø)
[BW1]   66: Gotoxy(3,23)
[U46]   67: Write('Ich verlasse Turbo-Pascal ... ')
[Y51]   68: wait(4Ø)
[ØX3]   69: type(chr(Ø),chr(68),'fqn')
[ØØØ]   7Ø:
```

Listing 2. »demo« zeigt Ihnen die Fähigkeiten von »Prokey«

```
[214]    1: PROGRAM TastaturCodes;
[5H1]    2: Uses crt;
[ØN3]    3: VAR FuncKey: Boolean;
[GC1]    4:     Ch: Char;
[MUØ]    5: BEGIN
[DB1]    6:    ClrScr;
[7V6]    7:    Writeln('Ermittlung der Tastaturcodes
[RC2]    8:            ' für Prokey');
[P31]    9:    REPEAT
[XF1]   1Ø:       Writeln;
[SU4]   11:       Writeln('Bitte Taste/Tasten'
[2N3]   12:        'kombination drücken'
[E73]   13:        ' (<ESC> für Ende)');
[DU2]   14:       FuncKey := False;
[152]   15:       Ch := ReadKey;
[ZT2]   16:       IF Ch = Chr(Ø) THEN
[MUØ]   17:       BEGIN
[152]   18:          Ch := ReadKey;
[ØR2]   19:          FuncKey := True;
[ZNØ]   2Ø:       END;
[YD2]   21:       {InKey := Ch;}
[NG2]   22:       IF FuncKey THEN
[U84]   23:          Writeln('Tastaturcodes:'
[LE2]   24:                  ' Ø und ',Ord(Ch))
[ZRØ]   25:       ELSE
[MUØ]   26:       BEGIN
[Z54]   27:          Writeln('Tastaturcode: ',
[DB1]   28:                  Ord(Ch));
[ZNØ]   29:       END;
[ØV2]   3Ø:    UNTIL Ch = Chr(27);
[KNØ]   31: END.
```

Listing 3. Das Hilfsprogramm »tastcode.pas« wird von »Prokey« erzeugt





SBasic/MASM/TASM

SBasic 2.0: Der »Nachbrenner« für GW-Basic

Seit Veröffentlichung der ersten Version von »SBasic« in der DOS Extra 6/89 wurde der GW-Basic-Compiler Schritt um Schritt weiter verbessert. Neben allen bislang erfolgten Erweiterungen wartet die Version 2.0 mit einem schnellen, in Assembler programmierten Präprozessor, einer leistungsfähigen Oberfläche und zahlreichen internen Optimierungen auf.

Fast jeder PC, AT, 80386 und 80486 wird serienmäßig neben dem Betriebssystem MS-DOS mit dem Basic-Interpreter »GW-Basic« ausgeliefert. Für zahlreiche Anwendungen ist die Ausführungsgeschwindigkeit dieser »interpretierten Programme« nicht hoch genug. Außerdem kann jedermann die Quelltexte einsehen und manipulieren. Die kommerziell vertriebenen Basic-Compiler, die diese Probleme beseitigen, sind jedoch nicht gerade billig und oft so hochgezüchtet, daß sie »normale« GW-Basic-Programme gar nicht mehr ohne größere Änderungen verarbeiten. Deshalb bieten wir Ihnen mit dem Basic-Compiler »SBasic 2.0« eine preiswerte Alternative, die einen Großteil des GW-Basic-Befehlssatzes beherrscht und diesen ohne Änderungen übersetzt.

SBasic eignet sich besonders gut zum Programmieren von kleinen Hilfsprogrammen (Utilities), die bei herkömmlichen Compilern die ganze Laufzeitbibliothek enthalten und deshalb sehr umfangreich werden. Bei SBasic ist dies nicht der Fall, so daß Sie viel Speicherplatz auf dem Datenträger und im Hauptspeicher sparen. In der Version 2.0 verarbeitet SBasic nun zusätzlich zu den »regulären« GW-Basic-Dateien, die im Token-Format (auch als Binärformat bezeichnet) aufgezeichnet sind, auch ASCII-Dateien. Dieses Format erhalten Sie, wenn Sie unter GW-Basic ein Programm mit der Anweisung »save"name",a« speichern oder zur Programmeingabe einen Editor verwenden. SBasic ist gewissermaßen ein »Nachbrenner«, der Ihren GW-Basic-Programmen bezüglich Geschwindigkeit und Leistung neue Welten erschließt.

Zudem steht Ihnen unter SBasic mehr Speicherplatz zur Verfügung. Wie Sie vielleicht wissen, legt GW-Basic alle Daten (Systemvariablen, Programmtext, Variablen, Felder und Zeichenketten) im gleichen Segment ab. Dadurch stehen maximal 64 KByte für die Programmierung zur Verfügung, die bei größeren Projekten sehr schnell belegt sind. Der SBasic-Compiler kann im Gegensatz zu GW-Basic den gesamten freien Arbeitsspeicher Ihres Computers nutzen, da er mit mehreren Segmenten arbeitet, was bereits an der Erweiterung »exe« der kompilierten Programme erkennbar ist. Die Nutzung des Arbeitsspeichers ist nach dem folgenden Schema organisiert:

- 64 KByte für den Programmcode
- 64 KByte für alle numerischen Variablen einschließlich der Felder
- 1024 Byte für den Stack
- der restliche Speicher für Strings (Zeichenketten) und Stringarrays (Zeichenkettenfelder)

Die maximale Anzahl der Programmzeilen ist in der Variablen »maxzeil« in Zeile 112 und die maximale Anzahl der Variablennamen in der Variablen »maxvar« in Zeile 114 des SBasic-Quelltextes festgelegt. Diese beiden Parameter können Sie nach Belieben an Ihre Bedürfnisse anpassen.

Zeichenketten verwaltet SBasic mit viel weniger Aufwand als GW-Basic

Die Strings werden von SBasic gänzlich anders verwaltet, als dies beim GW-Basic-Interpreter der Fall ist. Jeder String belegt konstant 256 Byte und beginnt immer an einer Segmentadresse. Das erste Byte enthält dabei die Stringlänge. Dieses Verfahren geht zwar mit dem verfügbaren Speicherplatz sehr großzügig um, doch macht es zeitaufwendige dynamische Speichervergabe-Algorithmen und Garbage-Collection-Vorgänge (wörtlich Müllsammlung, gemeint ist damit die Freigabe nicht mehr benötigten Variablenspeichers) überflüssig. Ein weiterer Vorteil von SBasic ist das Einbin-

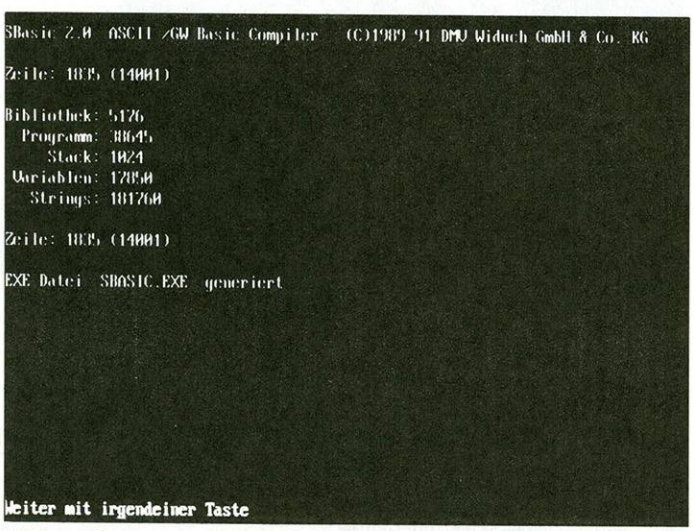


Bild 1. SBasic 2.0 wird als Schlußtest angewiesen, sich selbst nochmals zu übersetzen. Das ist ein anerkannter Test für die Qualität eines Compilers.

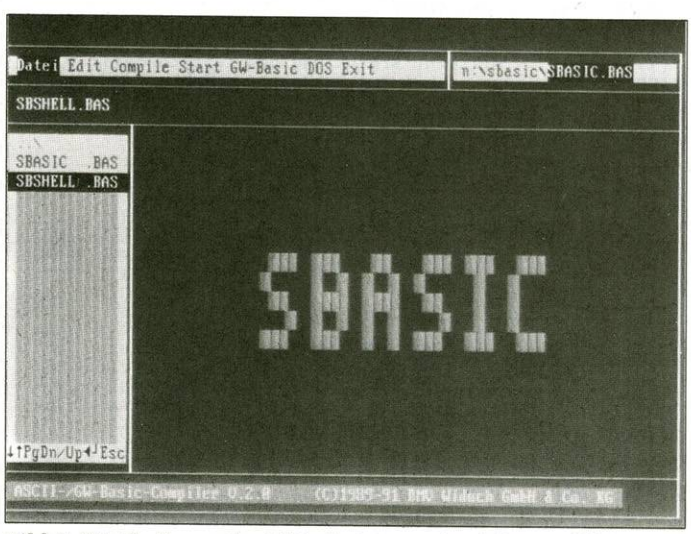


Bild 2. SB-Shell erweitert SBasic 2.0 um eine leistungsfähige Benutzeroberfläche. Über die DOS-Kommandozeile wurde »sbasic.bas« als zu bearbeitende Datei übergeben.

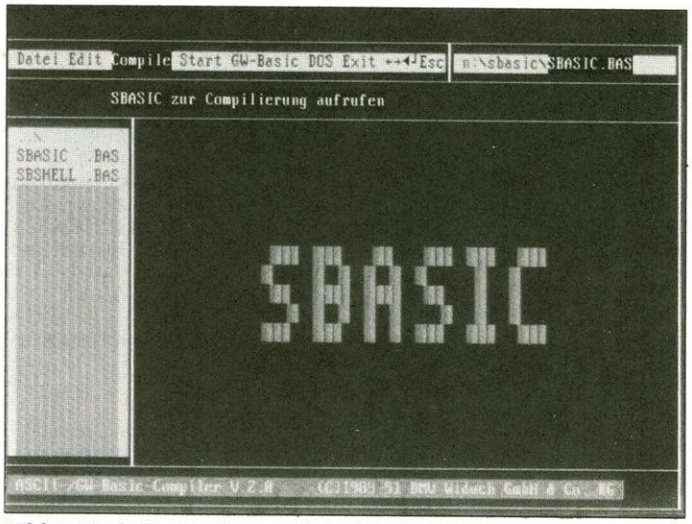


Bild 3. Nach der Festlegung der gewünschten Quelldatei (oben rechts im Bild ausgewiesen) muß bei SB-Shell lediglich »Compile« aufgerufen werden

den von Maschinencode-Unterprogrammen. Im Gegensatz zu anderen Compilern, bei denen Assembler-Programme erst an ein neues Format angepaßt werden müssen, verarbeitet SBasic die Maschinencode-Routinen im üblichen GW-Basic-Format. Es gibt allerdings zwei Einschränkungen, die die Strings betreffen. Zum einen muß ein String, in dem ein Maschinencode-Unterprogramm abgelegt ist, oder der als Parameter übergeben wird, im Zugriffsbereich des Datensegments liegen. Ist dies nicht der Fall, so erhält der Anwender den Fehler »String außer Reichweite« angezeigt. Definieren Sie dann die Zeichenkette einfach früher im Programmablauf, um sie näher an das Datensegment zu legen. Das zweite Problem tritt bei Maschinencode-Unterprogrammen auf, die mit String-Feldern arbeiten. Weil SBasic keine Deskriptoren kennt, muß es diese simulieren, und das geht natürlich nur bei einzelnen Strings gut, bei String-Feldern funktioniert es nicht.

An der alten Version sind nur wenige Änderungen notwendig, um die neuen Ergänzungen einzubauen

Sofern Sie nicht über SBasic in der Version 1.2 verfügen, ist es erforderlich, den Hexdump »sbasic.exe« (Listing 1) mit »SCheck« einzugeben, da die Version 2.0 sich nicht mehr mit GW-Basic selbst kompilieren kann. Dies ist lediglich noch mit Version 1.2 möglich. Außer dem Compiler selbst wird noch die Laufzeitbibliothek »sbasic.bin« (Listing 2) benötigt.

Wenn Sie SBasic bereits in der Version 1.2 besitzen, können Sie die Basic-Datei »sbasic.200« (Listing 3) zu SBasic 1.2 mit dem Merge-Befehl hinzuladen und diese neuere Version mit SBasic 1.2 kompilieren (Bild 1). Dadurch ist der Basic-Teil mit wenigen Eingaben auf dem neuesten Stand. Im Assembler-Teil (»sbasic.asm«) sind gegenüber der in DOS-Extra 10/90 erweiterten Laufzeitbibliothek folgende Korrekturen erforderlich, die den Lof-Befehl betreffen.

In Zeile 16 müssen Sie die Lof-Ende-Adresse »L9A« anfügen:

```
DW L90,L91,L92,L93,L94,L95,L96,L97,L98,L99,L9A
```

In den Zeilen 6317 und 6318 muß statt

```
MOV CH,CL
MOV DL,CH
```

jetzt

```
MOV CL,CH
MOV CH,DL
```

stehen. Dies setzt voraus, daß Sie die in der DOS Extra 6/89 erschienene Laufzeitbibliothek »sbasic.asm« eingegeben oder sich die Databox gekauft haben und über einen Assembler wie »MASM« oder »TASM« verfügen. Mit den folgenden Befehlen übersetzen Sie dann »sbasic.asm« in das erforderliche Binärformat.

```
masm sbasic;
link sbasic;
exe2bin sbasic.exe sbasic.com
```

Wenn Sie den Turbo-Assembler von Borland besitzen, ersetzen Sie »masm« und »link« durch »tasm« und »tlink«. Die Umgehung von »exe2bin« mit der Option »/t« von »tlink« funktioniert allerdings nicht, weil in »sbasic.asm« weder ein Einsprungspunkt angegeben ist, noch die Startadresse des Programms bei 100hex beginnt.

Ab dieser Stelle gibt es keinerlei Unterschiede mehr zwischen den Besitzern der Vorgängerversion von SBasic und allen Neueinsteigern. Bei dem in der Version 2.0 neu hinzugekommenen Präprozessor können Sie sich zwischen dem kürzeren Hexdump »sbasic.pre« (Listing 4) oder der Assembler-Quelltextversion »sbpre.asm« (Listing 5) entscheiden. Die letztere Alternative setzt einen Assembler wie »MASM« oder »TASM« voraus, erlaubt dafür aber beliebige Änderungen oder Anpassungen. Das Assemblieren und Linken mit verschiedenen Assemblern ist im Kopf der Quelltextversion (Listing 5) genau beschrieben.

Alle Teile des Compilers sind damit komplett. Die neue Oberfläche für SBasic finden Sie in Listing 6 (»sbshell.bas«). Zum komfortablen Aufruf der Oberfäche dient die kurze Stapeldatei »sb.bat« (Listing 7). »sbshell.bas« kann nun mit SBasic in ein eigenständig lauffähiges Programm umgewandelt werden.

SBasic kann mit einem neuen Befehl die Parameter aus der Kommandozeile einlesen

Nachdem alle Bestandteile des Basic-Compilers vorliegen, kann als erstes »Opfer« die Oberfläche »sbshell.bas« (Listing 6) übersetzt werden. Geben Sie nach dem Start von SBasic bei der Frage nach der zu kompilierenden Datei »sbshell« ein.

Während des Kompiliervorgangs zeigt SBasic immer die fortlaufende Nummer der aktuell bearbeiteten Zeile und in Klammern deren Zeilennummer im Quelltext an. Zur Halbzeit gibt SBasic eine kleine Statistik des Kompiliervorgangs aus. Diese Ausgabe erfolgt deswegen zur Halbzeit, weil es sich bei SBasic um einen Zwei-Pass-Compiler handelt, einen Compiler, der zum Übersetzen eines Programms zwei Durchläufe benötigt. Gegenüber professionellen Ein-Pass-Compilern entfällt somit das Anlegen einer speicheraufwendigen Vorwärtsreferenzliste, dafür arbeitet ein Zwei-Pass-Compiler allerdings auch langsamer. Gegenüber GW-Basic neu hinzugekommen ist bei SBasic die Funktion »command$()«. »command$« ist ein String-Feld, das die Parameter enthält, die beim Programmaufruf in der Kommandozeile übergeben wurden. Der Index beginnt ab 1, »command$(0)« ist immer leer. Alle Kleinbuchstaben wandelt SBasic in Großbuchstaben um, die Umlaute allerdings nicht. Zur Trennung der Parameter dient eine beliebige An-

Prioritätsebene	Operatoren
1	»–« (negatives Vorzeichen), »not«
2	»↑« (Potenz)
3	»*«, »/«, »\« (Integer-Division), »mod« (Rest einer Integer-Division)
4	»+«, »–« (Addition und Subtraktion)
5	»>«, »<«, »=«, »>=«, »<=«, »<>« (Vergleichsoperatoren)
6	»and«, »or«, »xor« (logische Operatoren)

Tabelle 1. Die Operatoren von SBasic und ihre Prioritätsebenen

Anweisung	Erklärung (und Besonderheiten gegenüber GW-Basic)
Befehle:	
BEEP	Ton ausgeben
BLOAD	Speicherblock lesen
BSAVE	Speicherblock schreiben
CALL	Maschinenprogramm aufrufen Der Call-Befehl des SBasic-Compilers ist kompatibel zum Call-Befehl des GW-Basic-Interpreters. (Bei Stringparametern kann jedoch der Fehler »String außer Reichweite« auftreten. In diesem Fall sollten die Strings am Anfang des Programms verwendet (definiert) werden, damit SBasic sie näher an das Datensegment legt. Stringfelder sind als Parameter nicht erlaubt. Ebenso sind Längenveränderungen von Strings in Maschinencode-Unterprogrammen nicht zulässig)
CHDIR	Aktuelles Verzeichnis wechseln
CLEAR	Variablen löschen und Dateien schließen (keine Zahlenangaben für Speicherplatz von Daten und Stack)
CLOSE	Eine oder alle Dateien schließen
CLS	Bildschirmfenster löschen
COLOR	Farben setzen
DATA	Konstanten definieren (die maximale Anzahl der Data-Zeilen ist in der Variablen »maxdata« in Programmzeile 118 von »sbasic.bas« festgelegt)
DEF FN	Benutzerfunktion definieren (maximal vier Parameter, maximale Anzahl der Funktionen: 50)
DEF SEG	Datensegment definieren
DEFINT, DEFSNG, DEFSTR	Variablentypen festlegen (keine dynamische Definition, das heißt, der Ausdruck »if test then defdbl a-z else defsng a-z« ist nicht zulässig)
DIM	Speicherplatz für Felder reservieren (maximal vier Dimensionen, keine dynamische Dimensionierung in der Art »dim feld(anzahl)«)
END	Programm beenden und Dateien schließen
FIELD	Datensatzfelder festlegen
FILES	Inhaltsverzeichnis ausgeben
FOR ... NEXT ... STEP	Zählschleife (die maximale Anzahl der For-Next-Schleifen ist in der Variablen »maxfor« in Zeile 120 von »sbasic.bas« festgelegt)
GET	Datensatz lesen
GOSUB	Unterprogramm aufrufen
GOTO	Zu neuer Adresse springen
IF ... THEN ... ELSE	Bedingte Verzweigung (die maximale Anzahl der If-Abfragen ist in der Variablen »maxif« in Programmzeile 116 von »sbasic.bas« festgelegt, auch verschachtelte Else-Konstrukte sind möglich)
INPUT, LINE INPUT	Zeichenkette lesen (der Insert-Modus bleibt bei Cursor-Bewegungen aktiv; wortweises Springen ist nicht möglich; [Home] führt an den Anfang der Zeile; eine Fehlermeldung in der Art »Redo from Start« unterbleibt bei Fehleingaben; alle Zeichen werden vom Bildschirm übernommen; [End] ist nicht nötig; Cursor-Bewegungen über den linken Eingabebeginn beziehungsweise den rechten Bildschirmrand hinaus werden verhindert; die Eingabeumleitung funktioniert; bei der Eingabe des Gerätes »kybd:« erscheint der Cursor)
KILL	Datei löschen (Wildcards dürfen nicht verwendet werden, in diesem Fall bietet sich als Ersatz »shell "del ..."« an)
LET	Zuweisung
LOCATE	Cursor positionieren
LSET	String linksbündig auffüllen
MKDIR	Unterverzeichnis anlegen
NAME	Datei umbenennen
ON ... GOSUB	Eines von mehreren Unterprogrammen aufrufen
ON ... GOTO	Zu einer von mehreren Adressen springen

OPEN	Datei öffnen
	(nur in der folgenden Form erlaubt:
	`open "name" [for modus] as #nummer [len=satzlänge]`
	das heißt, statt
	`open "o", #nummer, "name" [,satzlänge]`
	`open "i", #nummer, "name" [,satzlänge]`
	`open "r", #nummer, "name" [,satzlänge]`
	`open "a", #nummer, "name" [,satzlänge]`
	schreiben Sie:
	`open "name" for output as #nummer [len=satzlänge]`
	`open "name" for input as #nummer [len=satzlänge]`
	`open "name" for random as #nummer [len=satzlänge]`
	`open "name" for append as #nummer [len=satzlänge]`
	Für »satzlänge« sind maximal 128 Byte zulässig; maximal drei Dateien können gleichzeitig geöffnet sein; speziell behandelte Geräte sind »kybd:«, »scrn:«, »com1:« und »com2:«; für die serielle Schnittstelle ist der Eingabepuffer immer 64 Byte groß, »DTR« und »RTS« sind immer aktiv, standardmäßig sind acht Datenbits, keine Parität und ein Stopbit eingestellt, nur die Baudrate läßt sich verändern, auch »krumme« Baudraten von 3 bis 65535 sind erlaubt)

OUT	Wert auf Port ausgeben
PLAY	Musik spielen (die Hintergrundmodusanweisung »MB« wird ignoriert; Pausen müssen keine Längenangaben haben)
POKE	Wert in Speicher schreiben
PRINT, LPRINT	Werte oder Text ausgeben (die Zusatzanweisungen »tab« und »spc« werden unterstützt, aber die Sonderform »print using« und Ausgabeumleitungen sind nicht erlaubt. Wenn Sie in SBasic mehrere Werte durch Kommata getrennt in eine Datei schreiben möchten, müssen Sie dazu anstelle von »print #« den Befehl »write #« (gleiche Syntax) verwenden. Andernfalls lassen sich die Werte nicht mehr über den Input#-Befehl korrekt einlesen.)
PUT	Datensatz schreiben
RANDOMIZE	Zufallszahlenstartwert festlegen
READ	Datum lesen
REM	Kommentar des Programmierers
RESET	Dateien schließen
RESTORE	Data-Zeiger setzen
RETURN	Rücksprung von einem Unterprogramm (befindet sich keine Adresse auf dem Stack, so erfolgt Sprung ins Ungewisse)
RMDIR	Unterverzeichnis entfernen
RSET	String rechtsbündig auffüllen
RUN	Programm neu starten (Dateiangaben in der Art »run "name"« lassen sich nicht verwenden)
SCREEN	Videomodus setzen (Grafikmodi und Farbangaben sind nicht erlaubt, aber das Arbeiten mit mehreren Bildschirmseiten ist möglich)
SHELL	Befehlsinterpreter oder Programm aufrufen
SOUND	Ton erzeugen
STOP	Programm unter Angabe der Zeilennummer anhalten
SWAP	Variablen vertauschen
SYSTEM	Programm beenden
VIEW PRINT	Bildschirmfenster festlegen (die Standardeinstellung ist »1 to 24«, der maximal zulässige Wert darf auch 25 sein)
WAIT	Portadresse abfragen
WHILE ... WEND	Abweisende Schleife (die maximale Anzahl der While-Wend-Schleifen ist in der Variablen »maxwhile« in Zeile 122 von »sbasic.bas« festgelegt)
WIDTH	Bildschirmbreite festlegen (die Angabe eines Kanals ist nicht erlaubt)
WRITE	Variablenwerte ausgeben
Funktionen:	
ABS()	Absoluter Betrag
ASC()	Umwandlung eines ASCII-Zeichens in einen numerischen Wert
ATN()	Arkustangens
CHR$()	Umwandlung eines numerischen Wertes in ein ASCII-Zeichen
CINT()	Umwandlung eines Fließkommawertes in einen Integerwert
COMMAND$()	Enthält die Parameter aus der Kommandozeile beim Aufruf des Basic-Programms
COS()	Kosinus
CSRLIN	Zeile, in der sich der Cursor befindet
CVI()	Umwandlung eines Strings in einen Integerwert
CVS()	Umwandlung eines Strings in einen Fließkommawert
DATE$	Aktuelles Datum im amerikanischen Format
EOF()	Dateiende erreicht
EXP()	Exponentialfunktion »E ^ X« (die letzte Stelle ist unter Umständen nicht korrekt, weil keine zusätzlichen Stellen für die interne Verarbeitung vorgesehen sind)
FIX()	Vorkommaanteil
HEX$()	Umwandlung eines numerischen Wertes in einen hexadezimalen String
INKEY$	Gedrückte Taste
INP()	Status eines Port

Anweisung	Erklärung (und Besonderheiten gegenüber GW-Basic)
INPUT$()	Zeichen einlesen
INSTR()	String suchen
INT()	Ganzzahliger Wert
LEFT$()	String verkürzen
LEN()	Stringlänge
LOG()	Natürlicher Logarithmus (die letzte Ziffer ist unter Umständen nicht korrekt, weil keine zusätzlichen Stellen für die interne Verarbeitung vorgesehen sind)
MID$()	Teilstring (auch als Befehl erlaubt)
MKI$()	Umwandlung eines Integerwertes in einen String
MKS$()	Umwandlung eines Fließkommawertes in einen String
PEEK()	Wert einer Speicherzelle
POS()	Cursor-Spalte
RIGHT$()	Teilstring
RND()	Zufallszahl
SCREEN()	Bildschirmzeichen oder Attribut
SGN()	Vorzeichen
SIN()	Sinus
SPACE$()	String mit Leerzeichen
SQR()	Quadratwurzel (Genauigkeitsprobleme, weil »sqr(a)« dem Ausdruck »exp(log(a)/2« entspricht)
STICK()	Joystick-Stellung (ein nicht angeschlossener Joystick liefert den Wert 508 zurück)
STRIG()	Joystick-Knöpfe (diese Funktion gibt zurück, welcher Joystick-Knopf gerade gedrückt ist; die Tastenzustände werden nicht zwischengespeichert (gepuffert); das Schalten mit »strig(n) on/off/stop« ist nicht möglich)
STR$()	Umwandlung eines numerischen Wertes in einen String
STRING$()	Sich wiederholende Zeichen
TAN()	Tangens
TIME$	Systemzeit
TIMER	Anzahl der seit Mitternacht verstrichenen Sekunden
VAL()	Umwandlung eines String in einen numerischen Wert
VARPTR()	Zeiger auf Variable (unter Umständen treten hier Probleme mit Strings auf, siehe dazu »call«)

Tabelle 2. Diese Befehle unterstützt SBasic

zahl von Leerzeichen, ein Komma oder Schrägstrich. Letzterer erscheint zusätzlich im String. Im folgenden ein Beispiel für den Aufruf eines Programms mit dem Namen »test«:

```
test first,, third fourth/fifth
```

Nach diesem Programmaufruf enthält die Variable »command$(1)« den Text »first«, »command$(2)« bleibt aufgrund des Doppelkommas leer, »command$(3)« enthält den Text »third«, »command$(4)« ist auf »fourth« gesetzt und »command$(5)« enthält den Text »/fifth«. Daß die Command$-Funktion wirklich funktioniert, können Sie sofort feststellen, indem Sie in der MS-DOS-Kommandoebene eingeben

```
sbasic sbshell
```

Dadurch kompiliert SBasic wieder die Oberfäche, allerdings fragt es Sie nicht mehr nach dem Namen der zu kompilierenden Datei – den haben Sie ja bereits beim Aufruf angegeben.

Weil ein von SBasic kompiliertes Programm mehr Speicher als die entsprechende Interpreter-Version adressieren kann, läßt sich die Kapazität von SBasic durch folgenden Trick erhöhen: Erzeugen Sie eine zweite Version, in der Sie die Maximal-Parameter in den Zeilen 112 bis 123 an Ihre persönlichen Bedürfnisse anpassen. So ist es zum Beispiel ratsam, die Anzahl der For-Next-Schleifen und der Data-Zeilen zu erhöhen. Ändern Sie neben der jeweiligen Konstanten auch die Feldgrößen in den Dim-Anweisungen der folgenden Zeile, weil SBasic dynamische Felder in der Art

```
dim feld(max)
```

nicht unterstützt. Sichern Sie diese neue Version dann unter einem anderen Namen, und kompilieren Sie sie mit der ersten Version. Diese Technik kann natürlich nicht verwendet werden, wenn lediglich der Hexdump von SBasic eingegeben wurde.

Im Gegensatz zu GW-Basic, das grundsätzlich alle Zahlen vor einer Rechenoperation ins Fließkommaformat umwandelt, arbeitet SBasic mit echter Ganzzahl-Arithmetik. Als Beispiel können Sie einmal die Zahlen 2 und 32767 multiplizieren. Das Ergebnis lautet beim GW-Basic-Interpreter 65534, bei SBasic hingegen –2. Ist diese Form der Berechnung unerwünscht, so deklarieren Sie einfach eine der beiden Zahlen durch ein nachgestelltes Ausrufezeichen als einfachgenaue Variable; die Berechnung erfolgt dadurch im Fließkommaformat. Die logischen Operatoren »and«, »or«, »xor« und »not« führt SBasic übrigens immer im Integer-Format aus, auch wenn eine Fließkommazahl angegeben ist. Tabelle 1 zeigt die Operatoren von SBasic und ihre Prioritätsebenen, Tabelle 2 listet alle unter SBasic zulässigen Anweisungen mit Kurzbedeutung und Besonderheiten auf.

Zusätzlich erhalten Sie auch eine Benutzeroberfläche, die die Programmentwicklung mit SBasic ganz erheblich vereinfacht. Mit »sbshell.bas« (Listing 6) wählen Sie alle benötigten Funktionen durch die Cursortasten aus (Bild 2). Die Benutzeroberfläche von SBasic stellt die Funktionen »Datei«, »Edit«, »Compile«, »Start«, »GW-Basic«, »DOS« und »Exit« bereit. Wenn Sie zum Aufruf der Oberfläche die Stapeldatei »sb.bat« (Listing 7) verwenden, sind einige Parameter bereits vorbelegt.

SBasic kann mit SB-Shell eine zeitgemäße Benutzeroberfläche vorweisen

In der Kommandozeile können Sie hinter SBshell eine Reihe von Steueranweisungen angeben. So können durch »/SUFFIX=« auch andere Dateiendungen als »bas« als Vorgabe dienen. Nach »/EDITOR=« muß der gewünschte Editor oder die Textverarbeitung stehen, mit der Sie Ihre SBasic-Programme bearbeiten wollen. Sofern dieses Programm nicht durch den DOS-Befehl »path« von überall aus erreichbar ist, können Sie vor dem Namen des Editors auch noch einen kompletten Pfad angeben. Gleiches gilt auch für die Schalter »/SBASIC=« und »/GWBASIC=«, die der Oberfläche mitteilen, wo sich das SBasic- und das GW-Basic-Verzeichnis befinden. Der Start der Oberfläche erfolgt in diesem Fall durch

```
sb <Dateiname>
```

Der angegebene Name ist der eines Basic-Programms ohne Erweiterung, Sie sehen ihn nach dem Start der Benutzeroberfläche zur Kontrolle in der rechten oberen Ecke des Bildschirms. Möchten Sie ein anderes Programm bearbeiten, so aktivieren Sie den Menüpunkt »Datei«, indem Sie den Menübalken mit den Cursortasten auf diesen Menüpunkt positionieren und die Eingabe-Taste drücken. Direkt unterhalb des Menüpunktes »Datei« erscheint nun ein Auswahlfilter, das Sie mit [Eingabe] übernehmen oder verändern können. Der gewünschte Dateiname läßt sich nun mit Hilfe der Cursortasten auswählen, [Eingabe] legt die gewünschte Datei fest. Die Taste [Esc] kehrt jeweils ohne Änderungen in die vorherige Ebene zurück.

»Edit« ruft den mit dem Schalter in »sb.bat« festgelegten Editor auf, um das Quellprogramm schnell und komfortabel zu überarbeiten. »Compile« startet SBasic mit dem unter »Datei« festgelegten Namen (Bild 3), »Start« führt danach ein so übersetztes Programm aus. Wer sein Programm in GW-Basic

testen möchte, wählt den Menüpunkt »GW-Basic« und drückt [Eingabe]. Beachten Sie, daß die Systemvariablen »command$(n)« von SBasic im GW-Basic-Interpreter unbelegt sind. Verwenden Sie sie im Programm, so müssen Sie den Variablen im Direktmodus sinnvolle Werte zuweisen und das Programm anschließend mit dem Goto-Befehl starten. Zum Verlassen der Oberfläche dient »Exit«, die Taste [Esc] erfüllt jedoch den gleichen Zweck. Mit dem Menüpunkt »DOS« springen Sie zur Ausführung von Dienstprogrammen auf die MS-DOS-Befehlsebene. Von dort kehren Sie durch

```
exit
```

gefolgt von [Eingabe] zurück in die Oberfläche.

Der Präprozessor verleiht SBasic die Fähigkeit, Sprungmarken zu verarbeiten

Herkömmlichen GW-Basic-Quelltexten mangelt es stets an Übersichtlichkeit. Jeder Zeile muß eine Nummer voranstehen, damit der Interpreter sie akzeptiert. Muß man Zeilen einfügen, löschen oder überschreiben, kann es unter Umständen zu Problemen kommen, wenn die Abstände der Zeilennummern zu gering sind. Ein neu numeriertes Programm verwirrt dann nur noch zusätzlich, weil man sich die alten Zeilennummern bereits eingeprägt hat. Professionelle Basic-Compiler sind deshalb schon lange dazu übergegangen, auch Quelltexte zu übersetzen, deren Verzweigungen sich auf Marken (Label) beziehen. Aus diesem Grund wurde für den SBasic-Compiler ein schneller Präprozessor entwickelt, der ebenfalls die Verwendung von Sprungmarken erlaubt. Der Präprozessor »sbasic.pre« (Listing 4) wandelt Quelltexte, deren Verzweigungen sich auf Sprungmarken beziehen, in das für GW-Basic verträgliche Format mit Zeilennummern um. Dazu ein Beispiel:

```
ende: cls:end
```

Diese Programmzeile vereinbart die Sprungmarke »ende« an dieser Position in einem Programm. Die Anweisung

```
goto ende
```

bewirkt dadurch eine unbedingte Verzweigung zur Marke »ende«, an der dann der Bildschirm gelöscht und das Programm beendet wird. Die Marke muß sich übrigens nicht unbedingt am Zeilenanfang befinden, ihr dürfen aber nur Leerzeichen voranstehen. Der Then-Befehl bezieht sich unter Umständen ebenfalls auf eine Marke, weshalb Sie keine Marken verwenden dürfen, die wie ein alleinstehender Befehl lauten, also zum Beispiel »print« oder »wend«.

Bei der Entwicklung der Version 2.0 von SBasic stand die Steigerung der Arbeitsgeschwindigkeit im Vordergrund

Zielsetzung bei der Entwicklung der Version 2.0 von SBasic war es, einen Präprozessor für die Umwandlung von ASCII-/Label-Basic nach GW-Basic zu implementieren, der sich gegenüber einem vergleichbaren System (siehe DOS Extra 7/89) durch sehr hohe Leistungsdaten auszeichnet und SBasic professionellere Anwendungsbereiche erschließt. Bei relativ geringfügigen Änderungen des Hauptmoduls »sbasic.bas« wurde ein getrenntes Modul »sbasic.pre« neben der Laufzeitbibliothek »sbasic.bin« geschaffen, um bei eventuell vorhandener Laufzeitbibliothek diese nicht noch einmal neu eingeben zu müssen.
SBasic Version 2.0 ist zunächst voll kompatibel zu der Vorversion, das heißt, es verarbeitet eine Eingabe im tokenisierten GW-Basic-Format. Darüber hinaus können Basic-Programmdateien im ASCII-Format ohne Zwischenschaltung des GW-Basic-Interpreters direkt verarbeitet werden. Die Umwandlung in das GW-Basic-Format erfolgt mit einer mehr als 30fachen Geschwindigkeit gegenüber dem in DOS-Extra 7/89 vorgestellten System.

Schließlich lassen sich Zeilennummern und/oder Label als Sprungmarken nutzen. Bei Eingabe von Zeilennummern werden anschließende Zeilen ohne Zeilennummern in Einerschritten fortgezählt. Ein Label besteht aus alphanumerischen Zeichen einschließlich ».« und »_« mit beliebiger Länge. Die Endung mit Doppelpunkt ist nicht unbedingt erforderlich, sofern in der Zeile kein weiterer Befehl folgt, andererseits ist nach dem Doppelpunkt auch keine weitere Leerstelle notwendig. Eine Fehlermeldung erfolgt nicht nur bei fehlenden, sondern auch bei mehrfach definierten Sprungmarken. Die Umwandlung erfolgt gegenüber dem Präprozessor aus der DOS Extra 7/89 mit über 250facher Geschwindigkeit.

Viele kleine Verbesserungen erleichtern die Arbeit mit dem neuen Compiler

Wird SBasic mit einem zweiten Parameter »/GWB« beziehungsweise »/gwb« aufgerufen, so wird keine Exe-Datei, sondern eine GW-Basic-Datei mit der Dateiendung ».gwb« generiert. In dieser werden vorhandene Label mit dem Kommentar »'*[LABELNAME]*« dokumentiert. Dadurch können Sie eine strukturierte ASCII-/Label-Programmdatei zunächst mit dem GW-Basic-Interpreter austesten und dann in der Ausgangsdatei die entsprechenden Korrekturen vornehmen. Die zuvor genannte extrem kurze Umwandlungszeit erlaubt hierbei ein zügiges Vorgehen. Auch die eigentliche Kompilierung und Einbindung der Laufzeitmodule erfolgt jetzt schneller, da sie ohne mehrfaches Lesen von Dateien im Hauptspeicher abläuft. Die Begrenzung der Eingabedatei auf rund 60 KByte kann hingenommen werden, da diese im Bereich der Anwendungen von SBasic praktisch keine Bedeutung hat.

Die Fehlermeldungen, sowohl des Präprozessors als auch des Hauptprogramms, haben dasselbe Format (auch bei »Stop« beziehungsweise »Break«). Neben der Zeilennummer im Quelltext und der jeweils fortgezählten (logischen) Zeilennummer gibt SBasic auch den fehlerhaften Zeilentext aus, sofern der Quellcode im ASCII-Format vorliegt. Die ASCII-Eingabe verarbeitet nun auch Tabulatoren. Entsprechend den Konventionen von GW-Basic dürfen Sie die Anweisungsform »IF ... THEN GOTO« auch verkürzt als »IF ... GOTO« schreiben. Statt »GOTO« ist nun auch die Form »GO TO« erlaubt.

Wenn SBasic die angegebene Quelldatei nicht findet, legt es diese nicht mehr als Datei mit Länge Null im Verzeichnis an. Das Exe-Compilat wurde grundsätzlich durch Veränderung der Programmkopfinformationen um 480 Byte verkleinert. Sofern SBasic das als Freeware weit verbreitete Komprimierungsprogramm »lzexe.exe« für Exe-Dateien von Fabrice Bellard im gültigen Verzeichnis oder im MS-DOS-Suchpfad findet, erfolgt automatisch der Aufruf, wodurch sich das »exe«-Compilat deutlich verkleinert, bei »sbasic.exe« beispielsweise um 46 Prozent.

Die Module »sbasic.pre« und »sbasic.bin« müssen im gleichen Verzeichnis wie das Hauptprogramm »sbasic.exe« stehen. Dieses sollte im MS-DOS-Suchpfad, der über die Systemvariable »path« gesteuert wird, erreichbar sein. Den Dateinamen von SBasic, »sbasic«, können Sie beliebig verändern, aber »sbasic.pre« und »sbasic.bin« müssen immer denselben Namen haben. Mit SBasic programmieren Sie auf einfachste Weise, nämlich in Ihrer gewohnten Umgebung und mit der Sprache, die Sie beherrschen Anwendungen, die in punkto Ablaufgeschwindigkeit den Vergleich mit professionellen Programmen nicht zu scheuen brauchen. Wir wünschen viel Spaß mit dem verbesserten SBasic.

(Peter Knappe/Wolfgang Lorenz/Herbert Buckel/wn)

```
[EF5]    1: 4D 5A 4D Ø1 56 ØØ ØØ ØØ Ø2 ØØ BC 31 BC 31 B3 ØA
[325]    2: ØØ 1Ø ØØ ØØ 38 14 ØØ ØØ 1C ØØ ØØ ØØ ØØ ØØ ØØ ØØ
[WS5]    3: 8C DD 8C CA Ø3 C2 8E D8 8E CØ 23 DB 74 11 33 CØ
[5S5]    4: 33 FF B9 Ø8 ØØ F3 AB 8C CØ 4Ø 8E CØ 4B 75 EF 89
[1F5]    5: 2E 2A ØØ C6 Ø6 ØØ ØØ FF C6 Ø6 Ø4 ØØ Ø4 C6 Ø6 Ø9
[7M5]    6: ØØ Ø7 B4 ØF CD 1Ø FE CC 88 26 ØB ØØ C6 Ø6 ØD ØØ
[KE5]    7: 17 8C 1E 19 ØØ B4 Ø3 CD 1Ø 89 ØE ØF ØØ C7 Ø6 15
[RR5]    8: ØØ FF FF 8Ø FE 18 72 ØB B8 ØA ØE CD 1Ø B6 17 B4
[2N5]    9: Ø2 CD 1Ø B9 ØØ 1F B4 Ø1 CD 1Ø BA Ø7 Ø7 B4 ØF CD
[CB5]   1Ø: 1Ø 3C Ø7 75 Ø3 BA ØC ØB 89 16 24 ØØ C7 Ø6 26 ØØ
[9H5]   11: A4 8E C7 Ø6 28 ØØ 1F 7F C7 Ø6 2C ØØ 3Ø Ø4 C7 Ø6
[K45]   12: 2E ØØ 78 Ø7 C3 ØØ ØØ ØØ ØØ ØØ ØØ ØØ ØØ ØØ ØØ ØØ
[ØZ4]   13: ØØ ØØ ØØ ØØ ØØ ØØ ØØ ØØ ØØ ØØ ØØ ØØ ØØ ØØ ØØ ØØ
[ØZ4]   14: ØØ ØØ ØØ ØØ ØØ ØØ ØØ ØØ ØØ ØØ ØØ ØØ ØØ ØØ ØØ ØØ
[ØZ4]   15: ØØ ØØ ØØ ØØ ØØ ØØ ØØ ØØ ØØ ØØ ØØ ØØ ØØ ØØ ØØ ØØ
[ØZ4]   16: ØØ ØØ ØØ ØØ ØØ ØØ ØØ ØØ ØØ ØØ ØØ ØØ ØØ ØØ ØØ ØØ
[ØZ4]   17: ØØ ØØ ØØ ØØ ØØ ØØ ØØ ØØ ØØ ØØ ØØ ØØ ØØ ØØ ØØ ØØ
[ØZ4]   18: ØØ ØØ ØØ ØØ ØØ ØØ ØØ ØØ ØØ ØØ ØØ ØØ ØØ ØØ ØØ ØØ
[ØZ4]   19: ØØ ØØ ØØ ØØ ØØ ØØ ØØ ØØ ØØ ØØ ØØ ØØ ØØ ØØ ØØ ØØ
[ØZ4]   2Ø: ØØ ØØ ØØ ØØ ØØ ØØ ØØ ØØ ØØ ØØ ØØ ØØ ØØ ØØ ØØ ØØ
[XS5]   21: ØØ 5Ø 52 BA FD Ø3 EC DØ E8 73 1F B2 F8 EC 53 2E
[PM5]   22: 8A 1E 1D Ø1 43 83 E3 3F 2E 3A 1E 1E Ø1 74 ØA 2E
[PH5]   23: 88 87 95 ØØ 2E 88 1E 1D Ø1 5B BØ 64 E6 2Ø 5A 58
[3W5]   24: CF 5Ø 52 BA FD Ø2 EC DØ E8 73 1F B2 F8 EC 53 2E
[7K5]   25: 8A 1E 1F Ø1 43 83 E3 3F 2E 3A 1E 2Ø Ø1 74 ØA 2E
[6L5]   26: 88 87 D5 ØØ 2E 88 1E 1F Ø1 5B BØ 63 E6 2Ø 5A 58
[8M5]   27: CF 3C ØD 74 Ø9 3C ØA 74 Ø9 FE 47 Ø4 EB Ø4 C6 47
[LM5]   28: Ø4 ØØ 8A 1F B9 Ø1 ØØ A2 ØE ØØ BA ØE ØØ B4 4Ø CD
[NS5]   29: 21 5B C3 53 8A ØF 8Ø F9 FD 72 D6 5Ø 8Ø F9 FE 75
[WV5]   3Ø: ØE BA FD Ø3 EC A8 2Ø 74 FB B2 F8 58 EE 5B C3 8Ø
[1Z5]   31: F9 FD 75 ØE BA FD Ø2 EC A8 2Ø 74 FB B2 F8 58 EE
[ZP5]   32: 5B C3 B4 ØF CD 1Ø B4 Ø3 CD 1Ø 59 8A 1E Ø9 ØØ 8Ø
[ØP5]   33: F9 Ø7 75 Ø7 B8 Ø7 ØE CD 1Ø 5B C3 8Ø F9 Ø9 75 2E
[6L5]   34: E8 CC ØØ B4 Ø3 CD 1Ø 8A C2 24 Ø7 B9 Ø8 ØØ 2A C8
[ND5]   35: B8 2Ø Ø9 CD 1Ø C6 Ø6 ØA ØØ ØØ Ø2 D1 3A 16 ØB ØØ
[KS5]   36: 76 Ø6 FE CA FE ØE ØA ØØ B4 Ø2 CD 1Ø 5B C3 8Ø F9
[FL5]   37: ØA 74 Ø5 8Ø F9 ØD 75 Ø5 E8 A1 ØØ 5B C3 8Ø F9 ØB
[VC5]   38: 75 11 B2 ØØ 8A 36 ØC ØØ B4 Ø2 CD 1Ø C6 Ø6 ØA ØØ
[RN5]   39: ØØ 5B C3 8Ø F9 ØC 75 Ø8 B8 ØØ Ø6 E8 9C ØØ EB E2
[8H5]   4Ø: 8Ø F9 1C 75 12 B4 Ø3 CD 1Ø 3A 16 ØB ØØ 74 Ø4 FE
[LH5]   41: C2 EB 15 B2 ØØ EB Ø9 8Ø F9 1F 75 17 B4 Ø3 CD 1Ø
[4H5]   42: 3A 36 ØD ØØ 74 ØB FE C6 B4 Ø2 CD 1Ø C6 Ø6 ØA ØØ
[PG5]   43: ØØ 5B C3 8Ø F9 1D 75 13 B4 Ø3 CD 1Ø 8Ø FA ØØ 74
[ØS5]   44: Ø4 FE CA EB E3 8A 16 ØB ØØ EB Ø9 8Ø F9 1E 75 ØE
[FU5]   45: B4 Ø3 CD 1Ø 3A 36 ØC ØØ 74 D7 FE CE EB CA E8 ØE
[5M5]   46: ØØ 8A C1 B4 Ø9 B9 Ø1 ØØ CD 1Ø E8 46 ØØ 5B C3 3A
[1D5]   47: 16 ØB ØØ 75 3D 8Ø 3E ØA ØØ ØØ 74 36 51 B2 ØØ 3A
[FL5]   48: 36 ØD ØØ 75 Ø8 B8 Ø1 Ø6 E8 ØF ØØ EB Ø2 FE C6 B4
[GF5]   49: Ø2 CD 1Ø C6 Ø6 ØA ØØ ØØ 59 C3 53 51 52 8A FB B1
[DG5]   5Ø: ØØ 8A 2E ØC ØØ 8A 16 ØB ØØ 8A 36 ØD ØØ CD 1Ø 5A
[2P5]   51: 59 5B C3 B4 Ø3 CD 1Ø 3A 16 ØB ØØ 74 ØC FE C2 B4
[9G5]   52: Ø2 CD 1Ø C6 Ø6 ØA ØØ ØØ C3 C6 Ø6 ØA ØØ FF C3 BØ
[LK5]   53: 1A 8B 3E 17 ØØ 23 FF 74 ØF 2E 8A Ø5 47 3C 1A 75
[7M5]   54: Ø3 2E 8B 3D 89 3E 17 ØØ C3 23 DB 78 E2 8A Ø7 3C
[VN5]   55: FF 74 71 53 3C FE 75 2Ø 2E 8A 1E 1E Ø1 2E 3A 1E
[GP5]   56: 1D Ø1 74 F4 FE C3 8Ø E3 3F 2E 8A 87 95 ØØ 2E 88
[SL5]   57: 1E 1E Ø1 5B 88 7F 1Ø C3 3C FD 75 2Ø 2E 8A 1E 2Ø
[3S5]   58: Ø1 2E 3A 1E 1F Ø1 74 F4 FE C3 8Ø E3 3F 2E 8A 87
[YH5]   59: D5 ØØ 2E 88 1E 2Ø Ø1 5B 88 7F 1Ø C3 5B 8A 4F 1Ø
[5L5]   6Ø: 51 8Ø F9 1A 74 17 53 8A D8 B9 Ø1 ØØ BA ØE ØØ B4
[3G5]   61: 3F CD 21 5B 72 Ø7 23 CØ AØ ØE ØØ 75 Ø2 BØ 1A 88
[BL5]   62: 47 1Ø 58 C3 AØ ØE ØØ 3C FF 75 Ø5 B4 Ø8 CD 21 C3
[AL5]   63: 83 3E 15 ØØ FF 74 28 3C ØØ 75 Ø3 BØ 1A C3 FE ØE
[2K5]   64: ØE ØØ 53 B4 ØF CD 1Ø B4 Ø8 CD 1Ø 5Ø B4 Ø3 CD 1Ø
[NK5]   65: 3A 16 ØB ØØ 73 Ø6 FE C2 B4 Ø2 CD 1Ø 58 5B C3 53
[AL5]   66: B4 ØF CD 1Ø B4 Ø3 CD 1Ø 89 ØE 15 ØØ 52 33 FF 3A
[BE5]   67: 16 ØB ØØ 76 Ø4 8A 16 ØB ØØ B4 Ø2 CD 1Ø 8B ØE 24
[UE5]   68: ØØ 23 FF 74 Ø2 B5 Ø4 B4 Ø1 CD 1Ø B4 Ø8 CD 21 5Ø
[2F5]   69: B9 ØØ 1F B4 Ø1 CD 1Ø 58 3C 2Ø 72 44 8A 1E Ø9 ØØ
[ER5]   7Ø: 23 FF 74 31 5Ø 8A CA 8A 16 ØB ØØ FE CA 78 1F 3A
[VN5]   71: D1 72 1B B4 Ø2 CD 1Ø B4 Ø8 CD 1Ø FE C2 B4 Ø2 CD
[UR5]   72: 1Ø 51 B9 Ø1 ØØ B4 Ø9 CD 1Ø 59 FE CA EB DD 8A D1
[VN5]   73: B4 Ø2 CD 1Ø 58 B9 Ø1 ØØ B4 Ø9 CD 1Ø FE C2 EB 8F
[PM5]   74: 3C ØØ 75 4D B4 Ø8 CD 21 3C 4B 75 ØB 58 5Ø 3A DØ
[LØ6]   75: 76 Ø2 FE CA E9 78 FF 3C 4D 75 Ø4 FE C2 EB F5 3C
[2N5]   76: 47 75 Ø4 5A 52 EB ED 3C 4F 75 1A 8A 16 ØB ØØ 59
[XM5]   77: 51 B4 Ø2 CD 1Ø B4 Ø8 CD 1Ø 3C 2Ø 75 DE 3A D1 74
[GX5]   78: D3 FE CA EB EC 3C 53 74 14 3C 52 75 C7 F7 D7 EB
[PR5]   79: C3 3C Ø8 75 3E 59 51 3A D1 76 35 FE CA 52 8A 1E
[SL5]   8Ø: Ø9 ØØ 3A 16 ØB ØØ 73 1B FE C2 B4 Ø2 CD 1Ø B4 Ø8
[VT5]   81: CD 1Ø FE CA B4 Ø2 CD 1Ø B9 Ø1 ØØ B4 Ø9 CD 1Ø FE
[9M5]   82: C2 EB DF B4 Ø2 CD 1Ø B8 2Ø Ø9 B9 Ø1 ØØ CD 1Ø 5A
[8Y5]   83: E9 FC FE 3C ØD 74 Ø7 3C ØA 74 F5 E9 1E FF 59 8A
[8K5]   84: 16 ØB ØØ B4 Ø2 CD 1Ø B4 Ø8 CD 1Ø 3C 2Ø 75 Ø8 FE
[9U5]   85: CA 78 Ø4 3A D1 73 EC FE C2 2A D1 88 16 ØE ØØ 8A
[ØW5]   86: D1 B4 Ø2 CD 1Ø 5B E9 7B FE 8Ø 3F FF 75 13 BØ ØD
[HP5]   87: E8 6Ø FC 8B ØE 15 ØØ B4 Ø1 CD 1Ø C7 Ø6 15 ØØ FF
[XG5]   88: FF C2 Ø2 ØØ 8B EC 83 C5 Ø4 8A 8E ØØ Ø1 8A 6E ØØ
[8M5]   89: 33 FF E3 1C 8Ø F9 ØØ 74 14 8Ø FD ØØ 74 15 47 FE
[LM5]   9Ø: C9 FE CD 8A 83 ØØ Ø1 3A Ø3 74 E7 77 Ø6 BØ Ø2 C3
[HL5]   91: BØ Ø1 C3 BØ Ø4 C3 8B EC B2 ØØ 8A 76 Ø6 8B 76 Ø7
[ZN5]   92: 8A 5E Ø9 BØ ØØ 8A 66 Ø2 8B 7E Ø3 8A 7E Ø5 3A DF
[PT5]   93: 73 Ø5 92 87 F7 86 DF 8Ø FF ØØ 74 6A 8A CB 2A CF
[ZL5]   94: 8Ø F9 18 77 61 B5 ØØ 56 33 F7 9C 8B 76 FE 81 CE
[ZR5]   95: ØØ 8Ø 81 CF ØØ 8Ø E3 Ø6 D1 EF D1 D8 E2 FA 9D 78
[YU5]   96: ØE Ø3 DØ 13 F7 73 39 D1 DE D1 DA FE C3 EB 31 2B
[5M5]   97: DØ 1B F7 72 Ø8 75 14 23 D2 75 1Ø EB 1A F7 D2 83
[ØK5]   98: C2 Ø1 F7 D6 83 D6 ØØ 8Ø 76 FF 8Ø 23 F6 78 11 D1
[YU5]   99: E2 D1 D6 FE CB 75 F4 33 F6 33 D2 B3 ØØ 58 EB Ø6
[8M5]   1ØØ: 58 ØD FF 7F 23 FØ 88 76 Ø6 89 76 Ø7 88 5E Ø9 C2
[HG5]   1Ø1: Ø4 ØØ 8B EC 8Ø 7E Ø9 ØØ 74 5B 8Ø 7E Ø5 ØØ 74 55
[GT5]   1Ø2: 8A 5E Ø2 8A 7E Ø6 8A C3 F6 E7 8B F8 8B 76 Ø3 8B
[BU5]   1Ø3: 6E Ø7 8B C6 33 C5 ØD FF 7F 5Ø 81 CE ØØ 8Ø 81 CD
[9R5]   1Ø4: ØØ 8Ø 8B C6 F7 E5 8B C8 87 F2 BØ ØØ 8A E7 F7 E2
[FM5]   1Ø5: Ø3 F8 13 CA 83 D6 ØØ BØ ØØ 8A E3 F7 E5 Ø3 F8 13
[LT5]   1Ø6: CA 83 D6 ØØ 58 8B EC 8A 5E Ø5 8A 7E Ø9 81 F3 8Ø
[ØF5]   1Ø7: 8Ø Ø2 DF 71 ØB 33 CØ 89 46 Ø6 89 46 Ø8 C2 Ø4 ØØ
[BP5]   1Ø8: 8Ø F3 8Ø 23 F6 78 ØA D1 E7 D1 D1 D1 D6 FE CB 75
[CR5]   1Ø9: F2 81 C1 8Ø ØØ 83 D6 ØØ 73 Ø6 D1 DE D1 D9 FE C3
[5L5]   11Ø: 23 FØ 88 6E Ø6 89 76 Ø7 88 5E Ø9 C2 Ø4 ØØ 8B EC
[BF5]   111: 8Ø 7E Ø9 ØØ 74 76 8Ø 7E Ø5 ØØ 74 73 8B 76 Ø6 8A
[UL5]   112: 5E Ø8 B7 ØØ 8B 7E Ø2 8A 56 Ø4 B6 ØØ 8A C3 32 C2
```

```
[5M5]   113: ØC 7F 5Ø 8Ø CB 8Ø 8Ø CA 8Ø B1 19 33 ED BØ ØØ 2B
[8U5]   114: F7 1B DA 73 Ø4 Ø3 F7 13 DA F5 D1 D5 DØ DØ D1 E6
[YS5]   115: D1 D3 FE C9 22 CØ 79 E7 2B F7 1B DA F5 83 D5 ØØ
[CT5]   116: 14 ØØ 73 Ø6 DØ D8 D1 DD FE C1 5B 8B D5 8B EC 22
[KE5]   117: C3 89 56 Ø6 88 46 Ø8 8A 46 Ø9 8A 66 Ø5 35 8Ø 8Ø
[1A5]   118: Ø2 C1 2A C4 7Ø Ø9 35 8Ø ØØ 88 46 Ø9 C2 Ø4 ØØ 33
[ØH5]   119: CØ 89 46 Ø6 89 46 Ø8 C2 Ø4 ØØ 8B EC 8A 5E Ø5 22
[CE5]   12Ø: DB 74 ØF BØ 8Ø 84 46 Ø4 74 Ø9 33 CØ 89 46 Ø2 89
[BF5]   121: 46 Ø4 C3 88 46 Ø5 81 7E Ø4 35 8Ø 75 Ø5 81 7E Ø2
[MV5]   122: F4 Ø4 73 Ø5 FE 46 Ø5 FE CB 32 FF 53 FF 76 Ø4 FF
[NS5]   123: 76 Ø2 B8 8Ø 81 5Ø 33 CØ 5Ø E8 FA FD FF 76 1Ø FF
[XT5]   124: 76 ØE B8 ØØ 81 5Ø 33 CØ 5Ø E8 EA FD E8 1F FF 8B
[DH5]   125: 46 Ø8 89 46 1Ø 5Ø 8B 46 Ø6 89 46 ØE 5Ø E8 72 FE
[CR5]   126: 8B 46 Ø8 8B 5E ØA 53 5Ø 53 5Ø E8 65 FE B8 5E 7F
[SU5]   127: 5Ø B8 4D 57 5Ø E8 BE FD FF 76 ØC FF 76 ØA E8 51
[7V5]   128: FE B8 13 8Ø 5Ø B8 Ø8 9B 5Ø E8 AA FD FF 76 ØC FF
[5S5]   129: 76 ØA E8 3D FE B8 76 8Ø 5Ø B8 93 38 5Ø E8 96 FD
[AW5]   13Ø: E8 2F FE B8 38 82 5Ø B8 3B AA 5Ø E8 88 FD FF 76
[BN5]   131: 1Ø FF 76 ØE E8 1B FE 8A 46 ØA B4 ØØ 34 8Ø 74 17
[2R5]   132: 8A D8 79 Ø2 F6 D8 B4 88 22 CØ 78 Ø6 DØ EØ FE CC
[9S5]   133: EB F6 8Ø CB 7F 22 C3 5Ø 33 CØ 5Ø E8 58 FD B8 31
[VP5]   134: 8Ø 5Ø B8 18 72 5Ø E8 E9 FD 8F 46 ØE 8F 46 1Ø 58
[AR5]   135: C3 8B EC B2 ØØ 8B 76 Ø6 8B 46 Ø8 8B 7E Ø2 8B 5E
[1S5]   136: Ø4 8A C8 8A EB 32 E9 78 19 ØC 8Ø 8Ø CB 8Ø 3B C3
[7K5]   137: 75 Ø4 3B F7 74 ØB 77 Ø3 8Ø F1 8Ø DØ E1 72 Ø7 FE
[ØW5]   138: C2 C3 DØ EØ 73 F9 FE CA C3 53 8A 1F 8Ø FB FF 74
[BU5]   139: 4F 8Ø FB Ø5 72 4A 33 CØ 8E CØ 8Ø FB FE 75 1C BA
[XK5]   14Ø: F9 Ø3 EE E4 21 ØC 1Ø E6 21 2E A1 15 Ø1 26 A3 3Ø
[7K5]   141: ØØ 2E A1 17 Ø1 26 A3 32 ØØ EB 25 8Ø FB FD 75 1C
[1L5]   142: BA F9 Ø2 EE E4 21 ØC Ø8 E6 21 2E A1 19 Ø1 26 A3
[SL5]   143: 2C ØØ 2E A1 1B Ø1 26 A3 3E ØØ EB Ø4 B4 3E CD 21
[FN5]   144: 5B C3 Ø7 B8 38 81 5Ø B8 3B AA 5Ø E8 44 FD 33 D2
[ØN5]   145: 33 CØ 8B 5E Ø8 8Ø FF 81 72 24 8A E3 8Ø CC 8Ø 8A
[2N5]   146: 46 Ø7 8A 56 Ø6 8Ø FF 9Ø 73 Ø9 D1 E8 8Ø D6 ØØ FE
[WS5]   147: C7 EB F2 22 DB 79 Ø7 F7 D8 23 D2 74 Ø1 48 5Ø FF
[4K5]   148: 76 Ø8 FF 76 Ø6 B7 ØØ 23 CØ 74 ØC 79 Ø2 F7 D8 B7
[7V5]   149: 9Ø FE CF D1 EØ 79 FA 8Ø E3 8Ø 8Ø E4 7F ØA E3 8A
[4P5]   15Ø: DC 8A EØ BØ ØØ 8Ø F3 8Ø 53 5Ø E8 49 FC B8 34 71
[9N5]   151: 5Ø B8 3C 58 5Ø FF 76 Ø8 FF 76 Ø6 E8 D4 FC B8 16
[BX5]   152: 74 5Ø B8 B7 7E 5Ø E8 2D FC FF 76 ØC FF 76 ØA E8
[2Ø6]   153: CØ FC B8 2F 77 5Ø B8 DF EE 5Ø E8 19 FC FF 76 ØC
[HU5]   154: FF 76 ØA E8 AC FC B8 1D 7A 5Ø B8 17 84 5Ø E8 Ø5
[9W5]   155: FC FF 76 ØC FF 76 ØA E8 98 FC B8 63 7C 5Ø B8 59
[ØW5]   156: 59 5Ø E8 F1 FB FF 76 ØC FF 76 ØA E8 84 FC B8 75
[X16]   157: 7E 5Ø B8 E9 FD 5Ø E8 DD FB FF 76 ØC FF 76 ØA E8
[LR5]   158: 7Ø FC B8 31 8Ø 5Ø B8 18 72 5Ø E8 C9 FB FF 76 ØC
[7S5]   159: FF 76 ØA E8 5C FC B8 ØØ 81 5Ø 33 CØ 5Ø E8 B6 FB
[NH5]   16Ø: 58 5B 59 59 59 Ø2 F9 75 Ø4 33 CØ 33 DB 89 46 1Ø
[3N5]   161: 89 5E 12 Ø6 C3 8B EC 8B 46 Ø4 8Ø FC 98 73 5Ø 22
[NV5]   162: CØ 79 1C FF 76 Ø4 FF 76 Ø2 B8 FF 8Ø 5Ø B8 FF FF
[6P5]   163: 5Ø E8 82 FB 8B EC 83 C5 Ø4 8F 46 Ø4 2B 04 B8
[3T5]   164: FF FF B3 FF B1 9Ø 2A 4E Ø5 73 Ø7 8Ø C1 Ø8 D2 EØ
[PK5]   165: EB 17 BØ ØØ 8Ø E9 Ø8 73 Ø7 8Ø C1 Ø8 D2 E4 EB Ø9
[TE5]   166: B4 ØØ 8Ø F9 Ø8 73 Ø9 D2 E3 21 46 Ø2 2Ø 5E Ø4 C3
[ZF5]   167: 89 46 Ø2 89 46 Ø4 C3 Ø7 B8 49 81 5Ø B8 DA ØF 5Ø
[HS5]   168: E8 33 FB Ø6 Ø7 B8 22 7E 5Ø B8 81 F9 5Ø E8 C2 FB
[7U5]   169: 8B EC FF 76 Ø2 FF 76 ØØ E8 7A FF 8B EC 8Ø 7E Ø3
[2H5]   17Ø: ØØ 74 Ø4 8Ø 76 Ø2 8Ø E8 ØC FB 8B EC 8Ø 7E Ø3 ØØ
[XK5]   171: 74 Ø4 8Ø 76 Ø2 8Ø B8 ØØ 7F 5Ø 33 CØ 5Ø E8 F6 FA
[SH5]   172: 8B EC F6 46 Ø2 8Ø 74 19 B8 ØØ 8Ø 5Ø 33 CØ 5Ø E8
[YP5]   173: E4 FA 8B EC 8Ø 66 Ø2 7F B8 8Ø 7F 5Ø 33 CØ 5Ø EB
[BA5]   174: 11 8Ø 7E Ø3 ØØ 74 Ø4 8Ø 76 Ø2 8Ø B8 ØØ 7F 5Ø 33
[VU5]   175: CØ 5Ø E8 C1 FA 8B EC FF 76 Ø2 FF 76 ØØ FF 76 Ø2
[ØX5]   176: FF 76 ØØ E8 4C FB B8 E6 84 5Ø B8 2D 1A 5Ø 8B EC
[8T5]   177: FF 76 Ø6 FF 76 Ø4 E8 39 FB B8 28 86 5Ø B8 FB Ø7
[7V5]   178: 5Ø E8 92 FA 8B EC FF 76 Ø6 FF 76 Ø4 E8 23 FB B8
[ØX5]   179: 99 87 5Ø B8 99 68 5Ø E8 7C FA 8B EC FF 76 Ø6 FF
[TR5]   18Ø: 76 Ø4 E8 ØD FB B8 23 87 5Ø B8 DF 35 5Ø E8 66 FA
[PU5]   181: 8B EC FF 76 Ø6 FF 76 Ø4 E8 F7 FA B8 A5 86 5Ø B8
[6X5]   182: E7 5D 5Ø E8 5Ø FA 8B EC FF 76 Ø6 FF 76 Ø4 E8 E1
[GT5]   183: FA B8 49 83 5Ø B8 DA ØF 5Ø E8 3A FA 8B EC 8F 46
[NP5]   184: Ø4 8F 46 Ø6 E8 CB FA Ø6 C3 BB Ø3 ØØ E8 1A FD 4B
[RV5]   185: 75 FA 8B ØE ØF ØØ B4 Ø1 CD 1Ø B4 4C CD 21 8B EC
[ME5]   186: 33 D2 8B 46 Ø4 3B 46 Ø2 7E Ø1 4A 89 56 Ø4 C2 Ø2
[MH5]   187: ØØ 8B EC 33 D2 8B 46 Ø4 3B 46 Ø2 7D Ø1 4A 89 56
[7E5]   188: Ø4 C2 Ø2 ØØ 8B EC 33 D2 8B 46 Ø4 3B 46 Ø2 75 Ø1
[HL5]   189: 4A 89 56 Ø4 C2 Ø2 ØØ 8B EC 33 D2 8B 46 Ø4 3B 46
[5H5]   19Ø: Ø2 7C Ø1 4A 89 56 Ø4 C2 Ø2 ØØ 8B EC 33 D2 8B 46
[2G5]   191: Ø4 3B 46 Ø2 7F Ø1 4A 89 56 Ø4 C2 Ø2 ØØ 8B EC 33
[HB5]   192: D2 8B 46 Ø4 3B 46 Ø2 74 Ø1 4A 89 56 Ø4 C2 Ø2 ØØ
[2H5]   193: 8B EC 8B 46 Ø2 Ø1 46 Ø4 C2 Ø2 ØØ 8B EC 8B 46 Ø2
[HF5]   194: 29 46 Ø4 C2 Ø2 ØØ 8B EC 8B 46 Ø2 21 46 Ø4 C2 Ø2
[UG5]   195: ØØ 8B EC 8B 46 Ø2 Ø9 46 Ø4 C2 Ø2 ØØ 8B EC 8B 46
[8F5]   196: Ø4 F7 66 Ø2 89 46 Ø4 C2 Ø2 ØØ 8B EC 8B 46 Ø4 33
[5L5]   197: D2 F7 7E Ø2 89 46 Ø4 C2 Ø2 ØØ 8B EC F7 5E Ø2 C3
[1Y5]   198: 8B EC 8B 5E ØØ 8C DA C5 76 Ø2 81 EC FA ØØ 8B FC
[VZ5]   199: 8C DØ 8E CØ FC AC AA 8A C8 B5 ØØ F3 A4 8E DA FF
[TS5]   2ØØ: E3 8B F4 83 C6 Ø2 1E 8C DØ 8E D8 33 FF 36 8E 84
[TT5]   2Ø1: ØØ Ø1 FC AC AA 8A C8 B5 ØØ F3 A4 1F C2 Ø2 Ø1 8B
[TR5]   2Ø2: EC 8B 5E Ø2 BØ ØD BA A3 Ø1 FF D2 8Ø 3F FF 74 Ø7
[PR5]   2Ø3: BØ ØA BA A3 Ø1 FF D2 C3 8B EC 83 C5 Ø2 8B 9E ØØ
[FF5]   2Ø4: Ø1 8A 4E ØØ B5 ØØ E3 ØD 45 8A 46 ØØ 51 BA A3 Ø1
[PN5]   2Ø5: FF D2 59 E2 F3 C2 ØØ Ø1 81 EC FC ØØ 8B EC 8B 86
[DP5]   2Ø6: FE ØØ B3 2Ø 23 CØ 79 Ø4 F7 D8 B3 2D 88 5E Ø1 BB
[1M5]   2Ø7: ØA ØØ 33 C9 33 D2 F7 F3 8Ø CA 3Ø 52 23 CØ EØ F4
[4M5]   2Ø8: F7 D9 8A D9 BF Ø2 ØØ 58 88 Ø3 47 E2 FA C6 Ø3 2Ø
[2V5]   2Ø9: FE C3 88 5E ØØ FF A6 FC ØØ 33 DB BØ ØC BA A3 Ø1
[1V5]   21Ø: FF E2 8B F4 83 C6 Ø2 1E 8C DØ 8E D8 8E CØ 8A 8C
[XS5]   211: ØØ Ø1 B5 Ø1 8B F9 AC Ø3 FE B5 ØØ F7 D9 81 C1 FF
[TL5]   212: ØØ 3A C1 73 Ø2 8A C8 ØØ 8C FF ØØ F3 A4 1F C2 ØØ
[8R5]   213: Ø1 81 EC FC ØØ 8B EC BØ Ø1 8A A6 FE ØØ 89 46 ØØ
[VM5]   214: FF A6 FC ØØ 8B EC 33 D2 39 56 Ø2 74 Ø6 7F Ø3 4A
[9P5]   215: EB Ø1 42 89 56 Ø2 C3 8B EC B4 ØF CD 1Ø 8A 56 Ø2
[6E5]   216: 8A 76 Ø4 81 EA Ø1 Ø1 B4 Ø2 CD 1Ø C6 Ø6 ØA ØØ ØØ
[6X5]   217: C2 Ø4 ØØ 81 EC FE ØØ 8B EC FF B6 FE ØØ 8C DØ 8E
[AD5]   218: CØ 8D 7E Ø1 8B 9E Ø2 Ø1 8Ø 26 ØE ØØ 7F E8 54 ØØ
[8E5]   219: 74 49 3C 2Ø 74 F7 3C 22 75 21 E8 47 ØØ 74 3C 3C
```

Listing 1. »sbasic.exe« ist der Hexdump des Basic-Compilers. Geben Sie dieses Listing nur dann ein, wenn Sie keine frühere Version von SBasic besitzen.

```
[UT5]  22Ø: 22 74 ØD 57 2B FD 81 FF ØØ Ø1 5F 73 ED AA EB EA
[1H5]  221: E8 31 ØØ 74 26 3C 2C 75 F7 EB 2Ø 3C 2C 74 1Ø 57
[DR5]  222: 2B FD 81 FF ØØ Ø1 5F 73 Ø1 AA E8 17 ØØ 75 EC 4F
[FK5]  223: 3B FD 74 Ø6 26 8Ø 3D 2Ø 74 F5 47 8D 45 FF 2B C5
[SL5]  224: 88 46 ØØ C3 57 BA 39 Ø3 FF D2 5F 3C 1A 74 15 3C
[DM5]  225: ØD 75 11 8Ø 7F 1Ø ØA 75 Ø9 57 BA 39 Ø3 FF D2 5F
[6L5]  226: BØ ØD 3C ØD C3 8B EC 8A 46 Ø2 B4 ØØ 89 86 ØØ Ø1
[8M5]  227: C2 FE ØØ BA 54 Ø5 FF D2 33 D2 A8 Ø1 74 Ø1 4A 89
[WM5]  228: 96 FE Ø1 C2 FE Ø1 BA 54 Ø5 FF D2 33 D2 A8 Ø1 75
[ZV5]  229: Ø1 4A 89 96 FE Ø1 C2 FE Ø1 8B EC 8B 5E Ø2 BA 39
[6M5]  23Ø: Ø5 FF E2 8B EC 8A 8E Ø2 Ø1 8A 6E Ø2 8Ø FD ØØ 74
[LT5]  231: 2B 8B 9E Ø2 Ø2 8A C3 FE C8 Ø2 C5 72 1F 3A C1 77
[8L5]  232: 1B 8B F3 BF Ø1 ØØ 8A D5 46 47 8A 82 Ø1 Ø1 3A 43
[SW5]  233: Ø1 75 Ø6 FE CA 75 F1 EB Ø5 43 EB D9 33 DB 89 9E
[MG5]  234: Ø2 Ø2 C2 ØØ Ø2 8B EC 8B 46 Ø2 87 46 Ø4 83 C4 Ø4
[LS5]  235: FF EØ 8B EC 8E Ø6 19 ØØ 8B 7E Ø4 8A 46 Ø2 AA C2
[1K5]  236: Ø4 ØØ 8B EC 8E Ø6 19 ØØ 8B 7E Ø2 26 8A Ø5 B4 ØØ
[RK5]  237: 89 46 Ø2 C3 8B EC 8B 4E Ø2 33 CØ 8Ø F9 Ø1 72 Ø2
[BR5]  238: 8A C5 89 86 ØØ Ø1 C2 FE ØØ 8B EC 8A 46 Ø4 8A 66
[BL5]  239: Ø2 3A EØ 73 Ø2 8A C4 88 46 Ø4 C2 Ø2 ØØ 8B FC 83
[RW5]  24Ø: C7 Ø6 8C DA 8C DØ 8E D8 8E CØ 8A 5D FE 8Ø FB ØØ
[YN5]  241: 75 Ø2 B3 Ø1 B7 ØØ 8D 31 8A ØD B5 ØØ FE CB 2A CB
[8T5]  242: 73 Ø2 B1 ØØ 8A 45 FC 3A C8 76 Ø2 8A C8 8A C1 AA
[SS5]  243: F3 A4 8E DA C2 Ø4 ØØ 8B FC 83 C7 Ø4 8C DØ 8E CØ
[7M5]  244: 26 8A 8D FE ØØ B5 ØØ 26 8B 45 FE 3C ØØ 75 Ø2 B1
[7X5]  245: ØØ 8A C1 AA 8A C4 F3 AA C2 Ø2 ØØ 8B EC 8C C8 87
[UV5]  246: 46 Ø2 8C DB 8E C3 FF 36 19 ØØ 5Ø CB 8B EC 8B 7E
[ØG5]  247: ØØ 8B 46 Ø2 B3 ØØ 23 CØ 74 15 BB 9Ø 7F 79 Ø4 F7
[2W5]  248: D8 B7 FF 23 CØ 78 Ø6 D1 EØ FE CB EB F6 22 E7 C6
[UM5]  249: 46 ØØ ØØ 89 46 Ø1 88 5E Ø3 FF E7 81 EC FA ØØ 8B
[MS5]  25Ø: FC B2 ØØ 36 8A B5 FC ØØ 36 8B B5 FD ØØ 36 8A 9D
[UN5]  251: FF ØØ 8C DØ 8E CØ 47 BØ 2Ø 23 F6 79 Ø2 BØ 2D AA
[8R5]  252: 8Ø FB ØØ 75 14 BØ 3Ø AA 8B EC 8D 45 FF 2B C5 88
[ØN5]  253: 46 ØØ BØ 2Ø AA FF A6 FA ØØ 81 CE ØØ 8Ø 33 C9 8Ø
[2P5]  254: FB 81 72 Ø3 E9 D1 ØØ E8 7A ØØ FE C9 8Ø FB 8Ø 72
[9S5]  255: F6 81 C2 8Ø ØØ 83 D6 ØØ 73 Ø6 D1 DE D1 DA FE C3
[WL5]  256: 8Ø FB 81 73 Ø5 E8 5C ØØ FE C9 E8 86 ØØ 2A E9 8Ø
[TM5]  257: FD Ø7 77 33 8B EC 8B 46 Ø2 86 C4 89 46 Ø2 83 C5
[RM5]  258: Ø3 3B FD 75 Ø1 47 8B F7 F6 D1 B5 ØØ Ø3 F9 57 4E
[ZM5]  259: 4F 36 8A Ø4 36 88 Ø5 3B F5 77 F4 E3 Ø7 4F 36 C6
[EP5]  26Ø: Ø5 3Ø E2 F9 5F EB 81 BØ 45 AA BØ 2B 22 C9 79 Ø4
[ZP5]  261: F6 D9 BØ 2D AA 8A C1 B4 ØØ B1 ØA F6 F1 ØD 3Ø 3Ø
[136]  262: AB E9 64 FF 8B C2 8B EE D1 ED D1 D8 D1 ED D1 D8
[4R5]  263: 13 DØ 13 F5 73 Ø6 D1 DE D1 DA FE C3 8Ø C3 Ø3 C3
[BW5]  264: BØ ØØ D1 E2 D1 D6 DØ DØ FE CB 8Ø FB 8Ø 77 F3 ØC
[D66]  265: 3Ø AA C3 E8 EA FF BØ 2E AA E8 C8 FF E8 E1 FF FE
[TP5]  266: C5 8Ø FD Ø6 72 F3 26 8A 45 FF 3C 3Ø 75 Ø5 FE CD
[2V5]  267: 4F EB F3 3C 2E 74 F9 C3 E8 4B ØØ FE C1 8Ø FB 81
[MK5]  268: 73 F6 FE C9 81 C2 8Ø ØØ 83 D6 ØØ 73 Ø6 D1 DE D1
[416]  269: DA FE C3 E8 8E FF E8 BA FF 8Ø F9 Ø7 72 Ø3 E9 66
[AU5]  27Ø: FF 8B EC 83 C5 Ø3 FE C9 78 ØB 8B 46 ØØ 86 C4 89
[ZT5]  271: 46 ØØ 45 EB F1 3B FD 73 Ø2 8B FD 36 8Ø 7D FF 2E
[2T5]  272: 75 Ø1 4F E9 C2 FE 52 BD ØA ØØ 8B C6 33 D2 F7 F5
[3V5]  273: 8B FØ 58 F7 F5 92 B4 1A F6 E4 DØ EØ D1 D2 D1 D6
[CT5]  274: FE CB 23 F6 79 F4 C3 8B EC 8Ø 76 Ø4 8Ø C3 BA 86
[CP5]  275: Ø5 FF E2 83 EC Ø2 8B EC 8B 46 Ø6 87 46 Ø4 87 46
[7G5]  276: Ø2 89 46 ØØ 8B 46 Ø8 B3 ØØ 23 CØ 74 15 BB 9Ø 7F
[FX5]  277: 79 Ø4 F7 D8 B7 FF 23 CØ 78 Ø6 D1 EØ FE CB EB F6
[1F5]  278: 22 E7 C6 46 Ø6 ØØ 89 46 Ø7 88 5E Ø9 C3 BA 22 Ø6
[LX5]  279: FF E2 BA BE Ø6 FF E2 8B EC 8B 46 Ø8 23 CØ 74 2E
[DU5]  28Ø: 5Ø 24 7F 5Ø FF 76 Ø6 BA 4A Ø7 FF D2 8B EC FF 76
[1V5]  281: ØA FF 76 Ø8 BA 22 Ø6 FF D2 BA D2 Ø8 FF D2 58 5B
[4P5]  282: 59 8Ø E1 8Ø ØA D9 8B EC 89 46 Ø6 89 5E Ø8 C2 Ø4
[BN5]  283: ØØ 8B EC 8B 5E Ø4 4B 78 15 D1 E3 Ø3 5E Ø2 8B 56
[KH5]  284: ØØ 83 EA Ø3 3B DA 73 Ø6 2E 8B Ø7 89 46 ØØ C2 Ø4
[SV5]  285: ØØ BA E5 Ø9 FF E2 8B EC 8B 46 Ø3 8A 6E Ø5 22 ED
[TP5]  286: 74 16 8A DC 8Ø CC 8Ø B1 9Ø 2A CD 72 ØB D3 E8 15
[XK5]  287: ØØ ØØ 22 DB 79 Ø2 F7 D8 89 46 Ø4 C2 Ø2 ØØ Ø7 BA
[HG5]  288: 41 Ø8 FF D2 33 CØ 8Ø FA Ø1 75 Ø1 48 89 46 Ø8 Ø6
[MM5]  289: C2 Ø6 ØØ Ø7 BA 41 Ø8 FF D2 33 CØ 8Ø FA FF 75 Ø1
[7E5]  29Ø: 48 89 46 Ø8 Ø6 C2 Ø6 ØØ Ø7 BA 41 Ø8 FF D2 33 CØ
[KH5]  291: 8Ø FA ØØ 75 Ø1 48 89 46 Ø8 Ø6 C2 Ø6 ØØ 8B EC 8B
[EL5]  292: 46 Ø2 87 46 Ø4 87 46 Ø6 83 C4 Ø4 FF EØ 8B EC B4
[7M5]  293: ØF CD 1Ø B4 Ø3 CD 1Ø 8A D6 B6 ØØ 42 87 56 ØØ FF
[CN5]  294: E2 8B ØE ØF ØØ B4 Ø1 CD 1Ø 83 EC Ø2 8B EC 8B 76
[PC5]  295: Ø2 8A 46 Ø4 B4 ØD 89 46 Ø1 C6 46 Ø3 ØØ 22 CØ 74
[1F5]  296: 1E Ø4 Ø3 B4 2Ø 89 46 Ø1 C7 46 Ø3 2F 43 B4 ØØ 8B
[NH5]  297: F8 81 FF ØØ Ø1 76 Ø3 BF ØØ Ø1 C7 43 Ø2 ØD ØØ 1E
[GL5]  298: 56 16 45 55 8E 1E 2A ØØ 8E 1E 2C ØØ 1E 33 F6 E8
[6L5]  299: Ø8 ØØ 43 4F 4D 53 5Ø 45 43 1D 5B 8B D6 AC 22 CØ
[TS5]  29Ø: 74 1B 8B FB B9 Ø8 ØØ 24 DF 47 2E 3A 45 FF AC E1
[SU5]  2Ø1: F6 74 Ø7 AC 22 CØ 75 FB EB E1 83 C2 Ø8 8C DØ 8E
[ZM5]  2Ø2: CØ 8B DC E8 ØØ ØØ 5F 2E 8C 95 12 ØØ 2E 89 A5 17
[LP5]  2Ø3: ØØ B8 ØØ 4B CD 21 FA B8 ØØ ØØ 8E DØ BC ØØ ØØ FB
[8M5]  2Ø4: 8B EC 8E 5E Ø8 8B 6E Ø6 81 C4 ØE Ø1 B4 ØF CD 1Ø
[ZL5]  2Ø5: B4 Ø3 CD 1Ø 8Ø FE 18 72 ØB B8 ØA ØE CD 1Ø B6 17
[KP5]  2Ø6: B4 Ø2 CD 1Ø 8B ØE 24 ØØ B4 Ø1 CD 1Ø FF E5 8B EC
[BP5]  2Ø7: 8E Ø6 19 ØØ 1E 8C DØ 8E D8 8D 76 Ø4 8A ØC B5 ØØ
[8H5]  2Ø8: 8Ø F9 FA 72 Ø2 B1 FA Ø3 F1 8D 7C Ø1 8Ø F9 Ø4 76
[SK5]  2Ø9: Ø2 B1 Ø4 E3 Ø8 8Ø 3C 2E 74 ØF 4E E2 F8 C7 Ø5 2E
[2D5]  21Ø: 42 C7 45 Ø2 41 53 83 C7 Ø4 C6 Ø5 ØØ 8D 56 Ø5 B8
[ØL5]  211: ØØ 3D CD 21 72 35 8B D8 B9 Ø7 ØØ B4 3F CD 21 72
[8G5]  212: 26 3D Ø7 ØØ 75 21 8Ø 7E Ø5 FD 75 1B 8B 56 Ø2 83
[NT5]  213: FA FF 75 Ø8 8E 5E Ø6 8B 56 Ø8 EB Ø4 8C CØ 8E D8
[9R5]  214: 8B 4E ØA B4 3F CD 21 B4 3E CD 21 1F C2 Ø2 Ø1 1E
[4X5]  215: 8C DØ 8E D8 8B FC 83 C7 Ø5 8A 5D FF 8Ø FB FF 75
[PP5]  216: Ø2 FE CB B7 ØØ 88 39 8B D7 B4 41 CD 21 1F C2 ØØ
[HP5]  217: Ø1 1E 8C DØ 8E D8 8E CØ 8B F4 81 C6 Ø5 Ø1 8A 5C
[FØ6]  218: FF 8Ø FB FF 75 Ø2 FE CB B7 ØØ 88 38 8B D6 8D BC
[BZ5]  219: ØØ FF 8A 5D FF 8Ø FB FF 75 Ø2 FE CB 88 39 B4 56
[YT5]  22Ø: CD 21 1F C2 ØØ Ø2 81 EC FC ØØ 8B EC 8C DØ 8E CØ
[ZS5]  221: 1E BE 8Ø ØØ 8E 1E 2A ØØ AC 8A C8 B5 ØØ 8B 96 FE
[2D5]  222: ØØ 23 D2 74 32 8D 7E Ø1 E8 34 ØØ 74 22 3C 2Ø 74
[4L5]  223: F7 3C 2C 74 1A 3C 61 72 Ø6 3C 7A 77 Ø2 2C 2Ø AA
[YH5]  224: E8 1C ØØ 74 ØA 3C 2Ø 74 Ø6 3C 2F 75 E4 4E 41 4A
[5S5]  225: 75 D3 8D 55 FF 2B D5 88 56 ØØ 1F FF A6 FC ØØ BØ
[MV5]  226: ØD E3 Ø2 49 AC 3C ØD C3 B8 B3 ØB BB BC 3Ø E8 BF
[7L5]  227: EB C7 Ø6 17 ØØ ØØ ØØ E8 9F F8 B8 ØØ ØØ 5Ø E8 1E
[BF5]  228: F8 58 B8 ØØ ØØ 5Ø ØE E8 4C ØØ 4B 53 42 61 73 69
[NC5]  229: 63 2Ø 32 2E 3Ø 2Ø 2Ø 41 53 43 49 49 2D 2F 47 57
[GG5]  23Ø: 2D 42 61 73 69 63 2D 43 6F 6D 7Ø 69 6C 65 72 2Ø

[S95]  331: 2Ø 2Ø 28 43 29 31 39 38 39 2D 39 31 2Ø 44 4D 56
[CA5]  332: 2Ø 57 69 64 75 63 68 2Ø 47 6D 62 48 2Ø 26 2Ø 43
[FV5]  333: 6F 2E 2Ø 4B 47 2Ø E8 87 F7 E8 DC F7 E8 CØ F7 58
[GN5]  334: 8C D8 Ø5 5C Ø4 5Ø B8 Ø1 ØØ 5Ø E8 19 FF E8 91 F7
[SK5]  335: 8C D8 Ø5 5C Ø4 5Ø B8 ØØ ØØ 5Ø E8 63 F7 ØE E8 Ø1
[TK5]  336: ØØ ØØ E8 5B F7 E8 2B F9 58 21 CØ 75 Ø3 E9 39 ØØ
[AE5]  337: B8 ØØ ØØ 5Ø E8 88 F7 58 B8 ØØ ØØ 5Ø ØE E8 14 ØØ
[VD5]  338: 13 51 75 65 6C 6C 64 61 74 65 69 2Ø 5B 2E 42 41
[4N5]  339: 53 5D 3A 2Ø E8 29 F7 E8 7E F7 8C D8 Ø5 5C Ø4 5Ø
[VT5]  34Ø: E8 5Ø F8 E8 3B F7 E8 1Ø F9 8C D8 Ø5 5C Ø4 5Ø B8
[AK5]  341: ØØ ØØ 5Ø E8 ØA F7 ØE E8 Ø1 ØØ ØØ E8 Ø2 F7 E8 D2
[EL5]  342: F8 58 21 CØ 75 Ø3 E9 Ø3 ØØ E8 1D F6 8C D8 Ø5 6C
[EH5]  343: Ø4 5Ø 8C D8 Ø5 5C Ø4 5Ø B8 ØØ ØØ 5Ø E8 E1 F6 E8
[3G5]  344: FF F6 B8 3Ø ØØ 5Ø B8 Ø1 ØØ 5Ø 8C D8 Ø5 5C Ø4 5Ø
[WM5]  345: B8 ØØ ØØ 5Ø E8 C9 F6 ØE E8 Ø2 ØØ Ø1 2E E8 CØ F6
[VN5]  346: E8 CØ F8 58 5B 89 Ø7 B8 3Ø ØØ 5Ø 5B FF 37 B8 ØØ
[EL5]  347: ØØ 5Ø E8 ØF F6 58 21 CØ 75 Ø3 E9 28 ØØ 8C D8 Ø5
[XF5]  348: 5C Ø4 5Ø 8C D8 Ø5 5C Ø4 5Ø B8 ØØ ØØ 5Ø E8 9Ø F6
[3L5]  349: ØE E8 Ø5 ØØ Ø4 2E 42 41 53 E8 84 F6 E8 43 F7 E8
[NN5]  35Ø: 9F F6 E9 27 ØØ 8C D8 Ø5 6C Ø4 5Ø 8C D8 Ø5 5C Ø4
[EF5]  351: 5Ø B8 ØØ ØØ 5Ø E8 68 F6 B8 3Ø ØØ 5Ø 5B FF 37 B8
[LN5]  352: Ø1 ØØ 5Ø E8 15 F6 E8 EØ F8 E8 75 F6 E9 17 ØØ 5B
[2F5]  353: 8F Ø6 32 ØØ 53 B8 32 ØØ 5Ø 5B FF 37 B8 FF ØØ 5Ø
[YN5]  354: E8 Ø3 F6 E8 7F F8 E9 41 ØØ 5B 8F Ø6 34 ØØ 53 B8
[VT5]  355: 34 ØØ 5Ø 5B FF 37 B8 ØØ Ø1 5Ø E8 2F F9 E8 F3 FA
[ZS5]  356: E8 2F FB E8 8B FB B8 ØØ Ø1 5Ø B8 34 ØØ 5Ø 5B FF
[8R5]  357: 37 B8 ØØ ØØ 5Ø E8 59 F5 E8 E1 F5 E8 ØE F9 E8 C6
[VØ6]  358: FA E8 CA FA E8 6F FB E8 3B F8 E9 3F ØØ 5B 8F Ø6
[LL5]  359: 36 ØØ 8F Ø6 38 ØØ 53 B8 36 ØØ 5Ø 5B FF 77 Ø2 FF
[SB5]  36Ø: 37 B8 ØØ 91 5Ø B8 ØØ ØØ 5Ø B8 36 ØØ 5Ø 5B FF 77
[LX5]  361: Ø2 FF 37 B8 FF 7F 5Ø E8 D2 F8 E8 61 FB E8 CC F8
[Ø16]  362: E8 CA FA E8 88 FA E8 2D FB E8 F9 F7 E9 35 ØØ 5B
[3F5]  363: 8F Ø6 3A ØØ 53 B8 3A ØØ 5Ø 5B FF 37 B8 ØØ 91 5Ø
[TF5]  364: B8 ØØ ØØ 5Ø B8 3A ØØ 5Ø 5B FF 37 B8 ØØ ØØ 5Ø E8
[526]  365: DF F4 E8 97 F8 E8 95 FA E8 58 FA E8 49 FA E8 4D
[ØN5]  366: FA E8 59 FB B8 3C ØØ 5Ø B8 CF Ø7 5Ø 58 5B E8 Ø7
[2F5]  367: B8 7E 1F 5Ø B8 C7 ØØ 5Ø 58 5B 89 Ø7 B8 EØ 28 5Ø
[WC5]  368: B8 F3 Ø1 5Ø 58 5B 89 Ø7 B8 B2 3Ø 5Ø B8 63 ØØ 5Ø
[ND5]  369: 58 5B 89 Ø7 B8 44 32 5Ø B8 31 ØØ 5Ø 58 5B 89 Ø7
[NF5]  37Ø: B8 66 35 5Ø B8 31 ØØ 5Ø 58 5B 89 Ø7 8C D8 Ø5 3C
[4G5]  371: 17 5Ø ØE E8 Ø4 ØØ Ø3 45 58 45 E8 23 F5 E8 41 F5
[HK5]  372: B8 Ø2 ØØ 5Ø E8 BF FC ØE E8 Ø5 ØØ Ø4 2F 47 57 42
[VW5]  373: E8 ØD F5 E8 DD F6 B8 Ø2 ØØ 5Ø E8 A9 FC ØE E8 Ø5
[9S5]  374: ØØ Ø4 2F 67 77 62 E8 F7 F4 E8 C7 F6 E8 C2 F4 58
[1G5]  375: 21 CØ 75 Ø3 E9 14 ØØ 8C D8 Ø5 3C 17 5Ø ØE E8 Ø4
[NN5]  376: ØØ Ø3 47 57 42 E8 D8 F4 E8 F6 F4 8C D8 Ø5 4C 17
[CL5]  377: 5Ø 8C D8 Ø5 6C Ø4 5Ø B8 ØØ ØØ 5Ø E8 C2 F4 ØE E8
[9M5]  378: Ø2 ØØ Ø1 2E E8 B9 F4 E8 78 F5 8C D8 Ø5 3C 17 5Ø
[6S5]  379: B8 ØØ ØØ 5Ø E8 A9 F4 E8 68 F5 E8 C4 F4 8C D8 Ø5
[WF5]  38Ø: 5C 17 5Ø 8C D8 Ø5 5C Ø4 5Ø B8 ØØ ØØ 5Ø E8 9Ø F4
[UN5]  381: E8 AE F4 E8 5Ø 9Ø B8 ØØ ØØ 5Ø E8 C2 F4 58 B8 ØØ
[MG5]  382: ØØ 5Ø ØE E8 ØA ØØ Ø9 77 61 72 74 65 6E 2E 2E 2E
[RT5]  383: E8 6D F4 E8 C2 F4 E8 A6 F4 58 E8 5Ø FA B8 Ø2 ØØ
[LK5]  384: 5Ø E8 17 F4 B8 Ø1 ØØ 5Ø E8 6C F5 B8 ØØ 45 5Ø 5B
[UK5]  385: FF 77 Ø2 FF 37 B8 7Ø 85 5Ø B8 ØØ ØØ 5Ø E8 ØE F9
[MK5]  386: E8 B3 F9 8F Ø6 19 ØØ 8C D8 Ø5 6C 17 5Ø B8 ØØ ØØ
[9H5]  387: 5Ø E8 2C F4 ØE E8 Ø4 ØØ Ø3 5Ø 52 45 E8 21 F4 E8
[1V5]  388: EØ F4 B8 61 8E 5Ø B8 ØØ 1C 5Ø E8 89 F9 E8 DE FA
[4K5]  389: B8 47 38 5Ø E8 6B F6 B8 ØØ ØØ 5Ø E8 66 F3 58 21
[YF5]  39Ø: CØ 75 Ø3 E9 58 ØØ B8 ØØ ØØ 5Ø E8 32 F4 58 B8 ØØ
[GD5]  391: ØØ 5Ø ØE E8 ØE ØØ ØD 5Ø 72 65 7Ø 72 6F 63 65 73
[MN5]  392: 73 6F 72 2Ø E8 D9 F3 E8 2E F4 8C D8 Ø5 6C 17 5Ø
[ZH5]  393: B8 ØØ ØØ 5Ø E8 C9 F3 E8 1E F4 ØE E8 13 ØØ 12 5Ø
[KE5]  394: 52 45 2Ø 6E 69 63 68 74 2Ø 67 65 66 75 6E 64 65
[TZ5]  395: 6E E8 AC F3 E8 Ø1 F4 E8 E5 F3 58 E8 CB F2 E8 E9
[ZL5]  396: 8A B8 ØØ 45 5Ø 5B FF 77 Ø2 FF 37 E8 Ø8 F9 8F Ø6
[XD5]  397: 19 ØØ B8 Ø4 45 5Ø B8 2Ø 87 5Ø B8 ØØ ØØ 5Ø 58 5A
[WL5]  398: 5B 89 Ø7 89 57 Ø2 E8 4D 72 B8 Ø8 45 5Ø 5B FF 37
[BL5]  399: B8 Ø1 ØØ 5Ø E8 63 F3 E8 CA F2 58 21 CØ 75 Ø3 E9
[CH5]  4ØØ: Ø3 ØØ E9 68 ØØ B8 ØA 45 5Ø 5B FF 37 58 21 CØ 75
[EB5]  4Ø1: Ø3 E9 31 ØØ B8 B8 37 5Ø B8 Ø2 ØØ 5Ø B8 ØA 45 5Ø
[PT5]  4Ø2: 5B FF 37 B8 Ø1 ØØ 5Ø E8 F1 F2 E8 ØF F3 E8 EØ F2
[7S5]  4Ø3: B8 Ø8 45 5Ø 5B FF 37 B8 3Ø ØØ 5Ø 5B FF 37 E8 DA
[6F5]  4Ø4: F2 58 5B 89 Ø7 B8 3Ø ØØ 5Ø B8 Ø8 45 5Ø 5B FF 37
[ZK5]  4Ø5: 58 5B 89 Ø7 B8 ØA 45 5Ø B8 ØA 45 5Ø 5B FF 37 B8
[AP5]  4Ø6: Ø1 ØØ 5Ø E8 AA F2 58 5B 89 Ø7 E9 79 FF B8 ØC 45
[PF5]  4Ø7: 5Ø B8 3Ø ØØ 5Ø E8 F4 F5 58 5A 5B 89 Ø7 89 57 Ø2
[GG5]  4Ø8: E8 54 Ø6 B8 ØØ ØØ 5Ø E8 15 F3 58 B8 ØØ ØØ 5Ø E8
[ØL5]  4Ø9: ØD F3 58 B8 ØØ ØØ 5Ø ØE E8 ØC ØØ ØB 42 69 62 6C
[VN5]  41Ø: 69 6F 74 68 65 6B 3A E8 B6 F2 E8 ØB F3 B8 1Ø 45
[TS5]  411: 5Ø 5B FF 77 Ø2 FF 37 E8 E1 F5 FE 46 ØØ E8 F8 F2
[PF5]  412: E8 DC F2 58 B8 ØØ ØØ 5Ø ØE E8 ØC ØØ ØB 2Ø 2Ø 5Ø
[XR5]  413: 72 6F 67 72 61 6D 6D 3A E8 85 F2 E8 DA F2 B8 14
[8W5]  414: 45 5Ø 5B FF 77 Ø2 FF 37 E8 BØ F5 FE 46 ØØ E8 C7
[9H5]  415: F2 E8 AB F2 58 B8 ØØ ØØ 5Ø ØE E8 ØC ØØ ØB 2Ø 2Ø
[NL5]  416: 2Ø 2Ø 2Ø 53 74 61 63 6B 3A E8 54 F2 E8 A9 F2 B8
[BN5]  417: ØØ Ø4 5Ø E8 C2 F2 FE 46 ØØ E8 9C F2 E8 8Ø F2 58
[9B5]  418: B8 ØØ ØØ 5Ø ØE E8 ØC ØØ ØB 2Ø 56 61 72 69 61 62
[XW5]  419: 6C 65 6E 3A E8 29 F2 E8 7E F2 B8 ØC 45 5Ø 5B FF
[LV5]  42Ø: 77 Ø2 FF 37 E8 54 F5 FE 46 ØØ E8 6B F2 E8 4F F2
[YB5]  421: 58 B8 ØØ ØØ 5Ø ØE E8 ØC ØØ ØB 2Ø 2Ø 2Ø 53 74 72
[CT5]  422: 69 6E 67 73 3A E8 F8 F1 E8 4D F2 B8 18 45 5Ø 5B
[RR5]  423: FF 37 B8 ØØ 85 5Ø B8 ØØ ØØ 5Ø E8 B6 F6 E8 ED F6
[6S5]  424: E8 18 F5 FE 46 ØØ E8 2F F2 E8 13 F2 58 B8 ØØ 45
[HP5]  425: 5Ø 5B FF 77 Ø2 FF 37 E8 3C F7 8F Ø6 19 ØØ B8 1A
[4M5]  426: 45 5Ø B8 CØ Ø1 5Ø E8 C3 F4 58 5A 5B 89 Ø7 89 57
[2F5]  427: Ø2 B8 Ø8 45 5Ø B8 4D 5A 5Ø 58 5B 89 Ø7 E8 67 78
[BD5]  428: B8 Ø8 45 5Ø B8 2Ø ØØ 5Ø B8 1Ø 45 5Ø 5B FF 77 Ø2
[BS5]  429: FF 37 E8 5E F6 E8 56 F6 B8 14 45 5Ø 5B FF 77 Ø2
[6V5]  43Ø: FF 37 E8 49 F6 E8 85 FB B8 FF Ø1 5Ø E8 37 F1 58
[UF5]  431: 5B 89 Ø7 E8 31 78 B8 Ø8 45 5Ø B8 2Ø ØØ 5Ø B8 1Ø
[GS5]  432: 45 5Ø 5B FF 77 Ø2 FF 37 E8 28 F6 E8 2Ø F6 B8 14
[PU5]  433: 45 5Ø 5B FF 77 Ø2 FF 37 E8 13 F6 B8 FF Ø1 5Ø E8
[VN5]  434: 4A F4 E8 Ø9 F6 B8 ØØ Ø2 5Ø E8 4Ø F4 E8 43 F6 E8
[ZP5]  435: 9F F6 E8 A1 F6 58 5B 89 Ø7 E8 EB 77 B8 Ø8 45 5Ø
[ØK5]  436: B8 ØØ ØØ 5Ø 58 5B 89 Ø7 E8 DC 77 B8 Ø8 45 5Ø B8
[WH5]  437: Ø2 ØØ 5Ø 58 5B 89 Ø7 E8 CD 77 B8 Ø8 45 5Ø B8 ØØ
[HM5]  438: Ø1 5Ø B8 ØC 45 5Ø 5B FF 77 Ø2 FF 37 B8 ØF ØØ 5Ø
[NX5]  439: E8 F9 F3 E8 B8 F5 B8 1Ø ØØ 5Ø E8 EF F3 E8 F2 F5
[XV5]  44Ø: E8 4E F6 E8 AD F5 E8 A5 F5 B8 18 45 5Ø 5B FF 37
[WU5]  441: E8 D9 F3 E8 98 F5 E8 D4 FA 58 5B 89 Ø7 E8 87 77
```

```
[NP5]  442: E8 84 77 B8 Ø8 45 5Ø B8 1Ø 45 5Ø 5B FF 77 Ø2 FF
[7N5]  443: 37 B8 14 45 5Ø 5B FF 77 Ø2 FF 37 E8 7Ø F5 B8 ØF
[RP5]  444: ØØ 5Ø E8 A7 F3 E8 66 F5 B8 1Ø ØØ 5Ø E8 9D F3 E8
[UV5]  445: AØ F5 E8 FC F5 E8 95 FA 58 5B 89 Ø7 E8 48 77 B8
[UG5]  446: Ø8 45 5Ø B8 ØØ 1Ø 5Ø 58 5B 89 Ø7 E8 39 77 B8 Ø8
[GG5]  447: 45 5Ø B8 ØØ ØØ 5Ø 58 5B 89 Ø7 E8 2A 77 B8 Ø8 45
[DM5]  448: 5Ø B8 1Ø 45 5Ø 5B FF 77 Ø2 FF 37 E8 C8 F5 58 5B
[UC5]  449: 89 Ø7 E8 12 77 B8 Ø8 45 5Ø B8 ØØ ØØ 5Ø 58 5B 89
[ØE5]  45Ø: Ø7 E8 Ø3 77 B8 Ø8 45 5Ø B8 1C ØØ 5Ø 58 5B 89 Ø7
[2K5]  451: E8 F4 76 B8 Ø8 45 5Ø B8 ØØ ØØ 5Ø 58 5B 89 Ø7 B8
[7G5]  452: Ø1 ØØ 5Ø 8F Ø6 3Ø ØØ B8 Ø3 ØØ 5Ø 8F Ø6 1E 45 B8
[PG5]  453: Ø1 ØØ 5Ø 8F Ø6 2Ø 45 FF 36 2Ø 45 E8 Ø6 F1 58 8Ø
[ML5]  454: E4 Ø5 2E 88 26 52 1C E9 ØF ØØ FF 36 3Ø ØØ FF 36
[8L5]  455: 2Ø 45 E8 9B EF 8F Ø6 3Ø ØØ FF 36 3Ø ØØ FF 36 1E
[6N5]  456: 45 EB ØØ E8 64 EF EB Ø3 E8 4C EF 58 21 CØ 75 Ø3
[NH5]  457: E9 Ø6 ØØ E8 91 76 E9 D1 FF B8 1Ø 45 5Ø B8 ØØ ØØ
[SF5]  458: 5Ø E8 C8 F2 58 5A 5B 89 Ø7 89 57 Ø2 B8 3Ø ØØ 5Ø
[YB5]  459: B8 ØØ ØØ 5Ø 58 5B 89 Ø7 B8 48 39 5Ø B8 Ø2 ØØ 5Ø
[LS5]  46Ø: B8 3Ø ØØ 5Ø 5B FF 37 E8 72 EF E8 43 EF 5B FF 37
[RH5]  461: B8 ØØ ØØ 5Ø E8 ED EE 58 21 CØ 75 Ø3 E9 Ø3 ØØ E9
[DH5]  462: B4 ØØ B8 48 39 5Ø B8 Ø2 ØØ 5Ø B8 3Ø ØØ 5Ø 5B FF
[YV5]  463: 37 E8 48 EF E8 19 EF B8 1Ø 45 5Ø 5B FF 77 Ø2 FF
[PG5]  464: 37 E8 D2 F4 58 5B 89 Ø7 B8 1Ø 45 5Ø B8 1Ø 45 5Ø
[8M5]  465: 5B FF 77 Ø2 FF 37 B8 B8 37 5Ø B8 Ø2 ØØ 5Ø B8 3Ø
[HY5]  466: ØØ 5Ø 5B FF 37 E8 14 EF E8 E5 EE 5B FF 37 E8 3B
[LK5]  467: F2 E8 FA F3 58 5A 5B 89 Ø7 89 57 Ø2 B8 Ø4 45 5Ø
[WL5]  468: B8 Ø2 ØØ 5Ø B8 3Ø ØØ 5Ø 5B FF 37 E8 EE EE B8 51
[WM5]  469: ØØ 5Ø E8 BB EE E8 14 F2 58 5A 5B 89 Ø7 89 57 Ø2
[WK5]  47Ø: B8 22 45 5Ø B8 Ø4 45 5Ø 5B FF 77 Ø2 FF 37 E8 65
[ZM5]  471: F4 E8 4E F1 B8 8Ø ØØ 5Ø E8 95 EE 58 5B 89 Ø7 B8
[XM5]  472: Ø4 45 5Ø 5B FF 77 Ø2 FF 37 E8 4A F4 B8 22 45 5Ø
[3N5]  473: 5B FF 37 E8 1C F1 B8 3Ø ØØ 5Ø B8 3Ø ØØ 5Ø 5B FF
[YE5]  474: 37 B8 Ø1 ØØ 5Ø E8 68 EE 58 5B 89 Ø7 B8 3Ø ØØ 5Ø
[1V5]  475: 5B FF 37 B8 ØA 45 5Ø 5B FF 37 E8 F4 ED 58 21 CØ
[PH5]  476: 75 Ø3 E9 Ø3 ØØ E9 FØ FE B8 ØØ ØØ 5Ø 8F Ø6 3Ø ØØ
[1D5]  477: B8 19 ØØ 5Ø 8F Ø6 24 45 B8 Ø1 ØØ 5Ø 8F Ø6 26 45
[CS5]  478: FF 36 26 45 E8 7D EF 58 8Ø E4 Ø5 2E 88 26 DB 1D
[FP5]  479: E9 ØF ØØ FF 36 3Ø ØØ FF 36 26 45 E8 12 EE 8F Ø6
[RR5]  48Ø: 3Ø ØØ FF 36 3Ø ØØ FF 36 24 45 EB ØØ E8 DB ED EB
[DH5]  481: Ø3 E8 C3 ED 58 21 CØ 75 Ø3 E9 2Ø ØØ B8 94 36 5Ø
[ZP5]  482: B8 Ø2 ØØ 5Ø B8 3Ø ØØ 5Ø 5B FF 37 E8 ØE EE E8 DF
[ØM5]  483: ED B8 ØØ ØØ 5Ø 58 5B 89 Ø7 E9 B7 FF B8 28 45 5Ø
[2H5]  484: B8 Ø1 ØØ 5Ø 58 5B 89 Ø7 E8 8C Ø1 8C D8 Ø5 5C 17
[SK5]  485: 5Ø 8C D8 Ø5 4C 17 5Ø B8 ØØ ØØ 5Ø E8 Ø2 EE E8 2Ø
[MS5]  486: EE E8 AA 8B B8 ØØ 45 5Ø 5B FF 77 Ø2 FF 37 B8 7Ø
[MH5]  487: 85 5Ø B8 ØØ ØØ 5Ø E8 B5 F2 E8 5A F3 8F Ø6 19 ØØ
[SK5]  488: B8 Ø4 45 5Ø B8 4A 38 5Ø E8 E1 FØ 58 5A 5B 89 Ø7
[HT5]  489: 89 57 Ø2 FF 36 Ø6 45 FF 36 Ø4 45 E8 38 F3 E8 BA
[RN5]  49Ø: FØ B8 ØØ ØØ 5Ø E8 F7 ED 58 B8 ØØ ØØ 5Ø E8 EF ED
[HE5]  491: 58 B8 3Ø ØØ 5Ø E8 95 F3 58 5B 89 Ø7 ØE E8 Ø7 ØØ
[PH5]  492: Ø6 4C 5A 45 58 45 2Ø E8 96 ED 8C D8 Ø5 6C Ø4 5Ø
[1P5]  493: B8 ØØ ØØ 5Ø E8 89 ED E8 48 EE E8 84 F3 B8 2A 45
[7K5]  494: 5Ø E8 69 F3 58 5B 89 Ø7 B8 3Ø ØØ 5Ø 5B FF 37 B8
[NE5]  495: Ø1 ØØ 5Ø E8 81 EE B8 ØØ ØØ 5Ø B8 5Ø ØØ 5Ø ØE E8
[AW5]  496: Ø2 ØØ Ø1 2Ø E8 59 ED E8 2D FØ E8 AB ED E8 8F ED
[5M5]  497: 58 8C D8 Ø5 6C Ø4 5Ø B8 ØØ ØØ 5Ø E8 42 ED ØE E8
[AU5]  498: Ø5 ØØ Ø4 2E 4F 4C 44 E8 36 ED E8 F5 ED E8 7F F4
[ØP5]  499: B8 2A 45 5Ø 5B FF 37 B8 Ø1 ØØ 5Ø E8 DD EC B8 Ø3
[ZL5]  5ØØ: ØØ 5Ø B8 2A 45 5Ø 5B FF 37 B8 14 ØØ 5Ø E8 4E EC
[8T5]  5Ø1: E8 E9 EC E8 BA EC B8 Ø1 ØØ 5Ø E8 1A EE B8 ØØ ØØ
[WP5]  5Ø2: 5Ø ØE E8 Ø4 ØØ Ø3 45 58 45 E8 F4 EC E8 49 ED 58
[NA5]  5Ø3: B8 ØØ ØØ 5Ø ØE E8 Ø9 ØØ Ø8 2D 44 61 74 65 69 2Ø
[YR5]  5Ø4: 2D E8 DC EC E8 31 ED 8C D8 Ø5 4C 17 5Ø B8 ØØ ØØ
[KT5]  5Ø5: 5Ø E8 CC EC E8 21 ED ØE E8 ØC ØØ ØB 2D 2Ø 67 65
[ZT5]  5Ø6: 6E 65 72 69 65 72 74 E8 B6 EC E8 ØB ED B8 FF ØØ
[1W5]  5Ø7: 5Ø ØE E8 Ø2 ØØ Ø1 2Ø E8 A6 EC E8 7A EF E8 F8 EC
[WL5]  5Ø8: E8 DC EC 58 B8 ØØ ØØ 5Ø E8 D4 EC 58 B8 ØØ ØØ 5Ø
[XW5]  5Ø9: E8 CC EC 58 E8 B2 EB B8 ØØ 45 5Ø 5B FF 77 Ø2 FF
[KM5]  51Ø: 37 B8 68 8A 5Ø B8 ØØ 8Ø 5Ø E8 42 F1 E8 E7 F1 8F
[MN5]  511: Ø6 19 ØØ B8 Ø4 45 5Ø B8 2C 45 5Ø 5B FF 77 Ø2 FF
[XL5]  512: 37 B8 Ø1 ØØ 5Ø E8 64 EF E8 23 F1 58 5A 5B 89 Ø7
[DR5]  513: 89 57 Ø2 B8 1A 45 5Ø B8 B8 ØB 5Ø E8 4E EF 58 5A
[BL5]  514: 5B 89 Ø7 89 57 Ø2 B8 Ø8 45 5Ø B8 14 45 5Ø 5B FF
[KS5]  515: 77 Ø2 FF 37 B8 1Ø 45 5Ø 5B FF 77 Ø2 FF 37 E8 ED
[PM5]  516: FØ B8 ØF ØØ 5Ø E8 24 EF E8 E3 FØ B8 1Ø ØØ 5Ø E8
[3U5]  517: 1A EF E8 1D F1 E8 79 F1 B8 ØØ Ø1 5Ø E8 ØD EF E8
[HG5]  518: CC FØ E8 Ø8 F6 58 5B 89 Ø7 B8 14 45 5Ø B8 ØØ ØØ
[AK5]  519: 5Ø E8 F8 EE 58 5A 5B 89 Ø7 89 57 Ø2 B8 ØØ ØØ 5Ø
[2K5]  52Ø: E8 1C EC 58 B8 22 45 5Ø B8 B8 ØØ 5Ø 58 5B 89 Ø7
[5R5]  521: E8 EE 71 E8 5A 72 B8 22 45 5Ø B8 BB ØØ 5Ø 58 5B
[9M5]  522: 89 Ø7 E8 DC 71 B8 Ø8 45 5Ø B8 ØC 45 5Ø 5B FF 77
[PL5]  523: Ø2 FF 37 B8 ØF ØØ 5Ø E8 B2 EE E8 71 FØ B8 1Ø ØØ
[SW5]  524: 5Ø E8 A8 EE E8 AB FØ E8 Ø7 F1 B8 18 45 5Ø 5B FF
[6W5]  525: 37 E8 98 EE E8 57 FØ E8 93 F5 58 5B 89 Ø7 E8 ØF
[WC5]  526: 72 B8 3Ø 45 5Ø B8 ØØ ØØ 5Ø 58 5B 89 Ø7 E8 D6 72
[ZG5]  527: B8 Ø8 45 5Ø B8 C7 Ø6 5Ø 58 5B 89 Ø7 E8 F1 71 B8
[ML5]  528: Ø8 45 5Ø B8 17 ØØ 5Ø 58 5B 89 Ø7 E8 E2 71 B8 7C
[RN5]  529: 31 5Ø B8 Ø2 ØØ 5Ø B8 ØØ ØØ 5Ø E8 1F EB E8 FØ EA
[ZG5]  53Ø: 5B FF 37 58 21 CØ 75 Ø3 E9 33 ØØ B8 Ø8 45 5Ø B8
[1H5]  531: 1Ø 45 5Ø 5B FF 77 Ø2 FF 37 B8 7C 31 5Ø B8 Ø2 ØØ
[VU5]  532: 5Ø B8 ØØ ØØ 5Ø E8 F4 EA E8 C5 EA 5B FF 37 E8 1B
[EX5]  533: EE E8 DA EF E8 16 F5 58 5B 89 Ø7 E9 ØC ØØ B8 Ø8
[YE5]  534: 45 5Ø B8 ØØ ØØ 5Ø 58 5B 89 Ø7 E8 83 71 B8 32 45
[Ø95]  535: 5Ø B8 ØØ ØØ 5Ø 58 5B 89 Ø7 B8 34 45 5Ø B8 ØØ ØØ
[RB5]  536: 5Ø 58 5B 89 Ø7 B8 36 45 5Ø B8 ØØ ØØ 5Ø 58 5B 89
[CB5]  537: Ø7 B8 38 45 5Ø B8 ØØ ØØ 5Ø 58 5B 89 Ø7 B8 3A 45
[F95]  538: 5Ø B8 ØØ ØØ 5Ø 58 5B 89 Ø7 B8 3C 45 5Ø B8 ØØ ØØ
[6C5]  539: 5Ø 58 5B 89 Ø7 B8 3E 45 5Ø B8 ØØ ØØ 5Ø 58 5B 89
[2C5]  54Ø: Ø7 B8 4Ø 45 5Ø B8 ØØ ØØ 5Ø 58 5B 89 Ø7 E8 66 69
[9G5]  541: B8 Ø8 45 5Ø 5B FF 37 58 21 CØ 75 Ø3 E9 Ø3 ØØ E9
[1R5]  542: 5B ØØ B8 DE ØF 5Ø B8 Ø2 ØØ 5Ø B8 32 45 5Ø 5B FF
[HV5]  543: 37 E8 48 EA E8 19 EA B8 14 45 5Ø 5B FF 77 Ø2 FF
[HF5]  544: 37 E8 69 F4 58 5B 89 Ø7 B8 32 45 5Ø B8 32 45 5Ø
[PN5]  545: 5B FF 37 B8 Ø1 ØØ 5Ø E8 F6 E9 58 5B 89 Ø7 B8 42
[EH5]  546: 45 5Ø B8 32 45 5Ø 5B FF 37 58 5B 89 Ø7 B8 3Ø 45
[PF5]  547: 5Ø B8 Ø1 ØØ 5Ø 58 5B 89 Ø7 E8 8A 71 C3 E8 F6 68
[LS5]  548: B8 32 45 5Ø 5B FF 37 B8 3C ØØ 5Ø 5B FF 37 E8 4D
[VG5]  549: E9 58 21 CØ 75 Ø3 E9 2A ØØ B8 ØØ ØØ 5Ø ØE E8 18
[9C5]  55Ø: ØØ 17 5A 75 2Ø 76 69 65 6C 65 2Ø 5Ø 72 6F 67 72
[WT5]  551: 61 6D 6D 7A 65 69 6C 65 6E E8 E4 E9 E8 39 EA 58
[DK5]  552: E9 59 67 B8 ØØ ØØ 5Ø ØE E8 Ø7 ØØ Ø6 5A 65 69 6C
```

```
[4U5]  553: 65 3A E8 CB E9 E8 2Ø EA B8 32 45 5Ø 5B FF 37 B8
[PT5]  554: Ø1 ØØ 5Ø E8 6A E9 E8 2F EA FE 46 ØØ E8 Ø9 EA ØE
[YP5]  555: E8 Ø2 ØØ Ø1 28 E8 A8 E9 E8 FD E9 B8 Ø8 45 5Ø 5B
[KU5]  556: FF 37 E8 EA F3 E8 D3 EC B8 Ø2 ØØ 5Ø B8 FF ØØ 5Ø
[GS5]  557: E8 2A EC E8 E2 E9 ØE E8 Ø2 ØØ Ø1 29 E8 81 E9 E8
[ZW5]  558: D6 E9 B8 1E ØØ 5Ø E8 68 EA E8 CC E9 E8 BØ E9 58
[2G5]  559: B8 3E ØØ 5Ø B8 Ø2 ØØ 5Ø B8 32 45 5Ø 5B FF 37 E8
[9P5]  56Ø: 3A E9 E8 ØB E9 B8 Ø8 45 5Ø 5B FF 37 58 5B 89 Ø7
[9H5]  561: B8 DE ØF 5Ø B8 Ø2 ØØ 5Ø B8 32 45 5Ø 5B FF 37 E8
[RT5]  562: 1A E9 E8 EB E8 B8 14 45 5Ø 5B FF 77 Ø2 FF 37 E8
[YM5]  563: 3B F3 58 5B 89 Ø7 B8 32 45 5Ø B8 32 45 5Ø 5B FF
[2M5]  564: 37 B8 Ø1 ØØ 5Ø E8 C8 E8 58 5B 89 Ø7 E8 AE 67 B8
[GN5]  565: 44 45 5Ø 5B FF 37 B8 3A ØØ 5Ø E8 67 E8 B8 44 45
[1M5]  566: 5Ø 5B FF 37 B8 2Ø ØØ 5Ø E8 59 E8 E8 C3 E8 58 21
[NN5]  567: CØ 75 Ø3 E9 Ø3 ØØ E9 D3 FF B8 44 45 5Ø 5B FF 37
[SG5]  568: B8 ØØ ØØ 5Ø E8 3D E8 58 21 CØ 75 Ø3 E9 Ø3 ØØ E9
[LM5]  569: 3B FE B8 44 45 5Ø 5B FF 37 B8 41 ØØ 5Ø E8 37 E8
[GP5]  57Ø: B8 44 45 5Ø 5B FF 37 B8 5A ØØ 5Ø E8 3C E8 B8 75
[FD5]  571: E8 58 21 CØ 75 Ø3 E9 Ø3 ØØ E9 1C ØØ B8 44 45 5Ø
[XK5]  572: 5B FF 37 B8 88 ØØ 5Ø E8 33 E8 58 21 CØ 75 Ø3 E9
[HP5]  573: Ø3 ØØ E9 9F ØØ E8 A1 67 E8 BB 67 B8 46 45 5Ø 5B
[7G5]  574: FF 37 58 21 CØ 75 Ø3 E9 28 ØØ B8 ØØ ØØ 5Ø ØE E8
[RC5]  575: 16 ØØ 15 5A 75 77 65 69 73 75 6E 67 2Ø 61 6E 2Ø
[9S5]  576: 43 4F 4D 4D 41 4E 44 24 E8 55 E8 E8 AA E8 58 E9
[FR5]  577: CA 65 B8 48 45 5Ø B8 4A 45 5Ø 5B FF 37 58 5B 89
[8M5]  578: Ø7 B8 44 45 5Ø 5B FF 37 B8 E7 ØØ 5Ø E8 CE E7 58
[DG5]  579: 21 CØ 75 Ø3 E9 1F ØØ B8 ØØ ØØ 5Ø ØE E8 ØD ØØ ØC
[XF5]  58Ø: 27 3D 27 2Ø 65 72 77 61 72 74 65 74 E8 11 E8 E8
[DL5]  581: 66 E8 58 E9 86 65 E8 2Ø 67 E8 Ø6 3F B8 4C 45 5Ø
[VP5]  582: B8 48 45 5Ø 5B FF 37 58 5B 89 Ø7 E8 CE 73 E8 78
[1N5]  583: 7E E9 35 3E B8 44 45 5Ø 5B FF 37 B8 91 ØØ 5Ø E8
[6L5]  584: 7B E7 58 21 CØ 75 Ø3 E9 Ø3 ØØ E9 E2 Ø1 E8 E9 66
[CK5]  585: B8 44 45 5Ø 5B FF 37 B8 23 ØØ 5Ø E8 5F E7 58 21
[TE5]  586: CØ 75 Ø3 E9 12 ØØ B8 Ø8 45 5Ø B8 ØØ ØØ 5Ø 58 5B
[FS5]  587: 89 Ø7 E8 B9 6E E9 1B ØØ E8 BE 66 E8 85 74 8C D8
[KK5]  588: Ø5 7C 17 5Ø ØE E8 Ø2 ØØ Ø1 2C E8 93 E7 E8 B1 E7
[YU5]  589: E8 1F 74 B8 44 45 5Ø 5B FF 37 B8 CE ØØ 5Ø E8 1C
[YG5]  59Ø: E7 58 21 CØ 75 Ø3 E9 Ø3 ØØ E9 18 ØØ E8 8A 66 E8
[EL5]  591: 51 74 B8 3Ø 45 5Ø B8 5A ØØ 5Ø 58 5B 89 Ø7 E8 C5
[7L5]  592: 6E E9 2E ØØ B8 44 45 5Ø 5B FF 37 B8 D2 ØØ 5Ø E8
[ØK5]  593: EB E6 58 21 CØ 75 Ø3 E9 Ø3 ØØ E9 2A ØØ E8 59 66
[VK5]  594: E8 2Ø 74 B8 3Ø 45 5Ø B8 5B ØØ 5Ø 58 5B 89 Ø7 E8
[7K5]  595: 94 6E 8C D8 Ø5 7C 17 5Ø ØE E8 Ø2 ØØ Ø1 29 E8 1F
[RV5]  596: E7 E8 3D E7 E8 AB 73 E8 18 3E B8 4A 45 5Ø 5B FF
[HH5]  597: 37 E8 Ø8 ØØ 32 25 41 25 7F 89 6E 25 E8 52 EC E9
[SK5]  598: 4B ØØ B8 3Ø 45 5Ø B8 16 ØØ 5Ø 58 5B 89 Ø7 E9 ØC
[7D5]  599: ØØ B8 3Ø 45 5Ø B8 46 ØØ 5Ø 58 5B 89 Ø7 E8 46 6E
[DK5]  59Ø: B8 Ø8 45 5Ø B8 FE 46 5Ø 58 5B 89 Ø7 E8 61 6D B8
[XG5]  6Ø1: 22 45 5Ø B8 ØØ ØØ 5Ø 58 5B 89 Ø7 E8 E3 6C B8 3Ø
[SH5]  6Ø2: 45 5Ø B8 15 ØØ 5Ø 58 5B 89 Ø7 E8 19 6E B8 44 45
[KH5]  6Ø3: 5Ø 5B FF 37 B8 3B ØØ 5Ø E8 42 E6 58 21 CØ 75 Ø3
[XK5]  6Ø4: E9 Ø3 ØØ E9 3Ø ØØ E8 BØ 65 B8 44 45 5Ø 5B FF 37
[2S5]  6Ø5: B8 ØØ ØØ 5Ø E8 ED E5 B8 44 45 5Ø 5B FF 37 B8 3A
[HL5]  6Ø6: ØØ 5Ø E8 DF E5 E8 49 E6 58 21 CØ 75 Ø3 E9 Ø3 ØØ
[BP5]  6Ø7: E9 6A ØØ E9 B7 FF B8 44 45 5Ø 5B FF 37 B8 2C ØØ
[ØK5]  6Ø8: 5Ø E8 F9 E5 58 21 CØ 75 Ø3 E9 Ø3 ØØ E9 15 ØØ E8
[MH5]  6Ø9: 67 65 B8 3Ø 45 5Ø B8 21 ØØ 5Ø 58 5B 89 Ø7 E8 A5
[WR5]  61Ø: 6D E9 A5 FF B8 44 45 5Ø 5B FF 37 B8 ØØ 5Ø E8 E8
[VP5]  611: CB E5 B8 44 45 5Ø 5B FF 37 B8 3A ØØ 5Ø E8 BD E5
[KL5]  612: E8 E3 E5 58 21 CØ 75 Ø3 E9 Ø3 ØØ E9 85 FE B8 3Ø
[PH5]  613: 45 5Ø B8 14 ØØ 5Ø 58 5B 89 Ø7 E8 69 6D B8 22 45
[9H5]  614: 5Ø B8 58 ØØ 5Ø 58 5B 89 Ø7 E8 15 6C E9 3A 3C B8
[GN5]  615: 44 45 5Ø 5B FF 37 B8 9D ØØ 5Ø E8 8Ø E5 58 21 CØ
[ZG5]  616: 75 Ø3 E9 Ø3 ØØ E9 15 ØØ E8 EE 64 B8 Ø8 45 5Ø B8
[EM5]  617: Ø4 ØØ 5Ø 58 5B 89 Ø7 E8 D4 6C E9 36 FE B8 44 45
[KN5]  618: 5Ø 5B FF 37 B8 AC ØØ 5Ø E8 Ø6 E5 B8 44 45 5Ø 5B
[SU5]  619: FF 37 B8 AF ØØ 5Ø E8 E5 E4 E8 75 E5 58 21 CØ 75
[VF5]  62Ø: Ø3 E9 Ø3 ØØ E9 82 Ø1 B8 4A 45 5Ø B8 44 45 5Ø 5B
[BP5]  621: FF 37 B8 AC ØØ 5Ø E8 42 E5 58 5B 89 Ø7 B8 4A 45
[7H5]  622: 5Ø 5B FF 37 B8 ØØ ØØ 5Ø E8 D9 E4 58 21 CØ 75 Ø3
[3L5]  623: E9 ØC ØØ B8 4A 45 5Ø B8 Ø4 ØØ 5Ø 58 5B 89 Ø7 E8
[AL5]  624: 77 64 B8 44 45 5Ø 5B FF 37 B8 41 ØØ 5Ø E8 A1 E4
[LL5]  625: B8 44 45 5Ø 5B FF 37 B8 5A ØØ 5Ø E8 8Ø E4 E8 1Ø
[BE5]  626: E5 58 21 CØ 75 Ø3 E9 25 ØØ B8 ØØ ØØ 5Ø ØE E8 13
[A95]  627: ØØ 12 42 75 63 68 73 74 61 62 65 2Ø 65 72 77 61
[4S5]  628: 72 74 65 74 E8 19 E5 E8 6E E5 58 E9 8E 62 B8 Ø8
[AL5]  629: 45 5Ø B8 44 45 5Ø 5B FF 37 B8 41 ØØ 5Ø E8 BB E4
[8S5]  63Ø: 58 5B 89 Ø7 E8 12 64 B8 44 45 5Ø 5B FF 37 B8 EA
[3G5]  631: ØØ 5Ø E8 88 E4 58 21 CØ 75 Ø3 E9 Ø3 ØØ E9 A6 ØØ
[5M5]  632: E8 F6 63 B8 Ø8 45 5Ø 5B FF 37 8F Ø6 3Ø ØØ B8 44
[XK5]  633: 45 5Ø 5B FF 37 B8 41 ØØ 5Ø E8 7F E4 8F Ø6 4E 45
[PK5]  634: B8 Ø1 ØØ 5Ø 8F Ø6 5Ø 45 FF 36 5Ø 45 E8 B5 E5 58
[BP5]  635: 8Ø E4 Ø5 2E 88 26 A3 27 E9 ØF ØØ FF 36 3Ø ØØ FF
[1K5]  636: 36 5Ø 45 E8 4A E4 8F Ø6 3Ø ØØ FF 36 3Ø ØØ FF 36
[1U5]  637: 4E 45 EB ØØ E8 13 E4 EB Ø3 E8 FB E3 58 21 CØ 75
[695]  638: Ø3 E9 23 ØØ B8 94 36 5Ø B8 Ø2 ØØ 5Ø B8 3Ø ØØ 5Ø
[5R5]  639: 5B FF 37 E8 46 E4 E8 17 E4 B8 4A 45 5Ø 5B FF 37
[YT5]  64Ø: 58 5B 89 Ø7 E9 B4 FF E8 6F 63 B8 44 45 5Ø 5B FF
[RE5]  641: 37 B8 2C ØØ 5Ø E8 AC E3 58 21 CØ 75 Ø3 E9 Ø3 ØØ
[KP5]  642: E9 DC FE E9 83 3A B8 94 36 5Ø B8 Ø2 ØØ 5Ø B8 Ø8
[GS5]  643: 45 5Ø 5B FF 37 E8 Ø4 E4 E8 D5 E3 B8 4A 45 5Ø 5B
[FT5]  644: FF 37 58 5B 89 Ø7 E9 C1 FF B8 44 45 5Ø 5B FF 37
[TG5]  645: B8 86 ØØ 5Ø E8 A6 E3 58 21 CØ 75 Ø3 E9 Ø3 ØØ E9
[BP5]  646: D5 Ø5 E8 14 63 E8 2B 6E 8C D8 Ø5 8C 17 5Ø 8C D8
[SE5]  647: Ø5 8C 17 5Ø B8 ØØ ØØ 5Ø E8 E5 E3 ØE E8 Ø2 ØØ Ø1
[3X5]  648: 28 E8 DC E3 E8 9B E4 E8 F7 E3 B8 28 45 5Ø 5B FF
[PD5]  649: 37 58 21 CØ 75 Ø3 E9 Ø3 ØØ E9 EB ØØ B8 ØØ ØØ 5Ø
[MG5]  65Ø: 8F Ø6 3Ø ØØ B8 52 45 5Ø 5B FF 37 B8 Ø1 ØØ 5Ø E8
[MF5]  651: 69 E3 8F Ø6 54 45 B8 Ø1 ØØ 5Ø 8F Ø6 56 45 FF 36
[US5]  652: 56 45 E8 9F E4 58 8Ø E4 Ø5 2E 88 26 B9 28 E9 ØF
[TD5]  653: ØØ FF 36 3Ø ØØ FF 36 56 45 E8 34 E3 8F Ø6 3Ø ØØ
[RT5]  654: FF 36 3Ø ØØ FF 36 54 45 EB ØØ E8 FD E2 EB Ø3 E8
[TF5]  655: E5 E2 58 21 CØ 75 Ø3 E9 66 ØØ 8C D8 Ø5 8C 17 5Ø
[SF5]  656: B8 ØØ ØØ 5Ø E8 59 E3 8C D8 Ø5 7C Ø4 5Ø B8 1Ø ØØ
[NL5]  657: 5Ø B8 3Ø ØØ 5Ø 5B FF 37 E8 21 E3 E8 F2 E2 B8 ØØ
[PH5]  658: ØØ 5Ø E8 3B E3 E8 ØB E5 58 21 CØ 75 Ø3 E9 2D ØØ
```

Listing 1. »sbasic.exe« ist der Hexdump des Basic-Compilers. Geben Sie dieses Listing nur dann ein, wenn Sie keine frühere Version von SBasic besitzen.

```
[ZD5]  659: B8 ØØ ØØ 5Ø ØE E8 1B ØØ 1A 46 65 6C 64 2Ø 62 65
[RG5]  66Ø: 72 65 69 74 73 2Ø 64 69 6D 65 6E 73 69 6F 6E 69
[KW5]  661: 65 72 74 E8 ØA E3 E8 5F E3 58 E9 7F 6Ø E9 71 FF
[ZK5]  662: 8C D8 Ø5 7C Ø4 5Ø B8 1Ø ØØ 5Ø B8 52 45 5Ø 5B FF
[UM5]  663: 37 E8 C8 E2 E8 99 E2 8C D8 Ø5 8C 17 5Ø B8 ØØ ØØ
[XS5]  664: 5Ø E8 DC E2 E8 FA E2 8C D8 Ø5 7C 17 5Ø ØE E8 Ø2
[3P5]  665: ØØ Ø1 28 E8 CA E2 E8 E8 E2 E8 56 6F B8 44 45 5Ø
[RP5]  666: 5B FF 37 B8 1Ø ØØ 5Ø E8 F4 E1 B8 44 45 5Ø 5B FF
[YH5]  667: 37 B8 1B ØØ 5Ø E8 F9 E1 E8 6B E2 58 21 CØ 75 Ø3
[1M5]  668: E9 19 ØØ B8 Ø8 45 5Ø B8 44 45 5Ø 5B FF 37 B8 11
[UF5]  669: ØØ 5Ø E8 46 E2 58 5B 89 Ø7 E9 A2 ØØ B8 44 45 5Ø
[NN5]  67Ø: 5B FF 37 B8 ØF ØØ 5Ø E8 DA E1 58 21 CØ 75 Ø3 E9
[CF5]  671: 15 ØØ E8 Ø8 61 B8 Ø8 45 5Ø B8 44 45 5Ø 5B FF 37
[BP5]  672: 58 5B 89 Ø7 E9 77 ØØ B8 44 45 5Ø 5B FF 37 B8 1C
[4M5]  673: ØØ 5Ø E8 AF E1 B8 44 45 5Ø 5B FF 37 B8 ØB ØØ 5Ø
[GP5]  674: E8 A1 E1 E8 ØB E2 B8 44 45 5Ø 5B FF 37 B8 ØC ØØ
[BL5]  675: 5Ø E8 9Ø E1 E8 FA E1 58 21 CØ 75 Ø3 E9 Ø6 ØØ E8
[UD5]  676: F4 6Ø E9 39 ØØ B8 ØØ ØØ 5Ø ØE E8 27 ØØ 26 56 61
[LK5]  677: 72 69 61 62 6C 65 2Ø 44 69 6D 65 6E 73 69 6F 6E
[LH5]  678: 69 65 72 75 6E 67 2Ø 6E 69 63 68 74 2Ø 65 72 6C
[9T5]  679: 61 75 62 74 E8 E9 E1 E8 3E E2 58 E9 5E 5F B8 28
[PH5]  68Ø: 45 5Ø 5B FF 37 58 21 CØ 75 Ø3 E9 Ø3 ØØ E9 D5 ØØ
[TA5]  681: B8 3Ø ØØ 5Ø B8 1Ø 21 5Ø B8 ØA ØØ 5Ø B8 52 45 5Ø
[PG5]  682: 5B FF 37 E8 96 E1 E8 67 E1 B8 Ø2 ØØ 5Ø B8 ØØ ØØ
[RS5]  683: 5Ø E8 88 E1 E8 59 E1 5B FF 37 B8 Ø1 ØØ 5Ø E8 4F
[ØF5]  684: E1 58 5B 89 Ø7 B8 3Ø ØØ 5Ø 5B FF 37 B8 Ø4 ØØ 5Ø
[GH5]  685: E8 CB EØ 58 21 CØ 75 Ø3 E9 24 ØØ B8 ØØ ØØ 5Ø ØE
[8F5]  686: E8 12 ØØ 11 4D 61 78 69 6D 61 6C 2Ø 34 2Ø 49 6E
[2S5]  687: 64 69 7A 65 73 E8 68 E1 E8 BD E1 58 E9 DD 5E B8
[ZM5]  688: 1Ø 21 5Ø B8 ØA ØØ 5Ø B8 52 45 5Ø 5B FF 37 E8 2B
[DL5]  689: E1 E8 FC EØ B8 Ø2 ØØ 5Ø B8 3Ø ØØ 5Ø 5B FF 37 E8
[PM5]  69Ø: 1A E1 E8 EB EØ B8 ØØ 45 5Ø 5B FF 37 B8 Ø1 ØØ 5Ø
[5N5]  691: E8 DD EØ 58 5B 89 Ø7 B8 1Ø 21 5Ø B8 ØA ØØ 5Ø B8
[ZN5]  692: 52 45 5Ø 5B FF 37 E8 F3 EØ E8 C4 EØ B8 Ø2 ØØ 5Ø
[8N5]  693: B8 ØØ ØØ 5Ø E8 E5 EØ E8 B6 EØ B8 3Ø ØØ 5Ø 5B FF
[UG5]  694: 37 58 5B 89 Ø7 E8 11 6Ø B8 44 45 5Ø 5B FF 37 B8
[ZL5]  695: 2C ØØ 5Ø E8 4E EØ 58 21 CØ 75 Ø3 E9 Ø6 ØØ E8 F8
[9M5]  696: 5F E9 18 FE 8C D8 Ø5 7C 17 5Ø ØE E8 Ø2 ØØ Ø1 29
[MV5]  697: E8 CD EØ E8 EB EØ E8 59 6D B8 28 45 5Ø 5B FF 37
[6G5]  698: 58 21 CØ 75 Ø3 E9 Ø3 ØØ E9 7Ø Ø2 B8 4A 45 5Ø 5B
[1P5]  699: FF 37 B8 Ø4 ØØ 5Ø E8 ØB EØ 58 21 CØ 75 Ø3 E9 ØF
[5A5]  7ØØ: ØØ B8 58 45 5Ø B8 1Ø ØØ 5Ø 58 5B 89 Ø7 E9 1F ØØ
[YG5]  7Ø1: B8 58 45 5Ø B8 Ø2 ØØ 5Ø B8 4A 45 5Ø 5B FF 37 E8
[VW5]  7Ø2: 8A E3 E8 4E E5 E8 8F E5 E8 EB E5 58 5B 89 Ø7 B8
[2N5]  7Ø3: 1Ø 21 5Ø B8 ØA ØØ 5Ø B8 52 45 5Ø 5B FF 37 E8 3B
[UH5]  7Ø4: EØ E8 ØC EØ B8 Ø2 ØØ 5Ø B8 ØØ ØØ 5Ø E8 2D EØ E8
[3L5]  7Ø5: FE DF 5B FF 37 8F Ø6 3Ø ØØ B8 Ø1 ØØ 5Ø 8F Ø6 5A
[ZF5]  7Ø6: 45 B8 Ø1 ØØ 5Ø E8 32 EØ 8F Ø6 5C 45 FF 36 5C 45
[TP5]  7Ø7: E8 31 E1 58 8Ø E4 FF 37 E8 26 27 2C E9 ØF ØØ FF
[7N5]  7Ø8: 36 3Ø ØØ FF 36 5C 45 E8 C6 DF 8F Ø6 3Ø ØØ FF 36
[EV5]  7Ø9: 3Ø ØØ FF 36 5A 45 EB ØØ E8 8F DF EB Ø3 E8 77 DF
[1D5]  71Ø: 58 21 CØ 75 Ø3 E9 7E ØØ B8 Ø8 45 5Ø B8 1Ø 21 5Ø
[TS5]  711: B8 ØA ØØ 5Ø B8 52 45 5Ø 5B FF 37 E8 BE DF E8 8F
[CN5]  712: DF B8 Ø2 ØØ 5Ø B8 3Ø ØØ 5Ø 5B FF 37 E8 AD DF E8
[1K5]  713: 7E DF 5B FF 37 58 5B 89 Ø7 B8 1Ø 21 5Ø B8 ØA ØØ
[4P5]  714: 5Ø B8 52 45 5Ø 5B FF 37 E8 91 DF E8 62 DF B8 Ø2
[WR5]  715: ØØ 5Ø B8 3Ø ØØ 5Ø 5B FF 37 E8 8Ø DF E8 51 DF B8
[ØP5]  716: 58 45 5Ø 5B FF 37 58 5B 89 Ø7 B8 58 45 5Ø B8 58
[BR5]  717: 45 5Ø 5B FF 37 B8 Ø8 45 5Ø 5B FF 37 E8 5D DF 58
[8P5]  718: 5B 89 Ø7 E9 59 FF B8 4A 45 5Ø 5B FF 37 B8 Ø4 ØØ
[6K5]  719: 5Ø E8 DØ DE 58 21 CØ 75 Ø3 E9 3C ØØ B8 8Ø 1F 5Ø
[5S5]  72Ø: B8 Ø2 ØØ 5Ø B8 52 45 5Ø 5B FF 37 E8 2E DF E8 FF
[WK5]  721: DE B8 18 45 5Ø 5B FF 37 58 5B 89 Ø7 B8 18 45 5Ø
[6U5]  722: B8 18 45 5Ø 5B FF 37 B8 58 45 5Ø 5B FF 37 E8 DF
[CF5]  723: DE 58 5B 89 Ø7 E9 8F ØØ B8 8Ø 1F 5Ø B8 Ø2 ØØ 5Ø
[AS5]  724: B8 52 45 5Ø 5B FF 37 E8 F2 DE E8 C3 DE B8 ØC 45
[6R5]  725: 5Ø 5B FF 77 Ø2 FF 37 E8 13 E9 58 5B 89 Ø7 B8 ØC
[3N5]  726: 45 5Ø B8 ØC 45 5Ø 5B FF 77 Ø2 FF 37 B8 58 45 5Ø
[ØT5]  727: 5B FF 37 E8 F6 E1 E8 B5 E3 58 5A 5B 89 Ø7 89 57
[1M5]  728: Ø2 B8 ØC 45 5Ø 5B FF 77 Ø2 FF 37 B8 7F 9Ø 5Ø B8
[8G5]  729: ØØ FF 5Ø E8 68 E4 58 21 CØ 75 Ø3 E9 29 ØØ B8 ØØ
[LD5]  73Ø: ØØ 5Ø ØE E8 17 ØØ 16 56 61 72 69 61 62 6C 65 6E
[RM5]  731: 73 7Ø 65 69 63 68 65 72 2Ø 76 6F 6C 6C E8 AØ DE
[MN5]  732: E8 F5 DE 58 E9 15 5C B8 52 45 5Ø B8 52 45 5Ø 5B
[4M5]  733: FF 37 B8 Ø1 ØØ 5Ø E8 37 DE 58 5B 89 Ø7 B8 52 45
[PN5]  734: 5Ø 5B FF 37 B8 7E 1F 5Ø 5B FF 37 E8 BØ DD 58 21
[XD5]  735: CØ 75 Ø3 E9 25 ØØ B8 ØØ ØØ 5Ø ØE E8 13 ØØ 12 5A
[3D5]  736: 75 2Ø 76 69 65 6C 65 2Ø 56 61 72 69 61 62 6C 65
[FU5]  737: 6E E8 4C DE E8 A1 DE 58 E9 C1 5B B8 44 45 5Ø 5B
[CS5]  738: FF 37 B8 2C ØØ 5Ø E8 9B DD 58 21 CØ 75 Ø3 E9 Ø3
[BL5]  739: ØØ E9 2E FA E9 72 34 B8 3Ø 45 5Ø B8 44 45 5Ø 5B
[LS5]  74Ø: FF 37 B8 8D ØØ 5Ø E8 7B DD 58 5B 89 Ø7 B8 3Ø 45
[PG5]  741: 5Ø 5B FF 37 58 21 CØ 75 Ø3 E9 Ø3 ØØ E9 32 ØØ B8
[RP5]  742: 44 45 5Ø 5B FF 37 B8 ØE ØØ 5Ø E8 57 DD 58 21 CØ
[7G5]  743: 75 Ø3 E9 Ø3 ØØ E9 65 ØØ B8 44 45 5Ø 5B FF 37 B8
[6H5]  744: 89 ØØ 5Ø E8 77 DD 58 21 CØ 75 Ø3 E9 Ø3 ØØ E9 82
[2P5]  745: ØØ E8 E5 5C B8 44 45 5Ø 5B FF 37 B8 ØE ØØ 5Ø E8
[RF5]  746: 5B DD 58 21 CØ 75 Ø3 E9 33 ØØ B8 ØØ ØØ 5Ø ØE E8
[ØK5]  747: 21 ØØ 2Ø 5A 65 69 6C 65 6E 6E 75 6D 6D 65 72 2Ø
[BD5]  748: 6F 64 65 72 2Ø 4C 61 62 65 6C 2Ø 65 72 77 61 72
[6Ø6]  749: 74 65 74 E8 8A DD E8 DF DD 58 E9 FF 5A E8 67 72
[NG5]  75Ø: B8 3Ø 45 5Ø 5B FF 37 58 21 CØ 75 Ø3 E9 ØF ØØ B8
[2K5]  751: 22 45 5Ø B8 E8 ØØ 5Ø 58 5B 89 Ø7 E9 ØC ØØ B8 22
[YM5]  752: 45 5Ø B8 E9 ØØ 5Ø 58 5B 89 Ø7 E8 74 63 E8 6E 6A
[MU5]  753: E9 96 33 B8 44 45 5Ø 5B FF 37 B8 8F ØØ 5Ø E8 DC
[KK5]  754: DC 58 21 CØ 75 Ø3 E9 Ø3 ØØ E9 18 ØØ E8 CE 5B B8
[WM5]  755: 44 45 5Ø 5B FF 37 58 21 CØ 75 Ø3 E9 Ø3 ØØ E9 EB
[ØP5]  756: FF E9 65 33 B8 44 45 5Ø 5B FF 37 B8 8E ØØ 5Ø E8
[ZH5]  757: AB DC 58 21 CØ 75 Ø3 E9 Ø3 ØØ E9 15 ØØ E8 19 5C
[ZF5]  758: B8 22 45 5Ø B8 C3 ØØ 5Ø 58 5B 89 Ø7 E8 12 63 E9
[2N5]  759: 37 33 B8 44 45 5Ø 5B FF 37 B8 8B ØØ 5Ø E8 7D DC
[4M5]  76Ø: 58 21 CØ 75 Ø3 E9 Ø3 ØØ E9 63 Ø1 E8 EB 5B E8 D1
[SM5]  761: 33 B8 4A 45 5Ø 5B FF 37 E8 Ø8 ØØ B5 2F 76 2F 7F
[VL5]  762: 89 A6 2F E8 ØB E2 B8 Ø8 45 5Ø B8 31 CØ 5Ø 58 5B
[NF5]  763: 89 Ø7 E8 3B 63 B8 Ø8 45 5Ø B8 5Ø 5Ø 58 5B 89
[4D5]  764: Ø7 E8 2C 63 B8 3Ø 45 5Ø B8 57 ØØ 5Ø 58 5B 89 Ø7
[5L5]  765: E8 F3 63 E9 ØF ØØ B8 3Ø 45 5Ø B8 66 ØØ 5Ø 58 5B
[EK5]  766: 89 Ø7 E8 E1 63 B8 44 45 5Ø 5B FF 37 B8 89 ØØ 5Ø
[HY5]  767: E8 ØA DC B8 44 45 5Ø 5B FF 37 B8 CD ØØ 5Ø E8 FC
[4H5]  768: DB E8 22 DC 58 21 CØ 75 Ø3 E9 2A ØØ B8 ØØ ØØ 5Ø
[VC5]  769: ØE E8 18 ØØ 17 54 48 45 4E 2Ø 6F 64 65 72 2Ø 47
```

```
[GH5]  77Ø: 4F 54 4F 2Ø 65 72 77 61 72 74 65 74 E8 31 DC E8
[5R5]  771: 86 DC 58 E9 A6 59 B8 5E 45 5Ø B8 E2 28 5Ø B8 Ø2
[PU5]  772: ØØ 5Ø B8 34 45 5Ø 5B FF 37 E8 FØ DB E8 C1 DB 5B
[UR5]  773: FF 37 58 5B 89 Ø7 E8 A7 6C B8 CA 2C 5Ø B8 Ø2 ØØ
[CR5]  774: 5Ø B8 6Ø 45 5Ø 5B FF 37 E8 D1 DB E8 A2 DB B8 34
[1H5]  775: 45 5Ø 5B FF 37 58 5B 89 Ø7 B8 6Ø 45 5Ø B8 6Ø 45
[FM5]  776: 5Ø 5B FF 37 B8 Ø1 ØØ 5Ø E8 85 DB 58 5B 89 Ø7 B8
[LM5]  777: 34 45 5Ø B8 34 45 5Ø 5B FF 37 B8 Ø1 ØØ 5Ø E8 6F
[KK5]  778: DB 58 5B 89 Ø7 B8 34 45 5Ø 5B FF 37 B8 EØ 28 5Ø
[ZM5]  779: 5B FF 37 E8 E8 DA 58 21 CØ 75 Ø3 E9 2D ØØ B8 ØØ
[UE5]  78Ø: ØØ 5Ø ØE E8 1B ØØ 1A 5A 75 2Ø 76 69 65 6C 65 2Ø
[DG5]  781: 49 46 2F 54 48 45 4E 2D 53 63 68 6C 65 69 66 65
[LW5]  782: 6E E8 7C DB E8 D1 DB 58 E9 F1 58 E9 BB 31 B8 44
[DM5]  783: 45 5Ø 5B FF 37 B8 A1 ØØ 5Ø E8 Ø1 DB 58 21 CØ 75
[DM5]  784: Ø3 E9 Ø3 ØØ E9 BA ØØ E8 6F 5A B8 22 45 5Ø B8 E9
[1L5]  785: ØØ 5Ø 58 5B 89 Ø7 E8 68 61 B8 Ø8 45 5Ø B8 DE ØF
[TK5]  786: 5Ø B8 Ø2 ØØ 5Ø B8 32 45 5Ø 5B FF 37 E8 ØD DB E8
[LN5]  787: DE DA 5B FF 37 58 5B 89 Ø7 E8 42 68 B8 6Ø 45 5Ø
[KN5]  788: B8 6Ø 45 5Ø 5B FF 37 B8 Ø1 ØØ 5Ø E8 CD DA 58 5B
[SK5]  789: 89 Ø7 B8 6Ø 45 5Ø 5B FF 37 B8 ØØ ØØ 5Ø E8 51 DA
[8C5]  79Ø: 58 21 CØ 75 Ø3 E9 1F ØØ B8 ØØ ØØ 5Ø ØE E8 ØD ØØ
[4M5]  791: ØC 45 4C 53 45 2Ø 6F 68 6E 65 2Ø 49 46 E8 EØ DA
[2N5]  792: E8 35 DB 58 E9 55 58 B8 E2 28 5Ø B8 Ø2 ØØ 5Ø B8
[YR5]  793: CA 2C 5Ø B8 Ø2 ØØ 5Ø B8 6Ø 45 5Ø 5B FF 37 E8 9B
[CV5]  794: DA E8 6C DA 5B FF 37 E8 92 DA E8 63 DA B8 14 45
[BT5]  795: 5Ø 5B FF 77 Ø2 FF 37 E8 B3 E4 58 5B 89 Ø7 E9 8E
[ØM5]  796: F1 B8 44 45 5Ø 5B FF 37 B8 82 ØØ 5Ø E8 2E DA 58
[WK5]  797: 21 CØ 75 Ø3 E9 Ø3 ØØ E9 1E Ø6 E8 9C 59 B8 28 45
[HF5]  798: 5Ø 5B FF 37 58 21 CØ 75 Ø3 E9 47 ØØ B8 46 32 5Ø
[1P5]  799: B8 Ø2 ØØ 5Ø B8 36 45 5Ø 5B FF 37 E8 3E DA E8 ØF
[RL5]  79Ø: DA 5B FF 37 B8 ØØ ØØ 5Ø E8 B9 D9 58 21 CØ 75 Ø3
[7G5]  7Ø1: E9 2Ø ØØ B8 ØØ ØØ 5Ø ØE E8 ØE ØØ ØD 46 4F 52 2Ø
[4P5]  7Ø2: 6F 68 6E 65 2Ø 4E 45 58 54 E8 34 DA E8 89 DA 58
[9S5]  7Ø3: E9 A9 57 E8 D3 67 B8 62 45 5Ø B8 4A 45 5Ø 5B FF
[VC5]  7Ø4: 37 58 5B 89 Ø7 B8 AA 32 5Ø B8 Ø2 ØØ 5Ø B8 36 45
[AX5]  7Ø5: 5Ø 5B FF 37 E8 E5 D9 E8 B6 D9 B8 5E 45 5Ø 5B FF
[HG5]  7Ø6: 37 58 5B 89 Ø7 B8 44 45 5Ø 5B FF 37 B8 E7 ØØ 5Ø
[BK5]  7Ø7: E8 8A D9 58 21 CØ 75 Ø3 E9 1F ØØ B8 ØØ ØØ 5Ø ØE
[LD5]  7Ø8: E8 ØD ØØ ØC 27 3D 27 2Ø 65 72 77 61 72 74 65 74
[RV5]  7Ø9: E8 CD D9 E8 22 DA 58 E9 42 57 E8 DC 58 E8 C2 3Ø
[SK5]  71Ø: B8 4C 45 5Ø B8 62 45 5Ø 5B FF 37 58 5B 89 Ø7 E8
[8N5]  711: 8A 65 B8 5E 45 5Ø B8 AA 32 5Ø B8 Ø2 ØØ 5Ø B8 36
[EV5]  712: 45 5Ø 5B FF 37 E8 74 D9 E8 45 D9 5B FF 37 58 5B
[ØM5]  713: 89 Ø7 E8 E2 68 B8 44 45 5Ø 5B FF 37 B8 CC ØØ 5Ø
[2K5]  714: E8 1A D9 58 21 CØ 75 Ø3 E9 1E ØØ B8 ØØ ØØ 5Ø ØE
[CG5]  715: E8 ØC ØØ ØB 54 4F 2Ø 65 72 77 61 72 74 65 74 E8
[NT5]  716: 5E D9 E8 B3 D9 58 E9 D3 56 E8 6D 58 E8 53 3Ø B8
[PR5]  717: 4C 45 5Ø B8 62 45 5Ø 5B FF 37 58 5B 89 Ø7 E8 1B
[UH5]  718: 65 B8 28 45 5Ø 5B FF 37 B8 ØØ ØØ 5Ø E8 95 D8 58
[FF5]  719: 21 CØ 75 Ø3 E9 53 ØØ B8 ØE 33 5Ø B8 Ø2 ØØ 5Ø B8
[HT5]  72Ø: 36 45 5Ø 5B FF 37 E8 F3 D8 E8 C4 D8 B8 ØC 45 5Ø
[FM5]  721: 5B FF 77 Ø2 FF 37 E8 14 E3 58 5B 89 Ø7 B8 ØC 45
[KH5]  722: 5Ø B8 ØC 45 5Ø 5B FF 77 Ø2 FF 37 B8 Ø2 ØØ 5Ø B8
[XØ6]  723: 62 45 5Ø 5B FF 37 E8 F3 DB E8 B7 DD E8 F8 DD E8
[TS5]  724: AC DD 58 5A 5B 89 Ø7 89 57 Ø2 B8 5E 45 5Ø B8 ØE
[HL5]  725: 33 5Ø B8 Ø2 ØØ 5Ø B8 36 45 5Ø 5B FF 37 E8 9C D8
[4U5]  726: E8 6D D8 5B FF 37 58 5B 89 Ø7 E8 ØA 68 B8 44 45
[KK5]  727: 5Ø 5B FF 37 B8 CF ØØ 5Ø E8 Ø9 D8 58 21 CØ 75 Ø3
[7H5]  728: E9 Ø3 ØØ E9 1E ØØ B8 Ø8 45 5Ø B8 Ø1 ØØ 5Ø 58 5B
[HG5]  729: 89 Ø7 E8 99 5F B8 4A 45 5Ø B8 Ø1 ØØ 5Ø 58 5B 89
[DL5]  73Ø: Ø7 E9 Ø6 ØØ E8 92 57 E8 78 2F B8 4C 45 5Ø B8 62
[CH5]  731: 45 5Ø 5B FF 37 58 5B 89 Ø7 E8 4Ø 64 B8 28 45 5Ø
[UL5]  732: 5B FF 37 B8 ØØ ØØ 5Ø E8 BA D7 58 21 CØ 75 Ø3 E9
[YG5]  733: 53 ØØ B8 72 33 5Ø B8 Ø2 ØØ 5Ø B8 36 45 5Ø 5B FF
[FV5]  734: 37 E8 18 D8 E8 E9 D7 B8 ØC 45 5Ø 5B FF 77 Ø2 FF
[SG5]  735: 37 E8 39 E2 58 5B 89 Ø7 B8 ØC 45 5Ø B8 ØC 45 5Ø
[FN5]  736: 5B FF 77 Ø2 FF 37 B8 Ø2 ØØ 5Ø B8 62 45 5Ø 5B FF
[PZ5]  737: 37 E8 18 DB E8 DC DC E8 1D DD E8 D1 DC 58 5A 5B
[BB5]  738: 89 Ø7 89 57 Ø2 B8 5E 45 5Ø B8 72 33 5Ø B8 Ø2 ØØ
[BS5]  739: 5Ø B8 36 45 5Ø 5B FF 37 E8 C1 D7 E8 92 D7 5B FF
[TL5]  74Ø: 37 58 5B 89 Ø7 E8 2F 67 E8 CF 67 B8 62 45 5Ø 5B
[5R5]  741: FF 37 E8 Ø6 ØØ 6E 34 7D 34 7F 89 E8 13 DD B8 3Ø
[VH5]  742: 45 5Ø B8 1A ØØ 5Ø 58 5B 89 Ø7 E9 ØC ØØ B8 3Ø 45
[CE5]  743: 5Ø B8 59 ØØ 5Ø 58 5B 89 Ø7 E8 ØA 5F B8 Ø8 45 5Ø
[DK5]  744: B8 58 89 5Ø 58 5B 89 Ø7 E8 25 5E B8 Ø8 45 5Ø B8
[AM5]  745: E4 Ø5 5Ø 58 5B 89 Ø7 E8 16 5E B8 22 45 5Ø B8 2E
[AH5]  746: ØØ 5Ø 58 5B 89 Ø7 E8 98 5D B8 Ø8 45 5Ø B8 88 26
[BH5]  747: 5Ø 58 5B 89 Ø7 E8 F8 5D B8 Ø8 45 5Ø B8 1Ø 45 5Ø
[KM5]  748: 5B FF 77 Ø2 FF 37 B8 D6 33 5Ø B8 Ø2 ØØ 5Ø B8 36
[PX5]  749: 45 5Ø 5B FF 37 E8 24 D7 E8 F5 D6 5B FF 37 E8 8E
[SS5]  75Ø: E1 E8 ØA DC E8 46 E1 58 5B 89 Ø7 E8 C2 5D B8 22
[8L5]  751: 45 5Ø B8 E9 ØØ 5Ø 58 5B 89 Ø7 E8 44 5D B8 Ø8 45
[MC5]  752: 5Ø B8 3A 34 5Ø B8 Ø2 ØØ 5Ø B8 36 45 5Ø 5B FF 37
[WT5]  753: E8 E9 D6 E8 BA D6 5B FF 37 58 5B 89 Ø7 E8 1E 64
[ØH5]  754: B8 9E 34 5Ø B8 Ø2 ØØ 5Ø B8 36 45 5Ø 5B FF 37 E8
[8T5]  755: CA D6 E8 9B D6 B8 14 45 5Ø 5B FF 77 Ø2 FF 37 E8
[ØS5]  756: EB EØ 58 5B 89 Ø7 B8 5E 45 5Ø B8 AA 32 5Ø B8 Ø2
[2S5]  757: ØØ 5Ø B8 36 45 5Ø 5B FF 37 E8 AØ D6 E8 71 D6 5B
[9M5]  758: FF 37 58 5B 89 Ø7 E8 B1 66 B8 5E 45 5Ø B8 72 33
[SK5]  759: 5Ø B8 Ø2 ØØ 5Ø B8 36 45 5Ø 5B FF 37 E8 7D D6 E8
[WM5]  76Ø: 4E D6 5B FF 37 58 5B 89 Ø7 E8 8E 66 B8 62 45 5Ø
[DU5]  761: 5B FF 37 E8 Ø6 ØØ AF 35 BE 35 7F 89 E8 D2 DB B8
[FG5]  762: 3Ø 45 5Ø B8 Ø8 ØØ 5Ø 58 5B 89 Ø7 E9 ØC ØØ B8 3Ø
[LM5]  763: 45 5Ø B8 48 ØØ 5Ø 58 5B 89 Ø7 E8 C9 5D B8 5E 45
[4E5]  764: 5Ø B8 AA 32 5Ø B8 Ø2 ØØ 5Ø B8 36 45 5Ø 5B FF 37
[5U5]  765: E8 29 D6 E8 FA D5 5B FF 37 58 5B 89 Ø7 E8 97 65
[KG5]  766: B8 3A 34 5Ø B8 Ø2 ØØ 5Ø B8 36 45 5Ø 5B FF 37 E8
[ØT5]  767: ØA D6 E8 DB D5 B8 14 45 5Ø 5B FF 77 Ø2 FF 37 E8
[UP5]  768: 2B EØ 58 5B 89 Ø7 B8 5E 45 5Ø B8 AA 32 5Ø B8 Ø2
[SS5]  769: ØØ 5Ø B8 36 45 5Ø 5B FF 37 E8 EØ D5 E8 B1 D5 5B
[9M5]  77Ø: FF 37 58 5B 89 Ø7 E8 F1 65 B8 5E 45 5Ø B8 ØE 33
[2L5]  771: 5Ø B8 Ø2 ØØ 5Ø B8 36 45 5Ø 5B FF 37 E8 BD D5 E8
[GP5]  772: 8E D5 5B FF 37 58 5B 89 Ø7 E8 CE 65 B8 Ø8 45 5Ø
[FL5]  773: B8 EB ØØ 5Ø 58 5B 89 Ø7 E8 55 5C B8 D6 33 5Ø B8
[GR5]  774: Ø2 ØØ 5Ø B8 36 45 5Ø 5B FF 37 E8 8F D5 E8 6Ø D5
[5N5]  775: B8 14 45 5Ø 5B FF 77 Ø2 FF 37 B8 Ø1 ØØ 5Ø E8 AB
[RV5]  776: D8 E8 63 DA E8 67 DA E8 A3 DF 58 5B 89 Ø7 B8 62
[7N5]  777: 45 5Ø 5B FF 37 E8 Ø6 ØØ B1 36 CØ 36 7F 89 E8 DØ
[BC5]  778: DA B8 3Ø 45 5Ø B8 Ø6 ØØ 5Ø 58 5B 89 Ø7 E9 ØC ØØ
[FG5]  779: B8 3Ø 45 5Ø B8 56 ØØ 5Ø 58 5B 89 Ø7 E8 C7 5C B8
[WM5]  78Ø: Ø8 45 5Ø B8 EB Ø3 5Ø 58 5B 89 Ø7 E8 E2 5B B8 62
```

```
[6K5]  881: 45 5Ø 5B FF 37 E8 Ø6 ØØ F1 36 ØØ 37 7F 89 E8 9Ø
[AC5]  882: DA B8 3Ø 45 5Ø B8 Ø5 ØØ 5Ø 58 5B 89 Ø7 E9 ØC ØØ
[2G5]  883: B8 3Ø 45 5Ø B8 55 ØØ 5Ø 58 5B 89 Ø7 E8 87 5C B8
[DH5]  884: 5E 45 5Ø B8 46 32 5Ø B8 Ø2 ØØ 5Ø B8 36 45 5Ø 5B
[FW5]  885: FF 37 E8 E7 D4 E8 B8 D4 5B FF 37 58 5B 89 Ø7 E8
[YK5]  886: 9E 65 B8 Ø2 35 5Ø B8 Ø2 ØØ 5Ø B8 64 45 5Ø 5B FF
[AU5]  887: 37 E8 C8 D4 E8 99 D4 B8 36 45 5Ø 5B FF 37 58 5B
[LH5]  888: 89 Ø7 B8 64 45 5Ø B8 64 45 5Ø 5B FF 37 B8 Ø1 ØØ
[SF5]  889: 5Ø E8 7C D4 58 5B 89 Ø7 B8 36 45 5Ø B8 36 45 5Ø
[SM5]  89Ø: 5B FF 37 B8 Ø1 ØØ 5Ø E8 66 D4 58 5B 89 Ø7 B8 36
[8R5]  891: 45 5Ø 5B FF 37 B8 44 32 5Ø 5B FF 37 E8 DF D3 58
[AG5]  892: 21 CØ 75 Ø3 E9 2E ØØ B8 ØØ ØØ 5Ø ØE E8 1C ØØ 1B
[MF5]  893: 5A 75 2Ø 76 69 65 6C 65 2Ø 46 4F 52 2F 4E 45 58
[LM5]  894: 54 2D 53 63 68 6C 65 69 66 65 6E E8 72 D4 E8 C7
[ZR5]  895: D4 58 E9 E7 51 E9 B1 2A B8 44 45 5Ø 5B FF 37 B8
[VF5]  896: 83 ØØ 5Ø E8 F7 D3 58 21 CØ 75 Ø3 E9 Ø3 ØØ E9 12
[7G5]  897: Ø1 E8 65 53 B8 64 45 5Ø B8 64 45 5Ø 5B FF 37 B8
[5N5]  898: Ø1 ØØ 5Ø E8 F5 D3 58 5B 89 Ø7 B8 64 45 5Ø 5B FF
[CC5]  899: 37 B8 ØØ ØØ 5Ø E8 79 D3 58 21 CØ 75 Ø3 E9 2Ø ØØ
[AE5]  9ØØ: B8 ØØ ØØ 5Ø ØE E8 ØE ØØ ØD 4E 45 58 54 2Ø 6F 68
[9M5]  9Ø1: 6E 65 2Ø 46 4F 52 E8 Ø7 D4 E8 5C D4 58 E9 7C 51
[AG5]  9Ø2: B8 22 45 5Ø B8 E9 ØØ 5Ø 58 5B 89 Ø7 E8 12 5A B8
[3H5]  9Ø3: Ø8 45 5Ø B8 9E 34 5Ø B8 Ø2 ØØ 5Ø B8 Ø2 35 5Ø B8
[5S5]  9Ø4: Ø2 ØØ 5Ø B8 64 45 5Ø 5B FF 37 E8 AF D3 E8 8Ø D3
[CT5]  9Ø5: 5B FF 37 E8 A6 D3 E8 77 D3 5B FF 37 58 5B 89 Ø7
[ØK5]  9Ø6: E8 DB 6Ø B8 46 32 5Ø B8 Ø2 ØØ 5Ø B8 Ø2 35 5Ø B8
[VP5]  9Ø7: Ø2 ØØ 5Ø B8 64 45 5Ø 5B FF 37 E8 7F D3 E8 5Ø D3
[ZP5]  9Ø8: 5B FF 37 E8 76 D3 E8 47 D3 B8 14 45 5Ø 5B FF 77
[HR5]  9Ø9: Ø2 FF 37 E8 97 DD 58 5B 89 Ø7 B8 44 45 5Ø 5B FF
[DK5]  91Ø: 37 B8 41 ØØ 5Ø E8 C9 D2 B8 44 45 5Ø 5B FF 37 B8
[VL5]  911: 5A ØØ 5Ø E8 A8 D2 E8 38 D3 58 21 CØ 75 Ø3 E9 Ø3
[XP5]  912: ØØ E9 A5 29 E8 Ø2 61 B8 44 45 5Ø 5B FF 37 B8 2C
[LR5]  913: ØØ 5Ø E8 AF D2 58 21 CØ 75 Ø3 E9 Ø3 ØØ E9 F1 FE
[YR5]  914: E9 86 29 B8 44 45 5Ø 5B FF 37 B8 81 ØØ 5Ø E8 CC
[MF5]  915: D2 58 21 CØ 75 Ø3 E9 Ø3 ØØ E9 15 ØØ E8 3A 52 B8
[XH5]  916: 3Ø 45 5Ø B8 Ø1 ØØ 5Ø 58 5B 89 Ø7 E8 78 5A E9 58
[GL5]  917: 29 B8 44 45 5Ø 5B FF 37 B8 B1 ØØ 5Ø E8 9E D2 58
[HF5]  918: 21 CØ 75 Ø3 E9 Ø3 ØØ E9 5B Ø1 E8 9Ø 51 B8 44 45
[EH5]  919: 5Ø 5B FF 37 B8 E9 ØØ 5Ø E8 82 D2 58 21 CØ 75 Ø3
[SM5]  92Ø: E9 Ø3 ØØ E9 54 5F E8 FØ 51 B8 28 45 5Ø 5B FF 37
[5C5]  921: 58 21 CØ 75 Ø3 E9 49 ØØ B8 68 35 5Ø B8 Ø2 ØØ 5Ø
[VP5]  922: B8 38 45 5Ø 5B FF 37 E8 92 D2 E8 63 D2 5B FF 37
[YF5]  923: B8 ØØ ØØ 5Ø E8 ØD D2 58 21 CØ 75 Ø3 E9 22 ØØ B8
[ØG5]  924: ØØ ØØ 5Ø ØE E8 1Ø ØØ ØF 57 48 49 4C 45 2Ø 6F 68
[AV5]  925: 6E 65 2Ø 57 45 4E 44 E8 86 D2 E8 DB D2 58 E9 FB
[WG5]  926: 4F B8 CC 35 5Ø B8 Ø2 ØØ 5Ø B8 38 45 5Ø 5B FF 37
[FT5]  927: E8 49 D2 E8 1A D2 B8 14 45 5Ø 5B FF 77 Ø2 FF 37
[ZT5]  928: E8 6A DC 58 5B 89 Ø7 E8 39 5F B8 5E 45 5Ø B8 68
[7K5]  929: 35 5Ø B8 Ø2 ØØ 5Ø B8 38 45 5Ø 5B FF 37 E8 1C D2
[RT5]  93Ø: E8 ED D1 5B FF 37 58 5B 89 Ø7 E8 D3 62 B8 3Ø 36
[EL5]  931: 5Ø B8 Ø2 ØØ 5Ø B8 66 45 5Ø 5B FF 37 E8 FD D1 E8
[RN5]  932: CE D1 B8 38 45 5Ø 5B FF 37 58 5B 89 Ø7 B8 66 45
[LH5]  933: 5Ø B8 66 45 5Ø 5B FF 37 B8 Ø1 ØØ 5Ø E8 B1 D1 58
[RM5]  934: 5B 89 Ø7 B8 38 45 5Ø B8 38 45 5Ø 5B FF 37 B8 Ø1
[RL5]  935: ØØ 5Ø E8 9B D1 58 5B 89 Ø7 B8 38 45 5Ø 5B FF 37
[9H5]  936: B8 66 35 5Ø 5B FF 37 E8 14 D1 58 21 CØ 75 Ø3 E9
[KE5]  937: 3Ø ØØ B8 ØØ ØØ 5Ø ØE E8 1E ØØ 1D 5A 75 2Ø 76 69
[SF5]  938: 65 6C 65 2Ø 57 48 49 4C 45 2F 57 45 4E 44 2D 53
[TS5]  939: 63 68 6C 65 69 66 65 6E E8 A5 D1 E8 FA D1 58 E9
[7M5]  94Ø: 1A 4F E9 E4 27 B8 44 45 5Ø 5B FF 37 B8 B2 ØØ 5Ø
[RN5]  941: E8 2A D1 58 21 CØ 75 Ø3 E9 Ø3 ØØ E9 CE ØØ E8 98
[7E5]  942: 5Ø B8 66 45 5Ø B8 66 45 5Ø 5B FF 37 B8 Ø1 ØØ 5Ø
[HP5]  943: E8 28 D1 58 5B 89 Ø7 B8 66 45 5Ø 5B FF 37 B8 ØØ
[FF5]  944: ØØ 5Ø E8 AC DØ 58 21 CØ 75 Ø3 E9 22 ØØ B8 ØØ ØØ
[PG5]  945: 5Ø ØE E8 1Ø ØØ ØF 57 45 4E 44 2Ø 6F 68 6E 65 2Ø
[ZR5]  946: 57 48 49 4C 45 E8 38 D1 E8 8D D1 58 E9 AD 4E B8
[YH5]  947: 22 45 5Ø B8 E9 ØØ 5Ø 58 5B 89 Ø7 E8 43 57 B8 Ø8
[PG5]  948: 45 5Ø B8 CC 35 5Ø B8 Ø2 ØØ 5Ø B8 3Ø 36 5Ø B8 Ø2
[8S5]  949: ØØ 5Ø B8 66 45 5Ø 5B FF 37 E8 EØ DØ E8 B1 DØ 5B
[CV5]  95Ø: FF 37 E8 D7 DØ E8 A8 DØ 5B FF 37 58 5B 89 Ø7 E8
[FG5]  951: ØC 5E B8 68 35 5Ø B8 Ø2 ØØ 5Ø B8 3Ø 36 5Ø B8 Ø2
[KR5]  952: ØØ 5Ø B8 66 45 5Ø 5B FF 37 E8 BØ DØ E8 81 DØ 5B
[DP5]  953: FF 37 E8 A7 DØ E8 78 DØ B8 14 45 5Ø 5B FF 77 Ø2
[SS5]  954: FF 37 E8 C8 DA 58 5B 89 Ø7 E9 FD 26 B8 44 45 5Ø
[CK5]  955: 5B FF 37 B8 CØ ØØ 5Ø E8 43 DØ 58 21 CØ 75 Ø3 E9
[CF5]  956: Ø3 ØØ E9 15 ØØ E8 B1 4F B8 3Ø 45 5Ø B8 17 ØØ 5Ø
[FT5]  957: 58 5B 89 Ø7 E8 EF 57 E9 CF 26 B8 44 45 5Ø 5B FF
[WD5]  958: 37 B8 CA ØØ 5Ø E8 15 DØ 58 21 CØ 75 Ø3 E9 Ø3 ØØ
[MR5]  959: E9 53 Ø1 E8 83 4F E8 4A 5D B8 4A 45 5Ø 5B FF 37
[6K5]  96Ø: B8 ØØ ØØ 5Ø E8 BD CF 58 21 CØ 75 Ø3 E9 ØF ØØ B8
[XK5]  961: 3Ø 45 5Ø B8 5C ØØ 5Ø 58 5B 89 Ø7 E8 A8 57 B8 44
[VM5]  962: 45 5Ø 5B FF 37 B8 2C ØØ 5Ø E8 98 CF 58 21 CØ 75
[6N5]  963: Ø3 E9 Ø3 ØØ E8 42 4F E8 Ø9 5D B8 4A 45 5Ø 5B FF
[DF5]  964: 37 B8 ØØ ØØ 5Ø E8 7C CF 58 21 CØ 75 Ø3 E9 ØF ØØ
[KG5]  965: B8 3Ø 45 5Ø B8 5D ØØ 5Ø 58 5B 89 Ø7 E8 67 57 B8
[UH5]  966: 3Ø 45 5Ø B8 1B ØØ 5Ø 58 5B 89 Ø7 E8 58 57 B8 44
[GM5]  967: 45 5Ø 5B FF 37 B8 2C ØØ 5Ø E8 81 CF 58 21 CØ 75
[NN5]  968: Ø3 E9 Ø3 ØØ E9 22 26 E8 EF 4E E8 B6 5C B8 4A 45
[ZG5]  969: 5Ø 5B FF 37 B8 ØØ ØØ 5Ø E8 29 CF 58 21 CØ 75 Ø3
[ML5]  97Ø: E9 ØF ØØ B8 3Ø 45 5Ø B8 5F ØØ 5Ø 58 5B 89 Ø7 E8
[5L5]  971: 14 57 B8 44 45 5Ø 5B FF 37 B8 2C ØØ 5Ø E8 Ø4 CF
[RM5]  972: 58 21 CØ 75 Ø3 E9 Ø3 ØØ E8 AE 4E E8 75 5C B8 4A
[CN5]  973: 45 5Ø 5B FF 37 B8 ØØ ØØ E8 E8 CE 58 21 CØ 75
[AB5]  974: Ø3 E9 ØF ØØ B8 3Ø 45 5Ø B8 6Ø ØØ 5Ø 58 5B 89 Ø7
[ØS5]  975: E8 D3 56 B8 44 45 5Ø 5B FF 37 B8 2C ØØ 5Ø E8 C3
[3N5]  976: CE 58 21 CØ 75 Ø3 E9 Ø3 ØØ E8 6D 4E E8 34 5C B8
[FP5]  977: 4A 45 5Ø 5B FF 37 B8 ØØ ØØ 5Ø E8 A7 CE 58 21 CØ
[CE5]  978: 75 Ø3 E9 ØF ØØ B8 3Ø 45 5Ø B8 61 ØØ 5Ø 58 5B 89
[ME5]  979: Ø7 E8 92 56 B8 3Ø 45 5Ø B8 5E ØØ 5Ø 58 5B 89 Ø7
[HN5]  98Ø: E8 83 56 E9 63 25 B8 44 45 5Ø 5B FF 37 B8 BF ØØ
[WL5]  981: 5Ø E8 A9 CE 58 21 CØ 75 Ø3 E9 Ø3 ØØ E9 E6 ØØ E8
[5P5]  982: 17 4E E8 DE 5B B8 4A 45 5Ø 5B FF 37 B8 ØØ ØØ 5Ø
[5K5]  983: E8 51 CE 58 21 CØ 75 Ø3 E9 ØF ØØ B8 3Ø 45 5Ø B8
[DM5]  984: 62 ØØ 5Ø 58 5B 89 Ø7 E8 3C 56 B8 44 45 5Ø 5B FF
[KF5]  985: 37 B8 2C ØØ 5Ø E8 65 CE 58 21 CØ 75 Ø3 E9 ØF ØØ
[VF5]  986: B8 4A 45 5Ø B8 ØØ ØØ 5Ø 58 5B 89 Ø7 E9 18 ØØ 8C
[6M5]  987: D8 Ø5 7C 17 5Ø ØE E8 Ø2 ØØ Ø1 2C E8 A2 CE E8 CØ
[CU5]  988: CE E8 2E 5B E8 7C 5B B8 4A 45 5Ø 5B FF 37 B8 ØØ
[RN5]  989: ØØ 5Ø E8 EF CD 58 21 CØ 75 Ø3 E9 ØF ØØ B8 3Ø 45
[9E5]  99Ø: 5Ø B8 63 ØØ 5Ø 58 5B 89 Ø7 E8 DA 55 B8 3Ø 45 5Ø
[RK5]  991: B8 1C ØØ 5Ø 58 5B 89 Ø7 E8 CB 55 B8 44 45 5Ø 5B
[TR5]  992: FF 37 B8 2C ØØ 5Ø E8 F4 CD 58 21 CØ 75 Ø3 E9 Ø3
[TP5]  993: ØØ E9 95 24 E8 62 4D E8 29 5B B8 4A 45 5Ø 5B FF
[UE5]  994: 37 B8 ØØ ØØ 5Ø E8 9C CD 58 21 CØ 75 Ø3 E9 Ø3 ØØ
[ØK5]  995: E9 76 24 B8 3Ø 45 5Ø B8 87 ØØ 5Ø 58 5B 89 Ø7 E8
[KG5]  996: 84 55 E9 64 24 B8 44 45 5Ø 5B FF 37 B8 AØ ØØ 5Ø
[KL5]  997: E8 AA CD 58 21 CØ 75 Ø3 E9 Ø3 ØØ E9 18 ØØ E8 18
[9L5]  998: 4D E8 DF 5A B8 3Ø 45 5Ø B8 1D ØØ 5Ø 58 5B 89 Ø7
[MN5]  999: E8 53 55 E9 33 24 B8 44 45 5Ø 5B FF 37 B8 FE ØØ
[8K5]  1ØØØ: 5Ø E8 79 CD 58 21 CØ 75 Ø3 E9 Ø3 ØØ E9 E2 ØØ E8
[8R5]  1ØØ1: 6B 4C B8 44 45 5Ø 5B FF 37 B8 9C ØØ 5Ø E8 5D CD
[KK5]  1ØØ2: 58 21 CØ 75 Ø3 E9 Ø3 ØØ E9 32 13 E8 CB 4C B8 44
[2M5]  1ØØ3: 45 5Ø 5B FF 37 B8 91 ØØ 5Ø E8 41 CD 58 21 CØ 75
[8B5]  1ØØ4: Ø3 E9 21 ØØ B8 ØØ ØØ 5Ø ØE E8 ØF ØØ ØE 5Ø 52 49
[FL5]  1ØØ5: 4E 54 2Ø 65 72 77 61 72 74 65 74 E8 82 CD E8 D7
[NV5]  1ØØ6: CD 58 E9 F7 4A E8 91 4C E8 58 5A B8 4A 45 5Ø 5B
[LP5]  1ØØ7: FF 37 B8 ØØ ØØ 5Ø E8 CB CC 58 21 CØ 75 Ø3 E9 Ø3
[LK5]  1ØØ8: ØØ E9 4C ØØ B8 44 45 5Ø 5B FF 37 B8 CC ØØ 5Ø E8
[6K5]  1ØØ9: EB CC 58 21 CØ 75 Ø3 E9 1E ØØ B8 ØØ ØØ 5Ø ØE E8
[2H5]  1Ø1Ø: ØC ØØ ØB 54 4F 2Ø 65 72 77 61 72 74 65 74 E8 2F
[NV5]  1Ø11: CD E8 84 CD 58 E9 A4 4A E8 3E 4C E8 Ø5 5A B8 3Ø
[NK5]  1Ø12: 45 5Ø B8 1E ØØ 5Ø 58 5B 89 Ø7 E8 79 54 E9 59 23
[9G5]  1Ø13: B8 Ø8 45 5Ø B8 Ø1 ØØ 5Ø 58 5B 89 Ø7 E8 ØF 54 B8
[XL5]  1Ø14: Ø8 45 5Ø B8 17 ØØ 5Ø 58 5B 89 Ø7 E8 ØØ 54 E9 CD
[AR5]  1Ø15: FF B8 44 45 5Ø 5B FF 37 B8 BA ØØ 5Ø E8 7E CC 58
[UN5]  1Ø16: 21 CØ 75 Ø3 E9 Ø3 ØØ E9 E3 Ø2 E8 EC 4B E8 D2 23
[VL5]  1Ø17: B8 4A 45 5Ø 5B FF 37 B8 Ø4 ØØ 5Ø E8 5F CC 58 21
[6C5]  1Ø18: CØ 75 Ø3 E9 25 ØØ B8 ØØ ØØ 5Ø ØE E8 13 ØØ 12 44
[TD5]  1Ø19: 61 74 65 69 6E 61 6D 65 2Ø 65 72 77 61 72 74 65
[FS5]  1Ø2Ø: 74 E8 9C CC E8 F1 CC 58 E9 11 4A B8 44 45 5Ø 5B
[MP5]  1Ø21: FF 37 B8 82 ØØ 5Ø E8 24 CC 58 21 CØ 75 Ø3 E9 ØF
[3A5]  1Ø22: ØØ B8 Ø8 45 5Ø B8 Ø2 ØØ 5Ø 58 5B 89 Ø7 E9 6B Ø1
[WP5]  1Ø23: E8 86 4B B8 44 45 5Ø 5B FF 37 B8 85 ØØ 5Ø E8 C3
[LH5]  1Ø24: CB 58 21 CØ 75 Ø3 E9 12 ØØ E8 6D 4B B8 Ø8 45 5Ø
[1G5]  1Ø25: B8 ØØ ØØ 5Ø 58 5B 89 Ø7 E9 4Ø Ø1 8C D8 Ø5 7C 17
[3M5]  1Ø26: 5Ø ØE E8 Ø1 ØØ ØØ E8 37 CC E8 55 CC B8 44 45 5Ø
[BR5]  1Ø27: 5B FF 37 B8 41 ØØ 5Ø E8 77 CB B8 44 45 5Ø 5B FF
[3K5]  1Ø28: 37 B8 5A ØØ 5Ø E8 56 CB E8 E6 CB 58 21 CØ 75 Ø3
[5L5]  1Ø29: E9 Ø3 ØØ E9 29 ØØ 8C D8 Ø5 7C 17 5Ø 8C D8 Ø5 7C
[3M5]  1Ø3Ø: 17 5Ø B8 ØØ ØØ 5Ø E8 F7 CB B8 44 45 5Ø 5B FF 37
[LJ6]  1Ø31: E8 DE CC E8 AC CC E8 Ø8 CC E8 81 4A E9 AD FF 8C
[LG5]  1Ø32: D8 Ø5 7C 17 5Ø B8 ØØ ØØ 5Ø E8 D4 CB ØE E8 Ø7 ØØ
[1N5]  1Ø33: Ø6 4F 55 54 5Ø 55 54 E8 C6 CB E8 96 CD 58 21 CØ
[GD5]  1Ø34: 75 Ø3 E9 ØF ØØ B8 Ø8 45 5Ø B8 Ø1 ØØ 5Ø 58 5B 89
[UL5]  1Ø35: Ø7 E9 A7 ØØ 8C D8 Ø5 7C 17 5Ø B8 ØØ ØØ 5Ø E8 9F
[FR5]  1Ø36: CB ØE E8 Ø7 ØØ Ø6 52 41 4E 44 4F 4D E8 91 CB E8
[LG5]  1Ø37: 61 CD 58 21 CØ 75 Ø3 E9 ØF ØØ B8 Ø8 45 5Ø B8 Ø2
[5L5]  1Ø38: ØØ 5Ø 58 5B 89 Ø7 E9 72 ØØ 8C D8 Ø5 7C 17 5Ø B8
[CF5]  1Ø39: ØØ ØØ 5Ø E8 6A CB ØE E8 Ø7 ØØ Ø6 41 5Ø 5Ø 45 4E
[8P5]  1Ø4Ø: 44 E8 5C CB E8 2C CD 58 21 CØ 75 Ø3 E9 ØF ØØ B8
[KG5]  1Ø41: Ø8 45 5Ø B8 Ø3 ØØ 5Ø 58 5B 89 Ø7 E9 3D ØØ B8 ØØ
[8F5]  1Ø42: ØØ 5Ø ØE E8 2B ØØ 2A 49 4E 5Ø 55 54 2C 2Ø 4F 55
[5F5]  1Ø43: 54 5Ø 55 54 2C 2Ø 52 41 4E 44 4F 4D 2Ø 6F 64 65
[K95]  1Ø44: 72 2Ø 41 5Ø 5Ø 45 4E 44 2Ø 65 72 77 61 72 74 65
[2S5]  1Ø45: 74 E8 ØC CB E8 61 CB 58 E9 81 48 E8 1Ø 52 E8 1B
[BM5]  1Ø46: 4A E8 35 5E B8 44 45 5Ø 5B FF 37 B8 23 ØØ 5Ø E8
[MK5]  1Ø47: 52 CA 58 21 CØ 75 Ø3 E9 Ø3 ØØ E8 FC 49 B8 44 45
[ØN5]  1Ø48: 5Ø 5B FF 37 B8 3Ø ØØ 5Ø E8 4C CA B8 44 45 5Ø 5B
[RS5]  1Ø49: FF 37 B8 39 ØØ 5Ø E8 51 CA E8 8A CA 58 21 CØ 75
[XE5]  1Ø5Ø: Ø3 E9 16 ØØ B8 44 45 5Ø B8 44 45 5Ø 5B FF 37 B8
[MM5]  1Ø51: 1F ØØ 5Ø E8 65 CA 58 5B 89 Ø7 E8 86 57 B8 44 45
[FK5]  1Ø52: 5Ø 5B FF 37 B8 FF ØØ 5Ø E8 32 CA 58 21 CØ 75 Ø3
[3H5]  1Ø53: E9 12 ØØ B8 Ø8 45 5Ø B8 8Ø ØØ 5Ø 58 5B 89 Ø7 E8
[FL5]  1Ø54: 8C 51 E9 76 ØØ E8 15 49 B8 44 45 5Ø 5B FF 37 B8
[CG5]  1Ø55: 92 ØØ 5Ø E8 Ø7 CA 58 21 CØ 75 Ø3 E9 1F ØØ B8 ØØ
[KE5]  1Ø56: ØØ 5Ø ØE E8 ØD ØØ ØC 4C 45 4E 2Ø 65 72 77 61 72
[GX5]  1Ø57: 74 65 74 E8 4A CA E8 9F CA 58 E9 BF 47 E8 59 49
[WM5]  1Ø58: B8 44 45 5Ø 5B FF 37 B8 E7 ØØ 5Ø E8 CF C9 58 21
[VF5]  1Ø59: CØ 75 Ø3 E9 1F ØØ B8 ØØ ØØ 5Ø ØE E8 ØD ØØ ØC 27
[FH5]  1Ø6Ø: 3D 27 2Ø 65 72 77 61 72 74 65 74 E8 12 CA E8 67
[FN5]  1Ø61: CA 58 E9 87 47 E8 21 49 E8 56 B8 3Ø 45 5Ø B8
[XL5]  1Ø62: 1F ØØ 5Ø 58 5B 89 Ø7 E8 5C 51 E9 3C 2Ø B8 44 45
[ØK5]  1Ø63: 5Ø 5B FF 37 B8 BB ØØ 5Ø E8 82 C9 58 21 CØ 75 Ø3
[AM5]  1Ø64: E9 Ø3 ØØ E9 86 ØØ E8 FØ 48 B8 44 45 5Ø 5B FF 37
[3P5]  1Ø65: B8 3A ØØ 5Ø E8 2D C9 B8 44 45 5Ø 5B FF 37 B8 ØØ
[TK5]  1Ø66: ØØ 5Ø E8 1F C9 E8 89 C9 58 21 CØ 75 Ø3 E9 12 ØØ
[MF5]  1Ø67: B8 3Ø 45 5Ø B8 26 ØØ 5Ø 58 5B 89 Ø7 E8 Ø7 51 E9
[ØP5]  1Ø68: E7 1F B8 44 45 5Ø 5B FF 37 B8 23 ØØ 5Ø E8 F4 C8
[ZG5]  1Ø69: 58 21 CØ 75 Ø3 E9 Ø3 ØØ E8 9E 48 E8 65 56 B8 3Ø
[ZG5]  1Ø7Ø: 45 5Ø B8 2Ø ØØ 5Ø 58 5B 89 Ø7 E8 D9 5Ø B8 44 45
[AK5]  1Ø71: 5Ø 5B FF 37 B8 2C ØØ 5Ø E8 C9 C8 58 21 CØ 75 Ø3
[6S5]  1Ø72: E9 Ø6 ØØ E8 73 48 E9 B9 FF E9 9D 1F B8 44 45 5Ø
[VL5]  1Ø73: 5B FF 37 B8 BØ ØØ 5Ø E8 E3 C8 58 21 CØ 75 Ø3 E9
[PG5]  1Ø74: Ø3 ØØ E9 Ø4 Ø1 E8 51 48 B8 44 45 5Ø 5B FF 37 B8
[PF5]  1Ø75: 85 ØØ 5Ø E8 C7 C8 58 21 CØ 75 Ø3 E9 25 ØØ B8 ØØ
[AF5]  1Ø76: ØØ 5Ø ØE E8 13 ØØ 12 55 6E 62 65 6B 61 6E 6E 74
[6K5]  1Ø77: 65 72 2Ø 42 65 66 65 68 6C E8 Ø4 C9 E8 59 C9 58
[2L5]  1Ø78: E9 79 46 E8 13 48 B8 44 45 5Ø 5B FF 37 B8 23 ØØ
[KE5]  1Ø79: 5Ø E8 89 C8 58 21 CØ 75 Ø3 E9 12 ØØ B8 Ø8 45 5Ø
[MM5]  1Ø8Ø: B8 ØØ ØØ 5Ø 58 5B 89 Ø7 E8 E3 4F E9 1B ØØ E8 E8
[VK5]  1Ø81: 47 E8 AF 55 8C D8 Ø5 7C 17 5Ø ØE E8 Ø2 ØØ Ø1 2C
[HU5]  1Ø82: E8 BD C8 E8 DB C8 E8 49 55 B8 44 45 5Ø 5B FF 37
[MG5]  1Ø83: B8 22 ØØ 5Ø E8 ØD C8 58 21 CØ 75 Ø3 E9 15 ØØ E8
[EL5]  1Ø84: AØ 1F B8 3Ø 45 5Ø B8 15 ØØ 5Ø 58 5B 89 Ø7 E8 F5
[ØU5]  1Ø85: 4F E8 A5 47 E8 BF 47 B8 4A 45 5Ø 5B FF 37 B8 Ø4
[5F5]  1Ø86: ØØ 5Ø E8 18 C8 58 21 CØ 75 Ø3 E9 2A ØØ B8 ØØ ØØ
[1F5]  1Ø87: 5Ø ØE E8 18 ØØ 17 53 74 72 69 6E 67 76 61 72 69
[TD5]  1Ø88: 61 62 6C 65 2Ø 65 72 77 61 72 74 65 74 E8 5Ø C6
[SK5]  1Ø89: E8 A5 C8 58 E9 C5 45 B8 3Ø 45 5Ø B8 22 ØØ 5Ø 58
[4P5]  1Ø9Ø: 5B 89 Ø7 E8 AØ 4F E9 8Ø 1E B8 44 45 5Ø 5B FF 37
[NG5]  1Ø91: B8 85 ØØ 5Ø E8 C6 C7 58 21 CØ 75 Ø3 E9 Ø3 ØØ E9
[PF5]  1Ø92: Ø7 Ø1 E8 34 47 B8 44 45 5Ø 5B FF 37 B8 22 ØØ 5Ø
[FK5]  1Ø93: E8 AA C7 58 21 CØ 75 Ø3 E9 Ø3 ØØ E9 81 ØØ B8 Ø8
[XL5]  1Ø94: 45 5Ø B8 ØØ ØØ 5Ø 58 5B 89 Ø7 E8 Ø1 4F E8 F2 1E
[TF5]  1Ø95: B8 3Ø 45 5Ø B8 15 ØØ 5Ø 58 5B 89 Ø7 E8 47 4F B8
[6N5]  1Ø96: 44 45 5Ø 5B FF 37 B8 2C ØØ 5Ø E8 37 C7 58 21 CØ
[TK5]  1Ø97: 75 Ø3 E9 Ø6 ØØ E8 E1 46 E9 99 ØØ B8 44 45 5Ø 5B
[PP5]  1Ø98: FF 37 B8 3B ØØ 5Ø E8 1B C7 58 21 CØ 75 Ø3 E9 Ø6
```

Listing 1. »sbasic.exe« ist der Hexdump des Basic-Compilers. Geben Sie dieses Listing nur dann ein, wenn Sie keine frühere Version von SBasic besitzen.

```
[LG5] 1Ø99: ØØ E8 C5 46 E9 6E ØØ B8 ØØ ØØ 5Ø ØE E8 16 ØØ 15
[ZB5] 11ØØ: 27 3B 27 2Ø 6F 64 65 72 2Ø 27 2C 27 2Ø 65 72 77
[9R5] 1101: 61 72 74 65 74 E8 88 C7 E8 DD C7 58 E9 FD 44 B8
[4K5] 1102: 44 45 5Ø 5B FF 37 B8 23 ØØ 5Ø E8 1Ø C7 58 21 CØ
[7K5] 1103: 75 Ø3 E9 Ø3 ØØ E9 1E ØØ E8 7E 46 E8 45 54 8C D8
[GG5] 1104: Ø5 7C 17 5Ø ØE E8 Ø2 ØØ Ø1 2C E8 53 C7 E8 71 C7
[HL5] 1105: E8 DF 53 E9 1E ØØ B8 Ø8 45 5Ø B8 ØØ ØØ 5Ø 58 5B
[KF5] 1106: 89 Ø7 E8 49 4E B8 3Ø 45 5Ø B8 23 ØØ 5Ø 58 5B 89
[BH5] 1107: Ø7 E8 92 4E E8 Ø8 58 B8 3Ø 45 5Ø B8 3Ø ØØ 5Ø 58
[7N5] 1108: 5B 89 Ø7 E8 8Ø 4E E9 6Ø 1D B8 44 45 5Ø 5B FF 37
[PG5] 1109: B8 87 ØØ 5Ø E8 A6 C6 58 21 CØ 75 Ø3 E9 Ø3 ØØ E9
[WF5] 111Ø: 2A ØØ E8 14 46 B8 Ø8 45 5Ø B8 Ø1 ØØ 5Ø E8 EA C6
[UM5] 1111: 58 5B 89 Ø7 E8 F7 4D E8 C5 57 B8 22 45 5Ø B8 58
[EM5] 1112: ØØ 5Ø 58 5B 89 Ø7 E8 F8 4C E9 1D 1D B8 44 45 5Ø
[HK5] 1113: 5B FF 37 B8 84 ØØ 5Ø E8 63 C6 58 21 CØ 75 Ø3 E9
[FE5] 1114: Ø3 ØØ E9 8Ø Ø1 E8 D1 45 B8 22 45 5Ø B8 E9 ØØ 5Ø
[DR5] 1115: 58 5B 89 Ø7 E8 CA 4C B8 Ø8 45 5Ø B8 DE ØF 5Ø B8
[SP5] 1116: Ø2 ØØ 5Ø B8 32 45 5Ø 5B FF 37 E8 6F C6 E8 4Ø C6
[MN5] 1117: 5B FF 37 58 5B 89 Ø7 E8 A4 53 B8 7C 31 5Ø B8 Ø2
[HR5] 1118: ØØ 5Ø B8 3A 45 5Ø 5B FF 37 E8 5Ø C6 E8 21 C6 B8
[FN5] 1119: 14 45 5Ø 5B FF 77 Ø2 FF 37 E8 71 DØ 58 5B 89 Ø7
[TG5] 112Ø: B8 B4 3Ø 5Ø B8 Ø2 ØØ 5Ø B8 3A 45 5Ø 5B FF 37 E8
[WG5] 1121: 2A C6 E8 FB C5 B8 3E ØØ 5Ø B8 Ø2 ØØ 5Ø B8 32 45
[9U5] 1122: 5Ø 5B FF 37 B8 Ø1 ØØ 5Ø E8 FØ C5 E8 ØE C6 E8 DF
[BL5] 1123: C5 5B FF 37 58 5B 89 Ø7 B8 3A 45 5Ø B8 3A 45 5Ø
[ZP5] 1124: 5B FF 37 B8 Ø1 ØØ 5Ø E8 C6 C5 58 5B 89 Ø7 B8 3A
[3S5] 1125: 45 5Ø 5B FF 37 B8 B2 3Ø 5Ø 5B FF 37 E8 3F C5 58
[UC5] 1126: 21 CØ 75 Ø3 E9 27 ØØ B8 ØØ ØØ 5Ø ØE E8 15 ØØ 14
[BD5] 1127: 5A 75 2Ø 76 69 65 6C 65 2Ø 44 41 54 41 2D 5A 65
[CV5] 1128: 69 6C 65 6E E8 D9 C5 E8 2E C6 58 E9 4E 43 B8 44
[ØH5] 1129: 45 5Ø 5B FF 37 B8 ØØ ØØ 5Ø E8 28 C5 58 21 CØ 75
[EE5] 113Ø: Ø3 E9 Ø3 ØØ E9 18 ØØ B8 22 45 5Ø B8 44 45 5Ø 5B
[SW5] 1131: FF 37 58 5B 89 Ø7 E8 C8 4B E8 41 44 E9 CF FF B8
[ZK5] 1132: 22 45 5Ø B8 1A ØØ 5Ø 58 5B 89 Ø7 E8 B3 4B B8 Ø8
[HM5] 1133: 45 5Ø B8 7C 31 5Ø B8 Ø2 ØØ 5Ø B8 3A 45 5Ø 5B FF
[2T5] 1134: 37 E8 58 C5 E8 29 C5 5B FF 37 58 5B 89 Ø7 B8 Ø8
[6G5] 1135: 45 5Ø 5B FF 37 58 21 CØ 75 Ø3 E9 22 ØØ B8 Ø8 45
[4K5] 1136: 5Ø B8 1Ø 45 5Ø 5B FF 77 Ø2 FF 37 B8 Ø8 45 5Ø 5B
[AY5] 1137: FF 37 E8 9A CF E8 16 CA E8 52 CF 58 5B 89 Ø7 E8
[ZN5] 1138: CE 4B E9 84 1B B8 44 45 5Ø 5B FF 37 B8 8C ØØ 5Ø
[9L5] 1139: E8 CA C4 58 21 CØ 75 Ø3 E9 Ø3 ØØ E9 17 Ø1 E8 38
[PA5] 114Ø: 44 B8 Ø8 45 5Ø B8 ØØ ØØ 5Ø 58 5B 89 Ø7 B8 44 45
[VH5] 1141: 5Ø 5B FF 37 B8 ØE ØØ 5Ø E8 69 C4 58 21 CØ 75 Ø3
[FC5] 1142: E9 Ø6 ØØ E8 DØ 43 E8 1Ø 44 B8 3Ø ØØ 5Ø B8 ØØ ØØ
[KF5] 1143: 5Ø 58 5B 89 Ø7 B8 68 45 5Ø B8 Ø8 45 5Ø 5B FF 37
[KP5] 1144: E8 2C CF 58 5A 5B 89 Ø7 89 57 Ø2 B8 7C 31 5Ø B8
[BP5] 1145: Ø2 ØØ 5Ø B8 3Ø ØØ 5Ø 5B FF 37 E8 9F C4 E8 7Ø C4
[HK5] 1146: 5B FF 37 B8 ØØ ØØ 5Ø E8 1A C4 58 21 CØ 75 Ø3 E9
[LA5] 1147: Ø3 ØØ E9 49 ØØ B8 B4 3Ø 5Ø B8 Ø2 ØØ 5Ø B8 3Ø ØØ
[GY5] 1148: 5Ø 5B FF 37 E8 75 C4 E8 46 C4 5B FF 37 E8 DF CE
[FN5] 1149: B8 68 45 5Ø 5B FF 77 Ø2 FF 37 E8 36 CA 58 21 CØ
[LD5] 115Ø: 75 Ø3 E9 19 ØØ B8 3Ø ØØ 5Ø B8 3Ø ØØ 5Ø 5B FF 37
[3P5] 1151: B8 Ø1 ØØ 5Ø E8 19 C4 58 5B 89 Ø7 E9 8D FF B8 Ø8
[GM5] 1152: 45 5Ø B8 C7 Ø6 5Ø 58 5B 89 Ø7 E8 E3 4A B8 Ø8 45
[6F5] 1153: 5Ø B8 17 ØØ 5Ø 58 5B 89 Ø7 E8 D4 4A B8 Ø8 45 5Ø
[4M5] 1154: B8 1Ø 45 5Ø 5B FF 77 Ø2 FF 37 B8 7C 31 5Ø B8 Ø2
[AM5] 1155: ØØ 5Ø B8 3Ø ØØ 5Ø 5B FF 37 E8 ØØ C4 E8 D1 C3 5B
[2X5] 1156: FF 37 E8 27 C7 E8 E6 C8 E8 8B C9 58 5B 89 Ø7 E8
[6M5] 1157: 9E 4A E9 54 1A B8 44 45 5Ø 5B FF 37 B8 97 ØØ 5Ø
[1M5] 1158: E8 9A C3 58 21 CØ 75 Ø3 E9 Ø3 ØØ E9 2A Ø7 E8 Ø8
[ZK5] 1159: 43 B8 44 45 5Ø 5B FF 37 B8 D1 ØØ 5Ø E8 7E C3 58
[ØL5] 116Ø: 21 CØ 75 Ø3 E9 Ø3 ØØ E9 1Ø Ø6 E8 EC 42 E8 Ø3 4E
[RA5] 1161: B8 3Ø ØØ 5Ø B8 ØØ ØØ 5Ø 58 5B 89 Ø7 B8 3Ø ØØ 5Ø
[9S5] 1162: 5B FF 37 B8 6C 45 5Ø 5B FF 37 E8 17 C3 58 21 CØ
[YF5] 1163: 75 Ø3 E9 Ø3 ØØ E9 95 ØØ 8C D8 Ø5 8C 17 5Ø B8 ØØ
[9N5] 1164: ØØ 5Ø E8 9B C3 8C D8 Ø5 FC 1Ø 5Ø B8 1Ø ØØ 5Ø B8
[YH5] 1165: 3Ø ØØ 5Ø 5B FF 37 E8 63 C3 E8 34 C3 B8 ØØ ØØ 5Ø
[GN5] 1166: E8 7D C3 E8 4D C5 58 21 CØ 75 Ø3 E9 Ø3 ØØ E9 19
[TA5] 1167: ØØ B8 3Ø ØØ 5Ø B8 3Ø ØØ 5Ø 5B FF 37 B8 Ø1 ØØ 5Ø
[ST5] 1168: E8 ØD C3 58 5B 89 Ø7 E9 92 FF B8 28 45 5Ø 5B FF
[BE5] 1169: 37 B8 ØØ ØØ 5Ø E8 AC C2 58 21 CØ 75 Ø3 E9 2D ØØ
[DK5] 117Ø: B8 ØØ ØØ 5Ø ØE E8 1B ØØ 1A 46 75 6E 6B 74 69 6F
[BE5] 1171: 6E 2Ø 62 65 72 65 69 74 73 2Ø 64 65 66 69 6E 69
[1T5] 1172: 65 72 74 E8 1A C3 E8 6F C3 58 E9 8F 4Ø B8 6E 45
[EL5] 1173: 5Ø B8 3Ø ØØ 5Ø 5B FF 37 58 5B 89 Ø7 8C D8 Ø5 FC
[XH5] 1174: 1Ø 5Ø B8 1Ø ØØ 5Ø B8 3Ø ØØ 5Ø 5B FF 37 E8 CC C2
[HM5] 1175: E8 9D C2 8C D8 Ø5 8C 17 5Ø B8 ØØ ØØ 5Ø E8 EØ C2
[3R5] 1176: E8 FE C2 B8 EC 3F 5Ø B8 Ø2 ØØ 5Ø B8 3Ø ØØ 5Ø 5B
[AS5] 1177: FF 37 E8 A7 C2 E8 78 C2 B8 4A 45 5Ø 5B FF 37 58
[KP5] 1178: 5B 89 Ø7 B8 44 45 5Ø 5B FF 37 B8 28 ØØ 5Ø E8 4C
[TL5] 1179: C2 58 21 CØ 75 Ø3 E9 Ø3 ØØ E9 1E 4F E8 BA 41 E8
[4E5] 118Ø: D1 4C 8C D8 Ø5 1C 14 5Ø B8 1Ø ØØ 5Ø B8 7Ø 45 5Ø
[KS5] 1181: 5B FF 37 E8 66 C2 E8 37 C2 8C D8 Ø5 8C 17 5Ø B8
[YE5] 1182: ØØ ØØ 5Ø E8 7A C2 E8 98 C2 B8 5Ø 4Ø 5Ø B8 Ø2 ØØ
[2P5] 1183: 5Ø B8 7Ø 45 5Ø 5B FF 37 E8 41 C2 E8 12 C2 B8 4A
[VL5] 1184: 45 5Ø 5B FF 37 58 5B 89 Ø7 B8 28 45 5Ø 5B FF 37
[SG5] 1185: 58 21 CØ 75 Ø3 E9 Ø3 ØØ E9 C4 ØØ B8 4A 45 5Ø 5B
[LP5] 1186: FF 37 B8 Ø4 ØØ 5Ø E8 9B C1 58 21 CØ 75 Ø3 E9 4A
[ØD5] 1187: ØØ B8 B4 4Ø 5Ø B8 64 ØØ 5Ø B8 3Ø ØØ 5Ø 5B FF 37
[2S5] 1188: E8 F9 C1 E8 CA C1 B8 Ø2 ØØ 5Ø B8 7Ø 45 5Ø 5B FF
[7U5] 1189: 37 E8 E8 C1 E8 B9 C1 B8 18 45 5Ø 5B FF 37 58 5B
[SH5] 119Ø: 89 Ø7 B8 18 45 5Ø B8 18 45 5Ø 5B FF 37 B8 1Ø ØØ
[UK5] 1191: 5Ø E8 9C C1 58 5B 89 Ø7 E9 64 ØØ B8 B4 4Ø 5Ø B8
[4R5] 1192: 64 ØØ 5Ø B8 3Ø ØØ 5Ø 5B FF 37 E8 AF C1 E8 8Ø C1
[PN5] 1193: B8 Ø2 ØØ 5Ø B8 7Ø 45 5Ø 5B FF 37 E8 9E C1 E8 6F
[UR5] 1194: C1 B8 ØC 45 5Ø 5B FF 77 Ø2 FF 37 E8 BF CB 58 5B
[3S5] 1195: 89 Ø7 B8 ØC 45 5Ø B8 ØC 45 5Ø 5B FF 77 Ø2 FF 37
[BM5] 1196: B8 Ø2 ØØ 5Ø B8 4A 45 5Ø 5B FF 37 E8 9E C4 E8 62
[CR5] 1197: C6 E8 A3 C6 E8 57 C6 58 5A 5B 89 Ø7 89 57 Ø2 B8
[8M5] 1198: 44 42 5Ø B8 64 ØØ 5Ø B8 3Ø ØØ 5Ø 5B FF 37 E8 4B
[CH5] 1199: C1 E8 1C C1 B8 Ø2 ØØ 5Ø B8 7Ø 45 5Ø 5B FF 37 E8
[MN5] 12ØØ: 3A C1 E8 ØB C1 B8 4A 45 5Ø 5B FF 37 58 5B 89 Ø7
[SG5] 12Ø1: B8 9C 44 5Ø B8 Ø2 ØØ 5Ø B8 7Ø 45 5Ø 5B FF 37 E8
[DH5] 12Ø2: 1A C1 E8 EB CØ B8 B4 4Ø 5Ø B8 64 ØØ 5Ø B8 3Ø ØØ
[DN5] 12Ø3: 5Ø 5B FF 37 E8 Ø5 C1 E8 D6 CØ B8 Ø2 ØØ 5Ø B8 7Ø
[5W5] 12Ø4: 45 5Ø 5B FF 37 E8 F4 CØ E8 C5 CØ 5B FF 37 58 5B
[TG5] 12Ø5: 89 Ø7 B8 7Ø 45 5Ø B8 7Ø 45 5Ø 5B FF 37 B8 Ø1 ØØ
[1M5] 12Ø6: 5Ø E8 AC CØ 58 5B 89 Ø7 B8 7Ø 45 5Ø 5B FF 37 B8
[MD5] 12Ø7: 31 ØØ 5Ø E8 28 CØ 58 21 CØ 75 Ø3 E9 26 ØØ B8 ØØ
[2D5] 12Ø8: ØØ 5Ø ØE E8 14 ØØ 13 5A 75 2Ø 76 69 65 6C 65 2Ø
[SP5] 12Ø9: 46 75 6E 6B 74 69 6F 6E 65 6E E8 C3 CØ E8 18 C1
```

```
[3M5] 121Ø: 58 E9 38 3E B8 44 45 5Ø 5B FF 37 B8 29 ØØ 5Ø E8
[KC5] 1211: 12 CØ 58 21 CØ 75 Ø3 E9 Ø3 ØØ E9 8Ø ØØ B8 44 45
[4H5] 1212: 5Ø 5B FF 37 B8 2C ØØ 5Ø E8 32 CØ 58 21 CØ 75 Ø3
[HD5] 1213: E9 28 ØØ B8 ØØ ØØ 5Ø ØE E8 16 ØØ 15 27 2C 27 2Ø
[ØC5] 1214: 6F 64 65 72 2Ø 27 29 27 2Ø 65 72 77 61 72 74 65
[EN5] 1215: 74 E8 6C CØ E8 C1 CØ 58 E9 E1 3D B8 7Ø 45 5Ø 5B
[TR5] 1216: FF 37 B8 Ø4 ØØ 5Ø E8 BB BF 58 21 CØ 75 Ø3 E9 26
[6B5] 1217: ØØ B8 ØØ ØØ 5Ø ØE E8 14 ØØ 13 4D 61 78 69 6D 61
[ZC5] 1218: 6C 2Ø 34 2Ø 5Ø 61 72 61 6D 65 74 65 72 E8 3Ø CØ
[5W5] 1219: E8 85 CØ 58 E9 A5 3D E8 3F 3F E9 82 FD E8 39 3F
[HN5] 122Ø: B8 44 45 5Ø 5B FF 37 B8 E7 ØØ 5Ø E8 AF BF 58 21
[VF5] 1221: CØ 75 Ø3 E9 1F ØØ B8 ØØ ØØ 5Ø ØE E8 ØD ØØ ØC 27
[WK5] 1222: 3D 27 2Ø 65 72 77 61 72 74 65 74 E8 F2 BF E8 47
[7K5] 1223: CØ 58 E9 67 3D E8 Ø1 3F B8 22 45 5Ø B8 E9 ØØ 5Ø
[TL5] 1224: 58 5B 89 Ø7 E8 FA 45 B8 Ø8 45 5Ø B8 38 44 5Ø B8
[4U5] 1225: Ø2 ØØ 5Ø B8 3Ø ØØ 5Ø 5B FF 37 E8 9F BF E8 7Ø BF
[CP5] 1226: 5B FF 37 58 5B 89 Ø7 E8 D4 4C B8 D4 43 5Ø B8 Ø2
[ER5] 1227: ØØ 5Ø B8 3Ø ØØ 5Ø 5B FF 37 E8 8Ø BF E8 51 BF B8
[SR5] 1228: 14 45 5Ø 5B FF 77 Ø2 FF 37 E8 A1 C9 58 5B 89 Ø7
[CG5] 1229: B8 22 45 5Ø B8 5B ØØ 5Ø 58 5B 89 Ø7 E8 A2 45 B8
[ZP5] 123Ø: 7Ø 45 5Ø 5B FF 37 B8 Ø1 ØØ 5Ø E8 2E BF 8F Ø6 2A
[PC5] 1231: 45 B8 ØØ ØØ 5Ø 8F Ø6 72 45 B8 Ø1 ØØ 5Ø E8 5A BF
[VM5] 1232: 8F Ø6 74 45 FF 36 74 45 E8 59 CØ 58 8Ø E4 Ø5 2E
[DV5] 1233: 88 26 FF 4C E9 ØF ØØ FF 36 2A 45 FF 36 74 45 E8
[3S5] 1234: EE BE 8F Ø6 2A 45 FF 36 2A 45 FF 36 72 45 EB ØØ
[ØS5] 1235: E8 B7 BE EB Ø3 E8 9F BE 58 21 CØ 75 Ø3 E9 71 ØØ
[TC5] 1236: B8 4A 45 5Ø B8 5Ø 4Ø 5Ø B8 Ø2 ØØ 5Ø B8 2A 45 5Ø
[9X5] 1237: 5B FF 37 E8 E6 BE E8 B7 BE 5B FF 37 58 5B 89 Ø7
[GE5] 1238: B8 5E 45 5Ø B8 9C 44 5Ø B8 Ø2 ØØ 5Ø B8 2A 45 5Ø
[AW5] 1239: 5B FF 37 E8 C6 BE E8 97 BE 5B FF 37 58 5B 89 Ø7
[ZH5] 124Ø: B8 4A 45 5Ø 5B FF 37 B8 Ø4 ØØ 5Ø E8 23 BE 58 21
[HL5] 1241: CØ 75 Ø3 E9 Ø6 ØØ E8 1E 4E E9 12 ØØ E8 17 5Ø B8
[YG5] 1242: 3Ø 45 5Ø B8 32 ØØ 5Ø 58 5B 89 Ø7 E8 18 46 E9 66
[1G5] 1243: FF B8 22 45 5Ø B8 53 ØØ 5Ø 58 5B 89 Ø7 E8 C1 44
[VP5] 1244: E8 9F 15 B8 4C 45 5Ø B8 EC 3F 5Ø B8 Ø2 ØØ 5Ø B8
[3W5] 1245: 6E 45 5Ø 5B FF 37 E8 63 BE E8 34 BE 5B FF 37 58
[BH5] 1246: 5B 89 Ø7 E8 56 4A B8 7Ø 45 5Ø B8 ØØ ØØ 5Ø 58 5B
[GT5] 1247: 89 Ø7 B8 EC 3F 5Ø B8 Ø2 ØØ 5Ø B8 6E 45 5Ø 5B FF
[8V5] 1248: 37 E8 38 BE E8 Ø9 BE 5B FF 37 E8 Ø8 ØØ E8 4D F7
[DH5] 1249: 4D 7F 89 Ø6 4E E8 99 C3 B8 3Ø 45 5Ø B8 33 ØØ 5Ø
[LG5] 125Ø: 58 5B 89 Ø7 E9 1B ØØ B8 3Ø 45 5Ø B8 58 ØØ 5Ø 58
[LH5] 1251: 5B 89 Ø7 E9 ØC ØØ B8 3Ø 45 5Ø B8 34 ØØ 5Ø 58 5B
[1D5] 1252: 89 Ø7 E8 81 45 B8 38 44 5Ø B8 Ø2 ØØ 5Ø B8 6E 45
[HV5] 1253: 5Ø 5B FF 37 E8 E5 BD E8 B6 BD B8 14 45 5Ø 5B FF
[SP5] 1254: 77 Ø2 FF 37 E8 6F C3 58 5B 89 Ø7 B8 28 45 5Ø 5B
[PN5] 1255: FF 37 B8 ØØ ØØ 5Ø E8 4B BD 58 21 CØ 75 Ø3 E9 16
[RE5] 1256: ØØ B8 6C 45 5Ø B8 6C 45 5Ø 5B FF 37 B8 Ø1 ØØ 5Ø
[GP5] 1257: E8 7D BD 58 5B 89 Ø7 E9 ØF 14 8C D8 Ø5 7C 17 5Ø
[TU5] 1258: B8 44 45 5Ø 5B FF 37 E8 A7 BE E8 D4 BD E8 4D 3C
[ZF5] 1259: 8C D8 Ø5 7C 17 5Ø 8C D8 Ø5 7C 17 5Ø B8 ØØ ØØ 5Ø
[XX5] 126Ø: E8 9D BD B8 44 45 5Ø 5B FF 37 E8 84 BE E8 52 BE
[PN5] 1261: E8 AE BD 8C D8 Ø5 7C 17 5Ø B8 ØØ ØØ 5Ø E8 8Ø BD
[GM5] 1262: E8 42 BF B8 Ø3 ØØ 5Ø E8 C7 BC 58 21 CØ 75 Ø3 E9
[RP5] 1263: Ø3 ØØ E9 B8 FF 8C D8 Ø5 7C 17 5Ø B8 ØØ ØØ 5Ø E8
[6P5] 1264: 5E BD ØE E8 Ø4 ØØ Ø3 53 45 47 E8 53 BD E8 36 BF
[1C5] 1265: 58 21 CØ 75 Ø3 E9 2B ØØ B8 ØØ ØØ 5Ø ØE E8 19 ØØ
[NC5] 1266: 16 27 46 4E 27 2Ø 6F 64 65 72 2Ø 27 53 45 47 27
[UK5] 1267: 2Ø 65 72 77 61 72 74 65 74 E8 24 BD E8 79 BD 58
[MP5] 1268: E9 99 3A E8 33 3C B8 44 45 5Ø 5B FF 37 B8 E7 ØØ
[ØL5] 1269: 5Ø E8 A9 BC 58 21 CØ 75 Ø3 E9 Ø3 ØØ E9 27 ØØ E8
[ML5] 127Ø: 17 3C E8 DE 49 B8 Ø8 45 5Ø B8 8F Ø6 5Ø 58 5B 89
[ZD5] 1271: Ø7 E8 7C 43 B8 Ø8 45 5Ø B8 19 ØØ 5Ø 58 5B 89 Ø7
[LP5] 1272: E8 6D 43 E9 23 13 B8 Ø8 45 5Ø B8 8C 1E 5Ø 58 5B
[SU5] 1273: 89 Ø7 E8 5B 43 E9 DC FF B8 44 45 5Ø 5B FF 37 B8
[6G5] 1274: 98 ØØ 5Ø E8 57 BC 58 21 CØ 75 Ø3 E9 Ø3 ØØ E9 3Ø
[4P5] 1275: ØØ E8 C5 3B E8 8C 49 8C D8 Ø5 7C 17 5Ø ØE E8 Ø2
[BS5] 1276: ØØ Ø1 2C E8 9A BC E8 B8 BC E8 26 49 E8 74 49 B8
[SL5] 1277: 3Ø 45 5Ø B8 35 ØØ 5Ø 58 5B 89 Ø7 E8 E8 43 E9 C8
[XK5] 1278: 12 B8 44 45 5Ø 5B FF 37 B8 B3 ØØ 5Ø E8 ØE BC 58
[XL5] 1279: 21 CØ 75 Ø3 E9 Ø3 ØØ E9 B9 ØØ E8 7C 3B E8 93 46
[HR5] 128Ø: E8 89 4A B8 76 45 5Ø B8 4A 45 5Ø 5B FF 37 58 5B
[VP5] 1281: 89 Ø7 B8 78 45 5Ø B8 5E 45 5Ø 5B FF 37 58 5B 89
[PL5] 1282: Ø7 B8 44 45 5Ø 5B FF 37 B8 28 ØØ 5Ø E8 CE BB 58
[4P5] 1283: 21 CØ 75 Ø3 E9 Ø3 ØØ E9 37 ØØ E8 3C 3B E8 DF 4D
[3L5] 1284: B8 44 45 5Ø 5B FF 37 B8 2C ØØ 5Ø E8 76 BB 58 21
[PS5] 1285: CØ 75 Ø3 E9 Ø6 ØØ E8 2Ø 3B E9 E1 FF 8C D8 Ø5 7C
[RL5] 1286: 17 5Ø ØE E8 Ø2 ØØ Ø1 29 E8 F5 BB E8 13 BC E8 81
[7H5] 1287: 48 B8 4A 45 5Ø B8 76 45 5Ø 5B FF 37 58 5B 89 Ø7
[AL5] 1288: B8 5E 45 5Ø B8 78 45 5Ø 5B FF 37 58 5B 89 Ø7 E8
[EP5] 1289: C8 4B B8 4C 45 5Ø B8 Ø1 ØØ 5Ø 58 5B 89 Ø7 E8 9B
[MB5] 129Ø: 47 B8 3Ø 45 5Ø B8 3D ØØ 5Ø 58 5B 89 Ø7 E8 16 43
[EU5] 1291: E9 F6 11 B8 44 45 5Ø 5B FF 37 B8 FF ØØ 5Ø E8 3C
[5N5] 1292: BB 58 21 CØ 75 Ø3 E9 Ø3 ØØ E9 F8 Ø8 E8 2E 3A B8
[SL5] 1293: 44 45 5Ø 5B FF 37 B8 83 ØØ 5Ø E8 2Ø BB 58 21 CØ
[XL5] 1294: 75 Ø3 E9 Ø3 ØØ E9 F5 ØØ E8 8E 3A 8C D8 Ø5 7C 17
[2P5] 1295: 5Ø ØE E8 Ø2 ØØ Ø1 28 E8 66 BB E8 84 BB E8 F2 47
[3T5] 1296: E8 93 3A B8 4A 45 5Ø 5B FF 37 B8 Ø4 ØØ 5Ø E8 EC
[4L5] 1297: BA 58 21 CØ 75 Ø3 E9 Ø3 ØØ E9 93 38 8C D8 Ø5 7C
[LN5] 1298: 17 5Ø ØE E8 Ø2 ØØ Ø1 2C E8 35 BB E8 53 BB E8 C1
[BL5] 1299: 47 E8 ØF 48 B8 44 45 5Ø 5B FF 37 B8 2C ØØ 5Ø E8
[FN5] 13ØØ: BB BA 58 21 CØ 75 Ø3 E9 12 ØØ B8 Ø8 45 5Ø B8 FF
[ØH5] 13Ø1: ØØ 5Ø 58 5B 89 Ø7 E8 15 42 E9 Ø6 ØØ E8 1A 3A E8
[ML5] 13Ø2: E1 47 8C D8 Ø5 7C 17 5Ø ØE E8 Ø2 ØØ Ø1 29 E8 EF
[ZV5] 13Ø3: BA E8 ØD BB E8 7B 47 B8 44 45 5Ø 5B FF 37 B8 E7
[1H5] 13Ø4: ØØ 5Ø E8 78 BA 58 21 CØ 75 Ø3 E9 1F ØØ B8 ØØ ØØ
[HG5] 13Ø5: 5Ø ØE E8 ØD ØØ ØC 27 3D 27 2Ø 65 72 77 61 72 74
[BU5] 13Ø6: 65 74 E8 BB BA E8 1Ø BB 58 E9 3Ø 38 E8 CA 39 E8
[DM5] 13Ø7: BØ 11 B8 4A 45 5Ø 5B FF 37 B8 Ø4 ØØ 5Ø E8 3D BA
[RF5] 13Ø8: 58 21 CØ 75 Ø3 E9 Ø3 ØØ E9 E4 37 B8 3Ø 45 5Ø B8
[BP5] 13Ø9: 3E ØØ 5Ø 58 5B 89 Ø7 E8 EC 41 E9 CC 1Ø B8 44 45
[LG5] 131Ø: 5Ø 5B FF 37 B8 82 ØØ 5Ø E8 12 BA 58 21 CØ 75 Ø3
[KK5] 1311: E9 Ø3 ØØ E9 61 Ø2 E8 8Ø 39 B8 44 45 5Ø 5B FF 37
[ØH5] 1312: B8 23 ØØ 5Ø E8 BD B9 58 21 CØ 75 Ø3 E9 Ø3 ØØ E8
[CK5] 1313: 67 39 E8 2E 47 B8 28 45 5Ø 5B FF 37 B8 ØØ ØØ 5Ø
[PL5] 1314: E8 A1 B9 58 21 CØ 75 Ø3 E9 46 ØØ B8 D8 3A 5Ø B8
[HW5] 1315: Ø2 ØØ 5Ø B8 3E 45 5Ø 5B FF 37 E8 FF B9 E8 DØ B9
[XM5] 1316: B8 ØC 45 5Ø 5B FF 77 Ø2 FF 37 E8 2Ø C4 58 5B 89
[CH5] 1317: Ø7 B8 ØC 45 5Ø B8 ØC 45 5Ø 5B FF 77 Ø2 FF 37 B8
[CP5] 1318: Ø1 ØØ 5Ø E8 Ø6 BD E8 C5 BE 58 5A 5B 89 Ø7 89 57
[WC5] 1319: Ø2 B8 Ø8 45 5Ø B8 D8 3A 5Ø B8 Ø2 ØØ 5Ø B8 3E 45
[CU5] 132Ø: 5Ø 5B FF 37 E8 B5 B9 E8 86 B9 5B FF 37 58 5B 89
```

```
[BF5] 1321: Ø7 E8 DA 4Ø B8 7A 45 5Ø B8 ØØ ØØ 5Ø 58 5B 89 Ø7
[2N5] 1322: 8C D8 Ø5 7C 17 5Ø ØE E8 Ø2 ØØ Ø1 2C E8 B1 B9 E8
[HP5] 1323: CF B9 E8 3D 46 E8 8B 46 B8 5E 45 5Ø B8 D8 3A 5Ø
[WP5] 1324: B8 Ø2 ØØ 5Ø B8 3E 45 5Ø 5B FF 37 E8 6E B9 E8 3F
[VW5] 1325: B9 5B FF 37 E8 D8 C3 B8 Ø1 ØØ 5Ø E8 8E BC E8 4D
[9R5] 1326: BE B8 7A 45 5Ø 5B FF 37 B8 Ø4 ØØ 5Ø E8 4D B9 E8
[VT5] 1327: 7A BC E8 39 BE E8 75 C3 58 5B 89 Ø7 E8 B8 48 E8
[6K5] 1328: 97 4C E8 91 38 B8 4A 45 5Ø 5B FF 37 B8 Ø4 ØØ 5Ø
[HM5] 1329: E8 EA B8 58 21 CØ 75 Ø3 E9 Ø3 ØØ E9 91 36 B8 4A
[WL5] 133Ø: 45 5Ø B8 Ø1 ØØ 5Ø 58 5B 89 Ø7 B8 5E 45 5Ø B8 D8
[5S5] 1331: 3A 5Ø B8 Ø2 ØØ 5Ø B8 3E 45 5Ø 5B FF 37 E8 FC B8
[5T5] 1332: E8 CD B8 5B FF 37 E8 66 C3 B8 Ø3 ØØ 5Ø E8 1C BC
[AW5] 1333: E8 DB BD B8 7A 45 5Ø 5B FF 37 B8 Ø4 ØØ 5Ø E8 DB
[RT5] 1334: B8 E8 Ø8 BC E8 C7 BD E8 Ø3 C3 58 5B 89 Ø7 E8 46
[VF5] 1335: 48 B8 7A 45 5Ø B8 7A 45 5Ø 5B FF 37 B8 Ø1 ØØ 5Ø
[CR5] 1336: E8 8D B8 58 5B 89 Ø7 B8 28 45 5Ø 5B FF 37 B8 ØØ
[DG5] 1337: ØØ 5Ø E8 2F B8 58 21 CØ 75 Ø3 E9 2Ø ØØ B8 ØC 45
[1K5] 1338: 5Ø B8 ØC 45 5Ø 5B FF 77 Ø2 FF 37 B8 Ø4 ØØ 5Ø E8
[WP5] 1339: BA BB E8 79 BD 58 5A 5B 89 Ø7 89 57 Ø2 B8 44 45
[8K5] 134Ø: 5Ø 5B FF 37 B8 2C ØØ 5Ø E8 F9 B7 58 21 CØ 75 Ø3
[ST5] 1341: E9 Ø3 ØØ E9 CA FE B8 Ø8 45 5Ø B8 7A 45 5Ø 5B FF
[5F5] 1342: 37 58 5B 89 Ø7 E8 86 3F B8 3Ø 45 5Ø B8 4Ø ØØ 5Ø
[4R5] 1343: 58 5B 89 Ø7 E8 CF 3F B8 3E 45 5Ø B8 3E 45 5Ø 5B
[7L5] 1344: FF 37 B8 Ø1 ØØ 5Ø E8 Ø7 B8 58 5B 89 Ø7 B8 7A 45
[MG5] 1345: 5Ø 5B FF 37 B8 32 ØØ 5Ø E8 96 B7 58 21 CØ 75 Ø3
[FM5] 1346: E9 Ø6 ØØ E9 83 ØE E9 2E ØØ B8 ØØ ØØ 5Ø ØE E8 1C
[2D5] 1347: ØØ 1B 5A 75 2Ø 76 69 65 6C 65 2Ø 46 49 45 4C 44
[6M5] 1348: 2D 44 65 66 69 6E 69 74 69 6F 6E 65 6E E8 1Ø B8
[7S5] 1349: E8 65 B8 58 E9 85 35 B8 44 45 5Ø 5B FF 37 B8 89
[SE5] 135Ø: ØØ 5Ø E8 98 B7 58 21 CØ 75 Ø3 E9 Ø3 ØØ E9 49 ØØ
[YL5] 1351: E8 Ø6 37 B8 44 45 5Ø 5B FF 37 B8 23 ØØ 5Ø E8 43
[YL5] 1352: B7 58 21 CØ 75 Ø3 E9 Ø3 ØØ E8 ED 36 E8 B4 44 8C
[LL5] 1353: D8 Ø5 7C 17 5Ø ØE E8 Ø2 ØØ Ø1 2C E8 C2 B7 E8 EØ
[MG5] 1354: B7 E8 4E 44 E8 C5 44 B8 3Ø 45 5Ø B8 41 ØØ 5Ø 58
[8P5] 1355: 5B 89 Ø7 E8 1Ø 3F E9 FØ ØD B8 44 45 5Ø 5B FF 37
[1G5] 1356: B8 85 ØØ 5Ø E8 36 B7 58 21 CØ 75 Ø3 E9 Ø3 ØØ E9
[RD5] 1357: 15 ØØ E8 17 4B B8 3Ø 45 5Ø B8 42 ØØ 5Ø 58 5B 89
[ZS5] 1358: Ø7 E8 E2 3E E9 C2 ØD B8 44 45 5Ø 5B FF 37 B8 86
[DD5] 1359: ØØ 5Ø E8 Ø8 B7 58 21 CØ 75 Ø3 E9 Ø3 ØØ E9 15 ØØ
[KK5] 136Ø: E8 E9 4A B8 3Ø 45 5Ø B8 43 ØØ 5Ø 58 5B 89 Ø7 E8
[ØM5] 1361: B4 3E E9 94 ØD B8 44 45 5Ø 5B FF 37 B8 88 ØØ 5Ø
[NL5] 1362: E8 DA B6 58 21 CØ 75 Ø3 E9 Ø3 ØØ E9 49 ØØ E8 48
[SH5] 1363: 36 B8 44 45 5Ø 5B FF 37 B8 23 ØØ 5Ø E8 85 B6 58
[7M5] 1364: 21 CØ 75 Ø3 E9 Ø3 ØØ E8 2F 36 E8 F6 43 8C D8 Ø5
[PM5] 1365: 7C 17 5Ø ØE E8 Ø2 ØØ Ø1 2C E8 Ø4 B7 E8 22 B7 E8
[RC5] 1366: 9Ø 43 E8 Ø7 44 B8 3Ø 45 5Ø B8 44 ØØ 5Ø 58 5B 89
[BS5] 1367: Ø7 E8 52 3E E9 32 ØD B8 44 45 5Ø 5B FF 37 B8 8E
[RE5] 1368: ØØ 5Ø E8 78 B6 58 21 CØ 75 Ø3 E9 Ø3 ØØ E9 69 ØØ
[XS5] 1369: E8 E6 35 B8 44 45 5Ø 5B FF 37 B8 E7 ØØ 5Ø E8 5C
[AK5] 137Ø: B6 58 21 CØ 75 Ø3 E9 1F ØØ B8 ØØ ØØ 5Ø ØE E8 ØD
[AH5] 1371: ØØ ØC 27 3D 27 2Ø 65 72 77 61 72 74 65 74 E8 9F
[2T5] 1372: B6 E8 F4 B6 58 E9 14 34 E8 AE 35 E8 94 ØD B8 4A
[YG5] 1373: 45 5Ø 5B FF 37 B8 Ø4 ØØ 5Ø E8 21 B6 58 21 CØ 75
[1E5] 1374: Ø3 E9 Ø3 ØØ E9 C8 33 B8 3Ø 45 5Ø B8 76 ØØ 5Ø 58
[VP5] 1375: 5B 89 Ø7 E8 DØ 3D E9 BØ ØC B8 44 45 5Ø 5B FF 37
[BH5] 1376: B8 9A ØØ 5Ø E8 F6 B5 58 21 CØ 75 Ø3 E9 Ø3 ØØ E9
[ZM5] 1377: 59 ØØ E8 64 35 E8 4A ØD B8 4A 45 5Ø 5B FF 37 B8
[4F5] 1378: ØØ ØØ 5Ø E8 9E B5 58 21 CØ 75 Ø3 E9 12 ØØ B8 3Ø
[8L5] 1379: 45 5Ø B8 7B ØØ 5Ø 58 5B 89 Ø7 E8 89 3D E9 19 ØØ
[PL5] 138Ø: B8 4A 45 5Ø 5B FF 37 B8 Ø4 ØØ 5Ø E8 AF B5 58 21
[2E5] 1381: CØ 75 Ø3 E9 Ø3 ØØ E9 56 33 B8 3Ø 45 5Ø B8 7C ØØ
[5M5] 1382: 5Ø 58 5B 89 Ø7 E8 5E 3D E9 3E ØC B8 44 45 5Ø 5B
[9M5] 1383: FF 37 B8 98 ØØ 5Ø E8 84 B5 58 21 CØ 75 Ø3 E9 Ø3
[1P5] 1384: ØØ E9 31 ØØ E8 F2 34 E8 D8 ØC B8 4A 45 5Ø 5B FF
[5C5] 1385: 37 B8 Ø4 ØØ 5Ø E8 65 B5 58 21 CØ 75 Ø3 E9 Ø3 ØØ
[1M5] 1386: E9 ØC 33 B8 3Ø 45 5Ø B8 7F ØØ 5Ø 58 5B 89 Ø7 E8
[ØM5] 1387: 14 3D E9 F4 ØB B8 44 45 5Ø 5B FF 37 B8 99 ØØ 5Ø
[GL5] 1388: E8 3A B5 58 21 CØ 75 Ø3 E9 Ø3 ØØ E9 31 ØØ E8 A8
[8M5] 1389: 34 E8 8E ØC B8 4A 45 5Ø 5B FF 37 B8 Ø4 ØØ 5Ø E8
[SF5] 139Ø: 1B B5 58 21 CØ 75 Ø3 E9 Ø3 ØØ E9 C2 32 B8 3Ø 45
[TL5] 1391: 5Ø B8 8Ø ØØ 5Ø 58 5B 89 Ø7 E8 CA 3C E9 AA ØB B8
[XM5] 1392: 44 45 5Ø 5B FF 37 B8 97 ØØ 5Ø E8 FØ B4 58 21 CØ
[UK5] 1393: 75 Ø3 E9 Ø3 ØØ E9 31 ØØ E8 5E 34 E8 44 ØC B8 4A
[YK5] 1394: 45 5Ø 5B FF 37 B8 Ø4 ØØ 5Ø E8 D1 B4 58 21 CØ 75
[7D5] 1395: Ø3 E9 Ø3 ØØ E9 78 32 B8 3Ø 45 5Ø B8 81 ØØ 5Ø 58
[UM5] 1396: 5B 89 Ø7 E8 8Ø 3C E9 6Ø ØB B8 44 45 5Ø 5B FF 37
[4H5] 1397: B8 8A ØØ 5Ø E8 A6 B4 58 21 CØ 75 Ø3 E9 Ø3 ØØ E9
[VB5] 1398: 15 ØØ E8 14 34 B8 3Ø 45 5Ø B8 26 ØØ 5Ø 58 5B 89
[WN5] 1399: Ø7 E8 52 3C E9 32 ØB B8 44 45 5Ø 5B FF 37 B8 93
[XD5] 14ØØ: ØØ 5Ø E8 78 B4 58 21 CØ 75 Ø3 E9 Ø3 ØØ E9 31 ØØ
[HR5] 14Ø1: E8 E6 33 E8 CC ØB B8 4A 45 5Ø 5B FF 37 B8 Ø4 ØØ
[9F5] 14Ø2: 5Ø E8 59 B4 58 21 CØ 75 Ø3 E9 Ø3 ØØ E9 ØØ 32 B8
[9M5] 14Ø3: 3Ø 45 5Ø B8 89 ØØ 5Ø 58 5B 89 Ø7 E8 Ø8 3C E9 E8
[TL5] 14Ø4: ØA B8 44 45 5Ø 5B FF 37 B8 87 ØØ 5Ø E8 2E B4 58
[ZH5] 14Ø5: 21 CØ 75 Ø3 E9 Ø3 ØØ E9 31 ØØ E8 9C 33 E8 82 ØB
[KK5] 14Ø6: B8 4A 45 5Ø 5B FF 37 B8 Ø4 ØØ 5Ø E8 ØF B4 58 21
[7F5] 14Ø7: CØ 75 Ø3 E9 Ø3 ØØ E9 B6 31 B8 3Ø 45 5Ø B8 8C ØØ
[NN5] 14Ø8: 5Ø 58 5B 89 Ø7 E8 BE 3B E9 9E ØA B8 44 45 5Ø 5B
[BM5] 14Ø9: FF 37 B8 81 ØØ 5Ø E8 E4 B3 58 21 CØ 75 Ø3 E9 Ø3
[SM5] 141Ø: ØØ E9 62 ØØ E8 52 33 E8 38 ØB B8 4A 45 5Ø 5B FF
[2D5] 1411: 37 B8 ØØ ØØ 5Ø E8 8C B3 58 21 CØ 75 Ø3 E9 1B ØØ
[MG5] 1412: B8 3Ø 45 5Ø B8 7B ØØ 5Ø 58 5B 89 Ø7 E8 77 3B B8
[DK5] 1413: 4A 45 5Ø B8 Ø4 ØØ 5Ø 58 5B 89 Ø7 B8 4A 45 5Ø 5B
[VK5] 1414: FF 37 B8 Ø4 ØØ 5Ø E8 94 B3 58 21 CØ 75 Ø3 E9 Ø3
[ØD5] 1415: ØØ E9 3B 31 B8 3Ø 45 5Ø B8 8D ØØ 5Ø 58 5B 89 Ø7
[5M5] 1416: E8 43 3B E9 23 ØA B8 44 45 5Ø 5B FF 37 B8 84 ØØ
[9F5] 1417: 5Ø E8 69 B3 58 21 CØ 75 Ø3 E9 Ø3 ØØ E9 5Ø ØØ E8
[NN5] 1418: D7 32 E8 BD ØA B8 4A 45 5Ø 5B FF 37 B8 Ø4 ØØ 5Ø
[1N5] 1419: E8 4A B3 58 21 CØ 75 Ø3 E9 Ø3 ØØ E9 F1 3Ø E8 D8
[CM5] 142Ø: 46 E8 9E ØA B8 4A 45 5Ø 5B FF 37 B8 Ø4 ØØ 5Ø E8
[LF5] 1421: 2B B3 58 21 CØ 75 Ø3 E9 Ø3 ØØ E9 D2 3Ø B8 3Ø 45
[2M5] 1422: 5Ø B8 8E ØØ 5Ø 58 5B 89 Ø7 E8 DA 3A E9 BA Ø9 B8
[9M5] 1423: 44 45 5Ø 5B FF 37 B8 8D ØØ 5Ø E8 ØØ B3 58 21 CØ
[RG5] 1424: 75 Ø3 E9 Ø3 ØØ E9 69 ØØ E8 6E 32 B8 44 45 5Ø 5B
[PS5] 1425: FF 37 B8 E7 ØØ 5Ø E8 E4 B2 58 21 CØ 75 Ø3 E9 1F
[5E5] 1426: ØØ B8 ØØ ØØ 5Ø ØE E8 ØD ØØ ØC 27 3D 27 2Ø 65 72
[2K5] 1427: 77 61 72 74 65 74 E8 27 B3 E8 7C B3 58 E9 9C 3Ø
[SN5] 1428: E8 36 32 E8 1C ØA B8 4A 45 5Ø 5B FF 37 B8 Ø4 ØØ
[MG5] 1429: 5Ø E8 A9 B2 58 21 CØ 75 Ø3 E9 Ø3 ØØ E9 5Ø 3Ø B8
[9K5] 143Ø: 3Ø 45 5Ø B8 93 ØØ 5Ø 58 5B 89 Ø7 E8 58 3A E9 38
[WK5] 1431: Ø9 B8 44 45 5Ø 5B FF 37 B8 83 ØØ 5Ø E8 7E B2 58
```

```
[4H5] 1432: 21 CØ 75 Ø3 E9 Ø3 ØØ E9 15 ØØ E8 EC 31 B8 3Ø 45
[MF5] 1433: 5Ø B8 Ø1 ØØ 5Ø 58 5B 89 Ø7 E8 2A 3A E9 ØA Ø9 B8
[2G5] 1434: ØØ ØØ 5Ø ØE E8 13 ØØ 12 55 6E 62 65 6B 61 6E 6E
[8L5] 1435: 74 65 72 2Ø 42 65 66 65 68 6C E8 A3 B2 E8 F8 B2
[9H5] 1436: 58 E9 18 3Ø B8 44 45 5Ø 5B FF 37 B8 95 ØØ 5Ø E8
[4K5] 1437: 2B B2 58 21 CØ 75 Ø3 E9 Ø3 ØØ E9 CD Ø1 E8 99 31
[2R5] 1438: E8 6Ø 3F B8 3Ø 45 5Ø B8 44 45 5Ø 5B FF 37 B8 8D
[TM5] 1439: ØØ 5Ø E8 CF B1 58 5B 89 Ø7 B8 3Ø 45 5Ø 5B FF 37
[1M5] 144Ø: B8 44 45 5Ø 5B FF 37 B8 89 ØØ 5Ø E8 B6 B1 E8 2Ø
[TB5] 1441: B2 58 21 CØ 75 Ø3 E9 Ø3 ØØ E9 2B ØØ B8 ØØ ØØ 5Ø
[MD5] 1442: ØE E8 19 ØØ 18 47 4F 54 4F 2Ø 6F 64 65 72 2Ø 47
[AD5] 1443: 4F 53 55 42 2Ø 65 72 77 61 72 74 65 74 E8 2Ø B2
[ZL5] 1444: E8 75 B2 58 E9 95 2F B8 22 45 5Ø B8 E8 ØØ 5Ø 58
[NL5] 1445: 5B 89 Ø7 E8 2B 38 B8 Ø8 45 5Ø B8 Ø4 3C 5Ø B8 Ø2
[WP5] 1446: ØØ 5Ø B8 4Ø 45 5Ø 5B FF 37 E8 DØ B1 E8 A1 B1 5B
[9P5] 1447: FF 37 58 5B 89 Ø7 E8 Ø5 3F E8 FD 3Ø B8 44 45 5Ø
[BL5] 1448: 5B FF 37 B8 ØE ØØ 5Ø E8 73 B1 58 21 CØ 75 Ø3 E9
[5D5] 1449: 34 ØØ B8 ØØ ØØ 5Ø ØE E8 22 ØØ 21 5A 65 69 6C 65
[HK5] 145Ø: 6E 6E 75 6D 6D 65 72 2Ø 6F 64 65 72 2Ø 4C 61 62
[TK5] 1451: 65 6C 2Ø 65 72 77 61 72 74 65 74 2F E8 A1 B1 E8
[VM5] 1452: F6 B1 58 E9 E9 2E E8 7E 46 B8 Ø8 45 5Ø B8 1Ø 45
[WU5] 1453: 5Ø 5B FF 77 Ø2 FF 37 B8 Ø8 45 5Ø 5B FF 37 E8 CE
[KR5] 1454: BB E8 4A B6 E8 86 BB 58 5B 89 Ø7 E8 Ø2 38 B8 44
[8L5] 1455: 45 5Ø 5B FF 37 B8 2C ØØ 5Ø E8 C8 BØ 58 21 CØ 75
[PH5] 1456: Ø3 E9 Ø3 ØØ E9 72 FF B8 Ø4 3C 5Ø B8 Ø2 ØØ 5Ø B8
[NM5] 1457: 4Ø 45 5Ø 5B FF 37 E8 23 B1 E8 F4 BØ B8 14 45 5Ø
[WN5] 1458: 5B FF 77 Ø2 FF 37 E8 44 BB 58 5B 89 Ø7 B8 4Ø 45
[YG5] 1459: 5Ø B8 4Ø 45 5Ø 5B FF 37 B8 Ø1 ØØ 5Ø E8 D1 BØ 58
[NR5] 146Ø: 5B 89 Ø7 B8 4Ø 45 5Ø 5B FF 37 B8 F3 Ø1 5Ø E8 4D
[WD5] 1461: BØ 58 21 CØ 75 Ø3 E9 35 ØØ B8 ØØ ØØ 5Ø ØE E8 23
[3K5] 1462: ØØ 22 5A 75 2Ø 76 69 65 6C 65 2Ø 4F 4E 2F 47 4F
[DG5] 1463: 54 4F 2F 47 4F 53 55 42 2D 41 6E 77 65 69 73 75
[LV5] 1464: 6E 67 65 6E E8 D9 BØ E8 2E B1 58 E9 4E 2E B8 3Ø
[AM5] 1465: 45 5Ø B8 4E ØØ 5Ø B8 3Ø 45 5Ø 5B FF 37 E8 7B BØ
[SS5] 1466: 58 5B 89 Ø7 E8 1F 38 E9 FF Ø6 B8 44 45 5Ø 5B FF
[9C5] 1467: 37 B8 9Ø ØØ 5Ø E8 45 BØ 58 21 CØ 75 Ø3 E9 Ø3 ØØ
[FK5] 1468: E9 B5 ØØ E8 B3 2F B8 Ø8 45 5Ø B8 ØØ ØØ 5Ø 58 5B
[ZM5] 1469: 89 Ø7 E8 99 37 8C D8 Ø5 9C 17 5Ø ØE E8 Ø9 ØØ Ø8
[4R5] 147Ø: 42 72 65 61 6B 2Ø 69 6E E8 75 BØ E8 93 BØ E8 6F
[9D5] 1471: 44 B8 3Ø 45 5Ø B8 15 ØØ 5Ø 58 5B 89 Ø7 E8 C6 37
[YB5] 1472: B8 Ø8 45 5Ø B8 3E ØØ 5Ø B8 Ø2 ØØ 5Ø B8 32 45 5Ø
[TT5] 1473: 5B FF 37 B8 Ø1 ØØ 5Ø E8 Ø1 BØ E8 1F BØ E8 FØ AF
[9L5] 1474: 5B FF 37 58 5B 89 Ø7 E8 44 37 B8 3Ø 45 5Ø B8 16
[LF5] 1475: ØØ 5Ø 58 5B 89 Ø7 E8 8D 37 B8 3Ø 45 5Ø B8 15 ØØ
[8D5] 1476: 5Ø 58 5B 89 Ø7 E8 7E 37 B8 3Ø 45 5Ø B8 14 ØØ 5Ø
[MH5] 1477: 58 5B 89 Ø7 E8 6F 37 B8 22 45 5Ø B8 58 ØØ 5Ø 58
[8H5] 1478: 5B 89 Ø7 E8 1B 36 B8 3Ø 45 5Ø B8 Ø1 ØØ 5Ø 58 5B
[6K5] 1479: 89 Ø7 E8 51 37 E9 31 Ø6 B8 44 45 5Ø 5B FF 37 B8
[BH5] 148Ø: B7 ØØ 5Ø E8 77 AF 58 21 CØ 75 Ø3 E9 Ø3 ØØ E9 24
[KM5] 1481: Ø1 E8 E5 2E B8 44 45 5Ø 5B FF 37 B8 23 ØØ 5Ø E8
[4G5] 1482: 5B AF 58 21 CØ 75 Ø3 E9 12 ØØ B8 Ø8 45 5Ø B8 ØØ
[YL5] 1483: ØØ 5Ø 58 5B 89 Ø7 E8 B5 36 E9 1B ØØ E8 BA 2E E8
[YK5] 1484: 81 3C 8C D8 Ø5 7C 17 5Ø ØE E8 Ø2 ØØ Ø1 2C E8 8F
[4W5] 1485: AF E8 AD AF E8 1B 3C B8 44 45 5Ø 5B FF 37 B8 ØØ
[2P5] 1486: ØØ 5Ø E8 DF AE B8 44 45 5Ø 5B FF 37 B8 3A ØØ 5Ø
[VR5] 1487: E8 D1 AE E8 3B AF 58 21 CØ 75 Ø3 E9 Ø3 ØØ E9 7E
[GM5] 1488: ØØ E8 5E Ø6 B8 4A 45 5Ø 5B FF 37 E8 Ø8 ØØ E9 5C
[9K5] 1489: F8 5C 7F 89 16 5D E8 98 B4 B8 3Ø 45 5Ø B8 16 ØØ
[EB5] 149Ø: 5Ø 58 5B 89 Ø7 E9 ØC ØØ B8 3Ø 45 5Ø B8 46 ØØ 5Ø
[3H5] 1491: 58 5B 89 Ø7 E8 8F 36 B8 3Ø 45 5Ø B8 72 ØØ 5Ø 58
[VH5] 1492: 5B 89 Ø7 E9 ØC ØØ B8 3Ø 45 5Ø B8 73 ØØ 5Ø 58 5B
[NH5] 1493: 89 Ø7 E8 71 36 B8 44 45 5Ø 5B FF 37 B8 2C ØØ 5Ø
[EV5] 1494: E8 9A AE B8 44 45 5Ø 5B FF 37 B8 3B ØØ 5Ø E8 8C
[MN5] 1495: AE E8 B2 AE 58 21 CØ 75 Ø3 E9 Ø3 ØØ E9 15 ØØ E8
[ZK5] 1496: F7 2D B8 3Ø 45 5Ø B8 74 ØØ 5Ø 58 5B 89 Ø7 E8 35
[CH5] 1497: 36 E9 6D FF B8 3Ø 45 5Ø B8 14 ØØ 5Ø 58 5B 89 Ø7
[TH5] 1498: E8 23 36 B8 22 45 5Ø B8 58 ØØ 5Ø 58 5B 89 Ø7 E8
[AM5] 1499: CF 34 E9 F4 Ø4 B8 44 45 5Ø 5B FF 37 B8 96 ØØ 5Ø
[WN5] 15ØØ: E8 3A AE 58 21 CØ 75 Ø3 E9 Ø3 ØØ E9 5E ØØ E8 A8
[ØM5] 1501: 2D E8 6F 3B 8C D8 Ø5 7C 17 5Ø ØE E8 Ø2 ØØ Ø1 2C
[5U5] 1502: E8 7D AE E8 9B AE E8 Ø9 3B E8 57 3B B8 44 45 5Ø
[2M5] 1503: 5B FF 37 B8 2C ØØ 5Ø E8 Ø3 AE 58 21 CØ 75 Ø3 E9
[GH5] 1504: 12 ØØ B8 Ø8 45 5Ø B8 ØØ ØØ 5Ø 58 5B 89 Ø7 E8 5D
[XG5] 1505: 35 E9 Ø6 ØØ E8 62 2D E8 29 3B B8 3Ø 45 5Ø B8 75
[ØK5] 1506: ØØ 5Ø 58 5B 89 Ø7 E8 9D 35 E9 7D Ø4 B8 44 45 5Ø
[RM5] 1507: 5B FF 37 B8 A4 ØØ 5Ø E8 C3 AD 58 21 CØ 75 Ø3 E9
[NP5] 1508: Ø3 ØØ E9 8E ØØ E8 31 2D E8 4B 2D B8 7C 45 5Ø B8
[4N5] 1509: 4A 45 5Ø 5B FF 37 58 5B 89 Ø7 8C D8 Ø5 7C 17 5Ø
[AS5] 151Ø: ØE E8 Ø2 ØØ Ø1 2C E8 F7 AD E8 15 AE E8 83 3A E8
[8R5] 1511: 24 2D B8 4A 45 5Ø 5B FF 37 B8 7C 45 5Ø 5B FF 37
[MM5] 1512: E8 7A AD 58 21 CØ 75 Ø3 E9 Ø3 ØØ E9 21 2B B8 4A
[1P5] 1513: 45 5Ø 5B FF 37 E8 Ø8 ØØ 73 5E 82 5E 7F 89 91 5E
[NM5] 1514: E8 ØE B3 B8 3Ø 45 5Ø B8 77 ØØ 5Ø 58 5B 89 Ø7 E9
[CL5] 1515: 1B ØØ B8 3Ø 45 5Ø B8 78 ØØ 5Ø 58 5B 89 Ø7 E8 ØC
[HD5] 1516: ØØ B8 3Ø 45 5Ø B8 79 ØØ 5Ø 58 5B 89 Ø7 E8 F6 34
[LR5] 1517: E9 D6 Ø3 B8 44 45 5Ø 5B FF 37 B8 C4 ØØ 5Ø E8 1C
[UH5] 1518: AD 58 21 CØ 75 Ø3 E9 Ø3 ØØ E9 3Ø ØØ E8 8A 2C E8
[ZL5] 1519: 7A 3A 8C D8 Ø5 7C 17 5Ø ØE E8 Ø2 ØØ Ø1 2C E8 5F
[UU5] 152Ø: AD E8 7D AD E8 EB 39 E8 62 3A B8 3Ø 45 5Ø B8 7A
[NK5] 1521: ØØ 5Ø 58 5B 89 Ø7 E8 AD 34 E9 8D Ø3 B8 44 45 5Ø
[7N5] 1522: 5B FF 37 B8 C8 ØØ 5Ø E8 D3 AC 58 21 CØ 75 Ø3 E9
[5K5] 1523: Ø3 ØØ E9 74 ØØ E8 41 2C E8 Ø8 3A B8 44 45 5Ø 5B
[5R5] 1524: FF 37 B8 2C ØØ 5Ø E8 B4 AC 58 21 CØ 75 Ø3 E9 12
[Z95] 1525: ØØ B8 Ø8 45 5Ø B8 ØØ ØØ 5Ø 58 5B 89 Ø7 E8 ØE 34
[WP5] 1526: E9 Ø6 ØØ E8 13 2C E8 DA 39 B8 44 45 5Ø 5B FF 37
[PG5] 1527: B8 2C ØØ 5Ø E8 86 AC 58 21 CØ 75 Ø3 E9 12 ØØ B8
[VG5] 1528: Ø8 45 5Ø B8 ØØ ØØ 5Ø 58 5B 89 Ø7 E8 EØ 33 E9 Ø6
[VM5] 1529: ØØ E8 E5 2B E8 AC 39 B8 3Ø 45 5Ø B8 7D ØØ 5Ø 58
[HH5] 153Ø: 5B 89 Ø7 E8 2Ø 34 E9 ØØ Ø3 B8 44 45 5Ø 5B FF 37
[5G5] 1531: B8 92 ØØ 5Ø E8 46 AC 58 21 CØ 75 Ø3 E9 Ø3 ØØ E9
[NK5] 1532: 5D ØØ E8 B4 2B B8 22 45 5Ø B8 BB ØØ 5Ø 58 5B 89
[6H5] 1533: Ø7 E8 AD 32 B8 Ø8 45 5Ø B8 ØC 45 5Ø 5B FF 77 Ø2
[DN5] 1534: FF 37 B8 ØF ØØ 5Ø E8 83 AF E8 42 B1 B8 1Ø ØØ 5Ø
[FT5] 1535: E8 79 AF E8 7C B1 E8 D8 B1 E8 71 B6 B8 18 45 5Ø
[BR5] 1536: 5B FF 37 E8 ØA AC 58 5B 89 Ø7 E8 E3 32 B8 3Ø 45
[9H5] 1537: 5Ø B8 7E ØØ 5Ø 58 5B 89 Ø7 E8 AA 33 E9 8A Ø2 B8
```

Listing 1. »sbasic.exe« ist der Hexdump des Basic-Compilers. Geben Sie dieses Listing nur dann ein, wenn Sie keine frühere Version von SBasic besitzen.

```
[4R5] 1538: 44 45 5Ø 5B FF 37 B8 8A ØØ 5Ø E8 DØ AB 58 21 CØ
[EK5] 1539: 75 Ø3 E9 Ø3 ØØ E9 AF ØØ E8 3E 2B B8 22 45 5Ø B8
[7M5] 154Ø: BB ØØ 5Ø 58 5B 89 Ø7 E8 37 32 B8 Ø8 45 5Ø B8 ØC
[GS5] 1541: 45 5Ø 5B FF 77 Ø2 FF 37 B8 ØF ØØ 5Ø E8 ØD AF E8
[EP5] 1542: CC BØ B8 1Ø ØØ 5Ø E8 Ø3 AF E8 Ø6 B1 E8 62 B1 E8
[EK5] 1543: FB B5 58 5B 89 Ø7 E8 77 32 B8 3Ø 45 5Ø B8 7E ØØ
[8H5] 1544: 5Ø 58 5B 89 Ø7 E8 3E 33 B8 44 45 5Ø 5B FF 37 B8
[DL5] 1545: ØØ ØØ 5Ø E8 2E AB B8 44 45 5Ø 5B FF 37 B8 3A ØØ
[7L5] 1546: 5Ø E8 2Ø AB E8 8A AB 58 21 CØ 75 Ø3 E9 2Ø ØØ B8
[GK5] 1547: Ø8 45 5Ø B8 DE ØF 5Ø B8 Ø2 ØØ 5Ø B8 ØØ ØØ 5Ø E8
[MU5] 1548: 7A AB E8 4B AB 5B FF 37 58 5B 89 Ø7 E9 Ø3 ØØ E8
[EL5] 1549: 75 4Ø B8 22 45 5Ø B8 E9 ØØ 5Ø 58 B9 Ø7 E8 AØ
[NP5] 155Ø: 31 E8 9A 38 E9 C2 Ø1 B8 44 45 5Ø 5B FF 37 B8 B9
[FE5] 1551: ØØ 5Ø E8 Ø8 AB 58 21 CØ 75 Ø3 E9 Ø3 ØØ E9 18 ØØ
[VH5] 1552: E8 76 2A E8 66 38 B8 3Ø 45 5Ø B8 82 ØØ 5Ø 58 5B
[2L5] 1553: 89 Ø7 E8 B1 32 E9 91 Ø1 B8 44 45 5Ø 5B FF 37 B8
[SH5] 1554: C5 ØØ 5Ø E8 D7 AA 58 21 CØ 75 Ø3 E9 Ø3 ØØ E9 15
[GD5] 1555: ØØ E8 45 2A B8 3Ø 45 5Ø B8 83 ØØ 5Ø 58 5B 89 Ø7
[BM5] 1556: E8 83 32 E9 63 Ø1 B8 44 45 5Ø 5B FF 37 B8 C2 ØØ
[DK5] 1557: 5Ø E8 A9 AA 58 21 CØ 75 Ø3 E9 Ø3 ØØ E9 61 ØØ E8
[8M5] 1558: 17 2A E8 FD Ø1 B8 4A 45 5Ø 5B FF 37 B8 Ø4 ØØ 5Ø
[FL5] 1559: E8 8A AA 58 21 CØ 75 Ø3 E9 Ø3 ØØ E9 31 28 8C D8
[NN5] 156Ø: Ø5 7C 17 5Ø ØE E8 Ø2 ØØ Ø1 2C E8 D3 AA E8 F1 AA
[ZL5] 1561: E8 5F 37 E8 AD 37 8C D8 Ø5 7C 17 5Ø ØE E8 Ø2 ØØ
[ØW5] 1562: Ø1 2C E8 BB AA E8 D9 AA E8 47 37 E8 95 37 B8 3Ø
[TH5] 1563: 45 5Ø B8 85 ØØ 5Ø 58 5B 89 Ø7 E8 Ø9 32 E9 E9 ØØ
[SL5] 1564: B8 44 45 5Ø 5B FF 37 B8 C3 ØØ 5Ø E8 2F AA 58 21
[BH5] 1565: CØ 75 Ø3 E9 Ø3 ØØ E9 62 ØØ E8 9D 29 E8 83 Ø1 B8
[3M5] 1566: 4A 45 5Ø 5B FF 37 B8 Ø4 ØØ 5Ø E8 1Ø AA 58 21 CØ
[VK5] 1567: 75 Ø3 E9 Ø3 ØØ E9 B7 27 B8 44 45 5Ø 5B FF 37 B8
[SH5] 1568: 2C ØØ 5Ø E8 F7 A9 58 21 CØ 75 Ø3 E9 15 ØØ B8 Ø8
[TF5] 1569: 45 5Ø B8 Ø1 ØØ 5Ø E8 41 AA 58 5B 89 Ø7 E8 4E 31
[NG5] 157Ø: E9 Ø6 ØØ E8 53 29 E8 1A 37 B8 3Ø 45 5Ø B8 86 ØØ
[NH5] 1571: 5Ø 58 5B 89 Ø7 E8 8E 31 E9 6E ØØ B8 44 45 5Ø 5B
[XP5] 1572: FF 37 B8 9C ØØ 5Ø E8 B4 A9 58 21 CØ 75 Ø3 E9 Ø3
[4G5] 1573: ØØ E9 3Ø ØØ E8 22 29 E8 E9 36 8C D8 Ø5 7C 17 5Ø
[EP5] 1574: ØE E8 Ø2 ØØ Ø1 2C E8 F7 A9 E8 15 AA E8 83 36 E8
[4K5] 1575: D1 36 B8 3Ø 45 5Ø B8 88 ØØ 5Ø 58 5B 89 Ø7 E8 45
[H95] 1576: 31 E9 25 ØØ B8 ØØ ØØ 5Ø ØE E8 13 ØØ 12 55 6E 62
[6K5] 1577: 65 6B 61 6E 6E 74 65 72 2Ø 42 65 66 65 68 6C E8
[2R5] 1578: BE A9 E8 13 AA 58 E9 33 27 B8 44 45 5Ø 5B FF 37
[4H5] 1579: 58 21 CØ 75 Ø3 E9 Ø3 ØØ E9 91 CØ B8 ØØ ØØ 5Ø 8F
[MK5] 158Ø: Ø6 3Ø ØØ B8 6Ø 45 5Ø 5B FF 37 B8 Ø1 ØØ 5Ø E8 4A
[TE5] 1581: A9 8F Ø6 7E 45 B8 Ø1 ØØ 5Ø 8F Ø6 8Ø 45 FF 36 8Ø
[XK5] 1582: 45 E8 8Ø AA 58 8Ø E4 Ø5 2E 88 26 D8 62 E9 ØF ØØ
[ZP5] 1583: FF 36 3Ø ØØ FF 36 8Ø 45 E8 15 A9 8F Ø6 3Ø ØØ FF
[AV5] 1584: 36 3Ø ØØ FF 36 7E 45 EB ØØ E8 DE A8 EB Ø3 E8 C6
[AG5] 1585: A8 58 21 CØ 75 Ø3 E9 3A ØØ B8 E2 28 5Ø B8 Ø2 ØØ
[7F5] 1586: 5Ø B8 CA 2C 5Ø B8 Ø2 ØØ 5Ø B8 3Ø ØØ 5Ø 5B FF 37
[DX5] 1587: E8 Ø9 A9 E8 DA A8 5B FF 37 E8 ØØ A9 E8 D1 A8 B8
[1N5] 1588: 14 45 5Ø 5B FF 77 Ø2 FF 37 E8 21 B3 58 5B 89 Ø7
[5N5] 1589: E9 9D FF B8 6Ø 45 5Ø B8 ØØ ØØ 5Ø 58 5B 89 Ø7 E9
[WK5] 159Ø: 6B BE E8 17 28 B8 44 45 5Ø 5B FF 37 B8 ØØ ØØ 5Ø
[DR5] 1591: E8 51 A8 B8 44 45 5Ø 5B FF 37 B8 3A ØØ 5Ø E8 43
[HR5] 1592: A8 E8 AD A8 B8 44 45 5Ø 5B FF 37 B8 2C ØØ 5Ø E8
[SM5] 1593: 32 A8 E8 9C A8 B8 44 45 5Ø 5B FF 37 B8 3B ØØ 5Ø
[RP5] 1594: E8 21 A8 E8 8B A8 58 21 CØ 75 Ø3 E9 ØD ØØ B8 4A
[TK5] 1595: 45 5Ø B8 ØØ ØØ 5Ø 58 5B 89 Ø7 C3 B8 C8 36 5Ø B8
[6R5] 1596: Ø2 ØØ 5Ø 5B 82 45 5Ø 5B FF 37 E8 6F A8 E8 4Ø A8
[TD5] 1597: B8 F8 ØØ 5Ø 58 5B 89 Ø7 B8 82 45 5Ø B8 82 45 5Ø
[GM5] 1598: 5B FF 37 B8 Ø1 ØØ 5Ø E8 26 A8 58 5B 89 Ø7 B8 44
[BP5] 1599: 45 5Ø 5B FF 37 B8 EA ØØ 5Ø E8 C8 A7 58 21 CØ 75
[ØB5] 16ØØ: Ø3 E9 39 ØØ B8 C8 36 5Ø B8 Ø2 ØØ 5Ø B8 82 45 5Ø
[ØN5] 16Ø1: 5B FF 37 E8 26 A8 E8 F7 A7 B8 Ø1 ØØ 5Ø 58 5B 89
[UE5] 16Ø2: Ø7 B8 82 45 5Ø B8 82 45 5Ø 5B FF 37 B8 Ø1 ØØ 5Ø
[BU5] 16Ø3: E8 DD A7 58 5B 89 Ø7 E8 3F 27 E9 B1 FF B8 44 45
[MH5] 16Ø4: 5Ø 5B FF 37 B8 E9 ØØ 5Ø E8 79 A7 58 21 CØ 75 Ø3
[3P5] 16Ø5: E9 Ø6 ØØ E8 23 27 E9 95 FF B8 44 45 5Ø 5B FF 37
[5H5] 16Ø6: B8 D3 ØØ 5Ø E8 5D A7 58 21 CØ 75 Ø3 E9 39 ØØ B8
[FR5] 16Ø7: C8 36 5Ø B8 Ø2 ØØ 5Ø B8 82 45 5Ø 5B FF 37 E8 BB
[5G5] 16Ø8: A7 E8 8C A7 B8 ØØ ØØ 5Ø 58 5B 89 Ø7 B8 82 45 5Ø
[LK5] 16Ø9: B8 82 45 5Ø 5B FF 37 B8 Ø1 ØØ 5Ø E8 72 A7 58 5B
[1M5] 161Ø: 89 Ø7 E8 D4 26 E9 46 FF B8 4A 45 5Ø B8 Ø1 ØØ 5Ø
[ZH5] 1611: 58 5B 89 Ø7 B8 44 45 5Ø 5B FF 37 B8 11 ØØ 5Ø E8
[YP5] 1612: EF A6 B8 44 45 5Ø 5B FF 37 B8 1A ØØ 5Ø E8 CE A6
[ML5] 1613: E8 5E A7 58 21 CØ 75 Ø3 E9 Ø3 ØØ E9 1F ØØ B8 Ø8
[YG5] 1614: 45 5Ø B8 44 45 5Ø 5B FF 37 B8 11 ØØ 5Ø E8 2B A7
[2M5] 1615: 58 5B 89 Ø7 E8 77 2E E8 7F 26 E9 42 18 B8 44 45
[TH5] 1616: 5Ø 5B FF 37 B8 ØF ØØ 5Ø E8 F2 A6 58 21 CØ 75 Ø3
[EH5] 1617: E9 Ø3 ØØ E9 1B ØØ E8 E4 25 B8 Ø8 45 5Ø B8 44 45
[VR5] 1618: 5Ø 5B FF 37 58 5B 89 Ø7 E8 43 2E E8 4B 26 E9 ØE
[6N5] 1619: 18 B8 44 45 5Ø 5B FF 37 B8 1C ØØ 5Ø E8 BE A6 B8
[NS5] 162Ø: 44 45 5Ø 5B FF 37 B8 ØB ØØ 5Ø E8 BØ A6 E8 D6 A6
[XR5] 1621: B8 44 45 5Ø 5B FF 37 B8 ØC ØØ 5Ø E8 9F A6 E8 C5
[ØG5] 1622: A6 58 21 CØ 75 Ø3 E9 Ø3 ØØ E9 ØC ØØ E8 C7 25 E8
[1R5] 1623: FC 2D E8 Ø4 26 E9 C7 17 B8 44 45 5Ø 5B FF 37 B8
[XK5] 1624: 1D ØØ 5Ø E8 77 A6 58 21 CØ 75 Ø3 E9 Ø3 ØØ E9 5E
[GB5] 1625: ØØ B8 4A 45 5Ø B8 Ø2 ØØ 5Ø 58 5B 89 Ø7 E8 96 25
[FG5] 1626: B8 C8 36 5Ø B8 Ø2 ØØ 5Ø B8 82 45 5Ø 5B FF 37 E8
[7P5] 1627: 8A A6 E8 5B A6 B8 Ø8 45 5Ø 5B FF 37 58 5B 89 Ø7
[9N5] 1628: E8 73 25 E8 A8 2D B8 Ø8 45 5Ø B8 C8 36 5Ø B8 Ø2
[WN5] 1629: ØØ 5Ø B8 82 45 5Ø 5B FF 37 E8 6Ø A6 E8 31 A6 5B
[HR5] 163Ø: FF 37 58 5B 89 Ø7 E8 85 2D E8 8D 25 E9 5Ø 17 B8
[3M5] 1631: 44 45 5Ø 5B FF 37 B8 1F ØØ 5Ø E8 ØØ A6 58 21 CØ
[6C5] 1632: 75 Ø3 E9 Ø3 ØØ E9 1D Ø1 B8 4A 45 5Ø B8 Ø3 ØØ 5Ø
[PB5] 1633: 58 5B 89 Ø7 B8 ØØ ØØ 5Ø 8F Ø6 3Ø ØØ B8 Ø3 ØØ 5Ø
[PH5] 1634: 8F Ø6 84 45 B8 Ø1 ØØ 5Ø 8F Ø6 86 45 FF 36 86 45
[7N5] 1635: E8 31 A7 58 8Ø E4 Ø5 2E 88 26 27 66 E9 ØF ØØ FF
[BL5] 1636: 36 3Ø ØØ FF 36 86 45 E8 C6 A5 8F Ø6 3Ø ØØ FF 36
[3R5] 1637: 3Ø ØØ FF 36 84 45 EB ØØ E8 8F A5 EB Ø3 E8 77 A5
[9K5] 1638: 58 21 CØ 75 Ø3 E9 3Ø ØØ E8 CB 24 B8 C8 36 5Ø B8
[WL5] 1639: Ø2 ØØ 5Ø B8 82 45 5Ø 5B FF 37 B8 3Ø ØØ 5Ø 5B FF
[9T5] 164Ø: 37 E8 8C A5 E8 B5 A5 E8 86 A5 B8 Ø8 45 5Ø 5B FF
[EK5] 1641: 37 58 5B 89 Ø7 E9 A7 FF B8 Ø3 ØØ 5Ø 8F Ø6 3Ø ØØ
[EK5] 1642: B8 ØØ ØØ 5Ø 8F Ø6 88 45 B8 Ø1 ØØ 5Ø E8 AB A5 8F
[WN5] 1643: Ø6 8A 45 FF 36 8A 45 E8 AA A6 58 8Ø E4 Ø5 2E 88
[6T5] 1644: 26 AE 66 E9 ØF ØØ FF 36 3Ø ØØ FF 36 8A 45 E8 3F
[CF5] 1645: A5 8F Ø6 3Ø ØØ FF 36 3Ø ØØ FF 36 88 45 EB ØØ E8
[TH5] 1646: Ø8 A5 EB Ø3 E8 FØ A4 58 21 CØ 75 Ø3 E9 3Ø ØØ B8
[6H5] 1647: Ø8 45 5Ø B8 C8 36 5Ø B8 Ø2 ØØ 5Ø B8 82 45 5Ø 5B
[WS5] 1648: FF 37 B8 3Ø ØØ 5Ø 5B FF 37 E8 Ø4 A5 E8 2D A5 E8
[9W5] 1649: FE A4 5B FF 37 58 5B 89 Ø7 E8 52 2C E9 A7 FF E8
[EH5] 165Ø: 57 24 E9 1A 16 B8 44 45 5Ø 5B FF 37 B8 22 ØØ 5Ø
[CN5] 1651: E8 CA A4 58 21 CØ 75 Ø3 E9 Ø3 ØØ E9 8C ØØ B8 4A
[MF5] 1652: 45 5Ø B8 Ø4 ØØ 5Ø 58 5B 89 Ø7 8C D8 Ø5 9C 17 5Ø
[TL5] 1653: ØE E8 Ø1 ØØ ØØ E8 Ø8 A5 E8 26 A5 E8 9F 23 B8 44
[6G5] 1654: 45 5Ø 5B FF 37 B8 22 ØØ 5Ø E8 91 A4 B8 44 45 5Ø
[8N5] 1655: 5B FF 37 B8 ØØ ØØ 5Ø E8 83 A4 E8 A9 A4 58 21 CØ
[AM5] 1656: 75 Ø3 E9 26 ØØ 8C D8 Ø5 9C 17 5Ø 8C D8 Ø5 9C 17
[KH5] 1657: 5Ø B8 ØØ ØØ 5Ø E8 C8 A4 B8 44 45 5Ø 5B FF 37 E8
[UX5] 1658: AF A5 E8 7D A5 E8 D9 A4 E9 BØ FF B8 44 45 5Ø 5B
[RL5] 1659: FF 37 B8 22 ØØ 5Ø E8 ØB A4 58 21 CØ 75 Ø3 E9 Ø3
[8N5] 166Ø: ØØ E8 B5 23 E8 99 38 E9 75 15 B8 44 45 5Ø 5B FF
[CK5] 1661: 37 B8 41 ØØ 5Ø E8 D9 A3 B8 44 45 5Ø 5B FF 37 B8
[WL5] 1662: 5A ØØ 5Ø E8 B8 A3 E8 48 A4 58 21 CØ 75 Ø3 E9 Ø3
[EK5] 1663: ØØ E9 9C ØØ E8 9F 23 B8 46 45 5Ø 5B FF 37 58 21
[VL5] 1664: CØ 75 Ø3 E9 Ø3 ØØ E9 9C 12 B8 4A 45 5Ø 5B FF 37
[MP5] 1665: E8 Ø8 ØØ ØF 68 3Ø 68 7F 89 EE 67 E8 93 A9 B8 Ø8
[HF5] 1666: 45 5Ø B8 ØØ ØØ 5Ø 58 5B 89 Ø7 E8 41 2B B8 3Ø 45
[GF5] 1667: 5Ø B8 12 ØØ 5Ø 58 5B 89 Ø7 E8 8A 2B E9 ØØ 15 B8
[8K5] 1668: 22 45 5Ø B8 5B ØØ 5Ø 58 5B 89 Ø7 E8 33 2A B8 Ø8
[CS5] 1669: 45 5Ø B8 FF 37 5Ø 58 5B 89 Ø7 E8 93 2A E9 DF 14
[XL5] 167Ø: B8 Ø8 45 5Ø B8 5B FF 5Ø 58 5B 89 Ø7 E8 81 2A B8
[2K5] 1671: Ø8 45 5Ø B8 77 Ø2 5Ø 58 5B 89 Ø7 E8 72 2A B8 Ø8
[RR5] 1672: 45 5Ø B8 FF 37 5Ø 58 5B 89 Ø7 E8 63 2A E9 AF 14
[FK5] 1673: B8 44 45 5Ø 5B FF 37 B8 28 ØØ 5Ø E8 5F A3 58 21
[MR5] 1674: CØ 75 Ø3 E9 Ø3 ØØ E9 1E ØØ E8 CD 22 E8 B3 FA 8C
[PK5] 1675: D8 Ø5 7C 17 5Ø ØE E8 Ø2 ØØ Ø1 29 E8 A2 A3 E8 CØ
[6S5] 1676: A3 E8 2E 3Ø E9 78 14 B8 44 45 5Ø 5B FF 37 B8 FF
[ØG5] 1677: ØØ 5Ø E8 28 A3 58 21 CØ 75 Ø3 E9 Ø3 ØØ E9 68 ØA
[5R5] 1678: E8 1A 22 B8 8C 45 5Ø B8 44 45 5Ø 5B FF 37 58 5B
[2K5] 1679: 89 Ø7 E8 84 22 B8 8C 45 5Ø 5B FF 37 B8 96 ØØ 5Ø
[HM5] 168Ø: E8 FA A2 58 21 CØ 75 Ø3 E9 Ø3 ØØ E9 4B ØØ 8C D8
[DE5] 1681: Ø5 7C 17 5Ø ØE E8 Ø2 ØØ Ø1 28 E8 43 A3 E8 61 A3
[3M5] 1682: E8 CF 2F E8 1D 3Ø B8 3Ø 45 5Ø B8 19 ØØ 5Ø 58 5B
[ZE5] 1683: 89 Ø7 E8 91 2A B8 4A 45 5Ø B8 Ø4 ØØ 5Ø 58 5B 89
[YG5] 1684: Ø7 8C D8 Ø5 7C 17 5Ø ØE E8 Ø2 ØØ Ø1 29 E8 1Ø A3
[7V5] 1685: E8 2E A3 E8 9C 2F E9 E6 13 B8 8C 45 5Ø 5B FF 37
[TF5] 1686: B8 92 ØØ 5Ø E8 96 A2 58 21 CØ 75 Ø3 E9 Ø3 ØØ E9
[UL5] 1687: 4F ØØ 8C D8 Ø5 7C 17 5Ø ØE E8 Ø2 ØØ Ø1 28 E8 DF
[1X5] 1688: A2 E8 FD A2 E8 6B 2F E8 D8 F9 B8 4A 45 5Ø 5B FF
[PB5] 1689: 37 B8 Ø4 ØØ 5Ø E8 65 A2 58 21 CØ 75 Ø3 E9 Ø3 ØØ
[XH5] 169Ø: E9 ØC 2Ø B8 3Ø 45 5Ø B8 25 ØØ 5Ø 58 5B 89 Ø7 E8
[8K5] 1691: 14 2A B8 4A 45 5Ø B8 Ø1 ØØ 5Ø 58 5B 89 Ø7 E9 8Ø
[EP5] 1692: FF B8 8C 45 5Ø 5B FF 37 B8 A3 ØØ 5Ø E8 2E A2 58
[KE5] 1693: 21 CØ 75 Ø3 E9 Ø3 ØØ E9 36 ØØ 8C D8 Ø5 7C 17 5Ø
[YK5] 1694: ØE E8 Ø2 ØØ Ø1 28 E8 77 A2 E8 95 A2 E8 Ø3 2F E8
[WK5] 1695: 51 2F B8 3Ø 45 5Ø B8 27 ØØ 5Ø 58 5B 89 Ø7 E8 C5
[LF5] 1696: 29 B8 4A 45 5Ø B8 Ø1 ØØ 5Ø 58 5B 89 Ø7 E9 31 FF
[PM5] 1697: B8 8C 45 5Ø 5B FF 37 B8 97 ØØ 5Ø E8 DF A1 58 21
[5M5] 1698: CØ 75 Ø3 E9 Ø3 ØØ E9 2A ØØ 8C D8 Ø5 7C 17 5Ø ØE
[DL5] 1699: E8 Ø2 ØØ Ø1 28 E8 28 A2 E8 46 A2 E8 B4 2E E8 Ø2
[TD5] 17ØØ: 2F B8 3Ø 45 5Ø B8 36 ØØ 5Ø 58 5B 89 Ø7 E8 76 29
[CX5] 17Ø1: E9 AE FF B8 8C 45 5Ø 5B FF 37 B8 95 ØØ 5Ø E8 9C
[NF5] 17Ø2: A1 58 21 CØ 75 Ø3 E9 Ø3 ØØ E9 43 ØØ 8C D8 Ø5 7C
[NH5] 17Ø3: 17 5Ø ØE E8 Ø2 ØØ Ø1 28 E8 E5 A1 E8 Ø3 A2 E8 71
[ZR5] 17Ø4: 2E E8 DE F8 B8 4A 45 5Ø 5B FF 37 B8 Ø4 ØØ 5Ø E8
[VF5] 17Ø5: 6B A1 58 21 CØ 75 Ø3 E9 Ø3 ØØ E9 12 1F B8 3Ø 45
[BH5] 17Ø6: 5Ø B8 37 ØØ 5Ø 58 5B 89 Ø7 E8 1A 29 E9 52 FF B8
[PL5] 17Ø7: 8C 45 5Ø 5B FF 37 B8 81 ØØ 5Ø E8 4Ø A1 58 21 CØ
[TG5] 17Ø8: 75 Ø3 E9 Ø3 ØØ E9 83 ØØ 8C D8 Ø5 7C 17 5Ø ØE E8
[PN5] 17Ø9: Ø2 ØØ Ø1 28 E8 89 A1 E8 A7 A1 E8 15 2E E8 82 F8
[BK5] 171Ø: B8 4A 45 5Ø 5B FF 37 B8 Ø4 ØØ 5Ø E8 ØF A1 58 21
[EM5] 1711: CØ 75 Ø3 E9 Ø3 ØØ E9 B6 1E B8 44 45 5Ø 5B FF 37
[NG5] 1712: B8 29 ØØ 5Ø E8 BD AØ 58 21 CØ 75 Ø3 E9 12 ØØ B8
[XL5] 1713: Ø8 45 5Ø B8 FF ØØ 5Ø 58 5B 89 Ø7 E8 5Ø 28 E9 18
[7H5] 1714: ØØ 8C D8 Ø5 7C 17 5Ø ØE E8 Ø2 ØØ Ø1 2C E8 3Ø A1
[CR5] 1715: E8 4E A1 E8 BC 2D E8 ØA 2E B8 3Ø 45 5Ø B8 38 ØØ
[KR5] 1716: 5Ø 58 5B 89 Ø7 E8 7E 28 E9 EA FD B8 8C 45 5Ø 5B
[SL5] 1717: FF 37 B8 82 ØØ 5Ø E8 A4 AØ 58 21 CØ 75 Ø3 E9 Ø3
[XD5] 1718: ØØ E9 5B ØØ 8C D8 Ø5 7C 17 5Ø ØE E8 Ø2 ØØ Ø1 28
[3W5] 1719: E8 ED AØ E8 ØB A1 E8 79 2D E8 E6 F7 B8 4A 45 5Ø
[NH5] 172Ø: 5B FF 37 B8 Ø4 ØØ 5Ø E8 73 AØ 58 21 CØ 75 Ø3 E9
[GL5] 1721: Ø3 ØØ E9 1A 1E 8C D8 Ø5 7C 17 5Ø ØE E8 Ø2 ØØ Ø1
[KU5] 1722: 2C E8 BC AØ E8 DA AØ E8 48 2D E8 96 2D B8 3Ø 45
[DH5] 1723: 5Ø B8 39 ØØ 5Ø 58 5B 89 Ø7 E8 ØA 28 E9 76 FD B8
[SL5] 1724: 8C 45 5Ø 5B FF 37 B8 83 ØØ 5Ø E8 3Ø AØ 58 21 CØ
[CH5] 1725: 75 Ø3 E9 Ø3 ØØ E9 9B ØØ 8C D8 Ø5 7C 17 5Ø ØE E8
[MM5] 1726: Ø2 ØØ Ø1 28 E8 79 AØ E8 97 AØ E8 Ø5 2D E8 72 F7
[TM5] 1727: B8 4A 45 5Ø 5B FF 37 B8 Ø4 ØØ 5Ø E8 FF 9F 58 21
[4P5] 1728: CØ 75 Ø3 E9 Ø3 ØØ E9 A6 1D 8C D8 Ø5 7C 17 5Ø ØE
[MM5] 1729: E8 Ø2 ØØ Ø1 2C E8 48 AØ E8 66 AØ E8 D4 2C E8 22
[TK5] 173Ø: 2D B8 44 45 5Ø 5B FF 37 B8 29 ØØ 5Ø E8 95 9F 58
[8G5] 1731: 21 CØ 75 Ø3 E9 12 ØØ B8 Ø8 45 5Ø B8 FF ØØ 5Ø 58
[4N5] 1732: 5B 89 Ø7 E8 28 27 E9 18 ØØ 8C D8 Ø5 7C 17 5Ø ØE
[3N5] 1733: E8 Ø2 ØØ Ø1 2C E8 ØB AØ E8 26 AØ E8 94 2C E8 E2
[BD5] 1734: 2C B8 3Ø 45 5Ø B8 3A ØØ 5Ø 58 5B 89 Ø7 E8 56 27
[CX5] 1735: E9 C2 FC B8 8C 45 5Ø 5B FF 37 B8 8A ØØ 5Ø E8 7C
[3H5] 1736: 9F 58 21 CØ 75 Ø3 E9 Ø3 ØØ E9 36 ØØ 8C D8 Ø5 7C
[CL5] 1737: 17 5Ø ØE E8 Ø2 ØØ Ø1 28 E8 C5 9F E8 E3 9F E8 51
[GK5] 1738: 2C E8 C8 2C B8 3Ø 45 5Ø B8 4C ØØ 5Ø 58 5B 89 Ø7
[EK5] 1739: E8 13 27 B8 4A 45 5Ø B8 Ø2 ØØ 5Ø 58 5B 89 Ø7 E9
[WS5] 174Ø: 7F FC B8 8C 45 5Ø 5B FF 37 B8 85 ØØ 5Ø E8 2D 9F
[LG5] 1741: 58 21 CØ 75 Ø3 E9 Ø3 ØØ E9 2A ØØ 8C D8 Ø5 7C 17
[KM5] 1742: 5Ø ØE E8 Ø2 ØØ Ø1 28 E8 76 9F E8 94 9F E8 Ø2 2C
[RH5] 1743: E8 79 2C B8 3Ø 45 5Ø B8 5Ø ØØ 5Ø 58 5B 89 Ø7 E8
[7S5] 1744: C4 26 E9 AE FF B8 8C 45 5Ø 5B FF 37 B8 93 ØØ 5Ø
[CM5] 1745: E8 EA 9E 58 21 CØ 75 Ø3 E9 Ø3 ØØ E9 4E ØØ 8C D8
[3H5] 1746: Ø5 7C 17 5Ø ØE E8 Ø2 ØØ Ø1 28 E8 33 9F E8 51 9F
[WR5] 1747: E8 BF 2B E8 2C F6 B8 4A 45 5Ø 5B FF 37 E8 Ø8 ØØ
[9S5] 1748: 1B 6D 2A 6D 7F 89 7F 89 E8 66 A4 B8 3Ø 45 5Ø B8
[XG5] 1749: 16 ØØ 5Ø 58 5B 89 Ø7 E9 ØC ØØ B8 3Ø 45 5Ø B8 46
[PR5] 175Ø: ØØ 5Ø 58 5B 89 Ø7 E8 5D 26 E9 C9 FB B8 8C 45 5Ø
[BL5] 1751: 5B FF 37 B8 94 ØØ 5Ø E8 83 9E 58 21 CØ 75 Ø3 E9
[ØG5] 1752: Ø3 ØØ E9 43 ØØ 8C D8 Ø5 7C 17 5Ø ØE E8 Ø2 ØØ Ø1
[FW5] 1753: 28 E8 CC 9E E8 EA 9E E8 58 2B E8 C5 F5 B8 4A 45
[LF5] 1754: 5Ø 5B FF 37 B8 Ø4 ØØ 5Ø E8 52 9E 58 21 CØ 75 Ø3
[TK5] 1755: E9 Ø3 ØØ E9 F9 1B B8 3Ø 45 5Ø B8 64 ØØ 5Ø 58 5B
[BT5] 1756: 89 Ø7 E8 Ø1 26 E9 EB FE B8 8C 45 5Ø 5B FF 37 B8
[KK5] 1757: 88 ØØ 5Ø E8 27 9E 58 21 CØ 75 Ø3 E9 Ø3 ØØ E9 6D
[DH5] 1758: ØØ B8 44 45 5Ø 5B FF 37 B8 28 ØØ 5Ø E8 ØE 9E 58
[8M5] 1759: 21 CØ 75 Ø3 E9 Ø3 ØØ E9 18 ØØ E8 7C 1D E8 6C 2B
```

```
[8G5] 176Ø: B8 3Ø 45 5Ø B8 69 ØØ 5Ø 58 5B 89 Ø7 E8 B7 25 E9
[XM5] 1761: A1 FE B8 Ø8 45 5Ø B8 ØØ 81 5Ø 58 5B 89 Ø7 E8 4D
[ZA5] 1762: 25 B8 Ø8 45 5Ø B8 ØØ ØØ 5Ø 58 5B 89 Ø7 E8 3E 25
[HF5] 1763: B8 3Ø 45 5Ø B8 69 ØØ 5Ø 58 5B 89 Ø7 E8 87 25 B8
[DN5] 1764: 4A 45 5Ø B8 Ø2 ØØ 5Ø 58 5B 89 Ø7 E9 F1 ØE B8 8C
[SM5] 1765: 45 5Ø 5B FF 37 B8 91 ØØ 5Ø E8 A1 9D 58 21 CØ 75
[1G5] 1766: Ø3 E9 Ø3 ØØ E9 39 ØØ 8C D8 Ø5 7C 17 5Ø ØE E8 Ø2
[9S5] 1767: ØØ Ø1 28 E8 EA 9D E8 Ø8 9E E8 76 2A E8 C4 2A B8
[XG5] 1768: 22 45 5Ø B8 58 ØØ 5Ø 58 5B 89 Ø7 E8 F3 23 B8 3Ø
[CN5] 1769: 45 5Ø B8 5D ØØ 5Ø 58 5B 89 Ø7 E8 29 25 E9 12 FB
[SN5] 177Ø: B8 8C 45 5Ø 5B FF 37 B8 9C ØØ 5Ø E8 4F 9D 58 21
[6M5] 1771: CØ 75 Ø3 E9 Ø3 ØØ E9 1B ØØ 8C D8 Ø5 7C 17 5Ø ØE
[6N5] 1772: E8 Ø2 ØØ Ø1 28 E8 98 9D E8 B6 9D E8 24 2A E8 72
[9V5] 1773: 2A E9 DE FA B8 8C 45 5Ø 5B FF 37 B8 9F ØØ 5Ø E8
[PK5] 1774: 1B 9D 58 21 CØ 75 Ø3 E9 Ø3 ØØ E9 2A ØØ 8C D8 Ø5
[NM5] 1775: 7C 17 5Ø ØE E8 Ø2 ØØ Ø1 28 E8 64 9D E8 82 9D E8
[AG5] 1776: FØ 29 E8 67 2A B8 3Ø 45 5Ø B8 6F ØØ 5Ø 58 5B 89
[ØX5] 1777: Ø7 E8 B2 24 E9 9C FD B8 8C 45 5Ø 5B FF 37 B8 9A
[BG5] 1778: ØØ 5Ø E8 D8 9C 58 21 CØ 75 Ø3 E9 Ø3 ØØ E9 2A ØØ
[DK5] 1779: 8C D8 Ø5 7C 17 5Ø ØE E8 Ø2 ØØ Ø1 28 E8 21 9D E8
[5P5] 178Ø: 3F 9D E8 AD 29 E8 FB 29 B8 3Ø 45 5Ø B8 7Ø ØØ 5Ø
[9V5] 1781: 58 5B 89 Ø7 E8 6F 24 E9 DB F9 B8 8C 45 5Ø 5B FF
[EE5] 1782: 37 B8 9Ø ØØ 5Ø E8 95 9C 58 21 CØ 75 Ø3 E9 Ø3 ØØ
[EK5] 1783: E9 2A ØØ 8C D8 Ø5 7C 17 5Ø ØE E8 Ø2 ØØ Ø1 28 E8
[CV5] 1784: DE 9C E8 FC 9C E8 6A 29 E8 B8 29 B8 3Ø 45 5Ø B8
[3M5] 1785: 71 ØØ 5Ø 58 5B 89 Ø7 E8 2C 24 E9 15 FA B8 8C 45
[2G5] 1786: 5Ø 5B FF 37 B8 98 ØØ 5Ø E8 52 9C 58 21 CØ 75 Ø3
[GG5] 1787: E9 Ø3 ØØ E9 2A ØØ 8C D8 Ø5 7C 17 5Ø ØE E8 Ø2 ØØ
[8T5] 1788: Ø1 28 E8 9B 9C E8 B9 9C E8 27 29 E8 75 29 B8 3Ø
[RM5] 1789: 45 5Ø B8 84 ØØ 5Ø 58 5B 89 Ø7 E8 E9 23 E9 55 F9
[5M5] 179Ø: B8 8C 45 5Ø 5B FF 37 B8 86 ØØ 5Ø E8 ØF 9C 58 21
[VM5] 1791: CØ 75 Ø3 E9 Ø3 ØØ E9 4E ØØ 8C D8 Ø5 7C 17 5Ø ØE
[GM5] 1792: E8 Ø2 ØØ Ø1 28 E8 58 9C E8 76 9C E8 E4 28 E8 51
[BP5] 1793: F3 B8 4A 45 5Ø 5B FF 37 E8 Ø8 ØØ F6 6F Ø5 7Ø 7F
[SN5] 1794: 89 7F 89 E8 8B A1 B8 3Ø 45 5Ø B8 8A ØØ 5Ø 58 5B
[FF5] 1795: 89 Ø7 E9 ØC ØØ B8 3Ø 45 5Ø B8 8B ØØ 5Ø 58 5B 89
[UT5] 1796: Ø7 E8 82 23 E9 FA F8 B8 8C 45 5Ø 5B FF 37 B8 84
[BG5] 1797: ØØ 5Ø E8 A8 9B 58 21 CØ 75 Ø3 E9 Ø3 ØØ E9 4E ØØ
[ZK5] 1798: 8C D8 Ø5 7C 17 5Ø ØE E8 Ø2 ØØ Ø1 28 E8 F1 9B E8
[WV5] 1799: ØF 9C E8 7D 28 E8 EA F2 B8 4A 45 5Ø 5B FF 37 E8
[YH5] 18ØØ: Ø8 ØØ 5D 7Ø 6C 7Ø 7F 89 7F 89 E8 24 A1 B8 3Ø 45
[4H5] 18Ø1: 5Ø B8 1A ØØ 5Ø 58 5B 89 Ø7 E9 A5 FF B8 3Ø 45 5Ø
[FM5] 18Ø2: B8 59 ØØ 5Ø 58 5B 89 Ø7 E8 1B 23 E9 Ø4 F9 B8 8C
[WL5] 18Ø3: 45 5Ø 5B FF 37 B8 A1 ØØ 5Ø E8 41 9B 58 21 CØ 75
[GG5] 18Ø4: Ø3 E9 Ø3 ØØ E9 2A ØØ 8C D8 Ø5 7C 17 5Ø ØE E8 Ø2
[3P5] 18Ø5: ØØ Ø1 28 E8 8A 9B E8 A8 9B E8 16 28 E8 64 28 B8
[LM5] 18Ø6: 3Ø 45 5Ø B8 8F ØØ 5Ø 58 5B 89 Ø7 E8 D8 22 E9 C1
[5N5] 18Ø7: F8 B8 8C 45 5Ø 5B FF 37 B8 A2 ØØ 5Ø E8 FE 9A 58
[HF5] 18Ø8: 21 CØ 75 Ø3 E9 Ø3 ØØ E9 2A ØØ 8C D8 Ø5 7C 17 5Ø
[PM5] 18Ø9: ØE E8 Ø2 ØØ Ø1 28 E8 47 9B E8 65 9B E8 D3 27 E8
[NG5] 181Ø: 21 28 B8 3Ø 45 5Ø B8 9Ø ØØ 5Ø 58 5B 89 Ø7 E8 95
[PP5] 1811: 22 E9 7E F8 B8 8C 45 5Ø 5B FF 37 B8 8B ØØ 5Ø E8
[KL5] 1812: BB 9A 58 21 CØ 75 Ø3 E9 Ø3 ØØ E9 2A ØØ 8C D8 Ø5
[DL5] 1813: 7C 17 5Ø ØE E8 Ø2 ØØ Ø1 28 E8 Ø4 9B E8 22 9B E8
[PD5] 1814: 9Ø 27 E8 Ø7 28 B8 3Ø 45 5Ø B8 91 ØØ 5Ø 58 5B 89
[HT5] 1815: Ø7 E8 52 22 E9 3C FB B8 8C 45 5Ø 5B FF 37 B8 87
[YE5] 1816: ØØ 5Ø E8 78 9A 58 21 CØ 75 Ø3 E9 Ø3 ØØ E9 2A ØØ
[VK5] 1817: 8C D8 Ø5 7C 17 5Ø ØE E8 Ø2 ØØ Ø1 28 E8 C1 9A E8
[SM5] 1818: DF 9A E8 4D 27 E8 C4 27 B8 3Ø 45 5Ø B8 92 ØØ 5Ø
[UU5] 1819: 58 5B 89 Ø7 E8 ØF 22 E9 F9 FA B8 8C 45 5Ø 5B FF
[LD5] 182Ø: 37 B8 89 ØØ 5Ø E8 35 9A 58 21 CØ 75 Ø3 E9 Ø3 ØØ
[EK5] 1821: E9 2A ØØ 8C D8 Ø5 7C 17 5Ø ØE E8 Ø2 ØØ Ø1 28 E8
[7R5] 1822: 7E 9A E8 9C 9A E8 ØA 27 E8 81 27 B8 3Ø 45 5Ø B8
[6R5] 1823: 95 ØØ 5Ø 58 5B 89 Ø7 E8 CC 21 E9 B6 FA B8 8C 45
[YG5] 1824: 5Ø 5B FF 37 B8 8C ØØ 5Ø E8 F2 99 58 21 CØ 75 Ø3
[GG5] 1825: E9 Ø3 ØØ E9 2A ØØ 8C D8 Ø5 7C 17 5Ø ØE E8 Ø2 ØØ
[6S5] 1826: Ø1 28 E8 3B 9A E8 59 9A E8 C7 26 E8 3E 27 B8 3Ø
[VM5] 1827: 45 5Ø B8 96 ØØ 5Ø 58 5B 89 Ø7 E8 89 21 E9 73 FA
[XN5] 1828: B8 8C 45 5Ø 5B FF 37 B8 8D ØØ 5Ø E8 AF 99 58 21
[5M5] 1829: CØ 75 Ø3 E9 Ø3 ØØ E9 2A ØØ 8C D8 Ø5 7C 17 5Ø ØE
[XR5] 183Ø: E8 Ø2 ØØ Ø1 28 E8 F8 99 E8 16 9A E8 84 26 E8 FB
[PB5] 1831: 26 B8 3Ø 45 5Ø B8 97 ØØ 5Ø 58 5B 89 Ø7 E8 46 21
[3X5] 1832: E9 3Ø FA B8 8C 45 5Ø 5B FF 37 B8 8E ØØ 5Ø E8 6C
[3G5] 1833: 99 58 21 CØ 75 Ø3 E9 Ø3 ØØ E9 2A ØØ 8C D8 Ø5 7C
[ZH5] 1834: 17 5Ø ØE E8 Ø2 ØØ Ø1 28 E8 B5 99 E8 D3 99 E8 41
[ØF5] 1835: 26 E8 B8 26 B8 3Ø 45 5Ø B8 98 ØØ 5Ø 58 5B 89 Ø7
[ST5] 1836: E8 Ø3 21 E9 ED F9 B8 8C 45 5Ø 5B FF 37 B8 A5 ØØ
[ØF5] 1837: 5Ø E8 29 99 58 21 CØ 75 Ø3 E9 Ø3 ØØ E9 43 ØØ 8C
[7H5] 1838: D8 Ø5 7C 17 5Ø ØE E8 Ø2 ØØ Ø1 28 E8 72 99 E8 9Ø
[GM5] 1839: 99 E8 FE 25 B8 44 45 5Ø 5B FF 37 B8 23 ØØ 5Ø E8
[RG5] 184Ø: C2 98 58 21 CØ 75 Ø3 E9 Ø3 ØØ E8 6C 18 E8 33 26
[5G5] 1841: B8 3Ø 45 5Ø B8 99 ØØ 5Ø 58 5B 89 Ø7 E8 A7 2Ø E9
[AF5] 1842: 91 F9 B8 ØØ ØØ 5Ø ØE E8 14 ØØ 13 55 6E 62 65 6B
[DP5] 1843: 61 6E 6E 74 65 2Ø 46 75 6E 6B 74 69 6F 6E E8 1F
[ML5] 1844: 99 E8 74 99 58 E9 94 16 B8 44 45 5Ø 5B FF 37 B8
[XH5] 1845: DE ØØ 5Ø E8 A7 98 58 21 CØ 75 Ø3 E9 Ø3 ØØ E9 21
[1C5] 1846: ØØ E8 15 18 B8 3Ø 45 5Ø B8 28 ØØ 5Ø 58 5B 89 Ø7
[HK5] 1847: E8 53 2Ø B8 4A 45 5Ø B8 Ø4 ØØ 5Ø 58 5B 89 Ø7 E9
[KM5] 1848: BD Ø9 B8 44 45 5Ø 5B FF 37 B8 85 ØØ 5Ø E8 6D 98
[8L5] 1849: 58 21 CØ 75 Ø3 E9 Ø3 ØØ E9 89 ØØ E8 DB 17 8C D8
[2H5] 185Ø: Ø5 7C 17 5Ø ØE E8 Ø2 ØØ Ø1 24 E8 B3 ØØ E8 D1 98
[3L5] 1851: E8 3F 25 8C D8 Ø5 7C 17 5Ø ØE E8 Ø2 ØØ Ø1 28 E8
[LR5] 1852: 9E 98 E8 BC 98 E8 2A 25 E8 78 25 B8 44 45 5Ø 5B
[4M5] 1853: FF 37 B8 2C ØØ 5Ø E8 24 98 58 21 CØ 75 Ø3 E9 12
[XC5] 1854: ØØ B8 Ø8 45 5Ø B8 ØØ ØØ 5Ø 58 5B 89 Ø7 E8 7E 1F
[9L5] 1855: E9 1F ØØ E8 83 17 B8 44 45 5Ø 5B FF 37 B8 23 ØØ
[MH5] 1856: 5Ø E8 CØ 97 58 21 CØ 75 Ø3 E9 Ø3 ØØ E8 6A 17 E8
[CL5] 1857: 31 25 B8 3Ø 45 5Ø B8 2F ØØ 5Ø 58 5B 89 Ø7 E8 A5
[GN5] 1858: 1F E9 11 F5 B8 44 45 5Ø 5B FF 37 B8 D8 ØØ 5Ø E8
[CF5] 1859: CB 97 58 21 CØ 75 Ø3 E9 Ø3 ØØ E9 Ø5 Ø1 E8 39 17
[UG5] 186Ø: B8 Ø8 45 5Ø B8 Ø1 ØØ 5Ø 58 5B 89 Ø7 E8 1F 1F 8C
[2G5] 1861: D8 Ø5 7C 17 5Ø ØE E8 Ø2 ØØ Ø1 28 E8 Ø2 98 E8 2Ø
[GT5] 1862: 98 E8 8E 24 E8 FB EE B8 8E 45 5Ø B8 4A 45 5Ø 5B
[YM5] 1863: FF 37 B8 Ø4 ØØ 5Ø E8 4B 97 58 5B 89 Ø7 B8 8E 45
[LG5] 1864: 5Ø 5B FF 37 58 21 CØ 75 Ø3 E9 Ø3 ØØ E9 4Ø ØØ B8
[7M5] 1865: 4C 45 5Ø B8 Ø1 ØØ 5Ø 58 5B 89 Ø7 E8 9E 23 8C D8
[TH5] 1866: Ø5 7C 17 5Ø ØE E8 Ø2 ØØ Ø1 2C E8 B3 97 E8 D1 97
[LS5] 1867: E8 3F 24 E8 AC EE B8 4A 45 5Ø 5B FF 37 B8 Ø4 ØØ
[FF5] 1868: 5Ø E8 39 97 58 21 CØ 75 Ø3 E9 Ø3 ØØ E9 EØ 14 8C
[8K5] 1869: D8 Ø5 7C 17 5Ø ØE E8 Ø2 ØØ Ø1 2C E8 82 97 E8 AØ
[PS5] 187Ø: 97 E8 ØE 24 E8 7B EE B8 4A 45 5Ø 5B FF 37 B8 Ø4
```

```
[ZF5] 1871: ØØ 5Ø E8 Ø8 97 58 21 CØ 75 Ø3 E9 Ø3 ØØ E9 AF 14
[MF5] 1872: B8 3Ø 45 5Ø B8 31 ØØ 5Ø 58 5B 89 Ø7 E8 B7 1E B8
[FL5] 1873: 8E 45 5Ø 5B FF 37 58 21 CØ 75 Ø3 E9 Ø3 ØØ E9 91
[VK5] 1874: F4 B8 Ø8 45 5Ø B8 58 5B 5Ø 58 5B 89 Ø7 E8 CØ 1D
[BF5] 1875: B8 22 45 5Ø B8 5Ø ØØ 5Ø 58 5B 89 Ø7 E8 42 1D E9
[WM5] 1876: 7Ø F4 B8 44 45 5Ø 5B FF 37 B8 D1 ØØ 5Ø E8 AD 96
[UG5] 1877: 58 21 CØ 75 Ø3 E9 Ø3 ØØ E9 95 Ø3 E8 1B 16 E8 32
[585] 1878: 21 B8 3Ø ØØ 5Ø B8 ØØ ØØ 5Ø 58 5B 89 Ø7 B8 3Ø ØØ
[VL5] 1879: 5Ø 5B FF 37 B8 6C 45 5Ø 5B FF 37 E8 46 96 58 21
[2D5] 188Ø: CØ 75 Ø3 E9 28 ØØ B8 ØØ ØØ 5Ø ØE E8 16 ØØ 15 55
[PG5] 1881: 6E 64 65 66 69 6E 69 65 72 74 65 2Ø 46 75 6E 6B
[5U5] 1882: 74 69 6F 6E E8 B9 96 E8 ØE 97 58 E9 2E 14 8C D8
[5H5] 1883: Ø5 8C 17 5Ø B8 ØØ ØØ 5Ø E8 A5 96 8C D8 Ø5 FC 1Ø
[7G5] 1884: 5Ø B8 1Ø ØØ 5Ø B8 3Ø ØØ 5Ø 5B FF 37 E8 6D 96 E8
[7L5] 1885: 3E 96 B8 ØØ ØØ 5Ø E8 87 96 E8 6A 98 58 21 CØ 75
[4D5] 1886: Ø3 E9 19 ØØ B8 3Ø ØØ 5Ø B8 3Ø ØØ 5Ø 5B FF 37 B8
[3P5] 1887: Ø1 ØØ 5Ø E8 1A 96 58 5B 89 Ø7 E9 7Ø FF B8 C8 36
[RH5] 1888: 5Ø B8 Ø2 ØØ 5Ø B8 82 45 5Ø 5B FF 37 E8 2D 96 E8
[7M5] 1889: FE 95 B8 3Ø ØØ 5Ø 5B FF 37 58 5B 89 Ø7 B8 82 45
[KG5] 189Ø: 5Ø B8 82 45 5Ø 5B FF 37 B8 Ø1 ØØ 5Ø E8 E1 95 58
[DK5] 1891: 5B 89 Ø7 B8 C8 36 5Ø B8 Ø2 ØØ 5Ø B8 82 45 5Ø 5B
[XN5] 1892: FF 37 E8 F7 95 E8 C8 95 B8 ØØ ØØ 5Ø 58 5B 89 Ø7
[TG5] 1893: B8 82 45 5Ø B8 82 45 5Ø 5B FF 37 B8 Ø1 ØØ 5Ø E8
[AM5] 1894: AE 95 58 5B 89 Ø7 B8 2A 45 5Ø B8 Ø3 ØØ 5Ø 58 5B
[8M5] 1895: 89 Ø7 B8 C8 36 5Ø B8 Ø2 ØØ 5Ø B8 82 45 5Ø 5B FF
[2M5] 1896: 37 E8 B8 95 E8 89 95 B8 44 42 5Ø B8 64 ØØ 5Ø B8
[FK5] 1897: 3Ø ØØ 5Ø 5B FF 37 E8 A3 95 E8 74 95 B8 Ø2 ØØ 5Ø
[UP5] 1898: B8 2A 45 5Ø 5B FF 37 E8 92 95 E8 63 95 5B FF 37
[BD5] 1899: 58 5B 89 Ø7 B8 C8 36 5Ø B8 Ø2 ØØ 5Ø B8 82 45 5Ø
[3S5] 19ØØ: 5B FF 37 E8 76 95 E8 47 95 5B FF 37 58 21 CØ 75
[XE5] 19Ø1: Ø3 E9 16 ØØ B8 82 45 5Ø B8 82 45 5Ø 5B FF 37 B8
[WM5] 19Ø2: Ø1 ØØ 5Ø E8 2A 95 58 5B 89 Ø7 B8 2A 45 5Ø B8 2A
[2G5] 19Ø3: 45 5Ø 5B FF 37 B8 Ø1 ØØ 5Ø E8 1F 95 58 5B 89 Ø7
[UH5] 19Ø4: B8 2A 45 5Ø 5B FF 37 B8 ØØ ØØ 5Ø E8 C9 94 58 21
[ØN5] 19Ø5: CØ 75 Ø3 E9 Ø3 ØØ E9 59 FF B8 44 45 5Ø 5B FF 37
[BF5] 19Ø6: B8 28 ØØ 5Ø E8 D6 94 58 21 CØ 75 Ø3 E9 Ø3 ØØ E9
[4K5] 19Ø7: D7 ØØ E8 44 14 E8 2A EC B8 82 45 5Ø B8 82 45 5Ø
[HP5] 19Ø8: 5B FF 37 B8 Ø1 ØØ 5Ø E8 D1 94 58 5B 89 Ø7 B8 4C
[BL5] 19Ø9: 45 5Ø B8 C8 36 5Ø B8 Ø2 ØØ 5Ø B8 82 45 5Ø 5B FF
[UU5] 191Ø: 37 E8 D8 94 E8 A9 94 5B FF 37 58 5B 89 Ø7 B8 4C
[6G5] 1911: 45 5Ø 5B FF 37 B8 ØØ ØØ 5Ø E8 48 94 58 21 CØ 75
[995] 1912: Ø3 E9 25 ØØ B8 ØØ ØØ 5Ø ØE E8 13 ØØ 12 5A 75 2Ø
[FF5] 1913: 76 69 65 6C 65 2Ø 5Ø 61 72 61 6D 65 74 65 72 E8
[AH5] 1914: BE 94 E8 13 95 58 E9 33 12 E8 9Ø 2Ø B8 44 45 5Ø
[4L5] 1915: 5B FF 37 B8 29 ØØ 5Ø E8 ØA 94 58 21 CØ 75 Ø3 E9
[5F5] 1916: Ø3 ØØ E9 41 ØØ B8 44 45 5Ø 5B FF 37 B8 2C ØØ 5Ø
[TF5] 1917: E8 2A 94 58 21 CØ 75 Ø3 E9 28 ØØ B8 ØØ ØØ 5Ø ØE
[TB5] 1918: E8 16 ØØ 15 27 2C 27 2Ø 6F 64 65 72 2Ø 27 29 27
[6F5] 1919: 2Ø 65 72 77 61 72 74 65 74 E8 64 94 E8 B9 94 58
[ØM5] 192Ø: E9 D9 11 E9 2C FF E8 7Ø 13 B8 82 45 5Ø B8 82 45
[3K5] 1921: 5Ø 5B FF 37 B8 Ø1 ØØ 5Ø E8 ØØ 94 58 5B 89 Ø7 B8
[TN5] 1922: C8 36 5Ø B8 Ø2 ØØ 5Ø B8 82 45 5Ø 5B FF 37 E8 ØB
[1L5] 1923: 94 E8 DC 93 5B FF 37 58 21 CØ 75 Ø3 E9 26 ØØ B8
[3E5] 1924: ØØ ØØ 5Ø ØE E8 14 ØØ 13 5A 75 2Ø 77 65 6E 69 67
[UE5] 1925: 65 2Ø 5Ø 61 72 61 6D 65 74 65 72 E8 Ø2 94 E8 57
[YH5] 1926: 94 58 E9 77 11 B8 82 45 5Ø B8 82 45 5Ø 5B FF 37
[5G5] 1927: B8 Ø1 ØØ 5Ø E8 A4 93 58 5B 89 Ø7 B8 3Ø ØØ 5Ø B8
[CR5] 1928: C8 36 5Ø B8 Ø2 ØØ 5Ø B8 82 45 5Ø 5B FF 37 E8 AB
[LL5] 1929: 93 E8 7C 93 5B FF 37 58 5B 89 Ø7 B8 22 45 5Ø B8
[XM5] 193Ø: E8 ØØ 5Ø 58 5B 89 Ø7 E8 D7 19 B8 Ø8 45 5Ø B8 D4
[7G5] 1931: 43 5Ø B8 Ø2 ØØ 5Ø B8 3Ø ØØ 5Ø 5B FF 37 E8 7C 93
[MR5] 1932: E8 4D 93 5B FF 37 58 5B 89 Ø7 E8 B1 2Ø B8 4A 45
[ØG5] 1933: 5Ø B8 EC 3F 5Ø B8 Ø2 ØØ 5Ø B8 3Ø ØØ 5Ø 5B FF 37
[9P5] 1934: E8 59 93 E8 2A 93 5B FF 37 58 5B 89 Ø7 E9 4F Ø4
[5M5] 1935: B8 44 45 5Ø 5B FF 37 B8 D6 ØØ 5Ø E8 FF 92 58 21
[7K5] 1936: CØ 75 Ø3 E9 Ø3 ØØ E9 78 ØØ E8 6D 12 8C D8 Ø5 7C
[LH5] 1937: 17 5Ø ØE E8 Ø2 ØØ Ø1 28 E8 45 93 E8 63 93 E8 D1
[5H5] 1938: 1F E8 1F 2Ø 8C D8 Ø5 7C 17 5Ø ØE E8 Ø2 ØØ Ø1 2C
[NU5] 1939: E8 2D 93 E8 4B 93 E8 B9 1F E8 26 EA B8 4A 45 5Ø
[EM5] 194Ø: 5B FF 37 E8 Ø8 ØØ 3Ø 79 21 79 7F 89 3F 79 E8 6Ø
[TD5] 1941: 98 B8 3Ø 45 5Ø B8 51 ØØ 5Ø 58 5B 89 Ø7 E8 66 1A
[HF5] 1942: B8 3Ø 45 5Ø B8 19 ØØ 5Ø 58 5B 89 Ø7 E8 57 1A B8
[3M5] 1943: 3Ø 45 5Ø B8 3B ØØ 5Ø 58 5B 89 Ø7 E8 48 1A E9 B4
[GN5] 1944: EF B8 44 45 5Ø 5B FF 37 B8 DA ØØ 5Ø E8 6E 92 58
[TM5] 1945: 21 CØ 75 Ø3 E9 Ø3 ØØ E9 1E ØØ E8 DC 11 8C D8 Ø5
[RL5] 1946: 7C 17 5Ø ØE E8 Ø2 ØØ Ø1 28 E8 B4 92 E8 D2 92 E8
[NU5] 1947: 4Ø 1F E8 6A 24 E9 FA EF B8 44 45 5Ø 5B FF 37 B8
[5N5] 1948: FE ØØ 5Ø E8 37 92 58 21 CØ 75 Ø3 E9 Ø3 ØØ E9 FE
[VE5] 1949: ØØ E8 29 11 B8 8C 45 5Ø B8 44 45 5Ø 5B FF 37 58
[ØP5] 195Ø: 5B 89 Ø7 E8 93 11 B8 8C 45 5Ø 5B FF 37 B8 8E ØØ
[FE5] 1951: 5Ø E8 Ø9 92 58 21 CØ 75 Ø3 E9 Ø3 ØØ E9 1E ØØ B8
[SM5] 1952: 3Ø 45 5Ø B8 3F ØØ 5Ø 58 5B 89 Ø7 E8 B8 19 B8 4A
[PF5] 1953: 45 5Ø B8 Ø4 ØØ 5Ø 58 5B 89 Ø7 E9 22 Ø3 B8 8C 45
[PG5] 1954: 5Ø 5B FF 37 B8 8D ØØ 5Ø E8 D2 91 58 21 CØ 75 Ø3
[DH5] 1955: E9 Ø3 ØØ E9 12 ØØ B8 3Ø 45 5Ø B8 6B ØØ 5Ø 58 5B
[UT5] 1956: 89 Ø7 E8 81 19 E9 C6 FF B8 8C 45 5Ø 5B FF 37 B8
[LH5] 1957: 94 ØØ 5Ø E8 A7 91 58 21 CØ 75 Ø3 E9 Ø3 ØØ E9 1E
[1C5] 1958: ØØ B8 3Ø 45 5Ø B8 6C ØØ 5Ø 58 5B 89 Ø7 E8 56 19
[NF5] 1959: B8 4A 45 5Ø B8 Ø2 ØØ 5Ø 58 5B 89 Ø7 E9 CØ Ø2 B8
[8G5] 196Ø: ØØ ØØ 5Ø ØE E8 14 ØØ 13 55 6E 62 65 6B 61 6E 6E
[9L5] 1961: 74 65 2Ø 46 75 6E 6B 74 69 6F 6E E8 C2 91 E8 17
[LL5] 1962: 92 58 E9 37 ØF 8C D8 Ø5 7C 17 5Ø ØE E8 Ø2 ØØ Ø1
[PR5] 1963: 28 E8 AC 91 E8 CA 91 E8 38 1E E8 86 1E B8 3Ø 45
[HM5] 1964: 5Ø B8 94 ØØ 5Ø 58 5B 89 Ø7 E8 FA 18 E9 66 EE B8
[1N5] 1965: 44 45 5Ø 5B FF 37 B8 FD ØØ 5Ø E8 2Ø 91 58 21 CØ
[3H5] 1966: 75 Ø3 E9 Ø3 ØØ E9 6B Ø1 E8 12 1Ø B8 8C 45 5Ø B8
[CP5] 1967: 44 45 5Ø 5B FF 37 58 5B 89 Ø7 E8 7C 1Ø B8 8C 45
[RF5] 1968: 5Ø 5B FF 37 B8 84 ØØ 5Ø E8 F2 9Ø 58 21 CØ 75 Ø3
[GG5] 1969: E9 Ø3 ØØ E9 2A ØØ 8C D8 Ø5 7C 17 5Ø ØE E8 Ø2 ØØ
[7M5] 197Ø: Ø1 28 E8 3B 91 E8 59 91 E8 C7 1D E8 15 1E B8 3Ø
[GR5] 1971: 45 5Ø B8 67 ØØ 5Ø 58 5B 89 Ø7 E8 89 18 E9 F5 ED
[YL5] 1972: B8 8C 45 5Ø 5B FF 37 B8 85 ØØ 5Ø E8 AF 9Ø 58 21
[5M5] 1973: CØ 75 Ø3 E9 Ø3 ØØ E9 2A ØØ 8C D8 Ø5 7C 17 5Ø ØE
[4R5] 1974: E8 Ø2 ØØ Ø1 28 E8 F8 9Ø E8 16 91 E8 84 1D E8 FB
[CD5] 1975: 1D B8 3Ø 45 5Ø B8 68 ØØ 5Ø 58 5B 89 Ø7 E8 46 18
[5V5] 1976: E9 B2 ED B8 8C 45 5Ø 5B FF 37 B8 81 ØØ 5Ø E8 6C
[2F5] 1977: 9Ø 58 21 CØ 75 Ø3 E9 Ø3 ØØ E9 43 ØØ 8C D8 Ø5 7C
```

Listing 1. »sbasic.exe« ist der Hexdump des Basic-Compilers. Geben Sie dieses Listing nur dann ein, wenn Sie keine frühere Version von SBasic besitzen.

```
[4H5] 1978: 17 5Ø ØE E8 Ø2 ØØ Ø1 28 E8 B5 9Ø E8 D3 9Ø E8 41
[7R5] 1979: 1D E8 AE E7 B8 4A 45 5Ø 5B FF 37 B8 Ø4 ØØ 5Ø E8
[ZG5] 198Ø: 3B 9Ø 58 21 CØ 75 Ø3 E9 Ø3 ØØ E9 E2 ØD B8 3Ø 45
[MM5] 1981: 5Ø B8 6D ØØ 5Ø 58 5B 89 Ø7 E8 EA 17 E9 D3 ED B8
[YK5] 1982: 8C 45 5Ø 5B FF 37 B8 82 ØØ 5Ø E8 1Ø 9Ø 58 21 CØ
[9G5] 1983: 75 Ø3 E9 Ø3 ØØ E9 43 ØØ 8C D8 Ø5 7C 17 5Ø ØE E8
[NM5] 1984: Ø2 ØØ Ø1 28 E8 59 9Ø E8 77 9Ø E8 E5 1C E8 52 E7
[MM5] 1985: B8 4A 45 5Ø 5B FF 37 B8 Ø4 ØØ 5Ø E8 DF 8F 58 21
[PF5] 1986: CØ 75 Ø3 E9 Ø3 ØØ E9 86 ØD B8 3Ø 45 5Ø B8 6E ØØ
[3L5] 1987: 5Ø 58 5B 89 Ø7 E8 8E 17 E9 78 FØ B8 ØØ ØØ 5Ø B8
[2S5] 1988: 8C 45 5Ø 5B FF 37 E8 8F 9Ø FE 46 ØØ E8 69 9Ø 58
[5T5] 1989: E9 87 1C B8 44 45 5Ø 5B FF 37 B8 C8 ØØ 5Ø E8 9C
[TH5] 199Ø: 8F 58 21 CØ 75 Ø3 E9 Ø3 ØØ E9 73 ØØ E8 ØA ØF 8C
[1K5] 1991: D8 Ø5 7C 17 5Ø ØE E8 Ø2 ØØ Ø1 28 E8 E2 8F E8 ØØ
[SN5] 1992: 9Ø E8 6E 1C E8 BC 1C 8C D8 Ø5 7C 17 5Ø ØE E8 Ø2
[CU5] 1993: ØØ Ø1 2C E8 CA 8F E8 E8 8F E8 56 1C E8 A4 1C B8
[RN5] 1994: 44 45 5Ø 5B FF 37 B8 2C ØØ 5Ø E8 17 8F 58 21 CØ
[GP5] 1995: 75 Ø3 E9 Ø9 ØØ E8 C1 ØE E8 88 1C E9 ØF ØØ B8 Ø8
[KF5] 1996: 45 5Ø B8 ØØ ØØ 5Ø 58 5B 89 Ø7 E8 A1 16 B8 3Ø 45
[CM5] 1997: 5Ø B8 6A ØØ 5Ø 58 5B 89 Ø7 E8 EA 16 E9 D3 EC B8
[DP5] 1998: 44 45 5Ø 5B FF 37 B8 DB ØØ 5Ø E8 1Ø 8F 58 21 CØ
[SF5] 1999: 75 Ø3 E9 Ø3 ØØ E9 21 ØØ E8 7E ØE B8 3Ø 45 5Ø B8
[ØN5] 2ØØØ: 5C ØØ 5Ø 58 5B 89 Ø7 E8 BC 16 B8 4A 45 5Ø B8 Ø1
[HE5] 2ØØ1: ØØ 5Ø 58 5B 89 Ø7 E9 26 ØØ B8 ØØ ØØ 5Ø ØE E8 14
[4H5] 2ØØ2: ØØ 13 55 6E 62 65 6B 61 6E 6E 74 65 2Ø 46 75 6E
[1V5] 2ØØ3: 6B 74 69 6F 6E E8 28 8F E8 7D 8F 58 E9 9D ØC B8
[AP5] 2ØØ4: 9Ø 37 5Ø B8 Ø2 ØØ 5Ø B8 9Ø 45 5Ø 5B FF 37 E8 EB
[WS5] 2ØØ5: 8E E8 BC 8E B8 4A 45 5Ø 5B FF 37 58 5B 89 Ø7 B8
[WM5] 2ØØ6: 9Ø 45 5Ø B8 9Ø 45 5Ø 5B FF 37 B8 Ø1 ØØ 5Ø E8 9F
[GE5] 2ØØ7: 8E 58 5B 89 Ø7 B8 92 45 5Ø B8 ØØ ØØ 5Ø 58 5B 89
[GH5] 2ØØ8: Ø7 B8 44 45 5Ø 5B FF 37 B8 E6 ØØ 5Ø E8 35 8E 58
[AG5] 2ØØ9: 21 CØ 75 Ø3 E9 1C ØØ B8 92 45 5Ø B8 92 45 5Ø 5B
[RS5] 201Ø: FF 37 B8 29 ØØ 5Ø E8 88 8E 58 5B 89 Ø7 E8 C9 ØD
[AU5] 2Ø11: E9 CE FF B8 44 45 5Ø 5B FF 37 B8 E7 ØØ 5Ø E8 Ø3
[FH5] 2Ø12: 8E 58 21 CØ 75 Ø3 E9 1C ØØ B8 92 45 5Ø B8 92 45
[CM5] 2Ø13: 5Ø 5B FF 37 B8 2A ØØ 5Ø E8 56 8E 58 5B 89 Ø7 E8
[DR5] 2Ø14: 97 ØD E9 9C FF B8 44 45 5Ø 5B FF 37 B8 E8 ØØ 5Ø
[MH5] 2Ø15: E8 D1 8D 58 21 CØ 75 Ø3 E9 1C ØØ B8 92 45 5Ø B8
[FP5] 2Ø16: 92 45 5Ø 5B FF 37 B8 2C ØØ 5Ø E8 24 8E 58 5B 89
[ER5] 2Ø17: Ø7 E8 65 ØD E9 6A FF B8 92 45 5Ø 5B FF 37 58 21
[3L5] 2Ø18: CØ 75 Ø3 E9 Ø3 ØØ E9 7C Ø1 B8 44 45 5Ø 5B FF 37
[FH5] 2Ø19: B8 E9 ØØ 5Ø E8 8D 8D 58 21 CØ 75 Ø3 E9 ØC ØØ B8
[VF5] 2Ø2Ø: 92 45 5Ø B8 2Ø ØØ 5Ø 58 5B 89 Ø7 B8 44 45 5Ø 5B
[WU5] 2Ø21: FF 37 B8 EA ØØ 5Ø E8 6B 8D 58 21 CØ 75 Ø3 E9 ØC
[9B5] 2Ø22: ØØ B8 92 45 5Ø B8 21 ØØ 5Ø 58 5B 89 Ø7 B8 44 45
[CH5] 2Ø23: 5Ø 5B FF 37 B8 EB ØØ 5Ø E8 49 8D 58 21 CØ 75 Ø3
[5K5] 2Ø24: E9 ØC ØØ B8 92 45 5Ø B8 18 ØØ 5Ø 58 5B 89 Ø7 B8
[MN5] 2Ø25: 44 45 5Ø 5B FF 37 B8 F4 ØØ 5Ø E8 27 8D 58 21 CØ
[EE5] 2Ø26: 75 Ø3 E9 ØC ØØ B8 92 45 5Ø B8 19 ØØ 5Ø 58 5B 89
[7H5] 2Ø27: Ø7 B8 44 45 5Ø 5B FF 37 B8 F3 ØØ 5Ø E8 Ø5 8D 58
[GF5] 2Ø28: 21 CØ 75 Ø3 E9 ØC ØØ B8 92 45 5Ø B8 1A ØØ 5Ø 58
[AT5] 2Ø29: 5B 89 Ø7 B8 44 45 5Ø 5B FF 37 B8 EC ØØ 5Ø E8 E3
[PF5] 2Ø3Ø: 8C 58 21 CØ 75 Ø3 E9 ØC ØØ B8 92 45 5Ø B8 1B ØØ
[WG5] 2Ø31: 5Ø 58 5B 89 Ø7 B8 44 45 5Ø 5B FF 37 B8 EE ØØ 5Ø
[HH5] 2Ø32: E8 C1 8C 58 21 CØ 75 Ø3 E9 ØC ØØ B8 92 45 5Ø B8
[LP5] 2Ø33: 3Ø ØØ 5Ø 58 5B 89 Ø7 B8 44 45 5Ø 5B FF 37 B8 EF
[8M5] 2Ø34: ØØ 5Ø E8 9F 8C 58 21 CØ 75 Ø3 E9 ØC ØØ B8 92 45
[BE5] 2Ø35: 5Ø B8 31 ØØ 5Ø 58 5B 89 Ø7 B8 44 45 5Ø 5B FF 37
[WG5] 2Ø36: B8 FØ ØØ 5Ø E8 7D 8C 58 21 CØ 75 Ø3 E9 ØC ØØ B8
[3G5] 2Ø37: 92 45 5Ø B8 32 ØØ 5Ø 58 5B 89 Ø7 B8 44 45 5Ø 5B
[2V5] 2Ø38: FF 37 B8 ED ØØ 5Ø E8 5B 8C 58 21 CØ 75 Ø3 E9 ØC
[PB5] 2Ø39: ØØ B8 92 45 5Ø B8 1Ø ØØ 5Ø 58 5B 89 Ø7 B8 92 45
[HF5] 2Ø4Ø: 5Ø 5B FF 37 B8 ØØ ØØ 5Ø E8 39 8C 58 21 CØ 75 Ø3
[AL5] 2Ø41: E9 ØF ØØ B8 92 45 5Ø B8 FØ ØØ 5Ø 58 5B 89 Ø7 E9
[WH5] 2Ø42: Ø3 ØØ E8 D4 ØB B8 82 45 5Ø 5B FF 37 B8 ØØ ØØ 5Ø
[GK5] 2Ø43: E8 11 8C 58 21 CØ 75 Ø3 E9 Ø3 ØØ E9 55 Ø9 B8 C8
[4F5] 2Ø44: 36 5Ø B8 Ø2 ØØ 5Ø B8 82 45 5Ø 5B FF 37 B8 Ø1 ØØ
[HS5] 2Ø45: 5Ø E8 47 8C E8 65 8C E8 36 8C 5B FF 37 B8 F8 ØØ
[MM5] 2Ø46: 5Ø E8 42 8C B8 92 45 5Ø 5B FF 37 B8 F8 ØØ 5Ø E8
[EK5] 2Ø47: 34 8C E8 A9 8B 58 21 CØ 75 Ø3 E9 Ø3 ØØ E9 13 Ø9
[TG5] 2Ø48: B8 82 45 5Ø B8 82 45 5Ø 5B FF 37 B8 Ø1 ØØ 5Ø E8
[8L5] 2Ø49: Ø9 8C 58 5B 89 Ø7 B8 C8 36 5Ø B8 Ø2 ØØ 5Ø B8 82
[5T5] 2Ø5Ø: 45 5Ø 5B FF 37 E8 14 8C E8 E5 8B 5B FF 37 B8 Ø1
[CG5] 2Ø51: ØØ 5Ø E8 C8 8B 58 21 CØ 75 Ø3 E9 Ø3 ØØ E9 4B ØØ
[AF5] 2Ø52: B8 9Ø 37 5Ø B8 Ø2 ØØ 5Ø B8 9Ø 45 5Ø 5B FF 37 B8
[9V5] 2Ø53: Ø1 ØØ 5Ø E8 C5 8B E8 E3 8B E8 B4 8B 5B FF 37 E8
[CH5] 2Ø54: Ø8 ØØ 3D 8Ø 4C 8Ø 7F 89 7F 89 E8 44 91 B8 3Ø 45
[WC5] 2Ø55: 5Ø B8 11 ØØ 5Ø 58 5B 89 Ø7 E9 FC ØØ B8 3Ø 45 5Ø
[WK5] 2Ø56: B8 47 ØØ 5Ø 58 5B 89 Ø7 E9 ED ØØ B8 C8 36 5Ø B8
[MS5] 2Ø57: Ø2 ØØ 5Ø B8 82 45 5Ø 5B FF 37 E8 9F 8B E8 7Ø 8B
[ØK5] 2Ø58: 5B FF 37 B8 ØØ ØØ 5Ø E8 53 8B 58 21 CØ 75 Ø3 E9
[GD5] 2Ø59: Ø3 ØØ E9 3F ØØ B8 9Ø 37 5Ø B8 Ø2 ØØ 5Ø B8 9Ø 45
[ZR5] 2Ø6Ø: 5Ø 5B FF 37 B8 Ø1 ØØ 5Ø E8 5Ø 8B E8 6E 8B E8 3F
[RS5] 2Ø61: 8B 5B FF 37 E8 Ø8 ØØ B5 8Ø B2 8Ø 7F 89 7F 89 E8
[PE5] 2Ø62: CF 9Ø E8 82 22 B8 3Ø 45 5Ø B8 1Ø ØØ 5Ø 58 5B 89
[MG5] 2Ø63: Ø7 E9 84 ØØ E8 35 14 B8 C8 36 5Ø B8 Ø2 ØØ 5Ø B8
[TS5] 2Ø64: 82 45 5Ø 5B FF 37 E8 33 8B E8 Ø4 8B 5B FF 37 B8
[8K5] 2Ø65: 2Ø ØØ 5Ø E8 E7 8A 58 21 CØ 75 Ø3 E9 Ø3 ØØ E9 5D
[FD5] 2Ø66: ØØ B8 9Ø 37 5Ø B8 Ø2 ØØ 5Ø B8 9Ø 45 5Ø 5B FF 37
[HR5] 2Ø67: B8 Ø1 ØØ 5Ø E8 E4 8A E8 Ø2 8B E8 D3 8A 5B FF 37
[2M5] 2Ø68: E8 Ø8 ØØ 1E 81 2D 81 7F 89 3C 81 E8 63 9Ø B8 3Ø
[LG5] 2Ø69: 45 5Ø B8 Ø8 ØØ 5Ø 58 5B 89 Ø7 E9 1B ØØ B8 3Ø 45
[RA5] 2Ø7Ø: 5Ø B8 48 ØØ 5Ø 58 5B 89 Ø7 E9 ØC ØØ B8 3Ø 45 5Ø
[1N5] 2Ø71: B8 18 ØØ 5Ø 58 5B 89 Ø7 E8 4B 12 E9 27 FE B8 C8
[TM5] 2Ø72: 36 5Ø B8 Ø2 ØØ 5Ø B8 82 45 5Ø 5B FF 37 E8 AC 8A
[SP5] 2Ø73: E8 7D 8A 5B FF 37 B8 18 ØØ 5Ø E8 6Ø 8A 58 21 CØ
[7B5] 2Ø74: 75 Ø3 E9 Ø3 ØØ E9 72 ØØ B8 9Ø 37 5Ø B8 Ø2 ØØ 5Ø
[LP5] 2Ø75: B8 9Ø 45 5Ø 5B FF 37 B8 Ø1 ØØ 5Ø E8 5D 8A E8 7B
[ØR5] 2Ø76: 8A E8 4C 8A 5B FF 37 E8 Ø8 ØØ A5 81 B4 81 7F 89
[FM5] 2Ø77: 7F 89 E8 DC 8F B8 3Ø 45 5Ø B8 ØD ØØ 5Ø 58 5B 89
[2K5] 2Ø78: Ø7 E9 94 FF B8 3Ø 45 5Ø B8 4A ØØ 5Ø 58 5B 89 Ø7
[3M5] 2Ø79: E9 85 FF B8 9Ø 37 5Ø B8 Ø2 ØØ 5Ø B8 9Ø 45 5Ø 5B
[UN5] 2Ø8Ø: FF 37 B8 Ø1 ØØ 5Ø E8 12 8A E8 3Ø 8A E8 Ø1 8A B8
[LM5] 2Ø81: Ø1 ØØ 5Ø 58 5B 89 Ø7 E9 5E FF B8 C8 36 5Ø B8 Ø2
[AN5] 2Ø82: ØØ 5Ø B8 82 45 5Ø 5B FF 37 E8 1Ø 8A E8 E1 89 5B
[TN5] 2Ø83: FF 37 B8 2C ØØ 5Ø E8 C4 89 58 21 CØ 75 Ø3 E9 Ø3
[BA5] 2Ø84: ØØ E9 5A ØØ B8 9Ø 37 5Ø B8 Ø2 ØØ 5Ø B8 9Ø 45 5Ø
[8U5] 2Ø85: 5B FF 37 B8 Ø1 ØØ 5Ø E8 C1 89 E8 DF 89 E8 BØ 89
[1L5] 2Ø86: 5B FF 37 E8 Ø8 ØØ 41 82 5Ø 82 7F 89 5F 82 E8 4Ø
[5H5] 2Ø87: 8F B8 3Ø 45 5Ø B8 Ø3 ØØ 5Ø 58 5B 89 Ø7 E9 73 FF
[ØK5] 2Ø88: B8 3Ø 45 5Ø B8 53 ØØ 5Ø 58 5B 89 Ø7 E9 64 FF B8
```

```
[ØN5] 2Ø89: 3Ø 45 5Ø B8 2A ØØ 5Ø 58 5B 89 Ø7 E9 55 FF B8 C8
[NK5] 2Ø9Ø: 36 5Ø B8 Ø2 ØØ 5Ø B8 82 45 5Ø 5B FF 37 E8 8C 89
[RP5] 2Ø91: E8 5D 89 5B FF 37 B8 2A ØØ 5Ø E8 4Ø 89 58 21 CØ
[EB5] 2Ø92: 75 Ø3 E9 Ø3 ØØ E9 5A ØØ B8 9Ø 37 5Ø B8 Ø2 ØØ 5Ø
[TN5] 2Ø93: B8 9Ø 45 5Ø 5B FF 37 B8 Ø1 ØØ 5Ø E8 3D 89 E8 5B
[KN5] 2Ø94: 89 E8 2C 89 5B FF 37 E8 Ø8 ØØ C5 82 D4 82 7F 89
[BK5] 2Ø95: E3 82 E8 BC 8E B8 3Ø 45 5Ø B8 Ø4 ØØ 5Ø 58 5B 89
[7L5] 2Ø96: Ø7 E9 EF FE B8 3Ø 45 5Ø B8 54 ØØ 5Ø 58 5B 89 Ø7
[1N5] 2Ø97: E9 EØ FE B8 3Ø 45 5Ø B8 29 ØØ 5Ø 58 5B 89 Ø7 E9
[UN5] 2Ø98: D1 FE B8 C8 36 5Ø B8 Ø2 ØØ 5Ø B8 82 45 5Ø 5B FF
[7T5] 2Ø99: 37 E8 Ø8 89 E8 D9 88 5B FF 37 B8 29 ØØ 5Ø E8 BC
[YC5] 21ØØ: 88 58 21 CØ 75 Ø3 E9 Ø3 ØØ E9 5A ØØ B8 9Ø 37 5Ø
[TF5] 21Ø1: B8 Ø2 ØØ 5Ø B8 9Ø 45 5Ø 5B FF 37 B8 Ø1 ØØ 5Ø E8
[5S5] 21Ø2: B9 88 E8 D7 88 E8 A8 88 5B FF 37 E8 Ø8 ØØ 49 83
[XF5] 21Ø3: 58 83 7F 89 67 83 E8 38 8E B8 3Ø 45 5Ø B8 Ø2 ØØ
[MG5] 21Ø4: 5Ø 58 5B 89 Ø7 E9 6B FE B8 3Ø 45 5Ø B8 52 ØØ 5Ø
[YM5] 21Ø5: 58 5B 89 Ø7 E9 5C FE B8 3Ø 45 5Ø B8 2B ØØ 5Ø 58
[VN5] 21Ø6: 5B 89 Ø7 E9 4D FE B8 C8 36 5Ø B8 Ø2 ØØ 5Ø B8 82
[HU5] 21Ø7: 45 5Ø 5B FF 37 E8 84 88 E8 55 88 5B FF 37 B8 2E
[1E5] 21Ø8: ØØ 5Ø E8 38 88 58 21 CØ 75 Ø3 E9 Ø3 ØØ E9 5A ØØ
[AF5] 21Ø9: B8 9Ø 37 5Ø B8 Ø2 ØØ 5Ø B8 9Ø 45 5Ø 5B FF 37 B8
[ZL5] 21ıØ: Ø1 ØØ 5Ø E8 35 88 E8 53 88 E8 24 88 5B FF 37 E8
[3S5] 2111: Ø8 ØØ CD 83 DC 83 7F 89 EB 83 E8 B4 8D B8 3Ø 45
[ZG5] 2112: 5Ø B8 Ø6 ØØ 5Ø 58 5B 89 Ø7 E9 E7 FD B8 3Ø 45 5Ø
[NL5] 2113: B8 56 ØØ 5Ø 58 5B 89 Ø7 E9 D8 FD B8 3Ø 45 5Ø B8
[1P5] 2114: 2C ØØ 5Ø 58 5B 89 Ø7 E9 C9 FD B8 C8 36 5Ø B8 Ø2
[ZL5] 2115: ØØ 5Ø B8 82 45 5Ø 5B FF 37 E8 ØØ 88 E8 D1 87 5B
[CN5] 2116: FF 37 B8 2B ØØ 5Ø E8 B4 87 58 21 CØ 75 Ø3 E9 Ø3
[BA5] 2117: ØØ E9 5A ØØ B8 9Ø 37 5Ø B8 Ø2 ØØ 5Ø B8 9Ø 45 5Ø
[DT5] 2118: 5B FF 37 B8 Ø1 ØØ 5Ø E8 B1 87 E8 CF 87 E8 AØ 87
[PL5] 2119: 5B FF 37 E8 Ø8 ØØ 51 84 6Ø 84 7F 89 6F 84 E8 3Ø
[NG5] 212Ø: 8D B8 3Ø 45 5Ø B8 Ø5 ØØ 5Ø 58 5B 89 Ø7 E9 63 FD
[1K5] 2121: B8 3Ø 45 5Ø B8 55 ØØ 5Ø 58 5B 89 Ø7 E9 54 FD B8
[3N5] 2122: 3Ø 45 5Ø B8 2D ØØ 5Ø 58 5B 89 Ø7 E9 45 FD B8 C8
[9K5] 2123: 36 5Ø B8 Ø2 ØØ 5Ø B8 82 45 5Ø 5B FF 37 E8 7C 87
[RP5] 2124: E8 4D 87 5B FF 37 B8 2D ØØ 5Ø E8 3Ø 87 58 21 CØ
[EB5] 2125: 75 Ø3 E9 Ø3 ØØ E9 5A ØØ B8 9Ø 37 5Ø B8 Ø2 ØØ 5Ø
[GN5] 2126: B8 9Ø 45 5Ø 5B FF 37 B8 Ø1 ØØ 5Ø E8 2D 87 E8 4B
[KN5] 2127: 87 E8 1C 87 5B FF 37 E8 Ø8 ØØ D5 84 E4 84 7F 89
[BK5] 2128: F3 84 E8 AC 8C B8 3Ø 45 5Ø B8 Ø7 ØØ 5Ø 58 5B 89
[1L5] 2129: Ø7 E9 DF FC B8 3Ø 45 5Ø B8 57 ØØ 5Ø 58 5B 89 Ø7
[2P5] 213Ø: E9 DØ FC B8 3Ø 45 5Ø B8 2E ØØ 5Ø 58 5B 89 Ø7 E9
[MN5] 2131: C1 FC B8 C8 36 5Ø B8 Ø2 ØØ 5Ø B8 82 45 5Ø 5B FF
[CU5] 2132: 37 E8 F8 86 E8 C9 86 5B FF 37 B8 3Ø ØØ 5Ø E8 AC
[NC5] 2133: 86 58 21 CØ 75 Ø3 E9 Ø3 ØØ E9 3F ØØ B8 9Ø 37 5Ø
[TF5] 2134: B8 Ø2 ØØ 5Ø B8 9Ø 45 5Ø 5B FF 37 B8 Ø1 ØØ 5Ø E8
[4R5] 2135: A9 86 E8 C7 86 E8 98 86 5B FF 37 E8 Ø8 ØØ 5C 85
[RR5] 2136: 59 85 7F 89 7F 89 E8 28 8C E8 DB 1D B8 3Ø 45 5Ø
[NP5] 2137: B8 ØA ØØ 5Ø 58 5B 89 Ø7 E9 DD FB B8 C8 36 5Ø B8
[TN5] 2138: Ø2 ØØ 5Ø B8 82 45 5Ø 5B FF 37 E8 8F 86 E8 6Ø 86
[FH5] 2139: 5B FF 37 B8 31 ØØ 5Ø E8 43 86 58 21 CØ 75 Ø3 E9
[GD5] 214Ø: Ø3 ØØ E9 3F ØØ B8 9Ø 37 5Ø B8 Ø2 ØØ 5Ø B8 9Ø 45
[NP5] 2141: 5Ø 5B FF 37 B8 Ø1 ØØ 5Ø E8 4Ø 86 E8 5E 86 E8 2F
[BS5] 2142: 86 5B FF 37 E8 Ø8 ØØ C5 85 C2 85 7F 89 7F 89 E8
[PH5] 2143: BF 8B E8 72 1D B8 3Ø 45 5Ø B8 ØB ØØ 5Ø 58 5B 89
[1G5] 2144: Ø7 E9 74 FB B8 C8 36 5Ø B8 Ø2 ØØ 5Ø B8 82 45 5Ø
[5R5] 2145: 5B FF 37 E8 26 86 E8 F7 85 5B FF 37 B8 32 ØØ 5Ø
[MK5] 2146: E8 DA 85 58 21 CØ 75 Ø3 E9 Ø3 ØØ E9 3F ØØ B8 9Ø
[2F5] 2147: 37 5Ø B8 Ø2 ØØ 5Ø B8 9Ø 45 5Ø 5B FF 37 B8 Ø1 ØØ
[AP5] 2148: 5Ø E8 D7 85 E8 F5 85 E8 C6 85 5B FF 37 E8 Ø8 ØØ
[UP5] 2149: 2E 86 2B 86 7F 89 7F 89 E8 56 8B E8 Ø9 1D B8 3Ø
[7R5] 215Ø: 45 5Ø B8 ØC ØØ 5Ø 58 5B 89 Ø7 E9 ØB FB B8 C8 36
[2K5] 2151: 5Ø B8 Ø2 ØØ 5Ø B8 82 45 5Ø 5B FF 37 E8 BD 85 E8
[YK5] 2152: 8E 85 5B FF 37 B8 19 ØØ 5Ø E8 71 85 58 21 CØ 75
[UE5] 2153: Ø3 E9 Ø3 ØØ E9 3F ØØ B8 9Ø 37 5Ø B8 Ø2 ØØ 5Ø B8
[8N5] 2154: 9Ø 45 5Ø 5B FF 37 B8 Ø1 ØØ 5Ø E8 6E 85 E8 8C 85
[TS5] 2155: E8 5D 85 5B FF 37 E8 Ø8 ØØ 97 86 94 86 7F 89 7F
[MN5] 2156: 89 E8 ED 8A E8 AØ 1C B8 3Ø 45 5Ø B8 ØE ØØ 5Ø 58
[RM5] 2157: 5B 89 Ø7 E9 A2 FA B8 C8 36 5Ø B8 Ø2 ØØ 5Ø B8 82
[ØT5] 2158: 45 5Ø 5B FF 37 E8 54 85 E8 25 85 5B FF 37 B8 1A
[CD5] 2159: ØØ 5Ø E8 Ø8 85 58 21 CØ 75 Ø3 E9 Ø3 ØØ E9 3F ØØ
[AF5] 216Ø: B8 9Ø 37 5Ø B8 Ø2 ØØ 5Ø B8 9Ø 45 5Ø 5B FF 37 B8
[1N5] 2161: Ø1 ØØ 5Ø E8 Ø5 85 E8 23 85 E8 F4 84 5B FF 37 E8
[EP5] 2162: Ø8 ØØ ØØ 87 FD 86 7F 89 7F 89 E8 84 8A E8 37 1C
[3L5] 2163: B8 3Ø 45 5Ø B8 ØF ØØ 5Ø 58 5B 89 Ø7 E9 39 FA B8
[SR5] 2164: C8 36 5Ø B8 Ø2 ØØ 5Ø B8 82 45 5Ø 5B FF 37 E8 EB
[LM5] 2165: 84 E8 BC 84 5B FF 37 B8 21 ØØ 5Ø E8 9F 84 58 21
[FE5] 2166: CØ 75 Ø3 E9 Ø3 ØØ E9 5A ØØ B8 9Ø 37 5Ø B8 Ø2 ØØ
[RK5] 2167: 5Ø B8 9Ø 45 5Ø 5B FF 37 B8 Ø1 ØØ 5Ø E8 9C 84 E8
[1T5] 2168: BA 84 E8 8B 84 5B FF 37 E8 Ø8 ØØ 66 87 75 87 7F
[9M5] 2169: 89 7F 89 E8 1B 8A B8 3Ø 45 5Ø B8 Ø9 ØØ 5Ø 58 5B
[RG5] 217Ø: 89 Ø7 E9 D3 F9 B8 3Ø 45 5Ø B8 47 ØØ 5Ø 58 5B 89
[YD5] 2171: Ø7 E8 12 ØC B8 3Ø 45 5Ø B8 48 ØØ 5Ø 58 5B 89 Ø7
[EN5] 2172: E9 B5 F9 B8 C8 36 5Ø B8 Ø2 ØØ 5Ø B8 82 45 5Ø 5B
[WR5] 2173: FF 37 E8 67 84 E8 38 84 5B FF 37 B8 1B ØØ 5Ø E8
[5H5] 2174: 1B 84 58 21 CØ 75 Ø3 E9 Ø3 ØØ E9 7E ØØ B8 9Ø 37
[HC5] 2175: 5Ø B8 Ø2 ØØ 5Ø B8 9Ø 45 5Ø 5B FF 37 B8 Ø1 ØØ 5Ø
[3S5] 2176: E8 18 84 E8 36 84 E8 Ø7 84 5B FF 37 E8 Ø8 ØØ EA
[EP5] 2177: 87 2C 88 7F 89 7F 89 E8 97 89 B8 3Ø 45 5Ø B8 45
[1H5] 2178: ØØ 5Ø 58 5B 89 Ø7 E8 9D ØB B8 3Ø 45 5Ø B8 49 ØØ
[EE5] 2179: 5Ø 58 5B 89 Ø7 E8 8E ØB B8 9Ø 37 5Ø B8 Ø2 ØØ 5Ø
[XR5] 218Ø: B8 9Ø 45 5Ø 5B FF 37 B8 Ø1 ØØ 5Ø E8 CD 83 E8 EB
[ZE5] 2181: 83 E8 BC 83 B8 Ø2 ØØ 5Ø 58 5B 89 Ø7 B8 3Ø 45 5Ø
[YN5] 2182: B8 4B ØØ 5Ø 58 5B 89 Ø7 E9 ØD F9 B8 C8 36 5Ø B8
[EP5] 2183: Ø2 ØØ 5Ø B8 82 45 5Ø 5B FF 37 E8 BF 83 E8 9Ø 83
[8H5] 2184: 5B FF 37 B8 1Ø ØØ 5Ø E8 73 83 58 21 CØ 75 Ø3 E9
[HB5] 2185: Ø3 ØØ E9 45 1Ø B8 9Ø 37 5Ø B8 Ø2 ØØ 5Ø B8 9Ø 45
[YP5] 2186: 5Ø 5B FF 37 B8 Ø1 ØØ 5Ø E8 7Ø 83 E8 8E 83 E8 5F
[9R5] 2187: 83 5B FF 37 E8 Ø8 ØØ 92 88 D4 88 7F 89 7F 89 E8
[CN5] 2188: EF 88 B8 3Ø 45 5Ø B8 45 ØØ 5Ø 58 5B 89 Ø7 E8 F5
[ZF5] 2189: ØA B8 3Ø 45 5Ø B8 49 ØØ 5Ø 58 5B 89 Ø7 E8 E6 ØA
[AF5] 219Ø: B8 9Ø 37 5Ø B8 Ø2 ØØ 5Ø B8 9Ø 45 5Ø 5B FF 37 B8
[4D5] 2191: Ø1 ØØ 5Ø E8 25 83 E8 43 83 E8 14 83 B8 Ø2 ØØ 5Ø
[UE5] 2192: 58 5B 89 Ø7 B8 3Ø 45 5Ø B8 4D ØØ 5Ø 58 5B 89 Ø7
[DT5] 2193: E9 65 F8 B8 92 45 5Ø 5B FF 37 B8 FØ ØØ 5Ø E8 A3
[VF5] 2194: 82 58 21 CØ 75 Ø3 E9 4D ØØ B8 82 45 5Ø B8 82 45
[KK5] 2195: 5Ø 5B FF 37 B8 Ø1 ØØ 5Ø E8 EØ 82 58 5B 89 Ø7 B8
[6N5] 2196: 9Ø 45 5Ø B8 9Ø 45 5Ø 5B FF 37 B8 Ø1 ØØ 5Ø E8 CA
[8C5] 2197: 82 58 5B 89 Ø7 B8 4A 45 5Ø B8 9Ø 37 5Ø B8 Ø2 ØØ
[3R5] 2198: 5Ø B8 9Ø 45 5Ø 5B FF 37 E8 D1 82 E8 A2 82 5B FF
[LH5] 2199: 37 58 5B 89 Ø7 C3 B8 C8 36 5Ø B8 Ø2 ØØ 5Ø B8 82
[5P5] 22ØØ: 45 5Ø 5B FF 37 E8 B4 82 E8 85 82 B8 92 45 5Ø 5B
```

```
[3R5] 22Ø1: FF 37 58 5B 89 Ø7 B8 82 45 5Ø B8 82 45 5Ø 5B FF
[8K5] 22Ø2: 37 B8 Ø1 ØØ 5Ø E8 68 82 58 5B 89 Ø7 E9 3F DA B8
[2G5] 22Ø3: ØØ ØØ 5Ø ØE E8 1E ØØ 1D 55 6E 76 65 72 74 72 84
[1F5] 22Ø4: 67 6C 69 63 68 65 2Ø 56 61 72 69 61 62 6C 65 6E
[ZK5] 22Ø5: 74 79 7Ø 65 6E E8 88 82 E8 DD 82 58 B8 1A 45 5Ø
[TF5] 22Ø6: B8 3E ØØ 5Ø B8 Ø2 ØØ 5Ø B8 32 45 5Ø 5B FF 37 B8
[ØK5] 22Ø7: Ø1 ØØ 5Ø E8 25 82 E8 43 82 E8 14 82 5B FF 37 E8
[CN5] 22Ø8: AD 8C 58 5A 5B 89 Ø7 89 57 Ø2 B8 22 45 5Ø B8 1A
[LN5] 22Ø9: 45 5Ø 5B FF 77 Ø2 FF 37 B8 ØØ ØØ 5Ø E8 4D 85 E8
[1E5] 221Ø: Ø6 88 58 5B 89 Ø7 B8 ØØ ØØ 5Ø ØE E8 ØA ØØ Ø9 2Ø
[LM5] 2211: 69 6E 2Ø 5A 65 69 6C 65 E8 25 82 E8 7A 82 B8 32
[TP5] 2212: 45 5Ø 5B FF 37 E8 9Ø 82 FE 46 ØØ E8 6A 82 ØE E8
[FG5] 2213: Ø2 ØØ Ø1 28 E8 Ø9 82 E8 5E 82 ØE E8 Ø2 ØØ Ø1 3E
[FS5] 2214: E8 FD 81 B8 22 45 5Ø 5B FF 37 E8 7C 84 E8 48 82
[LN5] 2215: B8 1A 45 5Ø 5B FF 77 Ø2 FF 37 B8 7F 9Ø 5Ø B8 ØØ
[EU5] 2216: FF 5Ø B8 22 45 5Ø 5B FF 37 E8 EØ 84 E8 DE 86 E8
[2G5] 2217: 95 86 E8 99 86 E8 Ø3 85 B8 Ø2 ØØ 5Ø B8 FF ØØ 5Ø
[BK5] 2218: E8 5A 84 E8 12 82 ØE E8 Ø3 ØØ Ø2 29 3A E8 BØ 81
[SM5] 2219: E8 Ø5 82 E8 E9 81 58 B8 ØØ ØØ 5Ø E8 E1 81 58 8C
[WH5] 221Ø: D8 Ø5 5C 17 5Ø 8C D8 Ø5 5C Ø4 5Ø B8 ØØ ØØ 5Ø E8
[YM5] 2211: 8E 81 E8 AC 81 B8 1A 45 5Ø B8 32 45 5Ø 5B FF 37
[5M5] 2212: E8 89 84 58 5A 5B 89 Ø7 89 57 Ø2 E8 2Ø 1F B8 94
[HK5] 2213: 45 5Ø B8 Ø1 ØØ 5Ø 58 5B 89 Ø7 E9 AD 18 B8 44 45
[4N5] 2214: 5Ø B8 Ø4 45 5Ø 5B FF 77 Ø2 FF 37 E8 C8 86 E8 B1
[DF5] 2215: 83 58 5B 89 Ø7 B8 Ø4 45 5Ø B8 Ø4 45 5Ø 5B FF 77
[SH5] 2216: Ø2 FF 37 B8 Ø1 ØØ 5Ø E8 42 84 E8 Ø1 86 58 5A 5B
[9K5] 2217: 89 Ø7 89 57 Ø2 C3 E8 C4 FF B8 Ø8 45 5Ø B8 44 45
[2R5] 2218: 5Ø 5B FF 37 58 5B 89 Ø7 E8 B2 FF B8 Ø8 45 5Ø B8
[HN5] 2219: Ø8 45 5Ø 5B FF 37 B8 44 45 5Ø 5B FF 37 B8 ØØ 89
[TN5] 221Ø: 5Ø B8 ØØ ØØ 5Ø E8 CB 85 E8 Ø2 86 E8 C5 85 E8 BD
[CR5] 2211: 85 E8 F9 8A 58 5B 89 Ø7 C3 E8 81 FF B8 44 45 5Ø
[6K5] 2212: 5B FF 37 B8 2Ø ØØ 5Ø E8 3A 8Ø 58 21 CØ 75 Ø3 E9
[XR5] 2213: Ø3 ØØ E9 E4 FF C3 E8 FA ØA B8 96 45 5Ø B8 44 45
[YK5] 2214: 5Ø 5B FF 37 B8 28 ØØ 5Ø E8 19 8Ø 58 5B 89 Ø7 B8
[LE5] 2215: 46 45 5Ø 8C D8 Ø5 8C 17 5Ø B8 ØØ ØØ 5Ø E8 AØ 8Ø
[DF5] 2216: ØE E8 Ø9 ØØ Ø8 43 4F 4D 4D 41 4E 44 24 E8 9Ø 8Ø
[FP5] 2217: E8 6Ø 82 B8 96 45 5Ø 5B FF 37 E8 49 8Ø 58 5B 89
[ZE5] 2218: Ø7 B8 46 45 5Ø 5B FF 37 58 21 CØ 75 Ø3 E9 Ø1 ØØ
[TG5] 2219: C3 B8 96 45 5Ø 5B FF 37 58 21 CØ 75 Ø3 E9 25 ØØ
[3G5] 221Ø: 8C D8 Ø5 8C 17 5Ø 8C D8 Ø5 8C 17 5Ø B8 ØØ ØØ 5Ø
[1K5] 2211: E8 4D 8Ø ØE E8 Ø2 ØØ Ø1 28 E8 44 8Ø E8 Ø3 81 E8
[WP5] 2212: 5F 8Ø E8 54 FF B8 7Ø 45 5Ø 5B FF 37 58 21 CØ 75
[UG5] 2213: Ø3 E9 Ø3 ØØ E9 74 Ø5 B8 52 45 5Ø 5B FF 37 B8 ØØ
[NH5] 2214: ØØ 5Ø E8 7F 7F 58 21 CØ 75 Ø3 E9 Ø3 ØØ E9 7A ØØ
[ZE5] 2215: B8 3Ø ØØ 5Ø B8 ØØ ØØ 5Ø 58 5B 89 Ø7 8C D8 Ø5 8C
[EN5] 2216: 17 5Ø B8 ØØ ØØ 5Ø E8 F7 7F 8C D8 Ø5 7C Ø4 5Ø B8
[CT5] 2217: 1Ø ØØ 5Ø B8 3Ø ØØ 5Ø 5B FF 37 E8 BF 7F E8 9Ø 7F
[EH5] 2218: B8 ØØ ØØ 5Ø E8 D9 7F E8 A9 81 58 21 CØ 75 Ø3 E9
[EB5] 2219: Ø6 ØØ E9 E3 Ø1 E9 32 ØØ B8 3Ø ØØ 5Ø B8 3Ø ØØ 5Ø
[RM5] 221Ø: 5B FF 37 B8 Ø1 ØØ 5Ø E8 66 7F 58 5B 89 Ø7 B8 3Ø
[UM5] 2211: ØØ 5Ø 5B FF 37 B8 52 45 5Ø 5B FF 37 E8 F2 7E 58
[MF5] 2212: 21 CØ 75 Ø3 E9 Ø3 ØØ E9 92 FF B8 3Ø ØØ 5Ø B8 52
[XG5] 2213: 45 5Ø 5B FF 37 58 5B 89 Ø7 B8 52 45 5Ø B8 52 45
[TK5] 2214: 5Ø 5B FF 37 B8 Ø1 ØØ 5Ø E8 25 7F 58 5B 89 Ø7 B8
[CW5] 2215: 52 45 5Ø 5B FF 37 B8 7E 1F 5Ø 5B FF 37 E8 9E 7E
[EB5] 2216: 58 21 CØ 75 Ø3 E9 25 ØØ B8 ØØ ØØ 5Ø ØE E8 13 ØØ
[NC5] 2217: 12 5A 75 2Ø 76 69 65 6C 65 2Ø 56 61 72 69 61 62
[526] 2218: 6C 65 6E E8 3A 7F E8 8F 7F 58 E9 AF FC 8C D8 Ø5
[VP5] 2219: 7C Ø4 5Ø B8 1Ø ØØ 5Ø B8 3Ø ØØ 5Ø 5B FF 37 E8 FB
[NS5] 221Ø: 7E E8 CC 7E 8C D8 Ø5 8C 17 5Ø B8 ØØ ØØ 5Ø E8 ØF
[YK5] 2211: 7F E8 2D 7F B8 96 45 5Ø 5B FF 37 58 21 CØ 75 Ø3
[SF5] 2212: E9 2B ØØ B8 ØØ ØØ 5Ø ØE E8 19 ØØ 18 46 65 6C 64
[BK5] 2213: 2Ø 6E 69 63 68 74 2Ø 64 69 6D 65 6E 73 69 6F 6E
[AV5] 2214: 69 65 72 74 E8 D9 7E E8 2E 7F 58 E9 4E FC B8 4A
[TH5] 2215: 45 5Ø 5B FF 37 B8 Ø4 ØØ 5Ø E8 15 7E 58 21 CØ 75
[LA5] 2216: Ø3 E9 9E ØØ B8 8Ø 1F 5Ø B8 Ø2 ØØ 5Ø B8 3Ø ØØ 5Ø
[2T5] 2217: 5B FF 37 E8 86 7E E8 57 7E B8 ØC 45 5Ø 5B FF 77
[9R5] 2218: Ø2 FF 37 E8 A7 88 58 5B 89 Ø7 B8 ØC 45 5Ø B8 ØC
[BK5] 2219: 45 5Ø 5B FF 77 Ø2 FF 37 B8 Ø2 ØØ 5Ø B8 4A 45 5Ø
[3S5] 221Ø: 5B FF 37 E8 86 81 E8 4A 83 E8 8B 83 E8 3F 83 58
[KR5] 2211: 5A 5B 89 Ø7 89 57 Ø2 B8 ØC 45 5Ø 5B FF 77 Ø2 FF
[1B5] 2212: 37 B8 ØØ 91 5Ø B8 ØØ ØØ 5Ø E8 Ø7 84 58 21 CØ 75
[BD5] 2213: Ø3 E9 Ø6 ØØ E9 61 ØØ E9 28 ØØ B8 ØØ ØØ 5Ø ØE E8
[YB5] 2214: 16 ØØ 15 56 61 69 61 62 6C 65 6E 73 7Ø 65 69 63
[2R5] 2215: 68 65 72 2Ø 76 6F 6C 6C E8 25 7E E8 7A 7E 58 E9
[BM5] 2216: 9A FB B8 8Ø 1F 5Ø B8 Ø2 ØØ 5Ø B8 3Ø ØØ 5Ø 5B FF
[KU5] 2217: 37 E8 E8 7D E8 B9 7D B8 18 45 5Ø 5B FF 37 58 5B
[SH5] 2218: 89 Ø7 B8 18 45 5Ø B8 18 45 5Ø 5B FF 37 B8 1Ø ØØ
[CN5] 2219: 5Ø E8 9C 7D 58 5B 89 Ø7 B8 4A 45 5Ø 5B FF 37 B8
[HG5] 221Ø: Ø4 ØØ 5Ø E8 2B 7D 58 21 CØ 75 Ø3 E9 26 ØØ B8 Ø8
[8H5] 2211: 45 5Ø B8 8Ø 1F 5Ø B8 Ø2 ØØ 5Ø B8 3Ø ØØ 5Ø 5B FF
[ZW5] 2212: 37 E8 98 7D E8 69 7D 5B FF 37 58 5B 89 Ø7 E8 BD
[CB5] 2213: Ø4 E9 32 ØØ B8 5E 45 5Ø B8 8Ø 1F 5Ø B8 Ø2 ØØ 5Ø
[ZP5] 2214: B8 3Ø ØØ 5Ø 5B FF 37 E8 72 7D E8 43 7D 5B FF 37
[WH5] 2215: 58 5B 89 Ø7 B8 98 45 5Ø B8 5E 45 5Ø 5B FF 37 58
[EM5] 2216: 5B 89 Ø7 E8 DØ ØE B8 96 45 5Ø 5B FF 37 B8 ØØ ØØ
[6H5] 2217: 5Ø E8 DØ 7C 58 21 CØ 75 Ø3 E9 Ø1 ØØ C3 B8 9Ø 37
[ZH5] 2218: 5Ø B8 Ø2 ØØ 5Ø B8 9Ø 45 5Ø 5B FF 37 E8 2D 7D E8
[1S5] 2219: FE 7C B8 4A 45 5Ø 5B FF 37 58 5B 89 Ø7 B8 9Ø 45
[UG5] 221Ø: 5Ø B8 9Ø 45 5Ø 5B FF 37 B8 Ø1 ØØ 5Ø E8 E1 7C 58
[DK5] 2211: 5B 89 Ø7 B8 C8 36 5Ø B8 Ø2 ØØ 5Ø B8 82 45 5Ø 5B
[NS5] 2212: FF 37 E8 F7 7C E8 C8 7C B8 ØØ ØØ 5Ø 58 5B 89 Ø7
[TG5] 2213: B8 82 45 5Ø B8 82 45 5Ø 5B FF 37 B8 Ø1 ØØ 5Ø E8
[BL5] 2214: AE 7C 58 5B 89 Ø7 B8 1Ø 21 5Ø B8 ØA ØØ 5Ø B8 3Ø
[ØR5] 2215: ØØ 5Ø 5B FF 37 E8 C4 7C E8 95 7C B8 Ø2 ØØ 5Ø B8
[FP5] 2216: ØØ ØØ 5Ø E8 B6 7C E8 87 7C 5B FF 37 8F Ø6 Ø8 45
[4K5] 2217: B8 Ø1 ØØ 5Ø 8F Ø6 9A 45 B8 Ø1 ØØ 5Ø E8 BB 7C 8F
[UP5] 2218: Ø6 9C 45 FF 36 9C 45 E8 BA 7D 58 8Ø E4 Ø5 2E 88
[1V5] 2219: 26 9E 8F E9 ØF ØØ FF 36 Ø8 45 FF 36 9C 45 E8 4F
[5M5] 221Ø: 7C 8F Ø6 Ø8 45 FF 36 Ø8 45 FF 36 9A 45 EB ØØ E8
[4K5] 2211: 18 7C EB Ø3 E8 ØØ 7C 58 21 CØ 75 Ø3 E9 5B ØØ B8
[5P5] 2212: C8 36 5Ø B8 Ø2 ØØ 5Ø B8 82 45 5Ø 5B FF 37 E8 4B
[ME5] 2213: 7C E8 1C 7C B8 1Ø 21 5Ø B8 ØA ØØ 5Ø B8 3Ø ØØ 5Ø
[GK5] 2214: 5B FF 37 E8 36 7C E8 Ø7 7C B8 Ø2 ØØ 5Ø B8 Ø8 45
[XT5] 2215: 5Ø 5B FF 37 E8 25 7C E8 F6 7B 5B FF 37 58 5B 89
[UE5] 2216: Ø7 B8 82 45 5Ø B8 82 45 5Ø 5B FF 37 B8 Ø1 ØØ 5Ø
[4S5] 2217: E8 DD 7B 58 5B 89 Ø7 E9 7C FF B8 82 45 5Ø B8 82
[SL5] 2218: 45 5Ø 5B FF 37 B8 Ø1 ØØ 5Ø E8 CF 7B 58 5B 89 Ø7
[BC5] 2219: B8 Ø8 45 5Ø B8 C8 36 5Ø B8 Ø2 ØØ 5Ø B8 82 45 5Ø
[MV5] 221Ø: 5B FF 37 E8 D6 7B E8 A7 7B 5B FF 37 58 5B 89 Ø7
[TG5] 2211: B8 Ø8 45 5Ø 5B FF 37 B8 ØØ ØØ 5Ø E8 46 7B 58 21
[6H5] 2212: CØ 75 Ø3 E9 2D ØØ B8 ØØ ØØ 5Ø ØE E8 1B ØØ 1A 5A
```

```
[WE5] 2313: 75 2Ø 76 69 65 6C 65 2Ø 49 6E 64 69 7A 65 73 2Ø
[ØN5] 2314: 61 6E 67 65 67 65 62 65 6E E8 B4 7B E8 Ø9 7C 58
[RM5] 2315: E9 29 F9 E8 B8 Ø2 E8 8A Ø8 B8 3Ø 45 5Ø B8 ØD ØØ
[EE5] 2316: 5Ø 58 5B 89 Ø7 E8 FE Ø2 B8 3Ø 45 5Ø B8 Ø8 ØØ 5Ø
[TR5] 2317: 58 5B 89 Ø7 E8 EF Ø2 B8 44 45 5Ø 5B FF 37 B8 2C
[TP5] 2318: ØØ 5Ø E8 DF 7A 58 21 CØ 75 Ø3 E9 Ø6 ØØ E8 89 FA
[BN5] 2319: E9 47 FF 8C D8 Ø5 7C 17 5Ø ØE E8 Ø2 ØØ Ø1 29 E8
[LP5] 232Ø: 5E 7B E8 7C 7B E8 EA Ø7 B8 82 45 5Ø B8 82 45 5Ø
[4P5] 2321: 5B FF 37 B8 Ø1 ØØ 5Ø E8 Ø1 7B 58 5B 89 Ø7 B8 C8
[PK5] 2322: 36 5Ø B8 Ø2 ØØ 5Ø B8 82 45 5Ø 5B FF 37 E8 ØC 7B
[3R5] 2323: E8 DD 7A 5B FF 37 B8 ØØ ØØ 5Ø E8 CØ 7A 58 21 CØ
[4H5] 2324: 75 Ø3 E9 2D ØØ B8 ØØ ØØ 5Ø ØE E8 1B ØØ 1A 5A 75
[8G5] 2325: 2Ø 77 65 6E 69 67 2Ø 49 6E 64 69 7A 65 73 2Ø 61
[5S5] 2326: 6E 67 65 67 65 62 65 6E E8 F5 7A E8 4A 7B 58 E9
[HK5] 2327: 6A F8 B8 9Ø 45 5Ø B8 9Ø 45 5Ø 5B FF 37 B8 Ø1 ØØ
[GG5] 2328: 5Ø E8 97 7A 58 5B 89 Ø7 B8 4A 45 5Ø B8 9Ø 37 5Ø
[CP5] 2329: B8 Ø2 ØØ 5Ø B8 9Ø 45 5Ø 5B FF 37 E8 9E 7A E8 6F
[FR5] 233Ø: 7A 5B FF 37 58 5B 89 Ø7 E9 D1 F9 B8 3Ø ØØ 5Ø B8
[GD5] 2331: ØØ ØØ 5Ø 58 5B 89 Ø7 8C D8 Ø5 8C 17 5Ø B8 ØØ ØØ
[4H5] 2332: 5Ø E8 9C 7A 8C D8 Ø5 1C 14 5Ø B8 1Ø ØØ 5Ø B8 3Ø
[ZM5] 2333: ØØ 5Ø 5B FF 37 E8 64 7A E8 35 7A B8 ØØ ØØ 5Ø E8
[ML5] 2334: 7E 7A E8 4E 7C 58 21 CØ 75 Ø3 E9 Ø3 ØØ E9 35 ØØ
[XE5] 2335: B8 3Ø ØØ 5Ø B8 3Ø ØØ 5Ø 5B FF 37 B8 Ø1 ØØ 5Ø E8
[BM5] 2336: ØE 7A 58 5B 89 Ø7 B8 3Ø ØØ 5Ø 5B FF 37 B8 7Ø 45
[WK5] 2337: 5Ø 5B FF 37 E8 9A 79 58 21 CØ 75 Ø3 E9 Ø3 ØØ E9
[6M5] 2338: 95 FF E9 12 FA B8 4A 45 5Ø 5B FF 37 B8 Ø4 ØØ 5Ø
[CH5] 2339: E8 91 79 58 21 CØ 75 Ø3 E9 23 ØØ B8 5E 45 5Ø B8
[4R5] 234Ø: 9C 44 5Ø B8 Ø2 ØØ 5Ø B8 3Ø ØØ 5Ø 5B FF 37 E8 EB
[ØS5] 2341: 79 E8 BC 79 5B FF 37 58 5B 89 Ø7 E9 58 ØB B8 Ø8
[7K5] 2342: 45 5Ø B8 9C 44 5Ø B8 Ø2 ØØ 5Ø B8 3Ø ØØ 5Ø 5B FF
[XW5] 2343: 37 E8 C8 79 E8 99 79 5B FF 37 58 5B 89 Ø7 E9 ED
[1G5] 2344: ØØ B8 14 45 5Ø B8 14 45 5Ø 5B FF 77 Ø2 FF 37 B8
[LN5] 2345: Ø1 ØØ 5Ø E8 D6 7C E8 95 7E 58 5A 5B 89 Ø7 89 57
[AF5] 2346: Ø2 B8 28 45 5Ø 5B FF 37 B8 ØØ ØØ 5Ø E8 15 79 58
[3F5] 2347: 21 CØ 75 Ø3 E9 Ø1 ØØ C3 B8 1A 45 5Ø 5B FF 77 Ø2
[5S5] 2348: FF 37 E8 11 7F B8 22 45 5Ø 5B FF 37 E8 E3 7B B8
[SM5] 2349: 1A 45 5Ø B8 1A 45 5Ø 5B FF 77 Ø2 FF 37 B8 Ø1 ØØ
[WL5] 235Ø: 5Ø E8 88 7C E8 47 7E 58 5A 5B 89 Ø7 89 57 Ø2 C3
[PH5] 2351: B8 14 45 5Ø B8 14 45 5Ø 5B FF 77 Ø2 FF 37 B8 Ø2
[FK5] 2352: ØØ 5Ø E8 67 7C E8 26 7E 58 5A 5B 89 Ø7 89 57 Ø2
[1H5] 2353: B8 28 45 5Ø 5B FF 37 B8 ØØ ØØ 5Ø E8 A6 78 58 21
[ZL5] 2354: CØ 75 Ø3 E9 Ø1 ØØ C3 B8 9E 45 5Ø B8 Ø8 45 5Ø 5B
[FK5] 2355: FF 37 E8 F4 82 58 5B 89 Ø7 B8 22 45 5Ø B8 Ø8 45
[TL5] 2356: 5Ø 5B FF 37 B8 ØØ Ø1 5Ø B8 9E 45 5Ø 5B FF 37 E8
[ZR5] 2357: EA 78 E8 C6 78 58 5B 89 Ø7 E8 5C FF B8 22 45 5Ø
[VP5] 2358: B8 9E 45 5Ø 5B FF 37 58 5B 89 Ø7 E9 4A FF B8 14
[8K5] 2359: 45 5Ø B8 14 45 5Ø 5B FF 77 Ø2 FF 37 B8 Ø4 ØØ 5Ø
[4R5] 236Ø: E8 E9 7B E8 A8 7D 58 5A 5B 89 Ø7 89 57 Ø2 B8 28
[8G5] 2361: 45 5Ø 5B FF 37 B8 ØØ ØØ 5Ø E8 28 78 58 21 CØ 75
[9C5] 2362: Ø3 E9 Ø1 ØØ C3 B8 22 45 5Ø B8 B8 ØØ 5Ø 58 5B 89
[MM5] 2363: Ø7 E8 Ø4 FF E8 7Ø FF B8 22 45 5Ø B8 5Ø ØØ 5Ø 58
[UP5] 2364: 5B 89 Ø7 E9 F2 FE B8 28 45 5Ø 5B FF 37 58 21 CØ
[DC5] 2365: 75 Ø3 E9 Ø3 ØØ E9 9C ØØ B8 48 39 5Ø B8 Ø2 ØØ 5Ø
[EN5] 2366: B8 3Ø 45 5Ø 5B FF 37 E8 52 78 E8 23 78 5B FF 37
[LE5] 2367: B8 ØØ ØØ 5Ø E8 A7 77 58 21 CØ 75 Ø3 E9 Ø3 ØØ E9
[1H5] 2368: 51 ØØ B8 48 39 5Ø B8 Ø2 ØØ 5Ø B8 3Ø 45 5Ø 5B FF
[5K5] 2369: 37 E8 28 78 E8 F9 77 B8 Ø1 ØØ 5Ø 58 5B 89 Ø7 B8
[6N5] 237Ø: 1Ø 45 5Ø B8 1Ø 45 5Ø 5B FF 77 Ø2 FF 37 B8 B8 37
[EH5] 2371: 5Ø B8 Ø2 ØØ 5Ø B8 3Ø 45 5Ø 5B FF 37 E8 FD 77 E8
[9T5] 2372: CE 77 5B FF 37 E8 24 7B E8 E3 7C 58 5A 5B 89 Ø7
[2N5] 2373: 89 57 Ø2 B8 14 45 5Ø B8 14 45 5Ø 5B FF 77 Ø2 FF
[MG5] 2374: 37 B8 Ø3 ØØ 5Ø E8 Ø4 7B E8 C3 7C 58 5A 5B 89 Ø7
[BE5] 2375: 89 57 Ø2 C3 B8 22 45 5Ø B8 E8 ØØ 5Ø 58 5B 89 Ø7
[FP5] 2376: E8 FE FD B8 68 45 5Ø B8 48 39 5Ø B8 Ø2 ØØ 5Ø B8
[XR5] 2377: 3Ø 45 5Ø 5B FF 37 E8 A3 77 E8 74 77 5B FF 37 B8
[9T5] 2378: 1Ø 45 5Ø 5B FF 77 Ø2 FF 37 E8 87 7C E8 78 7C E8
[LT5] 2379: 7C 7C B8 14 45 5Ø 5B FF 77 Ø2 FF 37 E8 68 7C E8
[8S5] 238Ø: 6C 7C B8 Ø2 ØØ 5Ø E8 A3 7A E8 5B 7C E8 5F 7C 58
[XP5] 2381: 5A 5B 89 Ø7 89 57 Ø2 B8 68 45 5Ø 5B FF 77 Ø2 FF
[3E5] 2382: 37 B8 ØØ ØØ 5Ø E8 84 7A E8 28 7D 58 21 CØ 75 Ø3
[RP5] 2383: E9 21 ØØ B8 68 45 5Ø B8 68 45 5Ø 5B FF 77 Ø2 FF
[DD5] 2384: 37 B8 ØØ 91 5Ø B8 ØØ ØØ 5Ø E8 22 7C 58 5A 5B 89
[5E5] 2385: Ø7 89 57 Ø2 B8 Ø8 45 5Ø B8 68 45 5Ø 5B FF 77 Ø2
[SR5] 2386: FF 37 E8 48 81 58 5B 89 Ø7 E9 C4 FD B8 9Ø 45 5Ø
[MM5] 2387: B8 9Ø 45 5Ø 5B FF 37 B8 Ø1 ØØ 5Ø E8 DD 76 58 5B
[8K5] 2388: 89 Ø7 B8 9Ø 37 5Ø B8 Ø2 ØØ 5Ø B8 9Ø 45 5Ø 5B FF
[ØN5] 2389: 37 B8 Ø1 ØØ 5Ø E8 C3 76 E8 E1 76 E8 B2 76 5B FF
[AE5] 239Ø: 37 B8 9Ø 37 5Ø B8 Ø2 ØØ 5Ø B8 9Ø 45 5Ø 5B FF 37
{ØS5] 2391: E8 C9 76 E8 9A 76 5B FF 37 E8 48 76 58 21 CØ 75
[9A5] 2392: Ø3 E9 Ø1 ØØ C3 B8 9Ø 37 5Ø B8 Ø2 ØØ 5Ø B8 9Ø 45
[8R5] 2393: 5Ø 5B FF 37 B8 Ø1 ØØ 5Ø E8 8Ø 76 E8 9E 76 E8 6F
[5L5] 2394: 76 5B FF 37 B8 Ø2 ØØ 5Ø E8 19 76 B8 9Ø 37 5Ø B8
[9N5] 2395: Ø2 ØØ 5Ø B8 9Ø 45 5Ø 5B FF 37 E8 7F 76 E8 5Ø 76
[RN5] 2396: 5B FF 37 B8 Ø1 ØØ 5Ø E8 FA 75 E8 59 76 58 21 CØ
[8E5] 2397: 75 Ø3 E9 ØF ØØ B8 3Ø 45 5Ø B8 45 ØØ 5Ø 58 5B 89
[3H5] 2398: Ø7 E9 E2 FD B8 9Ø 37 5Ø B8 Ø2 ØØ 5Ø B8 9Ø 45 5Ø
[EM5] 2399: 5B FF 37 B8 Ø1 ØØ 5Ø E8 21 76 E8 3F 76 E8 1Ø 76
[CM5] 24ØØ: 5B FF 37 B8 Ø1 ØØ 5Ø E8 BA 75 B8 9Ø 37 5Ø B8 Ø2
[LL5] 24Ø1: ØØ 5Ø B8 9Ø 45 5Ø 5B FF 37 E8 2Ø 76 E8 F1 75 5B
[AR5] 24Ø2: FF 37 B8 Ø2 ØØ 5Ø E8 9B 75 E8 FA 75 58 21 CØ 75
[Y95] 24Ø3: Ø3 E9 33 ØØ B8 9Ø 37 5Ø B8 Ø2 ØØ 5Ø B8 9Ø 45 5Ø
[7T5] 24Ø4: 5B FF 37 B8 Ø1 ØØ 5Ø E8 D1 75 E8 EF 75 E8 C9 75
[3C5] 24Ø5: B8 Ø2 ØØ 5Ø 58 5B 89 Ø7 B8 3Ø 45 5Ø B8 49 ØØ 5Ø
[WM5] 24Ø6: 58 5B 89 Ø7 E9 5F FD B8 ØØ ØØ 5Ø ØE E8 1A ØØ 19
[MD5] 24Ø7: 55 6E 65 72 6C 61 75 62 74 65 2Ø 54 79 7Ø 76 65
[SL5] 24Ø8: 72 6B 6E 81 7Ø 66 75 6E 67 E8 D4 75 E8 29 76 58
[WN5] 24Ø9: E9 49 F3 B8 44 45 5Ø 5B FF 37 B8 41 ØØ 5Ø E8 1Ø
[DM5] 241Ø: 75 B8 44 45 5Ø 5B FF 37 B8 5A ØØ 5Ø E8 EF 74 E8
[CF5] 2411: 7F 75 58 21 CØ 75 Ø3 E9 29 ØØ B8 ØØ ØØ 5Ø ØE E8
[ZE5] 2412: 17 ØØ 16 56 61 72 69 61 62 6C 65 6E 6E 61 6D 65
[7F5] 2413: 2Ø 65 72 77 61 72 74 65 74 E8 84 75 E8 D9 75 58
[1S5] 2414: E9 F9 F2 8C D8 Ø5 8C 17 5Ø ØE E8 Ø1 ØØ ØØ E8 6F
[RK5] 2415: 75 E8 8D 75 8C D8 Ø5 8C 17 5Ø 8C D8 Ø5 8C 17 5Ø
[FL5] 2416: B8 ØØ ØØ 5Ø E8 59 75 B8 44 45 5Ø 5B FF 37 E8 4Ø
[8R5] 2417: 76 E8 ØE 76 E8 6A 75 E8 E3 F3 B8 44 45 5Ø 5B FF
[LH5] 2418: 37 B8 41 ØØ 5Ø E8 AF 74 B8 44 45 5Ø 5B FF 37 B8
[GR5] 2419: 5A ØØ 5Ø E8 B4 74 E8 ED 74 B8 44 45 5Ø 5B FF 37
```

Listing 1. »sbasic.exe« ist der Hexdump des Basic-Compilers. Geben Sie dieses Listing nur dann ein, wenn Sie keine frühere Version von SBasic besitzen.

```
[SL5] 242Ø: B8 3Ø ØØ 5Ø E8 9Ø 74 B8 44 45 5Ø 5B FF 37 B8 39
[UM5] 2421: ØØ 5Ø E8 95 74 E8 CE 74 E8 D6 74 B8 44 45 5Ø 5B
[HS5] 2422: FF 37 B8 2E ØØ 5Ø E8 5B 74 E8 C5 74 58 21 CØ 75
[8M5] 2423: Ø3 E9 Ø3 ØØ E9 7D FF E8 Ø2 F4 B8 4A 45 5Ø B8 Ø1
[9D5] 2424: ØØ 5Ø ØE E8 Ø5 ØØ Ø4 25 21 23 24 E8 D2 74 B8 44
[3R5] 2425: 45 5Ø 5B FF 37 E8 B9 75 E8 C8 76 58 5B 89 Ø7 B8
[WM5] 2426: 4A 45 5Ø 5B FF 37 58 21 CØ 75 Ø3 E9 Ø6 ØØ E8 4C
[NC5] 2427: F3 E9 52 ØØ B8 4A 45 5Ø B8 94 36 5Ø B8 Ø2 ØØ 5Ø
[YG5] 2428: 8C D8 Ø5 8C 17 5Ø B8 ØØ ØØ 5Ø E8 93 74 E8 Ø4 77
[6L5] 2429: B8 41 ØØ 5Ø E8 44 74 E8 62 74 E8 33 74 5B FF 37
[MK5] 243Ø: 58 5B 89 Ø7 B8 4A 45 5Ø 5B FF 37 B8 ØØ ØØ 5Ø E8
[BH5] 2431: D2 73 58 21 CØ 75 Ø3 E9 ØC ØØ B8 4A 45 5Ø B8 Ø2
[DK5] 2432: ØØ 5Ø 58 5B 89 Ø7 8C D8 Ø5 8C 17 5Ø 8C D8 Ø5 8C
[8D5] 2433: 17 5Ø B8 ØØ ØØ 5Ø E8 47 74 ØE E8 Ø5 ØØ Ø4 25 21
[MH5] 2434: 23 24 E8 3B 74 B8 4A 45 5Ø 5B FF 37 B8 Ø1 ØØ 5Ø
[RR5] 2435: E8 CA 76 E8 EC 74 E8 48 74 E9 4Ø F3 B8 4A 45 5Ø
[ZR5] 2436: 5B FF 37 B8 4C 45 5Ø 5B FF 37 E8 77 73 58 21 CØ
[RH5] 2437: 75 Ø3 E9 Ø1 ØØ C3 B8 4A 45 5Ø 5B FF 37 B8 Ø1 ØØ
[PH5] 2438: 5Ø E8 6Ø 73 B8 4C 45 5Ø 5B FF 37 B8 Ø2 ØØ 5Ø E8
[FH5] 2439: 52 73 E8 B1 73 58 21 CØ 75 Ø3 E9 1B ØØ B8 4A 45
[L95] 244Ø: 5Ø B8 Ø2 ØØ 5Ø 58 5B 89 Ø7 B8 3Ø 45 5Ø B8 45 ØØ
[NN5] 2441: 5Ø 58 5B 89 Ø7 E9 2E FB B8 4A 45 5Ø 5B FF 37 B8
[9L5] 2442: Ø2 ØØ 5Ø E8 1E 73 B8 4C 45 5Ø 5B FF 37 B8 Ø1 ØØ
[XF5] 2443: 5Ø E8 1Ø 73 E8 6F 73 58 21 CØ 75 Ø3 E9 1B ØØ B8
[EG5] 2444: 4A 45 5Ø B8 Ø1 ØØ 5Ø 58 5B 89 Ø7 B8 3Ø 45 5Ø B8
[SK5] 2445: 51 ØØ 5Ø 58 5B 89 Ø7 E9 EC FA B8 ØØ ØØ 5Ø ØE E8
[SK5] 2446: Ø6 ØØ Ø5 42 72 65 61 6B E8 75 73 E8 CA 73 58 E9
[GS5] 2447: EA FØ B8 44 45 5Ø 5B FF 37 8C D8 Ø5 7C 17 5Ø B8
[DP5] 2448: ØØ ØØ 5Ø E8 5A 73 E8 CB 75 E8 F1 72 58 21 CØ 75
[ND5] 2449: Ø3 E9 2C ØØ B8 ØØ ØØ 5Ø 8C D8 Ø5 7C 17 5Ø B8 ØØ
[ZE5] 245Ø: ØØ 5Ø E8 3B 73 E8 9Ø 73 ØE E8 ØA ØØ Ø9 2Ø 65 72
[4K5] 2451: 77 61 72 74 65 74 E8 27 73 E8 7C 73 58 E9 9C FØ
[PS5] 2452: E9 36 F2 E8 1C CA B8 4A 45 5Ø 5B FF 37 B8 ØØ ØØ
[8D5] 2453: 5Ø E8 7Ø 72 58 21 CØ 75 Ø3 E9 Ø1 ØØ C3 B8 4C 45
[TN5] 2454: 5Ø B8 Ø1 ØØ 5Ø 58 5B 89 Ø7 E9 DØ FE E8 F3 C9 B8
[BP5] 2455: 4C 45 5Ø B8 Ø2 ØØ 5Ø 58 5B 89 Ø7 E9 BE FE B8 68
[1L5] 2456: 45 5Ø B8 Ø8 45 5Ø 5B FF 37 E8 23 7D B8 14 45 5Ø
[2P5] 2457: 5B FF 77 Ø2 FF 37 E8 8E 77 E8 92 77 B8 Ø2 ØØ 5Ø
[SN5] 2458: E8 C9 75 E8 81 77 E8 85 77 58 5A 5B 89 Ø7 89 57
[ZP5] 2459: Ø2 B8 68 45 5Ø 5B FF 77 Ø2 FF 37 B8 FF 7F 5Ø E8
[DH5] 246Ø: 98 72 E8 A7 75 E8 4B 78 58 21 CØ 75 Ø3 E9 21 ØØ
[8K5] 2461: B8 68 45 5Ø B8 68 45 5Ø 5B FF 77 Ø2 FF 37 B8 ØØ
[CG5] 2462: 91 5Ø B8 ØØ ØØ 5Ø E8 45 77 58 5A 5B 89 Ø7 89 57
[8H5] 2463: Ø2 B8 Ø8 45 5Ø B8 68 45 5Ø 5B FF 77 Ø2 FF 37 E8
[RU5] 2464: 6B 7C 58 5B 89 Ø7 E9 E7 F8 E8 87 FC B8 4A 45 5Ø
[BL5] 2465: 5B FF 37 B8 Ø4 ØØ 5Ø E8 AA 71 58 21 CØ 75 Ø3 E9
[CE5] 2466: 2A ØØ B8 ØØ ØØ 5Ø ØE E8 18 ØØ 17 53 74 72 69 6E
[KC5] 2467: 67 2Ø 61 6C 73 2Ø 4C 61 75 66 76 61 72 69 61 62
[YN5] 2468: 6C 65 E8 1B 72 E8 7Ø 72 58 E9 9Ø EF B8 44 45 5Ø
[5L5] 2469: 5B FF 37 B8 28 ØØ 5Ø E8 6A 71 58 21 CØ 75 Ø3 E9
[8G5] 247Ø: 2A ØØ B8 ØØ ØØ 5Ø ØE E8 18 ØØ 17 49 6E 64 69 7A
[PB5] 2471: 69 65 72 74 65 2Ø 4C 61 75 66 76 61 72 69 61 62
[PN5] 2472: 6C 65 E8 DB 71 E8 3Ø 72 58 E9 5Ø EF B8 3Ø ØØ 5Ø
[MF5] 2473: B8 ØØ ØØ 5Ø 58 5B 89 Ø7 B8 3Ø ØØ 5Ø 5B FF 37 B8
[3M5] 2474: 52 45 5Ø 5B FF 37 E8 1B 71 58 21 CØ 75 Ø3 E9 Ø3
[LH5] 2475: ØØ E9 52 ØØ 8C D8 Ø5 8C 17 5Ø B8 ØØ ØØ 5Ø E8 9F
[1H5] 2476: 71 8C D8 Ø5 7C Ø4 5Ø B8 1Ø ØØ 5Ø B8 3Ø ØØ 5Ø 5B
[ZK5] 2477: FF 37 E8 67 71 E8 38 71 B8 ØØ ØØ 5Ø E8 81 71 E8
[WC5] 2478: 64 73 58 21 CØ 75 Ø3 E9 19 ØØ B8 3Ø ØØ 5Ø E8 3Ø
[3E5] 2479: ØØ 5Ø 5B FF 37 B8 Ø1 ØØ 5Ø E8 14 71 58 5B 89 Ø7
[5K5] 248Ø: E9 95 FF E9 9Ø ØØ 8C D8 Ø5 7C Ø4 5Ø B8 1Ø ØØ 5Ø
[7M5] 2481: B8 3Ø ØØ 5Ø 5B FF 37 E8 22 71 E8 F3 7Ø 8C D3 Ø5
[ZK5] 2482: 8C 17 5Ø B8 ØØ ØØ 5Ø E8 36 71 E8 54 71 B8 8Ø 1F
[7G5] 2483: 5Ø B8 Ø2 ØØ 5Ø B8 3Ø ØØ 5Ø 5B FF 37 E8 FD 7Ø 58
[8U5] 2484: CE 7Ø B8 ØC 45 5Ø 5B FF 77 Ø2 FF 37 E8 1E 7B 58
[TS5] 2485: 5B 89 Ø7 B8 ØC 45 5Ø B8 ØC 45 5Ø 5B FF 77 Ø2 FF
[GK5] 2486: 37 B8 Ø2 ØØ 5Ø B8 4A 45 5Ø 5B FF 37 E8 FD 73 E8
[RH5] 2487: C1 75 E8 Ø2 76 E8 B6 75 58 5A 5B 89 Ø7 89 57 Ø2
[LG5] 2488: B8 52 45 5Ø B8 52 45 5Ø 5B FF 37 B8 Ø1 ØØ 5Ø E8
[SN5] 2489: 7E 7Ø 58 5B 89 Ø7 B8 5E 45 5Ø B8 8Ø 1F 5Ø B8 Ø2
[TH5] 249Ø: ØØ 5Ø B8 3Ø ØØ 5Ø 5B FF 37 E8 9Ø 7Ø E8 61 7Ø 5B
[ZM5] 2491: FF 37 58 5B 89 Ø7 C3 B8 ØØ ØØ 5Ø 8F Ø6 3Ø ØØ B8
[KP5] 2492: Ø2 ØØ 5Ø B8 4A 45 5Ø 5B FF 37 E8 9F 73 E8 63 75
[CM5] 2493: E8 A4 75 B8 Ø2 ØØ 5Ø E8 92 73 E8 4A 75 E8 4E 75
[XN5] 2494: E8 F3 75 8F Ø6 AØ 45 B8 Ø2 ØØ 5Ø 8F Ø6 A2 45 FF
[4M5] 2495: 36 A2 45 E8 6E 71 58 8Ø E4 Ø5 2E 88 26 EA 9B E9
[NH5] 2496: ØF ØØ FF 36 3Ø ØØ FF 36 A2 45 E8 Ø3 7Ø 8F Ø6 3Ø
[PL5] 2497: ØØ FF 36 3Ø ØØ FF 36 AØ 45 EB ØØ E8 CC 6F EB Ø3
[RH5] 2498: E8 B4 6F 58 21 CØ 75 Ø3 E9 2E ØØ B8 Ø8 45 5Ø B8
[SP5] 2499: 8F Ø6 5Ø 58 5B 89 Ø7 E8 B6 F6 B8 Ø8 45 5Ø B8 5E
[ZL5] 25ØØ: 45 5Ø 5B FF 37 B8 3Ø ØØ 5Ø 5B FF 37 E8 C1 6F 58
[ZT5] 25Ø1: 5B 89 Ø7 E8 9A F6 E9 A9 FF C3 B8 Ø2 ØØ 5Ø B8 4A
[9M5] 25Ø2: 45 5Ø 5B FF 37 E8 Ø4 73 E8 C8 74 E8 Ø9 75 B8 Ø2
[HN5] 25Ø3: ØØ 5Ø E8 F7 72 E8 AF 74 E8 B3 74 E8 58 75 8F Ø6
[VE5] 25Ø4: 3Ø ØØ B8 ØØ ØØ 5Ø 8F Ø6 A4 45 B8 Ø2 ØØ 5Ø E8 C9
[4N5] 25Ø5: 6F 8F Ø6 A6 45 FF 36 A6 45 E8 C8 7Ø 58 8Ø E4 Ø5
[WK5] 25Ø6: 2E 88 26 9Ø 9C E9 ØF ØØ FF 36 3Ø ØØ FF 36 A6 45
[YS5] 25Ø7: E8 5D 6F 8F Ø6 3Ø ØØ FF 36 3Ø ØØ FF 36 A4 45 EB
[KS5] 25Ø8: ØØ E8 26 6F EB Ø3 E8 ØE 6F 58 21 CØ 75 Ø3 E9 2E
[YF5] 25Ø9: ØØ B8 Ø8 45 5Ø B8 FF 36 5Ø 58 5B 89 Ø7 E8 1Ø F6
[VH5] 251Ø: B8 Ø8 45 5Ø B8 5E 45 5Ø 5B FF 37 B8 3Ø ØØ 5Ø 5B
[KY5] 2511: FF 37 E8 1B 6F 58 5B 89 Ø7 E8 F4 F5 E9 A9 FF C3
[NK5] 2512: B8 Ø8 45 5Ø B8 58 21 5Ø 58 5B 89 Ø7 E8 E1 F5 B8
[8L5] 2513: Ø8 45 5Ø B8 CØ 75 5Ø 58 5B 89 Ø7 E8 D2 F5 B8 Ø8
[ML5] 2514: 45 5Ø B8 Ø3 E9 5Ø 58 5B 89 Ø7 E8 C3 F5 B8 Ø8 45
[9R5] 2515: 5Ø B8 5E 45 5Ø 5B FF 37 58 5B 89 Ø7 E9 3F FC E8
[6K5] 2516: 54 EE B8 3Ø 45 5Ø B8 24 ØØ 5Ø 58 5B 89 Ø7 E8 75
[ØP5] 2517: F6 B8 4A 45 5Ø 5B FF 37 E8 Ø8 ØØ 36 9D 57 9D 7F
[ØL5] 2518: 89 66 9D E8 4B 74 B8 3Ø 45 5Ø B8 64 ØØ 5Ø 58 5B
[PE5] 2519: 89 Ø7 E8 51 F6 B8 3Ø 45 5Ø B8 51 ØØ 5Ø 58 5B 89
[6H5] 252Ø: Ø7 E8 42 F6 E9 ØF ØØ B8 3Ø 45 5Ø B8 64 ØØ 5Ø 58
[YK5] 2521: 5B 89 Ø7 E8 3Ø F6 E8 5Ø Ø5 B8 44 45 5Ø 5B FF 37
[HG5] 2522: B8 2C ØØ 5Ø E8 1D 6E 58 21 CØ 75 Ø3 E9 Ø6 ØØ E8
[GW5] 2523: C7 ED E9 8A FF C3 B8 Ø8 45 5Ø B8 8C D8 5Ø 58 5B
[WF5] 2524: 89 Ø7 E8 2B F5 B8 22 45 5Ø B8 Ø5 ØØ 5Ø 58 5B 89
[7L5] 2525: Ø7 E8 AD F4 B8 Ø8 45 5Ø B8 ØC 45 5Ø 5B FF 77 Ø2
[1K5] 2526: FF 37 B8 ØF ØØ 5Ø E8 83 71 E8 42 73 B8 1Ø ØØ 5Ø
[BS5] 2527: E8 79 71 E8 7C 73 E8 D8 73 B8 5E 45 5Ø 5B FF 37
[UP5] 2528: E8 69 71 E8 28 73 E8 64 78 58 5B 89 Ø7 E8 EØ F4
[GH5] 2529: B8 22 45 5Ø B8 5Ø ØØ 5Ø 58 5B 89 Ø7 E9 62 F4 E8
[5R5] 253Ø: 74 ED B8 4A 45 5Ø 5B FF 37 B8 Ø4 ØØ 5Ø E8 CD 6D
[2G5] 2531: 58 21 CØ 75 Ø3 E9 Ø1 ØØ C3 B8 ØC 45 5Ø 5B FF 77
```

```
[KH5] 2532: Ø2 FF 37 B8 ØF ØØ 5Ø E8 22 71 E8 E1 72 B8 1Ø ØØ
[GN5] 2533: 5Ø E8 18 71 E8 1B 73 E8 77 73 B8 98 45 5Ø 5B FF
[TS5] 2534: 37 E8 Ø8 71 E8 C7 72 B8 FØ ØF 5Ø E8 FE 7Ø E8 8D
[DF5] 2535: 73 58 21 CØ 75 Ø3 E9 2A ØØ B8 ØØ ØØ 5Ø ØE E8 18
[7B5] 2536: ØØ 17 53 74 72 69 6E 67 2Ø 61 75 E1 65 72 2Ø 52
[WL5] 2537: 65 69 63 68 77 65 69 74 65 E8 C4 6D E8 19 6E 58
[AP5] 2538: E9 39 EB B8 28 45 5Ø 5B FF 37 B8 ØØ ØØ 5Ø E8 13
[3F5] 2539: 6D 58 21 CØ 75 Ø3 E9 46 ØØ B8 3C 3B 5Ø B8 Ø2 ØØ
[ZN5] 254Ø: 5Ø B8 3C 45 5Ø 5B FF 37 E8 71 6D E8 42 6D B8 ØC
[EP5] 2541: 45 5Ø 5B FF 77 Ø2 FF 37 E8 92 77 58 5B 89 Ø7 B8
[2N5] 2542: ØC 45 5Ø B8 ØC 45 5Ø 5B FF 77 Ø2 FF 37 B8 Ø3 ØØ
[9H5] 2543: 5Ø E8 78 7Ø E8 37 72 58 5A 5B 89 Ø7 89 57 Ø2 B8
[8L5] 2544: Ø8 45 5Ø B8 3C 3B 5Ø B8 Ø2 ØØ 5Ø B8 3C 45 5Ø 5B
[CW5] 2545: FF 37 E8 27 6D E8 F8 6C 5B FF 37 58 5B 89 Ø7 E8
[NM5] 2546: 4C F4 B8 3Ø 45 5Ø B8 3C ØØ 5Ø 58 5B 89 Ø7 E8 95
[RF5] 2547: F4 B8 3C 45 5Ø B8 3C 45 5Ø 5B FF 37 B8 Ø1 ØØ 5Ø
[LS5] 2548: E8 CD 6C 58 5B 89 Ø7 B8 3C 45 5Ø 5B FF 37 B8 63
[ØG5] 2549: ØØ 5Ø E8 49 6C 58 21 CØ 75 Ø3 E9 3B ØØ B8 ØØ ØØ
[GE5] 255Ø: 5Ø ØE E8 29 ØØ 28 5A 75 2Ø 76 69 65 6C 65 2Ø 53
[HD5] 2551: 74 72 69 6E 67 7Ø 61 72 61 6D 65 74 65 72 2Ø 62
[WH5] 2552: 65 69 2Ø 43 41 4C 4C 2F 56 41 52 5Ø 54 52 E8 CF
[3R5] 2553: 6C E8 24 6D 58 E9 44 EA C3 B8 44 45 5Ø 5B FF 37
[LE5] 2554: B8 41 ØØ 5Ø E8 56 6C 58 21 CØ 75 Ø3 E9 Ø3 ØØ E9
[NL5] 2555: 1C ØØ E8 48 EB B8 44 45 5Ø 5B FF 37 B8 53 ØØ 5Ø
[MK5] 2556: E8 Ø1 6C 58 21 CØ 75 Ø3 E9 Ø3 ØØ E9 AB EB B8 ØØ
[HB5] 2557: ØØ 5Ø ØE E8 ØC ØØ ØB 41 53 2Ø 65 72 77 61 72 74
[GØ6] 2558: 65 74 E8 7B 6C E8 DØ 6C 58 E9 FØ E9 E8 8A EB E8
[TR5] 2559: A4 EB B8 4A 45 5Ø 5B FF 37 B8 Ø4 ØØ 5Ø E8 FD 6B
[CK5] 256Ø: 58 21 CØ 75 Ø3 E9 Ø3 ØØ E9 A4 E9 B8 44 45 5Ø 5B
[ET5] 2561: FF 37 B8 E7 ØØ 5Ø E8 E4 6B 58 21 CØ 75 Ø3 E9 1F
[5E5] 2562: ØØ B8 ØØ ØØ 5Ø ØE E8 ØD ØØ ØC 27 3D 27 2Ø 65 72
[GP5] 2563: 77 61 72 74 65 74 E8 27 6C E8 7C 6C 58 E9 9C E9
[7S5] 2564: E8 36 EB E8 1C C3 B8 4A 45 5Ø 5B FF 37 B8 Ø4 ØØ
[GL5] 2565: 5Ø E8 A9 6B 58 21 CØ 75 Ø3 E9 Ø3 ØØ E9 5Ø E9 C3
[DN5] 2566: B8 Ø8 45 5Ø B8 ØE E8 5Ø 58 5B 89 Ø7 E8 81 F2 B8
[7H5] 2567: Ø8 45 5Ø 8C D8 Ø5 9C 17 5Ø B8 ØØ 5Ø E8 EØ 6B
[AR5] 2568: E8 A2 6D B8 Ø1 ØØ 5Ø E8 86 6B 58 5B 89 Ø7 E8 5F
[TH5] 2569: F2 B8 22 45 5Ø 8C D8 Ø5 9C 17 5Ø B8 ØØ ØØ 5Ø E8
[2P5] 257Ø: BE 6B E8 8Ø 6D 58 5B 89 Ø7 E8 D5 F1 B8 Ø1 ØØ 5Ø
[3L5] 2571: 8F Ø6 3Ø ØØ 8C D8 Ø5 9C 17 5Ø B8 ØØ ØØ 5Ø E8 9F
[CN5] 2572: 6B E8 61 6D 8F Ø6 A8 45 B8 Ø1 ØØ 5Ø 8F Ø6 AA 45
[HR5] 2573: FF 36 AA 45 E8 8D 6C 58 8Ø E4 Ø5 2E 88 26 CB AØ
[1R5] 2574: E9 ØF ØØ FF 36 3Ø ØØ FF 36 AA 45 E8 22 6B 8F Ø6
[3R5] 2575: 3Ø ØØ FF 36 3Ø ØØ FF 36 A8 45 EB ØØ E8 EB 6A EB
[3G5] 2576: Ø3 E8 D3 6A 58 21 CØ 75 Ø3 E9 2C ØØ B8 22 45 5Ø
[EG5] 2577: 8C D8 Ø5 9C 17 5Ø B8 ØØ ØØ 5Ø E8 43 6B B8 3Ø ØØ
[DN5] 2578: 5Ø 5B FF 37 B8 Ø1 ØØ 5Ø E8 D2 6D E8 A6 6D 58 5B
[6P5] 2579: 89 Ø7 E8 4C F1 E9 AB FF B8 3Ø 45 5Ø B8 12 ØØ 5Ø
[ZR5] 258Ø: 58 5B 89 Ø7 E9 7F F2 E8 B3 E9 B8 68 45 5Ø B8 44
[7L5] 2581: 45 5Ø 5B FF 37 E8 14 6E 58 5A 5B 89 Ø7 89 57 Ø2
[SU5] 2582: E8 9A E9 B8 68 45 5Ø B8 68 45 5Ø 5B FF 77 Ø2 FF
[PE5] 2583: 37 B8 44 45 5Ø 5B FF 37 B8 ØØ 89 5Ø B8 ØØ ØØ 5Ø
[1U5] 2584: E8 BØ 6F E8 E7 6F E8 AA5 6F 58 5A 5B 89 Ø7 89 57
[FF5] 2585: Ø2 B8 28 45 5Ø 5B FF 37 B8 ØØ ØØ 5Ø E8 25 6A 58
[1E5] 2586: 21 CØ 75 Ø3 E9 ØF ØØ B8 Ø8 45 5Ø B8 ØØ ØØ 5Ø 58
[DN5] 2587: 5B 89 Ø7 E9 C3 E9 B8 AC 45 5Ø B8 ØØ ØØ 5Ø 58 5B
[AM5] 2588: 89 Ø7 B8 AE 45 5Ø B8 42 45 5Ø 5B FF 37 B8 Ø1 ØØ
[HF5] 2589: 5Ø E8 47 6A 58 5B 89 Ø7 B8 3Ø ØØ 5Ø B8 AC 45 5Ø
[NR5] 259Ø: 5B FF 37 B8 AE 45 5Ø 5B FF 37 E8 23 6A B8 Ø2 ØØ
[TF5] 2591: 5Ø E8 56 6A 58 5B 89 Ø7 B8 BØ 45 5Ø B8 3E ØØ 5Ø
[KP5] 2592: B8 Ø2 ØØ 5Ø B8 3Ø ØØ 5Ø 5B FF 37 E8 2E 6A E8 FF
[EN5] 2593: 69 5B FF 37 E8 98 74 58 5A 5B 89 Ø7 89 57 Ø2 B8
[BR5] 2594: BØ 45 5Ø 5B FF 77 Ø2 FF 37 B8 68 45 5Ø 5B FF 77
[YH5] 2595: Ø2 FF 37 E8 F2 6F 58 21 CØ 75 Ø3 E9 23 ØØ B8 ØE
[YM5] 2596: 45 5Ø B8 DE ØF 5Ø B8 Ø2 ØØ 5Ø B8 3Ø ØØ 5Ø 5B FF
[VT5] 2597: 37 E8 E8 69 E8 B9 69 5B FF 37 58 5B 89 Ø7 E9 18
[4N5] 2598: E9 B8 3Ø ØØ 5Ø 5B FF 37 B8 AC 45 5Ø 5B FF 37 E8
[HE5] 2599: 52 69 58 21 CØ 75 Ø3 E9 2C ØØ B8 ØØ ØØ 5Ø ØE E8
[9E5] 26ØØ: 1A ØØ 19 55 6E 64 65 66 69 6E 69 65 72 74 65 2Ø
[MP5] 26Ø1: 5A 65 69 6C 65 6E 6E 75 6D 6D 65 72 E8 C1 69 E8
[KR5] 26Ø2: 16 6A 58 E9 36 E7 B8 BØ 45 5Ø 5B FF 77 Ø2 FF 37
[AN5] 26Ø3: B8 68 45 5Ø 5B FF 77 Ø2 FF 37 E8 56 6F 58 21 CØ
[2G5] 26Ø4: 75 Ø3 E9 12 ØØ B8 AC 45 5Ø B8 3Ø ØØ 5Ø 5B FF 37
[5P5] 26Ø5: 58 5B 89 Ø7 E9 Ø1 FF B8 AE 45 5Ø B8 3Ø ØØ 5Ø 5B
[LW5] 26Ø6: FF 37 58 5B 89 Ø7 E9 EF FE B8 4A 45 5Ø 5B FF 37
[5W5] 26Ø7: E8 Ø8 ØØ CE A2 EC A2 7F 89 28 A3 E8 B3 6E B8 Ø8
[9P5] 26Ø8: 45 5Ø B8 58 5B 5Ø 58 5B 89 Ø7 E8 E3 EF B8 Ø8 45
[9L5] 26Ø9: 5Ø B8 89 Ø7 5Ø 58 5B 89 Ø7 E9 D4 EF B8 Ø8 45 5Ø
[4N5] 261Ø: B8 58 5A 5Ø 58 5B 89 Ø7 E8 C5 EF B8 Ø8 45 5Ø B8
[CN5] 2611: 5B 89 5Ø 58 5B 89 Ø7 E8 B6 EF B8 Ø8 45 5Ø B8 Ø7
[CL5] 2612: 89 5Ø 58 5B 89 Ø7 E8 A7 EF B8 Ø8 45 5Ø B8 57 Ø2
[NG5] 2613: 5Ø 58 5B 89 Ø7 E9 98 EF B8 3Ø 45 5Ø B8 13 ØØ 5Ø
[2L5] 2614: 58 5B 89 Ø7 E9 5F FØ B8 3Ø 45 5Ø B8 51 ØØ 5Ø 58
[3H5] 2615: 5B 89 Ø7 E8 5Ø FØ B8 3Ø 45 5Ø B8 65 ØØ 5Ø 58 5B
[UD5] 2616: 89 Ø7 E8 41 FØ B8 9Ø 37 5Ø B8 Ø2 ØØ 5Ø B8 9Ø 45
[9R5] 2617: 5Ø 5B FF 37 B8 Ø1 ØØ 5Ø E8 8Ø 68 E8 9E 68 E8 6F
[2G5] 2618: 68 B8 Ø1 ØØ 5Ø 58 5B 89 Ø7 C3 B8 ØØ 45 5Ø 5B FF
[YM5] 2619: 77 Ø2 FF 37 B8 1E ØØ 5Ø E8 B1 6B E8 7Ø 6D E8 15
[YF5] 262Ø: 6E 8F Ø6 19 ØØ B8 Ø4 45 5Ø B8 61 8E 5Ø B8 ØØ 1C
[NF5] 2621: 5Ø 58 5A 5B 89 Ø7 89 57 Ø2 FF 36 Ø6 45 FF 36 Ø4
[LS5] 2622: 45 E8 F2 6D E8 74 6B B8 94 45 5Ø 5B FF 37 B8 Ø1
[KK5] 2623: ØØ 5Ø E8 CF 67 58 21 CØ 75 Ø3 E9 Ø3 ØØ E8 89 67
[EP5] 2624: B8 ØØ 45 5Ø 5B FF 77 Ø2 FF 37 E8 C9 6D 8F Ø6 19
[CD5] 2625: ØØ B8 2C 45 5Ø B8 ØØ ØØ 5Ø E8 A6 6A B8 ØØ 89 5Ø
[TL5] 2626: B8 ØØ ØØ 5Ø B8 Ø1 ØØ 5Ø E8 97 6A E8 3E 6B E8 3C
[US5] 2627: 6D E8 FF 6C E8 F7 6C 58 5A 5B 89 Ø7 89 57 Ø2 B8
[4H5] 2628: 1A 45 5Ø B8 Ø2 ØØ 5Ø E8 78 6A B8 ØØ 89 5Ø B8 ØØ
[HM5] 2629: ØØ 5Ø B8 Ø3 ØØ 5Ø E8 69 6A E8 1Ø 6B E8 ØE 6D E8
[9N5] 263Ø: D1 6C E8 C9 6C 58 5A 5B 89 Ø7 89 57 Ø2 B8 ØØ 5Ø
[RA5] 2631: 5Ø B8 Ø4 ØØ 5Ø E8 4A 6A B8 ØØ Ø1 5Ø B8 Ø5 ØØ 5Ø
[2N5] 2632: E8 3F 6A E8 B6 67 E8 87 67 58 5B 89 Ø7 B8 ØØ 45
[NU5] 2633: 5Ø 5B FF 77 Ø2 FF 37 B8 A2 Ø3 5Ø E8 CE 6A E8 8D
[NG5] 2634: 6C E8 32 6D 8F Ø6 19 ØØ B8 Ø4 45 5Ø B8 2C 45 5Ø
[CR5] 2635: 5B FF 77 Ø2 FF 37 58 5A 5B 89 Ø7 89 57 Ø2 E8 3C
[4K5] 2636: E6 B8 22 45 5Ø B8 Ø4 45 5Ø 5B FF 77 Ø2 FF 37 E8
[MP5] 2637: Ø4 6D E8 ED 69 58 5B 89 Ø7 B8 44 45 5Ø 5B FF 37
[CH5] 2638: B8 FF ØØ 5Ø E8 DD 66 58 21 CØ 75 Ø3 E9 28 ØØ B8
[KK5] 2639: 22 45 5Ø 5B FF 37 B8 ØØ ØØ 5Ø E8 C7 66 B8 22 45
[8H5] 264Ø: 5Ø 5B FF 37 B8 ØA ØØ 5Ø E8 93 66 E8 23 67 58 21
[HH5] 2641: CØ 75 Ø3 E9 Ø1 ØØ C3 B8 ØØ ØØ 5Ø E8 81 67 58 B8
[4R5] 2642: 44 45 5Ø 5B FF 37 B8 FE ØØ 5Ø E8 DØ 66 58 21 CØ
[AH5] 2643: 75 Ø3 E9 ØF ØØ B8 44 45 5Ø B8 22 45 5Ø 5B FF 37
```

```
[NK5] 2644: 58 5B 89 Ø7 B8 44 45 5Ø 5B FF 37 E8 14 ØØ 7D A5
[PS5] 2645: D3 A5 2E A6 74 A6 AB A6 D3 A6 FB A6 25 A7 4F A7
[WN5] 2646: AD A7 E8 4C 6C B8 ØØ ØØ 5Ø ØE E8 ØD ØØ ØC 51 75
[ZH5] 2647: 65 6C 6C 64 61 74 65 69 2Ø 2D E8 E3 66 E8 38 67
[SH5] 2648: 8C D8 Ø5 5C Ø4 5Ø B8 ØØ ØØ 5Ø E8 D3 66 E8 28 67
[PE5] 2649: ØE E8 ØC ØØ ØB 2D 2Ø 67 65 73 63 68 81 74 7A 74
[ZR5] 265Ø: E8 BD 66 E8 12 67 E8 F6 66 58 E8 DC 65 B8 ØØ ØØ
[BE5] 2651: 5Ø ØE E8 14 ØØ 13 4C 61 75 66 7A 65 69 74 62 69
[PN5] 2652: 62 6C 69 6F 74 68 65 6B 2Ø E8 94 66 E8 E9 66 8C
[RF5] 2653: D8 Ø5 6C 17 5Ø B8 ØØ ØØ 5Ø E8 84 66 E8 D9 66 ØE
[MC5] 2654: E8 13 ØØ 12 42 49 4E 2Ø 6E 69 63 68 74 2Ø 67 65
[VM5] 2655: 66 75 6E 64 65 6E E8 67 66 E8 BC 66 E8 AØ 66 58
[ØL5] 2656: E8 86 65 B8 ØØ ØØ 5Ø ØE E8 ØD ØØ ØC 51 75 65 6C
[NN5] 2657: 6C 64 61 74 65 69 2Ø 2D E8 45 66 E8 9A 66 8C D8
[DG5] 2658: Ø5 5C Ø4 5Ø B8 ØØ ØØ 5Ø E8 35 66 E8 8A 66 ØE E8
[3H5] 2659: 1F ØØ 1E 2D 2Ø 6E 69 63 68 74 2Ø 67 65 66 75 6E
[PE5] 266Ø: 64 65 6E 2Ø 6F 64 65 72 2Ø 75 6E 67 81 6C 74 69
[8L5] 2661: 67 E8 ØC 66 E8 61 66 E8 45 66 58 E8 2B 65 B8 ØØ
[RE5] 2662: ØØ 5Ø ØE E8 ØD ØØ ØC 51 75 65 6C 6C 64 61 74 65
[VS5] 2663: 69 2Ø 2D E8 EA 65 E8 3F 66 8C D8 Ø5 5C Ø4 5Ø B8
[MM5] 2664: ØØ ØØ 5Ø E8 DA 65 E8 2F 66 ØE E8 ØA ØØ Ø9 2D 2Ø
[8T5] 2665: 7A 75 2Ø 67 72 6F E1 E8 C6 65 E8 1B 66 E8 FF 65
[8F5] 2666: 58 E8 E5 64 B8 ØØ ØØ 5Ø ØE E8 25 ØØ 24 5A 65 69
[FH5] 2667: 6C 65 6E 6E 75 6D 6D 65 72 2Ø 6E 69 63 68 74 2Ø
[ØP5] 2668: 3E 2Ø 56 6F 72 7A 65 69 6C 65 6E 6E 75 6D 6D 65
[ZM5] 2669: 72 E8 8C 65 E8 E1 65 58 E9 2F E3 B8 ØØ ØØ 5Ø ØE
[3L5] 267Ø: E8 16 ØØ 15 5A 65 69 6C 65 6E 6E 75 6D 6D 65 72
[8G5] 2671: 2Ø 9A 62 65 72 6C 61 75 66 E8 64 65 E8 B9 65 58
[RH5] 2672: E9 Ø7 E3 B8 ØØ ØØ 5Ø ØE E8 16 ØØ 15 4C 61 62 65
[MF5] 2673: 6C 2Ø 6E 69 63 68 74 2Ø 64 65 66 69 6E 69 65 72
[PM5] 2674: 74 E8 3C 65 E8 91 65 58 E9 DF E2 B8 ØØ ØØ 5Ø ØE
[CE5] 2675: E8 18 ØØ 17 4C 61 62 65 6C 2Ø 62 65 72 65 69 74
[VG5] 2676: 73 2Ø 64 65 66 69 6E 69 65 72 74 E8 12 65 E8 67
[3K5] 2677: 65 58 E9 B5 E2 B8 ØØ ØØ 5Ø ØE E8 16 ØØ 17 44 4F
[XD5] 2678: 55 42 4C 45 2Ø 6E 69 63 68 74 2Ø 76 6F 72 67 65
[HS5] 2679: 73 65 68 65 6E E8 E8 64 E8 3D 65 58 E9 8B E2 B8
[WC5] 268Ø: ØØ ØØ 5Ø ØE E8 13 ØØ 12 2F 47 57 42 3A 2Ø 51 75
[3H5] 2681: 65 6C 6C 64 61 74 65 69 2Ø 2D E8 C3 64 E8 18 65
[WG5] 2682: 8C D8 Ø5 5C Ø4 5Ø B8 ØØ ØØ 5Ø E8 B3 64 E8 Ø8 65
[XD5] 2683: ØE E8 1C ØØ 1B 2D 2Ø 62 65 72 65 69 74 73 2Ø 69
[2G5] 2684: 6D 2Ø 47 57 42 41 53 49 43 2D 46 6F 72 6D 61 74
[FV5] 2685: E8 8D 64 E8 E2 64 E8 C6 64 58 E8 AC 63 8C D8 Ø5
[ØP5] 2686: 4C 17 5Ø B8 ØØ ØØ 5Ø E8 76 64 E8 C2 6B ØE E8 Ø8
[1E5] 2687: ØØ Ø7 23 23 23 2E 47 57 42 E8 64 64 8C D8 Ø5 4C
[BG5] 2688: 17 5Ø B8 ØØ ØØ 5Ø E8 57 64 E8 C5 6B B8 ØØ ØØ 5Ø
[CD5] 2689: ØE E8 Ø8 ØØ Ø7 47 57 42 41 53 49 43 E8 41 64 E8
[XG5] 269Ø: 96 64 58 E9 4A 77 B8 ØØ 45 5Ø B8 ØØ 89 5Ø B8 Ø5
[2M5] 2691: ØØ 5Ø B8 2B ØØ 5Ø E8 89 66 E8 3Ø 67 E8 2E 69 B8
[AP5] 2692: 2A ØØ 5Ø E8 7C 66 E8 23 67 E8 E2 68 58 5A 5B 89
[3K5] 2693: Ø7 89 57 Ø2 B8 ØØ 45 5Ø 5B FF 77 Ø2 FF 37 E8 75
[U95] 2694: 69 8F Ø6 19 ØØ B8 ØØ 89 5Ø B8 ØØ ØØ 5Ø B8 2D ØØ
[VM5] 2695: 5Ø E8 4E 66 E8 F5 66 E8 F3 68 B8 2C ØØ 5Ø E8 41
[YR5] 2696: 66 E8 E8 66 E8 A7 68 B8 Ø1 ØØ 5Ø E8 DE 66 E8 96
[XL5] 2697: 68 E8 9A 68 E8 3F 69 8F Ø6 19 ØØ B8 3Ø ØØ 5Ø B8
[VD5] 2698: ØØ ØØ 5Ø 58 5B 89 Ø7 B8 3Ø ØØ 5Ø B8 3Ø ØØ 5Ø 5B
[EG5] 2699: FF 37 B8 Ø1 ØØ 5Ø E8 57 63 58 5B 89 Ø7 B8 3Ø ØØ
[9N5] 27ØØ: 5Ø 5B FF 37 E8 FB 65 B8 ØØ ØØ 5Ø E8 F6 62 B8 3Ø
[GG5] 27Ø1: ØØ 5Ø 5B FF 37 B8 Ø1 ØØ 5Ø E8 34 63 E8 E3 65 B8
[2M5] 27Ø2: ØØ ØØ 5Ø E8 DE 62 E8 3D 63 B8 3Ø ØØ 5Ø 5B FF 37
[6K5] 27Ø3: B8 Ø2 ØØ 5Ø E8 19 63 E8 C8 65 B8 Ø1 ØØ 5Ø E8 C3
[MF5] 27Ø4: 62 E8 22 63 B8 3Ø ØØ 5Ø 5B FF 37 B8 Ø3 ØØ 5Ø E8
[ØN5] 27Ø5: FE 62 E8 AD 65 B8 ØØ ØØ 5Ø E8 A8 62 E8 Ø7 63 58
[7E5] 27Ø6: 21 CØ 75 Ø3 E9 2A ØØ B8 3Ø ØØ 5Ø B8 3Ø ØØ 5Ø 5B
[CN5] 27Ø7: FF 37 B8 Ø3 ØØ 5Ø E8 D7 62 58 5B 89 Ø7 8C D8 Ø5
[BH5] 27Ø8: 6C 17 5Ø ØE E8 Ø1 ØØ ØØ E8 15 63 E8 33 63 E9 Ø3
[ZH5] 27Ø9: ØØ E9 53 FF B8 3Ø ØØ 5Ø B8 3Ø ØØ 5Ø 5B FF 37 B8
[CL5] 271Ø: Ø1 ØØ 5Ø E8 AA 62 58 5B 89 Ø7 B8 B4 45 5Ø B8 3Ø
[CM5] 2711: ØØ 5Ø 5B FF 37 E8 4A 65 58 5B 89 Ø7 8C D8 Ø5 6C
[LF5] 2712: 17 5Ø 8C D8 Ø5 6C 17 5Ø B8 ØØ ØØ 5Ø E8 D1 62 B8
[3S5] 2713: B4 45 5Ø 5B FF 37 E8 B8 63 E8 86 63 E8 E2 62 B8
[BM5] 2714: B4 45 5Ø 5B FF 37 B8 2E ØØ 5Ø E8 5Ø 62 58 21 CØ
[4H5] 2715: 75 Ø3 E9 Ø3 ØØ E9 9C FF B8 ØØ 45 5Ø B8 ØØ 45 5Ø
[ØR5] 2716: 5B FF 77 Ø2 FF 37 B8 ØØ 27 5Ø E8 9F 65 E8 5E 67
[GG5] 2717: 58 5A 5B 89 Ø7 89 57 Ø2 8C D8 Ø5 3C 17 5Ø B8 ØØ
[RF5] 2718: ØØ 5Ø E8 7B 62 ØE E8 Ø4 ØØ Ø3 47 57 42 E8 7Ø 62
[7G5] 2719: E8 4Ø 64 58 21 CØ 75 Ø3 E9 13 ØØ B8 1A 45 5Ø B8
[GN5] 272Ø: FF FF 5Ø E8 66 65 58 5A 5B 89 Ø7 89 57 Ø2 B8 ØØ
[6S5] 2721: 45 5Ø 5B FF 77 Ø2 FF 37 E8 BB 67 8F Ø6 19 ØØ 8C
[3K5] 2722: D8 Ø5 5C 17 5Ø 8C D8 Ø5 5C 17 5Ø B8 ØØ ØØ 5Ø E8
[ZL5] 2723: 2E 62 B8 ØØ ØØ 5Ø E8 18 63 E8 E6 62 8C D8 Ø5 6C
[VE5] 2724: 17 5Ø B8 ØØ ØØ 5Ø E8 17 62 E8 D6 62 ØE E8 Ø4 ØØ
[KL5] 2725: Ø3 42 49 4E E8 Ø9 62 E8 C8 62 B8 ØØ ØØ 5Ø E8 FØ
[LM5] 2726: 62 E8 BE 62 E8 1A 62 B8 22 45 5Ø B8 1A 45 5Ø 5B
[RK5] 2727: FF 77 Ø2 FF 37 B8 ØØ 89 5Ø B8 ØØ ØØ 5Ø E8 F2 66
[4M5] 2728: E8 4E 67 E8 5Ø 67 58 5B 89 Ø7 B8 ØØ ØØ 5Ø B8 1A
[NH5] 2729: 45 5Ø 5B FF 77 Ø2 FF 37 B8 ØØ 89 5Ø B8 ØØ ØØ 5Ø
[TM5] 273Ø: B8 22 45 5Ø 5B FF 37 E8 C2 64 E8 CØ 66 E8 77 66
[KH5] 2731: E8 7B 66 E8 2Ø 67 E8 F9 63 B8 Ø1 ØØ 5Ø B8 22 45
[KM5] 2732: 5Ø 5B FF 37 E8 EB 63 B8 Ø1 ØØ 5Ø 8F Ø6 3Ø ØØ 8C
[2F5] 2733: D8 Ø5 5C 17 5Ø B8 ØØ ØØ 5Ø E8 84 61 E8 46 63 8F
[LM5] 2734: Ø6 B6 45 B8 Ø1 ØØ 5Ø 8F Ø6 B8 45 FF 36 B8 45 E8
[7M5] 2735: 72 62 58 8Ø E4 Ø5 2E 88 26 E6 AA E9 ØF ØØ FF 36
[6F5] 2736: 3Ø ØØ FF 36 B8 45 E8 Ø7 61 8F Ø6 3Ø ØØ FF 36 3Ø
[WL5] 2737: ØØ FF 36 B6 45 EB ØØ E8 DØ 6Ø EB Ø3 E8 B8 6Ø 58
[DG5] 2738: 21 CØ 75 Ø3 E9 32 ØØ B8 3Ø ØØ 5Ø 5B FF 37 B8 Ø1
[UM5] 2739: ØØ 5Ø E8 DB 6Ø 8C D8 Ø5 5C 17 5Ø B8 ØØ ØØ 5Ø E8
[AH5] 274Ø: 1E 61 B8 3Ø ØØ 5Ø 5B FF 37 B8 Ø1 ØØ 5Ø E8 AD 63
[AR4] 2741: E8 81 63 E8 5C 63 E9 A5 FF C3 E8 2C 6Ø
```

Listing 1. »sbasic.exe« ist der Hexdump des Basic-Compilers. Geben Sie dieses Listing nur dann ein, wenn Sie keine frühere Version von SBasic besitzen.

```
[CU5]    1: 38 Ø1 91 ØC A6 ØC B9 ØC CC ØC DF ØC F2 ØC Ø5 ØD
[9R5]    2: 18 ØD 23 ØD 2E ØD 39 ØD 44 ØD 4F ØD 5D ØD 6D ØD
[VU5]    3: 7D ØD 83 ØD 89 ØD AA ØD C8 ØD E1 ØD Ø1 ØE 42 ØE
[WX5]    4: 4B ØE 7A ØE 8D ØE AØ ØE BC ØE EØ ØE 28 ØF 46 ØF
[S85]    5: CD 1Ø DA 1Ø 12 11 6Ø 11 7Ø 11 Ø2 12 1Ø 12 1C 12
[FD5]    6: 34 12 62 12 75 12 88 12 9B 12 AE 12 C1 12 D4 12
[9G5]    7: FF 12 Ø9 13 4B 13 67 13 74 13 9A 13 AA 13 BC 13
```

```
[7K5]    8: D1 13 E5 13 ØE 14 48 14 6C 14 8E 14 9F 14 EØ 14
[3E5]    9: 1C 15 4C 15 EØ 15 13 16 46 16 A6 16 D5 16 61 18
[RH5]   1Ø: 68 18 6D 18 A7 18 AC I8 B1 18 B6 18 FØ 18 1Ø 19
[1L5]   11: 36 19 3B 19 63 19 78 19 8D 19 A2 19 B7 19 CC 19
[PV5]   12: E1 19 F1 19 ØA 1A 7C 1A 97 1A AB 1A BD 1A D9 1A
[PY5]   13: EC 1A F8 1A Ø4 1B 1C 1B 2D 1B B1 1C DF 1C F1 1C
[PU5]   14: Ø6 1D 22 1D 9D 1D D2 1D 23 1E 6B 1E 7F 1E 9C 1E
[HY5]   15: DC 1E ØE 1F 1A 1F 3E 1F 6C 1F 78 1F 8D 1F D2 1F
[8F5]   16: E3 1F ØØ 2Ø 2E 2Ø BA 2Ø C8 2Ø 95 21 B1 21 D6 21
[ZE5]   17: F8 21 1A 22 3C 22 47 22 4D 22 6C 22 EC 22 6D 23
[2D5]   18: 7B 23 87 23 11 26 1D 26 24 26 46 26 4E 28 83 28
[DR5]   19: 14 29 43 29 48 29 F6 29 47 2A A9 2A AE 2A B3 2A
[G26]   2Ø: DB 2A 7A 2C CA 2C FF FF 8C DD 8C CA Ø3 C2 8E D8
[EP5]   21: 8E CØ 23 DB 74 11 33 CØ 33 FF B9 Ø8 ØØ F3 AB 8C
[HP5]   22: CØ 4Ø 8E CØ 4B 75 EF 89 2E 2A ØØ C6 Ø6 ØØ ØØ FF
[4M5]   23: C6 Ø6 Ø4 ØØ Ø4 C6 Ø6 Ø9 ØØ Ø7 B4 ØF CD 1Ø FE CC
[BD5]   24: 88 26 ØB ØØ C6 Ø6 ØD ØØ 17 8C 1E 19 ØØ B4 Ø3 CD
[HK5]   25: 1Ø 89 ØE ØF ØØ C7 Ø6 15 ØØ FF FF 8Ø FE 18 72 ØB
[9L5]   26: B8 ØA ØE CD 1Ø B6 17 B4 Ø2 CD 1Ø B9 ØØ 1F B4 Ø1
[TL5]   27: CD 1Ø BA Ø7 Ø7 B4 ØF CD 1Ø 3C Ø7 75 Ø3 BA ØC ØB
[7F5]   28: 89 16 24 ØØ C7 Ø6 26 ØØ A4 8E C7 Ø6 28 ØØ 1F 7F
[DB5]   29: C7 Ø6 2C ØØ 3Ø Ø4 C7 Ø6 2E ØØ 78 Ø7 C3 ØØ ØØ ØØ
[ØZ4]   3Ø: ØØ ØØ ØØ ØØ ØØ ØØ ØØ ØØ ØØ ØØ ØØ ØØ ØØ ØØ ØØ ØØ
[ØZ4]   31: ØØ ØØ ØØ ØØ ØØ ØØ ØØ ØØ ØØ ØØ ØØ ØØ ØØ ØØ ØØ ØØ
[ØZ4]   32: ØØ ØØ ØØ ØØ ØØ ØØ ØØ ØØ ØØ ØØ ØØ ØØ ØØ ØØ ØØ ØØ
[ØZ4]   33: ØØ ØØ ØØ ØØ ØØ ØØ ØØ ØØ ØØ ØØ ØØ ØØ ØØ ØØ ØØ ØØ
[ØZ4]   34: ØØ ØØ ØØ ØØ ØØ ØØ ØØ ØØ ØØ ØØ ØØ ØØ ØØ ØØ ØØ ØØ
[ØZ4]   35: ØØ ØØ ØØ ØØ ØØ ØØ ØØ ØØ ØØ ØØ ØØ ØØ ØØ ØØ ØØ ØØ
[ØZ4]   36: ØØ ØØ ØØ ØØ ØØ ØØ ØØ ØØ ØØ ØØ ØØ ØØ ØØ ØØ ØØ ØØ
[ØZ4]   37: ØØ ØØ ØØ ØØ ØØ ØØ ØØ ØØ ØØ ØØ ØØ ØØ ØØ ØØ ØØ ØØ
[6C5]   38: ØØ ØØ ØØ ØØ ØØ ØØ ØØ ØØ ØØ 5Ø 52 BA FD Ø3 EC DØ
[SP5]   39: E8 73 1F B2 F8 EC 53 2E 8A 1E 1D Ø1 43 83 E3 3F
[3L5]   4Ø: 2E 3A 1E 1E Ø1 74 ØA 2E 88 87 95 ØØ 2E 88 1E 1D
[MS5]   41: Ø1 5B BØ 64 E6 2Ø 5A 58 CF 5Ø 52 BA FD Ø2 EC DØ
[UP5]   42: E8 73 1F B2 F8 EC 53 2E 8A 1E 1F Ø1 43 83 E3 3F
[6K5]   43: 2E 3A 1E 2Ø Ø1 74 ØA 2E 88 87 D5 ØØ 2E 88 1E 1F
[8K5]   44: Ø1 5B BØ 63 E6 2Ø 5A 58 CF 3C ØD 74 Ø9 3C ØA 74
[SK5]   45: Ø9 FE 47 Ø4 EB Ø4 C6 47 Ø4 ØØ 8A 1F B9 Ø1 ØØ A2
[ES5]   46: ØE ØØ BA ØE ØØ B4 4Ø CD 21 5B C3 53 8A ØF 8Ø F9
[2T5]   47: FD 72 D6 5Ø 8Ø F9 FE 75 ØE BA FD Ø3 EC A8 2Ø 74
[WZ5]   48: FB B2 F8 58 EE 5B C3 8Ø F9 FD 75 ØE BA FD Ø2 EC
[XV5]   49: A8 2Ø 74 FB B2 F8 58 EE 5B C3 B4 ØF CD 1Ø B4 Ø3
[1N5]   5Ø: CD 1Ø 59 8A 1E Ø9 ØØ 8Ø F9 Ø7 75 Ø7 B8 Ø7 ØE CD
[BN5]   51: 1Ø 5B C3 8Ø F9 Ø9 75 2E E8 CC ØØ B4 Ø3 CD 1Ø 8A
[WH5]   52: C2 24 Ø7 B9 Ø8 ØØ 2A C8 B8 2Ø Ø9 CD 1Ø C6 Ø6 ØA
[PH5]   53: ØØ ØØ Ø2 D1 3A 16 ØB ØØ 76 Ø6 FE CA FE ØE ØA ØØ
[TK5]   54: B4 Ø2 CD 1Ø 5B C3 8Ø F9 ØA 74 Ø5 8Ø F9 ØD 75 Ø5
[VH5]   55: E8 A1 ØØ 5B C3 8Ø F9 ØB 75 11 B2 ØØ 8A 36 ØC ØØ
[AG5]   56: B4 Ø2 CD 1Ø C6 Ø6 ØA ØØ ØØ 5B C3 8Ø F9 ØC 75 Ø8
[ZM5]   57: B8 ØØ Ø6 E8 9C ØØ EB E2 8Ø F9 1C 75 12 B4 Ø3 CD
[8G5]   58: 1Ø 3A 16 ØB ØØ 74 Ø4 FE C2 EB 15 B2 ØØ EB Ø9 8Ø
[5M5]   59: F9 1F 75 17 B4 Ø3 CD 1Ø 3A 36 ØD ØØ 74 ØB FE C6
[WF5]   6Ø: B4 Ø2 CD 1Ø C6 Ø6 ØA ØØ ØØ 5B C3 8Ø F9 1D 75 13
[5P5]   61: B4 Ø3 CD 1Ø 8Ø FA ØØ 74 Ø4 FE CA EB E3 8A 16 ØB
[NG5]   62: ØØ EB Ø9 8Ø F9 1E 75 ØE B4 Ø3 CD 1Ø 3A 36 ØC ØØ
[SU5]   63: 74 D7 FE CE EB CA E8 ØE ØØ 8A C1 B4 Ø9 B9 Ø1 ØØ
[SM5]   64: CD 1Ø E8 46 ØØ 5B C3 3A 16 ØB ØØ 75 3D 8Ø 3E ØA
[DA5]   65: ØØ ØØ 74 36 51 B2 ØØ 3A 36 ØD ØØ 75 Ø8 B8 Ø1 Ø6
[XL5]   66: E8 ØF ØØ EB Ø2 FE C6 B4 Ø2 CD 1Ø C6 Ø6 ØA ØØ ØØ
[NH5]   67: 59 C3 53 51 52 8A FB B1 ØØ 8A 2E ØC ØØ 8A 16 ØB
[PL5]   68: ØØ 8A 36 ØD ØØ CD 1Ø 5A 59 5B C3 B4 Ø3 CD 1Ø 3A
[MG5]   69: 16 ØB ØØ 74 ØC FE C2 B4 Ø2 CD 1Ø C6 Ø6 ØA ØØ ØØ
[VL5]   7Ø: C3 C6 Ø6 ØA ØØ FF C3 BØ 1A 8B 3E 17 ØØ 23 FF 74
[UH5]   71: ØF 2E 8A Ø5 47 3C 1A 75 Ø3 2E 8B 3D 89 3E 17 ØØ
[ZN5]   72: C3 23 DB 78 E2 8A Ø7 3C FF 74 71 53 3C FE 75 2Ø
[1S5]   73: 2E 8A 1E 1E Ø1 2E 3A 1E 1D Ø1 74 F4 FE C3 8Ø E3
[RM5]   74: 3F 2E 8A 87 95 ØØ 2E 88 1E 1E Ø1 5B 88 7F 1Ø C3
[BL5]   75: 3C FD 75 2Ø 2E 8A 1E 2Ø Ø1 2E 3A 1E 1F Ø1 74 F4
[FP5]   76: FE C3 8Ø E3 3F 2E 8A 87 D5 ØØ 2E 88 1E 2Ø Ø1 5B
[GL5]   77: 88 7F 1Ø C3 5B 8A 4F 1Ø 51 8Ø F9 1A 74 17 53 8A
[NK5]   78: D8 B9 Ø1 ØØ BA ØE ØØ B4 3F CD 21 5B 72 Ø7 23 CØ
[UE5]   79: AØ ØE ØØ 75 Ø2 BØ 1A 88 47 1Ø 58 C3 AØ ØE ØØ 3C
[WL5]   8Ø: FF 75 Ø5 B4 Ø8 CD 21 C3 83 3E 15 ØØ FF 74 28 3C
[WM5]   81: ØØ 75 Ø3 BØ 1A C3 FE ØE ØE ØØ 53 B4 ØF CD 1Ø B4
[8G5]   82: Ø8 CD 1Ø 5Ø B4 Ø3 CD 1Ø 3A 16 ØB ØØ 73 Ø6 FE C2
[4M5]   83: B4 Ø2 CD 1Ø 58 5B C3 53 B4 ØF CD 1Ø B4 Ø3 CD 1Ø
[ZF5]   84: 89 ØE 15 ØØ 52 33 FF 3A 16 ØB ØØ 76 Ø4 8A 16 ØB
[5F5]   85: ØØ B4 Ø2 CD 1Ø 8B ØE 24 ØØ 23 FF 74 Ø2 B5 Ø4 B4
[3F5]   86: Ø1 CD 1Ø B4 Ø8 CD 21 5Ø B9 ØØ 1F B4 Ø1 CD 1Ø 58
[VH5]   87: 3C 2Ø 72 44 8A 1E Ø9 ØØ 23 FF 74 31 5Ø 8A CA 8A
[2N5]   88: 16 ØB ØØ FE CA 78 1F 3A D1 72 1B B4 Ø2 CD 1Ø B4
[8N5]   89: Ø8 CD 1Ø FE C2 B4 Ø2 CD 1Ø 51 B9 Ø1 ØØ B4 Ø9 CD
[EP5]   9Ø: 1Ø 59 FE CA EB DD 8A D1 B4 Ø2 CD 1Ø 58 B9 Ø1 ØØ
[3U5]   91: B4 Ø9 CD 1Ø FE C2 EB 8F 3C ØØ 75 4D B4 Ø8 CD 21
[XU5]   92: 3C 4B 75 ØB 58 5Ø 3A DØ 76 Ø2 FE CA E9 78 FF 3C
[HV5]   93: 4D 75 Ø4 FE C2 EB F5 3C 47 75 Ø4 5A 52 EB ED 3C
[PL5]   94: 4F 75 1A 8A 16 ØB ØØ 59 51 B4 Ø2 CD 1Ø B4 Ø8 CD
[VU5]   95: 1Ø 3C 2Ø 75 DE 3A D1 74 D3 FE CA EB EC 3C 53 74
[UP5]   96: 14 3C 52 75 C7 F7 D7 EB C3 3C Ø8 75 3E 59 51 3A
[2N5]   97: D1 76 35 FE CA 52 8A 1E Ø9 ØØ 3A 16 ØB ØØ 73 1B
[9V5]   98: FE C2 B4 Ø2 CD 1Ø B4 Ø8 CD 1Ø FE CA B4 Ø2 CD 1Ø
[3S5]   99: B9 Ø1 ØØ B4 Ø9 CD 1Ø FE C2 EB DF B4 Ø2 CD 1Ø B8
[7P5]  1ØØ: 2Ø Ø9 B9 Ø1 ØØ CD 1Ø 5A E9 FC FE 3C ØD 74 Ø7 3C
[7R5]  1Ø1: ØA 74 F5 E9 1E FF 59 8A 16 ØB ØØ B4 Ø2 CD 1Ø B4
[NT5]  1Ø2: Ø8 CD 1Ø 3C 2Ø 75 Ø8 FE CA 78 Ø4 3A D1 73 EC FE
[8S5]  1Ø3: C2 2A D1 88 16 ØE ØØ 8A D1 B4 Ø2 CD 1Ø 5B E9 7B
[EV5]  1Ø4: FE 8Ø 3F FF 75 13 BØ ØD E8 6Ø FC 8B ØE 15 ØØ B4
[6N5]  1Ø5: Ø1 CD 1Ø C7 Ø6 15 ØØ FF FF C2 Ø2 ØØ 8B EC 83 C5
[TF5]  1Ø6: Ø4 8A 8E ØØ Ø1 8A 6E ØØ 33 FF E3 1C 8Ø F9 ØØ 74
[SP5]  1Ø7: 14 8Ø FD ØØ 74 15 47 FE C9 FE CD 8A 83 ØØ Ø1 3A
[WL5]  1Ø8: Ø3 74 E7 77 Ø6 BØ Ø2 C3 BØ Ø1 C3 BØ Ø4 C3 8B EC
[5F5]  1Ø9: B2 ØØ 8A 76 Ø6 8B 76 Ø7 8A 5E Ø9 BØ ØØ 8A 66 Ø2
[CS5]  11Ø: 8B 7E Ø3 8A 7E Ø5 3A DF 73 Ø5 92 87 F7 86 DF 8Ø
[FN5]  111: FF ØØ 74 6A 8A CB 2A CF 8Ø F9 18 77 61 B5 ØØ 56
[9R5]  112: 33 F7 9C 8B 76 FE 81 CE ØØ 8Ø 81 CF ØØ 8Ø E3 Ø6
[NS5]  113: D1 EF D1 D8 E2 FA 9D 78 ØE Ø3 DØ 13 F7 73 39 D1
[3T5]  114: DE D1 DA FE C3 EB 31 2B DØ 1B F7 72 Ø8 75 14 23
```

Listing 2. »sbasic.bin« stellt die neueste Version der Laufzeitbibliothek dar. Bitte nur dann eingeben, wenn Sie die aktuelle Assembler-Laufzeitbibliothek von SBasic nicht besitzen.

```
[BN5]  115: D2 75 1Ø EB 1A F7 D2 83 C2 Ø1 F7 D6 83 D6 ØØ 8Ø
[YS5]  116: 76 FF 8Ø 23 F6 78 11 D1 E2 D1 D6 FE CB 75 F4 33
[BR5]  117: F6 33 D2 B3 ØØ 58 EB Ø6 58 ØD FF 7F 23 FØ 88 76
[YF5]  118: Ø6 89 76 Ø7 88 5E Ø9 C2 Ø4 ØØ 8B EC 8Ø 7E Ø9 ØØ
[XM5]  119: 74 5B 8Ø 7E Ø5 ØØ 74 55 8A 5E Ø2 8A 7E Ø6 8A C3
[7U5]  12Ø: F6 E7 8B F8 8B 76 Ø3 8B 6E Ø7 8B C6 33 C5 ØD FF
[BS5]  121: 7F 5Ø 81 CE ØØ 8Ø 81 CD ØØ 8Ø 8B C6 F7 E5 8B C8
[NM5]  122: 87 F2 BØ ØØ 8A E7 F7 E2 Ø3 F8 13 CA 83 D6 ØØ BØ
[VT5]  123: ØØ 8A E3 F7 E5 Ø3 F8 13 CA 83 D6 ØØ 58 8B EC 8A
[ZN5]  124: 5E Ø5 8A 7E Ø9 81 F3 8Ø 8Ø Ø2 DF 71 ØB 33 CØ 89
[AC5]  125: 46 Ø6 89 46 Ø8 C2 Ø4 ØØ 8Ø F3 8Ø 23 F6 78 ØA D1
[CP5]  126: E7 D1 D1 D1 D6 FE CB 75 F2 81 C1 8Ø ØØ 83 D6 ØØ
[2S5]  127: 73 Ø6 D1 DE D1 D9 FE C3 23 FØ 88 6E Ø6 89 76 Ø7
[AK5]  128: 88 5E Ø9 C2 Ø4 ØØ 8B EC 8Ø 7E Ø9 ØØ 74 76 8Ø 7E
[8L5]  129: Ø5 ØØ 74 73 8B 76 Ø6 8A 5E Ø8 B7 ØØ 8B 7E Ø2 8A
[PM5]  13Ø: 56 Ø4 B6 ØØ 8A C3 32 C2 ØC 7F 5Ø 8Ø CB 8Ø 8Ø CA
[MK5]  131: 8Ø B1 19 33 ED BØ ØØ 2B F7 1B DA 73 Ø4 Ø3 F7 13
[CW5]  132: DA F5 D1 D5 DØ DØ D1 E6 D1 D3 FE C9 22 CØ 79 E7
[ØR5]  133: 2B F7 1B DA F5 83 D5 ØØ 14 ØØ 73 Ø6 DØ D8 D1 DD
[NP5]  134: FE C1 5B 8B D5 8B EC 22 C3 89 56 Ø6 88 46 Ø8 8A
[1C5]  135: 46 Ø9 8A 66 Ø5 35 8Ø 8Ø Ø2 C1 2A C4 7Ø Ø9 35 8Ø
[DB5]  136: ØØ 88 46 Ø9 C2 Ø4 ØØ 33 CØ 89 46 Ø6 89 46 Ø8 C2
[DH5]  137: Ø4 ØØ 8B EC 8A 5E Ø5 22 DB 74 ØF BØ 8Ø 84 46 Ø4
[4H5]  138: 74 Ø9 33 CØ 89 46 Ø2 89 46 Ø4 C3 88 46 Ø5 81 7E
[XG5]  139: Ø4 35 8Ø 75 Ø5 81 7E Ø2 F4 Ø4 73 Ø5 FE 46 Ø5 FE
[RP5]  14Ø: CB 32 FF 53 FF 76 Ø4 FF 76 Ø2 B8 8Ø 81 5Ø 33 CØ
[3T5]  141: 5Ø E8 FA FD FF 76 1Ø FF 76 ØE B8 ØØ 81 5Ø 33 CØ
[RS5]  142: 5Ø E8 EA FD E8 1F FF 8B 46 Ø8 89 46 1Ø 5Ø 8B 46
[8N5]  143: Ø6 89 46 ØE 5Ø E8 72 FE 8B 46 Ø8 8B 5E Ø8 53 5Ø
[PT5]  144: 53 5Ø E8 65 FE B8 5E 7F 5Ø B8 4D 57 5Ø E8 BE FD
[2S5]  145: FF 76 ØC FF 76 ØA E8 51 FE B8 13 8Ø 5Ø B8 Ø8 9B
[2Z5]  146: 5Ø E8 AA FD FF 76 ØC FF 76 ØA E8 3D FE B8 76 8Ø
[ES5]  147: 5Ø B8 93 38 5Ø E8 96 FD E8 2F FE B8 38 82 5Ø B8
[DX5]  148: 3B AA 5Ø E8 88 FD FF 76 1Ø FF 76 ØE E8 1B FE 8A
[VH5]  149: 46 ØA B4 ØØ 34 8Ø 74 17 8A D8 79 Ø2 F6 D8 B4 88
[ZT5]  15Ø: 22 CØ 78 Ø6 DØ EØ FE CC EB F6 8Ø CB 7F 22 C3 5Ø
[KM5]  151: 33 CØ 5Ø E8 58 FD B8 31 8Ø 5Ø B8 18 72 5Ø E8 E9
[1R5]  152: FD 8F 46 ØE 8F 46 1Ø 58 C3 8B EC B2 ØØ 8B 76 Ø6
[3V5]  153: 8B 46 Ø8 8B 7E Ø2 8B 5E Ø4 8A C8 8A EB 32 E6 91
[GH5]  154: 19 ØC 8Ø 8Ø CB 8Ø 3B C3 75 Ø4 3B F7 74 ØB 77 Ø3
[PT5]  155: 8Ø F1 8Ø DØ E1 72 Ø7 FE C2 C3 DØ EØ 73 F9 FE CA
[NP5]  156: C3 53 8A 1F 8Ø FB FF 74 4F 8Ø FB Ø5 72 4A 33 CØ
[6W5]  157: 8E CØ 8Ø FB FE 75 1C BA F9 Ø3 EE E4 21 ØC 1Ø E6
[9E5]  158: 21 2E A1 15 Ø1 26 A3 3Ø ØØ 2E A1 17 Ø1 26 A3 32
[FP5]  159: ØØ EB 25 8Ø FB FD 75 1C BA F9 Ø2 EE E4 21 ØC Ø8
[SF5]  16Ø: E6 21 2E A1 19 Ø1 26 A3 2C ØØ 2E A1 1B Ø1 26 A3
[8N5]  161: 2E ØØ EB Ø4 B4 3E CD 21 5B C3 Ø7 B8 38 81 5Ø B8
[8P5]  162: 3B AA 5Ø E8 44 FD 33 D2 33 CØ 8B 5E Ø8 8Ø FF 81
[2L5]  163: 72 24 8A E3 8Ø CC 8Ø 8A 46 Ø7 8A 56 Ø6 8Ø FF 9Ø
[RT5]  164: 73 Ø9 D1 E8 8Ø D6 ØØ FE C7 EB F2 22 DB 79 Ø7 F7
[LL5]  165: D8 23 D2 74 Ø1 48 5Ø FF 76 Ø8 FF 76 Ø6 B7 ØØ 23
[DR5]  166: CØ 74 ØC 79 Ø2 F7 D8 B7 9Ø FE CF D1 EØ 79 FA 8Ø
[ZT5]  167: E3 8Ø 8Ø E4 7F ØA E3 8A DC 8A EØ BØ ØØ 8Ø F3 8Ø
[WH5]  168: 53 5Ø E8 49 FC B8 34 71 5Ø B8 3C 58 5Ø FF 76 Ø8
[6V5]  169: FF 76 Ø6 E8 D4 FC B8 16 74 5Ø B8 B7 7E 5Ø E8 2D
[2Ø6]  17Ø: FC FF 76 ØC FF 76 ØA E8 CØ FC B8 2F 77 5Ø B8 DF
[HØ6]  171: EE 5Ø E8 19 FC FF 76 ØC FF 76 ØA E8 AC FC B8 1D
[CT5]  172: 7A 5Ø B8 17 84 5Ø E8 Ø5 FC FF 76 ØC FF 76 ØA E8
[1V5]  173: 98 FC B8 63 7C 5Ø B8 59 59 5Ø E8 F1 FB FF 76 ØC
[WZ5]  174: FF 76 ØA E8 84 FC B8 75 7E 5Ø B8 E9 FD 5Ø E8 DD
[KS5]  175: FB FF 76 ØC FF 76 ØA E8 7Ø FC B8 31 8Ø 5Ø B8 18
[4W5]  176: 72 5Ø E8 C9 FB FF 76 ØC FF 76 ØA E8 5C FC B8 ØØ
[NM5]  177: 81 5Ø 33 CØ 5Ø E8 B6 FB 58 5B 59 59 59 Ø2 F9 75
[1M5]  178: Ø4 33 CØ 33 DB 89 46 1Ø 89 5E 12 Ø6 C3 8B EC 8B
[WN5]  179: 46 Ø4 8Ø FC 98 73 5Ø 22 CØ 79 1C FF 76 Ø4 FF 76
[EW5]  18Ø: Ø2 B8 FF 8Ø 5Ø B8 FF FF 5Ø E8 82 FB 8B EC 83 C5
[ST5]  181: Ø4 8F 46 Ø2 8F 46 Ø4 B8 FF FF B3 FF B1 9Ø 2A 4E
[WF5]  182: Ø5 73 Ø7 8Ø C1 Ø8 D2 EØ EB 17 BØ ØØ 8Ø E9 Ø8 73
[7L5]  183: Ø7 8Ø C1 Ø8 D2 E4 EB Ø9 B4 ØØ 8Ø F9 Ø8 73 Ø9 D2
[RD5]  184: E3 21 46 Ø2 2Ø 5E Ø4 C3 89 46 Ø2 89 46 Ø4 C3 Ø7
[GL5]  185: B8 49 81 5Ø B8 DA ØF 5Ø E8 33 FB Ø6 Ø7 B8 22 7E
[ØT5]  186: 5Ø B8 81 F9 5Ø E8 C2 FB 8B EC FF 76 Ø2 FF 76 ØØ
[FM5]  187: E8 7A FF 8B EC 8Ø 7E Ø3 ØØ 74 Ø4 8Ø 76 Ø2 8Ø E8
[1H5]  188: ØC FB 8B EC 8Ø 7E Ø3 ØØ 74 Ø4 8Ø 76 Ø2 8Ø B8 ØØ
[8R5]  189: 7F 5Ø 33 CØ 5Ø E8 F6 FA 8B EC F6 46 Ø2 8Ø 74 19
[WM5]  19Ø: B8 ØØ 8Ø 5Ø 33 CØ 5Ø E8 E4 FA 8B EC 8Ø 66 Ø2 7F
[RC5]  191: B8 8Ø 7F 5Ø 33 CØ 5Ø EB 11 8Ø 7E Ø3 ØØ 74 Ø4 8Ø
[6T5]  192: 76 Ø2 8Ø B8 ØØ 7F 5Ø 33 CØ 5Ø E8 C1 FA 8B EC FF
[DS5]  193: 76 Ø2 FF 76 ØØ FF 76 Ø2 FF 76 ØØ E8 4C FB B8 E6
[GV5]  194: 84 5Ø B8 2D 1A 5Ø 8B EC FF 76 Ø6 FF 76 Ø4 E8 39
[DU5]  195: FB B8 28 86 5Ø B8 FB Ø7 5Ø E8 92 FA 8B EC FF 76
[LP5]  196: Ø6 FF 76 Ø4 E8 23 FB B8 99 87 5Ø B8 99 68 5Ø E8
[MY5]  197: 7C FA 8B EC FF 76 Ø6 FF 76 Ø4 E8 ØD FB B8 23 87
[4S5]  198: 5Ø B8 DF 35 5Ø E8 66 FA 8B EC FF 76 Ø6 FF 76 Ø4
[KW5]  199: E8 F7 FA B8 A5 86 5Ø B8 E7 5D 5Ø E8 5Ø FA 8B EC
[BV5]  19Ø: FF 76 Ø6 FF 76 Ø4 E8 E1 FA B8 49 83 5Ø B8 DA ØF
[NR5]  2Ø1: 5Ø E8 3A FA 8B EC 8F 46 Ø4 8F 46 Ø6 E8 CB FA Ø6
[UP5]  2Ø2: C3 BB Ø3 ØØ E8 1A FD 4B 75 FA 8B ØE ØF ØØ B4 Ø1
[SP5]  2Ø3: CD 1Ø B4 4C CD 21 8B EC 33 D2 8B 46 Ø4 3B 46 Ø2
[EH5]  2Ø4: 7E Ø1 4A 89 56 Ø4 C2 Ø2 ØØ 8B EC 33 D2 8B 46 Ø4
[5G5]  2Ø5: 3B 46 Ø2 7D Ø1 4A 89 56 Ø4 C2 Ø2 ØØ 8B EC 33 D2
[XE5]  2Ø6: 8B 46 Ø4 3B 46 Ø2 75 Ø1 4A 89 56 Ø4 C2 Ø2 ØØ 8B
[7L5]  2Ø7: EC 33 D2 8B 46 Ø4 3B 46 Ø2 7C Ø1 4A 89 56 Ø4 C2
[ØH5]  2Ø8: Ø2 ØØ 8B EC 33 D2 8B 46 Ø4 3B 46 Ø2 7F Ø1 4A 89
[NF5]  2Ø9: 56 Ø4 C2 Ø2 ØØ 8B EC 33 D2 8B 46 Ø4 3B 46 Ø2 74
[3C5]  21Ø: Ø1 4A 89 56 Ø4 C2 Ø2 ØØ 8B EC 8B 46 Ø2 Ø1 46 Ø4
[MK5]  211: C2 Ø2 ØØ 8B EC 8B 46 Ø2 29 46 Ø4 C2 Ø2 ØØ 8B EC
[SF5]  212: 8B 46 Ø2 21 46 Ø4 C2 Ø2 ØØ 8B EC 8B 46 Ø2 Ø9 46
[XC5]  213: Ø4 C2 Ø2 ØØ 8B EC 8B 46 Ø2 31 46 Ø4 C2 Ø2 ØØ 8B
[HM5]  214: EC 8B 46 Ø4 F7 66 Ø2 89 46 Ø4 C2 Ø2 ØØ 8B EC 8B
[GL5]  215: 46 Ø4 33 D2 F7, 7E Ø2 89 46 Ø4 C2 Ø2 ØØ 8B EC 8B
[GL5]  216: 46 Ø4 33 D2 F7 7E Ø2 89 56 Ø4 C2 Ø2 ØØ 8B EC F7
[5W5]  217: 56 Ø2 C3 8B EC F7 5E Ø2 C3 8B EC 8B 5E ØØ 8C DA
[ØY5]  218: C5 76 Ø2 81 EC FA ØØ 8B FC 8C DØ 8E CØ FC AC AA
[MT5]  219: 8A C8 B5 ØØ F3 A4 8E DA FF E3 8B F4 83 C6 Ø2 1E
[ØV5]  22Ø: 8C DØ 8E D8 33 FF 36 8E 84 ØØ Ø1 FC AC AA 8A C8
[EP5]  221: B5 ØØ F3 A4 1F C2 Ø2 Ø1 8B EC 8B 5E Ø2 BØ ØD BA
[2T5]  222: A3 Ø1 FF D2 8Ø 3F FF 74 Ø7 BØ ØA BA A3 Ø1 FF D2
[UM5]  223: C3 8B EC 83 C5 Ø2 8B 9E ØØ Ø1 8A 4E ØØ B5 ØØ E3
[2M5]  224: ØD 45 8A 46 ØØ 51 BA A3 Ø1 FF D2 59 E2 F3 C2 ØØ
[AS5]  225: Ø1 81 EC FC ØØ 8B EC 8B 86 FE ØØ B3 2Ø 23 CØ 79
[UN5]  226: Ø4 F7 D8 B3 2D 88 5E Ø1 BB ØA ØØ 33 C9 33 D2 F7
```

```
[UN5]  227: F3 8Ø CA 3Ø 52 23 CØ EØ F4 F7 D9 8A D9 BF Ø2 ØØ
[SP5]  228: 58 88 Ø3 47 E2 FA C6 Ø3 2Ø FE C3 88 5E ØØ FF A6
[NU5]  229: FC ØØ 33 DB BØ ØC BA A3 Ø1 FF E2 8B F4 83 C6 Ø2
[8Y5]  23Ø: 1E 8C DØ 8E D8 8E CØ 8A 8C ØØ Ø1 B5 Ø1 8B F9 AC
[8N5]  231: Ø3 FE B5 ØØ F7 D9 81 C1 FF ØØ 3A C1 73 Ø2 8A C8
[NR5]  232: ØØ 8C FF ØØ F3 A4 1F C2 ØØ Ø1 81 EC FC ØØ 8B EC
[WR5]  233: BØ Ø1 8A A6 FE ØØ 89 46 ØØ FF A6 FC ØØ 8B EC 33
[ND5]  234: D2 39 56 Ø2 74 Ø6 7F Ø3 4A EB Ø1 42 89 56 Ø2 C3
[HL5]  235: 8B EC B4 ØF CD 1Ø 8A 56 Ø2 8A 76 Ø4 81 EA Ø1 Ø1
[XD5]  236: B4 Ø2 CD 1Ø C6 Ø6 ØA ØØ ØØ C2 Ø4 ØØ 8B EC 8A 46
[NK5]  237: Ø4 8A 66 Ø2 B1 Ø4 D2 E4 8Ø E4 7Ø 8A D8 24 ØF ØA
[8F5]  238: C4 B1 Ø3 D2 E3 8Ø E3 8Ø ØA C3 A2 Ø9 ØØ C2 Ø4 ØØ
[BR5]  239: B4 ØF CD 1Ø B4 Ø3 CD 1Ø 8B EC 8A 5E Ø2 8Ø FB 28
[4G5]  24Ø: 75 2Ø 8Ø FC 28 74 14 3C Ø6 74 Ø4 BØ Ø1 EB Ø2 BØ
[SL5]  241: Ø5 FE CB 88 1E ØB ØØ B4 ØØ CD 1Ø B4 Ø1 CD 1Ø C2
[PN5]  242: Ø2 ØØ 8Ø FB 5Ø 75 F4 8Ø FC 5Ø 74 EF 3C Ø4 73 Ø4
[3X5]  243: BØ Ø3 EB DD BØ Ø6 EB D9 8B EC 8A 46 Ø4 FE C8 A2
[KR5]  244: ØC ØØ 8A 46 Ø2 FE C8 A2 ØD ØØ 33 DB BØ ØB BA A3
[9R5]  245: Ø1 FF D2 C2 Ø4 ØØ 8B F4 83 C6 Ø8 8B DE 1E 8C DØ
[6M5]  246: 8E D8 AC B4 ØØ 8B D6 Ø3 FØ 88 24 8Ø 7F Ø5 3A 75
[LR5]  247: 18 8B 47 Ø1 25 DF DF 8B 4F Ø3 81 E1 DF DF 3D 53
[PM5]  248: 43 75 ØE 81 F9 52 4E 74 Ø3 E9 C2 ØØ BØ FF E9 DC
[1L5]  249: ØØ 3D 4B 59 75 Ø8 81 F9 42 44 74 FØ EB EB 3D 43
[HW5]  25Ø: 4F 75 E6 B6 Ø3 81 F9 4D 11 74 Ø8 FE CE 81 F9 4D
[KS5]  251: 12 75 D6 B2 FB BØ 83 EE 8D 77 Ø6 52 33 C9 AC 2C
[1L5]  252: 3Ø 72 12 3C ØA 73 ØE B4 ØØ 8B F8 BØ ØA F7 E1 Ø3
[XT5]  253: C7 8B C8 EB E9 23 C9 75 Ø3 B9 2C Ø1 B8 ØØ C2 BA
[LU5]  254: Ø1 ØØ F7 F1 5A B2 F8 EE 42 8A C4 EE 42 42 BØ Ø3
[7Z5]  255: EE 42 BØ ØB EE B2 F9 BØ Ø1 EE 4A EC 1E 33 CØ 8E
[HM5]  256: D8 B1 FE 8Ø FE Ø3 75 1F 2E A3 1D Ø1 B8 21 Ø1 87
[7F5]  257: Ø6 3Ø ØØ 2E A3 15 Ø1 8C C8 87 Ø6 32 ØØ 2E A3 17
[1P5]  258: Ø1 E4 21 24 EF EB 1F FE C9 2E A3 1F Ø1 B8 51 Ø1
[1F5]  259: 87 Ø6 2C ØØ 2E A3 19 Ø1 8C C8 87 Ø6 2E ØØ 2E A3
[7R5]  26Ø: 1B Ø1 E4 21 24 F7 E6 21 1F EC 8A C1 EB 1F 8A 47
[XF5]  261: FE 3C Ø1 74 12 3C Ø3 75 Ø2 BØ Ø1 B4 3D CD 21 73
[UR5]  262: ØC 8Ø 7F FE ØØ 74 64 B1 2Ø B4 3C CD 21 8A 67 FE
[9T5]  263: 8B 5F FC 1F 88 Ø7 C6 47 Ø4 ØØ C6 47 1Ø 1A 3C FD
[SK5]  264: 73 46 8Ø FC Ø3 75 ØD 8A D8 33 C9 33 D2 B8 Ø2 42
[RH5]  265: CD 21 EB 2C 8Ø FC ØØ 75 ØB C6 47 1Ø ØØ BA 39 Ø3
[7P5]  266: FF D2 EB 1C 8Ø FC Ø2 75 17 53 8A 1F 33 C9 33 D2
[MF5]  267: B8 Ø2 42 CD 21 5B 72 Ø8 ØB C2 74 Ø4 C6 47 1Ø ØØ
[YN5]  268: 8B EC 8A 46 Ø2 88 47 1A C2 Ø6 Ø1 8B 5F FC 1F C6
[PL5]  269: Ø7 ØØ C6 47 Ø4 ØØ C6 47 1Ø 1A C2 Ø6 Ø1 8B EC 8B
[YR5]  27Ø: 5E Ø2 BA 79 Ø8 FF D2 C2 Ø2 ØØ 8B EC 8B 5E Ø2 8Ø
[SL5]  271: 3F FF 75 14 B4 Ø3 CD 1Ø AØ ØB ØØ 2C 17 3A DØ 72
[BR5]  272: ØA BØ ØD BA A3 Ø1 FF E2 8A 57 Ø4 8A C2 B4 ØØ B9
[RR5]  273: ØE ØØ F6 F1 2A CC BØ 2Ø 51 BA A3 Ø1 FF D2 59 E2
[WN5]  274: F5 C3 58 8B EC 8E 46 ØØ 89 46 ØØ BF Ø1 ØØ 8B 5E
[7L5]  275: Ø2 C6 Ø6 ØE ØØ ØØ 57 BA 39 Ø3 FF D2 5F 3C ØØ 74
[TM5]  276: F5 3C 1A 74 1F 3C ØD 74 Ø9 81 FF ØØ Ø1 73 E7 AA
[ØP5]  277: EB E4 8Ø 3F FF 74 ØD 8Ø 7F 1Ø ØA 75 Ø7 57 BA 39
[CU5]  278: Ø3 FF D2 5F 8D 45 FF 26 A2 ØØ ØØ BA 39 Ø5 FF E2
[XV5]  279: 33 DB BØ 3F BA A3 Ø1 FF D2 BØ 2Ø BA A3 Ø1 FF E2
[HX5]  28Ø: 81 EC FE ØØ 8B EC FF B6 FE ØØ 8C DØ 8E CØ 8D 7E
[8F5]  281: Ø1 8B 9E Ø2 Ø1 8Ø 26 ØE ØØ 7F E8 54 ØØ 74 49 3C
[HL5]  282: 2Ø 74 F7 3C 22 75 21 E8 47 ØØ 74 3C 3C 22 74 ØD
[2U5]  283: 57 2B FD 81 FF ØØ Ø1 5F 73 ED AA EB EA E8 31 ØØ
[XL5]  284: 74 26 3C 2C 75 F7 EB 2Ø 3C 2C 74 1Ø 57 2B FD 81
[8U5]  285: FF ØØ Ø1 5F 73 Ø1 AA E8 17 ØØ 75 EC 4F 3B FD 74
[AM5]  286: Ø6 26 8Ø 3D 2Ø 74 F5 47 8D 45 FF 2B C5 88 46 ØØ
[MP5]  287: C3 57 BA 39 Ø3 FF D2 5F 3C 1A 74 15 3C ØD 75 11
[FP5]  288: 8Ø 7F 1Ø ØA 75 Ø9 57 BA 39 Ø3 FF D2 5F BØ ØD 3C
[MK5]  289: ØD C3 8B EC 8A 46 Ø2 B4 ØØ 89 86 ØØ Ø1 C2 FE ØØ
[HX5]  29Ø: BB Ø4 ØØ BA 79 Ø8 FF D2 4B 75 F8 C3 8B EC 8B 5E
[FG5]  291: Ø2 BØ FF 38 Ø7 74 Ø6 8Ø 7F 1Ø 1A 74 Ø2 F6 DØ 98
[YV5]  292: 89 46 Ø2 C3 81 EC FE ØØ 8B EC B4 ØB CD 21 B4 FF
[ØG5]  293: 3C ØØ 74 Ø8 B4 Ø8 CD 21 8A EØ BØ Ø1 89 46 ØØ 8Ø
[RR5]  294: FC ØØ 75 ØA B4 Ø8 CD 21 88 46 Ø2 FE 46 ØØ FF A6
[NL5]  295: FE ØØ BA 54 Ø5 FF D2 33 D2 A8 Ø1 74 Ø1 4A 89 96
[FP5]  296: FE Ø1 C2 FE Ø1 BA 54 Ø5 FF D2 33 D2 A8 Ø2 74 Ø1
[ZU5]  297: 4A 89 96 FE Ø1 C2 FE Ø1 BA 54 Ø5 FF D2 33 D2 A8
[PN5]  298: Ø4 74 Ø1 4A 89 96 FE Ø1 C2 FE Ø1 BA 54 Ø5 FF D2
[3L5]  299: 33 D2 A8 Ø3 74 Ø1 4A 89 96 FE Ø1 C2 FE Ø1 BA 54
[7R5]  3ØØ: Ø5 FF D2 33 D2 A8 Ø5 74 Ø1 4A 89 96 FE Ø1 C2 FE
[KM5]  3Ø1: Ø1 BA 54 Ø5 FF D2 33 D2 A8 Ø1 75 Ø1 4A 89 96 FE
[UL5]  3Ø2: Ø1 C2 FE Ø1 5E 5B 59 81 EC ØØ Ø1 8B EC 88 4E ØØ
[HN5]  3Ø3: E3 16 C6 Ø6 ØE ØØ FF 45 51 BA 39 Ø3 FF D2 59 3C
[NU5]  3Ø4: 1A 74 Ø5 88 46 ØØ E2 EF 8B EC 28 4E ØØ FF E6 8B
[UV5]  3Ø5: EC 8B 5E Ø2 BA 39 Ø5 FF E2 8B EC 8A 8E Ø2 Ø1 8A
[8R5]  3Ø6: 6E Ø2 8Ø FD ØØ 74 2B 8B 9E Ø2 Ø2 8A C3 FE C8 Ø2
[EK5]  3Ø7: C5 72 1F 3A C1 77 1B 8B F3 BF Ø1 ØØ 8A D5 46 47
[YF5]  3Ø8: 8A 82 Ø1 Ø1 3A 43 Ø1 75 Ø6 FE CA 75 F1 EB Ø5 43
[US5]  3Ø9: EB D9 33 DB 89 9E Ø2 Ø2 C2 ØØ Ø2 8B F4 8C DA 8C
[H16]  31Ø: DØ 8E D8 AD AD 8E CØ 33 FF FC AC AA 8A C8 B5 ØØ
[YL5]  311: F3 A4 8E DA C2 Ø2 Ø1 8B EC 8B 46 Ø2 87 46 Ø4 83
[PX5]  312: C4 Ø4 FF EØ 8B F4 83 C6 Ø2 8C DA 8C DØ 8E D8 8E
[8N5]  313: CØ 8B 9C ØØ Ø1 8A ØC B5 ØØ Ø3 F1 8D 7C Ø2 41 FD
[WV5]  314: F3 A4 FC 8E DA 83 C4 Ø4 FF E3 8B EC 8E Ø6 19 ØØ
[WM5]  315: 8B 7E Ø4 8A 46 Ø2 AA C2 Ø4 ØØ 8B EC 8E Ø6 19 ØØ
[5N5]  316: 8B 7E Ø2 26 8A Ø5 B4 ØØ 89 46 Ø2 C3 8B EC 8B 4E
[1L5]  317: Ø2 33 CØ 8Ø F9 Ø1 72 Ø2 8A C5 89 86 ØØ Ø1 C2 FE
[4N5]  318: ØØ 8B EC 8A 46 Ø4 8A 66 Ø2 3A EØ 73 Ø2 8A C4 88
[PN5]  319: 46 Ø4 C2 Ø2 ØØ 8B FC 83 C7 Ø4 36 8A 4D FE 36 8A
[SS5]  32Ø: 1D 2A D9 76 16 B7 ØØ 8D 71 Ø1 8C DA 8C DØ 8E D8
[KU5]  321: 8E CØ B5 ØØ 8A C1 AA F3 A4 8E DA C2 Ø2 ØØ 8B FC
[LØ6]  322: 83 C7 Ø6 8C DA 8C DØ 8E D8 8E CØ 8A 5D FE 8Ø FB
[TH5]  323: ØØ 75 Ø2 B3 Ø1 B7 ØØ 8D 31 8A ØD B5 ØØ FE CB 2A
[CN5]  324: CB 73 Ø2 B1 ØØ 8A 45 FC 3A C8 76 Ø2 8A C8 8A C1
[BV5]  325: AA F3 A4 8E DA C2 Ø4 ØØ 8B FC 83 C7 Ø4 8C DØ 8E
[UN5]  326: CØ 26 8A 8D FE ØØ B5 ØØ 26 8B 45 FE 3C ØØ 75 Ø2
[RP5]  327: B1 ØØ 8A C1 AA 8A C4 F3 AA C2 Ø2 ØØ 8B EC 8B 46
[MS5]  328: Ø4 8E CØ 8C DB 2B C3 B1 Ø4 D3 EØ 4Ø 8B 5E Ø2 89
[6F5]  329: 47 Ø1 26 AØ ØØ ØØ 88 Ø7 89 5E Ø4 C2 Ø2 ØØ 8B EC
[6S5]  33Ø: 8C C8 87 46 Ø2 8C DB 8E C3 FF 36 19 ØØ 5Ø CB 8B
[5P5]  331: F4 83 C6 Ø2 8C DA 8C DØ 8E D8 8A 8C ØØ Ø1 B5 ØØ
[XP5]  332: AC 3A C1 73 Ø2 8A C8 8E 84 Ø3 Ø1 8A 9C Ø1 Ø1 B7
[EH5]  333: ØØ 8Ø FB ØØ 77 Ø1 43 8B FB 26 AØ ØØ ØØ 2A C3 72
[5T5]  334: ØA FE CØ 3A C8 76 Ø2 8A C8 F3 A4 8E DA C2 Ø6 Ø1
[4T5]  335: 81 EC FE ØØ 8B EC B4 2C CD 21 C6 46 ØØ Ø8 8A C5
[YH5]  336: E8 1F ØØ 89 46 Ø1 C6 46 Ø3 3A 8A C1 E8 13 ØØ 89
[YM5]  337: 46 Ø4 C6 46 Ø6 3A 8A C6 E8 Ø7 ØØ 89 46 Ø7 FF A6
[BP5]  338: FE ØØ B4 ØØ B3 ØA F6 F3 Ø5 3Ø 3Ø C3 8B EC 8A 46
```

```
[9N5]  339: Ø2 8B 76 Ø4 8B 5E Ø6 88 Ø4 D1 E3 89 77 1C AC 8A
[CP5]  34Ø: DØ AD B4 ØØ 8B C8 AD 8E CØ 26 88 ØE ØØ ØØ BF Ø1
[SP5]  341: ØØ BØ 2Ø F3 AA FE CA 75 E8 C2 Ø6 ØØ 8B EC 8B 5E
[FR5]  342: Ø6 D1 E3 8B 77 1C D1 EB 8A 47 1A B4 ØØ 5Ø 8B 56
[MM5]  343: Ø2 8B 4E Ø4 8Ø C9 8Ø 8Ø FD 98 73 Ø8 DØ E9 D1 DA
[XR5]  344: FE C5 EB F3 83 EA Ø1 8Ø D9 ØØ F6 E1 8B C8 58 5Ø
[SP5]  345: F7 E2 Ø3 D1 8B DØ 8B FB 8A 1F B8 ØØ 42 CD 21 59
[VT5]  346: 57 Ø3 C1 83 D2 ØØ 8E CØ 8B EA AC B4 ØØ 8B F8 AD
[PR5]  347: 8B C8 1E AD 8E D8 88 ØE ØØ ØØ BA Ø1 ØØ B4 3F CD
[UL5]  348: 21 1F 72 24 3B C1 75 2Ø 4F 75 E4 33 C9 33 D2 B8
[HK5]  349: Ø2 42 CD 21 72 12 3B EA 75 Ø6 8C C6 3B FØ 74 Ø8
[1A5]  35Ø: 5B C6 47 1Ø ØØ C2 Ø6 ØØ 5B C6 47 1Ø 1A C2 Ø6 ØØ
[UV5]  351: 8B F4 83 C6 Ø2 8C DA 8C DØ 8E D8 8E 84 ØØ Ø1 BF
[KF5]  352: Ø1 ØØ 26 8A ØE ØØ ØØ B5 ØØ B3 ØØ AC 3A C1 73 Ø6
[7X5]  353: 8A D9 2A D8 8A C8 F3 A4 8A CB BØ 2Ø F3 AA 8E DA
[FP5]  354: C2 Ø2 Ø1 8B F4 83 C6 Ø2 8C DA 8C DØ 8E D8 8E 84
[KF5]  355: ØØ Ø1 BF Ø1 ØØ 26 8A 1E ØØ ØØ B5 ØØ B1 ØØ AC 3A
[1W5]  356: C3 73 Ø6 8A CB 2A C8 8A D8 BØ 2Ø F3 AA 8A CB F3
[LT5]  357: A4 8E DA C2 Ø2 Ø1 8B EC 8B 5E Ø6 D1 E3 8B 77 1C
[ØM5]  358: D1 EB 8A 47 1A B4 ØØ 5Ø 8B 56 Ø2 8B 4E Ø4 8Ø C9
[PX5]  359: 8Ø 8Ø FD 98 73 Ø8 DØ E9 D1 DA FE C5 EB F3 83 EA
[FP5]  36Ø: Ø1 8Ø D9 ØØ F6 E1 8B C8 58 F7 E2 Ø3 D1 8B DØ 8A
[HV5]  361: 1F B8 ØØ 42 CD 21 AC B4 ØØ 8B F8 AD 8B C8 1E AD
[FK5]  362: 8E D8 BA Ø1 ØØ B4 4Ø CD 21 1F 72 Ø7 3B C1 75 Ø3
[7R5]  363: 4F 75 E8 C2 Ø6 ØØ 8B EC 8B 7E ØØ 8B 46 Ø2 B3 ØØ
[PP5]  364: 23 CØ 74 15 BB 9Ø 7F 79 Ø4 F7 D8 B7 FF 23 CØ 78
[VM5]  365: Ø6 D1 EØ FE CB EB F6 22 E7 C6 46 ØØ ØØ 89 46 Ø1
[9V5]  366: 88 5E Ø3 FF E7 81 EC FA ØØ 8B FC B2 ØØ 36 8A B5
[PR5]  367: FC ØØ 36 8B B5 FD ØØ 36 8A 9D FF ØØ 8C DØ 8E CØ
[4H5]  368: 47 BØ 2Ø 23 F6 79 Ø2 BØ 2D AA 8Ø FB ØØ 75 14 BØ
[4V5]  369: 3Ø AA 8B EC 8D 45 FF 2B C5 88 46 ØØ BØ 2Ø AA FF
[8M5]  37Ø: A6 FA ØØ 81 CE ØØ 8Ø 33 C9 8Ø FB 81 72 Ø3 E9 D1
[TK5]  371: ØØ E8 7A ØØ FE C9 8Ø FB 8Ø 72 F6 81 C2 8Ø ØØ 83
[XT5]  372: D6 ØØ 73 Ø6 D1 DE D1 DA FE C3 8Ø FB 81 73 Ø5 E8
[BU5]  373: 5C ØØ FE C9 E8 86 ØØ 2A E9 8Ø FD Ø7 77 33 8B EC
[YF5]  374: 8B 46 Ø2 86 C4 89 46 Ø2 83 C5 Ø3 3B FD 75 Ø1 47
[DN5]  375: 8B F7 F6 D1 B5 ØØ Ø3 F9 57 4E 4F 36 8A Ø4 36 88
[GN5]  376: Ø5 3B F5 77 F4 E3 Ø7 4F 36 C6 Ø5 3Ø E2 F9 5F EB
[GT5]  377: 81 BØ 45 AA BØ 2B 22 C9 79 Ø4 F6 D9 BØ 2D AA 8A
[AR5]  378: C1 B4 ØØ B1 ØA F6 F1 ØD 3Ø 3Ø AB E9 64 FF 8B C2
[8V5]  379: 8B EE D1 ED D1 D8 D1 ED D1 D8 13 DØ 13 F5 73 Ø6
[PU5]  38Ø: D1 DE D1 DA FE C3 8Ø C3 Ø3 C3 BØ ØØ D1 E2 D1 D6
[PY5]  381: DØ DØ FE CB 8Ø FB 8Ø 77 F3 ØC 3Ø AA C3 E8 EA FF
[UZ5]  382: BØ 2E AA E8 C8 FF E8 E1 FF FE C5 8Ø FD Ø6 72 F3
[TT5]  383: 26 8A 45 FF 3C 3Ø 75 Ø5 FE CD 4F EB F3 3C 2E 74
[CV5]  384: F9 C3 E8 4B ØØ FE C1 8Ø FB 81 73 F6 FE C9 81 C2
[HU5]  385: 8Ø ØØ 83 D6 ØØ 73 Ø6 D1 DE D1 DA FE C3 E8 8E FF
[EU5]  386: E8 BA FF 8Ø F9 Ø7 72 Ø3 E9 66 FF 8B EC 83 C5 Ø3
[YN5]  387: FE C9 78 ØB 8B 46 ØØ 86 C4 89 46 ØØ 45 EB F1 3B
[7Y5]  388: FD 73 Ø2 8B FD 36 8Ø 7D FF 2E 75 Ø1 4F E9 C2 FE
[EM5]  389: 52 BD ØA ØØ 8B C6 33 D2 F7 F5 8B FØ 58 F7 F5 92
[HW5]  39Ø: B4 1A F6 E4 DØ EØ D1 D2 D1 D6 FE CB 23 F6 79 F4
[YR5]  391: C3 8B EC 8Ø 76 Ø4 8Ø C3 BA 86 Ø5 FF E2 83 EC Ø2
[8F5]  392: 8B EC 8B 46 Ø6 87 46 Ø4 87 46 Ø2 89 46 ØØ 8B 46
[6N5]  393: Ø8 B3 ØØ 23 CØ 74 15 BB 9Ø 7F 79 Ø4 F7 D8 B7 FF
[ZM5]  394: 23 CØ 78 Ø6 D1 EØ FE CB EB F6 22 E7 C6 46 Ø6 ØØ
[WV5]  395: 89 46 Ø7 88 5E Ø9 C3 BA 22 Ø6 FF E2 BA BE Ø6 FF
[PM5]  396: E2 BA 4A Ø7 FF E2 8B EC 8B 46 Ø8 23 CØ 74 2E 5Ø
[UX5]  397: 24 7F 5Ø FF 76 Ø6 BA 4A Ø7 FF D2 8B EC FF 76 ØA
[SU5]  398: FF 76 Ø8 BA 22 Ø6 FF D2 BA D2 Ø8 FF D2 58 5B 59
[2L5]  399: 8Ø E1 8Ø ØA D9 8B EC 89 46 Ø6 89 5E Ø8 C2 Ø4 ØØ
[XK5]  4ØØ: 8B EC 8B 5E Ø4 4B 78 15 D1 E3 Ø3 5E Ø2 8B 56 Ø8
[8G5]  4Ø1: 83 EA Ø3 3B DA 73 Ø6 2E 8B Ø7 89 46 ØØ C2 Ø4 ØØ
[HM5]  4Ø2: 8B EC 8B 5E Ø4 4B 78 1B D1 E3 Ø3 5E Ø2 8B 56 ØØ
[SH5]  4Ø3: 83 EA Ø3 3B DA 73 ØC 2E 8B Ø7 87 46 ØØ 89 46 Ø4
[DP5]  4Ø4: C2 Ø2 ØØ C2 Ø4 ØØ BA E5 Ø9 FF E2 8B EC 8B 46 Ø3
[LR5]  4Ø5: 8A 6E Ø5 22 ED 74 16 8A DC 8Ø CC 8Ø B1 9Ø 2A CD
[ØK5]  4Ø6: 72 ØB D3 E8 15 ØØ ØØ 22 DB 79 Ø2 F7 D8 89 46 Ø4
[CK5]  4Ø7: C2 Ø2 ØØ Ø7 BA 41 Ø8 FF D2 33 CØ 8Ø FA Ø1 75 Ø1
[7E5]  4Ø8: 48 89 46 Ø8 Ø6 C2 Ø6 ØØ Ø7 BA 41 Ø8 FF D2 33 CØ
[4H5]  4Ø9: 8Ø FA FF 75 Ø1 48 89 46 Ø8 Ø6 C2 Ø6 ØØ Ø7 BA 41
[WG5]  41Ø: Ø8 FF D2 33 CØ 8Ø FA ØØ 75 Ø1 48 89 46 Ø8 Ø6 C2
[KN5]  411: Ø6 ØØ Ø7 BA 41 Ø8 FF D2 33 CØ 8Ø FA FF 74 Ø1 48
[9H5]  412: 89 46 Ø8 Ø6 C2 Ø6 ØØ Ø7 BA 41 Ø8 FF D2 33 CØ 8Ø
[YB5]  413: FA Ø1 74 Ø1 48 89 46 Ø8 Ø6 C2 Ø6 ØØ Ø7 BA 41 Ø8
[9G5]  414: FF D2 33 CØ 8Ø FA ØØ 74 Ø1 48 89 46 Ø8 Ø6 C2 Ø6
[YK5]  415: ØØ 8B EC 8B 46 Ø2 87 46 Ø4 87 46 Ø6 83 C4 Ø4 FF
[3G5]  416: EØ 8B EC 33 D2 8B 46 Ø4 8Ø FC ØØ 74 Ø7 42 22 CØ
[HN5]  417: 79 Ø2 F7 DA 89 56 Ø4 C2 Ø2 ØØ 8B EC 8A 4E Ø2 22
[WM5]  418: C9 74 Ø2 FE C9 8B 5E Ø4 8Ø 3F FF 74 25 38 4F Ø4
[2T5]  419: 76 1Ø 51 BØ ØD BA A3 Ø1 FF D2 BØ ØA A3 Ø1 FF
[3L5]  42Ø: D2 59 38 4F Ø4 73 42 51 BØ 2Ø BA A3 Ø1 FF D2 59
[8W5]  421: EB FØ B4 ØF CD 1Ø 2A CC 73 FC Ø2 CC 51 B4 Ø3 CD
[MM5]  422: 1Ø 59 51 3A D1 76 Ø9 33 DB BØ ØD BA A3 Ø1 FF D2
[7P5]  423: B4 ØF CD 1Ø B4 Ø3 CD 1Ø 59 3A D1 73 ØC 51 33 DB
[7T5]  424: BØ 2Ø BA A3 Ø1 FF D2 EB E7 C2 Ø2 ØØ 8B EC 8B 4E
[YM5]  425: Ø2 83 F9 ØØ 76 ØE 8B 5E Ø4 51 BØ 2Ø BA A3 Ø1 FF
[5T5]  426: D2 59 E2 F5 C2 Ø2 ØØ 8B EC B4 ØF CD 1Ø B4 Ø3 CD
[US5]  427: 1Ø 8A D6 B6 ØØ 42 87 56 ØØ FF E2 8B EC B4 ØF CD
[BN5]  428: 1Ø B4 Ø3 CD 1Ø B6 ØØ 42 87 56 ØØ FF E2 8B EC 8A
[SE5]  429: 4E Ø2 8A 6E Ø4 89 ØE 24 ØØ 83 7E Ø6 ØØ 75 Ø3 B9
[6P5]  43Ø: ØØ 1F B4 Ø1 CD 1Ø C2 Ø6 ØØ 8B EC B4 ØF CD 1Ø B4
[EN5]  431: Ø3 CD 1Ø 81 E9 ØØ 1F 87 4E ØØ FF E1 8B EC AØ 25
[YG5]  432: ØØ B4 ØØ 87 46 ØØ FF EØ 8B EC AØ 24 ØØ B4 ØØ 87
[7P5]  433: 46 ØØ FF EØ 8B EC AØ Ø9 ØØ 8A D8 25 ØF ØØ B1 Ø4
[9S5]  434: D2 EB 8Ø E3 1Ø ØA C3 87 46 ØØ FF EØ 8B EC AØ Ø9
[4H5]  435: ØØ B1 Ø4 D2 E8 25 Ø7 ØØ 87 46 ØØ FF EØ 1E 8C DØ
[ZP5]  436: 8E D8 8B F4 83 C6 Ø4 AC 8A C8 B5 ØØ 33 DB 33 D2
[4M5]  437: 33 ED E3 4D E8 6Ø Ø1 74 48 3C 2B 74 F7 3C 2D 75
[PN5]  438: Ø4 F7 D5 EB EF 3C 26 75 6A E8 4B Ø1 74 33 3C 68
[BH5]  439: 74 Ø4 3C 48 75 2B E8 3E Ø1 74 26 2C 3Ø 72 22 3C
[BD5]  44Ø: ØA 72 12 2C Ø7 3C ØA 72 18 3C 1Ø 72 Ø8 2C 2Ø 72
[WR5]  441: 1Ø 3C 1Ø 73 ØC D1 E2 D1 E2 D1 E2 D1 E2 Ø2 DØ EB
[5S5]  442: D5 23 D2 74 1C B3 9Ø 79 Ø4 F7 DA F7 D5 22 F6 78
[NY5]  443: Ø6 D1 E2 FE CB EB F6 8Ø E6 7F 23 ED 74 Ø3 8Ø CE
[DL5]  444: 8Ø 8B EC 88 BE ØØ Ø1 89 96 Ø1 Ø1 88 9E Ø3 Ø1 1F
[GN5]  445: C2 FC ØØ 55 33 FF 3C 2E 75 53 23 FF 75 Ø6 4F E8
[5H5]  446: D5 ØØ 75 F2 23 FF 74 Ø1 47 23 FF 74 2A 57 53 52
[CH5]  447: 78 12 B8 2Ø 84 5Ø 33 CØ 5Ø BA 22 Ø6 FF D2 5A 5B
[WW5]  448: 5F 4F EB E5 B8 4C 7D 5Ø B8 CD CC 5Ø BA 22 Ø6 FF
[HM5]  449: D2 5A 5B 5F 47 EB D2 58 23 CØ 74 Ø3 8Ø F3 8Ø 8B
[WK5]  45Ø: EC 89 96 ØØ Ø1 89 9E Ø2 Ø1 1F C2 FC ØØ 3C 65 74
```

```
[4N5]  451: Ø4 3C 45 75 42 23 FF 74 Ø1 47 33 ED 8E C5 E8 76
[EU5]  452: ØØ 74 A6 3C 2B 74 F7 3C 2D 75 Ø8 8C CØ F7 DØ 8E
[2R5]  453: CØ EB EB 2C 3Ø 72 1Ø 3C ØA 73 ØC 95 B4 ØA F6 E4
[YM5]  454: Ø3 E8 E8 52 ØØ 75 EC 8B C5 98 8C C5 23 ED 79 Ø2
[9R5]  455: F7 D8 Ø3 F8 E9 72 FF B4 ØØ 2C 3Ø 73 Ø3 E9 64 FF
[LU5]  456: 74 ØE 3C ØA 73 F7 B4 88 FE CC DØ EØ 79 FA 24 7F
[AA5]  457: 51 56 57 5Ø 33 CØ 5Ø B8 2Ø 84 5Ø 33 CØ 5Ø 53 52
[5T5]  458: BA 22 Ø6 FF D2 BA 86 Ø5 FF D2 5A 5B 5F 5E 59 23
[1V5]  459: FF 74 Ø1 4F E9 28 FF 23 C9 74 Ø5 AC 3C 2Ø E1 F7
[YS5]  46Ø: C3 8B EC 8B 46 Ø5 8A 6E Ø7 22 ED 74 16 8A DC 8Ø
[3M5]  461: CC 8Ø B1 9Ø 2A CD 72 ØB D3 E8 15 ØØ ØØ 22 DB 79
[GF5]  462: Ø2 F7 D8 89 46 Ø6 8B 46 Ø2 89 46 Ø4 C2 Ø2 ØØ 8B
[RM5]  463: EC 33 D2 8Ø 7E Ø2 ØØ 74 Ø1 4A 89 96 ØØ Ø1 C2 FE
[2S5]  464: ØØ 81 EC FC ØØ 8B EC 8B 86 FE ØØ 89 46 Ø1 C6 46
[FW5]  465: ØØ Ø2 FF A6 FC ØØ 81 EC FA ØØ 8B EC 8B 86 FC ØØ
[HL5]  466: 89 46 Ø1 8B 86 FE ØØ 89 46 Ø3 C6 46 ØØ Ø4 FF A6
[1R5]  467: FA ØØ 8B EC 8B 46 Ø4 22 CØ 79 ØA 8B 5E Ø2 89 1E
[KE5]  468: 26 ØØ A3 28 ØØ FF 36 28 ØØ FF 36 26 ØØ 22 E4 74
[5P5]  469: 4B B8 A7 98 5Ø B8 EØ 18 5Ø BA 22 Ø6 FF D2 B8 28
[GR5]  47Ø: 68 5Ø B8 46 B1 5Ø BA 86 Ø5 FF D2 8B 46 Ø6 8B 5E
[BL5]  471: Ø8 34 AA 8Ø F3 55 86 C3 8Ø F4 C3 B7 ØØ 23 CØ 75
[GN5]  472: Ø4 22 DB 74 11 B7 8Ø 22 DB 78 Ø8 D1 EØ DØ D3 FE
[PR5]  473: CF EB F4 8Ø E3 7F 89 46 Ø6 89 5E Ø8 8B EC 58 89
[MG5]  474: 46 Ø6 A3 26 ØØ 58 89 46 Ø8 A3 28 ØØ C3 8B EC B4
[DK5]  475: ØF CD 1Ø B4 Ø3 CD 1Ø 8B F2 8A 56 Ø4 8A 76 Ø6 81
[VL5]  476: EA Ø1 Ø1 B4 Ø2 CD 1Ø B4 Ø8 CD 1Ø 8B 4E Ø2 23 C9
[NK5]  477: 74 Ø2 8A C4 B4 ØØ 89 46 Ø6 8B D6 B4 Ø2 CD 1Ø C2
[5V5]  478: Ø4 ØØ 81 EC FE ØØ 8B EC B4 2A CD 21 8A C6 E8 38
[XF5]  479: ØØ 89 46 Ø1 C6 46 Ø3 2D 8A C2 E8 2C ØØ 89 46 Ø4
[HG5]  48Ø: C6 46 Ø6 2D B8 31 39 81 E9 6C Ø7 8Ø F9 64 72 Ø6
[NK5]  481: B8 32 3Ø 8Ø E9 64 89 46 Ø7 8A C1 E8 ØB ØØ 89 46
[5N5]  482: Ø9 C6 46 ØØ ØA FF A6 FE ØØ B4 ØØ B3 ØA F6 F3 Ø5
[ZS5]  483: 3Ø 3Ø C3 83 EC Ø2 B4 ØØ CD 1A 8A DA 8A D6 8A F1
[5F5]  484: 8A CD B5 ØØ 22 DB 75 Ø8 23 D2 75 Ø4 23 C9 74 13
[RW5]  485: B5 AØ 22 C9 78 ØA DØ E3 D1 D2 DØ D1 FE CD EB F2
[PG5]  486: 8Ø E1 7F 51 52 B8 6Ø 7C 5Ø B8 7A F9 5Ø BA 22 Ø6
[KU5]  487: FF D2 8F 46 ØA 58 87 46 ØC FF EØ 8B EC 33 CØ 8Ø
[2E5]  488: 7E Ø2 Ø2 75 Ø3 8B 46 Ø3 89 86 ØØ Ø1 C2 FE ØØ 8B
[4M5]  489: EC 33 CØ 33 DB 8Ø 7E Ø2 Ø4 75 Ø6 8B 46 Ø3 8B 5E
[7N5]  49Ø: Ø5 89 86 FE ØØ 89 9E ØØ Ø1 C2 FC ØØ 8B EC 8Ø 7E
[UP5]  491: Ø5 98 73 3Ø B8 FF FF B3 FF B1 9Ø 2A 4E Ø5 73 Ø7
[WF5]  492: 8Ø C1 Ø8 D2 EØ EB 17 BØ ØØ 8Ø E9 Ø8 73 Ø7 8Ø C1
[CK5]  493: Ø8 D2 E4 EB Ø9 B4 ØØ 8Ø F9 Ø8 73 Ø9 D2 E3 21 46
[5E5]  494: Ø2 2Ø 5E Ø4 C3 89 46 Ø2 89 46 Ø4 C3 81 EC FC ØØ
[2P5]  495: 8B EC 8B 86 FE ØØ BB 1Ø ØØ 33 C9 33 D2 F7 F3 52
[BL5]  496: 23 CØ EØ F7 F7 D9 88 4E ØØ BF Ø1 ØØ 58 Ø4 3Ø 3C
[3N5]  497: 39 76 Ø2 Ø4 Ø7 88 Ø3 47 E2 F2 FF A6 FC ØØ 8B EC
[NN5]  498: 8B 56 Ø2 EC B4 ØØ 89 46 Ø2 C3 8B EC 83 C5 Ø2 8B
[8G5]  499: 9E ØØ Ø1 8A 4E ØØ B5 ØØ E3 11 45 8A 46 ØØ 3C 2Ø
[YS5]  49Ø: 74 Ø7 51 BA A3 Ø1 FF D2 59 E2 EF C2 ØØ Ø1 8B EC
[ØN5]  491: 83 C5 Ø2 8B 9E ØØ Ø1 BØ 22 BA A3 Ø1 FF D2 8A 4E
[FK5]  492: ØØ B5 ØØ E3 ØD 45 8A 46 ØØ 51 BA A3 Ø1 FF D2 59
[AR5]  493: E2 F3 BØ 22 BA A3 Ø1 FF D2 C2 ØØ Ø1 8B EC 8B 5E
[9R5]  494: Ø2 BØ 2C BA A3 Ø1 FF E2 8B EC 8A 5E Ø2 8A 4E Ø4
[LM5]  495: 8B 56 Ø6 EC 32 C3 22 C1 74 F9 C2 Ø6 ØØ 8B EC 83
[CS5]  496: C5 Ø3 8A 4E FF B5 ØØ E8 15 ØØ 8A FE E8 1Ø ØØ 8A
[1N5]  497: DE E8 ØB ØØ B2 32 8B CB B4 2D CD 21 C2 ØØ Ø1 B6
[GD5]  498: ØØ E8 14 ØØ 2C 3Ø 72 19 3C ØA 73 15 5Ø BØ ØA F6
[MN5]  499: E6 8A FØ 58 Ø2 FØ EB E9 BØ ØØ E3 Ø5 8A 46 ØØ 45
[LM5]  49Ø: 49 C3 8B EC 8B 76 Ø2 8B 7E Ø4 8B Ø4 87 Ø5 89 Ø4
[YP5]  511: C2 Ø4 ØØ 8B EC 8C D8 8E CØ 8B 76 Ø2 8B 7E Ø4 26
[CL5]  512: 8B Ø5 A5 89 44 FE 26 8B Ø5 A5 89 44 FE C2 Ø4 ØØ
[HS5]  513: 8B EC 8C DA 8E 5E Ø2 8E 46 Ø4 8A ØE ØØ ØØ 26 8A
[5N5]  514: 2E ØØ ØØ 3A CD 73 Ø2 8A CD B5 ØØ 41 33 F6 33 FF
[FT5]  515: 26 8A Ø5 A4 88 44 FF E2 F7 8E DA C2 Ø4 ØØ 8B EC
[MT5]  516: B8 11 95 5Ø B8 FØ A6 5Ø FF 76 Ø8 FF 76 Ø6 BA BE
[ER5]  517: Ø6 FF D2 58 5B 22 FF 74 12 8Ø CB 8Ø 8A C4 8A E3
[TY5]  518: 8Ø FF 9Ø 73 Ø6 D1 E8 FE C7 EB F5 8B 5E ØE 22 FF
[LN5]  519: 74 13 8B 4E ØC 8Ø CB 8Ø 8Ø FF 9Ø 73 Ø8 DØ EB D1
[2S5]  52Ø: D9 FE C7 EB F3 22 DB 75 Ø2 E3 2C 5Ø BØ B6 E6 43
[EL5]  521: BØ 14 E6 43 BØ ØØ E6 4Ø 58 E6 42 8A C4 E6 42 E4
[ZH5]  522: 61 ØC Ø3 E6 61 E4 4Ø 22 CØ 79 FA E4 4Ø 22 CØ 78
[VY5]  523: FA E2 F2 FE CB 79 EE E4 61 24 FD E6 61 BØ 36 E6
[ØP5]  524: 43 BØ ØØ E6 4Ø E6 4Ø C2 Ø8 ØØ 81 EC FE ØØ 8B EC
[VH5]  525: C6 46 ØØ ØØ FF A6 FE ØØ 8B ØE ØF ØØ B4 Ø1 CD 1Ø
[WK5]  526: 83 EC Ø2 8B EC 8B 76 Ø2 8A 46 Ø4 B4 ØD 89 46 Ø1
[MD5]  527: C6 46 Ø3 ØØ 22 CØ 74 1E Ø4 Ø3 B4 2Ø 89 46 Ø1 C7
[RE5]  528: 46 Ø3 2F 43 B4 ØØ 8B F8 81 FF ØØ Ø1 76 Ø3 BF ØØ
[LG5]  529: Ø1 C7 43 Ø2 ØD ØØ 1E 56 16 45 55 8E 1E 2A ØØ 8E
[YG5]  53Ø: 1E 2C ØØ 1E 33 F6 E8 Ø8 ØØ 43 4F 4D 53 5Ø 45 43
[ZV5]  531: 1D 5B 8B D6 AC 22 CØ 74 1B 8B FB B9 Ø8 ØØ 24 DF
[FV5]  532: 47 2E 3A 45 FF AC E1 F6 74 Ø7 AC 22 CØ 75 FB EB
[2R5]  533: E1 83 C2 Ø8 8C DØ 8E CØ 8B DC E8 ØØ ØØ 5F 2E 8C
[7H5]  534: 95 12 ØØ 2E 89 A5 17 ØØ B8 ØØ 4B CD 21 FA B8 ØØ
[4T5]  535: ØØ 8E DØ BC ØØ ØØ FB 8B EC 8E 5E Ø8 8B 6E Ø6 81
[CM5]  536: C4 ØE Ø1 B4 ØF CD 1Ø B4 Ø3 CD 1Ø 8Ø FE 18 72 Ø8
[1M5]  537: B8 ØA ØE CD 1Ø B6 17 B4 Ø2 CD 1Ø 8B ØE 24 ØØ B4
[TT5]  538: Ø1 CD 1Ø FF E5 8B EC B4 ØF CD 1Ø B4 Ø3 CD 1Ø 8A
[5R5]  539: 46 Ø2 B4 Ø5 CD 1Ø B4 ØF CD 1Ø B4 Ø2 CD 1Ø C2 Ø6
[DL5]  54Ø: ØØ 8C DA 83 C2 Ø3 83 EB Ø3 74 ØF 33 CØ 8E C2 33
[6L5]  541: FF B9 Ø8 ØØ F3 AB 42 4B 75 F3 BB Ø4 ØØ BA 79 Ø8
[RV5]  542: FF D2 4B 75 F8 C3 1E 8C DØ 8E D8 8B FC 83 C7 Ø5
[VU5]  543: 8A 5D FF 8Ø FB FF 75 Ø2 FE CB B7 ØØ 88 39 8B D7
[MS5]  544: B4 39 CD 21 1F C2 ØØ Ø1 1E 8C DØ 8E D8 8B FC 83
[CY5]  545: C7 Ø5 8A 5D FF 8Ø FB FF 75 Ø2 FE CB B7 ØØ 88 39
[LT5]  546: 8B D7 B4 3A CD 21 1F C2 ØØ Ø1 1E 8C DØ 8E D8 8B
[EV5]  547: FC 83 C7 Ø5 8A 5D FF 8Ø FB FF 75 Ø2 FE CB B7 ØØ
[WK5]  548: 88 39 8B D7 B4 3B CD 21 1F C2 ØØ Ø1 5A 8F Ø6 26
[YR5]  549: ØØ 8F Ø6 28 ØØ FF E2 B8 Ø7 ØE CD 1Ø C3 81 EC FC
[ZS5]  55Ø: ØØ 8B FC 8C DØ 8E CØ 36 8A 85 FE ØØ B4 ØØ 8B C8
[7U5]  551: 36 8B 95 FC ØØ AA BØ 2Ø F3 AA FF E2 8B EC 8E Ø6
[PR5]  552: 19 ØØ 1E 8C DØ 8E D8 8D 76 Ø6 8A ØC B5 ØØ 8Ø F9
[PP5]  553: FA 72 Ø2 B1 FA Ø3 F1 8D 7C Ø1 8Ø F9 Ø4 76 Ø2 B1
[1P5]  554: Ø4 E3 Ø8 8Ø 3C 2E 74 ØF 4E E2 F8 C7 Ø5 2E 42 C7
[RC5]  555: 45 Ø2 41 53 83 C7 Ø4 C6 Ø5 ØØ 8D 56 Ø7 B1 2Ø B4
[ØL5]  556: 3C CD 21 72 33 8B D8 8C CØ 87 46 Ø4 89 46 Ø6 B4
[5K5]  557: FD 87 46 Ø2 89 46 Ø8 B9 Ø7 ØØ 8D 56 Ø3 B4 4Ø CD
```

Listing 2. »sbasic.bin« stellt die neueste Version der Laufzeitbibliothek dar. Bitte nur dann eingeben, wenn Sie die aktuelle Assembler-Laufzeitbibliothek von SBasic nicht besitzen.

```
[FH5]  558: 21 72 11 3B C1 75 ØD 8E 5E Ø4 8B 56 Ø6 8B 4E Ø8
[PL5]  559: B4 4Ø CD 21 B4 3E CD 21 1F C2 Ø4 Ø1 8B EC 8E Ø6
[KR5]  56Ø: 19 ØØ 1E 8C DØ 8E D8 8D 76 Ø4 8A ØC B5 ØØ 8Ø F9
[PP5]  561: FA 72 Ø2 B1 FA Ø3 F1 8D 7C Ø1 8Ø F9 Ø4 76 Ø2 B1
[1P5]  562: Ø4 E3 Ø8 8Ø 3C 2E 74 ØF 4E E2 F8 C7 Ø5 2E 42 C7
[AD5]  563: 45 Ø2 41 53 83 C7 Ø4 C6 Ø5 ØØ 8D 56 Ø5 B8 ØØ 3D
[PM5]  564: CD 21 72 35 8B D8 B9 Ø7 ØØ B4 3F CD 21 72 26 3D
[AN5]  565: Ø7 ØØ 75 21 8Ø 7E Ø5 FD 75 1B 8B 56 Ø2 83 FA FF
[PR5]  566: 75 Ø8 8E 5E Ø6 8B 56 Ø8 EB Ø4 8C CØ 8E D8 8B 4E
[ZS5]  567: ØA B4 3F CD 21 B4 3E CD 21 1F C2 Ø2 Ø1 8B EC 8A
[WL5]  568: 5E Ø2 B7 ØØ B4 ØB CD 1Ø C2 Ø2 ØØ 8B EC 8B 56 Ø4
[AK5]  569: 8A 46 Ø2 EE C2 Ø4 ØØ BØ B6 E6 43 BØ 14 E6 43 BØ
[AH5]  57Ø: ØØ E6 4Ø 8B EC 8A 4E Ø2 83 C5 Ø3 E8 48 Ø2 22 CØ
[4K5]  571: 75 ØD BØ 36 E6 43 BØ ØØ E6 4Ø E6 4Ø C2 ØØ Ø1 3C
[VF5]  572: 41 72 2E 3C 47 77 2A 2C 41 B4 ØØ E8 Ø7 ØØ ØA ØC
[SD5]  573: Ø1 Ø3 Ø5 Ø6 Ø8 5B Ø3 D8 2E 8A Ø7 Ø2 Ø6 2C ØØ 8A
[5L5]  574: D8 E8 12 Ø2 8A 3E 2D ØØ 3C 23 75 3Ø FE C3 E9 87
[BH5]  575: ØØ 3C 5Ø 75 Ø4 BØ ØØ EB E6 3C 4F 75 11 E8 F6 Ø1
[ZM5]  576: E8 Ø7 Ø2 5Ø BØ ØC F6 E2 A2 2C ØØ 58 EB AØ 3C 4E
[WR5]  577: 75 12 E8 E1 Ø1 E8 F2 Ø1 8A DA EB C8 3C 2B 75 52
[2U5]  578: FE C3 EB 54 3C 4C 75 ØD E8 CB Ø1 E8 DC Ø1 88 16
[YR5]  579: 2D ØØ E9 79 FF 3C 54 75 ØD E8 BA Ø1 E8 CB Ø1 88
[9L5]  58Ø: 16 2E ØØ E9 68 FF 3C 3C 75 12 8Ø 3E 2C ØØ ØC 72
[RP5]  581: Ø5 8Ø 2E 2C ØØ ØC E8 9D Ø1 E9 52 FF 3C 3E 75 35
[4E5]  582: 8Ø 3E 2C ØØ 48 73 Ø5 8Ø Ø6 2C ØØ ØC E8 87 Ø1 E9
[XS5]  583: 3C FF 3C 2D 75 Ø5 FE CB E8 7B Ø1 3C 3Ø 72 Ø9 3C
[MP5]  584: 39 77 Ø5 E8 84 Ø1 8A FA B4 ØØ 3C 2E 75 33 FE C4
[9L5]  585: E8 63 Ø1 EB F5 3C 4D 75 22 E8 5A Ø1 3C 4E 75 Ø7
[RF5]  586: C6 Ø6 2F ØØ Ø7 EB 14 3C 4C 75 Ø7 C6 Ø6 2F ØØ Ø8
[MP5]  587: EB Ø9 3C 53 75 Ø5 C6 Ø6 2F ØØ Ø6 E8 38 Ø1 E9 ED
[RM5]  588: FE E8 AA ØØ Ø1 ØØ 43 47 43 43 7C 3F EC 3B 8F 38
[XP5]  589: 62 35 63 32 8F 2F E4 2C 5F 2A FE 27 CØ 25 A1 23
[MM5]  59Ø: A1 21 BE 1F F6 1D 48 1C B1 1A 32 19 C8 17 72 16
[BP5]  591: 3Ø 15 FF 13 EØ 12 D1 11 D1 1Ø DF ØF FB ØE 24 ØE
[9M5]  592: 59 ØD 99 ØC E4 ØB 39 ØB 98 ØA ØØ ØA 7Ø Ø9 E8 Ø8
[FH5]  593: 68 Ø8 FØ Ø7 7E Ø7 12 Ø7 AC Ø6 4C Ø6 F2 Ø5 9D Ø5
[UE5]  594: 4C Ø5 ØØ Ø5 B8 Ø4 74 Ø4 34 Ø4 F8 Ø3 BF Ø3 89 Ø3
[WC5]  595: 56 Ø3 26 Ø3 F9 Ø2 CE Ø2 A6 Ø2 8Ø Ø2 5C Ø2 3A Ø2
[TH5]  596: 1A Ø2 FC Ø1 DF Ø1 C4 Ø1 AB Ø1 93 Ø1 7C Ø1 67 Ø1
[RC5]  597: 53 Ø1 4Ø Ø1 2E Ø1 1D Ø1 ØD Ø1 FE ØØ FØ ØØ E2 ØØ
[6G5]  598: D6 ØØ CA ØØ BE ØØ B4 ØØ A9 ØØ AØ ØØ 97 ØØ 5E 5Ø
[PN5]  599: 8A C3 B4 ØØ D1 EØ Ø3 FØ 2E 8B Ø4 E6 42 8A C4 E6
[NS5]  6ØØ: 42 8A DF B7 ØØ D1 E3 D1 E3 D1 E3 D1 E3 58 5Ø 8A
[ZR5]  6Ø1: C4 B4 ØØ 8B F8 BØ AØ F6 26 2F ØØ 8B F7 4E 78 ØB
[3U5]  6Ø2: BA Ø3 ØØ F7 E2 D1 EA D1 D8 EB F2 8A 16 2E ØØ B6
[TM5]  6Ø3: ØØ F7 E2 F7 F3 8B FØ B8 ØØ Ø5 4F 78 ØB BA Ø3 ØØ
[PU5]  6Ø4: F7 E2 D1 EA D1 D8 EB F2 8A 16 2E ØØ B6 ØØ F7 E2
[XM5]  6Ø5: F7 F3 8B F8 E4 61 ØC Ø3 E6 61 E4 4Ø 22 CØ 78 FA
[GL5]  6Ø6: E4 4Ø 22 CØ 79 FA 4E 75 Ø6 E4 61 24 FD E6 61 4F
[PL5]  6Ø7: 75 E8 58 E9 B8 FD BØ ØØ E3 ØF 8A 46 ØØ 3C 61 72
[VH5]  6Ø8: Ø6 3C 7A 77 Ø2 24 DF 45 49 C3 BA ØØ ØA 2C 3Ø 86
[HU5]  6Ø9: C2 F6 E6 Ø2 DØ E8 DE FF 3C 3Ø 72 Ø4 3C 39 76 ED
[DN5]  61Ø: C3 8B EC F6 46 Ø3 8Ø 79 Ø3 F7 5E Ø2 C3 8B EC 8Ø
[SV5]  611: 66 Ø4 7F C3 1E 8C DØ 8E D8 8B FC 83 C7 Ø5 8A 5D
[VX5]  612: FF 8Ø FB FF 75 Ø2 FE CB B7 ØØ 88 39 8B D7 B4 41
[FP5]  613: CD 21 1F C2 ØØ Ø1 8B EC 83 C5 Ø3 8A 5E FF B7 ØØ
[VV5]  614: 8B FB 23 DB 74 ØC 8Ø 7B FF 3A 74 Ø6 8Ø 7B FF 5C
[AL5]  615: 75 1Ø 8Ø FB FC 77 ØB C7 Ø3 2A 2E C6 43 Ø2 2A 8Ø
[RU5]  616: C3 Ø3 8Ø FB FF 75 Ø2 B3 FE 88 5E FF Ø3 DD B2 ØØ
[4C5]  617: 36 88 17 8Ø 7E Ø1 3A 75 11 8A 46 ØØ 24 DF 2C 4Ø
[SN5]  618: 72 Ø8 3C 1A 77 Ø4 8A DØ EB Ø8 B4 19 CD 21 8A DØ
[NS5]  619: FE C2 52 33 DB 8A C2 Ø4 4Ø BA A3 Ø1 FF D2 BØ 3A
[VW5]  62Ø: BA A3 Ø1 FF D2 BØ 5C BA A3 Ø1 FF D2 5A 52 81 EC
[8N5]  621: 81 ØØ 8B F4 36 C6 Ø4 5C 46 1E 8C DØ 8E D8 B4 47
[FW5]  622: CD 21 8A 4E FF B5 ØØ 8B FD Ø3 F9 4F 36 8Ø 3D 5C
[YS5]  623: 74 Ø4 E2 F7 EB 27 3B FD 75 Ø1 47 36 FF 35 52 8B
[GK5]  624: D5 36 C6 Ø5 ØØ B4 3B CD 21 5A 36 8F Ø5 72 ØE 83
[KS5]  625: C6 4Ø B4 47 CD 21 8D 54 BF B4 3B CD 21 1F 36 8A
[2P5]  626: Ø4 46 22 CØ 74 Ø7 BA A3 Ø1 FF D2 EB F1 81 C4 81
[PN5]  627: ØØ BØ ØD BA A3 Ø1 FF D2 B4 2F CD 21 8D 77 1E 33
[BS5]  628: DB 1E 8C DØ 8E D8 8B D4 83 C2 Ø7 B1 31 B4 4E CD
[HK5]  629: 21 1F 73 Ø3 E9 93 ØØ 33 ED 8B FE 26 8A Ø3 22 CØ
[PU5]  63Ø: 74 1E 3C 2E 74 Ø8 BA A3 Ø1 FF D2 45 EB ED BØ 2E
[2U5]  631: 83 FD Ø8 73 F1 BØ 2Ø BA A3 Ø1 FF D2 45 4F EB EE
[ØU5]  632: 83 FD ØC 73 ØA BØ 2Ø BA A3 Ø1 FF D2 45 EB F1 BD
[VK5]  633: Ø6 ØØ 26 F6 44 F7 1Ø 74 26 BØ 3C BA A3 Ø1 FF D2
[WR5]  634: BØ 44 BA A3 Ø1 FF D2 BØ 49 BA A3 Ø1 FF D2 BØ 52
[VT5]  635: BA A3 Ø1 FF D2 BØ 3E BA A3 Ø1 FF D2 BD Ø1 ØØ BØ
[NP5]  636: 2Ø BA A3 Ø1 FF D2 4D 75 F6 B4 ØF CD 1Ø B4 Ø3 CD
[2H5]  637: 1Ø 33 DB 8Ø C2 12 3A 16 ØB ØØ 72 Ø7 BØ ØD BA A3
[LV5]  638: Ø1 FF D2 B4 4F CD 21 E9 68 FF B4 ØF CD 1Ø B4 Ø3
[5R5]  639: CD 1Ø 33 DB 22 D2 74 Ø7 BØ ØD BA A3 Ø1 FF D2 BØ
[MT5]  64Ø: 2Ø BA A3 Ø1 FF D2 5A B4 36 CD 21 F7 E1 F7 E3 BB
[WP5]  641: ØA ØØ 33 C9 8B FØ 8B C2 33 D2 F7 F3 96 F7 F3 52
[4S5]  642: 96 23 F6 EØ F3 23 CØ 75 EF 33 DB F7 D9 58 ØC 3Ø
[ZG5]  643: 51 BA A3 Ø1 FF D2 59 E2 F4 E8 ØF ØØ 2Ø 42 79 74
[7E5]  644: 65 73 2Ø 66 72 65 65 2Ø ØD ØD ØØ 5E 2E 8A Ø4 22
[2R5]  645: CØ 74 Ø8 BA A3 Ø1 FF D2 46 EB F1 C2 ØØ Ø1 1E 8C
[PW5]  646: DØ 8E D8 8E CØ 8B F4 81 C6 Ø5 Ø1 8A 5C FF 8Ø FB
[UY5]  647: FF 75 Ø2 FE CB B7 ØØ 88 38 8B D6 8D BC ØØ FF 8A
[LX5]  648: 5D FF 8Ø FB FF 75 Ø2 FE CB 88 39 B4 56 CD 21 1F
[CK5]  649: C2 ØØ Ø2 8B EC 8B 5E Ø2 23 DB 75 74 53 33 DB 33
[9S5]  65Ø: ED 33 F6 33 FF B9 7F ØØ E4 61 24 FC ØC Ø1 E6 61
[7M5]  651: BØ A4 E6 43 BØ ØØ E6 42 BØ 94 E6 43 BØ 1F E6 42
[SS5]  652: BA Ø1 Ø2 EE EC DØ E8 83 D3 ØØ DØ E8 83 D5 ØØ DØ
[WK5]  653: E8 83 D6 ØØ DØ E8 83 D7 ØØ E4 42 24 1Ø 74 FA E4
[EU5]  654: 42 24 1Ø 75 FA E2 DD FB B1 Ø2 D3 E3 D3 E5 D3 E6
[2C5]  655: D3 E7 E8 Ø8 ØØ ØØ ØØ ØØ ØØ ØØ ØØ ØØ 8B C3 5B
[GN5]  656: 2E 89 Ø7 2E 89 6F Ø2 2E 89 77 Ø4 2E 89 7F Ø6 5B
[MP5]  657: E8 ØØ ØØ 5E 83 E3 Ø3 D1 E3 2E 8B 8Ø E2 FF 8B EC
[3K5]  658: 89 46 Ø2 C3 8B EC BA Ø1 Ø2 EC 33 D2 8A 66 Ø2 8Ø
[WH5]  659: E4 Ø6 75 Ø4 A8 1Ø EB 14 8Ø FC Ø4 75 Ø4 A8 2Ø EB
[KK5]  66Ø: ØB 8Ø FC Ø2 75 Ø4 A8 4Ø EB Ø2 A8 8Ø 75 Ø1 4A 89
[ZT5]  661: 56 Ø2 C3 BA D2 Ø8 FF E2 8B EC B2 ØØ 8A 76 Ø2 8B
[AK5]  662: 7E Ø3 8A 5E Ø5 81 CF ØØ 8Ø 8Ø FB ØØ 75 Ø1 C3 8Ø
[MX5]  663: F3 8Ø ØA DB 79 ØE F6 DB DØ EB 73 Ø4 D1 EF D1 DA
[7Y5]  664: F6 DB EB ØA DØ EB 73 Ø6 D1 EF D1 DA FE C3 8Ø F3
[YH5]  665: 8Ø 88 5E Ø5 1E B9 18 ØØ BE Ø1 ØØ 33 CØ 8B E8 8B
[2X5]  666: D8 D1 E2 D1 D7 D1 D5 D1 E2 D1 D7 D1 D5 2B EE 1B
[TV5]  667: D8 D1 E2 D1 D7 D1 D5 D1 D3 D1 E2 D1 D7 D1 D5 D1
[6X5]  668: D3 D1 E6 D1 DØ 8E DB 8E C5 2B EE 1B D8 72 ØE 2B
[DR5]  669: EE 1B D8 72 Ø8 83 ED Ø1 83 DB ØØ 73 Ø6 8C DB 8C
```

```
[FM5]  67Ø: C5 EB Ø3 83 CE Ø1 E2 C9 83 C6 Ø1 15 ØØ ØØ D1 E8
[GR5]  671: D1 DE 1F 8B EC 8A 5E Ø4 24 7F 8Ø E3 8Ø ØA C3 88
[ZM5]  672: 46 Ø4 89 76 Ø2 C3 8B EC 83 C5 Ø3 8A 4E FF B5 ØØ
[VN5]  673: E8 1E ØØ 8A FA E8 19 ØØ 8A DA E8 14 ØØ 83 FA 64
[HS5]  674: 73 Ø4 81 C2 6C Ø7 8B CA 8B D3 B4 2B CD 21 C2 ØØ
[AF5]  675: Ø1 33 D2 E8 17 ØØ 2C 3Ø 72 1C 3C ØA 73 18 B4 ØØ
[XL5]  676: 5Ø B8 ØA ØØ F7 E2 8B DØ 58 Ø3 DØ EB E6 BØ ØØ E3
[LS5]  677: Ø5 8A 46 ØØ 45 49 C3 81 EC FC ØØ 8B EC 8C DØ 8E
[ML5]  678: CØ 1E BE 8Ø ØØ 8E 1E 2A ØØ AC 8A C8 B5 ØØ 8B 96
[NE5]  679: FE ØØ 23 D2 74 32 8D 7E Ø1 E8 34 ØØ 74 22 3C 2Ø
[1F5]  68Ø: 74 F7 3C 2C 74 1A 3C 61 72 Ø6 3C 7A 77 Ø2 2C 2Ø
[4F5]  681: AA E8 1C ØØ 74 ØA 3C 2Ø 74 Ø6 3C 2F 75 E4 4E 41
[PT5]  682: 4A 75 D3 8D 55 FF 2B D5 88 56 ØØ 1F FF A6 FC ØØ
[VS5]  683: BØ ØD E3 Ø2 49 AC 3C ØD C3 BA 54 ØA FF E2 BA 47
[9Y5]  684: ØA FF E2 8B EC FF 76 Ø4 FF 76 Ø2 BA 54 ØA FF D2
[VV5]  685: 8B EC FF 76 Ø8 FF 76 Ø6 BA 47 ØA FF D2 BA BE Ø6
[9W5]  686: FF D2 8B EC 8F 46 Ø6 8F 46 Ø8 C3 8B EC 8B 46 Ø4
[ØP5]  687: 24 7F 8Ø FC 81 72 12 BB ØØ 81 53 33 DB 53 5Ø FF
[LZ5]  688: 76 Ø2 BA BE Ø6 FF D2 EB Ø4 5Ø FF 76 Ø2 8B EC FF
[EP5]  689: 76 Ø2 FF 76 ØØ FF 76 Ø2 FF 76 ØØ BA 22 Ø6 FF D2
[FN5]  69Ø: B8 B3 76 5Ø B8 BD 83 5Ø 8B EC FF 76 Ø6 FF 76 Ø4
[ZN5]  691: BA 22 Ø6 FF D2 B8 1E 79 5Ø B8 A6 F4 5Ø BA 86 Ø5
[FY5]  692: FF D2 8B EC FF 76 Ø6 FF 76 Ø4 BA 22 Ø6 FF D2 B8
[PV5]  693: 83 7B 5Ø B8 BØ FC 5Ø BA 86 Ø5 FF D2 8B EC FF 76
[DP5]  694: Ø6 FF 76 Ø4 BA 22 Ø6 FF D2 B8 ØC 7C 5Ø B8 67 1F
[BU5]  695: 5Ø BA 86 Ø5 FF D2 8B EC FF 76 Ø6 FF 76 Ø4 BA 22
[DX5]  696: Ø6 FF D2 B8 DE 7C 5Ø B8 CB 53 5Ø BA 86 Ø5 FF D2
[WT5]  697: 8B EC FF 76 Ø6 FF 76 Ø4 BA 22 Ø6 FF D2 B8 14 7D
[6T5]  698: 5Ø B8 7Ø 64 5Ø BA 86 Ø5 FF D2 8B EC FF 76 Ø6 FF
[FS5]  699: 76 Ø4 BA 22 Ø6 FF D2 B8 B7 7D 5Ø B8 51 EA 5Ø BA
[XU5]  7ØØ: 86 Ø5 FF D2 8B EC FF 76 Ø6 FF 76 Ø4 BA 22 Ø6 FF
[8P5]  7Ø1: D2 B8 63 7D 5Ø B8 88 3Ø 5Ø BA 86 Ø5 FF D2 8B EC
[ZT5]  7Ø2: FF 76 Ø6 FF 76 Ø4 BA 22 Ø6 FF D2 B8 92 7E 5Ø B8
[7T5]  7Ø3: 99 44 5Ø BA 86 Ø5 FF D2 8B EC FF 76 Ø6 FF 76 Ø4
[7S5]  7Ø4: BA 22 Ø6 FF D2 B8 4C 7E 5Ø B8 91 CC 5Ø BA 86 Ø5
[FY5]  7Ø5: FF D2 8B EC FF 76 Ø6 FF 76 Ø4 BA 22 Ø6 FF D2 B8
[2Z5]  7Ø6: AA 7F 5Ø B8 AA AA 5Ø BA 86 Ø5 FF D2 8B EC FF 76
[XH5]  7Ø7: Ø6 FF 76 Ø4 BA 22 Ø6 FF D2 B8 ØØ 81 5Ø 33 CØ 5Ø
[ZR5]  7Ø8: BA 86 Ø5 FF D2 8B EC 8F 46 Ø4 8F 46 Ø6 BA 22 Ø6
[KL5]  7Ø9: FF D2 8B EC 8Ø 7E Ø9 81 72 19 8Ø 7E Ø3 ØØ 74 Ø4
[ØK5]  71Ø: 8Ø 76 Ø2 8Ø B8 49 81 5Ø B8 DA ØF 5Ø BA 86 Ø5 FF
[DG5]  711: D2 8B EC F6 46 Ø8 8Ø 74 ØA 8Ø 7E Ø3 ØØ 74 Ø4 8Ø
[1N5]  712: 76 Ø2 8Ø 8F 46 Ø6 8F 46 Ø8 C3 Ø7 5B 8A 1F 22 DB
[HD5]  713: 78 42 33 C9 33 D2 B8 Ø1 42 CD 21 72 37 52 5Ø 33
[2S5]  714: C9 33 D2 B8 Ø2 42 CD 21 8B FØ 8B FA 5A 59 B8 ØØ
[ØP5]  715: 42 CD 21 72 1F 8B CE 8B D7 B7 AØ 23 D2 78 Ø8 D1
[836]  716: E1 D1 D2 FE CF 75 F4 8A DE 8Ø E3 7F 53 8A CD 8A
[Z83]  717: EA 51 Ø6 C3 33 CØ 5Ø 5Ø Ø6 C3
```

Listing 2. »sbasic.bin« stellt die neueste Version der Laufzeitbibliothek dar. Bitte nur dann eingeben, wenn Sie die aktuelle Assembler-Laufzeitbibliothek von SBasic nicht besitzen.

```
[XØR] 1Ø1 REM SBasic 2.Ø, (C) 1989-91 DMV Widuch GmbH &
      Co.KG, DOS International, Autor: Wolfgang Lorenz
      (ASCII-/LABEL-BASIC-Preprocessor: Peter Knappe)
[HHE] 1Ø2 CLS:PRINT:PRINT "SBasic 2.Ø  ASCII-/GW-Basic-
      Compiler   (C)1989-91 DMV Widuch GmbH & Co KG"
[5H3] 1Ø3 EINDAT$=COMMAND$(1)
[4W9] 1Ø4 IF EINDAT$="" THEN PRINT:INPUT "Quelldatei [.
      BAS]: ",EINDAT$
[3R3] 1Ø5 IF EINDAT$="" THEN END
[H66] 1Ø6 PREFIX$=EINDAT$:I=INSTR(EINDAT$,".")
[4DA] 1Ø7 IF I=Ø THEN EINDAT$=EINDAT$+".BAS" ELSE PREFI
      X$=LEFT$(EINDAT$,I-1)
[GXA] 127 EXT$="EXE":IF COMMAND$(2)="/GWB" OR COMMAND$(
      2)="/gwb" THEN EXT$="GWB"
[CF8] 128 AUSDAT$=PREFIX$+"."+EXT$:PARAM$=EINDAT$:GOSUB
       136ØØ
[5PD] 129 PRINT:PRINT "warten...":LOCATE CSRLIN-2,1:DEF
      SEG=SEGADR!+3Ø!:BLOAD PATH$+"PRE",144Ø7!
[6CD] 13Ø IF PEEK(144Ø7)=Ø THEN PRINT:PRINT "Preprocess
      or ";PATH$;"PRE nicht gefunden":END
[ME6] 131 GOSUB 135ØØ:DEF SEG=SEGADR!:BUFADR!=8Ø!
[742] 132 GOSUB 1ØØØØ
[2R1] 137 GOTO 132
[151] 147 REM
[LV4] 148 DEF SEG=SEGADR!:EXEADR!=448
[S96] 151 WORT=FNI(32+RC!+PC!) AND 511:GOSUB 11Ø43
[4M6] 152 WORT=INT((32+RC!+PC!+511)/512):GOSUB 11Ø43
[WA3] 154 WORT=2:GOSUB 11Ø43
[ZA3] 161 WORT=Ø:FOR I=1 TO 3
[5Ø1] 166 I=Ø
[HM3] 167 IF UPADR(I)=Ø THEN 174
[E41] 17Ø REM
[UH2] 171 BUFADR!=2*I+81
[7V3] 172 AUS=PEEK(BUFADR!)+128
[A83] 173 POKE BUFADR!,AUS
[651] 176 REM
[RK4] 182 PARAM$=AUSDAT$:GOSUB 136Ø7
[4C7] 183 DEF SEG=SEGADR!+3Ø!:BUFADR!=1441Ø:CALL BUFADR
      !
[BU7] 184 PRINT:PRINT:I=CSRLIN:SHELL "LZEXE "+PREFIX$
[FV6] 185 J=CSRLIN:LOCATE I,1:PRINT STRING$(8Ø," ")
[GM8] 186 KILL PREFIX$+".OLD":LOCATE J-1+3*(J>2Ø),1:PRI
      NT "EXE";
[FP8] 187 PRINT "-Datei -";AUSDAT$;"- generiert";STRING
      $(255," ")
[4C3] 188 PRINT:PRINT:END
[BW4] 1ØØØ REM *** Quelldaten lesen ***
[1N3] 1Ø01 DEF SEG=SEGADR!+93Ø!
[8D3] 1Ø02 BUFADR!=SAVADR!+1
[LD2] 1Ø03 EXEADR!=3ØØØ
[X31] 1Ø04 REM
```

```
[Y31] 1ØØ5 REM
[Z31] 1ØØ6 REM
[941] 1Ø34 REM
[BRB] 1155 IF EIN<>14 THEN PRINT "Zeilennummer oder Lab
      el erwartet";:GOTO 999Ø
[AZB] 1181 IF EIN<>137 AND EIN<>2Ø5 THEN PRINT "THEN od
      er GOTO erwartet";:GOTO 999Ø
[4X3] 1281 IF EIN<>233 THEN 116Ø4
[3R3] 1486 IF EIN<>4Ø THEN 116Ø4
[NRB] 2Ø1Ø IF EIN<>14 THEN PRINT "Zeilennummer oder Lab
      el erwartet/";:GOTO 998Ø
[PL6] 216Ø PRINT "Unbekannter Befehl";:GOTO 999Ø
[W44] 673Ø PRINT FEIN;:GOTO 116Ø4
[KM4] 7123 IF STACK(SP)<>16 THEN 116Ø4
[PD8] 999Ø REM *** Zeilennummer und Fehlerzeile ausgebe
      n ***
[ED5] 9991 EXEADR!=FNR!(ZEILNR(ZEILE-1))
[CY2] 9992 AUS=(EXEADR!=Ø)
[27C] 9993 PRINT " in Zeile";ZEILE;"(";LEFT$(">",AUS);M
      ID$(STR$(EXEADR!-65535!*AUS),2);")":"
[94B] 9994 PRINT:PARAM$=EINDAT$:EXEADR!=ZEILE:GOSUB 136
      Ø7:ERRKZ=1:GOTO 135ØØ
[E61] 9996 REM
[NA3] 9997 EIN=PEEK(BUFADR!)
[AG3] 9998 BUFADR!=BUFADR!+1
[663] 11Ø03 POKE EXEADR!,AUS
[3B3] 11Ø04 EXEADR!=EXEADR!+1
[P61] 11Ø05 REM
[N56] 11Ø43 AUS1=FNHI(WORT):AUS=WORT-256*AUS1
[M92] 11Ø44 GOSUB 11ØØ3
[VZ1] 11Ø45 AUS=AUS1
[B22] 11Ø46 GOTO 11ØØ3
[PF4] 116Ø4 PRINT "Break";:GOTO 999Ø
[RG5] 135ØØ REM *** Preprocessor-Aufruf ***
[ØS5] 135Ø1 DEF SEG=SEGADR!+3Ø:BUFADR!=144Ø7!
[1P5] 135Ø2 CALL BUFADR!:IF ERRKZ=1 THEN END
[8A7] 135Ø3 DEF SEG=SEGADR!:SAVADR!=PEEK(Ø)+256!*PEEK(1
      )
[6K8] 135Ø4 EXEADR!=PEEK(2)+256!*PEEK(3):ZEILE=PEEK(4)+
      256*PEEK(5)
[9UA] 135Ø5 DEF SEG=SEGADR!+93Ø:BUFADR!=SAVADR!:GOSUB 9
      995:AUS=PEEK(BUFADR!)
[HV7] 135Ø6 IF EIN=255 THEN IF AUS=Ø OR AUS>1Ø THEN RET
      URN
[KL5] 135Ø7 PRINT:IF EIN<>254 THEN EIN=AUS
[FU9] 135Ø8 ON EIN GOTO 1351Ø,13511,13512,13513,13514,1
      3515,13516,13517,13518,13519
[KD8] 135Ø9 PRINT "Quelldatei -";EINDAT$;"- geschützt":
      END
[7EA] 1351Ø PRINT "Laufzeitbibliothek ";PATH$;"BIN nich
      t gefunden":END
[APB] 13511 PRINT "Quelldatei -";EINDAT$;"- nicht gefun
      den oder ungültig":END
[U28] 13512 PRINT "Quelldatei -";EINDAT$;"- zu groß":EN
      D
[W3A] 13513 PRINT "Zeilennummer nicht > Vorzeilennummer
      ";:GOTO 9992
[6P7] 13514 PRINT "Zeilennummer Überlauf";:GOTO 9992
[B37] 13515 PRINT "Label nicht definiert";:GOTO 9992
[2K7] 13516 PRINT "Label bereits definiert";:GOTO 9992
[MR7] 13517 PRINT "DOUBLE nicht vorgesehen";:GOTO 9992
[YHB] 13518 PRINT "/GWB: Quelldatei -";EINDAT$;"- berei
      ts im GWBASIC-Format":END
[EF6] 13519 KILL AUSDAT$:NAME "###.GWB" AS AUSDAT$
[SY4] 1352Ø PRINT "GWBASIC";:GOTO 187
[ML7] 136ØØ REM *** SBASIC.EXE-PATH / SBASIC.PRE-PARAM
      ***
[EZ4] 136Ø1 SEGADR!=256!*PEEK(43)+PEEK(42)
[BA8] 136Ø2 DEF SEG=SEGADR!:DEF SEG=256!*PEEK(45)+PEEK(
      44)-1:I=Ø
[WVE] 136Ø3 I=I+1:IF PEEK(I)=Ø AND PEEK(I+1)=Ø AND PEEK
      (I+2)=1 AND PEEK(I+3)=Ø THEN I=I+3:PATH$="" ELSE
      136Ø3
[VS8] 136Ø4 I=I+1:P=PEEK(I):PATH$=PATH$+CHR$(P):IF P<>4
      6 THEN 136Ø4
[AZ3] 136Ø5 SEGADR!=SEGADR!+&H27ØØ
[VZ5] 136Ø6 IF EXT$="GWB" THEN EXEADR!=&HFFFF
[4E9] 136Ø7 DEF SEG=SEGADR!:PARAM$=PARAM$+CHR$(Ø)+PATH$
      +"BIN"+CHR$(Ø)
[C29] 136Ø8 AUS=INT(EXEADR!/256!):POKE Ø,EXEADR!-256!*A
      US:POKE 1,AUS
[ZD9] 136Ø9 FOR I=1 TO LEN(PARAM$):POKE I+1,ASC(MID$(PA
      RAM$,I,1)):NEXT
[EW1] 1361Ø RETURN
[VS5] 14ØØØ REM *** Speicherreservierung ***
[ZX2] 14ØØ1 DIM BUFFER$(4ØØ)
```

Listing 3. Die Merge-Datei ergibt zusammen mit Version 1.2 den aktuellen Basic-Teil von SBasic

```
[4G5]    1: FD ØØ ØØ ØØ ØØ 98 ØC EB 4A 9Ø E9 93 Ø7 98 ØC ØØ
[ØZ4]    2: ØØ ØØ ØØ ØØ ØØ ØØ ØØ ØØ ØØ ØØ ØØ ØØ ØØ ØØ ØØ ØØ
[XZ4]    3: ØØ ØØ ØØ ØØ ØØ ØØ ØØ ØØ ØØ ØØ ØØ ØØ ØØ ØØ 23 23
[NM5]    4: 23 2E 47 57 42 ØØ 3E 3D 3C 2B 2D 2A 2F 5E 2D 2D
[ZX5]    5: 2D 2D 2D 2D 5C 2Ø 28 3A 3B 2C F4 E7 E6 E9 E8 EB
[8U5]    6: EA ED EC 1E 2E 8Ø 3E 25 ØØ Ø2 74 49 8C CD 83 ED
[CU5]    7: 1E 8E DD 81 C5 A2 Ø3 BE Ø2 ØØ AC ØA CØ 75 FB 8B
[5S5]    8: D6 B8 ØØ 3D CD 21 8B D8 8E DD BD ØØ Ø1 72 4C 8C
[1U5]    9: CD 8E DD 2B D2 B9 4Ø 38 B4 3F CD 21 B4 3E CD 21
[FW5]   1Ø: 83 ED 19 8E C5 2B F6 2B FF AD AB 4Ø 75 FB BF 9Ø
```

```
[VU5]   11: Ø1 D1 F9 F2 A5 8C CD 83 ED 1E 8E DD 81 C5 A2 Ø3
[YM5]   12: BA Ø2 ØØ B8 ØØ 3D CD 21 8B D8 8B 16 ØØ ØØ 8E DD
[8H5]   13: 8E C5 89 16 15 ØØ BD ØØ Ø2 73 Ø5 2B FF E9 6Ø Ø6
[E95]   14: 8B 16 ØD ØØ 8Ø 3E 25 ØØ ØØ 75 Ø4 81 C2 ØØ Ø4 89
[MA5]   15: 16 ØF ØØ 42 89 16 11 ØØ 81 C2 FF Ø3 89 16 13 ØØ
[4T5]   16: 8B CA 83 C1 Ø2 F7 D1 51 B4 3F CD 21 8B C8 B4 3E
[9P5]   17: CD 21 58 3B C8 BD ØØ Ø3 74 C1 8B F2 8B FA Ø3 F9
[YL5]   18: 8Ø 3C FE 72 ØC 8Ø 7C Ø1 ØA 75 Ø3 FE 44 Ø1 E9 15
[AH5]   19: Ø6 B8 ØD ØA AB BØ 1A AA C7 Ø6 1B ØØ ØØ ØØ C7 Ø6
[7C5]   2Ø: 1D ØØ ØØ ØØ FF Ø6 1D ØØ 8Ø 3E 25 ØØ Ø2 75 1B 8B
[ML5]   21: 16 1D ØØ 39 16 15 ØØ 75 ØA AC CD 29 3C ØD 75 F9
[NR5]   22: E9 4B Ø6 AC 3C ØA 75 FB EB DA C7 Ø6 26 ØØ ØØ ØØ
[4C5]   23: C6 Ø6 2A ØØ ØØ C6 Ø6 29 ØØ Ø1 C6 Ø6 2B ØØ Ø1 C6
[SH5]   24: Ø6 2D ØØ 1C 8B 3E 11 ØØ 2B CØ AB 8Ø 3C 1A 74 Ø6
[6R5]   25: 8Ø 7C Ø2 1A 75 Ø3 E9 82 Ø5 2B C9 BD ØØ Ø5 E8 6F
[ZM5]   26: Ø5 3C 2Ø 74 F9 2A E4 2C 3Ø 91 8Ø F9 Ø9 77 1Ø BA
[MP5]   27: ØA ØØ F7 EA ØB D2 75 46 Ø3 C1 72 42 91 EB DF ØB
[MA5]   28: CØ 75 ØB A1 1B ØØ 4Ø 74 35 C6 Ø6 26 ØØ Ø1 39 Ø6
[CM5]   29: 1B ØØ 72 ØA BD ØØ Ø4 A1 1B ØØ 4Ø EB 23 9Ø A3 1B
[LL5]   3Ø: ØØ AA 8A C4 AA 57 8F Ø6 1F ØØ 8A C1 Ø4 3Ø 3C ØD
[YR5]   31: 75 Ø9 5Ø B8 3A 8F AB BØ D9 AA 58 EB 17 9Ø 2B CØ
[EK5]   32: A3 1B ØØ E9 3A Ø5 AA C6 Ø6 2B ØØ ØØ E8 Ø1 Ø5 C6
[L65]   33: Ø6 26 ØØ ØØ 8Ø 3E 25 ØØ ØØ 75 11 8Ø 3E 28 ØØ Ø1
[CK5]   34: 75 ØA 8Ø 3E 26 ØØ ØØ 75 Ø3 E9 37 FF 3C ØD 75 Ø3
[4D5]   35: E9 99 ØØ 77 Ø6 BD ØØ Ø2 E9 Ø5 Ø5 3C 2Ø 75 Ø9 8Ø
[1E5]   36: 3E 29 ØØ Ø1 75 Ø2 EB C4 C6 Ø6 29 ØØ Ø1 3C 22 75
[GE5]   37: Ø3 E9 48 Ø3 3C 26 75 ØD 8Ø 3C 48 74 Ø5 8Ø 3C 68
[ME5]   38: 75 Ø3 E9 65 Ø3 3C 27 75 Ø3 E9 48 Ø3 3C 2C 75 Ø2
[ZU5]   39: EB 94 57 BF 36 ØØ B9 ØF ØØ F2 AE 5F 75 19 BØ F4
[7L5]   4Ø: 2A C1 39 3E 1F ØØ 74 Ø7 8Ø 7D FF 2Ø 75 Ø1 4F C6
[KS5]   41: Ø6 2D ØØ 1C E9 6F FF 3C 28 75 ØF 8Ø 7D FF CE 74
[8M5]   42: Ø6 8Ø 7D FF D2 75 Ø3 E9 62 FF 3C 2E 74 Ø8 3C 3Ø
[SK5]   43: 72 DD 3C 39 77 Ø3 E9 4D Ø3 3C 41 72 D2 3C 5B 72
[CH5]   44: 54 3C 61 72 CA 3C 7A 77 C6 EB 4A 9Ø 8Ø 3E 25 ØØ
[BC5]   45: Ø1 75 2F 8Ø 3E 27 ØØ Ø1 75 28 8Ø 3E 2A ØØ ØØ 74
[4V5]   46: Ø5 B8 2Ø 5B 75 Ø7 B8 3A 8F AB B8 D9 2A AB 8B 1E
[DM5]   47: 21 ØØ 83 EB Ø2 8A Ø7 3C 2Ø 76 Ø4 AA 43 EB F6 BØ
[WK5]   48: 2A AA 2A CØ AA 8B 1E 11 ØØ 89 3F 89 3E 11 ØØ E8
[5P5]   49: FE Ø3 E9 2F FE 8A 26 2B ØØ 88 26 2C ØØ 8A 24 25
[LW5]   5Ø: DF DF 3D 46 4E 75 Ø9 BØ D1 AA E8 E3 Ø3 E9 DC FE
[ØG5]   51: E8 ØA Ø1 5Ø 56 51 8B Ø5 57 ØA C9 74 1B E8 87 Ø1
[6F5]   52: 74 19 8Ø 3E 2D ØØ ØE 75 ØF BD ØØ Ø6 8Ø 3E 25 ØØ
[ØE5]   53: Ø1 74 25 C6 Ø6 28 ØØ Ø1 E9 BE ØØ 81 FB 98 ØC 72
[V85]   54: 28 8Ø 3E 26 ØØ ØØ 74 17 C6 Ø6 27 ØØ Ø1 8Ø 3E 25
[9F5]   55: ØØ Ø1 74 11 BD ØØ Ø7 57 5F 59 5E 58 E9 C1 Ø3 BØ
[ØS5]   56: ØE AA 8B Ø7 AB 59 E9 A5 ØØ C6 Ø6 2D ØØ 1C 81 FB
[9R5]   57: 78 ØA 72 16 B8 FD FF 81 FB A4 ØA 72 ØA FE CØ 81
[DR5]   58: FB BØ ØB 72 Ø2 FE CØ E8 AB Ø1 8A Ø7 2B DB ØA E4
[DR5]   59: 75 53 3C AF BD ØØ Ø8 74 CØ 3C CD 74 2E 3C A1 75
[KR5]   6Ø: ØA 5Ø BØ 3A E8 8E Ø1 58 EB 21 9Ø 3C 8C 74 1C 3C
[6L5]   61: A8 74 18 3C 89 74 14 3C 8D 74 1Ø 3C CC 75 11 81
[5R5]   62: 7D FE 47 4F 75 ØA BØ 89 83 EF Ø2 C6 Ø6 2D ØØ ØE
[7P5]   63: 3C B1 75 Ø5 E8 5E Ø1 BØ E9 3C 8F 75 Ø2 FE C3 3C
[GN5]   64: 84 75 Ø2 FE C7 E8 4D Ø1 59 5E 58 ØB DB 74 2B ØA
[BK5]   65: DB 75 Ø3 E9 61 Ø1 E9 A5 Ø1 C6 Ø6 29 ØØ ØØ 8B 3E
[MB5]   66: 19 ØØ 5E 59 8Ø 3E 26 ØØ Ø1 75 Ø3 E8 ØØ Ø1 5E 58
[PP5]   67: 3C 23 75 Ø6 BD ØØ Ø8 E9 Ø6 Ø3 E9 D2 FD 57 2B C9
[5R5]   68: AA E8 CC Ø2 FE C1 3C 2E 74 F6 3C 5F 74 F2 3C 24
[9G5]   69: 74 EE 3C 3Ø 72 16 3C 39 76 E6 3C 7A 77 ØE 3C 61
[PH5]   7Ø: 72 Ø2 24 DF 3C 41 72 Ø4 3C 5B 72 D4 8Ø 3E 2D ØØ
[SF5]   71: ØE 74 Ø7 8Ø 3E 26 ØØ ØØ 74 2B 3C 2Ø 75 Ø5 E8 8F
[1P5]   72: Ø2 EB F7 3C ØD 74 1E 3C 27 74 1A 3C 3A 74 16 C6
[BF5]   73: Ø6 26 ØØ ØØ 3C 2C 74 ØD 3C 45 74 Ø9 3C 65 74 Ø5
[WG5]   74: C6 Ø6 2D ØØ 1C FE C9 ØA C9 75 13 8Ø 3E 26 ØØ Ø1
[BM5]   75: 74 Ø7 8Ø 3E 2D ØØ ØE 75 Ø5 FE C1 88 ØD 47 C6 Ø5
[4G5]   76: 2Ø 89 3E 19 ØØ 5F C3 8F Ø6 23 ØØ 8B 16 ØD ØØ 8Ø
[ZG5]   77: 3E 26 ØØ Ø1 74 ØA 8Ø 3E 2D ØØ ØE 74 Ø3 BA 98 ØC
[YN5]   78: 81 EA Ø6 Ø8 D1 FA BF Ø6 Ø8 8B CA F2 AF 75 3A 8Ø
[YD5]   79: 7D FD 8Ø 72 F6 8Ø 3E 26 ØØ Ø1 75 Ø4 89 3E 21 ØØ
[2P5]   8Ø: 8B D1 5E 59 51 56 83 C6 Ø2 57 F3 A6 8B DF 5F 75
[GR5]   81: D8 8A E3 DØ FC F5 83 D3 ØØ 5F 39 3E 1F ØØ 74 Ø7
[HK5]   82: 8Ø 7D FF 2Ø 75 Ø1 4F 2A E4 FF 36 23 ØØ C3 56 8B
[5T5]   83: 3E ØD ØØ 8Ø C1 Ø2 8B D9 AC AA E2 FC D1 FB 72 Ø3
[WK5]   84: BØ 2Ø AA A1 1B ØØ AB BØ 8Ø AA C6 Ø6 28 ØØ Ø1 89
[AM5]   85: 3E ØD ØØ 5F C3 8Ø 3E 2C ØØ Ø1 74 19 5Ø 8A 45 FF
[DL5]   86: 57 BF 45 ØØ B9 ØE ØØ F2 AE 5F 74 Ø3 BØ 2Ø AA 58
[2K5]   87: C6 Ø6 2C ØØ Ø1 AA C3 3C 22 74 12 3C ØD 74 1A 3C
[1V5]   88: 27 74 16 3C 3A 74 12 AA E8 85 Ø1 EB EA AA E8 7F
[GU5]   89: Ø1 3C 22 74 F2 3C ØD 75 F4 E9 73 FC E8 BD FF EB
[3N5]   9Ø: Ø2 9Ø AA E8 6A Ø1 3C 22 75 Ø3 E9 E2 FC 3C ØD 75
[5V5]   91: F1 E9 18 FD B8 3A 8F AB BØ D9 AA E8 52 Ø1 3C ØD
[UN5]   92: 75 F8 C6 Ø6 2A ØØ Ø1 E9 Ø2 FD E8 43 Ø1 2B D2 E8
[MH5]   93: 3E Ø1 8A E8 2C 3Ø 3C Ø9 76 14 2C Ø7 3C ØA 72 16
[GM5]   94: 3C ØF 76 ØA 2C 2Ø 3C ØA 72 ØC 3C ØF 77 Ø8 B1 Ø4
[PW5]   95: D3 E2 Ø2 DØ EB D9 BØ ØC AA 8B C2 AB 8A C5 C6 Ø6
[XL5]   96: 2B ØØ ØØ E9 Ø9 FC C6 Ø6 2B ØØ ØØ 8A 2E 2D ØØ 89
[KK5]   97: 36 17 ØØ 89 3E 19 ØØ 2B DB EB Ø8 9Ø E8 F1 ØØ 3C
[VP5]   98: 2Ø 74 F9 8A C8 2C 3Ø 3C Ø9 77 14 2A E4 93 BA ØA
[LN5]   99: ØØ F7 EA ØB D2 75 46 Ø3 C3 72 42 8B D8 EB DD 8Ø
[CP5]   1ØØ: FD ØE 74 ØD 83 FB Ø9 77 Ø8 8B C3 Ø4 11 AA EB Ø7
[4R5]   1Ø1: 9Ø 8A C5 AA 8B C3 AB 8A C1 3C 2E 74 2Ø 3C 21 74
[EL5]   1Ø2: 1C 3C 23 BD ØØ Ø8 74 21 8Ø E1 DF 8Ø F9 45 75 ØA
[4X5]   1Ø3: 8A ØC 8Ø E1 DF 8Ø F9 4C 75 Ø3 E9 92 FB 8Ø FD ØE
[3F5]   1Ø4: 75 ØA BD ØØ Ø5 3C 21 74 F1 E9 B4 ØØ 8B 36 17 ØØ
[3W5]   1Ø5: 4E 16 Ø7 81 EC ØØ Ø1 8B FC 8B DF 2A CØ AA AC AA
[YG5]   1Ø6: 26 FE Ø7 E8 6A ØØ 3C 2Ø 74 F9 3C 21 74 F5 3C 2E
[6V5]   1Ø7: 74 ED 8Ø 7D FF 45 75 Ø8 3C 2B 74 E3 3C 2D 74 DF
[GL5]   1Ø8: 3C 3Ø 72 14 3C 3A 72 D7 24 DF 3C 45 75 ØA 8Ø 3C
[9N5]   1Ø9: 4C 74 Ø5 8Ø 3C 6C 75 C7 89 36 17 ØØ 8C CD 83 ED
[PL5]   11Ø: 19 8E C5 26 8B 16 C8 ØØ 26 2B 16 ØØ ØØ FF D2 1E
[SK5]   111: Ø7 8B 3E 19 ØØ BØ 1D AA 58 AB 58 8B 36 17 ØØ
[SU5]   112: 8A 44 FF 3C 23 75 Ø6 BD ØØ Ø8 EB 34 9Ø E9 FF FA
[LK5]   113: AC 3C Ø9 75 Ø5 BØ 2Ø 88 44 FF C3 BD ØØ Ø2 39 Ø6
[CB5]   114: 1B ØØ 74 1C 8Ø 3E 25 ØØ Ø1 74 ØF 8Ø 3E 28 ØØ Ø1
[6R5]   115: 75 Ø8 C6 Ø6 25 ØØ Ø1 E9 7B F9 BØ 1A AA BD FF ØØ
[CP5]   116: 8B 36 ØF ØØ 89 2C C6 Ø6 25 ØØ Ø2 8B CF 2B CE 8B
[336]   117: F7 4E BF FF FF FD F2 A4 FC 47 FF 36 1D ØØ FF 36
[YL5]   118: 1B ØØ 8C CB 83 EB 1E 8E DB 89 3E ØØ ØØ 8F Ø6 Ø2
[FK5]   119: ØØ 8F Ø6 Ø4 ØØ Ø6 1F BA 2E ØØ B4 41 CD 21 83 3E
[AM5]   12Ø: 15 ØØ FF 75 29 8Ø 3D FF 75 24 8Ø 7D Ø1 ØØ C6 45
```

Listing 4. Hexdump des Präprozessors »sbasic.pre«

```
[CN5]   121: Ø1 Ø9 75 1A C6 45 Ø1 ØA B4 3C 2B C9 CD 21 8B D8
[816]   122: 8B D7 F7 DF 8B CF B4 4Ø CD 21 B4 3E CD 21 1F CB
[3X5]   123: 1E 8C CD 8E C5 83 ED 1E 8E DD FF 36 ØØ ØØ 33 C9
[9P5]   124: BA Ø2 ØØ B4 3C CD 21 8B D8 BE 5Ø ØØ 33 FF 8B 2C
[BZ5]   125: 81 E5 FF 7F 81 ED EØ Ø1 AD 8Ø FC FF 74 1A 8Ø FC
[UU5]   126: 8Ø 72 F5 25 FF 7F 8B ØC 81 E1 FF 7F 2B C8 2B C5
[VU5]   127: 56 8B FØ F2 A4 5E EB EØ BA CØ Ø1 83 C7 2Ø 8B CF
[HS5]   128: B4 4Ø CD 21 8C CD 81 C5 84 Ø3 8E DD BA B8 ØB 59
[NF5]   129: 2B CA EB 92 ØØ 8Ø 45 4E 44 2Ø 2Ø 81 46 4F 52 2Ø
[ZA5]   13Ø: 2Ø 82 4E 45 58 54 2Ø 83 44 41 54 41 2Ø 84 49 4E
[U95]   131: 5Ø 55 54 2Ø 2Ø 85 44 49 4D 2Ø 2Ø 86 52 45 41 44
[ND5]   132: 2Ø 87 4C 45 54 2Ø 2Ø 88 47 4F 54 4F 2Ø 89 52 55
[AF5]   133: 4E 2Ø 2Ø 8A 49 46 2Ø 8B 52 45 53 54 4F 52 45 2Ø
[CE5]   134: 2Ø 8C 47 4F 53 55 42 2Ø 2Ø 8D 52 45 54 55 52 4E
[RC5]   135: 2Ø 8E 52 45 4D 2Ø 2Ø 8F 53 54 4F 5Ø 2Ø 9Ø 5Ø 52
[6B5]   136: 49 4E 54 2Ø 2Ø 91 43 4C 45 41 52 2Ø 2Ø 92 4C 49
[2C5]   137: 53 54 2Ø 93 4E 45 57 2Ø 2Ø 94 4F 4E 2Ø 95 57 41
[BD5]   138: 49 54 2Ø 96 44 45 46 2Ø 2Ø 97 5Ø 4F 4B 45 2Ø 98
[ØF5]   139: 43 4F 4E 54 2Ø 99 4F 55 54 2Ø 2Ø 9C 4C 5Ø 52 49
[XH5]   14Ø: 4E 54 2Ø 9D 4C 4C 49 53 54 2Ø 2Ø 9E 57 49 44 54
[PD5]   141: 48 2Ø 2Ø AØ 45 4C 53 45 2Ø A1 54 52 4F 4E 2Ø A2
[V95]   142: 54 52 4F 46 46 2Ø 2Ø A3 53 57 41 5Ø 2Ø A4 45 52
[YB5]   143: 41 53 45 2Ø 2Ø A5 45 44 49 54 2Ø A6 45 52 52 4F
[LD5]   144: 52 2Ø 2Ø A7 52 45 53 55 4D 45 2Ø A8 44 45 4C 45
[6F5]   145: 54 45 2Ø A9 41 55 54 4F 2Ø AA 52 45 4E 55 4D 2Ø
[4C5]   146: 2Ø AB 44 45 46 53 54 52 2Ø AC 44 45 46 49 4E 54
[6E5]   147: 2Ø AD 44 45 46 53 4E 47 2Ø AE 44 45 46 44 42 4C
[6G5]   148: 2Ø AF 4C 49 4E 45 2Ø BØ 57 48 49 4C 45 2Ø 2Ø B1
[8B5]   149: 57 45 4E 44 2Ø B2 49 4E 53 54 52 2Ø 2Ø D8 43 41
[ZF5]   15Ø: 4C 4C 2Ø B3 57 52 49 54 45 2Ø 2Ø B7 4F 5Ø 54 49
[AL5]   151: 4F 4E 2Ø B8 52 41 4E 44 4F 4D 49 5A 45 2Ø 2Ø B9
[SL5]   152: 4F 5Ø 45 4E 2Ø BA 43 4C 4F 53 45 2Ø 2Ø BB 4C 4F
[BG5]   153: 41 44 2Ø BC 4D 45 52 47 45 2Ø 2Ø BD 53 41 56 45
[7H5]   154: 2Ø BE 43 4F 4C 4F 52 2Ø 2Ø BF 43 4C 53 2Ø 2Ø CØ
[UE5]   155: 4D 4F 54 4F 52 2Ø 2Ø C1 42 53 41 56 45 2Ø 2Ø C2
[ME5]   156: 42 4C 4F 41 44 2Ø 2Ø C3 53 4F 55 4E 44 2Ø 2Ø C4
[G95]   157: 42 45 45 5Ø 2Ø C5 5Ø 53 45 54 2Ø C6 5Ø 52 45 53
[9E5]   158: 45 54 2Ø C7 53 43 52 45 45 4E 2Ø C8 4B 45 59 2Ø
[NG5]   159: 2Ø C9 4C 4F 43 41 54 45 2Ø CA 54 4F 2Ø CC 54 48
[PG5]   16Ø: 45 4E 2Ø CD 54 41 42 2Ø 2Ø CE 53 54 45 5Ø 2Ø CF
[KA5]   161: 55 53 52 2Ø 2Ø DØ 53 5Ø 43 2Ø 2Ø D2 4E 4F 54 2Ø
[NA5]   162: 2Ø D3 45 52 4C 2Ø 2Ø D4 45 52 52 2Ø 2Ø D5 53 54
[XE5]   163: 52 49 4E 47 24 2Ø 2Ø D6 55 53 49 4E 47 2Ø 2Ø D7
[NH5]   164: 56 41 52 5Ø 54 52 2Ø DA 43 53 52 4C 49 4E 2Ø DB
[AL5]   165: 5Ø 4F 49 4E 54 2Ø 2Ø DC 4F 46 46 2Ø 2Ø DD 49 4E
[NN5]   166: 4B 45 59 24 2Ø DE 41 4E 44 2Ø 2Ø EE 4F 52 2Ø EF
[5B5]   167: 58 4F 52 2Ø 2Ø FØ 45 51 56 2Ø 2Ø F1 49 4D 5Ø 2Ø
[VB5]   168: 2Ø F2 4D 4F 44 2Ø 2Ø F3 43 56 49 2Ø 2Ø 81 43 56
[ØA5]   169: 53 2Ø 2Ø 82 43 56 44 2Ø 2Ø 83 4D 4B 49 24 2Ø 84
[1E5]   17Ø: 4D 4B 53 24 2Ø 85 4D 4B 44 24 2Ø 86 45 58 54 45
[AD5]   171: 52 52 2Ø 8B 46 49 4C 45 53 2Ø 2Ø 81 46 49 45 4C
[YB5]   172: 44 2Ø 2Ø 82 53 59 53 54 45 4D 2Ø 83 4E 41 4D 45
[3A5]   173: 2Ø 84 4C 53 45 54 2Ø 85 52 53 45 54 2Ø 86 4B 49
[DB5]   174: 4C 4C 2Ø 87 5Ø 55 54 2Ø 2Ø 88 47 45 54 2Ø 2Ø 89
[KK5]   175: 52 45 53 45 54 2Ø 2Ø 8A 43 4F 4D 4D 4F 4E 2Ø 8B
[FE5]   176: 43 48 41 49 4E 2Ø 2Ø 8C 44 41 54 45 24 2Ø 2Ø 8D
[FF5]   177: 54 49 4D 45 24 2Ø 2Ø 8E 5Ø 41 49 4E 54 2Ø 2Ø 8F
[K95]   178: 43 4F 4D 2Ø 2Ø 9Ø 43 49 52 43 4C 45 2Ø 91 44 52
[AB5]   179: 41 57 2Ø 92 5Ø 4C 41 59 2Ø 93 54 49 4D 45 52 2Ø
[Y95]   18Ø: 2Ø 94 45 52 44 45 56 2Ø 2Ø 95 49 4F 43 54 4C 2Ø
[RA5]   181: 2Ø 96 43 48 44 49 52 2Ø 2Ø 97 4D 4B 44 49 52 2Ø
[1C5]   182: 2Ø 98 52 4D 44 49 52 2Ø 2Ø 99 53 48 45 4C 4C 2Ø
[PK5]   183: 2Ø 9A 45 4E 56 49 52 4F 4E 2Ø 2Ø 9B 56 49 45 57
[EH5]   184: 2Ø 9C 57 49 4E 44 4F 57 2Ø 9D 5Ø 4D 41 5Ø 2Ø 9E
[RA5]   185: 5Ø 41 4C 45 54 54 45 2Ø 2Ø 9F 4C 43 4F 5Ø 59 2Ø
[NA5]   186: 2Ø AØ 43 41 4C 4C 53 2Ø 2Ø A1 5Ø 43 4F 5Ø 59 2Ø
[SL5]   187: 2Ø A5 4C 4F 43 4B 2Ø A7 55 4E 4C 4F 43 4B 2Ø A8
[BA5]   188: 4C 45 46 54 24 2Ø 2Ø 81 52 49 47 48 54 24 2Ø 82
[XC5]   189: 4D 49 44 24 2Ø 83 53 47 4E 2Ø 2Ø 84 49 4E 54 2Ø
[E95]   19Ø: 2Ø 85 41 42 53 2Ø 2Ø 86 53 51 52 2Ø 2Ø 87 52 4E
[KD5]   191: 44 2Ø 2Ø 88 53 49 4E 2Ø 2Ø 89 4C 4F 47 2Ø 2Ø 8A
[NC5]   192: 45 58 5Ø 2Ø 2Ø 8B 43 4F 53 2Ø 2Ø 8C 54 41 4E 2Ø
[GF5]   193: 2Ø 8D 41 54 4E 2Ø 2Ø 8E 46 52 45 2Ø 2Ø 8F 49 4E
[C95]   194: 5Ø 2Ø 2Ø 9Ø 5Ø 4F 53 2Ø 2Ø 91 4C 45 4E 2Ø 2Ø 92
[E85]   195: 53 54 52 24 2Ø 93 56 41 4C 2Ø 2Ø 94 41 53 43 2Ø
[DA5]   196: 2Ø 95 43 48 52 24 2Ø 96 5Ø 45 45 4B 2Ø 97 53 5Ø
[YB5]   197: 41 43 45 24 2Ø 98 4F 4B 54 24 2Ø 99 48 45 58 24
[DE5]   198: 2Ø 9A 4C 5Ø 4F 53 2Ø 9B 43 49 4E 54 2Ø 9C 43 53
[FK5]   199: 4E 47 2Ø 9D 43 44 42 4C 2Ø 9E 46 49 58 2Ø 2Ø 9F
[G85]   2ØØ: 5Ø 45 4E 2Ø 2Ø AØ 53 54 49 43 4B 2Ø 2Ø A1 53 54
[2D5]   2Ø1: 52 49 47 2Ø 2Ø A2 45 4F 46 2Ø 2Ø A3 4C 4F 43 2Ø
[ZZ2]   2Ø2: 2Ø A4 4C 4F 46 2Ø 2Ø A5 ØØ
```

Listing 4. Hexdump des Präprozessors »sbasic.pre«

```
[5U4]     1: ;-----------------------------------------
[TS6]     2: ; Programm-Modul: SBPRE.ASM (SBASIC.PRE)
[9M2]     3: ; Version       : 2.Ø
[794]     4: ; Autor         : Peter Knappe
[ØE4]     5: ; (C) 1991 DMV Widuch GmbH &Co KG
[P3Ø]     6: ;
[R52]     7: ; Funktionen:
[C37]     8: ; 1. schnelle Vorübersetzung ASCII-/LABEL-
[GF3]     9: ;    BASIC nach GWBASIC
[3F6]    1Ø: ; 2. Fehlerzeilen-Ausgabe für SBASIC.BAS
[F26]    11: ; 3. Binden und Ausgabe der EXE-Datei
[P3Ø]    12: ;
[UL6]    13: ; Laden und Aufruf erfolgt von SBASIC.BAS
[DF6]    14: ; (SBASIC.EXE);aufgerufen wird SBASIC.BIN
[P3Ø]    15: ;
[554]    16: ; Assemblieren und Linken:
[5U1]    17: ; mit MASM:
[1B2]    18: ;    masm sbpre;
[NB2]    19: ;    link sbpre;
[115]    2Ø: ;    exe2bin sbpre.exe sbasic.pre
[FK2]    21: ;    del sbpre.obj
[2L2]    22: ;    del sbpre.exe
[VP3]    23: ; mit Quick-Assembler:
```

```
[BM2]    24: ;    qcl sbpre.asm
[5G4]    25: ;    weiter mit exe2bin s.o.
[HD1]    26: ; mit A86:
[MT4]    27: ;    a86 sbpre.asm to sbasic.pre
[4Ø3]    28: ;    del sbasic.sym
[P3Ø]    29: ;
[5U4]    3Ø: ;-----------------------------------------
[ØØØ]    31:
[B52]    32: CODE    SEGMENT
[3U4]    33:         ASSUME CS:CODE,DS:CODE,ES:CODE
[G11]    34:         ORG   Ø
[ØØØ]    35:
[B22]    36:     ;BLOAD-Header
[M71]    37:         DB    ØFDH
[A13]    38:         DW    Ø,Ø,OFFSET LBLS
[ØØØ]    39:
[4U4]    4Ø:     ;Einsprung zum Preprocessor
[RC1]    41:         JMP   PRE
[1B4]    42:     ;Einsprung zur EXE-Ausgabe
[FC1]    43:         JMP   LNK
[ØØØ]    44:
[5U4]    45: ;-----------------------------------------
[KWØ]    46: ; DATA
[5U4]    47: ;-----------------------------------------
[K96]    48: BUFØ    DW    OFFSET LBLS   ;Ende LABEL-Tab.
[PW5]    49: BUF1    DW    Ø     ;Adr.GWBASIC-Code-Anfang
[AH5]    5Ø: BUF2    DW    Ø     ;Adr.GWBASIC-Code-Ende
[896]    51: BUF3    DW    Ø     ;Adresse Quellcode-Anfang
[1H6]    52: SAVØ    DW    Ø     ;Adr.d.Übergabe-Parameter
[AY5]    53: SVSI    DW    Ø     ;Sicherung Register SI
[KU5]    54: SVDI    DW    Ø     ;Sicherung Register DI
[7K6]    55: LONR    DW    Ø     ;letzte log.Zeilen-Nummer
[GX5]    56: PHNR    DW    Ø     ;letzte phys.Zeilen-Nr.
[VL5]    57: ZBEG    DW    Ø     ;GWBASIC-Zeilen-Anfang
[MX5]    58: SLBL    DW    Ø     ;Adr.e.in Tab.gef.LABELS
[SV5]    59: SRET    DW    Ø     ;Sicherung RET-Adresse
[884]    6Ø: PASS    DB    Ø     ;Ø=1.Durchlauf
[9W3]    61:                     ;1=2.Durchlauf bei LABELS
[834]    62:                     ;2=Fehlerzeilen-Ausgabe
[EY5]    63: LBL1    DB    Ø     ;1=Zeilenbeg.m.pot.LABEL
[DS5]    64: LBL2    DB    Ø     ;1=Zeile beginnt m.LABEL
[RL5]    65: LBL3    DB    Ø     ;1=LABEL-Tab. generiert
[X86]    66: SPCZ    DB    Ø     ;1=mehrf.Leerz.überlesen
[8N5]    67: REMZ    DB    Ø     ;1=Zeile enthält REM / '
[L16]    68: TKZ1    DB    Ø     ;Ø=TOKEN innerhalb Zeile
[C14]    69:                     ;1=TOKEN am Zeilenbeginn
[LA6]    7Ø: TKZ2    DB    Ø     ;Ø=Leerz.vor TOKEN setzen
[RZ3]    71:                     ;1=kein Leerz.vor TOKEN
[176]    72: NUMZ    DB    Ø     ;ØEH=Zeilen-Ref.-Nr.folgt
[7V3]    73:                     ;1CH=norm. INTEGER folgt
[WØ5]    74:     ;vorl. GWBASIC-File-Name ("/GWB")
[9U2]    75: FGWB    DB    "###.GWB",Ø
[8A6]    76:     ;Tabelle der 1-Byte-Operationszeichen
[KP3]    77: TBX1    DB    ">=<+-*/^------\"
[NK6]    78:     ;Tabelle d.Zeichen ohne Leerz.vor TOKEN
[HZ2]    79: TBX2    DB    " (:;,",ØF4H
[NP4]    8Ø:         DW    ØE6E7H,ØE8E9H,ØEAEBH,ØECEDH
[ØØØ]    81:
[5U4]    82: ;-----------------------------------------
[S75]    83: ; PRE - START DES PREPROCESSORS
[5U4]    84: ;-----------------------------------------
[FC2]    85: PRE:    PUSH  DS
[AA6]    86:     ;Durchlauf für Fehlerzeilen-Ausgabe?
[S44]    87:         CMP   BYTE PTR CS:[PASS],2
[SA1]    88:         JE    PRE3
[FB5]    89:     ;nein - Datei SBASIC.BIN öffnen
[4V1]    9Ø:         MOV   BP,CS
[PL1]    91:         SUB   BP,3Ø
[ØV1]    92:         MOV   DS,BP
[NL1]    93:         ADD   BP,93Ø
[2H1]    94:         MOV   SI,2
[LV1]    95: PRE1:   LODSB
[FL1]    96:         OR    AL,AL
[NK1]    97:         JNZ   PRE1
[WV1]    98:         MOV   DX,SI
[432]    99:         MOV   AX,3DØØH
[N81]   1ØØ:         INT   21H
[UV1]   1Ø1:         MOV   BX,AX
[ØV1]   1Ø2:         MOV   DS,BP
[5S4]   1Ø3:     ;Fehler, wenn nicht gefunden
[2V1]   1Ø4:         MOV   BP,1ØØH
[KA1]   1Ø5:         JB    PRE4
[9M6]   1Ø6:     ;Datei SBASIC.BIN einlesen u. schließen
[4V1]   1Ø7:         MOV   BP,CS
[ØV1]   1Ø8:         MOV   DS,BP
[TU1]   1Ø9:         SUB   DX,DX
[2Z1]   11Ø:         MOV   CX,144ØØ
[WT1]   111:         MOV   AH,3FH
[N81]   112:         INT   21H
[ST1]   113:         MOV   AH,3EH
[N81]   114:         INT   21H
[RH5]   115:     ;SBASIC-BIN Adressblock sichern
[7M1]   116:         SUB   BP,25
[4V1]   117:         MOV   ES,BP
[RV1]   118:         SUB   SI,SI
[ES1]   119:         SUB   DI,DI
[9W1]   12Ø: PRE2:   LODSW
[ZYØ]   121:         STOSW
[451]   122:         INC   AX
[SK1]   123:         JNZ   PRE2
```

```
[S86]  124:    ;SBASIC-BIN Bibliothekscode sichern
[WP1]  125:        MOV   DI,4ØØ
[ØE1]  126:        SAR   CX,1
[T42]  127:        REPNZ MOVSW
[8B3]  128:    ;Quelldatei öffnen
[FN2]  129: PRE3:  MOV   BP,CS
[PL1]  13Ø:        SUB   BP,3Ø
[ØV1]  131:        MOV   DS,BP
[NL1]  132:        ADD   BP,93Ø
[NF1]  133:        MOV   DX,2
[432]  134:        MOV   AX,3DØØH
[N81]  135:        INT   21H
[UV1]  136:        MOV   BX,AX
[4T5]  137:    ;Param. sichern ("/GWB",Fehler-Z.-Nr.)
[BL2]  138:        MOV   DX,DS:[Ø]
[ØV1]  139:        MOV   DS,BP
[4V1]  14Ø:        MOV   ES,BP
[Ø43]  141:        MOV   DS:[SAVØ],DX
[WM3]  142:    ;Quelldatei gefunden?
[5V1]  143:        MOV   BP,2ØØH
[1G1]  144:        JNB   PRE5
[TD5]  145:    ;nein - zur Fehler-Exit-Routine
[MK2]  146: PRE4:  SUB   DI,DI
[5H1]  147:        JMP   EXT2
[PW5]  148:    ;ja - Aktualisierung Puffer-Zeiger
[1X3]  149: PRE5:  MOV   DX,DS:[BUFØ]
[L44]  15Ø:        CMP   BYTE PTR DS:[PASS],Ø
[GG1]  151:        JNE   PRE6
[UR1]  152:        ADD   DX,4ØØH
[6W3]  153: PRE6:  MOV   DS:[BUF1],DX
[G51]  154:        INC   DX
[P13]  155:        MOV   DS:[BUF2],DX
[5V1]  156:        ADD   DX,3FFH
[T13]  157:        MOV   DS:[BUF3],DX
[KV5]  158:    ;max. Quellcode-Größe ermitteln
[7W1]  159:        MOV   CX,DX
[GB1]  16Ø:        ADD   CX,2
[X71]  161:        NOT   CX
[R91]  162:        PUSH  CX
[Y37]  163:    ;Quellcode einlesen und Datei schließen
[WT1]  164:        MOV   AH,3FH
[N81]  165:        INT   21H
[YV1]  166:        MOV   CX,AX
[ST1]  167:        MOV   AH,3EH
[N81]  168:        INT   21H
[X55]  169:    ;Fehler, wenn Datei zu groß
[971]  17Ø:        POP   AX
[CU1]  171:        CMP   CX,AX
[8V1]  172:        MOV   BP,3ØØH
[VA1]  173:        JE    PRE4
[SX1]  174:        MOV   SI,DX
[WV1]  175:        MOV   DI,DX
[SR1]  176:        ADD   DI,CX
[KH6]  177:    ;Quelldatei bereits im GWBASIC-Format?
[EC4]  178:        CMP   BYTE PTR DS:[SI],ØFEH
[YA1]  179:        JB    PRE8
[R55]  18Ø:    ;ja - evtl. "/GWB"-Kennz. anpassen
[K74]  181:        CMP   BYTE PTR DS:[SI+1],ØAH
[LG1]  182:        JNE   PRE7
[TM3]  183:        INC   BYTE PTR DS:[SI+1]
[KE2]  184: PRE7:  JMP   EXT3
[ØU3]  185:    ;Puffer abschließen
[9Ø3]  186: PRE8:  MOV   AX,ØAØDH
[ZYØ]  187:        STOSW
[6T1]  188:        MOV   AL,1AH
[NXØ]  189:        STOSB
[Ø35]  19Ø:    ;Zeilenzähler zurücksetzen
[D84]  191:        MOV   WORD PTR DS:[LONR],Ø
[384]  192:        MOV   WORD PTR DS:[PHNR],Ø
[ØØØ]  193:
[5U4]  194: ;-----------------------------------------
[1X6]  195: ; ZNR - ZEILENNUMMER-VERARB./-GENERIERUNG
[5U4]  196: ;-----------------------------------------
[AZ1]  197:    ;neue Zeile
[R25]  198: ZNR:   INC   WORD PTR DS:[PHNR]
[Z56]  199:    ;Duchlauf für Fehler-Zeilen-Ausgabe?
[T44]  2ØØ:        CMP   BYTE PTR DS:[PASS],2
[5K1]  2Ø1:        JNE   ZNR3
[YH4]  2Ø2:    ;ja - Fehler-Zeile erreicht?
[Ø43]  2Ø3:        MOV   DX,DS:[PHNR]
[E23]  2Ø4:        CMP   DS:[SAVØ],DX
[8H5]  2Ø5:    ;nein - zum Restzeilen-Überlesen
[2K1]  2Ø6:        JNE   ZNR2
[RG5]  2Ø7:    ;ja - Ausgabe Fehler-Zeile und Ende
[FX1]  2Ø8: ZNR1:  LODSB
[X81]  2Ø9:        INT   29H
[XR1]  21Ø:        CMP   AL,ØDH
[ZH1]  211:        JNE   ZNR1
[EH1]  212:        JMP   EXT5
[LH3]  213:    ;Restzeile überlesen
[GX1]  214: ZNR2:  LODSB
[HR1]  215:        CMP   AL,ØAH
[2K1]  216:        JNE   ZNR2
[5R2]  217:    ;zur neuen Zeile
[NE1]  218:        JMP   ZNR
[K65]  219: ZNR3:  MOV   WORD PTR DS:[LBL1],Ø
[T64]  22Ø:        MOV   BYTE PTR DS:[REMZ],Ø
[N64]  221:        MOV   BYTE PTR DS:[SPCZ],1
[D34]  222:        MOV   BYTE PTR DS:[TKZ1],1
[CK4]  223:        MOV   BYTE PTR DS:[NUMZ],1CH
[L13]  224:        MOV   DI,DS:[BUF2]
[354]  225:    ;Folgeadresse vorbelegen
[4U1]  226:        SUB   AX,AX
[ZYØ]  227:        STOSW
[KZ2]  228:    ;Quelldaten-Ende?
[E34]  229:        CMP   BYTE PTR DS:[SI],1AH
[TC1]  23Ø:        JE    ZNR4
[T74]  231:        CMP   BYTE PTR DS:[SI+2],1AH
[BK1]  232:        JNE   ZNR5
[MM3]  233:    ;ja - zur EXIT-Routine
[5D2]  234: ZNR4:  JMP   EXT
[7B4]  235:    ;nein - Zeilennummer lesen
[NR2]  236: ZNR5:  SUB   CX,CX
[EV1]  237:        MOV   BP,5ØØH
[PX2]  238:    ;Zeichen einlesen
[AL2]  239: ZNR6:  CALL  GETB
[Ø24]  24Ø:    ;Leerzeichen überlesen
[LN1]  241:        CMP   AL,2ØH
[ZC1]  242:        JE    ZNR6
[MR1]  243:        SUB   AH,AH
[PN1]  244:        SUB   AL,3ØH
[XU1]  245:        XCHG  AX,CX
[X33]  246:    ;gültige Ziffer?
[8E1]  247:        CMP   CL,9
[PC1]  248:        JA    ZNR7
[BØ4]  249:    ;ja - Ziffer kumulieren
[VT1]  25Ø:        MOV   DX,ØAH
[191]  251:        IMUL  DX
[F13]  252:    ;Überlauf (>65535)?
[ØP1]  253:        OR    DX,DX
[KL1]  254:        JNE   ZNRB
[VR1]  255:        ADD   AX,CX
[UD1]  256:        JB    ZNRB
[XW4]  257:    ;nein - zum nächsten Zeichen
[XU1]  258:        XCHG  AX,CX
[RL1]  259:        JMP   ZNR6
[Z84]  26Ø:    ;Zeilennummer vorhanden?
[XL2]  261: ZNR7:  OR    AX,AX
[MK1]  262:        JNE   ZNR8
[AD4]  263:    ;nein - Folgenummer bilden
[443]  264:        MOV   AX,DS:[LONR]
[451]  265:        INC   AX
[EU2]  266:    ;Überlauf (65536)?
[2G1]  267:        JZ    ZNRB
[UD6]  268:    ;Kennzeichnung für potentiellen LABEL
[914]  269:        MOV   BYTE PTR DS:[LBL1],1
[SZ5]  27Ø:    ;vorhergehende Z.-Nr. < neuer Z.-Nr.?
[HZ3]  271: ZNR8:  CMP   DS:[LONR],AX
[ZC1]  272:        JB    ZNR9
[PC4]  273:    ;nein - zur Fehler-Routine
[BV1]  274:        MOV   BP,4ØØH
[443]  275:        MOV   AX,DS:[LONR]
[451]  276:        INC   AX
[YM1]  277:        JMP   ZNRC
[1G6]  278:    ;ja - Zeilennummer sichern und ausgeben
[KØ4]  279: ZNR9:  MOV   DS:[LONR],AX
[NXØ]  28Ø:        STOSB
[AT1]  281:        MOV   AL,AH
[NXØ]  282:        STOSB
[4L3]  283:    ;Zeilenanfang sichern
[C91]  284:        PUSH  DI
[KR2]  285:        POP   DS:[ZBEG]
[RL4]  286:    ;Folgezeichen restaurieren
[ØU1]  287:        MOV   AL,CL
[7L1]  288:        ADD   AL,3ØH
[D75]  289:    ;Leerz. in REM-Zeile umwandeln?
[XR1]  29Ø:        CMP   AL,ØDH
[FL1]  291:        JNE   ZNRA
[AB4]  292:    ;ja - REM-Zeile generieren
[F91]  293:        PUSH  AX
[Ø52]  294:        MOV   AX,8F3AH
[ZYØ]  295:        STOSW
[GX1]  296:        MOV   AL,ØD9H
[NXØ]  297:        STOSB
[971]  298:        POP   AX
[D95]  299:    ;zur Zeilen-Code Vorprüfung
[LF2]  3ØØ: ZNRA:  JMP   ZCD3
[FC5]  3Ø1:    ;Zeilennummer-Überlauf-Fehler
[SR2]  3Ø2: ZNRB:  SUB   AX,AX
[CZ4]  3Ø3:    ;Fehler-Zeilen-Nummer sichern
[WØ4]  3Ø4: ZNRC:  MOV   DS:[LONR],AX
[ZA4]  3Ø5:    ;zur Fehler-Exit-Routine
[5H1]  3Ø6:        JMP   EXT2
[ØØØ]  3Ø7:
[5U4]  3Ø8: ;-----------------------------------------
[HF4]  3Ø9: ; ZCD - ZEILEN-CODE PRÜFEN
[5U4]  31Ø: ;-----------------------------------------
[XU1]  311: ZCD:   STOSB
[A34]  312:        MOV   BYTE PTR DS:[TKZ1],Ø
[CG2]  313: ZCD1:  CALL  GETB
[535]  314: ZCD2:  MOV   BYTE PTR DS:[LBL1],Ø
[6C6]  315:    ;1.Durchlauf, LABELS vorh., pot. LABEL?
[965]  316: ZCD3:  CMP   BYTE PTR DS:[PASS],Ø
[1H1]  317:        JNE   ZCD4
[ZZ3]  318:        CMP   BYTE PTR DS:[LBL3],1
[1H1]  319:        JNE   ZCD4
[MZ3]  32Ø:        CMP   BYTE PTR DS:[LBL1],Ø
```

Listing 5. Alternativ zu Listing 4 kann auch die Assembler-Quelltextversion »sbpre.asm« eingegeben werden

```
[1H1]  321:         JNE   ZCD4
[CD4]  322:     ;ja - Restzeile überlesen
[BL1]  323:         JMP   ZNR2
[PK4]  324:     ;nein - Zeilenende erreicht?
[DT2]  325: ZCD4:   CMP   AL,ØDH
[4H1]  326:         JNE   ZCD5
[2T4]  327:     ;ja - zur Zeilenende-Routine
[SE1]  328:         JMP   ZEX
[KB4]  329:     ;nein - Quelldatei gültig?
[C72]  33Ø: ZCD5:   JA    ZCD6
[TD5]  331:     ;nein - zur Fehler-Exit-Routine
[5V1]  332:         MOV   BP,2ØØH
[5H1]  333:         JMP   EXT2
[5E7]  334:     ;ja - überzählige Leerzeichen überlesen
[RR2]  335: ZCD6:   CMP   AL,2ØH
[AH1]  336:         JNE   ZCD7
[354]  337:         CMP   BYTE PTR DS:[SPCZ],1
[AH1]  338:         JNE   ZCD7
[1K1]  339:         JMP   ZCD1
[Z75]  34Ø: ZCD7:   MOV   BYTE PTR DS:[SPCZ],1
[3W2]  341:     ;folgt TEXT-Code?
[VN1]  342:         CMP   AL,22H
[DH1]  343:         JNE   ZCD8
[UD4]  344:     ;ja - zur Text-Übertragung
[BE1]  345:         JMP   TXT
[8L2]  346:     ;folgt HEX-Code?
[BS2]  347: ZCD8:   CMP   AL,26H
[8K1]  348:         JNE   ZCDA
[VZ3]  349:         CMP   BYTE PTR DS:[SI],"H"
[1C1]  35Ø:         JE    ZCD9
[VZ3]  351:         CMP   BYTE PTR DS:[SI],"h"
[8K1]  352:         JNE   ZCDA
[BN4]  353:     ;ja - zur HEX-Tokenisierung
[K82]  354: ZCD9:   JMP   HEX
[HK2]  355:     ;folgt REM-Code?
[RS2]  356: ZCDA:   CMP   AL,27H
[BK1]  357:         JNE   ZCDB
[MF4]  358:     ;ja - zur REM-Übertragung
[2D1]  359:         JMP   REM
[3T2]  36Ø:     ;Komma überlesen
[WT2]  361: ZCDB:   CMP   AL,2CH
[EK1]  362:         JNE   ZCDC
[ED1]  363:         JMP   ZCD
[FD5]  364:     ;folgt 1-Byte-Operations-TOKEN?
[7C2]  365: ZCDC:   PUSH  DI
[ZB3]  366:         MOV   DI,OFFSET TBX1
[BU1]  367:         MOV   CX,ØFH
[712]  368:         REPNZ SCASB
[671]  369:         POP   DI
[8N1]  37Ø:         JNZ   ZCDE
[WG5]  371:     ;ja - Operations-TOKEN generieren
[LX1]  372:         MOV   AL,ØF4H
[ES1]  373:         SUB   AL,CL
[GR6]  374:     ;überlesenes Leerzeichen zurückholen
[133]  375:         CMP   DS:[ZBEG],DI
[2D1]  376:         JE    ZCDD
[B54]  377:         CMP   BYTE PTR DS:[DI-1],2ØH
[HK1]  378:         JNE   ZCDD
[W31]  379:         DEC   DI
[1V6]  38Ø:     ;INTEGER-Kennung setzen, Code speichern
[UR5]  381: ZCDD:   MOV   BYTE PTR DS:[NUMZ],1CH
[ED1]  382:         JMP   ZCD
[5K4]  383:     ;"(" nach TAB/SPC übergehen
[PH2]  384: ZCDE:   CMP   AL,"("
[UK1]  385:         JNE   ZCDG
[YB4]  386:         CMP   BYTE PTR DS:[DI-1],ØCEH
[8D1]  387:         JE    ZCDF
[FB4]  388:         CMP   BYTE PTR DS:[DI-1],ØD2H
[UK1]  389:         JNE   ZCDG
[ES6]  39Ø:     ;zur weiteren Zeilen-Code-Verarbeitung
[VC2]  391: ZCDF:   JMP   ZCD1
[YZ3]  392:     ;folgt numerischer Code?
[AK2]  393: ZCDG:   CMP   AL,"."
[ED1]  394:         JE    ZCDH
[3R3]  395:     ;Zeichen < "Ø" überlesen
[PN1]  396:         CMP   AL,3ØH
[TC1]  397:         JB    ZCDD
[UP1]  398:         CMP   AL,39H
[5D1]  399:         JA    ZCDI
[4C5]  4ØØ:     ;ja - zur INTEGER-Tokenisierung
[8B2]  4Ø1: ZCDH:   JMP   NUM
[N86]  4Ø2:     ;nein - Nicht-Alpha-Zeichen überlesen
[HS2]  4Ø3: ZCDI:   CMP   AL,41H
[TC1]  4Ø4:         JB    ZCDD
[4S1]  4Ø5:         CMP   AL,5BH
[H61]  4Ø6:         JB    TKN
[3P1]  4Ø7:         CMP   AL,61H
[TC1]  4Ø8:         JB    ZCDD
[6S1]  4Ø9:         CMP   AL,7AH
[PC1]  41Ø:         JA    ZCDD
[LX5]  411:     ;Code ist TOKEN, LABEL oder VARIABLE
[KD1]  412:         JMP   TKN
[ØØØ]  413:
[5U4]  414: ;-----------------------------------------
[TH4]  415: ; ZEX - ZEILEN-ENDE-ROUTINE
[5U4]  416: ;-----------------------------------------
[6S6]  417:     ;2.Durchlauf und Zeilen-LABEL vorhanden?
[M75]  418: ZEX:    CMP   BYTE PTR DS:[PASS],1
[EK1]  419:         JNE   ZEX5
[VZ3]  42Ø:         CMP   BYTE PTR DS:[LBL2],1
```

```
[EK1]  421:         JNE   ZEX5
[ZF6]  422:     ;ja - "'*LABEL*" anfügen z.Dokumentation
[754]  423:         CMP   BYTE PTR DS:[REMZ],Ø
[KE1]  424:         JZ    ZEX1
[432]  425:         MOV   AX,5B2ØH
[TM1]  426:         JNZ   ZEX2
[333]  427: ZEX1:   MOV   AX,8F3AH
[ZYØ]  428:         STOSW
[V32]  429:         MOV   AX,2AD9H
[MZ1]  43Ø: ZEX2:   STOSW
[T33]  431:         MOV   BX,DS:[SLBL]
[VD1]  432:         SUB   BX,2
[US3]  433: ZEX3:   MOV   AL,DS:[BX]
[LN1]  434:         CMP   AL,2ØH
[7H1]  435:         JBE   ZEX4
[NXØ]  436:         STOSB
[851]  437:         INC   BX
[HL1]  438:         JMP   ZEX3
[YW2]  439: ZEX4:   MOV   AL,2AH
[NXØ]  44Ø:         STOSB
[PT2]  441:     ;Zeilenende-Code
[MM2]  442: ZEX5:   SUB   AL,AL
[NXØ]  443:         STOSB
[6W6]  444:     ;Zeilenende-Pointer a. Zeilenende setzen
[U13]  445:         MOV   BX,DS:[BUF2]
[YL2]  446:         MOV   DS:[BX],DI
[9Ø3]  447:         MOV   DS:[BUF2],DI
[MH1]  448:         CALL  GETB
[5R2]  449:     ;zur neuen Zeile
[NE1]  45Ø:         JMP   ZNR
[ØØØ]  451:
[5U4]  452: ;-----------------------------------------
[4H6]  453: ; TKN - BASIC TOKENISIEREN / LABEL VERARB.
[5U4]  454: ;-----------------------------------------
[TØ3]  455: TKN:    MOV   AH,TKZ1
[633]  456:         MOV   DS:[TKZ2],AH
[5P2]  457:         MOV   AH,DS:[SI]
[GA2]  458:         AND   AX,ØDFDFH
[PØ3]  459:     ;Funktion "FNxxx"?
[9U1]  46Ø:         CMP   AX,"NF"
[WG1]  461:         JNE   TKN1
[7W3]  462:     ;ja - FN-TOKEN speichern
[8X1]  463:         MOV   AL,ØD1H
[NXØ]  464:         STOSB
[ES6]  465:     ;zur weiteren Zeilen-Code-Verarbeitung
[MH1]  466:         CALL  GETB
[1K1]  467:         JMP   ZCD1
[LY5]  468:     ;Zeichenkette übertragen / checken
[FE2]  469: TKN1:   CALL  TCH
[F91]  47Ø:         PUSH  AX
[8B1]  471:         PUSH  SI
[R91]  472:         PUSH  CX
[8N2]  473:         MOV   AX,DS:[DI]
[C91]  474:         PUSH  DI
[9T2]  475:     ;1-Byte-VARIABLE?
[XL1]  476:         OR    CL,CL
[GB1]  477:         JE    TKN2
[CB4]  478:     ;nein - TOKEN / LABEL suchen
[DC1]  479:         CALL  TTB
[KD1]  48Ø:         JZ    TKN3
[X76]  481:     ;nicht gefunden, Ref.-LABEL erwartet?
[ZG4]  482:         CMP   BYTE PTR DS:[NUMZ],ØEH
[LL1]  483:         JNZ   TKN2
[9H6]  484:     ;ja - zur Fehler-Routine bei 2.Durchlauf
[HV1]  485:         MOV   BP,6ØØH
[P44]  486:         CMP   BYTE PTR DS:[PASS],1
[PB1]  487:         JE    TKN4
[946]  488:     ;Kennzeichnung für vorhandenen LABEL
[H14]  489:         MOV   BYTE PTR DS:[LBL3],1
[266]  49Ø:     ;Zeichenkette ist LABEL oder VARIABLE
[3G2]  491: TKN2:   JMP   TKNI
[LL4]  492:     ;TOKEN oder LABEL gefunden?
[894]  493: TKN3:   CMP   BX,OFFSET LBLS
[TB1]  494:         JB    TKN8
[VN6]  495:     ;LABEL - entspricht dieser Referenz-Nr.?
[MZ3]  496:         CMP   BYTE PTR DS:[LBL1],Ø
[WB1]  497:         JE    TKN6
[KU5]  498:     ;nein - LABEL entspricht Zeilen-Nr.
[D14]  499:         MOV   BYTE PTR DS:[LBL2],1
[P44]  5ØØ:         CMP   BYTE PTR DS:[PASS],1
[ZB1]  5Ø1:         JE    TKN7
[XY3]  5Ø2:     ;LABEL bereits definiert
[MV1]  5Ø3:         MOV   BP,7ØØH
[C91]  5Ø4:         PUSH  DI
[ZA4]  5Ø5:     ;zur Fehler-Exit-Routine
[R72]  5Ø6: TKN4:   POP   DI
[H92]  5Ø7: TKN5:   POP   CX
[291]  5Ø8:         POP   SI
[971]  5Ø9:         POP   AX
[5H1]  51Ø:         JMP   EXT2
[Y26]  511:     ;LABEL durch Zeilen-Nummer ersetzen
[6W2]  512: TKN6:   MOV   AL,ØEH
[NXØ]  513:         STOSB
[YP2]  514:         MOV   AX,DS:[BX]
[ZYØ]  515:         STOSW
[UH1]  516:     ;zum Ende
[L92]  517: TKN7:   POP   CX
[GM1]  518:         JMP   TKNJ
[CV6]  519:     ;TOKEN gefunden, INTEGER-Kennung setzen
[2T5]  52Ø: TKN8:   MOV   BYTE PTR DS:[NUMZ],1CH
```

```
[DA2]  521:     ;1-Byte-TOKEN?
[U93]  522:         CMP   BX,OFFSET TKFD
[MC1]  523:         JB    TKNA
[FG6]  524:     ;nein - TOKEN-Kz. ermitteln u. speichern
[TD2]  525:         MOV   AX,ØFFFDH
[V93]  526:         CMP   BX,OFFSET TKFE
[WB1]  527:         JB    TKN9
[S41]  528:         INC   AL
[W93]  529:         CMP   BX,OFFSET TKFF
[WB1]  53Ø:         JB    TKN9
[S41]  531:         INC   AL
[UY2]  532:     ;TOKEN übertragen
[5H2]  533: TKN9:   CALL  TTR
[ØV3]  534:     ;TOKEN aus Tabelle holen
[6S3]  535: TKNA:   MOV   AL,DS:[BX]
[BU1]  536:         SUB   BX,BX
[G42]  537:     ;1-Byte-TOKEN
[VK1]  538:         OR    AH,AH
[YK1]  539:         JNE   TKNG
[1U3]  54Ø:     ;ja - ist TOKEN=DEFDBL?
[ZV1]  541:         CMP   AL,ØAFH
[RV1]  542:         MOV   BP,8ØØH
[TB1]  543:         JE    TKN5
[6U2]  544:     ;ist TOKEN=THEN?
[5W1]  545:         CMP   AL,ØCDH
[3D1]  546:         JE    TKNC
[5U2]  547:     ;ist TOKEN=ELSE?
[AV1]  548:         CMP   AL,ØA1H
[FK1]  549:         JNE   TKNB
[3Y4]  55Ø:     ;ja - Doppelpunkt hinzufügen
[F91]  551:         PUSH  AX
[CT1]  552:         MOV   AL,3AH
[UY2]  553:     ;TOKEN übertragen
[DD1]  554:         CALL  TTR
[971]  555:         POP   AX
[VL1]  556:         JMP   TKNC
[1M6]  557:     ;TOKEN=Zeilen-Bezugs-Befehl (z.B.GOTO)?
[WV2]  558: TKNB:   CMP   AL,8CH
[3D1]  559:         JE    TKNC
[HV1]  56Ø:         CMP   AL,ØA8H
[3D1]  561:         JE    TKNC
[9R1]  562:         CMP   AL,89H
[3D1]  563:         JE    TKNC
[NS1]  564:         CMP   AL,8DH
[3D1]  565:         JE    TKNC
[4W1]  566:         CMP   AL,ØCCH
[NK1]  567:         JNE   TKND
[FA3]  568:     ;"GO TO" statt "GOTO"
[1C4]  569:         CMP   WORD PTR DS:[DI-2],"OG"
[NK1]  57Ø:         JNE   TKND
[VS1]  571:         MOV   AL,89H
[MD1]  572:         SUB   DI,2
[PE6]  573:     ;ja - INTEGER-Kennung f. Ref.-Nr. setzen
[HT5]  574: TKNC:   MOV   BYTE PTR DS:[NUMZ],ØEH
[HY2]  575:     ;ist TOKEN=WHILE?
[RU2]  576: TKND:   CMP   AL,ØB1H
[SK1]  577:         JNE   TKNE
[4E3]  578:     ;ja - TOKEN ergänzen
[DD1]  579:         CALL  TTR
[MX1]  58Ø:         MOV   AL,ØE9H
[VP2]  581:     ;ist TOKEN=REM?
[8W2]  582: TKNE:   CMP   AL,8FH
[VK1]  583:         JNE   TKNF
[S43]  584:     ;ja - BL auf 1 setzen
[W41]  585:         INC   BL
[1T2]  586:     ;ist TOKEN=DATA?
[LU2]  587: TKNF:   CMP   AL,84H
[YK1]  588:         JNE   TKNG
[M43]  589:     ;ja - BH auf 1 setzen
[S41]  59Ø:         INC   BH
[UY2]  591:     ;TOKEN übertragen
[LH2]  592: TKNG:   CALL  TTR
[H71]  593:         POP   CX
[291]  594:         POP   SI
[971]  595:         POP   AX
[KN1]  596:         OR    BX,BX
[UF1]  597:         JZ    TKNK
[PL1]  598:         OR    BL,BL
[AZ3]  599:     ;zur DATA-Übertragung
[NN1]  6ØØ:         JNZ   TKNH
[MB1]  6Ø1:         JMP   DAT
[BV3]  6Ø2:     ;zur REM-Übertragung
[CE2]  6Ø3: TKNH:   JMP   REM2
[266]  6Ø4:     ;Zeichenkette ist LABEL oder VARIABLE
[3A5]  6Ø5: TKNI:   MOV   BYTE PTR DS:[SPCZ],Ø
[P43]  6Ø6:         MOV   DI,DS:[SVDI]
[291]  6Ø7:         POP   SI
[H71]  6Ø8:         POP   CX
[A81]  6Ø9:     ;LABEL?
[RZ3]  61Ø:         CMP   BYTE PTR DS:[LBL1],1
[7L1]  611:         JNE   TKNJ
[7U4]  612:     ;ja - LABEL nach TOKEN-Tabelle
[5C1]  613:         CALL  TLB
[U92]  614: TKNJ:   POP   SI
[971]  615:         POP   AX
[RX2]  616:     ;DOUBLE-Variable?
[9H1]  617:         CMP   AL,"#"
[AL1]  618:         JNE   TKNK
[4V4]  619:     ;ja - zur Fehler-Exit-Routine
[RV1]  62Ø:         MOV   BP,8ØØH
```

```
[5H1]  621:         JMP   EXT2
[ES6]  622:     ;zur weiteren Zeilen-Code-Verarbeitung
[PE2]  623: TKNK:   JMP   ZCD2
[ØØØ]  624:
[5U4]  625: ;-----------------------------------------
[1P7]  626: ; TCH - ZEICHENKETTE ÜBERTRAGEN/TYP CHECKEN
[5U4]  627: ;-----------------------------------------
[EA2]  628: TCH:    PUSH  DI
[KU1]  629:         SUB   CX,CX
[Z25]  63Ø:     ;gültiges Zeichen speichern
[CW1]  631: TCH1:   STOSB
[TD7]  632:     ;nächstes Zeichen lesen u.Zähler erhöhen
[MH1]  633:         CALL  GETB
[Ø51]  634:         INC   CL
[215]  635:     ;auf gültige Zeichen checken
[NK1]  636:         CMP   AL,"."
[TA1]  637:         JE    TCH1
[TS1]  638:         CMP   AL,"_"
[TA1]  639:         JE    TCH1
[DH1]  64Ø:         CMP   AL,"$"
[TA1]  641:         JE    TCH1
[PN1]  642:         CMP   AL,3ØH
[PA1]  643:         JB    TCH3
[UP1]  644:         CMP   AL,39H
[4F1]  645:         JBE   TCH1
[6S1]  646:         CMP   AL,7AH
[LA1]  647:         JA    TCH3
[3P1]  648:         CMP   AL,61H
[LA1]  649:         JB    TCH2
[UU1]  65Ø:         AND   AL,ØDFH
[YR2]  651: TCH2:   CMP   AL,41H
[PA1]  652:         JB    TCH3
[4S1]  653:         CMP   AL,5BH
[GA1]  654:         JB    TCH1
[XA5]  655:     ;Ende der gültigen Zeichenkette
[HP5]  656: TCH3:   CMP   BYTE PTR DS:[NUMZ],ØEH
[2B1]  657:         JE    TCH4
[MZ3]  658:         CMP   BYTE PTR DS:[LBL1],Ø
[8B1]  659:         JE    TCH6
[NR6]  66Ø:     ;mögl. LABEL o. Referenz-LABEL erwartet
[Ø24]  661:     ;Leerzeichen überlesen
[TR2]  662: TCH4:   CMP   AL,2ØH
[MG1]  663:         JNE   TCH5
[MH1]  664:         CALL  GETB
[UH1]  665:         JMP   TCH4
[D95]  666:     ;gültiges LABEL-Folgezeichen?
[KT2]  667: TCH5:   CMP   AL,ØDH
[8B1]  668:         JE    TCH6
[FP1]  669:         CMP   AL,27H
[8B1]  67Ø:         JE    TCH6
[5M1]  671:         CMP   AL,":"
[8B1]  672:         JE    TCH6
[656]  673:     ;nein - potent. LABEL-Kennung löschen
[614]  674:         MOV   BYTE PTR DS:[LBL1],Ø
[7W6]  675:     ;gültiges Referenz-LABEL-Folgezeichen?
[DK1]  676:         CMP   AL,","
[8B1]  677:         JE    TCH6
[HN1]  678:         CMP   AL,"E"
[8B1]  679:         JE    TCH6
[HN1]  68Ø:         CMP   AL,"e"
[8B1]  681:         JE    TCH6
[Y55]  682:     ;nein - INTEGER-Kennung setzen
[CK4]  683:         MOV   BYTE PTR DS:[NUMZ],1CH
[B42]  684: TCH6:   DEC   CL
[SA5]  685:     ;Zeichenkette nur 1 Byte groß?
[XL1]  686:         OR    CL,CL
[HL1]  687:         JNZ   TCH8
[DU3]  688:     ;ja - ist Zeichen LABEL?
[RZ3]  689:         CMP   BYTE PTR DS:[LBL1],1
[BB1]  69Ø:         JE    TCH7
[ZG4]  691:         CMP   BYTE PTR DS:[NUMZ],ØEH
[XG1]  692:         JNE   TCH8
[G34]  693:     ;ja - auf 2 Byte ergänzen
[352]  694: TCH7:   INC   CL
[PL2]  695:         MOV   DS:[DI],CL
[151]  696:         INC   DI
[687]  697:     ;Zeichenkette m. Leerzeichen abschließen
[HZ4]  698: TCH8:   MOV   BYTE PTR DS:[DI],2ØH
[F43]  699:         MOV   DS:[SVDI],DI
[671]  7ØØ:         POP   DI
[3PØ]  7Ø1:         RET
[ØØØ]  7Ø2:
[5U4]  7Ø3: ;-----------------------------------------
[E76]  7Ø4: ; TTB - TOKEN / LABEL IN TABELLE SUCHEN
[5U4]  7Ø5: ;-----------------------------------------
[HG3]  7Ø6: TTB:    POP   DS:[SRET]
[VA5]  7Ø7:     ;Ende des Suchbereichs festlegen
[Y13]  7Ø8:         MOV   DX,DS:[BUFØ]
[RZ3]  7Ø9:         CMP   BYTE PTR DS:[LBL1],1
[YA1]  71Ø:         JE    TTB1
[ZG4]  711:         CMP   BYTE PTR DS:[NUMZ],ØEH
[YA1]  712:         JE    TTB1
[GB3]  713:         MOV   DX,OFFSET LBLS
[GV5]  714:     ;Länge des Suchbereichs ermitteln
[724]  715: TTB1:   SUB   DX,OFFSET TKØØ
[4E1]  716:         SAR   DX,1
[T73]  717:         MOV   DI,OFFSET TKØØ
```

Listing 5. Alternativ zu Listing 4 kann auch die Assembler-Quelltextversion »sbpre.asm« eingegeben werden

```
[4R2]  718: TTB2:  MOV   CX,DX
[4R5]  719:    ;nach den ersten zwei Bytes suchen
[2Y2]  72Ø: TTB3:  REPNZ SCASW
[R64]  721:    ;nicht gefunden, zum Ende
[GL1]  722:        JNZ   TTB6
[9C5]  723:    ;gefunden, jedoch zu Wortbeginn?
[564]  724:        CMP   BYTE PTR DS:[DI-3],8ØH
[3B3]  725:    ;nein - weitersuchen
[VA1]  726:        JB    TTB3
[S45]  727:    ;ja - evtl. LABEL-Beginn sichern
[RZ3]  728:        CMP   BYTE PTR DS:[LBL1],1
[PG1]  729:        JNE   TTB4
[A43]  73Ø:        MOV   DS:[SLBL],DI
[5P6]  731:    ;restliche Zeichen auf Gleichheit prüfen
[7R2]  732: TTB4:  MOV   DX,CX
[291]  733:        POP   SI
[H71]  734:        POP   CX
[R91]  735:        PUSH  CX
[8B1]  736:        PUSH  SI
[1D1]  737:        ADD   SI,2
[C91]  738:        PUSH  DI
[7X1]  739:        REPZ  CMPSB
[7U1]  74Ø:        MOV   BX,DI
[671]  741:        POP   DI
[VZ4]  742:    ;nicht gefunden, weitersuchen
[4L1]  743:        JNZ   TTB2
[BP6]  744:    ;gefunden, BX auf Wortgrenze ausrichten
[TT1]  745:        MOV   AH,BL
[8D1]  746:        SAR   AH,1
[TLØ]  747:        CMC
[2B1]  748:        ADC   BX,Ø
[671]  749:        POP   DI
[PC5]  75Ø:    ;evtl. Leerzeichen zurückholen
[133]  751:        CMP   DS:[ZBEG],DI
[AB1]  752:        JE    TTB5
[B54]  753:        CMP   BYTE PTR DS:[DI-1],2ØH
[TG1]  754:        JNE   TTB5
[W31]  755:        DEC   DI
[1K2]  756: TTB5:  SUB   AH,AH
[PR3]  757: TTB6:  PUSH  DS:[SRET]
[3PØ]  758:        RET
[ØØØ]  759:
[5U4]  76Ø: ;-----------------------------------------
[UD6]  761: ; TLB - LABEL AN TOKEN-TABELLE ANHÄNGEN
[5U4]  762: ;-----------------------------------------
[YB2]  763: TLB:   PUSH  SI
[F13]  764:        MOV   DI,DS:[BUFØ]
[4B1]  765:        ADD   CL,2
[ØW1]  766:        MOV   BX,CX
[XU1]  767: TLB1:  LODSB
[NXØ]  768:        STOSB
[4L1]  769:        LOOP  TLB1
[FU6]  77Ø:    ;evtl. Leerzeichen f. Wortgrenze anfügen
[WD1]  771:        SAR   BX,1
[GA1]  772:        JB    TLB2
[5R1]  773:        MOV   AL,2ØH
[NXØ]  774:        STOSB
[EX6]  775:    ;Zeilennummmer und Pseudo-TOKEN anfügen
[3Y3]  776: TLB2:  MOV   AX,DS:[LONR]
[ZYØ]  777:        STOSW
[RR1]  778:        MOV   AL,8ØH
[NXØ]  779:        STOSB
[946]  78Ø:    ;Kennzeichnung für vorhandenen LABEL
[H14]  781:        MOV   BYTE PTR DS:[LBL3],1
[RL4]  782:    ;neues Puffer-Ende speichern
[3Ø3]  783:        MOV   DS:[BUFØ],DI
[671]  784:        POP   DI
[3PØ]  785:        RET
[ØØØ]  786:
[5U4]  787: ;-----------------------------------------
[T64]  788: ; TTR - TOKEN ÜBERTRAGEN
[5U4]  789: ;-----------------------------------------
[265]  79Ø: TTR:   CMP   BYTE PTR DS:[TKZ2],1
[4C1]  791:        JE    TTR3
[7G2]  792: TTR1:  PUSH  AX
[L35]  793:    ;vorangehendes Zeichen prüfen
[5Z2]  794:        MOV   AL,DS:[DI-1]
[C91]  795:        PUSH  DI
[ØC3]  796:        MOV   DI,OFFSET TBX2
[7U1]  797:        MOV   CX,ØEH
[712]  798:        REPNZ SCASB
[671]  799:        POP   DI
[1C1]  8ØØ:        JE    TTR2
[ZD4]  8Ø1:    ;Leerstelle voranstellen
[5R1]  8Ø2:        MOV   AL,2ØH
[NXØ]  8Ø3:        STOSB
[LA2]  8Ø4: TTR2:  POP   AX
[H34]  8Ø5:        MOV   BYTE PTR DS:[TKZ2],1
[BS2]  8Ø6:    ;TOKEN speichern
[9Z1]  8Ø7: TTR3:  STOSB
[3PØ]  8Ø8:        RET
[ØØØ]  8Ø9:
[5U4]  81Ø: ;-----------------------------------------
[SZ3]  811: ; DAT - DATA ÜBERTRAGEN
[5U4]  812: ;-----------------------------------------
[ØP1]  813:    ;DATA-Zahl
[VP2]  814: DAT:   CMP   AL,22H
[K91]  815:        JE    DAT2
[XR1]  816:        CMP   AL,ØDH
[N91]  817:        JE    DAT3
[FP1]  818:        CMP   AL,27H
[N91]  819:        JE    DAT3
[UR1]  82Ø:        CMP   AL,3AH
[N91]  821:        JE    DAT3
[PW1]  822: DAT1:  STOSB
[MH1]  823:        CALL  GETB
[MB1]  824:        JMP   DAT
[4R1]  825:    ;DATA-Text
[RW1]  826: DAT2:  STOSB
[MH1]  827:        CALL  GETB
[VN1]  828:        CMP   AL,22H
[F91]  829:        JE    DAT1
[XR1]  83Ø:        CMP   AL,ØDH
[1F1]  831:        JNE   DAT2
[ES6]  832:    ;zur weiteren Zeilen-Code-Verarbeitung
[SC2]  833: DAT3:  JMP   ZCD2
[ØØØ]  834:
[5U4]  835: ;-----------------------------------------
[Y44]  836: ; TXT - TEXT ÜBERTRAGEN
[5U4]  837: ;-----------------------------------------
[TK5]  838:    ;evtl. Leerzeichen vor '"' einfügen
[LM2]  839: TXT:   CALL  TTR1
[1L1]  84Ø:        JMP   TXT2
[VN3]  841:    ;Literal übertragen
[VZ1]  842: TXT1:  STOSB
[ØM2]  843: TXT2:  CALL  GETB
[VN1]  844:        CMP   AL,22H
[VH1]  845:        JNE   TXT3
[ES6]  846:    ;zur weiteren Zeilen-Code-Verarbeitung
[UL1]  847:        JMP   ZCDD
[ZW2]  848: TXT3:  CMP   AL,ØDH
[NH1]  849:        JNE   TXT1
[L25]  85Ø:    ;zur Zeilenende-Verarbeitung
[SE1]  851:        JMP   ZEX
[ØØØ]  852:
[5U4]  853: ;-----------------------------------------
[5Y5]  854: ; REM - KOMMENTARZEILEN ÜBERTRAGEN
[5U4]  855: ;-----------------------------------------
[WZ2]  856: REM:   MOV   AX,8F3AH
[ZYØ]  857:        STOSW
[GX1]  858:        MOV   AL,ØD9H
[2X1]  859: REM1:  STOSB
[MH1]  86Ø:        CALL  GETB
[5U2]  861: REM2:  CMP   AL,ØDH
[CG1]  862:        JNE   REM1
[FB4]  863:    ;REM-Kennzeichnung setzen
[W64]  864:        MOV   BYTE PTR DS:[REMZ],1
[L25]  865:    ;zur Zeilenende-Verarbeitung
[SE1]  866:        JMP   ZEX
[ØØØ]  867:
[5U4]  868: ;-----------------------------------------
[6U6]  869: ; HEX - HEXADEZIMALE ZAHLEN TOKENISIEREN
[5U4]  87Ø: ;-----------------------------------------
[LG2]  871: HEX:   CALL  GETB
[TU1]  872:        SUB   DX,DX
[YH2]  873: HEX1:  CALL  GETB
[YT1]  874:        MOV   CH,AL
[PN1]  875:        SUB   AL,3ØH
[WY1]  876:    ;Zeichen Ø-9?
[ØE1]  877:        CMP   AL,9
[TE1]  878:        JBE   HEX2
[UD1]  879:        SUB   AL,7
[DK2]  88Ø:    ;Zeichen A/a-F/f?
[HR1]  881:        CMP   AL,ØAH
[8A1]  882:        JB    HEX3
[5S1]  883:        CMP   AL,ØFH
[TE1]  884:        JBE   HEX2
[LN1]  885:        SUB   AL,2ØH
[HR1]  886:        CMP   AL,ØAH
[8A1]  887:        JB    HEX3
[5S1]  888:        CMP   AL,ØFH
[5A1]  889:        JA    HEX3
[354]  89Ø:    ;HEX-Zeichen kumulieren
[ZC2]  891: HEX2:  MOV   CL,4
[7T1]  892:        SHL   DX,CL
[5P1]  893:        ADD   DL,AL
[3H1]  894:        JMP   HEX1
[WX4]  895:    ;HEX-Ende, HEX-TOKEN speichern
[VV2]  896: HEX3:  MOV   AL,ØCH
[NXØ]  897:        STOSB
[ZV1]  898:        MOV   AX,DX
[ZYØ]  899:        STOSW
[GT1]  9ØØ:        MOV   AL,CH
[A34]  9Ø1:        MOV   BYTE PTR DS:[TKZ1],Ø
[ES6]  9Ø2:    ;zur weiteren Zeilen-Code-Verarbeitung
[4K1]  9Ø3:        JMP   ZCD2
[ØØØ]  9Ø4:
[5U4]  9Ø5: ;-----------------------------------------
[ZX5]  9Ø6: ; NUM - INTEGER-ZAHLEN TOKENISIEREN
[5U4]  9Ø7: ;-----------------------------------------
[GF6]  9Ø8:    ;Status sichern, falls später SINGLE
[Ø65]  9Ø9: NUM:   MOV   BYTE PTR DS:[TKZ1],Ø
[953]  91Ø:        MOV   CH,DS:[NUMZ]
[B63]  911:        MOV   DS:[SVSI],SI
[F43]  912:        MOV   DS:[SVDI],DI
[BU1]  913:        SUB   BX,BX
[SH1]  914:        JMP   NUM2
[EK2]  915: NUM1:  CALL  GETB
[Ø24]  916:    ;Leerzeichen überlesen
[LN1]  917:        CMP   AL,2ØH
```

```
[XA1]  918:          JE    NUM1
[PT2]  919:      ;Zeichen sichern
[DN2]  92Ø: NUM2:  MOV   CL,AL
[PN1]  921:        SUB   AL,3ØH
[WY1]  922:      ;Zeichen Ø-9?
[ØE1]  923:        CMP   AL,9
[RA1]  924:        JA    NUM3
[GØ4]  925:      ;ja - Zeichen kumulieren
[MR1]  926:        SUB   AH,AH
[UU1]  927:        XCHG  AX,BX
[VT1]  928:        MOV   DX,ØAH
[191]  929:        IMUL  DX
[1A3]  93Ø:      ;INTEGER-Überlauf?
[ØP1]  931:        OR    DX,DX
[KL1]  932:        JNZ   NUM7
[SR1]  933:        ADD   AX,BX
[6B1]  934:        JB    NUM7
[W14]  935:      ;kein INTEGER-Überlauf
[UV1]  936:        MOV   BX,AX
[DZ3]  937:      ;zum nächsten Zeichen
[NH1]  938:        JMP   NUM1
[4D6]  939:      ;INTEGER-Ende, Zeilennummer erwartet?
[7V2]  94Ø: NUM3:  CMP   CH,ØEH
[6B1]  941:        JE    NUM4
[1D2]  942:      ;nein - Zahl > 9?
[FZ1]  943:        CMP   BX,ØØØ9H
[UA1]  944:        JA    NUM4
[C46]  945:      ;nein - 1 Byte-INTEGER-TOKEN speichern
[TV1]  946:        MOV   AX,BX
[5L1]  947:        ADD   AL,11H
[NXØ]  948:        STOSB
[1K1]  949:        JMP   NUM5
[H44]  95Ø:      ;INTEGER-TOKEN speichern
[1N2]  951: NUM4:  MOV   AL,CH
[NXØ]  952:        STOSB
[TV1]  953:        MOV   AX,BX
[ZYØ]  954:        STOSW
[D45]  955:      ;letztes Zeichen zurückholen
[EN2]  956: NUM5:  MOV   AL,CL
[ØG2]  957:      ;Dezimalpunkt?
[NK1]  958:        CMP   AL,"."
[FB1]  959:        JE    NUM7
[EV2]  96Ø:      ;SINGLE-Kennung?
[1H1]  961:        CMP   AL,"!"
[FB1]  962:        JE    NUM7
[PU2]  963:      ;DOUBLE-Kennung?
[9H1]  964:        CMP   AL,"#"
[BØ4]  965:      ;ja - zur Fehler-Routine
[RV1]  966:        MOV   BP,8ØØH
[KB1]  967:        JE    NUM8
[2V1]  968:        AND   CL,ØDFH
[A92]  969:      ;EXP-Kennung?
[TN1]  97Ø:        CMP   CL,"E"
[VG1]  971:        JNE   NUM6
[HP2]  972:        MOV   CL,DS:[SI]
[2V1]  973:        AND   CL,ØDFH
[UX1]  974:      ;evtl ELSE?
[NP1]  975:        CMP   CL,"L"
[YG1]  976:        JNE   NUM7
[KX6]  977:      ;INTEGER ok, zur weiteren Zeilen-Verarb.
[FE2]  978: NUM6:  JMP   ZCD2
[WG6]  979:      ;Status: INTEGER-Überlauf oder SINGLE
[H64]  98Ø:      ;Zeilennummer erwartet?
[BV2]  981: NUM7:  CMP   CH,ØEH
[PD4]  982:      ;nein - zur SINGLE-Routine
[UB1]  983:        JNE   SNG
[5U4]  984:      ;ja - SINGLE ohne Überlauf?
[EV1]  985:        MOV   BP,5ØØH
[1H1]  986:        CMP   AL,"!"
[L36]  987:      ;ja - zur weiteren Zeilen-Code-Verarb.
[CB1]  988:        JE    NUM6
[TD5]  989:      ;nein - zur Fehler-Exit-Routine
[PF2]  99Ø: NUM8:  JMP   EXT2
[ØØØ]  991:
[5U4]  992: ;-----------------------------------------
[KK5]  993: ; SNG - SINGLE-ZAHLEN TOKENISIEREN
[5U4]  994: ;-----------------------------------------
[HZ5]  995:      ;Status vor INTEGER restaurieren
[RY3]  996: SNG:   MOV   SI,DS:[SVSI]
[S51]  997:        DEC   SI
[6G6]  998:      ;Register für späteren Aufruf der VAL-
[E76]  999:      ;Funktion aus SBASIC.BIN vorbereiten
[RM5] 1ØØØ:      ;und Zeichen auf Stack übertragen
[KB1] 1ØØ1:        PUSH  SS
[M71] 1ØØ2:        POP   ES
[1S1] 1ØØ3:        SUB   SP,256
[9W1] 1ØØ4:        MOV   DI,SP
[7U1] 1ØØ5:        MOV   BX,DI
[8S1] 1ØØ6:        SUB   AL,AL
[NXØ] 1ØØ7:        STOSB
[CVØ] 1ØØ8:        LODSB
[7X1] 1ØØ9: SNG1:  STOSB
[WG3] 1Ø1Ø:        INC   BYTE PTR ES:[BX]
[CH2] 1Ø11: SNG2:  CALL  GETB
[Ø24] 1Ø12:      ;Leerzeichen überlesen
[LN1] 1Ø13:        CMP   AL,2ØH
[1B1] 1Ø14:        JE    SNG2
[V62] 1Ø15:      ;SINGLE-Ende?
[1H1] 1Ø16:        CMP   AL,"!"
[1B1] 1Ø17:        JE    SNG2
```

```
[4G3] 1Ø18:      ;nein - Dezimalpunkt?
[NK1] 1Ø19:        CMP   AL,"."
[YA1] 1Ø2Ø:        JE    SNG1
[1H2] 1Ø21:      ;EXP-Funktion?
[954] 1Ø22:        CMP   BYTE PTR DS:[DI-1],"E"
[LG1] 1Ø23:        JNE   SNG3
[6T2] 1Ø24:      ;ja - Vorzeichen?
[9K1] 1Ø25:        CMP   AL,"+"
[YA1] 1Ø26:        JE    SNG1
[HK1] 1Ø27:        CMP   AL,"-"
[YA1] 1Ø28:        JE    SNG1
[S13] 1Ø29:      ;nein - Zeichen Ø-9?
[NS2] 1Ø3Ø: SNG3:  CMP   AL,3ØH
[YA1] 1Ø31:        JB    SNG4
[UR1] 1Ø32:        CMP   AL,3AH
[NA1] 1Ø33:        JB    SNG1
[5H3] 1Ø34:      ;nein - EXP-Funktion?
[UU1] 1Ø35:        AND   AL,ØDFH
[HN1] 1Ø36:        CMP   AL,"E"
[PG1] 1Ø37:        JNE   SNG4
[3Ø4] 1Ø38:        CMP   BYTE PTR DS:[SI],"L"
[7B1] 1Ø39:        JE    SNG4
[3Ø4] 1Ø4Ø:        CMP   BYTE PTR DS:[SI],"l"
[DG1] 1Ø41:        JNE   SNG1
[WS6] 1Ø42:      ;nein - Status: Zeichenübertrag. beendet
[2W6] 1Ø43:      ;Vorbereitung u.VAL-Aufruf in SBASIC.BIN
[BZ3] 1Ø44: SNG4:  MOV   DS:[SVSI],SI
[4V1] 1Ø45:        MOV   BP,CS
[7M1] 1Ø46:        SUB   BP,25
[4V1] 1Ø47:        MOV   ES,BP
[AN2] 1Ø48:        MOV   DX,ES:[2ØØ]
[WH2] 1Ø49:        SUB   DX,ES:[Ø]
[D61] 1Ø5Ø:        CALL  DX
[CY3] 1Ø51:      ;Register restaurieren
[P91] 1Ø52:        PUSH  DS
[M71] 1Ø53:        POP   ES
[P43] 1Ø54:        MOV   DI,DS:[SVDI]
[MT3] 1Ø55:      ;SINGLE-TOKEN ausgeben
[KT1] 1Ø56:        MOV   AL,1DH
[NXØ] 1Ø57:        STOSB
[971] 1Ø58:        POP   AX
[ZYØ] 1Ø59:        STOSW
[971] 1Ø6Ø:        POP   AX
[ZYØ] 1Ø61:        STOSW
[LG6] 1Ø62:      ;SI und letztes Zeichen restaurieren
[173] 1Ø63:        MOV   SI,DS:[SVSI]
[KØ3] 1Ø64:        MOV   AL,DS:[SI-1]
[9U2] 1Ø65:      ;DOUBLE Kennung?
[9H1] 1Ø66:        CMP   AL,"#"
[TG1] 1Ø67:        JNE   SNG5
[4V4] 1Ø68:      ;ja - zur Fehler-Exit-Routine
[RV1] 1Ø69:        MOV   BP,8ØØH
[5H1] 1Ø7Ø:        JMP   EXT2
[ES6] 1Ø71:      ;zur weiteren Zeilen-Code-Verarbeitung
[BD2] 1Ø72: SNG5:  JMP   ZCD2
[ØØØ] 1Ø73:
[5U4] 1Ø74: ;-----------------------------------------
[8C5] 1Ø75: ; GETB - ZEICHEN AUS PUFFER LESEN
[5U4] 1Ø76: ;-----------------------------------------
[7W1] 1Ø77: GETB:  LODSB
[9B3] 1Ø78:      ;TABULATOR-Zeichen?
[ØE1] 1Ø79:        CMP   AL,9
[9R1] 1Ø8Ø:        JNE   GETB2
[RC5] 1Ø81:      ;ja - Umwandlung in Leerzeichen
[ZV2] 1Ø82: GETB1: MOV   AL,2ØH
[FØ3] 1Ø83:        MOV   DS:[SI-1],AL
[TK1] 1Ø84: GETB2: RET
[ØØØ] 1Ø85:
[5U4] 1Ø86: ;-----------------------------------------
[5D3] 1Ø87: ; EXT - EXIT-ROUTINEN
[5U4] 1Ø88: ;-----------------------------------------
[RT2] 1Ø89: EXT:   MOV   BP,2ØØH
[NY2] 1Ø9Ø:      ;Quelldatei leer?
[G43] 1Ø91:        CMP   DS:[LONR],AX
[4X4] 1Ø92:      ;ja zur Fehler-Exit-Routine
[DA1] 1Ø93:        JE    EXT2
[5L5] 1Ø94:      ;zweiter Durchgang erforderlich?
[P44] 1Ø95:        CMP   BYTE PTR DS:[PASS],1
[AA1] 1Ø96:        JE    EXT1
[ZZ3] 1Ø97:        CMP   BYTE PTR DS:[LBL3],1
[TF1] 1Ø98:        JNE   EXT1
[WE3] 1Ø99:      ;ja - zum 2. Durchlauf
[864] 11ØØ:        MOV   BYTE PTR DS:[PASS],1
[GH1] 11Ø1:        JMP   PRE3
[BH6] 11Ø2:      ;nein - Dateiende-Kennzeichnung setzen
[SW2] 11Ø3: EXT1:  MOV   AL,1AH
[NXØ] 11Ø4:        STOSB
[DY1] 11Ø5:        MOV   BP,ØFFH
[4G6] 11Ø6:      ;Speichern GW-BASIC-Kennung/Fehlercode
[8Y3] 11Ø7: EXT2:  MOV   SI,DS:[BUF1]
[KN2] 11Ø8:        MOV   DS:[SI],BP
[MR5] 11Ø9:      ;Code-Verschiebung zum Segmentende
[2A5] 111Ø: EXT3:  MOV   BYTE PTR DS:[PASS],2
[BU1] 1111:        MOV   CX,DI
[6U1] 1112:        SUB   CX,SI
[WV1] 1113:        MOV   SI,DI
[S51] 1114:        DEC   SI
```

Listing 5. Alternativ zu Listing 4 kann auch die Assembler-Quelltextversion »sbpre.asm« eingegeben werden

```
[WD2] 1115:         MOV   DI,ØFFFFH
[KNØ] 1116:         STD
[522] 1117:         REPNZ MOVSB
[ULØ] 1118:         CLD
[151] 1119:         INC   DI
[FV6] 112Ø:     ;Übergabe Codeadresse/Fehler-Zeilen-Nr.
[9U2] 1121:         PUSH  DS:[PHNR]
[ØU2] 1122:         PUSH  DS:[LONR]
[CV1] 1123:         MOV   BX,CS
[YL1] 1124:         SUB   BX,3Ø
[ØW1] 1125:         MOV   DS,BX
[CF2] 1126:         MOV   DS:[Ø],DI
[UW1] 1127:         POP   DS:[2]
[2X1] 1128:         POP   DS:[4]
[U91] 1129:         PUSH  ES
[G71] 113Ø:         POP   DS
[LU2] 1131:     ;Löschen "###.GWB"
[EC3] 1132:         MOV   DX,OFFSET FGWB
[BR1] 1133:         MOV   AH,41H
[N81] 1134:         INT   21H
[E64] 1135:     ;"/GWB" Parameter gesetzt?
[K15] 1136:         CMP   WORD PTR DS:[SAVØ],ØFFFFH
[5G1] 1137:         JNE   EXT5
[2Z4] 1138:     ;ja - Programmcode fehlerfrei?
[2C4] 1139:         CMP   BYTE PTR DS:[DI],ØFFH
[5G1] 114Ø:         JNE   EXT5
[2U6] 1141:     ;ja - Quellcode bereits GWBASIC-Format?
[RW3] 1142:         CMP   BYTE PTR DS:[DI+1],Ø
[4Z3] 1143:         MOV   BYTE PTR DS:[DI+1],9
[5G1] 1144:         JNE   EXT5
[MV6] 1145:     ;nein - Code für GWBASIC-Ausgabe setzen
[N84] 1146:         MOV   BYTE PTR DS:[DI+1],ØAH
[LP3] 1147:     ;Datei "###.GWB" öffnen
[GT1] 1148:         MOV   AH,3CH
[KU1] 1149:         SUB   CX,CX
[N81] 115Ø:         INT   21H
[UV1] 1151:         MOV   BX,AX
[GP3] 1152:     ;Code-Länge ermitteln
[FU1] 1153:         MOV   DX,DI
[Ø51] 1154:         NEG   DI
[BU1] 1155:         MOV   CX,DI
[VK5] 1156:     ;Datei beschreiben und schließen
[YU2] 1157: EXT4:   MOV   AH,4ØH
[N81] 1158:         INT   21H
[ST1] 1159:         MOV   AH,3EH
[N81] 116Ø:         INT   21H
[1A2] 1161: EXT5:   POP   DS
[9SØ] 1162:         RETF
[ØØØ] 1163:
[5U4] 1164: ;-----------------------------------------
[Ø16] 1165: ; LNK - LINK UND AUSGABE DER EXE-DATEI
[5U4] 1166: ;-----------------------------------------
[LC2] 1167: LNK:    PUSH  DS
[4V1] 1168:         MOV   BP,CS
[4V1] 1169:         MOV   ES,BP
[PL1] 117Ø:         SUB   BP,3Ø
[ØV1] 1171:         MOV   DS,BP
[4V6] 1172:     ;Adresse des Programmcode-Endes sichern
[SY1] 1173:         PUSH  DS:[Ø]
[XU2] 1174:     ;EXE-Datei öffnen
[AW1] 1175:         XOR   CX,CX
[NF1] 1176:         MOV   DX,2
[GT1] 1177:         MOV   AH,3CH
[N81] 1178:         INT   21H
[UV1] 1179:         MOV   BX,AX
[C56] 118Ø:     ;markierte SBASIC.BIN-Routinen linken
[MR1] 1181:         MOV   SI,8Ø
[6U1] 1182:         XOR   DI,DI
[HP2] 1183:         MOV   BP,DS:[SI]
[L22] 1184:         AND   BP,7FFFH
[9P1] 1185:         SUB   BP,48Ø
[CW1] 1186: LNK1:   LODSW
[LX5] 1187:     ;Ende der SBASIC-BIN-Adresstabelle?
[FW1] 1188:         CMP   AH,ØFFH
[DA1] 1189:         JE    LNK2
[1X4] 119Ø:     ;nein - ist Adresse markiert?
[1P1] 1191:         CMP   AH,8ØH
[1A1] 1192:         JB    LNK1
[EF6] 1193:     ;ja - Startadresse und Länge ermitteln
[R22] 1194:         AND   AX,7FFFH
[XP2] 1195:         MOV   CX,DS:[SI]
[Z22] 1196:         AND   CX,7FFFH
[CU1] 1197:         SUB   CX,AX
[XF4] 1198:     ;Unterprogramm übertragen
[7T1] 1199:         SUB   AX,BP
[8B1] 12ØØ:         PUSH  SI
[FX1] 12Ø1:         MOV   SI,AX
[522] 12Ø2:         REPNZ MOVSB
[291] 12Ø3:         POP   SI
[2H1] 12Ø4:         JMP   LNK1
[156] 12Ø5:     ;EXE-Header und Bibliothek schreiben
[XT2] 12Ø6: LNK2:   MOV   DX,448
[GH1] 12Ø7:         ADD   DI,32
[BU1] 12Ø8:         MOV   CX,DI
[7R1] 12Ø9:         MOV   AH,4ØH
[N81] 121Ø:         INT   21H
[W44] 1211:     ;Programmcode schreiben
[4V1] 1212:         MOV   BP,CS
```

```
[9L1] 1213:         ADD   BP,9ØØ
[ØV1] 1214:         MOV   DS,BP
[8U1] 1215:         MOV   DX,3ØØØ
[H71] 1216:         POP   CX
[NU1] 1217:         SUB   CX,DX
[BH1] 1218:         JMP   EXT4
[ØØØ] 1219:
[5U4] 122Ø: ;-----------------------------------------
[DM6] 1221: ; BASIC-TOKEN-TABELLE / OPT. LABEL-PUFFER
[5U4] 1222: ;-----------------------------------------
[ERØ] 1223: EVEN
[DF1] 1224:         DW    8ØØØH
[734] 1225:     ;Tabelle der 1-Byte-TOKEN
[ZR2] 1226: TKØØ    DB    "END ",81H
[UZ1] 1227:         DB    "FOR ",82H
[H62] 1228:         DB    "NEXT ",83H
[832] 1229:         DB    "DATA ",84H
[PG2] 123Ø:         DB    "INPUT ",85H
[1Z1] 1231:         DB    "DIM ",86H
[M22] 1232:         DB    "READ ",87H
[SZ1] 1233:         DB    "LET ",88H
[462] 1234:         DB    "GOTO ",89H
[NØ2] 1235:         DB    "RUN ",8AH
[NU1] 1236:         DB    "IF ",8BH
[SS2] 1237:         DB    "RESTORE ",8CH
[UH2] 1238:         DB    "GOSUB ",8DH
[3N2] 1239:         DB    "RETURN ",8EH
[RZ1] 124Ø:         DB    "REM ",8FH
[A62] 1241:         DB    "STOP ",9ØH
[AF2] 1242:         DB    "PRINT ",91H
[KC2] 1243:         DB    "CLEAR ",92H
[C62] 1244:         DB    "LIST ",93H
[3Ø2] 1245:         DB    "NEW ",94H
[6U1] 1246:         DB    "ON ",95H
[C52] 1247:         DB    "WAIT ",96H
[9Y1] 1248:         DB    "DEF ",97H
[D42] 1249:         DB    "POKE ",98H
[662] 125Ø:         DB    "CONT ",99H
[A12] 1251:         DB    "OUT ",9CH
[4M2] 1252:         DB    "LPRINT ",9DH
[AH2] 1253:         DB    "LLIST ",9EH
[2F2] 1254:         DB    "WIDTH ",ØAØH
[392] 1255:         DB    "ELSE ",ØA1H
[SA2] 1256:         DB    "TRON ",ØA2H
[4F2] 1257:         DB    "TROFF ",ØA3H
[3A2] 1258:         DB    "SWAP ",ØA4H
[6F2] 1259:         DB    "ERASE ",ØA5H
[V92] 126Ø:         DB    "EDIT ",ØA6H
[YG2] 1261:         DB    "ERROR ",ØA7H
[NL2] 1262:         DB    "RESUME ",ØA8H
[XG2] 1263:         DB    "DELETE ",ØA9H
[WB2] 1264:         DB    "AUTO ",ØAAH
[4K2] 1265:         DB    "RENUM ",ØABH
[5M2] 1266:         DB    "DEFSTR ",ØACH
[ZK2] 1267:         DB    "DEFINT ",ØADH
[HL2] 1268:         DB    "DEFSNG ",ØAEH
[RH2] 1269:         DB    "DEFDBL ",ØAFH
[M82] 127Ø:         DB    "LINE ",ØBØH
[LE2] 1271:         DB    "WHILE ",ØB1H
[R82] 1272:         DB    "WEND ",ØB2H
[YH2] 1273:         DB    "INSTR ",ØD8H
[R82] 1274:         DB    "CALL ",ØB3H
[ZG2] 1275:         DB    "WRITE ",ØB7H
[KL2] 1276:         DB    "OPTION ",ØB8H
[V73] 1277:         DB    "RANDOMIZE ",ØB9H
[GA2] 1278:         DB    "OPEN ",ØBAH
[TH2] 1279:         DB    "CLOSE ",ØBBH
[V82] 128Ø:         DB    "LOAD ",ØBCH
[PG2] 1281:         DB    "MERGE ",ØBDH
[DA2] 1282:         DB    "SAVE ",ØBEH
[3K2] 1283:         DB    "COLOR ",ØBFH
[D42] 1284:         DB    "CLS ",ØCØH
[MG2] 1285:         DB    "MOTOR ",ØC1H
[BF2] 1286:         DB    "BSAVE ",ØC2H
[PD2] 1287:         DB    "BLOAD ",ØC3H
[RG2] 1288:         DB    "SOUND ",ØC4H
[Y82] 1289:         DB    "BEEP ",ØC5H
[UA2] 129Ø:         DB    "PSET ",ØC6H
[LL2] 1291:         DB    "PRESET ",ØC7H
[ZH2] 1292:         DB    "SCREEN ",ØC8H
[B52] 1293:         DB    "KEY ",ØC9H
[MH2] 1294:         DB    "LOCATE ",ØCAH
[TW1] 1295:         DB    "TO ",ØCCH
[CA2] 1296:         DB    "THEN ",ØCDH
[Z32] 1297:         DB    "TAB ",ØCEH
[FB2] 1298:         DB    "STEP ",ØCFH
[B52] 1299:         DB    "USR ",ØDØH
[U32] 13ØØ:         DB    "SPC ",ØD2H
[852] 13Ø1:         DB    "NOT ",ØD3H
[F42] 13Ø2:         DB    "ERL ",ØD4H
[352] 13Ø3:         DB    "ERR ",ØD5H
[3T2] 13Ø4:         DB    "STRING$ ",ØD6H
[FG2] 13Ø5:         DB    "USING ",ØD7H
[8N2] 13Ø6:         DB    "VARPTR ",ØDAH
[UM2] 13Ø7:         DB    "CSRLIN ",ØDBH
[ZH2] 13Ø8:         DB    "POINT ",ØDCH
[F42] 13Ø9:         DB    "OFF ",ØDDH
[DH2] 131Ø:         DB    "INKEY$ ",ØDEH
```

```
[E42] 1311:         DB    "AND  ",ØEEH
[CX1] 1312:         DB    "OR ",ØEFH
[852] 1313:         DB    "XOR  ",ØFØH
[752] 1314:         DB    "EQV  ",ØF1H
[L42] 1315:         DB    "IMP  ",ØF2H
[T32] 1316:         DB    "MOD  ",ØF3H
[C76] 1317:     ;Tabelle der TOKEN mit "FD"-Vorkennung
[NV2] 1318: TKFD    DB    "CVI  ",81H
[8Ø2] 1319:         DB    "CVS  ",82H
[WY1] 132Ø:         DB    "CVD  ",83H
[GZ1] 1321:         DB    "MKI$ ",84H
[FØ2] 1322:         DB    "MKS$ ",85H
[3Z1] 1323:         DB    "MKD$ ",86H
[VL2] 1324:         DB    "EXTERR ",8BH
[F76] 1325:     ;Tabelle der TOKEN mit "FE"-Vorkennung
[GB3] 1326: TKFE    DB    "FILES  ",81H
[BD2] 1327:         DB    "FIELD  ",82H
[ØM2] 1328:         DB    "SYSTEM ",83H
[D32] 1329:         DB    "NAME ",84H
[M52] 133Ø:         DB    "LSET ",85H
[V52] 1331:         DB    "RSET ",86H
[P42] 1332:         DB    "KILL ",87H
[WØ2] 1333:         DB    "PUT  ",88H
[MZ1] 1334:         DB    "GET  ",89H
[UF2] 1335:         DB    "RESET  ",8AH
[7L2] 1336:         DB    "COMMON ",8BH
[BE2] 1337:         DB    "CHAIN  ",8CH
[ØE2] 1338:         DB    "DATE$  ",8DH
[EE2] 1339:         DB    "TIME$  ",8EH
[5G2] 134Ø:         DB    "PAINT  ",8FH
[AZ1] 1341:         DB    "COM  ",9ØH
[WF2] 1342:         DB    "CIRCLE ",91H
[B52] 1343:         DB    "DRAW ",92H
[M52] 1344:         DB    "PLAY ",93H
[BE2] 1345:         DB    "TIMER  ",94H
[UD2] 1346:         DB    "ERDEV  ",95H
[AF2] 1347:         DB    "IOCTL  ",96H
[PD2] 1348:         DB    "CHDIR  ",97H
[9E2] 1349:         DB    "MKDIR  ",98H
[NE2] 135Ø:         DB    "RMDIR  ",99H
[DF2] 1351:         DB    "SHELL  ",9AH
[NS2] 1352:         DB    "ENVIRON  ",9BH
[862] 1353:         DB    "VIEW ",9CH
[8L2] 1354:         DB    "WINDOW ",9DH
[452] 1355:         DB    "PMAP ",9EH
[DP2] 1356:         DB    "PALETTE  ",9FH
[DF2] 1357:         DB    "LCOPY  ",ØAØH
[5E2] 1358:         DB    "CALLS  ",ØA1H
[5G2] 1359:         DB    "PCOPY  ",ØA5H
[492] 136Ø:         DB    "LOCK ",ØA7H
[ZK2] 1361:         DB    "UNLOCK ",ØA8H
[K76] 1362:     ;Tabelle der TOKEN mit "FF"-Vorkennung
[A73] 1363: TKFF    DB    "LEFT$  ",81H
[5E2] 1364:         DB    "RIGHT$ ",82H
[WY1] 1365:         DB    "MID$ ",83H
[DZ1] 1366:         DB    "SGN  ",84H
[4Ø2] 1367:         DB    "INT  ",85H
[2Z1] 1368:         DB    "ABS  ",86H
[GØ2] 1369:         DB    "SQR  ",87H
[ØZ1] 137Ø:         DB    "RND  ",88H
[PZ1] 1371:         DB    "SIN  ",89H
[EZ1] 1372:         DB    "LOG  ",8AH
[NØ2] 1373:         DB    "EXP  ",8BH
[BØ2] 1374:         DB    "COS  ",8CH
[KZ1] 1375:         DB    "TAN  ",8DH
[6Ø2] 1376:         DB    "ATN  ",8EH
[DZ1] 1377:         DB    "FRE  ",8FH
[RZ1] 1378:         DB    "INP  ",9ØH
[AØ2] 1379:         DB    "POS  ",91H
[4Z1] 138Ø:         DB    "LEN  ",92H
[612] 1381:         DB    "STR$ ",93H
[2Z1] 1382:         DB    "VAL  ",94H
[RY1] 1383:         DB    "ASC  ",95H
[1Ø2] 1384:         DB    "CHR$ ",96H
[Y32] 1385:         DB    "PEEK ",97H
[LD2] 1386:         DB    "SPACE$ ",98H
[WØ2] 1387:         DB    "OKT$ ",99H
[XØ2] 1388:         DB    "HEX$ ",9AH
[T62] 1389:         DB    "LPOS ",9BH
[462] 139Ø:         DB    "CINT ",9CH
[552] 1391:         DB    "CSNG ",9DH
[S32] 1392:         DB    "CDBL ",9EH
[SØ2] 1393:         DB    "FIX  ",9FH
[V32] 1394:         DB    "PEN  ",ØAØH
[6E2] 1395:         DB    "STICK  ",ØA1H
[TF2] 1396:         DB    "STRIG  ",ØA2H
[K32] 1397:         DB    "EOF  ",ØA3H
[K32] 1398:         DB    "LOC  ",ØA4H
[X32] 1399:         DB    "LOF  ",ØA5H
[EH6] 14ØØ:     ;Beginn der reservierten LABEL-Tabelle
[TM1] 14Ø1: LBLS    DB    Ø
[ØØØ] 14Ø2:
[1R1] 14Ø3: CODE    ENDS
[4MØ] 14Ø4:         END
[5U4] 14Ø5: ;-----------------------------------------
```

Listing 5. Alternativ zu Listing 4 kann auch die Assembler-Quelltextversion »sbpre.asm« eingegeben werden

```
[T84]    1: REM Programm: SBSHELL.BAS
[GW6]    2: REM Benutzeroberfläche für SBASIC V.2.Ø
[L14]    3: REM Autor: Peter Knappe
[7B5]    4: REM (C) 1991 DMV Widuch GmbH & Co KG
[NF4]    5: REM          DOS International
[ØØØ]    6:
[XØ5]    7: REM Parameter-Vorbelegungen:
[UM2]    8: EDITOR$="EDITOR"
[ØH2]    9: SBASIC$="SBASIC"
[8X2]   1Ø: GWBASIC$="GWBASIC"
[FØ2]   11: SUFFIX$="BAS"
[ØØØ]   12:
[TL1]   13: DEFINT A-Z
[SH2]   14: DIM DIR$(8Ø),A(53)
[NGØ]   15: M=1
[4R3]   16: DEF FNL$(X)=LEFT$(DAT$,X)
[YA5]   17: DEF FNM$(X)=MID$(DAT$,X,LEN(DAT$))
[SU1]   18: FOR I=1 TO 5
[HL2]   19:  DAT$=COMMAND$(I)
[GD6]   2Ø:  IF FNL$(1)<>"/" AND DAT$<>"" THEN DATEI$=D
        AT$
[5X5]   21:  IF FNL$(7)="/DATEI=" THEN DATEI$=FNM$(8)
[C76]   22:  IF FNL$(8)="/EDITOR=" THEN EDITOR$=FNM$(9)
[Ø46]   23:  IF FNL$(8)="/SBASIC=" THEN SBASIC$=FNM$(9)
[P96]   24:  IF FNL$(8)="/SUFFIX=" THEN SUFFIX$=FNM$(9)
[7R6]   25:  IF FNL$(9)="/GWBASIC=" THEN GWBASIC$=FNM$(
        1Ø)
[U11]   26: NEXT I
[KU1]   27: DAT$=DATEI$
[Ø12]   28: GOSUB DATEI4
[BT2]   29: IF DAT$<>"" THEN M=2
[ØØØ]   3Ø:
[PRØ]   31: MENU
[232]   32:  GOSUB RAHMEN
[Ø22]   33:  GOSUB TITEL
[XF1]   34:  GOSUB DIR
[SUØ]   35: MENU1
[4F5]   36:  IF M=Ø THEN M=7 ELSE IF M=8 THEN M=1
[851]   37:  S=Ø:M1=Ø
[9P1]   38:  LOCATE 2,2,Ø
[ZX1]   39:  FOR I=1 TO 44
[U59]   4Ø:   A$=MID$(" Datei Edit Compile Start GW-Bas
        ic DOS Exit ",I,1)
[AKB]   41:   IF A$=" " THEN S=S+1:COLOR Ø,11:ELSE IF S
        =M THEN COLOR 7,12:IF M1=Ø THEN M1=I+1
[ED1]   42:   PRINT A$;
[U11]   43:  NEXT I
[7F1]   44:  COLOR 4,11
[5U6]   45:  PRINT CHR$(27);CHR$(26);CHR$(17);CHR$(217)
        ;"Esc"
[RE1]   46:  COLOR Ø,11
[2P1]   47:  LOCATE 2,54
[VW1]   48:  A=LEN(PASS$)
[V24]   49:  PRINT " ";PASS$;SPACE$(25-A)
[4T1]   5Ø:  DATEI$=DAT$
[FF6]   51:  IF M=4 AND DAT$<>"" THEN DATEI$=PREFIX$+".
        EXE"
[PF1]   52:  COLOR 7,12
[EZ1]   53:  LOCATE 2,55+A)
[VY3]   54:  IF DATEI$="" THEN DATEI$=" "
[3N3]   55:  PRINT LEFT$(DATEI$,25-A)
[LE1]   56:  COLOR 11,Ø
[BF1]   57:  LOCATE 4,2
[WZ2]   58:  PRINT SPACE$(78);C$
[NU2]   59:  GOSUB HILFSTEXT
[UUØ]   6Ø: MENU2
[S32]   61:  GOSUB TASTE1
[WZ2]   62:  IF IN1=27 THEN EXIT
[2T4]   63:  IF IN2=75 THEN M=M-1:GOTO MENU1
[YS4]   64:  IF IN2=77 THEN M=M+1:GOTO MENU1
[UV4]   65:  IF IN1<>13 THEN BEEP:GOTO MENU2
[E65]   66:  IF M<2 OR M>5 OR DAT$<>"" THEN MENU3
[RNØ]   67:  BEEP
[CZ1]   68:  GOSUB DATEI
[YW2]   69:  IF DAT$="" THEN MENU
[WUØ]   7Ø: MENU3
[1F1]   71:  LOCATE Ø,Ø
[1N9]   72:  ON M GOSUB DATEI,EDITOR,COMPILER,START,GW_
        BASIC,DOS,EXIT
[B74]   73:  IF M=1 THEN MENU1 ELSE MENU
[ØØØ]   74:
[KVØ]   75: DATEI
[7F1]   76:    COLOR 4,11
[7P1]   77:    LOCATE 2,46
[1C3]   78:    PRINT CHR$(26);CHR$(221)
[ME1]   79:    COLOR 12,Ø
[EF1]   8Ø:    LOCATE 4,3
[WA9]   81:    IF DAT$="" THEN KZ=1:PRINT "*.";SUFFIX$;S
        PACE$(72):GOTO DATEI1
[8PØ]   82:    KZ=Ø
[YA1]   83:    COLOR 7,Ø
[FR4]   84:    PRINT DAT$;SPACE$(69-LEN(DAT$));
[5Ø1]   85: DATEI1
[7F1]   86:    COLOR 4,11
[WS4]   87:    IF D=Ø THEN LOCATE 2,46:PRINT "  "
```

Listing 6. Eine leistungsfähige Benutzeroberfläche für SBasic ist »sbshell.bas«

```
[YA1]   88:   COLOR 7,Ø
[V72]   89:   LOCATE 4,3,1,1,1Ø
[S32]   9Ø:   GOSUB TASTE1
[P73]   91:   IF IN1=27 THEN RETURN
[KD5]   92:   IF IN1>31 THEN KZ=Ø:PRINT SPACE$(78)
[VØ4]   93:   IF IN2=77 THEN IN1=ASC(DAT$)
[766]   94:   IF IN2=83 THEN KZ=Ø:PRINT FNM$(2);" ":IN1
        =1
[SR5]   95:   IF IN1=Ø OR (IN1=9 AND D=Ø) THEN DATEI1
[7F1]   96:   COLOR 4,11
[7P1]   97:   LOCATE 2,46
[YA1]   98:   PRINT "  "
[KN1]   99:   LOCATE 2,5Ø
[ØF1]  1ØØ:   PRINT "  "
[423]  1Ø1:   IF IN1=9 THEN DATEI2
[V83]  1Ø2:   IF IN1<>13 THEN DATEI3
[DØ4]  1Ø3:   IF KZ=Ø OR D=Ø THEN RETURN
[8Ø1]  1Ø4: DATEI2
[S22]  1Ø5:   GOSUB SELECT
[LX1]  1Ø6:   GOTO DATEI4
[BØ1]  1Ø7: DATEI3
[8Ø4]  1Ø8:   IF IN1>1 THEN GOSUB KEYIN
[YA1]  1Ø9:   COLOR 7,Ø
[NV2]  11Ø:   LOCATE 4,3+LEN(DAT$)
[444]  111:   PRINT SPACE$(78-LEN(DAT$))
[EF1]  112:   LOCATE 4,3
[7W1]  113:   INPUT "",DAT$
[UØ3]  114:   IF DAT$="" THEN DATEI
[5X1]  115:   LOCATE ,,Ø,Ø,Ø
[LY1]  116:   GOSUB CHECK
[EØ1]  117: DATEI4
[BV4]  118:   IF MID$(DAT$,2,1)<>":" THEN DATEI5
[2R3]  119:   A(Ø)=&HBFB8+&H1ØØ*ASC(DAT$)
[ZH1]  12Ø:   A(1)=&H88ØE
[LK1]  121:   A(2)=&HCDC2
[WG1]  122:   A(3)=&HCB21
[Ø91]  123:   DEF SEG
[2Ø3]  124:   DRIVE!=VARPTR(A(Ø))
[K64]  125:   IF A(Ø)>&HB7 THEN CALL DRIVE!
[YL4]  126:   IF MID$(DAT$,3,1)="\" THEN DATEI5
[GF3]  127:   DAT$=FNL$(2)+"\"+FNM$(3)
[HØ1]  128: DATEI5
[VFØ]  129:   B=Ø
[894]  13Ø:   FOR C=LEN(DAT$)+1 TO 1 STEP-1
[2R4]  131:    IF MID$(DAT$,C,1)="\" THEN B=C:C=1
[811]  132:   NEXT C
[KPØ]  133:   C=B+1
[KP2]  134:   IF B=Ø THEN DATEI7
[LV4]  135:   IF RIGHT$(DAT$,2)<>":\" THEN B=B-1
[TU1]  136:   CHDIR FNL$(B)
[UØ1]  137: DATEI8
[XF1]  138:   GOSUB DIR
[YY2]  139:   IN1=13:GOSUB KEYIN
[RØ1]  14Ø: DATEI7
[5W1]  141:   DAT$=FNM$(C)
[1A9]  142:   IF FNL$(2)="*." THEN SUFFIX$=MID$(DAT$,3,
       3):DAT$="":GOTO DATEI8
[S95]  143:   IF DAT$=SPACE$(LEN(DAT$)) THEN DAT$=""
[PØ2]  144:   PREFIX$=DAT$
[PL2]  145:   I=INSTR(DAT$,".")
[FL9]  146:   IF I>Ø OR DAT$="" THEN PREFIX$=FNL$(I-1)
       ELSE DAT$=DAT$+"."+SUFFIX$
[651]  147:  RETURN
[ØØØ]  148:
[231]  149: EDITOR
[L62]  15Ø:   GOSUB ZEILE25
[573]  151:   SHELL EDITOR$+" "+DAT$
[651]  152:  RETURN
[ØØØ]  153:
[LE1]  154: COMPILER
[L62]  155:   GOSUB ZEILE25
[173]  156:   SHELL SBASIC$+" "+DAT$
[722]  157:   GOSUB TASTE
[651]  158:  RETURN
[ØØØ]  159:
[ØXØ]  16Ø: START
[L62]  161:   GOSUB ZEILE25
[842]  162:   SHELL PREFIX$
[722]  163:   GOSUB TASTE
[651]  164:  RETURN
[ØØØ]  165:
[LH1]  166: GW_BASIC
[L62]  167:   GOSUB ZEILE25
[S23]  168:   GWB$=PREFIX$+".GWB"
[8F1]  169:   KILL GWB$
[DØ4]  17Ø:   SHELL "SBASIC "+DAT$+" /GWB"
[KT3]  171:   OPEN "$.$" FOR OUTPUT AS #1
[DG1]  172:   C$=CHR$(34)
[GU5]  173:   PRINT #1,"1 CLS:KEY OFF:KILL";C$;"$.$";
[GZ4]  174:   PRINT #1,C$;":PRINT";C$;"LOAD";C$;
[Ø74]  175:   PRINT #1,";CHR$(34);";C$;GWB$;
[6W3]  176:   PRINT #1,C$;":LOAD";C$;GWB$"
[Ø51]  177:   CLOSE #1
[722]  178:   GOSUB TASTE
[UL5]  179:   SHELL "IF EXIST "+GWB$+" "+GWBASIC$+" $.$
       "
[651]  18Ø:  RETURN
```

```
[ØØØ]  181:
[1PØ]  182: DOS
[L62]  183:   GOSUB ZEILE25
[BC2]  184:   SHELL "COMMAND"
[651]  185:  RETURN
[ØØØ]  186:
[ASØ]  187: EXIT
[L62]  188:   GOSUB ZEILE25
[4MØ]  189:  END
[ØØØ]  19Ø:
[NWØ]  191: TASTE
[PE1]  192:   COLOR 14,Ø
[TT1]  193:   LOCATE 25,1,Ø
[386]  194:   PRINT "Weiter mit irgendeiner Taste ";
[811]  195: TASTE1
[MK1]  196:   IN$=INKEY$
[123]  197:   IF IN$="" THEN TASTE1
[N73]  198:   IN1=ASC(MID$(IN$,1,1))
[W73]  199:   IN2=ASC(MID$(IN$,2,1))
[651]  2ØØ:  RETURN
[ØØØ]  2Ø1:
[EXØ]  2Ø2: KEYIN
[XD1]  2Ø3:   DEF SEG=Ø
[6N1]  2Ø4:   POKE 1Ø5Ø,3Ø
[CN1]  2Ø5:   POKE 1Ø52,32
[6W1]  2Ø6:   POKE 1Ø54,IN1
[651]  2Ø7:  RETURN
[ØØØ]  2Ø8:
[BUØ]  2Ø9: CHECK
[NT6]  21Ø:   T$="Ø123456789ABCDEFGHIJKLMNOPQTSTUVWXYZ"
[UG3]  211:   T$=T$+"$&#%'()-@{}~!_*.\"
[YZ2]  212:   FOR I=1 TO LEN(DAT$)
[2Y1]  213:    J=ASC(FNM$(I))
[N24]  214:    IF J<97 OR J>122 THEN CHECK1
[RC4]  215:    MID$(DAT$,I,1)=CHR$(J AND &HDF)
[YYØ]  216: CHECK1
[PW3]  217:    J=INSTR(T$,MID$(DAT$,I,1))
[L87]  218:    IF J=Ø THEN I=99:BEEP:DAT$="":IN1=13:GOS
       UB KEYIN
[U11]  219:   NEXT I
[651]  22Ø:  RETURN
[ØØØ]  221:
[771]  222: ZEILE25
[YA1]  223:   COLOR 7,Ø
[DN1]  224:   LOCATE 25,1
[GG2]  225:   PRINT SPACE$(8Ø)
[PNØ]  226:   CLS
[651]  227:  RETURN
[ØØØ]  228:
[821]  229: RAHMEN
[YA1]  23Ø:   COLOR 7,Ø
[PNØ]  231:   CLS
[VL1]  232:   C$=CHR$(179)
[4G2]  233:   L$=STRING$(78,196)
[DG3]  234:   L1$=CHR$(195)+L$+CHR$(18Ø)
[LE1]  235:   COLOR 11,Ø
[5P1]  236:   LOCATE 1,1,Ø
[9B4]  237:   PRINT CHR$(218);L$;CHR$(191);C$
[VN1]  238:   LOCATE 2,8Ø
[HA2]  239:   PRINT C$;L1$;C$
[XN1]  24Ø:   LOCATE 1,53
[R72]  241:   PRINT CHR$(194)
[YN1]  242:   LOCATE 2,53
[ØA1]  243:   PRINT C$
[ZN1]  244:   LOCATE 3,53
[M72]  245:   PRINT CHR$(193)
[XN1]  246:   LOCATE 4,8Ø
[ØA1]  247:   PRINT C$
[9F1]  248:   LOCATE 5,1
[TD1]  249:   PRINT L1$
[1P1]  25Ø:   LOCATE 5,16
[R72]  251:   PRINT CHR$(194)
[HY1]  252:   FOR I=6 TO 24
[XF1]  253:    LOCATE I,1
[PD1]  254:    PRINT C$;
[6Ø5]  255:    IF I<23 THEN LOCATE I,16:PRINT C$
[KP1]  256:    LOCATE I,8Ø
[PD1]  257:    PRINT C$;
[U11]  258:   NEXT I
[9N1]  259:   LOCATE 23,1
[TD1]  26Ø:   PRINT L1$
[ZP1]  261:   LOCATE 23,16
[M72]  262:   PRINT CHR$(193)
[DN1]  263:   LOCATE 25,1
[FØ4]  264:   PRINT CHR$(192);L$;CHR$(217);
[651]  265:  RETURN
[ØØØ]  266:
[GXØ]  267: TITEL
[MR3]  268:   DATA " !### !###   !##   !### !###  !###"
[EN3]  269:   DATA "!#    !# !# !# !# !#    !#  !#   "
[SP3]  27Ø:   DATA " !##  !#### !####  !##   !#  !#   "
[KN3]  271:   DATA "   !# !# !# !# !#    !#  !#  !#   "
[PR3]  272:   DATA "!###  !###  !# !# !###  !###  !###"
[BC2]  273:   RESTORE TITEL
[W52]  274:   FOR I=12 TO 16
[E21]  275:    READ A$
[7P1]  276:    LOCATE I,31
```

```
[ØY1]  277:     FOR J=1 TO 34
[YX2]  278:      K=ASC(MID$(A$,J,1))-32
[ED1]  279:      COLOR K,Ø
[T83]  28Ø:      PRINT CHR$(244-22.5*K);
[X11]  281:     NEXT J
[U11]  282:    NEXT I
[FN1]  283:    LOCATE 24,2
[GA1]  284:    COLOR 3,1
[ZG6]  285:    PRINT " ASCII-/GW-Basic-Compiler V.2.Ø
               ";
[KB6]  286:    PRINT "(C)1989-91 DMV Widuch GmbH & Co. K
       G ";
[651]  287:   RETURN
[ØØØ]  288:
[RM1]  289: HILFSTEXT
[6TB]  29Ø:    DATA "Laufwerk / Pfad / ASCII-/GW-Basic-D
       ateinamen eingeben / auswählen"
[M26]  291:    DATA "ASCII-Datei in den Editor laden"
[DR6]  292:    DATA "SBASIC zur Compilierung aufrufen"
[MD8]  293:    DATA "Das zuvor compilierte Programm ausf
       ühren"
[VB8]  294:    DATA "Datei [.GWB] generieren und mit GW-
       Basic laden"
[P47]  295:    DATA "DOS-Ebene aufrufen; zurück mit EXIT
       "
[4T3]  296:    DATA "Programm beenden"
[K33]  297:    RESTORE HILFSTEXT
[GV1]  298:    FOR I=1 TO M
[E21]  299:     READ A$
[U11]  3ØØ:    NEXT I
[6B1]  3Ø1:    COLOR 7,4
[ØS1]  3Ø2:    LOCATE 4,M1
[S91]  3Ø3:    PRINT A$
[651]  3Ø4:   RETURN
[ØØØ]  3Ø5:
[C31]  3Ø6: SELECT
[RE1]  3Ø7:    COLOR Ø,11
[VU1]  3Ø8:    LOCATE 2,48,Ø
[YA1]  3Ø9:    PRINT "  "
[P21]  31Ø:    K=Ø:L=Ø
[8RØ]  311:    I=D-1
[G91]  312: SELECT1
[WX1]  313:    FOR J=Ø TO 15
[5A3]  314:     N=INSTR(DIR$(J+L),"\")
[9A6]  315:     IF J<>K THEN COLOR 1+(N=Ø),11:GOTO SELEC
       T2
[W72]  316:     DAT1$=DIR$(J+L)
[7S4]  317:     IF N=9 THEN DAT1$=LEFT$(DAT1$,9)
[PP2]  318:     N=INSTR(DAT1$," ")
[777]  319:     IF N>Ø THEN DAT1$=LEFT$(DAT1$,N-1)+RIGHT
       $(DAT1$,4)
[YA1]  32Ø:     COLOR 7,Ø
[P42]  321:     LOCATE 4,3,Ø,Ø,Ø
[NY4]  322:     PRINT DAT1$;SPACE$(76-LEN(DAT1$));
[6B1]  323:     COLOR 7,4
[M91]  324: SELECT2
[DR1]  325:     LOCATE J+6,2
[Ø43]  326:     PRINT " ";DIR$(J+L);" ";
[X11]  327:    NEXT J
[RT1]  328:    LOCATE 22,2,Ø
[7F1]  329:    COLOR 4,11
[FB8]  33Ø:    PRINT CHR$(25);CHR$(24);"PgDn/Up";CHR$(17
       );CHR$(217);"Esc";
[S91]  331: SELECT3
[S32]  332:    GOSUB TASTE1
[SCA]  333:    IF IN1=27 OR I<Ø THEN GOSUB DIR5:IN1=13:G
       OSUB KEYIN:GOTO SELECT9
[XP7]  334:    IF IN1=13 THEN DAT$=DAT1$:GOSUB DIR5:GOTO
        SELECT9
[SK3]  335:    IF IN2<>8Ø THEN SELECT5
[W91]  336: SELECT4
[Ø33]  337:    IF (K+L)<I THEN K=K+1
[SS7]  338:    IF K>15 THEN IF (K+L)<(I+1) THEN K=Ø:L=L+
       16 ELSE K=K-1
[742]  339:    GOTO SELECT1
[ØA1]  34Ø: SELECT5
[4L3]  341:    IF IN2<>72 THEN SELECT7
[4A1]  342: SELECT6
[VØ3]  343:    IF (K+L)>Ø THEN K=K-1
[VV6]  344:    IF K<Ø THEN IF L>15 THEN K=15:L=L-16 ELSE
        K=K+1
[742]  345:    GOTO SELECT1
[8A1]  346: SELECT7
[7L3]  347:    IF IN2<>81 THEN SELECT8
[445]  348:    IF (L+15)<I THEN K=15:GOTO SELECT4
[DGØ]  349:    K=Ø
[742]  35Ø:    GOTO SELECT1
[CA1]  351: SELECT8
[B35]  352:    IF IN2<>73 THEN BEEP:GOTO SELECT3
[HGØ]  353:    K=1
[YT2]  354:    IF L>15 THEN L=L-16
[C42]  355:    GOTO SELECT6
[GA1]  356: SELECT9
[651]  357:   RETURN
[ØØØ]  358:
[BNØ]  359: DIR
[9G5]  36Ø:    DATA 1E,ØE,1F,B4,2F,CD,21,89,DE,83,C6,1E
[995]  361:    DATA E9,2D,ØØ,89,DF,47,B9,1Ø,ØØ,88,6D,Ø2
[WD5]  362:    DATA 89,DA,B4,4F,38,2F,74,Ø2,FE,CC,CD,21
[U65]  363:    DATA 72,13,2B,CØ,88,27,Ø6,1E,Ø7,1F,8A,44
[ØK5]  364:    DATA F7,AB,B1,ØC,AC,AB,49,75,FB,1F,CB,9Ø
[FB5]  365:    DATA E8,ØØ,ØØ,5B,83,C3,Ø6,EB,CA,2A,2E,2A
[U75]  366:    DATA ØØ,ØØ,1E,2B,CØ,8E,D8,B4,19,CD,21,Ø4
[Ø55]  367:    DATA 41,A2,ØØ,Ø2,C7,Ø6,Ø1,Ø2,3A,5C,BE,Ø3
[894]  368:    DATA Ø2,2A,D2,B4,47,CD,21,1F,CB
[642]  369:    RESTORE DIR
[Ø91]  37Ø:    DEF SEG
[KT3]  371:    DIRECTORY!=VARPTR(A(Ø))
[222]  372:    FOR D=Ø TO 1Ø4
[E21]  373:     READ A$
[TE4]  374:     POKE DIRECTORY!+D,VAL("&H"+A$)
[B11]  375:    NEXT D
[B83]  376:    PASS!=DIRECTORY!+74
[NK1]  377:    CALL PASS!
[XD1]  378:    DEF SEG=Ø
[B51]  379:    PASS$=""
[DT2]  38Ø:    FOR D=&H2ØØ TO &H22Ø
[ZA1]  381:     A=PEEK(D)
[TZ3]  382:     IF A>64 AND A<91 THEN A=A+32
[WL6]  383:     IF A>Ø THEN PASS$=PASS$+CHR$(A) ELSE D=&
       H22Ø
[B11]  384:    NEXT D
[Ø91]  385:    DEF SEG
[BGØ]  386:    D=3
[WPØ]  387: DIR1
[E53]  388:    A=INSTR(D,PASS$,"\")
[6P9]  389:    IF A>Ø THEN PASS$=LEFT$(PASS$,D-1)+MID$(P
       ASS$,A):D=D+1:GOTO DIR1
[FL5]  39Ø:    IF LEN(PASS$)>3 THEN PASS$=PASS$+"\"
[DX1]  391:    FOR D=Ø TO 8Ø
[SD3]  392:     DIR$(D)=STRING$(12,177)
[B11]  393:    NEXT D
[ZFØ]  394:    D=Ø
[XPØ]  395: DIR2
[CL2]  396:    CALL DIRECTORY!
[4W2]  397:    IF A(36)=Ø THEN DIR5
[YKØ]  398:    L$=""
[Z62]  399:    FOR P=36 TO 47
[LU5]  4ØØ:     IF A(P)>Ø THEN L$=L$+CHR$(A(P)) ELSE P=4
       7
[F21]  4Ø1:    NEXT P
[P65]  4Ø2:    IF A(35)=16 THEN L$=L$+"\":GOTO DIR3
[TN5]  4Ø3:    IF RIGHT$(L$,4)<>"."+SUFFIX$ THEN DIR2
[R82]  4Ø4:    P=INSTR(L$,".")
[7E7]  4Ø5:    IF P>Ø THEN L$=LEFT$(L$,P-1)+SPACE$(9-P)+
       MID$(L$,P)
[YPØ]  4Ø6: DIR3
[WC3]  4Ø7:    L$=L$+SPACE$(12-LEN(L$))
[3Ø7]  4Ø8:    IF LEFT$(L$,1)="." THEN D=1:DIR$(Ø)=L$:GO
       TO DIR2
[5V1]  4Ø9:    FOR J=Ø TO D
[ZA3]  41Ø:     IF L$>DIR$(J) THEN DIR4
[G33]  411:     FOR K=D TO J STEP -1
[CC2]  412:      DIR$(K+1)=DIR$(K)
[Ø21]  413:     NEXT K
[2F1]  414:     DIR$(J)=L$
[VKØ]  415:     J=D
[ZPØ]  416: DIR4
[X11]  417:    NEXT J
[WPØ]  418:    D=D+1
[GF2]  419:    IF D<8Ø THEN DIR2
[ØRØ]  42Ø: DIR5
[YX1]  421:    FOR J=Ø TO 16
[D53]  422:     N=INSTR(DIR$(J),"\")
[E82]  423:     COLOR 1+(N=Ø),11
[DR1]  424:     LOCATE J+6,2
[ØX2]  425:     PRINT " ";DIR$(J);" ";
[LE1]  426:     COLOR 11,Ø
[PD1]  427:     PRINT C$;
[X11]  428:    NEXT J
[651]  429:   RETURN
```

Listing 6. Eine leistungsfähige Benutzeroberfläche für SBasic ist »sbshell.bas«

```
[DA1]    1: echo off
[WSC]    2: SBSHELL /DATEI=%1 /SUFFIX=BAS /EDITOR=EDITO
            R /SBASIC=SBASIC /GWBASIC=GWBASIC
```

Listing 7. Die Stapeldatei »sb.bat« ermöglicht den komfortablen Aufruf der Oberfäche

Impressum

Herausgeber
Christian Widuch

Leitung des Redaktionsbüros
Volker Everts

Chefredakteur
Volker Everts (ev)

Stellvertretender Chefredakteur
Ulrich Eike (ue)

Chef vom Dienst, Aktuelles
Gabi Börger (gb)

Ressort Anwendung
Marina Baader (ba), Alois Kneisle (al), Herwig Weihe (hw), Gerda Hutt (Prakt.)

Ressort Hardware
Leitung: Wolfgang Hofner (wh)
Jörn Hellwig (jh), Ralph Kunze (rk)

Ressort Programmierung
Leitung: Dieter Bäumler (db)
Ralf Morgenstern (rm), Wolfgang Nefzger (wn), Erhard Thomas (et), Kerrin Uhlmann (Prakt.)

Redaktionsmarketing
Ingo Everts (ie)

Schlußredaktion
Karin Adlmannseder (ka), Elisabeth Foessleitner (el)

Redaktionsassistenz
Anke Ludwig, Christine Reichherzer

Produktionsleitung
Uwe Siebert

Layout
Patricia Reifenhausen, Michaela Küppers

Fotografie
Leitung: Achim Schulte
Heinz Harcuba

Fotosatz
Leitung: Claudia Ebbrecht
Gabriela Joseph

Montage/Reprografie
Leitung: Margarete Schenk, Helmut Skoupy
Manuela Eska, Andrea Gundlach

Werbeleitung
Jutta Himmelmann

Werbegrafik
Mohamed Hawa
Uwe Kalischinsky

Anzeigenleitung
Wolfgang Brill

Anzeigenverkauf für PLZ 1, 4, 5
Gerlinde Rachow (0 56 51) 80 93 90
Sylvia Stephani (0 56 51) 80 93 80
Bernd Heckmann (0 56 51) 80 93 81
Karina Ehrlich (0 56 51) 80 93 71

Anzeigenverkauf für PLZ 2 + 3
DMV-Verlagsbüro Hamburg
Ohlsdorfer Straße 34, 2000 Hamburg 60
Tel.: (0 40) 46 12 33, Telefax (0 40) 47 43 10
Leitung: Sylvia Ehrenpfordt
Anzeigenverkauf: Sabine Bindseil, Ralph Streng

Anzeigenverkauf für PLZ 6 – 8
DMV-Verlagsbüro München
Zaunkönigweg 2c, 8000 München 82,
Telefon: (0 89) 4 39 10 87, Telefax: (0 89) 4 39 10 80
Leitung: Britta Fiebig
Anzeigenverkauf: Peter Schätzle, Hannelore David, Ilona Sehm

Repräsentant für England: Gerry Rhoades-Brown

Anzeigenverwaltung und Disposition
Andrea Giese, Karina Ehrlich, Beate Kranz, Christina Wabra

Anschrift Redaktion:
Redaktion DOS International
Gruberstraße 46c, 8011 Poing bei München
Telefon: (0 81 21) 7 69-0, Telefax: (0 81 21) 7 90 46
Btx: * dmv # oder * 20072 #

Sie erreichen die Redaktion werktags telefonisch von 15 bis 17 Uhr.

Anschrift Verlag:
DMV Daten- und Medien-Verlag
Widuch GmbH & Co. KG
Fuldaer Straße 6, 3440 Eschwege
Telefon: (0 56 51) 8 09-0
Telefax: (0 56 51) 80 93 33

Vertrieb
Verlagsunion Erich Pabel-Arthur Moewig KG (VPM)
Friedrich-Bergius-Straße 20
6200 Wiesbaden

Druck
Druckerei Jungfer, 3420 Herzberg

Bezugspreise
DOS International erscheint monatlich.
Einzelpreis DM 7,50/sfr. 7,50/öS 65, – /DR 1000

Abonnementpreise
Die Preise verstehen sich grundsätzlich einschließlich Porto und Verpackung.

Inland:
Bezugspreis 12 Ausgaben: DM 82,50
Bezugspreis 6 Ausgaben: DM 42, –
Europäisches Ausland:
Bezugspreis 12 Ausgaben: DM 110, –
Bezugspreis 6 Ausgaben: DM 55, –
Außereuropäisches Ausland:
Bezugspreis 12 Ausgaben: DM 150, –
Bezugspreis 6 Ausgaben: DM 75, –

Bankverbindungen:
Postscheck 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 6 bzw. 12 Monate, wenn es nicht mindestens 6 Wochen vor Ablauf beim Verlag schriftlich gekündigt wird.

Manuskripte, Programme:
bitte an die Redaktion
DOS International, Gruberstraße 46c, 8011 Poing, 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 Basteleien 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.

Mitglied der Informationsgemeinschaft zur Feststellung der Verbreitung von Werbeträgern e.V. (IVW),
Bad Godesberg.
ISSN 0933-1557

BCi DisAsm i86
Intelligenter Disassembler Version 1.0

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, das 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 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 compilierten 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.

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**

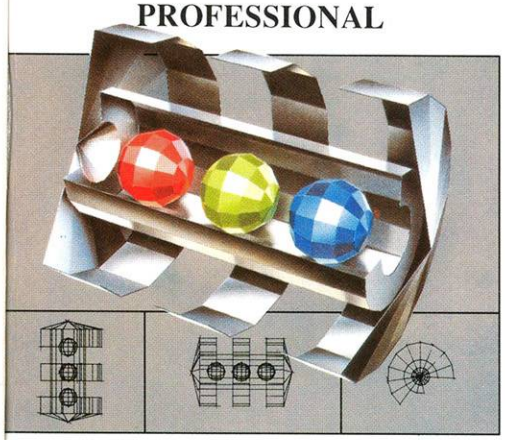
Bitte benutzen Sie die Bestellkarte.

DMV-Verlag · Postfach 250 · 3440 Eschwege

DMV Daten- und Medienverlag

DOS SOFTWARE

3D-DRAW PROFESSIONAL



Komfortables, dreidimensionales Konstruktionsprogramm

Die Leistungsmerkmale von 3D-DRAW PROFESSIONAL:

3D-Modus

• Kantenmodelle und Flächenmodelle mit und ohne Lichtquelle animierbar • anwählbare Lichtquelle • Rotation und Translation um beliebige Raumachsen • manipulierbare Bereiche • konfigurierbare Druckertreiber • 2000 Punkte und Flächen sowie 4000 Kanten darstellbar • Einzelbehandlung von Körpern • Leuchtbalken-Menüs zur besseren Anwählbarkeit

2D-Modus

• Erhöhte Arbeitsgeschwindigkeit • Mauszeiger-Bedienung auch über die Tastatur • Zurücknehmen des letzten Arbeitsschrittes • Quick-Modus für einfache Grundformen wie Kreise oder Vielecke • Werkzeuge zum Pressen, Stauchen und Dehnen von Objekten, Körpern und Bereichen

Systemvoraussetzungen: PC, XT, AT oder Kompatibler mit mindestens 640 KByte RAM. **Grafikkarte:** Alle üblichen Typen. **Ausgabemedien:** Epson-kompatible Nadeldrucker, HP- kompatible Plotter, Postscript-Drucker. **Sonstiges:** Microsoft-kompatible Maus unbedingt erforderlich, MS-DOS ab Version 3.0

nur DM 199,– *

3D-DRAW 1.0 **nur DM 99,– ***

S O F T W A R E

Wetterberatung und Flugplanung für den Microsoft Flugsimulator III und IV

Der neu überarbeitete NAVIGATOR 4.5 bietet jetzt noch mehr Bedienungskomfort und arbeitet noch enger mit dem Microsoft Flugsimulator zusammen.

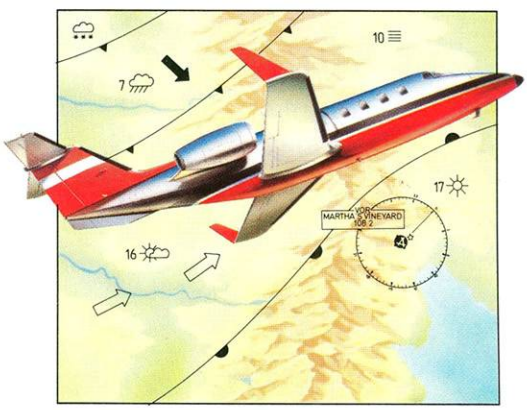
Das Leistungsspektrum auf einen Blick:

• Auf Wunsch automatische Datenübertragung an den Flugsimulator • 1237 Flughäfen stehen zur Verfügung • eine interne Datenbank bietet 610 komplette Flugrouten mit allen zu überfliegenden Funkfeuern, Kursen und Zwischenzeiten • eine weitere Datenbank enthält Informationen über 900 Funkfeuer • 400 zusätzliche Routen lassen sich auf Wunsch anfertigen • von jedem Flughafen aus sind Rundflüge möglich • NAVIGATOR 4.5 läßt sich jetzt per Maus bedienen

Systemvoraussetzungen:
Microsoft Flugsimulator 3.0 oder 4.0, 512 KByte Arbeitsspeicher, jede, vom Flugsimulator unterstützte Grafikkarte

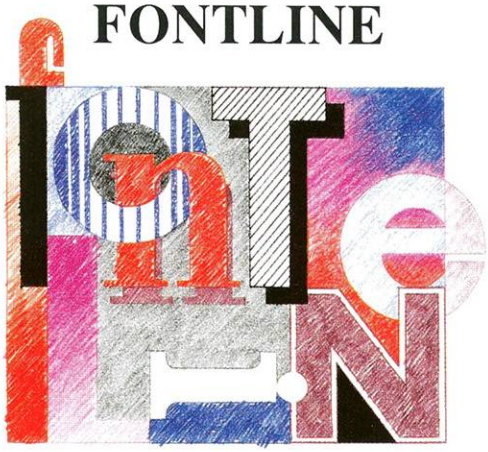
nur DM 69,– *

NAVIGATOR 4.5



S O F T W A R E

FONTLINE



Softfont-Manipulation und Druckertreiber-Generierung für Microsoft-Word 4.0/5.0 sowie Ventura Publisher 1.1/1.2/2.0

FONTLINE erzeugt aus bestehenden Softfonts für HP-Laserjet Plus und Serie II, sowie dazu kompatible Laserdrucker nahezu beliebig viele neue Variationen. Die bekanntesten wie Inline, Outline, 3D, Schatten, Relief, Invers, Kursiv und Landscape können sofort von allen ladbaren Zeichensätzen angefertigt werden.

Erzeugen Sie Zeichensätze mit Marmorstruktur, Chrom-Glanz, Holzmaserung oder beliebigen anderen Mustern, indem Sie bekannte Zeichenprogramme wie Publisher's Paintbrush oder Deluxe Paint II zu Hilfe nehmen.

FONTLINE macht alle vorhandenen und neu erzeugten Fonts auch für Microsoft Word 4/5 und Ventura Publisher zugänglich.

Ihr Programm unterstützt die Einbindung neuer Fonts beispielsweise in Microsoft Windows 2.1 oder Aldus Pagemaker? Dann versorgt Sie zukünftig FONTLINE mit einer umfangreichen Schriftenvielfalt.

Mitgelieferte, einsatzfertige Softfonts in diversen Größen: Alte Schwabacher, Futura Black, Bembo, Keys, Sybolic, PC-Linedraw

Systemvoraussetzungen: PC, XT, AT, PS/2 und Kompatible mit mindestens 512 KByte RAM, HP- Laserjet Plus/Serie II oder kompatibler Laserdrucker, MS-DOS 3.2 und höher

Erscheinungstermin: 1. Quartal 1991

nur DM 299,– *

S O F T W A R E

* 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.

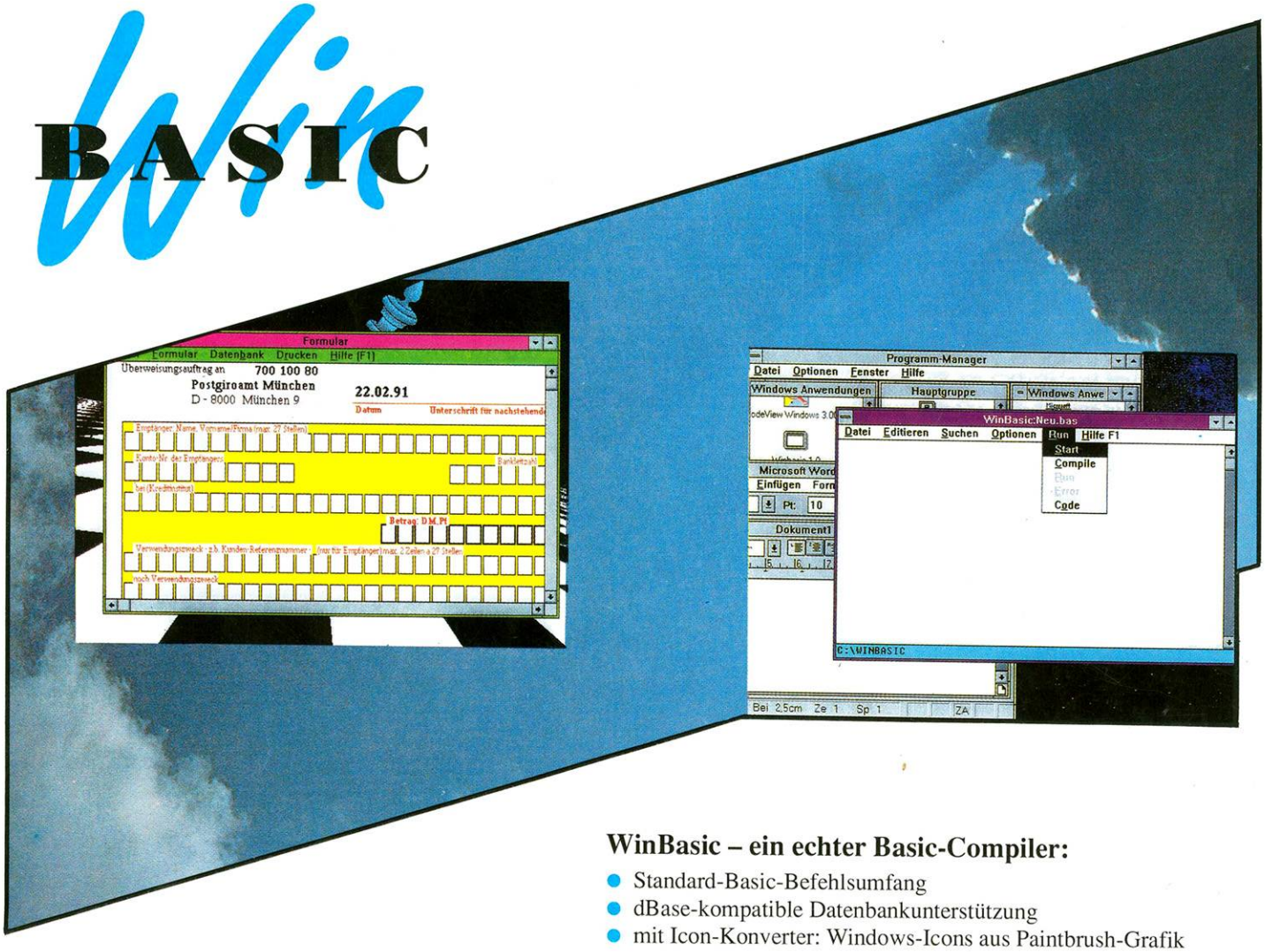
DMV-Verlag · Postfach 250 · 3440 Eschwege

DMV Daten- und Medienverlag

Für Microsoft Windows ab Version 3

Der einfache Weg zur Windows-Programmierung

WinBASIC



Kein Windows-Software-Development-Kit nötig! Einfach Windows und WinBasic laden, und schon kann's losgehen!

WinBasic – ein echter Basic-Compiler:

- Standard-Basic-Befehlsumfang
- dBase-kompatible Datenbankunterstützung
- mit Icon-Konverter: Windows-Icons aus Paintbrush-Grafik

WinBasic unterstützt alle wichtigen Windows-Funktionen:

Maus • Fenster • Dialogboxen • Dateiauswahlboxen • EMS und virtueller Speicher • Fonts • Multitasking • Clipboard-Datenaustausch

WinBasic, Basic unter Microsoft Windows **DM 299,— ***

Zur Entspannung und als Konzentrationstraining

MEMO, Das Gesellschaftsspiel für helle Köpfe **DM 69,— ***

* Unabhängig von der Anzahl der bestellten Programme berechnen wir für das Inland DM 4,— bzw. für das Ausland DM 6,— Porto und Verpackung. Bitte benutzen Sie die Bestellkarte.

DMV-Verlag • Postfach 250 • 3440 Eschwege

DMV Daten- und Medienverlag