Januar/Februar '85 DM 6,50

PEEKER

MAGAZIN FÜR APPLE-COMPUTER

1/2 - 85

Poor Man's RAM-Disk

Die RAM-Karten von IBS

Universal-Modem WS 2000

Quickcopy für ProDOS

Garbage-Collection

Pascal-Directory

Ikonen und Deixis

Mit Sonderteil **Steuer '84**

Hüthig PUBLIKATION

EDITORIAL

Dieses Peeker-Heft bringt eine geballte Phalanx nützlicher und extrem schneller Programme.

Unter dem Rubrum „RAM-Disk" beginnen wir mit einer Serie von RAM-Disk-Drivern und sonstigen Utilities zu RAM-Karten unterschiedlicher Produzenten (Apple-LC, Apple-64K-Erweiterung, IBS-RAM-Karten, Balfer-Interface, Titan-Karten u. a.). An einer kleinen „Poor Man's RAM-Disk" wird zunächst das Prinzip eines RAM-Disk-Drivers erläutert und dann anhand der großen IBS-RAM-Karten technisch vertieft.

QUICKCOPY ist ein ProDOS-Kopierprogramm, das für Laufwerke mit 35–80 Spuren ausgelegt ist und eine normale 35-Spur-Diskette in nur 33 Sekunden dupliziert. Daneben wird ein nicht ganz so schnelles PRODOS.COPYA gelistet, das wie das alte DOS.COPYA ohne 64K-Karte auskommt.

Wenn Sie bislang Probleme mit der Applesoft-Stringverwaltung unter DOS 3.3 hatten, so können Sie jetzt aufatmen. Eine „blitzartige" Garbage-Collection-Routine, die teilweise auf dem ProDOS-FRE-Befehl basiert, ist sage und schreibe bis zu 2000 mal schneller als die alte Interpreter-Routine. Da sich diese neue Routine mit einem in die Language Card gelegten DOS 3.3 verträgt, kann in den unteren 48K ein riesiger Stringpool verwaltet werden, was unter ProDOS leider nicht mehr möglich ist.

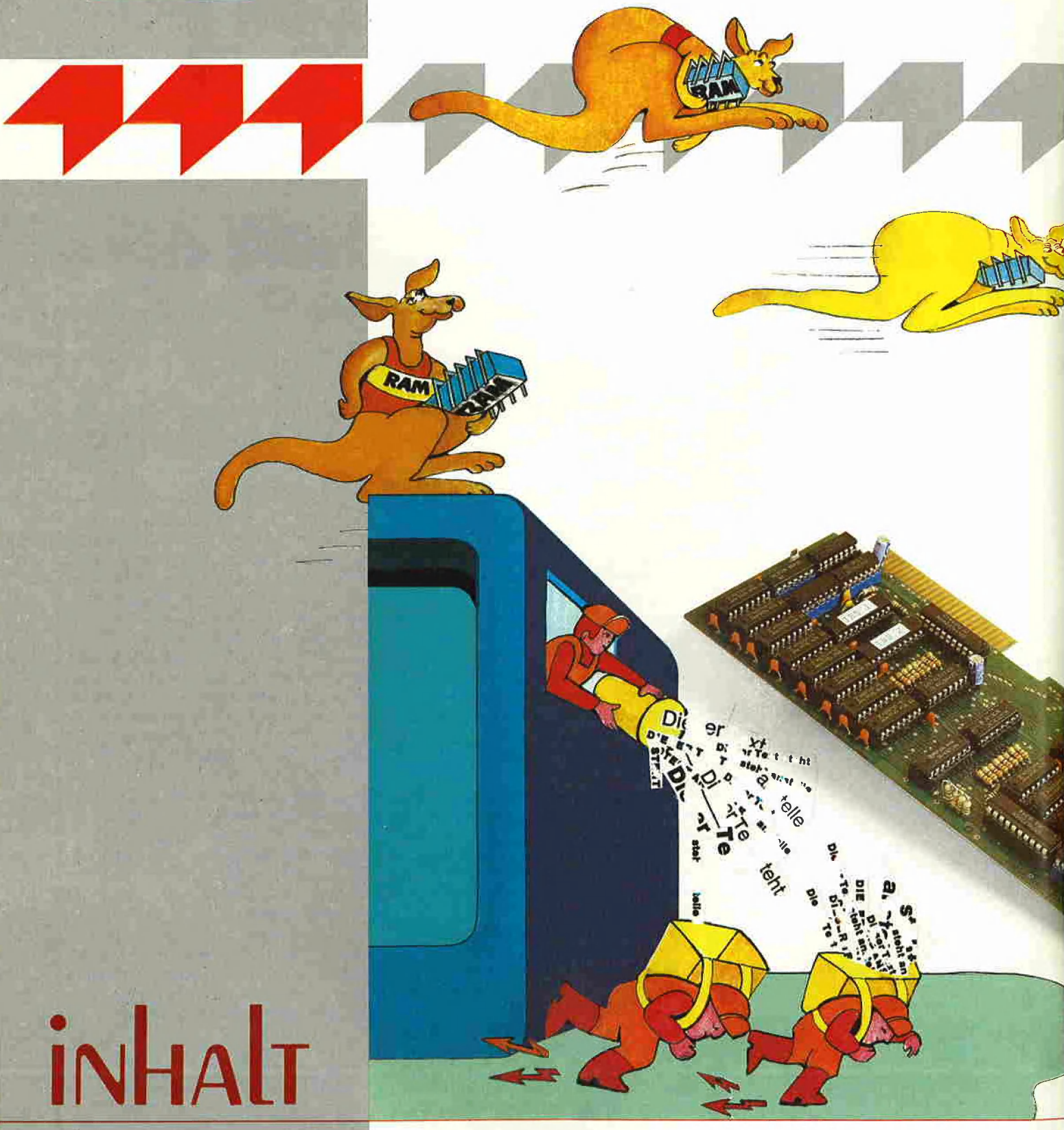
Schließlich bringen wir neben einer Reihe weiterer Utilities in unserem Sonderteil ein sehr umfangreiches MBASIC-Programm für die Lohn- und Einkommensteuererklärung 1984.

Leser, die nur selten oder gar nicht programmieren, werden sicherlich einige der im „Peeker" enthaltenen professionellen Programme nur mit Mühe verstehen können. Deshalb gibt es die „Peeker"-Sammeldiskette, die alle abgedruckten Programme in sofort einsatzfähiger Form enthalten. Viele unserer Utilities sind auf dem Softwaremarkt sonst nur für teures Geld zu haben. Beispielsweise ist gerade in den USA ein ProDOS-Kopierprogramm für $ 30 (rund DM 100,–) erschienen, das genauso schnell wie unser QUICKCOPY ist. Für einen RAM-Disk-Driver würde man einen ähnlichen Betrag zahlen müssen, und für ein Einkommen- und Lohnsteuerprogramm müßte man noch erheblich tiefer in die Tasche greifen. Bei uns bekommen Sie das alles zusammen für eine nominale Gebühr von nur DM 20,– pro Diskette im Fortsetzungsbezug.

Noch ein Hinweis in eigener Sache: Das Peeker-Poster wird erst am 25. 1. 85 verschickt, weil in die Tabellen noch neuere Erkenntnisse zu ProDOS sowie zum 65C02 eingearbeitet werden mußten.

Ulrich Stiehl

Hüthig
PUBLIKATION



inhalt

1/2-85

Impressum

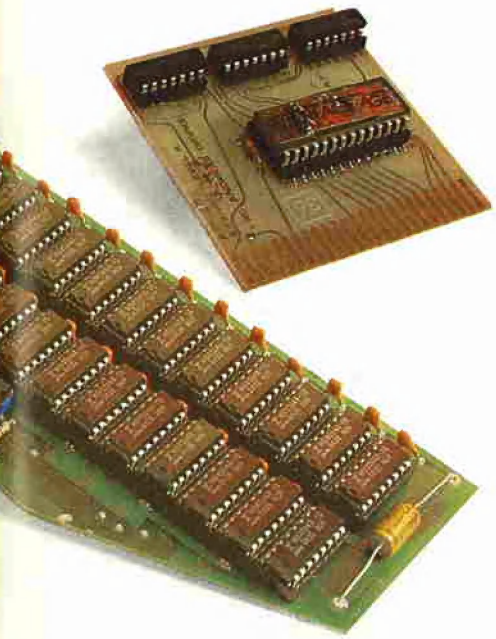
Peeker
Magazin für Apple-Computer
2. Jahrgang 1985
ISSN 0176–9200

Verleger und Herausgeber:
Dipl.-Kfm. Holger Hüthig
Geschäftsführung Zeitschriften:
Heinz Melcher
Chefredakteur:
Ulrich Stiehl (us) Tel. (0 62 21) 48 93 52

Anzeigenleitung:
Jürgen Maurer, Tel. (0 62 21) 48 92 18
z. Zt. gilt Anzeigenpreisliste Nr. 2
Vertriebsleitung:
Ruth Biller, Tel. (0 62 21) 48 92 80
Produktionsleitung: Gunter Sokollek
Gestaltung: Rainer Schmitt

PEEKER

MAGAZIN FÜR APPLE-COMPUTER



Verlag:
Dr. Alfred Hüthig Verlag GmbH
Im Weiher 10, Postfach 10 28 69
6900 Heidelberg
Telefon (0 62 21) 4 89-1
Telex 4-6 17 27 hued d.

Erscheinungsweise:
12 Hefte jährlich,
Jan./Febr. Doppelheft.
Erscheinungstag jeweils 1 Woche vor Monatsbeginn.
Jahresabonnement DM 58,–, einschließlich MWSt, im Inland portofrei.
Einzelheft DM 6,50

Zahlungen: an den Dr. Alfred Hüthig-Verlag GmbH, D-6900 Heidelberg 1: **Postscheckkonten:** BRD: Karlsruhe 48545-753; Österreich: Wien 7555888; Schweiz: Basel 40-24417; Niederlande: Den Haag 145728; Italien: Mailand 47718; Belgien: Brüssel 723026; Dänemark: Kopenhagen 34969; Norwegen: Oslo 99424; Schweden: Stockholm 547776-5

Bankkonten: Landeszentralbank Heidelberg 67207341; BLZ 67200000; Deutsche Bank Heidelberg 0 2165 041; BLZ 67270003; Bezirkssparkasse Heidelberg 20451, BLZ 67250020.

Herstellung: Heidelberger Verlagsanstalt
Printed in Germany

Die Language Card (LC) des Apple II ist ursprünglich als Sprachkarte für Integer-Basic, Pascal usw. gedacht worden, doch kann sie auch generell als Datenspeicher sowie speziell als RAM-Disk (Pseudo-Disk, Disk-Emulator) eingesetzt werden. Nachstehend wird ein mit allen 48K-DOS-Versionen (DOS 3.3, Diversi-DOS, David-DOS usw.) verträglicher und damit sehr flexibler Driver vorgestellt, der mit einer Datenübertragungsrate von über 50K/s aufwarten kann. Unsere Serie über RAM-Disk-Driver beginnt mit diesem LC-Driver, weil sich hier die Prinzipien einer RAM-Disk besonders einfach erläutern lassen. Die Besonderheit dieses LC-Drivers gegenüber ähnlichen LC-Drivern, die bereits in „Nibble" und anderen Zeitschriften publiziert worden sind, liegt darin, daß unser Driver selbst in der LC residiert, womit keine Kompatibilitätsprobleme mit unterschiedlichen DOS-Varianten auftreten können.

Poor Man's RAM-Disk

von Ulrich Stiehl

RAM-Disk-Driver für die Language Card

1. Allgemeines zur RWTS

DOS 3.3 läßt sich in 3 Bereiche gliedern:

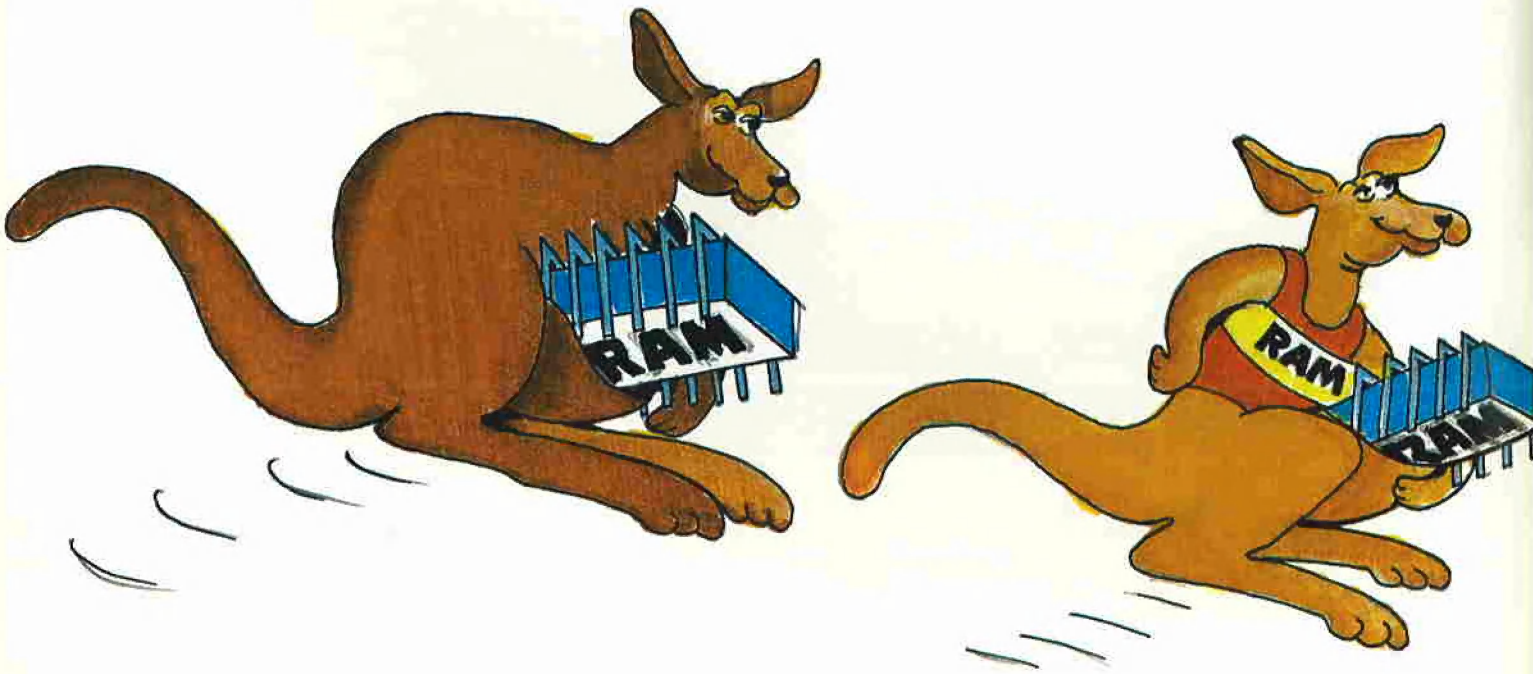
1. Der **Command-Interpreter** („Befehlsdeuter") analysiert die eingegebenen Befehle wie LOAD PROGRAMM usw.
2. Der **File-Manager** („Dateiverwalter") überwacht das Laden/Speichern von Dateien.
3. Der **Disk-Driver** („Diskettentreiber") oder die RWTS (= Read Write Track Sector) führt das physische Laden/Speichern auf Spur-Sektor-Ebene durch.

Der Command-Interpreter gibt die Befehle an den File-Manager weiter, der seinerseits die benötigten Sektoren über den Disk-Driver abruft. Der normale Disk-Driver für physische Laufwerke beginnt, wenn DOS 3.3 in den unteren 48K liegt, stets bei der Speicherstelle $BD00. Vor dem Sprung zum Disk-Driver bzw. zur RWTS sind bereits vom File-Manager die gewünschten Parameter in dem sog. **IOB** (= Input-Output-Block) ab Speicherstelle $B7E8 gesetzt worden, die dann von der RWTS übernommen werden. Ein Maschinenprogramm kann anstelle des normalen IOB ab $B7E8 seinen eigenen IOB, der irgendwo im Speicher liegen kann, benutzen und im übrigen den File-Manager ganz umgehen, doch muß man dann vor dem Sprung zur RWTS (= JSR $BD00 oder besser JSR $03D9 = RWTS-Vektoradresse) das Y-Register mit dem Low Byte und das A-Register mit dem High Byte der Adresse des eigenen IOB laden.

Der IOB umfaßt folgende Parameter:

Input-Output-Block
(ab $0300 mit Beispielwerten)

0 $0300: 01 Konstante
1 $0301: 30 Jetzt-Slot 3 (3 * 16)
1 $0302: 01 Jetzt-Drive 1
0 $0303: FE Jetzt-Volume-Nummer 254
1 $0304: 11 Spur 17
1 $0305: 00 Sektor 0
0 $0306: 11 LL DCT ($0311)
0 $0307: 03 HH DCT
1 $0308: 00 LL Puffer ($1000)
1 $0309: 10 HH Puffer
0 $030A: 00 entfällt
0 $030B: 00 entfällt
1 $030C: 01 Read-Befehl (02 = Write)



0 $030D: 00 Fehler-Code (danach)
0 $030E: 00 Vorher-Volume-Nummer
0 $030F: 60 Vorher-Slot 6
0 $0310: 02 Vorher-Drive 2
0 $0311: 00 DCT-Konstante
0 $0312: 01 DCT-Konstante
0 $0313: EF DCT-Konstante
0 $0314: D8 DCT-Konstante

Aufruf der RWTS

```
$0315: A0 00      LDY #$00
                  (Low $0300)
$0317: A9 03      LDA #$03
                  (High $0300)
$0319: 20 D9 03   JSR $03D9
                  (RWTS-Vektor)
$031C: 60         RTS
```

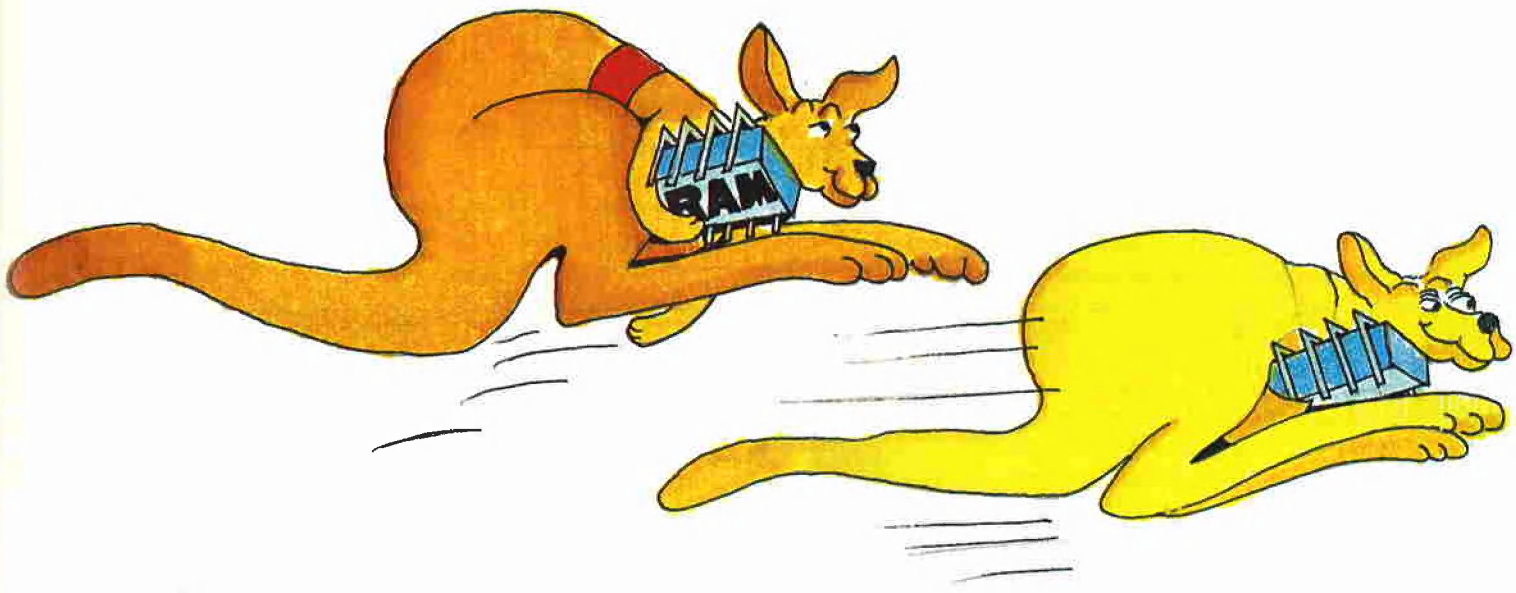
Das obige RWTS-Beispiel würde – nach CALL –151 mit 315G gestartet – den Sektor 0 von Spur 17 von der LC-RAM-Disk im imaginären Slot 3 in den Puffer $1000-$10FF einlesen. DCT steht übrigens für Device Characteristics Table und umfaßt 4 Konstanten für den Laufwerk-Controller, die allerdings bei einem RAM-Disk-Driver nicht benötigt werden. Als Jetzt-Parameter gelten die IOB-Slot-Drive-Volume-Werte vor dem RWTS-Aufruf. Nach der Rückkehr von der RWTS werden die Jetzt-Parameter zu Vorher-Parametern. (Um den Versuch der normalen RWTS, den nicht-vorhandenen Motor der RAM-Disk abzustellen, zu unterbinden, aktualisiert unser LC-RAM-Disk-Driver nicht die Vorher-Parameter.)
Vor den einzelnen Bytes des IOB steht entweder eine 0 oder eine 1. Falls man wie hier mit einem externen IOB die RAM-Disk per RWTS direkt anspricht, so brauchen vor dem Aufruf der RWTS theoretisch nur die mit 1 markierten Speicherstellen initialisiert werden, da die mit 0 markierten Speicherstellen bei der RAM-RWTS im Gegensatz zur normalen RWTS keine Rolle spielen. Aus Gründen der Mehrfachnutzung eines einzigen IOB innerhalb eines Assemblerprogramms empfiehlt es sich jedoch, den IOB vollständig anzulegen.

Eine RAM-Disk unterscheidet sich von der normalen Diskette lediglich dadurch, daß die Daten auf der RAM-Karte anstatt auf dem magnetischen Datenträger gespeichert werden. Dies hat den Nachteil, daß mit dem Ausschalten des Apple der RAM-Disk-Inhalt verlorengeht, der somit vor Beendigung der Arbeit stets auf einer physischen Diskette gesichert werden muß. Der Vorteil der RAM-Disk liegt in dem ungleich schnelleren Zugriff, weil im Gegensatz zum Diskettenlaufwerk keine physischen Teile bewegt werden (Einschalten des Laufwerkmotors, Rotieren der Diskette, Positionieren des Lesekopfs usw.).

Der normale Disk-Driver übernimmt u.a. die Spur- und Sektornummer sowie die Pufferadresse (= Pufferanfang) aus dem IOB. Wenn ein Lesebefehl vorliegt (01 = Read), so wird der 256 Bytes umfassende Sektor von der physischen Diskette in den 256 Bytes umfassenden Puffer eingelesen. Wenn umgekehrt ein Schreibbefehl vorliegt (02 = Write), so wird der Pufferinhalt auf den entsprechenden Diskettensektor geschrieben. De facto ist eine Nibble-Kodierung zwischengeschaltet, doch können wir dies hier ignorieren.
Ein RAM-Disk-Driver verfährt analog zum normalen Disk-Driver. Beim Lesebefehl wird der Pufferinhalt in einen bestimmten Bereich der RAM-Karte eingelesen. Umgekehrt wird beim Schreibbefehl ein bestimmter Bereich der RAM-Karte in den Puffer geschrieben. Lesen und Schreiben sind hier Move- oder Speicherverschiebebefehle.
Ein RAM-Disk-Driver besteht grundsätzlich aus 2 Teilprogrammen: einem Modul, das die RAM-Disk „initialisiert", sowie einem Modul, das den eigentlichen Driver installiert. Beide Module können wie bei unserer **RAMDISKLC**-Utility zu einem Gesamtprogramm vereint werden.

2. RAM-Disk-Organisation

Der für die RAM-Disk zur Verfügung stehende Speicherbereich – in unserem Fall die LC – kann völlig beliebig in „Spuren" und „Sektoren" (256-Byte-Bereiche) eingeteilt werden. Aus **Diagramm 1** ersehen wir, daß z.B. der Bereich $D000-$DFFF der LC-Bank 2 der Spur 19 mit den Sektoren 0–15 entspricht. Sektor 0 von Spur 19 nimmt den Teilbereich $D000-$D0FF ein, Sektor 1 den Teilbereich $D100-$D1FF usw. Bei einer normalen Diskette gibt es 35 Spuren. Die Spuren 0–2 enthalten das DOS, die Spur 17 umfaßt den Catalog, und die restlichen Spuren ab Spur 18 aufwärts sowie ab Spur 16 abwärts enthalten die eigentlichen Dateien. Da auf einer RAM-Disk normalerweise nicht 35 Spuren un-



tergebracht werden können und im übrigen auch von einer RAM-Disk nicht gebootet werden kann, muß das sog. VTOC (= Volume Table of Contents), das sich immer auf Sektor 0 von Spur 17 befindet, so modifiziert bzw. „gekürzt" werden, daß das RAM-Disk-VTOC die tatsächlich verfügbaren Spuren und Sektoren widerspiegelt.

Im RAMDISKLC-Listing ist das VTOC in den Programmzeilen 191–310 bzw. (*vor* der Verschiebung in die LC) im Speicher $6200-$62FF enthalten. Die Speicherstelle $6200 entspricht damit dem 0. Byte und die Speicherstelle $62FF dem 255. Bytes des VTOC-Sektors. Die wichtigsten relativen Bytes $00–$FF des VTOC – weitere Details entnehme man dem Listing – wollen wir nunmehr kurz besprechen.

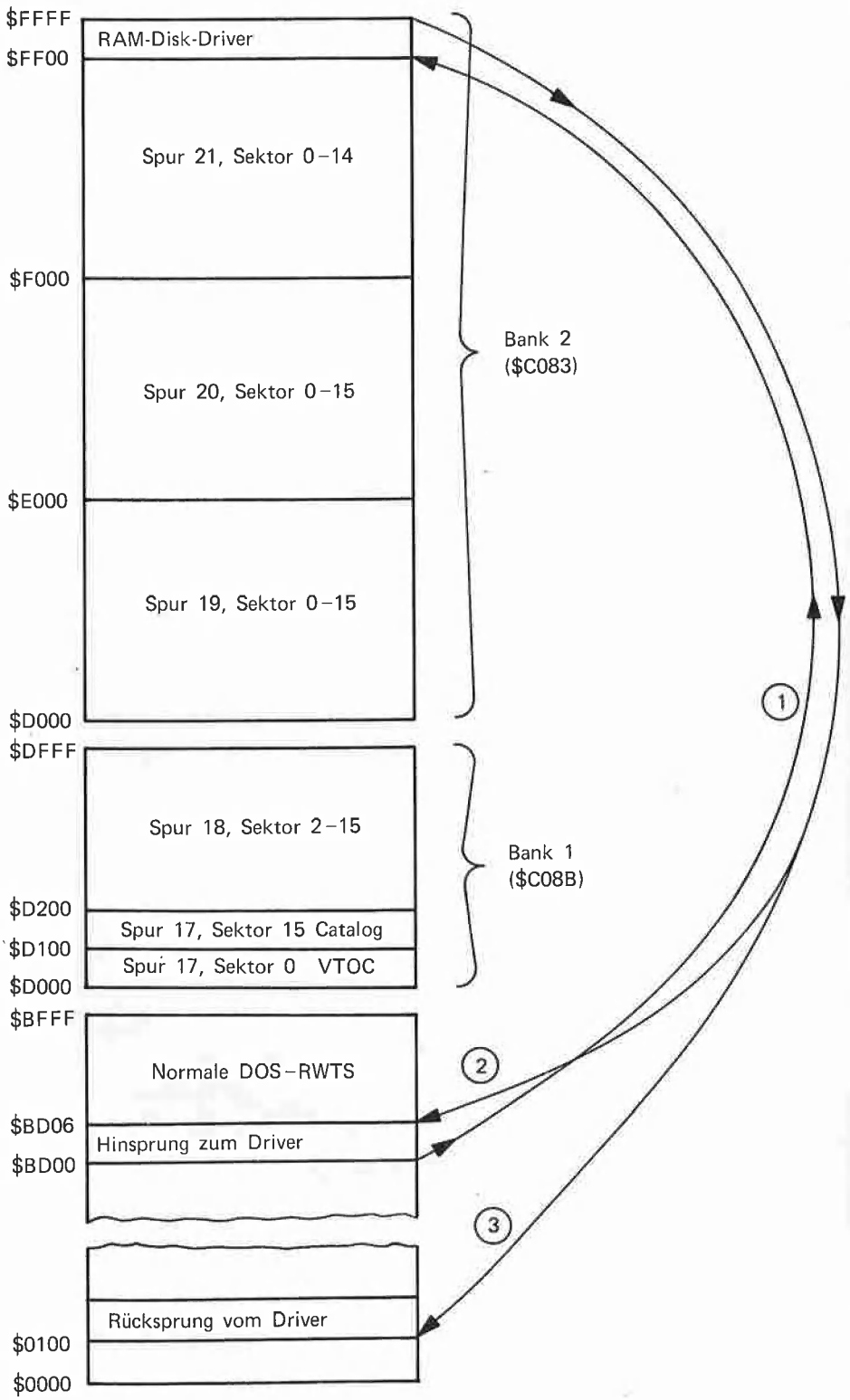
$01: Spur-Nr. des 1. Dateinamensektors (17)
$02: Sektor-Nr. des 1. Dateinamensektors (15)
$03: 3 (= Version DOS 3.3)
$06: Volume-Nr. (254)
$27: maximale Anzahl der Spur-Sektor-Paare innerhalb einer Spur-Sektor-Liste (122)
$30: zuletzt benutzte Spur
$31: Richtung: 1 für ab Spur 18 aufwärts, 255 für ab Spur 16 abwärts
$34: Anzahl der Spuren pro Diskette (35)
$35: Anzahl der Sektoren pro Spur (16)
$36: Anzahl der Bytes pro Sektor (in Low-High: $0100 = 256)

Die eben genannten Bytes von $00–$37 machen den „Kopf" des VTOC aus. Ab Byte $38 folgen je 4 Bytes für je 1 Spur (beginnend mit Spur 0) als Spurenbelegung-Bitmap. Von den je 4 Bytes werden nur jeweils die ersten 2 Bytes zur Verschlüsselung der belegten bzw. freien Sektoren einer Spur benutzt. Beispiel:

F E D C B A 9 8 7 6 5 4 3 2 1 0 Bit-Nrn.
1 1 1 1 1 1 1 1 1 1 1 1 1 1 0 0 Bits selbst

Nehmen wir an, für die Spur 18 seien die 4 Bytes $FF $FC $00 $00 eingetragen. Wir ignorieren die letzten beiden Bytes und erstellen ein Bitmuster der ersten beiden Bytes. Das 0. Bit ($0) steht für Sektor 0 und und das 15. Bit ($F) für Sektor 15. Wenn ein Bit den Wert 1 hat, dann ist der entsprechende Sektor frei, und wenn es den Wert 0 hat, belegt. Das obige Bitmuster für Spur 18 besagt also, daß die Sektoren 0 und 1 bereits belegt sind, während die restlichen Sektoren noch zur freien Verfügung stehen.

Diagramm 1: Speicherverteilung bei RAMDISKLC, „Slot 3"
① Hinsprung zum RAM-Disk-Driver
② Rücksprung, falls Slot <> 3
③ Rücksprung, falls Slot = 3



APPLE II PASCAL

Betriebssystem und Sprache

Erstes deutsches Referenzwerk sämtlicher Befehle und Systemroutinen von Apple II Pascal – mit Addendum einschließlich Version Pascal 1.2!

Gültig für Apple II, II Plus, IIe einschließlich der 128K/80 Zeichen-Konfiguration.

Betriebssystem kommentiert ausführlich und in Deutsch Funktion und Benutzung der fast 60 Systemroutinen des Apple II Pascal Betriebssystems.

Sprache ist das vollständige, deutsche Referenzwerk der „Apple Pascal"-Programmiersprache mit u. a. Informationen über professionelle Pascal-Programmierung, Turtlegraphics, Programmbibliothek etc.

In Vorbereitung: Addendum Pascal 1.2, ein Zusatz zum Buch „Betriebssystem" für 1.2-Benutzer in Deutsch.

„Nach Unterlagen von Apple Deutschland hergestellt"

Apple II Betriebssystem,
272 Seiten, DM 49,–

Apple II Sprache,
216 Seiten, DM 39,–

Pascal 1.2 Addendum, etwa 100 Seiten,
DM 36,– (1. Quartal '85)

te-wi Verlag GmbH
Theo-Prosel-Weg 1
8000 München 40

Weiterführende Literatur...

APPLE II –
Anwenderhandbuch
(L. Poole)
Erst mit Hilfe dieses Leitfadens werden Sie Ihren Apple II erfolgreich einsetzen, denn Text und Bildmaterial gehen weit über das hinaus, was herstellerseitig an Literatur angeboten wird.
416 Seiten, Softcover, DM 56,–

LOGO –
Jeder kann programmieren
(Daniel Watt)
Buch des Jahres in den USA. Für die Computer C64, ATARI, APPLE II, IBM-PC und TI-99.
Hochwertiges Textbuch für Logo-Kurse für zu Hause und im Lehrbereich.
A4, DM 59,–

APPLE II PASCAL –
Eine praktische Anleitung
(A. Luehrmann, H. Peckham)
Unentbehrlich für alle, die die Programmiersprache PASCAL lernen wollen und Zugang zu einem Apple-Computer haben.
544 Seiten, Softcover, DM 59,–

APPLE II – Bewegte 3D-Graphik
(Phil Cohen)
Selbstentworfene Graphiken und Diagramme – animiert oder als Standbilder – eben oder räumlich: alle erforderlichen BASIC-Programme mit Erklärung finden Sie in diesem Buch.
ca. 190 Seiten, Softcover, DM 49,–
(4. Quartal 85)

Computer für Kinder
(Sally Greenwood Larson)
Ein Buch für Kinder, ihre Lehrer und Eltern.
„Computer für Kinder" richtet sich an Kinder im Alter von 8 bis 13 Jahren, für deren Interesse an Computern dieses Buch bewußt geschrieben wurde.
Unterhaltsam und leicht verständlich.
A4 quer, Fadenheftung, DM 29,80

Apple Maschinensprache
Für BASIC-Programmierer der einfachste Zugang zur Muttersprache des Apple. Wesentlich schnellere Maschinenprogramme, direkte Manipulation des Mikroprozessors 6502 im Apple – als Brücke dorthin benötigt dieses Buch nur die drei BASIC-Befehle, POKE, CALL, PEEK. D. Inman/K. Inman. DM 49,–

tm 4153

Noch im Programm:
6502 – Programmieren in Assembler DM 59,–
VisiCalc, 50 Programme auf Diskette, DM 79,–

In Vorbereitung:
Macintosh Programmier-Handbuch DM 59,–

Die Initialisierung der RAM-Disk muß insbesondere für ein dem RAM-Speicher entsprechendes VTOC sorgen. Ferner müssen auch die Dateinamensektoren auf Spur 17 angelegt werden. Bei einer normalen Diskette ist Sektor 15 in Spur 17 der erste und Sektor 1 in Spur 17 der letzte Dateinamensektor, wobei jeder dieser Sektoren 7 Dateieinträge aufnehmen kann. Die Beschreibung der Struktur eines Dateinamensektors würde den Rahmen dieses Aufsatzes sprengen, doch sei darauf hingewiesen, daß die Dateinamensektoren untereinander verkettet sind, d.h. Sektor 15 als 1. Datennamensektor enthält einen Zeiger auf Sektor 14 als 2. Dateinamensektor usw. Der letzte Dateinamensektor (Sektor 1) enthält einen „Null"-Zeiger, womit das Ende der Dateieintragsliste erreicht wird. Die Zeiger in der Form
Spur-Nr.– Sektor-Nr., z.B.
$11 – $0E, d.h. 17 – 14
befinden sich stets in den relativen Bytes $01–$02 eines jeden Dateinamensektors. Im übrigen enthält eine frisch initialisierte Diskette in den Dateinamensektoren zunächst nur „Nullen" ($00). Aus Platzgründen legen wir für unsere Mini-LC-RAM-Disk nur einen einzigen Dateinamensektor (Spur 17, Sektor 15) an. Zu diesem Zweck brauchen wir lediglich den entsprechenden Speicherbereich auf Null zu setzen, womit auch der Zeiger zum theoretisch nächsten Dateinamensektor zum „Null"-Zeiger wird.

Fassen wir zusammen: Zur „Initialisierung" einer RAM-Disk muß der korrekte VTOC-Sektor sowie mindestens 1 Dateinamensektor angelegt werden. Die eigentlichen Datensektoren selbst müssen nicht initialisiert, d.h. auf Null gesetzt werden, obgleich unsere RAMDISKLC-Utility dies zusätzlich tut, damit man beim Experimentieren mit der RAM-Disk sieht, welche Sektoren tatsächlich bereits benutzt wurden.

3. RAM-Disk-Driver

Die Logik eines RAM-Disk-Drivers funktioniert prinzipiell in der Art der aus **Diagramm 1** ersichtlichen eingekreisten Ziffern:

1. Am Anfang des normalen Disk-Drivers (= $BD00) wird ein Sprung zum RAM-Disk-Driver vorgeschaltet, der sich seinerseits irgendwo im Speicher befinden kann. Unser Driver befindet sich selbst in der LC, womit erstens kein Speicherraum in den unteren 48K verlorengeht und zweitens keine Konflikte mit gepatchtem DOS 3.3 auftreten. Wir benötigen bei der Speicherstelle $BD00 nur 2 Assemblerbefehle:
$BD00: BIT $C080
$BD03: JMP $FF00
$BD06: Fortsetzung der RWTS
BIT $C080 macht die Bank 2 der LC lesefähig und JMP $FF00 bewirkt den Sprung zum bei $FF00 beginnenden RAM-Disk-Driver.
Wenn nunmehr die RWTS vom eigenen Assemblerprogramm oder vom File-Manager aufgerufen wird, so erfolgt zunächst ein Sprung via $BD00 zum RAM-Disk-Driver.

2. Dieser wertet den IOB aus und überprüft zunächst, ob der für die RAM-Disk willkürlich festgelegte Slot, bei unserem Driver Slot 3, als Jetzt-Slot vorgegeben ist. Ist dies nicht der Fall, so wird die Language Card wieder abgeschaltet und die normale RWTS bei $BD06 fortgesetzt, wobei selbstverständlich bereits diejenigen Befehle, die ursprünglich bei $BD00–$BD05 standen, vom RAM-Disk-Driver erledigt worden sind.

3. Stimmt jedoch der für die RAM-Disk festgelegte Slot 3 mit dem IOB-Jetzt-Slot überein, so holt sich der RAM-Disk-Driver die erforderlichen Parameter aus der RWTS-IOB und überträgt entweder den DOS-Puffer in den entsprechenden LC-Bereich oder umgekehrt. Danach wird der IOB aktualisiert (insbesondere durch Eintrag von Fehler-Nummer 0) und der Driver, nachdem die LC wieder abgestellt worden ist, über RTS verlassen.

Da das Abstellen der LC nicht auf der LC selbst geschehen kann, wird eine 6 Bytes umfassende Mini-Routine in den Stackbereich $0100-$0105 kopiert und von dort aus die LC desaktiviert.
Insgesamt gibt es 3 Typen des Rücksprungs vom RAM-Disk-Driver in die unteren 48K:
a) Sprung über den Stack nach $BD06, wenn die RAM-Disk laut IOB nicht verlangt war.
b) Sprung über den Stack auf ein RTS bei $0103 mit gesetzten Carry-Flag und I/O-Error in IOB, falls die RAM-Disk illegal angesprochen wurde (z.B. INIT XXX, S3 oder Lesen einer Spur, die laut VTOC gar nicht existiert).
c) Sprung über den Stack auf ein RTS bei $0103 mit zurückgesetztem Carry-Flag und Error = 0, falls alle Parameter der IOB gültig waren.

Hinweis: Auf die LC-Softswitches sowie auf Memory-Management-Probleme schlechthin wird in einem späteren Peeker-Artikel eingegangen.

4. Technische Anmerkungen

a) Das Programm RAMDISKLC wird mit BRUN RAMDISKLC gestartet. Danach erscheint ein Kurzmenü mit der Frage „RAMDISKLC mit Init J/N", worauf Sie mit „J" antworten.
b) Die LC-RAM-Disk wird mit Slot 3, Drive 1 oder Slot 3, Drive 2 angesprochen.
c) Der 16K-RAM-Disk stehen 61 reine Datensektoren zur Verfügung, weil die 3 restlichen 256-Byte-Blöcke durch VTOC, Dateinamensektor und den Driver selbst benutzt werden.
d) Der eine und einzige Dateinamensektor kann 7 Dateieinträge aufnehmen. Das Kopieren von Dateien auf die und von der RAM-Disk mit dem FID-Programm der System Master Diskette ist problemlos möglich. Doch sei darauf hingewiesen, daß FID einen Bug hat, der nicht an unserem RAMDISKLC liegt. Dieser Bug läßt FID stets dann „durchdrehen", wenn die Catalog-Spur voll ist, was bei uns nach 7 Dateieinträgen bereits der Fall ist. Derselbe Bug würde auch auftreten, wenn man mehr als 105 (14 · 7) Dateieinträge auf einer normalen Diskette mit FID erzeugen wollte. Dann können Sie sich nur noch mit Reset retten!
e) Wenn aus Versehen neu gebootet worden ist, so starte man RAMDISKLC und antworte auf die Frage „Init J/N" mit „N". Durch Neubooten wird der Inhalt der LC nicht zerstört, doch wird durch DOS 3.3 in die Speicherstelle $E000 der Wert $00 gepokt, der eine in diesem Bereich liegende RAM-Disk-Datei modifizieren würde. Bei einer nach dem Patch
$BFD3: EA EA EA
initialisierten DOS-Diskette findet dieser Poke nicht mehr statt.
Im übrigen starten Sie RAMDISKLC nur dann ein zweites Mal, wenn Sie zuvor neu gebootet haben.
f) RAMDISKLC benutzt in den unteren 48K nur je 6 Bytes ab $BD00 sowie ab $0100.
g) Der Driver ist relativ gut gegen illegale Parameter abgesichert (besser als DOS 3.3 selbst), so daß man RAMDISKLC zum Üben der RWTS benutzten kann. Beispielsweise ist es nicht möglich, einen Binärfile in die Karte zu BLOADen oder eine Spur zu lesen, die auf Grund der VTOC gar nicht existiert.

Die Mikrocomputer-Zeitschrift, die ihre Leser jeden Monat weiterbringt.

Ob Sie die Mikrocomputerei als Freizeitvergnügen oder aus beruflicher Notwendigkeit betreiben: Mit mc dringen Sie jeden Monat tiefer ein in die Mikrocomputertechnik. Und einsteigen können Sie jederzeit...

mc testet Hardware und prüft Programme...

mc informiert Sie umfassend über Computer und Peripherie, über Programmiersprachen und Betriebssysteme...

mc liefert Software-Ideen und regt an zum Selberbauen. Sie finden Applikationen vom einfachen Interface bis zum kompletten Selbstbausystem...

mc gibt Ihnen viel, wenn Sie mc mit ein wenig technischem Verständnis entgegenkommen...

mc macht ihre Leser Schritt für Schritt zu Mikrocomputer-Profis. Interessiert? Dann machen Sie doch einfach von unserem nebenstehenden Kennenlern-Angebot Gebrauch...

Die Mikrocomputer-Zeitschrift

Kennenlern-Angebot

Ich möchte die mc unverbindlich kennenlernen. Schicken Sie mir die beiden neuesten Ausgaben kostenlos. Informiere ich Sie danach nicht anders, abonniere ich die mc ab _____ zum Jahrespreis von DM 66,– (im Ausland DM 72,–) inkl. Porto.

Name _____

Beruf _____

Straße _____

PLZ/Ort _____

Datum/Unterschrift _____

Die Kündigung ist jeweils 8 Wochen vor Ablauf des Abonnements möglich. Wichtig: Nach Erhalt des 2. kostenlosen Heftes kann ich innerhalb von 10 Tagen durch einfache schriftliche Mitteilung an den Verlag von einem Abonnement Abstand nehmen.

Bitte hier Ihre zweite Unterschrift _____ PE1

Bitte ausschneiden und einsenden an:

Franzis' Franzis Verlag, Postfach 37 01 20, 8000 München 37

```
                        1             ORG  $611D
                        3     *
                        4     * RAMDISKLC
                        5     * ═══════
                        6     *
                        7     * von U.Stiehl/01.12.84
                        8     *
                        9     * RAM-Diskdriver für
                        10    * 16K-Language-Card
                        11    * Apple II+/IIe/IIc
                        12    *
                        13    *  61 Datensektoren
                        14    *   1 VTOC-Sektor (Catalog)
                        15    *   1 Dateinamen-Sektor (Catalog)
                        16    *  ──────────────────────────
                        17    *  63 Sektoren insgesamt
                        18    *
                        19    * Speicherverteilung
                        20    * ──────────────────
                        21    *
                        22    * $D000-$D0FF Bk1 T.11,S.00 VTOC
                        23    * $D100-$D1FF Bk1 T.11,S.0F CAT
                        24    * $D200-$DFFF Bk1 T.12,S.02-S.0F
                        25    * $D000-$DFFF Bk2 T.13,S.00-S.0F
                        26    * $E000-$EFFF Bk2 T.14,S.00-S.0F
                        27    * $F000-$FEFF Bk2 T.15.S.00-S.0E
                        28    * $FF00-$FFFF RAM-Diskdriver
                        29    *
                        30    PTR      EQU  $CE
                        31    IOB      EQU  $48
                        32    SECTOR   EQU  $FC          ;Temp!
                        33    RAM      EQU  $FC          ;RAM-Disk
                        34    BUF      EQU  $FE          ;DOS-Puffer
                        35    STACK    EQU  $0100        ;Rücksprung
                        36    DOSWARM  EQU  $03D0
                        37    PRINT    EQU  $FDED
                        38    RDKEY    EQU  $FD0C
                        39    HOME     EQU  $FC58
                        40    *
611D: A2 00             41            LDX  #0
611F: BD 49 61          42    MITINIT? LDA  MENUE,X
6122: F0 06             43            BEQ  JANEIN
6124: 20 ED FD          44            JSR  PRINT
6127: E8                45            INX
6128: D0 F5             46            BNE  MITINIT?
612A: 20 0C FD          47    JANEIN   JSR  RDKEY
612D: C9 CE             48            CMP  #"N"
612F: F0 11             49            BEQ  NURSOFT
6131: C9 EE             50            CMP  #"n"
6133: F0 0D             51            BEQ  NURSOFT
6135: C9 CA             52            CMP  #"J"
6137: F0 06             53            BEQ  MITINIT
6139: C9 EA             54            CMP  #"j"
613B: F0 02             55            BEQ  MITINIT
613D: D0 EB             56            BNE  JANEIN
                        57    *
613F: 20 BB 61          58    MITINIT  JSR  INITLC
6142: 20 6E 61          59    NURSOFT  JSR  SOFTLC
                        60    *
6145: EA                61    INITENDE HEX  EA          ;60
                        62    *
6146: 4C D0 03          63            JMP  DOSWARM
                        64    *
6149: 8D                65    MENUE    HEX  8D
614A: D2 C1 CD          66            ASC  "RAMDISKLC mit Init"
615C: A0 CA AF          67            ASC  " J/N "
6161: 00                68            HEX  00
                        69    *
                        70    *═══════════════════════════
                        71    *
                        72    * SOFTLC
                        73    * ══════
                        74    *
                        75    * DOS-Patch bei RWTS-Entry
                        76    *
                        77    * $BD00: 84 48    STY $48
                        78    * $BD02: 85 49    STA $49
                        79    * $BD04: A0 02    LDY #$02
                        80    *
                        81    * $BD00: 2C 80 C0 BIT $C080 RDBK2
                        82    * $BD03: 4C 00 FF JMP $FF00
                        83    *
6162: 84 48 85          84    BD00ALT  HEX  84488549A002
6168: 2C 80 C0          85    BD00NEU  HEX  2C80C04C00FF
                        86    *
616E: A2 05             87    SOFTLC   LDX  #5
6170: BD 00 BD          88    DOSOKAY  LDA  $BD00,X
6173: DD 62 61          89             CMP  BD00ALT,X
6176: D0 20             90             BNE  NOTDOS
6178: CA                91             DEX
6179: 10 F5             92             BPL  DOSOKAY
                        93    *
617B: A2 05             94             LDX  #5
617D: BD 68 61          95    PATCH1   LDA  BD00NEU,X
6180: 9D 00 BD          96             STA  $BD00,X
6183: CA                97             DEX
6184: 10 F7             98             BPL  PATCH1
                        99    *
                        100   * Verschiebe "Driver" nach $FF00
                        101   *
6186: AD 81 C0          102            LDA  $C081       ;RDROM
6189: AD 81 C0          103            LDA  $C081       ;WRBK2
618C: A2 00             104            LDX  #$00
618E: BD 00 63          105   MOVER1   LDA  $6300,X
6191: 9D 00 FF          106            STA  $FF00,X
6194: E8                107            INX
6195: D0 F7             108            BNE  MOVER1
6197: 60                109            RTS              ;Exit
                        110   *
                        111   ********************************
                        112   *
6198: A2 00             113   NOTDOS   LDX  #$00
619A: BD A8 61          114   NOTDOS1  LDA  NOTDOS3,X
619D: F0 06             115            BEQ  NOTDOS2
619F: 20 ED FD          116            JSR  PRINT
61A2: E8                117            INX
61A3: D0 F5             118            BNE  NOTDOS1
61A5: 4C 45 61          119   NOTDOS2  JMP  INITENDE
61A8: 8D                120   NOTDOS3  HEX  8D
61A9: C4 CF D3          121            ASC  "DOS modifiziert"
61B8: 87 8D 00          122            HEX  878D00
                        123   *
                        124   *═══════════════════════════
                        125   *
                        126   * INITLC
                        127   * ══════
                        128   *
                        129   * $D000-$FFFF auf 0 setzen
                        130   *
                        131   * Bereich $D000-FFFF Bk2 löschen
                        132   *
61BB: AC 83 C0          133   INITLC   LDY  $C083       ;RDBK2
61BE: AC 83 C0          134            LDY  $C083       ;WRBK2
61C1: A9 D0             135            LDA  #$D0        ;$D000
61C3: 85 CF             136            STA  PTR+1
61C5: A9 00             137            LDA  #0
61C7: 85 CE             138            STA  PTR
61C9: A8                139            TAY
61CA: 91 CE             140   L1       STA  (PTR),Y
61CC: C8                141            INY
61CD: D0 FB             142            BNE  L1
61CF: E6 CF             143            INC  PTR+1
61D1: D0 F7             144            BNE  L1          ;$FFFF+1
                        145   *
                        146   * Bereich $D100-DFFF Bk1 löschen
                        147   *
61D3: AC 8B C0          148            LDY  $C08B       ;RDBK1
61D6: AC 8B C0          149            LDY  $C08B       ;WRBK1
61D9: A0 D1             150            LDY  #$D1        ;$D100
61DB: 84 CF             151            STY  PTR+1
61DD: A8                152            TAY
61DE: 91 CE             153   L2       STA  (PTR),Y
61E0: C8                154            INY
61E1: D0 FB             155            BNE  L2
61E3: E6 CF             156            INC  PTR+1
61E5: A6 CF             157            LDX  PTR+1
61E7: E0 E0             158            CPX  #$E0        ;$E000
61E9: D0 F3             159            BNE  L2
                        160   *
                        161   ********************************
                        162   *
                        163   * Catalog-Spur anlegen
                        164   *
                        165   * Track $11, Sektor $00 = VTOC
                        166   * nach $D000-$D0FF von Bank1
                        167   * verschieben.
                        168   *
61EB: A2 00             169            LDX  #0
61ED: BD 00 62          170   VTOC1    LDA  VTOC2,X
61F0: 9D 00 D0          171            STA  $D000,X
61F3: E8                172            INX
61F4: D0 F7             173            BNE  VTOC1
                        174   *
                        175   * Kein Link-Byte anlegen, da nur
```

```
                        176  * 1 Sektor für Dateinamen reser-
                        177  * viert ist = 7 Dateien max.
                        178  *
61F6: AD 81 C0          179          LDA  $C081       ;RDROM
61F9: AD 81 C0          180          LDA  $C081       ;WRBK2
61FC: 20 58 FC          181          JSR  HOME
61FF: 60                182          RTS              ;Exit
                        183  *
                        184  ********************************
                        185  *
                        186  * Bank 1: $D000-$DFFF
                        187  * ===================
                        188  *
                        189  * T.$11, S.$00 = VTOC = $D000-$DFFF
                        190  *
6200: 00                191  VTOC2    HEX  00          ;unused
                        192  *
6201: 11                193          HEX  11          ;T. $11
6202: 0F                194          HEX  0F          ;S. $0F
                        195  *
6203: 03                196          HEX  03          ;DOS 3.3
6204: 00 00             197          HEX  0000        ;unused
6206: FE                198          HEX  FE          ;Vol.254
                        199  *
6207: 00 00 00          200          HEX  0000000000 ;unused
620C: 00 00 00          201          HEX  0000000000
6211: 00 00 00          202          HEX  0000000000
6216: 00 00 00          203          HEX  0000000000
621B: 00 00 00          204          HEX  0000000000
6220: 00 00 00          205          HEX  0000000000
6225: 00 00             206          HEX  0000
                        207  *
                        208  * Anzahl der T/S-Paare pro TSL
                        209  *
6227: 7A                210          HEX  7A          ;122max.
6228: 00 00 00          211          HEX  00000000    ;unused
622C: 00 00 00          212          HEX  00000000
6230: 11                213          HEX  11          ;letzter T.
6231: 01                214          HEX  01          ;Richtung
6232: 00 00             215          HEX  0000        ;unused
6234: 23                216          HEX  23          ;35 Trk.
6235: 10                217          HEX  10          ;16 Sek.
6236: 00 01             218          HEX  0001        ;$0100
                        219  *
                        220  * Spurenbelegung-Bitmap
                        221  * =====================
                        222  *
                        223  * Bit 1 = frei, Bit 0 = belegt
                        224  *
                        225  * Nichtbenutzte Spuren: $00-$10
                        226  *
                        227  * FEDCBA9876543210
                        228  * 0000000000000000 belegt
                        229  *
6238: 00 00 00          230          HEX  00000000    ;T.00
623C: 00 00 00          231          HEX  00000000    ;T.01
6240: 00 00 00          232          HEX  00000000    ;T.02
6244: 00 00 00          233          HEX  00000000    ;T.03
6248: 00 00 00          234          HEX  00000000    ;T.04
624C: 00 00 00          235          HEX  00000000    ;T.05
6250: 00 00 00          236          HEX  00000000    ;T.06
6254: 00 00 00          237          HEX  00000000    ;T.07
6258: 00 00 00          238          HEX  00000000    ;T.08
625C: 00 00 00          239          HEX  00000000    ;T.09
6260: 00 00 00          240          HEX  00000000    ;T.0A
6264: 00 00 00          241          HEX  00000000    ;T.0B
6268: 00 00 00          242          HEX  00000000    ;T.0C
626C: 00 00 00          243          HEX  00000000    ;T.0D
6270: 00 00 00          244          HEX  00000000    ;T.0E
6274: 00 00 00          245          HEX  00000000    ;T.0F
6278: 00 00 00          246          HEX  00000000    ;T.10
                        247  *
                        248  * Catalog: Es gibt nur 1 Datei-
                        249  * namen-Sektor $0F sowie 1 VTOC-
                        250  * Sektor $00. Die Bitmap für die
                        251  * Catalog-Spur gilt als belegt.
                        252  *
627C: 00 00 00          253  T11      HEX  00000000    ;T.11
                        254  *
                        255  * T.$12 = $D200-$DFFF = S.$02-$0F
                        256  *
                        257  * FEDCBA9876543210
                        258  * 1111111111111100
                        259  *
6280: FF FC 00          260  T12      HEX  FFFC0000    ;T.12
                        261  *
```

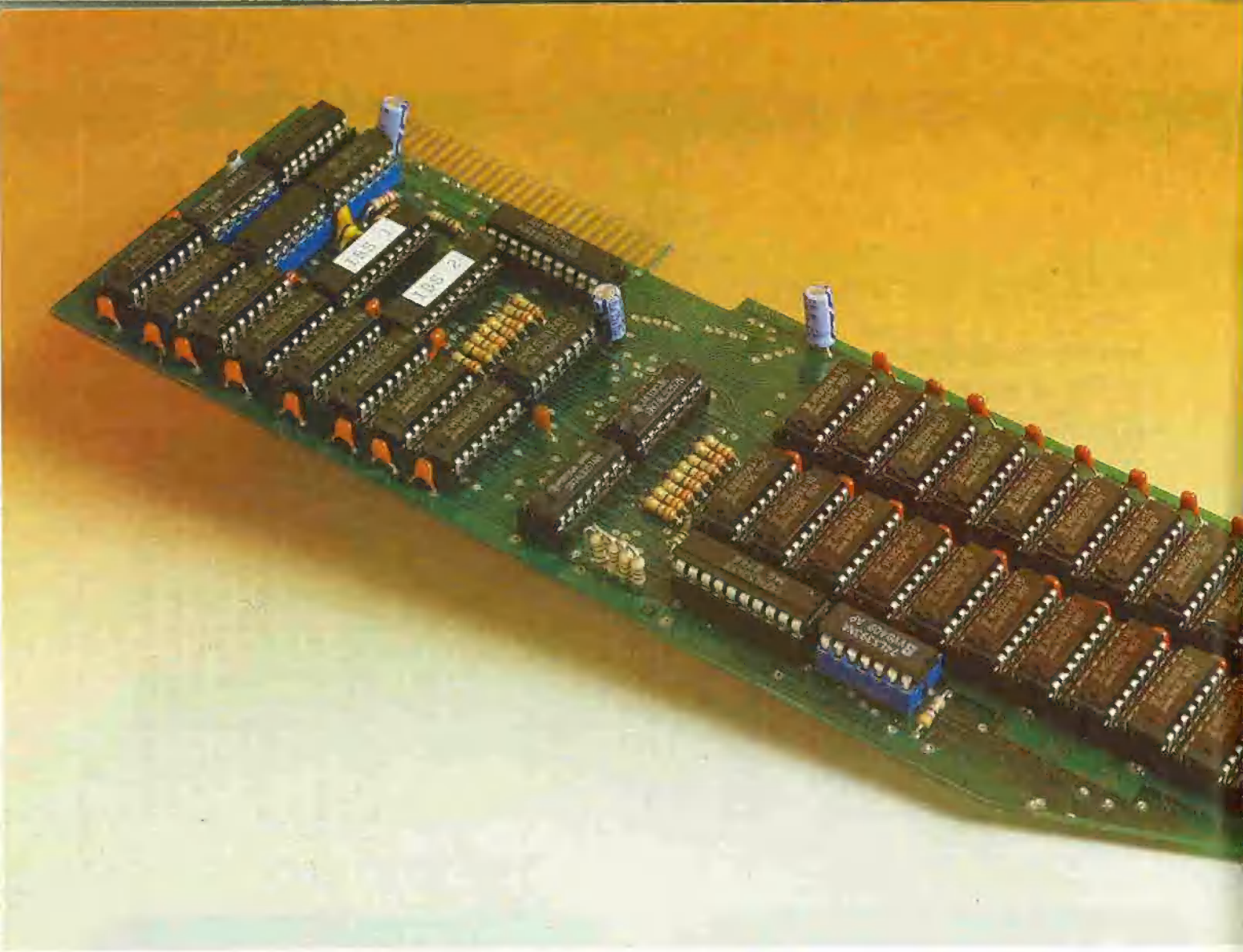
```
                        262  * Bank 2: $D000-$FFFF
                        263  * ===================
                        264  *
                        265  * T.$13 = $D000-$DFFF
                        266  * T.$14 = $E000-$EFFF
                        267  *
6284: FF FF 00          268          HEX  FFFF0000    ;T.13
6288: FF FF 00          269          HEX  FFFF0000    ;T.14
                        270  *
                        271  * T.$15 = $F000-$FEFF = S.$00-$0E
                        272  *
                        273  * FEDCBA9876543210
                        274  * 0111111111111111
                        275  *
628C: 7F FF 00          276          HEX  7FFF0000    ;T.15
                        277  *
                        278  * Nichtbenutzte Spuren: $10-$22
                        279  *
6290: 00 00 00          280          HEX  00000000    ;T.16
6294: 00 00 00          281          HEX  00000000    ;T.17
6298: 00 00 00          282          HEX  00000000    ;T.18
629C: 00 00 00          283          HEX  00000000    ;T.19
62A0: 00 00 00          284          HEX  00000000    ;T.1A
62A4: 00 00 00          285          HEX  00000000    ;T.1B
62A8: 00 00 00          286          HEX  00000000    ;T.1C
62AC: 00 00 00          287          HEX  00000000    ;T.1D
62B0: 00 00 00          288          HEX  00000000    ;T.1E
62B4: 00 00 00          289          HEX  00000000    ;T.1F
62B8: 00 00 00          290          HEX  00000000    ;T.20
62BC: 00 00 00          291          HEX  00000000    ;T.21
62C0: 00 00 00          292          HEX  00000000    ;T.22
                        293  *
                        294  * Rest stets frei
                        295  *
62C4: 00 00 00          296          HEX  00000000
62C8: 00 00 00          297          HEX  00000000
62CC: 00 00 00          298          HEX  00000000
62D0: 00 00 00          299          HEX  00000000
62D4: 00 00 00          300          HEX  00000000
62D8: 00 00 00          301          HEX  00000000
62DC: 00 00 00          302          HEX  00000000
62E0: 00 00 00          303          HEX  00000000
62E4: 00 00 00          304          HEX  00000000
62E8: 00 00 00          305          HEX  00000000
62EC: 00 00 00          306          HEX  00000000
62F0: 00 00 00          307          HEX  00000000
62F4: 00 00 00          308          HEX  00000000
62F8: 00 00 00          309          HEX  00000000
62FC: 00 00 00          310          HEX  00000000
                        311  *
                        312  * Muß bei $62FF enden!
                        313  *=========================
                        314  *
                        315  * Eigentlicher Driver
                        316  *
                        317          ORG  $FF00       ;$6300!
                        318  *
FF00: 4C 0A FF          319  DRIVER1  JMP  DRIVER2     ;0000
                        320  *
FF03: 30                321  SLOT     HEX  30          ;Slot 3
FF04: 00                322  TRACK    HEX  00
FF05: 00                323  RDWRFLAG HEX  00
FF06: 00 00 00          324  ZERO     HEX  00000000
                        325  *
                        326  * Rette IOB-Pointer
                        327  *
FF0A: 84 48             328  DRIVER2  STY  IOB
FF0C: 85 49             329          STA  IOB+1
                        330  *
                        331  * Zunächst Bank 1 Read/Write
                        332  *
FF0E: AD 8B C0          333          LDA  $C08B       ;RDBK1
FF11: AD 8B C0          334          LDA  $C08B       ;WRBK1
                        335  *
                        336  * Rette Zero-Page $00FC-$00FF
                        337  *
FF14: A2 03             338          LDX  #3
FF16: B5 FC             339  ZEROSAVE LDA  RAM,X
FF18: 9D 06 FF          340          STA  ZERO,X
FF1B: CA                341          DEX
FF1C: 10 F8             342          BPL  ZEROSAVE
                        343  *
                        344  * Pseudo-Disk-Slot?
                        345  *
FF1E: A0 01             346  SLOT?    LDY  #$01
FF20: B1 48             347          LDA  (IOB),Y     ;Slot
FF22: CD 03 FF          348          CMP  SLOT
```

```
FF25: F0 0D     350           BEQ  RDWR?
                351  *
                352  * Zurück zur normalen Rwts
                353  *
FF27: 4C C8 FF  354  RWTS     JMP  EXIT1      ;Rwts
                355  *
                356  * Drive-Error
                357  *
FF2A: A0 0D     358  IOERROR  LDY  #$0D
FF2C: A9 40     359           LDA  #$40       ;IO-Err.
FF2E: 91 48     360           STA  (IOB),Y
FF30: 38        361           SEC             ;Carry!
FF31: 4C CE FF  362           JMP  EXIT2      ;IO-Err.
                363  *
                364  * Befehl: 1=Read oder 2=Write?
                365  *
FF34: A0 0C     366  RDWR?    LDY  #$0C
FF36: B1 48     367           LDA  (IOB),Y    ;Befehl
FF38: F0 F0     368           BEQ  IOERROR    ;Seek
FF3A: C9 03     369           CMP  #3         ;>3?
FF3C: B0 EC     370           BCS  IOERROR
                371  *
                372  * DOS-Pufferadresse
                373  *
FF3E: 8D 05 FF  374  PUFFER   STA  RDWRFLAG
FF41: A0 08     375           LDY  #$08
FF43: B1 48     376           LDA  (IOB),Y
FF45: 85 FE     377           STA  BUF
FF47: C8        378           INY
FF48: B1 48     379           LDA  (IOB),Y
FF4A: 85 FF     380           STA  BUF+1
                381  *
                382  * Track und Sektor in Bereich
                383  * der LC-Karte umwandeln
                384  *
                385  * T.$11 S.$00 -> $D000-$DFFF Bk1
                386  * T.$11 S.$0F -> $D100-$D1FF Bk1
                387  * T.$12 02-0F -> $D200-$DFFF Bk1
                388  * T.$13 00-0F -> $D000-$DFFF Bk2
                389  * T.$14 00-0F -> $E000-$EFFF Bk2
                390  * T.$15 00-0E -> $F000-$FEFF Bk2
                391  *
FF4C: A0 04     392           LDY  #$04
FF4E: B1 48     393           LDA  (IOB),Y    ;Track
FF50: 8D 04 FF  394           STA  TRACK
FF53: C8        395           INY
FF54: B1 48     396           LDA  (IOB),Y    ;Sector
FF56: 85 FC     397           STA  SECTOR
FF58: AA        398           TAX             ;X=Sector
                399  *
FF59: AD 04 FF  400  T11?     LDA  TRACK
FF5C: C9 11     401           CMP  #$11
FF5E: D0 0C     402           BNE  T12?
FF60: E0 00     403           CPX  #$00       ;S.00/0F?
FF62: F0 22     404           BEQ  D000BK1
FF64: E0 0F     405           CPX  #$0F
FF66: D0 C2     406           BNE  IOERROR
FF68: A9 D1     407           LDA  #$D1       ;$D100
FF6A: D0 1F     408           BNE  KEINADD
FF6C: C9 12     409  T12?     CMP  #$12
FF6E: D0 04     410           BNE  T13?
FF70: E0 02     411           CPX  #$02       ;S.02-0F?
FF72: B0 12     412           BCS  D000BK1
FF74: C9 13     413  T13?     CMP  #$13
FF76: F0 17     414           BEQ  D000BK2
FF78: C9 14     415  T14?     CMP  #$14
FF7A: F0 17     416           BEQ  E000BK2
FF7C: C9 15     417  T15?     CMP  #$15
FF7E: D0 AA     418           BNE  IOERROR
FF80: E0 0F     419           CPX  #$0F
FF82: D0 13     420           BNE  F000BK2
FF84: F0 A4     421           BEQ  IOERROR    ;$FF00!
                422  *
FF86: A9 D0     423  D000BK1  LDA  #$D0       ;$D000
FF88: 18        424  ADDSEC1  CLC
FF89: 65 FC     425           ADC  SECTOR
FF8B: 85 FD     426  KEINADD  STA  RAM+1
FF8D: D0 15     427           BNE  READ?      ;stets
FF8F: A9 D0     428  D000BK2  LDA  #$D0       ;$D000
FF91: D0 06     429           BNE  ADDSEC2
FF93: A9 E0     430  E000BK2  LDA  #$E0       ;$E000
FF95: D0 02     431           BNE  ADDSEC2
FF97: A9 F0     432  F000BK2  LDA  #$F0       ;$F000
FF99: 18        433  ADDSEC2  CLC
FF9A: 65 FC     434           ADC  SECTOR
FF9C: 85 FD     435           STA  RAM+1
FF9E: AD 83 C0  436           LDA  $C083      ;RDBK2
FFA1: AD 83 C0  437           LDA  $C083      ;WRBK2
                438  *
FFA4: A0 00     439  READ?    LDY  #0         ;Y=0
FFA6: 84 FC     440           STY  RAM        ;Low=0
FFA8: AD 05 FF  441           LDA  RDWRFLAG
FFAB: C9 01     442           CMP  #1         ;1=Read
FFAD: D0 09     443           BNE  WRITER
                444  *
                445  * Read = Von der LC nach "unten"
                446  *
FFAF: B1 FC     447  READER   LDA  (RAM),Y
FFB1: 91 FE     448           STA  (BUF),Y
FFB3: C8        449           INY
FFB4: D0 F9     450           BNE  READER
FFB6: F0 07     451           BEQ  ZURUECK1
                452  *
                453  * Write = Von "unten" zur LC
                454  *
FFB8: B1 FE     455  WRITER   LDA  (BUF),Y
FFBA: 91 FC     456           STA  (RAM),Y
FFBC: C8        457           INY
FFBD: D0 F9     458           BNE  WRITER
                459  *
FFBF: 18        460  ZURUECK1 CLC             ;okay!
FFC0: A0 0D     461           LDY  #$0D
FFC2: A9 00     462           LDA  #$00
FFC4: 91 48     463           STA  (IOB),Y    ;o.Fehler
FFC6: 90 06     464           BCC  EXIT2
                465  *
                466  ********************************
                467  *
                468  * EXIT1: Fortsetzung der normalen
                469  *        Rwts bei $BD06
                470  *
FFC8: 20 D9 FF  471  EXIT1    JSR  ZEROLOAD   ;Rwts
FFCB: 4C 00 01  472           JMP  STACK
                473  *
                474  * EXIT2: Entweder IO-Error mit BCS
                475  *        oder normaler Exit von
                476  *        RAM-Diskdriver mit BCC
                477  *
FFCE: 20 D9 FF  478  EXIT2    JSR  ZEROLOAD   ;IO-Error
FFD1: A9 60     479           LDA  #$60       ;RTS
FFD3: 8D 03 01  480           STA  STACK+3
FFD6: 4C 00 01  481           JMP  STACK
                482  *
                483  * Restore Zero-Page
                484  *
FFD9: A2 03     485  ZEROLOAD LDX  #3
FFDB: BD 06 FF  486  ZERO1    LDA  ZERO,X
FFDE: 95 FC     487           STA  RAM,X
FFE0: CA        488           DEX
FFE1: 10 F8     489           BPL  ZERO1
                490  *
                491  * RDROM auf Stack schieben
                492  *
                493  * $0100: 2C 82 C0  BIT $C082
                494  * $0103: 4C 06 BD  JMP $BD06
                495  *
FFE3: A2 05     496           LDX  #5
FFE5: BD F1 FF  497  STACKMOV LDA  STACKPRO,X
FFE8: 9D 00 01  498           STA  STACK,X
FFEB: CA        499           DEX
FFEC: 10 F7     500           BPL  STACKMOV
                501  *
                502  * LDY #$02 bei $BD04 vornehmen
                503  *
FFEE: A0 02     504           LDY  #$02
FFF0: 60        505           RTS
                506  *
FFF1: 2C 82 C0  507  STACKPRO HEX  2C82C0
FFF4: 4C 06 BD  508           HEX  4C06BD
                509  *
FFF7: AD 82 C0  510  RESET    LDA  $C082      ;RDROM
FFFA: F7 FF     511           DA   RESET
FFFC: F7 FF     512           DA   RESET
FFFE: F7 FF     513           DA   RESET

739 bytes
```

```
RAMDISKLC

BSAVE RAMDISKLC, A24861, L739

$6118: 00 00 00 00 00 A2 00 BD
$6120: 49 61 F0 06 20 ED FD E8
$6128: D0 F5 20 0C FD C9 CE F0
$6130: 11 C9 EE F0 0D C9 CA F0
$6138: 06 C9 EA F0 02 D0 EB 20
$6140: BB 61 20 6E 61 EA 4C D0
$6148: 03 8D D2 C1 CD C4 C9 D3
$6150: CB CC C3 A0 ED E9 F4 A0
$6158: C9 EE E9 F4 A0 CA AF CE
$6160: A0 00 84 48 85 49 A0 02
$6168: 2C 80 C0 4C 00 FF A2 05
$6170: BD 00 BD DD 62 61 D0 20
$6178: CA 10 F5 A2 05 BD 68 61
$6180: 9D 00 BD CA 10 F7 AD 81
$6188: C0 AD 81 C0 A2 00 BD 00
$6190: 63 9D 00 FF E8 D0 F7 60
$6198: A2 00 BD A8 61 F0 06 20
$61A0: ED FD E8 D0 F5 4C 45 61
$61A8: 8D C4 CF D3 A0 ED EF E4
$61B0: E9 E6 E9 FA E9 E5 F2 F4
$61B8: 87 8D 00 AC 83 C0 AC 83
$61C0: C0 A9 D0 85 CF A9 00 85
$61C8: CE A8 91 CE C8 D0 FB E6
$61D0: CF D0 F7 AC 8B C0 AC 8B
$61D8: C0 A0 D1 84 CF A8 91 CE
$61E0: C8 D0 FB E6 CF A6 CF E0
$61E8: E0 D0 F3 A2 00 BD 00 62
$61F0: 9D 00 D0 E8 D0 F7 AD 81
$61F8: C0 AD 81 C0 20 58 FC 60
```

```
RAM-Disk-VTOC
$6200: 00 11 0F 03 00 00 FE 00
$6208: 00 00 00 00 00 00 00 00
$6210: 00 00 00 00 00 00 00 00
$6218: 00 00 00 00 00 00 00 00
$6220: 00 00 00 00 00 00 00 7A
$6228: 00 00 00 00 00 00 00 00
$6230: 11 01 00 00 23 10 00 01
$6238: 00 00 00 00 00 00 00 00
$6240: 00 00 00 00 00 00 00 00
$6248: 00 00 00 00 00 00 00 00
$6250: 00 00 00 00 00 00 00 00
$6258: 00 00 00 00 00 00 00 00
$6260: 00 00 00 00 00 00 00 00
$6268: 00 00 00 00 00 00 00 00
$6270: 00 00 00 00 00 00 00 00
$6278: 00 00 00 00 00 00 00 00
$6280: FF FC 00 00 FF FF 00 00
$6288: FF FF 00 00 7F FF 00 00
$6290: 00 00 00 00 00 00 00 00
$6298: 00 00 00 00 00 00 00 00
$62A0: 00 00 00 00 00 00 00 00
$62A8: 00 00 00 00 00 00 00 00
$62B0: 00 00 00 00 00 00 00 00
$62B8: 00 00 00 00 00 00 00 00
$62C0: 00 00 00 00 00 00 00 00
$62C8: 00 00 00 00 00 00 00 00
$62D0: 00 00 00 00 00 00 00 00
$62D8: 00 00 00 00 00 00 00 00
$62E0: 00 00 00 00 00 00 00 00
$62E8: 00 00 00 00 00 00 00 00
$62F0: 00 00 00 00 00 00 00 00
$62F8: 00 00 00 00 00 00 00 00
```

```
Eigentlicher Driver
$6300: 4C 0A FF 30 00 00 00 00
$6308: 00 00 84 48 85 49 AD 8B
$6310: C0 AD 8B C0 A2 03 B5 FC
$6318: 9D 06 FF CA 10 F8 A0 01
$6320: B1 48 CD 03 FF F0 0D 4C
$6328: C8 FF A0 0D A9 40 91 48
$6330: 38 4C CE FF A0 0C B1 48
$6338: F0 F0 C9 03 B0 EC 8D 05
$6340: FF A0 08 B1 48 85 FE C8
$6348: B1 48 85 FF A0 04 B1 48
$6350: 8D 04 FF C8 B1 48 85 FC
$6358: AA AD 04 FF C9 11 D0 0C
$6360: E0 00 F0 22 E0 0F D0 C2
$6368: A9 D1 D0 1F C9 12 D0 04
$6370: E0 02 B0 12 C9 13 F0 17
$6378: C9 14 F0 17 C9 15 D0 AA
$6380: E0 0F D0 13 F0 A4 A9 D0
$6388: 18 65 FC 85 FD D0 15 A9
$6390: D0 D0 06 A9 E0 D0 02 A9
$6398: F0 18 65 FC 85 FD AD 83
$63A0: C0 AD 83 C0 A0 00 84 FC
$63A8: AD 05 FF C9 01 D0 09 B1
$63B0: FC 91 FE C8 D0 F9 F0 07
$63B8: B1 FE 91 FC C8 D0 F9 18
$63C0: A0 0D A9 00 91 48 90 06
$63C8: 20 D9 FF 4C 00 01 20 D9
$63D0: FF A9 60 8D 03 01 4C 00
$63D8: 01 A2 03 BD 06 FF 95 FC
$63E0: CA 10 F8 A2 05 BD F1 FF
$63E8: 9D 00 01 CA 10 F7 A0 02
$63F0: 60 2C 82 C0 4C 06 BD AD
$63F8: 82 C0 F7 FF F7 FF F7 FF
```



Die RAM-Karten vo

Mit Quellcode eines RAM-Disk-Drivers

Die Firma IBS (Ingenieurbüro Specovius, Olper Str. 10, 4800 Bielefeld) ist bekannt durch eine riesige Auswahl an Zusatzkarten für den Apple II Plus und IIe (nicht IIc, da letzteres Gerät keine Slots hat). Wer will, kann z.B. seinen Apple durch eine 68000-Karte (AP 20) oder Z80B-Karte (AP 22) aufwerten. Darüber hinaus sind diverse Interface-Karten für Techniker lieferbar. Im nachfolgenden Beitrag beschränken wir uns auf die RAM-Karten.

RAM-Karten lassen sich in drei Gruppen einteilen:

1. Reine RAM-Karten

Wer eine RAM-Karte nur als RAM-Disk oder als sonstigen schnellen Zwischenspeicher für Daten benutzen will, wird sich selbstverständlich eine reine RAM-Karte kaufen. Zu den reinen RAM-Karten von IBS gehören

– AP 17: RAM-Karte mit maximal 256K, die in unterschiedlichen Ausbaustufen (64K, 128K, 196K und schließlich 256K) geliefert werden kann. Die oben abgebildeten Karten zeigen drei der möglichen Ausbaustufen. Es liegen RAM-Disk-Driver für DOS 3.3, Pascal 1.1 und CP/M 2.2 vor, d.h. ProDOS fehlt z.Zt. noch.
– AP 33: RAM-Karte mit wahlweise 256K oder 1024K (= 1M) RAM. Es liegen die gleichen RAM-Disk-Driver wie bei der AP 17 vor.

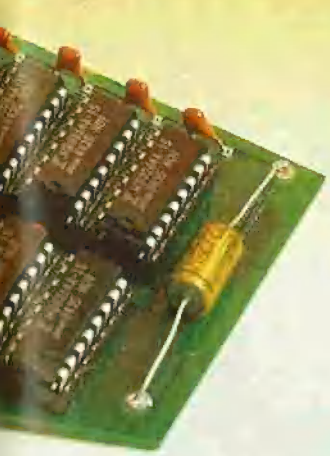
2. Prozessor-RAM-Karten

Prozessorkarten, die über zusätzliches RAM verfügen, lassen sich zwar ebenfalls meist als RAM-Disks einsetzen, doch wäre es eine Geldverschwendung, etwa eine 68000-Karte nur als Pseudo-Disk einzusetzen. Umgekehrt, auch wenn man eine Prozessor-Karte normalerweise für den Einsatz eines applefremden Betriebssystems (CP/M usw.) benötigt, ist es ganz nützlich zu wissen, daß man unter DOS 3.3 oder Pascal die Karte nicht brachliegen lassen muß. Zu den IBS-Prozessor-RAM-Karten gehören

– AP 20: INTEMEX 68000-Karte mit wahlweise 128K oder 512K RAM. Es liegen RAM-Disk-Driver für DOS 3.3, Pascal 1.1 und CP/M 2.2 vor, also auch hier noch nicht für ProDOS.
– AP 25: INTEMEX 8086 mit wahlweise 128K oder 512K RAM. Mit denselben Drivern wie bei der AP 20.

3. Mehrfunktion-RAM-Karten

Unter diesem Begriff fassen wir RAM-Karten zusammen, die vielfältig eingesetzt werden können. Hierzu gehört insbesondere die 64K-Karte des Apple IIe (bei IBS als AP 27 lieferbar), die zusätzlich folgenden Funktionen dienen kann:
a) 80-Zeichendarstellung



IBS

b) Double-Lores-Grafik
c) Double-Hires-Grafik
d) „Pseudo-Coprocessing"
Die letztgenannte Funktion wird Gegenstand eines gesonderten „Peeker"-Artikels sein.

128K-RAM-Karte

Nachfolgend beschränken wir uns auf die AP 20 mit 128K RAM (funktionsgleich mit der AP 17 in der 128K-Ausbaustufe), für die wir bereits vor einem Jahr Utilities für den Transfer von Meßdaten in unserer Hüthig-Zeitschrift „CAL" veröffentlicht haben. Diese Transfer- oder Move-Utilities umgehen den RAM-Disk-Driver, wodurch eine noch höhere Datenübertragungsrate erzielt werden kann, was bei Meßdatenerfassung bisweilen notwendig ist.
Wie aus **Diagramm 1** ersichtlich ist, werden die 128K ($000000–$01F0000) der AP 20 softswitchmäßig in 32 Blöcke zu je 4K eingeteilt. Da der 6502-Prozessor nur 64K adressieren kann, läßt sich ein Byte von der oder auf die Karte nicht mit z.B.

```
LDA $01FFFF
STA $2000
oder
LDA $1111
STA $011111
```

direkt übertragen. Deshalb werden 3 4K-Blöcke $D000–$DFFF, $E000–$EFFF und $F000–FFFF als Pseudo-Adressen für die 128K-RAM-Karte benutzt. Beispiel:

```
LDA $C082   LC abstellen
STA $C0CF   AP einschalten
LDA $1000   $1000 laden und in
STA $F000   $01F000 speichern
STA $C0C0   AP abstellen
RTS
```

Block-Nr. für 32 4K-Blöcke	Softswitch für Slot 4	Pseudo-Adresse für LDA und STA	Echte 128K-Adresse
$00	$C0C2	$E000	$000000
$01	$C0C4	$D000	$001000
$02	$C0C4	$E000	$002000
$03	$C0C4	$F000	$003000
$04	$C0C2	$D000	$004000
$1D	$C0CF	$D000	$01D000
$1E	$C0CF	$E000	$01E000
$1E	$C0CF	$F000	$01F000

Diagramm 1: Organisation der IBS-128K-RAM-Karte

Dieses Mini-Programm würde das Byte der Speicherstelle $1000 des Apple-RAM in die Speicherstelle $01F000 der AP 20 übertragen. Man beachte also, daß STA $F000 nicht das Byte in die Apple-Language-Card $F000 oder gar in das Apple-ROM $F000 pokt. Welche Softswitches im einzelnen angesprochen werden müssen, ergibt sich aus einer (für Anfänger leider nicht besonders übersichtlichen) Tabelle, die in dem AP-20-Handbuch enthalten ist. Die Softswitches sind slotabhängig, wobei die RAM-Karte in einen beliebigen Slot gesteckt werden kann. Der Slotbereich, z.B. $C400-$C4FF ist nicht ROM, sondern RAM, so daß man den Driver für die RAM-Disk oder eine sonstige Transfer-Utility in diesen Bereich BLOADen kann, falls das Programm dort platzmäßig unterzubringen ist. Diese Methode hat jedoch einen kleinen Schönheitsfehler, weil der Bereich $01FF00–$01FFFF in den Slotbereich $Cn00–$CnFF „gemappt" ist, so daß bei einer RAM-Disk dieser $01FF00-Bereich (= 1 Sektor) freigehalten werden muß, falls der Driver im Slot-RAM liegt.

Die skizzierte Methode des Bank-Selecting mit 4K-Blöcken und dem Pseudo-Adreßbereich $D000–$FFFF hat den Vorteil, daß die Datenübertragungsrate selbst bei einer RAM-Disk, falls diese per RWTS angesprochen wird, 64K/s beträgt. Nachteilig ist dagegen der Umstand, daß ein *direkter* Datentransfer zwischen 128K-Karte und Apple-LC nicht möglich ist.

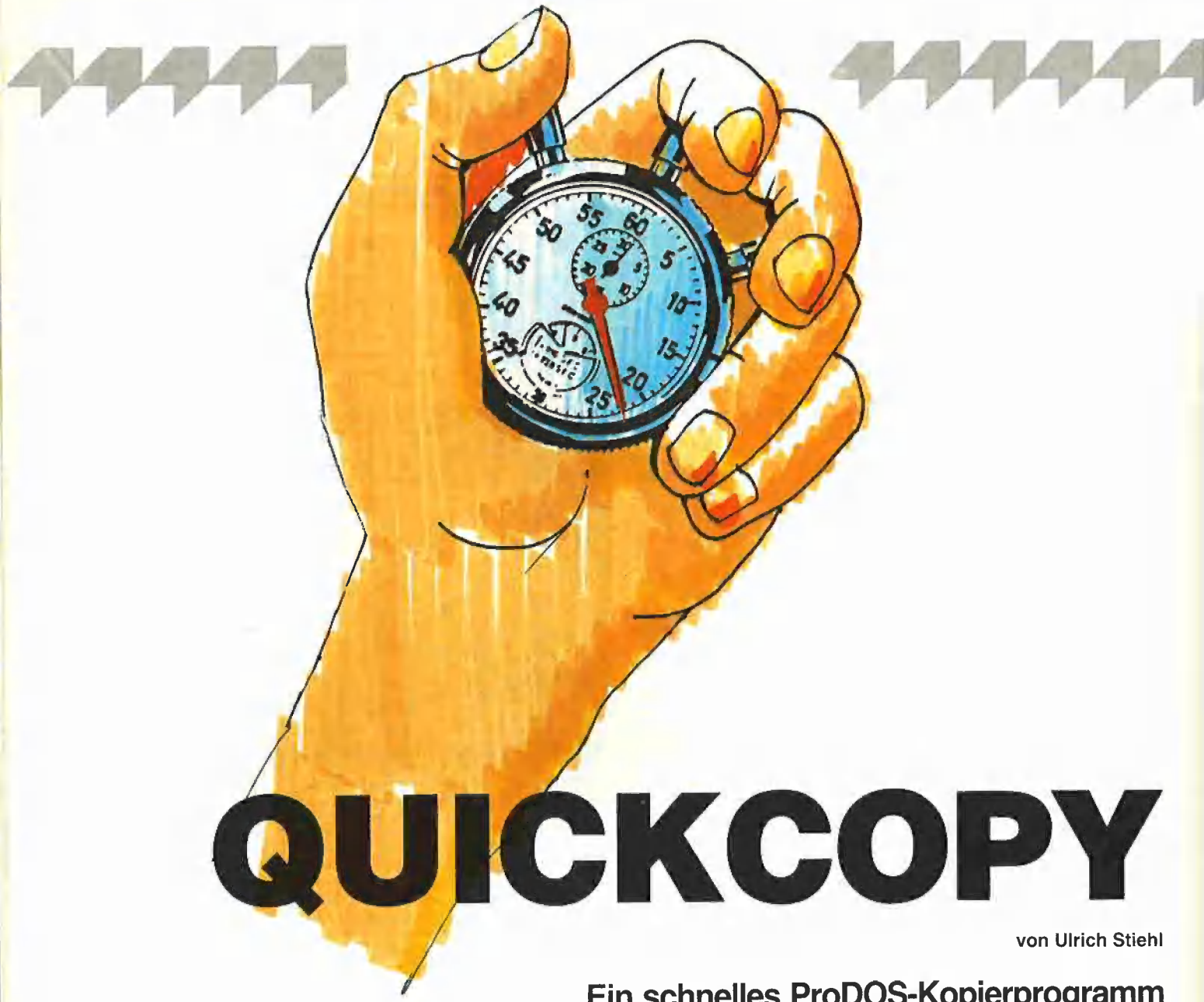
Deshalb dürfte ein ProDOS-RAM-Disk-Driver nur über einen Zwischenpuffer in den unteren 48K zu realisieren sein, weil PRODOS in der Bank 1 der LC liegt. Dadurch sinkt jedoch die Datenübertragungsrate stark ab.

Hinweis zu den Listings: Das Programm **IBS-RAM-Diskdriver** stellt den diassemblierten Objekt-Code eines von Gerd Blanke für die Firma IBS entwickelten Drivers für die AP 20 dar. Man beachte, daß dieser Driver neben $BD00 auch $A5B2 als absolute DOS-Adressen benutzt, weshalb manche modifizierten DOS-Versionen entsprechend angepaßt werden müssen. Der **AP20.RAMDISKTEST** zeigt, wie man auf die RAM-Disk per RWTS direkt zugreifen kann, womit eine Datenübertragungsrate erzielt werden kann, die einer Transfer-Utility (ohne RAM-Disk-Driver) sehr nahekommt.
us

```
                    1           ORG  $C400
                    2   *
                    3   * IBS-RAM-Diskdriver
                    4   * ==================
                    5   *
                    6   * RAM-Diskdriver für IBS-AP20
                    7   * 128K-RAM-Karte in Slot 4.
                    8   * Von Gerd Blanke für IBS,
                    9   * Bielefeld, geschrieben
                   10   * Disassembliert von U.Stiehl.
                   11   * Bei dieser Karte ist der
                   12   * Bereich $Cn00-$CnFF Slot-RAM,
                   13   * nicht Slot-ROM!
                   14   * (Man beachte, daß nur der Driver
                   15   * und nicht die SOFT- und INIT-Module
                   16   * disassembliert wurden.)
                   17   *
                   18   IOB       EQU  $0048       ;von Rwts
                   19   RDWR      EQU  $003C       ;Read/Write
                   20   TRSEC     EQU  $003D       ;Track/Sektor
                   21   RWTS      EQU  $BD04       ;weiter
                   22   SETROM    EQU  $A5B2       ;FPINT
                   23   INTERPR   EQU  $E000       ;FPINT
                   24   *
                   25   * Wenn Slot 4, dann RAM-Disk,
                   26   * andernfalls normale RWTS
                   27   *
C400: 84 48        28   ENTRY     STY  IOB         ;von $BD00
C402: 85 49        29             STA  IOB+1
C404: A0 01        30             LDY  #$01        ;Jetztslot
C406: B1 48        31             LDA  (IOB),Y
C408: C9 40        32             CMP  #$40        ;Slot 4
C40A: 08           33             PHP
C40B: A0 0F        34             LDY  #$0F        ;Vorslot
C40D: B1 48        35             LDA  (IOB),Y
C40F: 28           36             PLP
C410: F0 10        37             BEQ  SLDR1
C412: C9 40        38             CMP  #$40
C414: D0 09        39             BNE  OLDRWTS
C416: A9 60        40   VORSLOT   LDA  #$60        ;gepokt
C418: 91 48        41             STA  (IOB),Y     ;Vorslot
C41A: C8           42             INY
C41B: A9 01        43   VORDRIVE  LDA  #$01        ;gepokt
C41D: 91 48        44             STA  (IOB),Y     ;Vordrive
C41F: 4C 04 BD     45   OLDRWTS   JMP  RWTS
                   46   *
                   47   * Vorslot und Vordrive poken
                   48   *
C422: C9 40        49   SLDR1     CMP  #$40
C424: F0 0D        50             BEQ  SLDR2
C426: 8D 17 C4     51             STA  VORSLOT+1
C429: A9 40        52             LDA  #$40        ;Slot 4
C42B: 91 48        53             STA  (IOB),Y
C42D: C8           54             INY
C42E: B1 48        55             LDA  (IOB),Y
C430: 8D 1C C4     56             STA  VORDRIVE+1
C433: A0 02        57   SLDR2     LDY  #$02        ;Jetztdrive
C435: B1 48        58             LDA  (IOB),Y
C437: A0 10        59             LDY  #$10        ;Vordrive
C439: 91 48        60             STA  (IOB),Y
C43B: C9 02        61             CMP  #$02        ;Drive > 2?
C43D: B0 10        62             BCS  IOERROR
                   63   *
                   64   * Befehl ermitteln
                   65   *
C43F: A0 0C        66             LDY  #$0C        ;Befehl
C441: B1 48        67             LDA  (IOB),Y
C443: D0 06        68             BNE  INIT?
                   69   *
                   70   * Bei Seek + Init sofort Exit
                   71   *
C445: 18           72   EXIT      CLC              ;00=Seek
C446: A0 0D        73   ERRNUM    LDY  #$0D        ;Error-Nr.
C448: 91 48        74             STA  (IOB),Y
C44A: 60           75             RTS              ;Ende
                   76   *
C44B: C9 04        77   INIT?     CMP  #$04        ;04=Init
C44D: D0 05        78             BNE  READ?
C44F: A9 40        79   IOERROR   LDA  #$40        ;I/O-Error
C451: 38           80             SEC
C452: B0 F2        81             BCS  ERRNUM
                   82   *
                   83   * Read=01 (BMI), Write=02 (BPL)
                   84   *
C454: 4A           85   READ?     LSR              ;Bit 0 bei Read
C455: 66 3C        86             ROR  RDWR        ;Bit 7 bei RDWR
                   87   *
                   88   * Language Card (wegen Integer Basic) aktiv?
                   89   *
C457: AD 00 E0     90             LDA  INTERPR     ;Basic-Anf.
C45A: 8D CD C4     91             STA  FPINT+1
                   92   *
                   93   * Track/Sektor in RAM-Disk-Bereich umwandeln
                   94   *
C45D: A0 04        95             LDY  #$04        ;Spur
C45F: B1 48        96             LDA  (IOB),Y
C461: 38           97             SEC
C462: E9 01        98             SBC  #$01
C464: 4A           99             LSR
C465: AA          100             TAX
C466: BD F0 C4    101             LDA  AP20BANK,X
C469: A2 00       102             LDX  #$00        ;Low=00
C46B: 48          103             PHA
C46C: 29 F0       104             AND  #$F0
C46E: 90 02       105             BCC  TRSEC1
C470: 69 07       106             ADC  #$07
C472: 85 3D       107   TRSEC1    STA  TRSEC
C474: C8          108             INY
C475: B1 48       109             LDA  (IOB),Y     ;Sektor
C477: 29 07       110             AND  #$07
C479: 05 3D       111             ORA  TRSEC
C47B: 24 3C       112             BIT  RDWR
C47D: 30 08       113             BMI  TRSEC2
C47F: 8D C5 C4    114             STA  DUMMY2+2    ;Write
C482: 8E C4 C4    115             STX  DUMMY2+1
C485: 10 06       116             BPL  TRSEC3
C487: 8D C2 C4    117   TRSEC2    STA  DUMMY1+2    ;Read
C48A: 8E C1 C4    118             STX  DUMMY1+1
C48D: B1 48       119   TRSEC3    LDA  (IOB),Y
C48F: 29 08       120             AND  #$08
C491: 85 3D       121             STA  TRSEC
C493: 68          122             PLA
C494: 29 0F       123             AND  #$0F
C496: 05 3D       124             ORA  TRSEC
C498: 09 C0       125             ORA  #$C0
C49A: 8D BE C4    126             STA  SOFTON+1
                  127   *
                  128   * DOS-Puffer ermitteln
                  129   *
C49D: A0 08       130             LDY  #$08
C49F: B1 48       131             LDA  (IOB),Y     ;Low-Buffer
C4A1: 48          132             PHA
C4A2: C8          133             INY
C4A3: B1 48       134             LDA  (IOB),Y     ;High-Buffer
C4A5: 24 3C       135             BIT  RDWR
C4A7: 30 0A       136             BMI  DOSPUFF
C4A9: 8D C2 C4    137             STA  DUMMY1+2    ;Write
C4AC: 68          138             PLA
C4AD: 8D C1 C4    139             STA  DUMMY1+1
C4B0: 4C BA C4    140             JMP  RDROM
C4B3: 8D C5 C4    141   DOSPUFF   STA  DUMMY2+2    ;Read
C4B6: 68          142             PLA
C4B7: 8D C4 C4    143             STA  DUMMY2+1
                  144   *
                  145   * Softswitches aktivieren
                  146   *
C4BA: 8D 82 C0    147   RDROM     STA  $C082       ;LCOFF
C4BD: 8D C2 C0    148   SOFTON    STA  $C0C2       ;gepokt
                  149   *
                  150   * Total gepokte Move-Routine:
                  151   * Entweder Read AP20,   Write Puffer
                  152   * oder     Read Puffer, Write AP20
                  153   *
C4C0: BD 00 10    154   DUMMY1    LDA  $1000,X     ;Dummy
C4C3: 9D 00 10    155   DUMMY2    STA  $1000,X     ;Dummy
C4C6: E8          156             INX
C4C7: D0 F7       157             BNE  DUMMY1
C4C9: 8D C0 C0    158   SOFTOFF   STA  $C0C0
                  159   *
                  160   * LC ein- oder ausschalten
                  161   *
C4CC: A9 4C       162   FPINT     LDA  #$4C        ;gepokt
C4CE: 20 B2 A5    163             JSR  SETROM
C4D1: 8A          164             TXA              ;Error=0
C4D2: 4C 45 C4    165             JMP  EXIT
                  166   *
C4D5: AA AA AA    167   COPYRT    ASC  "***(C) 1983 "
C4E1: C7 C5 D2    168             ASC  "GERD BLANKE****"
                  169   *
                  170   * Tabelle der AP20-Bank-Softswitches
                  171   *
C4F0: E2 D4 E4    172   AP20BANK  HEX  E2D4E4F4
C4F4: F2 D6 E6    173             HEX  F2D6E6F6
C4F8: E3 D5 E5    174             HEX  E3D5E5F5
C4FC: F3 D7 E7    175             HEX  F3D7E7F7
256 bytes
```

RAM-disk

```
                        1           ORG  $300
                        2   *
                        3   * AP20.RAMDISKTEST
                        4   * ================
                        5   *
                        6   * Für RAM-Disk mit 128K:
                        7   * Dauer des Testlaufs: 4s
                        8   * Es werden also ca. 64K/s
                        9   * gelesen oder geschrieben
                       10   *
                       11   RWTS     EQU  $03D9
                       12   PUFFER   EQU  $2000
                       13   PRINT    EQU  $FDED
                       14   *
0300: 4C 1B 03         15           JMP  START
                       16   *
0303: 01               17   IOB      DFB  1
0304: 40               18            DFB  $40          ;Slot 4
0305: 01               19            DFB  1            ;Drive 1
0306: 00               20            DFB  0            ;Volume
0307: 00               21   TRACK    DFB  0            ;Track
0308: 00               22   SECTOR   DFB  0            ;Sektor
0309: 14               23            DFB  #<DCT
030A: 03               24            DFB  #>DCT
030B: 00               25   PUFFLOW  DFB  $00          ;$4000
030C: 20               26   PUFFHIGH DFB  $20
030D: 00 00            27            DFB  0,0          ;frei
030F: 01               28   READWRIT DFB  1            ;1=R;2=W
0310: 00               29   ERRCODE  DFB  0
0311: 00               30            DFB  0            ;volume
0312: 40               31            DFB  $40          ;Slot 4
0313: 01               32            DFB  1            ;Drive 1
                       33   *
0314: 00 01 EF         34   DCT      DFB  0,1,239,216
0317: D8
                       35   *
0318: 01               36   TRCKANF  DFB  1
0319: 20               37   TRCKEND  DFB  32
031A: 00               38   TRCKCNT  DFB  0            ;Zähler
                       39   *
                       40   * Von Track 1-32 je Read/Write
                       41   *
031B: AD 18 03         42   START    LDA  TRCKANF
031E: 8D 1A 03         43            STA  TRCKCNT
0321: 8D 07 03         44   TRCKLOOP STA  TRACK
0324: A9 01            45            LDA  #1           ;Read
0326: 8D 0F 03         46            STA  READWRIT
0329: 20 42 03         47            JSR  READWRT
032C: A9 02            48            LDA  #2           ;Write
032E: 8D 0F 03         49            STA  READWRIT
0331: 20 42 03         50            JSR  READWRT
0334: EE 1A 03         51            INC  TRCKCNT
0337: AD 1A 03         52            LDA  TRCKCNT
033A: CD 19 03         53            CMP  TRCKEND
033D: F0 E2            54            BEQ  TRCKLOOP
033F: 90 E0            55            BCC  TRCKLOOP
0341: 60               56            RTS
                       57   *
                       58   * Read/Write eines Tracks
                       59   * von/in Sektor $0F-$00
                       60   * in/aus Puffer $2000-2FFF
                       61   *
0342: A9 10            62   READWRT  LDA  #16          ;16-1
0344: 8D 08 03         63            STA  SECTOR
0347: A9 00            64            LDA  #<PUFFER
0349: 8D 0B 03         65            STA  PUFFLOW
034C: A9 1F            66            LDA  #>PUFFER-1   ;-256
034E: 8D 0C 03         67            STA  PUFFHIGH     ;$1F00
0351: EE 0C 03         68   SECTLOOP INC  PUFFHIGH
0354: CE 08 03         69            DEC  SECTOR
0357: AD 08 03         70            LDA  SECTOR
035A: 30 13            71            BMI  RETURN
035C: A9 03            72            LDA  #>IOB
035E: A0 03            73            LDY  #<IOB
0360: 20 D9 03         74            JSR  RWTS
0363: B0 03            75            BCS  ERROR
0365: 4C 51 03         76            JMP  SECTLOOP
0368: 68               77   ERROR    PLA
0369: 68               78            PLA
036A: A9 87            79            LDA  #$87         ;Bell
036C: 20 ED FD         80            JSR  PRINT
036F: 60               81   RETURN   RTS

112 bytes
```



QUICKCOPY

von Ulrich Stiehl

Ein schnelles ProDOS-Kopierprogramm für 35–80 Spuren

Das im ProDOS-FILER enthaltene Teilprogramm zum Kopieren ganzer Disketten hat drei Nachteile:

– Erstens ist es zu langsam, da immer nur schubweise 4 Spuren kopiert werden mit der Folge, daß beispielsweise bei der klassischen 35-Spur-Diskette 9 Durchläufe für Lesen und Schreiben erforderlich sind, was zu einer beachtlichen Geschwindigkeitsreduzierung führt, denn der Laufwerkmotor muß insgesamt 19mal ein- und ausgeschaltet werden (1mal für Formatieren, 9mal für Lesen und 9mal für Schreiben).

– Zweitens können neben dem Profile-Festplattenlaufwerk und dem 35-Spur-Diskettenlaufwerk keine anderen Disk-Drives, z.B. solche für 40- oder 80-Spur-Disketten benutzt werden.

– Drittens kann man vom ProDOS-FILER nicht, wie dies bei dem früheren COPYA für DOS 3.3 der Fall war, direkt in den Applesoft-Modus zurück, sondern muß vielmehr das BASIC.SYSTEM neu laden, wodurch unnötige Zeit verlorengeht.

Nachfolgend werden deshalb zwei neue ProDOS-Kopierprogramme – PRODOS.-COPYA und QUICKCOPY – vorgestellt. Für beide Programme gilt gemeinsam:

1. Sie sind nur für Diskettenlaufwerke, nicht für Festplattenlaufwerke gedacht, doch können beliebig viele Spuren kopiert werden, so daß auch 40- und 80-Spur-Disketten duplizierbar sind.
2. Sie können verlassen werden, ohne daß man danach neu booten muß, d.h. das BASIC.SYSTEM wird nicht gelöscht oder verändert.
3. Sie sind erheblich schneller als der ProDOS-FILER. Das Kopieren einer 35-Spur-Diskette dauert beim ProDOS-FILER 76s, beim PRODOS.COPYA 63s und beim QUICKCOPY 57s.
4. Sie sind für Besitzer von 2 Laufwerken gedacht. (Für diejenigen, die nur über ein einziges Drive verfügen, können auf Wunsch der Leser spezielle 1-Drive-Kopierprogramme veröffentlicht werden. Schreiben Sie an die Redaktion von „Peeker"!)
5. Sie enthalten als Kleinzitat eine leicht modifizierte Version der Formatierroutine, die Bestandteil des ProDOS-FILERs ist. Diese Routine kann unabhängig vom FILER mit
BLOAD FILER, TSYS, A$7900, B$5900, L$043A
eingeladen werden. Assembler-Programmierer können die Routine dann mit
LDA UNITNUMBER
JSR $7900
aufrufen. Zuvor kann man noch mit
LDA VOLUME-NUMMER
STA $79EF
die Volume-Nummer und mit
LDA SPUR-ANZAHL
STA $79F4
die Anzahl der Spuren festlegen.
6. Sie können neben ProDOS- auch DOS-, Pascal- und CP/M-Disketten kopie-

ren. Allerdings muß vorher auf alle Fälle das ProDOS-Betriebssystem gestartet worden sein.

QUICKCOPY

QUICKCOPY ist als professionelles Kopierprogramm für Besitzer des Apple IIc oder des Apple IIe mit 64K-Karte gedacht. Es wird mit BRUN QUICKCOPY gestartet und meldet sich mit „Start J/N". Bevor man „J" oder „j" tippt, muß man bereits Original- und Duplikatdiskette eingelegt haben. Ferner sollte man daran denken, daß durch QUICKCOPY der Inhalt der RAM-Disk zerstört wird.
Man muß bei QUICKCOPY im Gegensatz zum ProDOS-FILER und zum PRODOS.-COPYA keinen Fragenkatalog durchgehen, sondern kann direkt mit dem Kopieren beginnen. QUICKCOPY gestattet eine individuelle Konfigurierung durch entsprechende Pokes:

BLOAD QUICKCOPY

POKE 2222 + 3, Original-Slot (1-7)
POKE 2222 + 4, Original-Drive (1-2)
POKE 2222 + 5, Duplikat-Slot (1-7)
POKE 2222 + 6, Duplikat-Drive (1-2)
POKE 2222 + 7, Volume-Nummer (1-254)
POKE 2222 + 8, INIT (0 = nein; 1 = ja)
POKE 2222 + 9, Spur-Anzahl (35-160)
POKE 2222 + 10, Durchläufe (?)
POKE 2222 + 11, Puffer-Ende (?)

BSAVE QUICKCOPY, A2222, L1881

Anmerkungen:

a) Man beachte, daß QUICKCOPY für 2-Drive-Besitzer gedacht ist, d.h. Original-Slot und -Drive dürfen nicht mit Duplikat-Slot und -Drive identisch sein. Die Drives brauchen indes nicht an demselben Slot angeschlossen zu sein.
b) Die Volume-Nummer kann man ignorieren, da sie auch vom ProDOS-Betriebssystem ignoriert wird. Der ProDOS-FILER formatiert übrigens stets mit Volume-Nummer 1.
c) Das INIT-Flag bestimmt, ob die Duplikatdiskette vor dem Kopiervorgang formatiert wird oder nicht. Wenn man als Duplikatdisketten bereits früher initialisierte Disketten benutzt, kann man viel Zeit sparen, da dann QUICKCOPY nur 35s für den gesamten Kopiervorgang benötigt. QUICKCOPY prüft bei INIT-Flag = 0 (= kein INIT), ob die Diskette bereits formatiert war. Wenn dies nicht der Fall sein sollte, erfolgt die Formatierung automatisch. Man könnte also das INIT-Flag permanent auf 0 setzen.
d) Spur-Anzahl, Anzahl der Durchläufe und Puffer-Ende (als Page) sind technischer Natur. Beispiel: Eine 35-Spur-Diskette hat 140K Speicherkapazität. Da der interne Kopierpuffer von QUICKCOPY maximal 70K beträgt, benötigt man für eine solche Diskette 2 Kopierdurchläufe (70K mal 2 = 140K). Die Pufferobergrenze beträgt dann $9A00 und die Page wäre $9A. Das kleine Applesoft-Programm QUICKCOPY.PUFFER errechnet nach Eingabe der Spur-Anzahl automatisch die zu pokenden Werte für die Durchläufe und das Puffer-Ende.
e) Die Ersatz-Parameter sind bei dem gelisteten Programm auf Slot 6, Drive 1 und 2, INIT ja und 35 Spuren eingestellt, also die normale Konfiguration für 2-Drive-Besitzer.

QUICKCOPY gehört zu jenen sich selbst verschiebenden Programmen, die sehr schwer durchschaubar wären, wenn man nicht über den Quell-Code verfügte. Die nachfolgenden Anmerkungen beziehen sich auf die Zeilennummern des Assembler-Listings:

1–118: Umfaßt Parameter-Liste und Sprung zum Menü.

119–767: Enthält modifizierte und neu assemblierte Formatierroutine aus dem ProDOS-FILER (hier nicht gelistet).
768–801: Tastaturabfrage für „Start J/N".
802–844: Dieser Teil kopiert die Global Page von den unteren 48K des Motherboards in die oberen 48K der 64K-Karte, sichert den momentanen Inhalt der Zero-Page und verschiebt dann das eigentliche Kopierprogramm komplett in die Zero-Page. Dies ist ungewöhnlich, jedoch wegen der Softswitches der 64K-Karte hier besonders vorteilhaft, da somit die Disketten-Blocks direkt in die 64K-Karte geladen werden können, womit zeitraubende Move-Routinen entfallen. Die Datenübertragungsrate von ca. 8K/s wäre sonst für den 64K-Karte-Puffer nicht zu erreichen (280K : 35s = 8K/s).
845–900: Aufruf der Formatierroutine.
901–920: Aufruf der Kopierroutine in der Zero-Page. Danach wird der alte Zustand der Zero-Page wieder hergestellt und zum Menü zurückgesprungen.
921–940: Unterroutine zum Sichern und Wiederherstellen der Zero-Page.
941–951: Unterroutine, die Slot- und Drive-Nummer zur sog. Unit-Number in ein einziges Byte packt.
952–978: Hier wird erstens die PRODOS-Spurprüfroutine gepatcht, so daß nunmehr auch Disketten mit mehr als 280 Blocks (=

ProDOS-Editor 1.0

Applesoft-Editor unter ProDOS-Betriebssystem

von U. Stiehl

1984, Diskette und Manual, DM 98,–
ISBN 3-7785-1024-X

Mit diesem neuen Editor – übrigens der bislang einzige deutsche ProDOS-Editor – wird dem Applesoft-Programmierer ein Werkzeug zur effektiven Programmierung unter dem Betriebssystem ProDOS gegeben, denn die früheren Editoren sind allesamt unter ProDOS nicht mehr lauffähig.

Unter anderem sind folgende Features implementiert worden:

- Zeilenorientierter Editor mit jedem erdenklichen Redigierkomfort (Insert, Delete, Tab, Restore, freie Cursorbewegung in allen vier Richtungen, Eingabe von Ctrl-Buchstaben in Applesoft-Zeilen usw.)
- Renumber (Zeilen-Umnumerierung)
- Xreference (sortierte Variablenliste)
- Suchen von Tokens, Strings und Variablen
- dezimale und hexadezimale Umrechnungen
- Ausführung von Monitorbefehlen aus dem Editor heraus
- Listen des Applesoft-Programms in speicherinterner Form als Hex-Dump
- Suchen von Hex-Folgen, Adressen oder Speicherstellen im gesamten RAM-Bereich einschließlich der Language-Card
- frei definierbare Tastatur-Macrobefehle

Der Applesoft-Editor liegt in einem von ProDOS geschützten Bereich und läßt sich per Tastendruck vorübergehend abschalten und ebenso einfach wieder aktivieren.

Gerätevoraussetzung: Apple II+, IIe oder IIc

Hüthig Software Service,
Postfach 10 28 69,
D-6900 Heidelberg

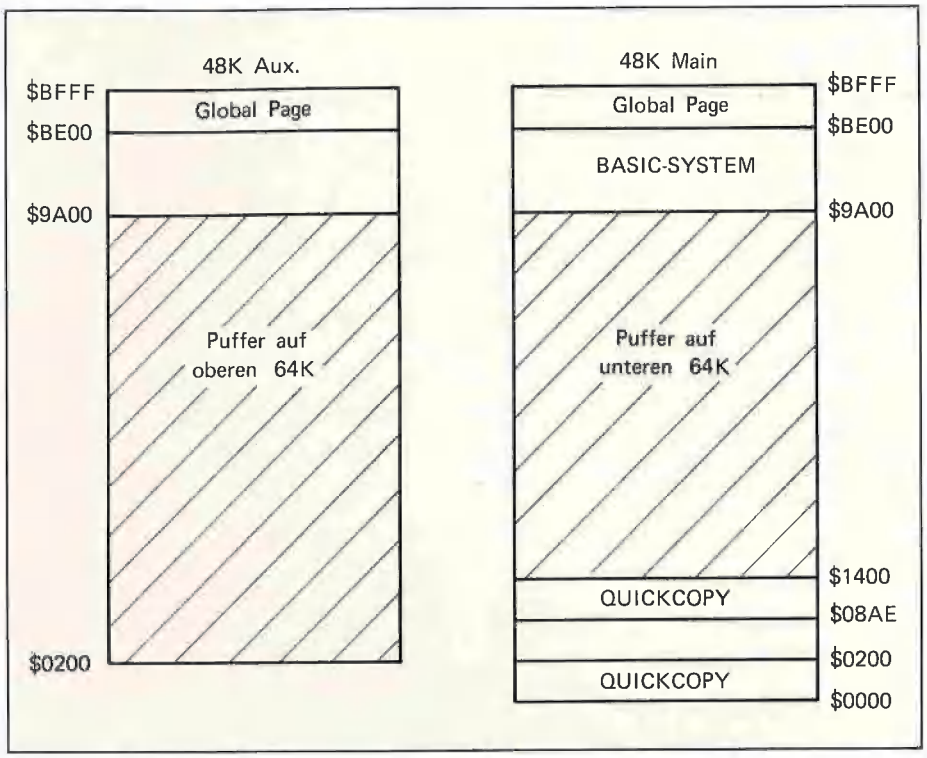


Diagramm 1: Speicherverteilung bei QUICKCOPY

35 Spuren) kopiert werden können, und zweitens wird die RAM-Disk abgestellt, da die 64K-Karte als Puffer verwendet wird.
979–1103: Dies ist die eigentliche Kopierroutine. Wenn eine 35-Spur-Diskette dupliziert wird, wird in nur zwei Durchläufen oder Schüben von je 70K, die sich in 36.5K auf der 64K-Karte und 33.5K auf den unteren 48K einteilen, die ganze Originaldiskette auf die Duplikatdiskette kopiert. Zeile 992–996 kontrolliert die Anzahl der Durchläufe. Zeile 1003–1024 ist die 70K-Hauptlese- und -schreibschleife. Zeile 1037–1045 aktiviert und deaktiviert die 64K-Karte. Zeile 1047–1079 kontrolliert die 70K-Lese-Schreibvorgänge, die durch die Read- und Write-Block-Routinen in Zeile 1081–1103 auf Blockebene durchgeführt werden. Die Block-Routinen gehören zum sog. MLI = Machine Language Interface, das in einem späteren Heft von „Peeker" analysiert wird.

Das **Diagramm 1** veranschaulicht die Speicherorganisation von QUICKCOPY.

2. PRODOS.COPYA

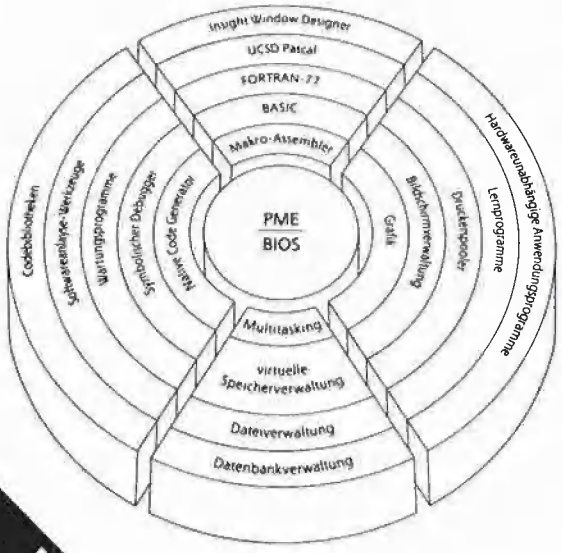
PRODOS.COPYA – das dem alten COPYA für DOS 3.3 nachempfunden ist – läuft auf jedem Apple mit 64K, z.B. Apple II Plus oder Apple IIe ohne 64K-Karte usw. Es benutzt einen konstanten, 32K großen Datenpuffer, der 8 Spuren aufnehmen kann. Für 80-Spurlaufwerke wären damit 10 Durchläufe erforderlich (gegenüber 5 Durchläufen bei QUICKCOPY). PRODOS.-COPYA wird durch das Applesoft-Programm namens PRODOS.COPYA gestartet, welches seinerseits PRODOS.-COPYOBJ einlädt. In dem Applesoft-Programm können in Zeile 120 die folgenden Variablen geändert werden:

OS = Original-Slot
OD = Original-Drive
DS = Duplikat-Slot
DD = Duplikat-Drive
S = Spurenanzahl

Wenn man mit dem Programm erst einmal vertraut ist, kann man auf PRODOS.-COPYA gänzlich verzichten und mit
BRUN PRODOS.COPYOBJ
das Kopierprogramm direkt starten, da es ein eigenes Menü mit „START J/N" hat. Aus Platzgründen ist zu PRODOS.-COPYOBJ nur der Hex-Dump abgedruckt.

Hinweis: In einem der nächsten Hefte von „Peeker" wird ein FID-ähnliches Programm zum Kopieren von Dateien veröffentlicht werden.

Das UCSD p-System.



Jetzt auch für Apple Macintosh

FOCUS Computer GmbH
Friesenstrasse 14 · 3000 Hannover 1

Professionelle Softwarewerkzeuge von FOCUS:

UCSD p-System: Das komplette Entwicklungssystem mit UCSD Pascal, FORTRAN-77, BASIC, acht verschiedenen Assemblern und dem INSIGHT Window Designer. Für mehr als 100 Rechner: Ihre Software läßt sich leicht auf alle diese Rechner übertragen.

MacAdvantage Der UCSD Pascal Compiler für den Macintosh, der direkt mit dem Mac-Betriebssystem zusammenarbeitet. Zugriff auf alle Macintosh-Features direkt von Pascal aus, volle Unterstützung für den 512K-'Fat Mac', symbolischer Debugger, Resource File Compiler. Der mitgelieferte Maus-Editor kann in mehreren Fenstern mehrere, beliebig lange Dateien bearbeiten.

T.L.D.U. Die programmierbare Disk-Utility für alle Computer der Apple II Serie. Mit Unterstützung für 35-, 40-, 80- und 160-Spur-Laufwerke und Festplatten. Für DOS, CP/M, UCSD oder ProDOS einstellbar. 100 Seiten deutsches Handbuch mit Tutorium, damit auch Anfänger Disketten editieren und reparieren können. Empfohlener Verkaufspreis DM 99,-- einschließlich Mehrwertsteuer.

Sie können sich diese Produkte bei Ihrem Apple-Händler ansehen oder uns unter 0511-345461 anrufen.

The Last Disk Utility

---

weidemann electronic

Apple Service Händler Level I
24 Std. Reparaturservice

Achtung: alle Geräte mit Seriennummer vom Hersteller

Apple II e

Apple IIe mit 64 kb	2418,– DM
Einstiegspaket (IIe, Monitor, Disk mit Contr.)	3368,– DM
Disk m. C. (Prodos, DOS)	1177,– DM
Disk o. Contr.	866,– DM
DuoDisk incl. Contr.	2016,– DM
Profile 5 MB incl. Interf.	4299,– DM
II MOuse incl. Interface	376,– DM

Apple IIc

Apple IIc	2655,– DM
Komplettpaket (IIc incl. Monitor und Ständer)	3249,– DM
IIc 2. laufwerk	855,– DM
IIc Mouse incl. Mousepaint	262,– DM

Drucker + Plotter

Apple Scribe incl. Kabel	849,– DM
Apple Imagewriter 12" incl. Kabel	1484,– DM
Apple Imagewriter 15" incl. Kabel	2176,– DM
Apple A3 Plotter 4 Farben	2252,– DM
Epson RX-80	957,– DM
Epson FX-80	1479,– DM
Epson FX-100	1917,– DM
Parallel Interface incl. Kabel	211,– DM
Seriell Interf. für Apple Drucker	211,– DM
parallel/seriell Interface	229,– DM

Macintosh

Macintosh 128 kb	6299,– DM
Macintosh 128 kb (110 V. Vers.)	5999,– DM
Macintosh 512 kb	9188,– DM
Aufrüstung auf 512 kb	2948,– DM
Externes Laufwerk	1239,– DM

Fordern Sie unsere Softwareliste an!

Monitore + Karten

Apple IIe Monitor	562,– DM
Apple IIc Monitor	512,– DM
Zenith ZVM-122-EA bernstein	**299,– DM**
Zenith ZVM-123-EA rün	307,– DM
Taxan RGB Vision I, 15 MHz.	755,– DM
Taxan RGB Vision II, 18 MHz.	1073,– DM
Taxan RGB Vision III, 20 MHz.	1368,– DM
80 Zeichenkarte IIe	171,– DM
80 Zeichenkarte + 64 kb	376,– DM
RGB + 80 Zeichenkarte	399,– DM
RGB + 80 Zeichen + 64 kb	707,– DM

Software

Wir liefern jede Software für Apple
z.B. Aztek C-Compiler für Macintosh
1699,– DM
Fordern Sie unsere Listen an!

Auf alle Apple Produkte 1 Jahr Garantie
Sonstige 6 Monate (Epson ohne Druckkopf)
Alle Preise incl. Mehrwertsteuer

Anfragen + Bestellungen

Tel. 0 26 34 / 36 36 / 36 37 / 36 38 / 36 39
Telex: 86 85 13
IMCA/Auge Mailbox an U. Weidemann
5455 Rengsdorf Wohnplatz Weidemann

```
                        1               ORG  2222
                        3   *
                        4   * QUICKCOPY für ProDOS
                        5   * ═══════════════════
                        6   *
                        7   * Gerätevoraussetzung: Apple IIc
                        8   * oder IIe mit 64K-Karte und
                        9   * 2 Laufwerken
                       10   *
                       11   * von U.Stiehl/06.10.84
                       12   *
                       13   HOME      EQU  $FC58
                       14   PRINT     EQU  $FDED
                       15   PRERR     EQU  $FF2D
                       16   HEXOUT    EQU  $FDDA
                       17   RDKEY     EQU  $FD0C
                       18   MLI       EQU  $BF00
                       19   *
                       20   * Main $1400-$9A00 = 33.5K
                       21   * Aux. $0800-$9A00 = 36.5K
                       22   * ------------------------
                       23   * Zusammen          = 70.0K
                       24   *
                       25   PUFFBEGM EQU  $1400       ;Main
                       26   PUFFBEGA EQU  $0800       ;Aux.
                       27   *
08AE: 4C BD 08         28               JMP  START1
                       29   *
                       30   *-------------------------------
                       31   *
                       32   * In diesem Bereich können die
                       33   * gewünschten Parameter gepokt
                       34   * werden.
                       35   *
                       36   * Original-Slot und -Drive
                       37   *
08B1: 06               38   OSLOT     HEX  06
08B2: 01               39   ODRIVE    HEX  01
                       40   *
                       41   * Duplikat-Slot und -Drive
                       42   *
08B3: 06               43   DSLOT     HEX  06
08B4: 02               44   DDRIVE    HEX  02
                       45   *
                       46   * Volume-Nummer 1-254
                       47   *
08B5: 01               48   VOLUME    HEX  01
                       49   *
                       50   * Initflag: 0=nein, 1=ja
                       51   *
08B6: 01               52   INITFLAG HEX  01          ;INIT=ja
                       53   *
                       54   *-------------------------------
                       55   *
                       56   * Anzahl der Spuren, Anzahl der
                       57   * Durchläufe und Ende des Puffers
                       58   * müssen aufeinander abgestimmt
                       59   * sein. Es gilt im einzelnen:
                       60   *
                       61   * $23 Spuren, $02 Läufe, $9A Ende
                       62   * $24 Spuren, $03 Läufe, $6E Ende
                       63   * $28 Spuren, $04 Läufe, $5E Ende
                       64   * $50 Spuren, $05 Läufe, $8E Ende
                       65   * $A0 Spuren, $0A Läufe, $8E Ende
                       66   *
                       67   * Jeder Lauf ist gleich groß,
                       68   * es gibt also keinen kleineren
                       69   * Restlauf. 100% optimal sind
                       70   * nur 35 = $23 Spuren, für die
                       71   * das Programm speziell
                       72   * geschrieben worden ist.
                       73   *
                       74   * Spuren
                       75   *
08B7: 23               76   SPUREN1  HEX  23          ;35
                       77   *
                       78   * Läufe
                       79   *
08B8: 02               80   LAEUFE1  HEX  02          ;2 mal
                       81   *
                       82   * Ende
                       83   *
08B9: 9A               84   PUFFEND1 HEX  9A          ;$9A00
                       85   *
                       86   *-------------------------------
                       87   *
```

```
                       88   * Nachträglich ERROR-Nummer hier
                       89   *
08BA: 00               90   ERROR     HEX  00
                       91   *
                       92   *-------------------------------
                       93   *
                       94   * Original- und Duplikat-Unit
                       95   *
08BB: 60               96   OUNIT     HEX  60          ;S6,D1
08BC: E0               97   DUNIT     HEX  E0          ;S6,D2
                       98   *
                       99   *-------------------------------
                      100   *
08BD: 20 58 FC        101   START1    JSR  HOME
08C0: A2 00           102             LDX  #0
08C2: BD D0 08        103   START2    LDA  STRING,X
08C5: F0 06           104             BEQ  START3
08C7: 20 ED FD        105             JSR  PRINT
08CA: E8              106             INX
08CB: D0 F5           107             BNE  START2
08CD: 4C 3A 0E        108   START3    JMP  START4
                      109   *
08D0: 2A 11 15        110   STRING    INV  "*QUICK-COPY*"
08DC: 8D              111             HEX  8D
08DD: 15 2E 20        112             INV  "U. STIEHL 84"
08E9: 8D 8D           113             HEX  8D8D
08EB: D3 D4 C1        114             ASC  "START J/N "
08F5: 00              115             HEX  00
08F6: 00 00 00        116             HEX  0000000000
08FB: 00 00 00        117             HEX  0000000000
                      118   *
                      119   *-------------------------------
                      120   *
                      121   * ProDOS-Format
                      122   * ═══════════
                      123   *
                      124   * Dem FILER der Firma Apple
                      125   * mit Modifikationen entnommen.
                      126   * Hier nicht gelistet.
                      127   *
                      767   *═══════════════════════════════
                      768   *
                      769   ZAREA     DS   256
                      770   *
                      771   * ProDOS-Quickcopy
                      772   * ═══════════════
                      773   *
0E3A: 2C 10 C0        774   START4    BIT  $C010       ;Strobe
0E3D: A9 00           775             LDA  #0
0E3F: 8D BA 08        776             STA  ERROR       ;okay
                      777   *
                      778   * Durchläufe und Pufferende poken
                      779   *
0E42: AD B8 08        780             LDA  LAEUFE1
0E45: 8D 0A 00        781             STA: LAEUFE2
0E48: AD B9 08        782             LDA  PUFFEND1
0E4B: 8D 0B 00        783             STA: PUFFEND2
                      784   *
                      785   * MenÜ
                      786   *
0E4E: 20 0C FD        787   START5    JSR  RDKEY
0E51: C9 CA           788             CMP  #"J"
0E53: F0 0F           789             BEQ  COPY1
0E55: C9 EA           790             CMP  #"j"
0E57: F0 0B           791             BEQ  COPY1
0E59: C9 CE           792             CMP  #"N"
0E5B: F0 06           793             BEQ  EXIT1
0E5D: C9 EE           794             CMP  #"n"
0E5F: F0 02           795             BEQ  EXIT1
0E61: D0 EB           796             BNE  START5
                      797   *
                      798   * Okay-Exit nach Menü
                      799   *
0E63: 60              800   EXIT1     RTS              ;Exit
                      801   *
                      802   *-------------------------------
                      803   *
                      804   * Zero-Page retten und
                      805   * MLI-Routine in Zero-Page
                      806   * kopieren
                      807   *
0E64: 20 03 0F        808   COPY1     JSR  ZSAVER
                      809   *
                      810   * Global Page in 64K-Karte
                      811   * duplizieren
                      812   *
```

```
0E67: 8D 05 C0  813           STA  $C005       ;WRAUX
0E6A: A2 00     814           LDX  #0
0E6C: BD 00 BF  815 MLICOPY   LDA  MLI,X
0E6F: 9D 00 BF  816           STA  MLI,X
0E72: E8        817           INX
0E73: D0 F7     818           BNE  MLICOPY
0E75: 8D 04 C0  819           STA  $C004       ;WRMAIN
                820 *
                821 * Mehr als 280 Blocks zulassen
                822 *
0E78: 20 2C 0F  823           JSR  PATCH       ;Spuren
                824 *
                825 * Volume-Nummer und Spur-Anzahl
                826 * für Format poken
                827 *
0E7B: AD B5 08  828           LDA  VOLUME
0E7E: 8D 6F 09  829           STA  VOLNR+1
0E81: AD B7 08  830           LDA  SPUREN1
0E84: 8D F4 09  831           STA  SPURNR+1
                832 *
                833 * Slot/Drive in Unit-Number
                834 * packen
                835 *
0E87: AD B1 08  836           LDA  OSLOT
0E8A: AE B2 08  837           LDX  ODRIVE
0E8D: 20 21 0F  838           JSR  SLOTDRIV
0E90: 8D BB 08  839           STA  OUNIT
0E93: AD B3 08  840           LDA  DSLOT
0E96: AE B4 08  841           LDX  DDRIVE
0E99: 20 21 0F  842           JSR  SLOTDRIV
0E9C: 8D BC 08  843           STA  DUNIT
                844 *
                845 *-------------------------------
                846 *
                847 * INIT-Routine aus PRODOS-FILER
                848 * der Firma Apple aufrufen:
                849 *
                850 * Entry mit A = Unit-Number
                851 * Exit  mit A = Fehler-Number
                852 *
                853 * Mit Block $0000 anfangen
                854 *
0E9F: A9 00     855           LDA  #0
0EA1: 8D A0 00  856           STA: RDBLOCK
0EA4: 8D A1 00  857           STA: RDBLOCK+1
0EA7: 8D AD 00  858           STA: WRBLOCK
0EAA: 8D AE 00  859           STA: WRBLOCK+1
                860 *
                861 * Block 0 von nicht-formatierter
                862 * Diskette lesen, um Lesekopf
                863 * zu justieren
                864 *
0EAD: A9 14     865           LDA  #>PUFFBEGM
0EAF: 8D 9F 00  866           STA: RDPUFF+1
0EB2: AD BC 08  867           LDA  DUNIT
0EB5: 8D 9D 00  868           STA: RDUNIT
0EB8: 20 95 00  869           JSR  READER
                870 *
                871 * Falls Lesefehler auf Duplikat,
                872 * dann stets INIT, andernfalls
                873 * INIT ja,   wenn INITFLAG = 1
                874 * INIT nein, wenn INITFLAG = 0
                875 *
0EBB: B0 05     876           BCS  INIT        ;Fehler
0EBD: AD B6 08  877           LDA  INITFLAG
0EC0: F0 2A     878           BEQ  COPY2
                879 *
                880 * Duplikat formatieren
                881 *
0EC2: AD BC 08  882 INIT      LDA  DUNIT
0EC5: 20 00 09  883           JSR  FORMAT
0EC8: 8D BA 08  884           STA  ERROR
0ECB: F0 1F     885           BEQ  COPY2       ;okay
                886 *
                887 * Fehler-Exit
                888 *
0ECD: 20 15 0F  889 EXIT2     JSR  ZLOADER
0ED0: 20 58 FC  890           JSR  HOME
0ED3: 20 2D FF  891           JSR  PRERR
0ED6: A9 BA     892           LDA  #":"
0ED8: 20 ED FD  893           JSR  PRINT
0EDB: AD BA 08  894           LDA  ERROR
0EDE: 20 DA FD  895           JSR  HEXOUT
0EE1: A9 A0     896           LDA  #$A0
0EE3: 20 ED FD  897           JSR  PRINT
0EE6: 20 0C FD  898           JSR  RDKEY
```

```
0EE9: 4C BD 08  899           JMP  START1      ;Exit
                900 *
                901 *-------------------------------
                902 *
                903 * Kopieren
                904 * ========
                905 *
                906 * MLI-Unit-Number poken
                907 *
0EEC: AD BB 08  908 COPY2     LDA  OUNIT
0EEF: 8D 9D 00  909           STA: RDUNIT
0EF2: AD BC 08  910           LDA  DUNIT
0EF5: 8D AA 00  911           STA: WRUNIT
                912 *
0EF8: 20 00 00  913           JSR  COPY3
0EFB: B0 D0     914           BCS  EXIT2
                915 *
                916 * Okay-Exit nach Kopie
                917 *
0EFD: 20 15 0F  918           JSR  ZLOADER
0F00: 4C BD 08  919           JMP  START1      ;Exit
                920 *
                921 *-------------------------------
                922 *
                923 * Zero-Page saven und loaden
                924 *
0F03: A2 00     925 ZSAVER    LDX  #0
0F05: BD 00 00  926 ZSAVE     LDA: $0000,X
0F08: 9D 3A 0D  927           STA  ZAREA,X
0F0B: BD 58 0F  928           LDA  MARKER+1,X
0F0E: 9D 00 00  929           STA: $0000,X
0F11: E8        930           INX
0F12: D0 F1     931           BNE  ZSAVE
0F14: 60        932           RTS
                933 *
0F15: A2 00     934 ZLOADER   LDX  #0
0F17: BD 3A 0D  935 ZLOAD     LDA  ZAREA,X
0F1A: 9D 00 00  936           STA: $0000,X
0F1D: E8        937           INX
0F1E: D0 F7     938           BNE  ZLOAD
0F20: 60        939           RTS
                940 *
                941 * Slot + Drive in Unit-Number
                942 *
0F21: 0A        943 SLOTDRIV  ASL
0F22: 0A        944           ASL
0F23: 0A        945           ASL
0F24: 0A        946           ASL
0F25: E0 01     947           CPX  #1          ;Drive 1
0F27: F0 02     948           BEQ  SLOTDR
0F29: 09 80     949           ORA  #%10000000
0F2B: 60        950 SLOTDR    RTS
                951 *
                952 * Spurprüfroutine patchen
                953 *
0F2C: AD 83 C0  954 PATCH     LDA  $C083
0F2F: AD 83 C0  955           LDA  $C083
                956 *
0F32: A9 18     957           LDA  #$18
0F34: 8D 09 F8  958           STA  $F809
0F37: A9 90     959           LDA  #$90
0F39: 8D 0A F8  960           STA  $F80A
0F3C: A9 04     961           LDA  #$04
0F3E: 8D 0B F8  962           STA  $F80B
                963 *
                964 * RAM-Disk abstellen
                965 *
0F41: A9 4C     966           LDA  #$4C
0F43: 8D 18 FF  967           STA  $FF18
0F46: A9 3B     968           LDA  #$3B
0F48: 8D 19 FF  969           STA  $FF19
0F4B: A9 FF     970           LDA  #$FF
0F4D: 8D 1A FF  971           STA  $FF1A
                972 *
0F50: AD 81 C0  973           LDA  $C081
0F53: AD 81 C0  974           LDA  $C081
0F56: 60        975           RTS
                976 *
0F57: EA        977 MARKER    NOP
                978 *
                979 *-------------------------------
                980 *
                981 * In 2 Schüben je 70K kopieren
                982 * $0800-$9A00 AUX  36.5K
                983 * $1400-$9A00 MAIN 33.5K
                984 *
```

```
                        985  * Kopierroutine in Zero-Page!
                        986  * --------------------------
                        987  *
                        988           ORG  $0000
                        989  *
                        990  * 2 mal 70K lesen und schreiben
                        991  *
0000: 20 0C 00          992  COPY3    JSR  COPY4
0003: B0 4B             993           BCS  MAINRAM     ;Fehler
0005: C6 0A             994           DEC  LAEUFE2
0007: D0 F7             995           BNE  COPY3
0009: 60                996           RTS
                        997  *
000A: 02                998  LAEUFE2  HEX  02          ;2 mal
000B: 9A                999  PUFFEND2 HEX  9A          ;$9A00
                        1000 *
                        1001 * 70K lesen
                        1002 *
000C: 20 5C 00          1003 COPY4    JSR  AUXRAM
000F: A9 08             1004          LDA  #>PUFFBEGA
0011: 20 63 00          1005          JSR  RDHUNK1
0014: B0 3A             1006          BCS  MAINRAM     ;Fehler
                        1007 *
0016: 20 50 00          1008          JSR  MAINRAM
0019: A9 14             1009          LDA  #>PUFFBEGM
001B: 20 63 00          1010          JSR  RDHUNK1
001E: B0 30             1011          BCS  MAINRAM     ;Fehler
                        1012 *
                        1013 * 70K schreiben
                        1014 *
0020: 20 5C 00          1015          JSR  AUXRAM
0023: A9 08             1016          LDA  #>PUFFBEGA
0025: 20 7C 00          1017          JSR  WRHUNK1
0028: B0 26             1018          BCS  MAINRAM     ;Fehler
                        1019 *
002A: 20 50 00          1020          JSR  MAINRAM
002D: A9 14             1021          LDA  #>PUFFBEGM
002F: 20 7C 00          1022          JSR  WRHUNK1
0032: B0 1C             1023          BCS  MAINRAM     ;Fehler
0034: 60                1024          RTS
                        1025 *
                        1026 * Füller für MLI-Zero-Page
                        1027 * MLI benutzt $003A-$004F
                        1028 *
0035: 00 00 00          1029          HEX  000000
0038: 00 00 00          1030          HEX  0000000000000000
0040: 00 00 00          1031          HEX  0000000000000000
0048: 00 00 00          1032          HEX  0000000000000000
                        1033 *
                        1034 * Zwischen unteren und oberen
                        1035 * 47.5K $0200-$BFFF umschalten
                        1036 *
0050: 8D 02 C0          1037 MAINRAM  STA  $C002       ;RDMAIN
0053: 8D 04 C0          1038          STA  $C004       ;WRMAIN
0056: 90 03             1039          BCC  MAINRAM1
0058: 8D BA 08          1040          STA  ERROR
005B: 60                1041 MAINRAM1 RTS
                        1042 *
005C: 8D 03 C0          1043 AUXRAM   STA  $C003       ;RDAUX
005F: 8D 05 C0          1044          STA  $C005       ;WRAUX
0062: 60                1045          RTS
                        1046 *
                        1047 * Read
                        1048 * ----
                        1049 *
0063: 85 9F             1050 RDHUNK1  STA  RDPUFF+1
0065: 20 95 00          1051 RDHUNK2  JSR  READER
0068: B0 11             1052          BCS  RDHUNK4
006A: E6 9F             1053          INC  RDPUFF+1
006C: E6 9F             1054          INC  RDPUFF+1
006E: E6 A0             1055          INC  RDBLOCK
0070: D0 02             1056          BNE  RDHUNK3
0072: E6 A1             1057          INC  RDBLOCK+1
0074: A5 9F             1058 RDHUNK3  LDA  RDPUFF+1
0076: C5 0B             1059          CMP  PUFFEND2
0078: 90 EB             1060          BCC  RDHUNK2
007A: 18                1061          CLC              ;C=0!
007B: 60                1062 RDHUNK4  RTS
                        1063 *
                        1064 * Write
                        1065 * -----
                        1066 *
007C: 85 AC             1067 WRHUNK1  STA  WRPUFF+1
007E: 20 A2 00          1068 WRHUNK2  JSR  WRITER
0081: B0 11             1069          BCS  WRHUNK4
0083: E6 AC             1070          INC  WRPUFF+1
0085: E6 AC             1071          INC  WRPUFF+1
0087: E6 AD             1072          INC  WRBLOCK
0089: D0 02             1073          BNE  WRHUNK3
008B: E6 AE             1074          INC  WRBLOCK+1
008D: A5 AC             1075 WRHUNK3  LDA  WRPUFF+1
008F: C5 0B             1076          CMP  PUFFEND2
0091: 90 EB             1077          BCC  WRHUNK2
0093: 18                1078          CLC              ;C=0!
0094: 60                1079 WRHUNK4  RTS
                        1080 *
                        1081 * Read-Block
                        1082 *
0095: 20 00 BF          1083 READER   JSR  MLI
0098: 80                1084 RDCOM    HEX  80          ;Read
0099: 9C 00             1085          DA   RDCOUNT
009B: 60                1086          RTS              ;C=0/1
                        1087 *
009C: 03                1088 RDCOUNT  HEX  03
009D: 60                1089 RDUNIT   HEX  60
009E: 00 16             1090 RDPUFF   HEX  0016        ;LLHH
00A0: 00 00             1091 RDBLOCK  HEX  0000        ;LLHH
                        1092 *
                        1093 * Write-Block
                        1094 *
00A2: 20 00 BF          1095 WRITER   JSR  MLI
00A5: 81                1096 WRCOM    HEX  81          ;Write
00A6: A9 00             1097          DA   WRCOUNT
00A8: 60                1098          RTS              ;C=0/1
                        1099 *
00A9: 03                1100 WRCOUNT  HEX  03
00AA: E0                1101 WRUNIT   HEX  E0
00AB: 00 16             1102 WRPUFF   HEX  0016        ;LLHH
00AD: 00 00             1103 WRBLOCK  HEX  0000        ;LLHH
```

1881 bytes

PRODOS.COPYA

```
100  ON  PEEK (3000) = 76 AND  PEEK (3001) = 198 AND
     PEEK (3002) = 11 GOTO 110: PRINT  CHR$ (4)"BLOAD
     PRODOS.COPYOBJ"
110  PRINT  CHR$ (21): HOME : INVERSE : HTAB 14: PRINT
     "PRODOS.COPYA": HTAB 14: PRINT "U. STIEHL 84":
     NORMAL : PRINT : PRINT
120 OS = 6:OD = 1:DS = 6:DD = 2:S = 35
130  IF OS = DS AND OD = DD THEN  PRINT "NUR FUER 2
     DRIVES!"; CHR$ (7): END
140  PRINT "ORIGINAL-SLOT: "OS;: HTAB 22: PRINT
     "DUPLIKAT-SLOT: "DS: PRINT "ORIGINAL-DRIVE:"OD;:
     HTAB 22: PRINT "DUPLIKAT-DRIVE:"DD
150  PRINT : HTAB 13: PRINT "SPURENANZAHL:"S
160  PRINT : HTAB 14: PRINT "START J/N ";
170  GET X$: ON X$ = "J" OR X$ = "j" GOTO 180: ON X$ <  >
     "N" AND X$ <  > "n" GOTO 170: HOME : END
180  VTAB 8:: HTAB 11: CALL  - 958: PRINT "DISKETTEN
     EINLEGEN": PRINT : HTAB 14: PRINT "W = WEITER ";
190  GET X$: IF X$ <  > "W" AND X$ <  > "w" GOTO 190
200 C = 3000: POKE C + 3,OS: POKE C + 4,OD: POKE C +
     5,DS: POKE C + 6,DD: POKE C + 7,S
210 B = S * 8: POKE C + 9, INT (B / 256): POKE C + 8,B -
     PEEK (C + 9) * 256
220  CALL C
230  IF  PEEK (C + 10) <  > 0 THEN  HOME : PRINT "FEHLER
     "; CHR$ (7);: GET X$
240  RUN 110
```

QUICKCOPY.PUFFER

```
100  INPUT "Spuren:";S
110 BG = 140: REM Blockanzahl
120 PE = 154: REM HH $9A00
130 B = S * 8
140 L = B / BG: REM L=Läufe
150  IF L =  INT (L) THEN 170
160 L =  INT (L) + 1
170 T = BG -  INT (B / L)
180 PE = 154 - T
190  HOME : PRINT "Spuren:"S
200  PRINT "Läufe:"L
210  PRINT "Pufferende:"PE
220 G = (PE * 256 - 5120
    + PE * 256 - 2048) * L
230  PRINT "Gesamt:"G:
     PRINT "Kilobyte:"G / 1024
240 BL = G / 512: IF B > BL
    THEN  GOTO 160
250  PRINT "Blocks:"BL
260  PRINT "Puffergröße:"G / L
```

QUICKCOPY Hex-Dump

BSAVE QUICKCOPY, A2222, L1881
Start mit BRUN QUICKCOPY
Neustart mit CALL 2222

```
$08A8: 00 00 00 00 00 00 4C BD
$08B0: 08 06 01 06 02 01 01 23
$08B8: 02 9A 00 60 E0 20 58 FC
$08C0: A2 00 BD D0 08 F0 06 20
$08C8: ED FD E8 D0 F5 4C 3A 0E
$08D0: 2A 11 15 09 03 0B 2D 03
$08D8: 0F 10 19 2A 8D 15 2E 20
$08E0: 13 14 09 05 08 0C 20 38
$08E8: 34 8D 8D D3 D4 C1 D2 D4
$08F0: A0 CA AF CE A0 00 00 00
$08F8: 00 00 00 00 00 00 00 00
$0900: 08 78 20 3A 09 28 C9 00
$0908: D0 02 18 60 C9 02 D0 05
$0910: A9 2B 4C 21 09 C9 01 D0
$0918: 05 A9 27 4C 21 09 18 69
$0920: 30 38 60 0A 0E 24 0D 8D
$0928: 36 0D 8A 4A 4A 4A 4A A8
$0930: AD 36 0D 20 C6 0A 4E 24
$0938: 0D 60 AA 29 70 8D 23 0D
$0940: 8A AE 23 0D 2A A9 00 2A
$0948: D0 06 BD 8A C0 4C 53 09
$0950: BD 8B C0 BD 89 C0 A9 D7
$0958: 85 DA A9 50 8D 24 0D A9
$0960: 00 20 23 09 A5 DA F0 06
$0968: 20 3A 0C 4C 64 09 A9 01
$0970: 85 D3 A9 AA 85 D0 AD 20
$0978: 0D 18 69 02 85 D4 A9 00
$0980: 85 D1 A5 D1 AE 23 0D 20
$0988: 23 09 AE 23 0D BD 8D C0
$0990: BD 8E C0 A8 BD 8E C0 BD
$0998: 8C C0 98 10 05 A9 02 4C
$09A0: F9 09 20 63 0C 90 0E A9
$09A8: 01 A4 D4 CC 1F 0D B0 02
$09B0: A9 04 4C F9 09 A4 D4 CC
$09B8: 1F 0D B0 05 A9 04 4C F9
$09C0: 09 CC 20 0D 90 05 A9 03
$09C8: 4C F9 09 AD 22 0D 8D 25
$09D0: 0D CE 25 0D D0 05 A9 01
$09D8: 4C F9 09 AE 23 0D 20 6A
$09E0: 0A B0 EE A5 D8 D0 EA AE
$09E8: 23 0D 20 07 0A B0 E2 E6
$09F0: D1 A5 D1 C9 23 90 8B A9
$09F8: 00 48 AE 23 0D BD 88 C0
$0A00: A9 00 20 23 09 68 60 A0
$0A08: 20 88 F0 5C BD 8C C0 10
$0A10: FB 49 D5 D0 F4 EA BD 8C
$0A18: C0 10 FB C9 AA D0 F2 A0
$0A20: 56 BD 8C C0 10 FB C9 AD
$0A28: D0 E7 A9 00 88 84 D5 BD
$0A30: 8C C0 10 FB C9 96 D0 30
$0A38: A4 D5 D0 F0 84 D5 BD 8C
$0A40: C0 10 FB C9 96 D0 21 A4
$0A48: D5 C8 D0 F0 BD 8C C0 10
$0A50: FB C9 96 D0 13 BD 8C C0
$0A58: 10 FB C9 DE D0 0A EA BD
$0A60: 8C C0 10 FB C9 AA F0 5C
$0A68: 38 60 A0 FC 84 DC C8 D0
$0A70: 04 E6 DC F0 F3 BD 8C C0
$0A78: 10 FB C9 D5 D0 F0 EA BD
$0A80: 8C C0 10 FB C9 AA D0 F2
$0A88: A0 03 BD 8C C0 10 FB C9
$0A90: 96 D0 E7 A9 00 85 DB BD
$0A98: 8C C0 10 FB 2A 85 DD BD
$0AA0: 8C C0 10 FB 25 DD 99 D7
$0AA8: 00 45 DB 88 10 E7 A8 D0
$0AB0: B7 BD 8C C0 10 FB C9 DE
$0AB8: D0 AE EA BD 8C C0 10 FB
$0AC0: C9 AA D0 A4 18 60 8E 37
$0AC8: 0D 8D 36 0D CD 24 0D F0
$0AD0: 5C A9 00 8D 38 0D AD 24
$0AD8: 0D 8D 39 0D 38 ED 36 0D
$0AE0: F0 37 B0 07 49 FF EE 24
$0AE8: 0D 90 05 69 FE CE 24 0D
$0AF0: CD 38 0D 90 03 AD 38 0D
$0AF8: C9 0C B0 01 A8 38 20 1D
$0B00: 0B B9 4B 0C 20 3A 0C AD
$0B08: 39 0D 18 20 20 0B B9 57
$0B10: 0C 20 3A 0C EE 38 0D D0
$0B18: BD 20 3A 0C 18 AD 24 0D
$0B20: 29 03 2A 0D 37 0D AA BD
$0B28: 80 C0 AE 37 0D 60 20 0E
$0B30: 0D BD 8D C0 BD 8E C0 A9
$0B38: FF 9D 8F C0 DD 8C C0 48
$0B40: 68 EA A0 04 48 68 20 A5
$0B48: 0B 88 D0 F8 A9 D5 20 A4
$0B50: 0B A9 AA 20 A4 0B A9 AD
$0B58: 20 A4 0B A0 56 EA EA EA
$0B60: D0 03 20 0E 0D EA EA A9
$0B68: 96 9D 8D C0 DD 8C C0 88
$0B70: D0 F0 24 00 EA 20 0E 0D
$0B78: A9 96 9D 8D C0 DD 8C C0
$0B80: A9 96 EA C8 D0 EF 20 A4
$0B88: 0B A9 DE 20 A4 0B A9 AA
$0B90: 20 A4 0B A9 EB 20 A4 0B
$0B98: A9 FF 20 A4 0B BD 8E C0
$0BA0: BD 8C C0 60 EA 48 68 9D
$0BA8: 8D C0 DD 8C C0 60 38 BD
$0BB0: 8D C0 BD 8E C0 30 5E A9
$0BB8: FF 9D 8F C0 DD 8C C0 48
$0BC0: 68 20 1B 0C 20 1B 0C 9D
$0BC8: 8D C0 DD 8C C0 EA 88 D0
$0BD0: F0 A9 D5 20 2D 0C A9 AA
$0BD8: 20 2D 0C A9 96 20 2D 0C
$0BE0: A5 D3 20 1C 0C A5 D1 20
$0BE8: 1C 0C A5 D2 20 1C 0C A5
$0BF0: D3 45 D1 45 D2 48 4A 05
$0BF8: D0 9D 8D C0 BD 8C C0 68
$0C00: 09 AA 20 2C 0C A9 DE 20
$0C08: 2D 0C A9 AA 20 2D 0C A9
$0C10: EB 20 2D 0C 18 BD 8E C0
$0C18: BD 8C C0 60 48 4A 05 D0
$0C20: 9D 8D C0 DD 8C C0 68 EA
$0C28: EA EA 09 AA EA EA 48 68
$0C30: 9D 8D C0 DD 8C C0 60 00
$0C38: 00 00 A2 11 CA D0 FD E6
$0C40: D9 D0 02 E6 DA 38 E9 01
$0C48: D0 F0 60 01 30 28 24 20
$0C50: 1E 1D 1C 1C 1C 1C 1C 70
$0C58: 2C 26 22 1F 1E 1D 1C 1C
$0C60: 1C 1C 1C AD 21 0D 85 D6
$0C68: A0 80 A9 00 85 D2 4C 73
$0C70: 0C A4 D4 AE 23 0D 20 AE
$0C78: 0B 90 03 4C 0E 0D AE 23
$0C80: 0D 20 2E 0B E6 D2 A5 D2
$0C88: C9 10 90 E5 A0 0F 84 D2
$0C90: AD 22 0D 8D 25 0D 99 26
$0C98: 0D 88 10 FA A5 D4 38 E9
$0CA0: 05 A8 20 0E 0D 20 0E 0D
$0CA8: 48 68 EA EA 88 D0 F3 AE
$0CB0: 23 0D 20 6A 0A B0 3C A5
$0CB8: D8 F0 13 C6 D4 A5 D4 CD
$0CC0: 1F 0D B0 2F 38 60 AE 23
$0CC8: 0D 20 6A 0A B0 1A AE 23
$0CD0: 0D 20 07 0A B0 12 A4 D8
$0CD8: B9 26 0D 30 0B A9 FF 99
$0CE0: 26 0D C6 D2 10 E0 18 60
$0CE8: CE 25 0D D0 D9 C6 D6 D0
$0CF0: 02 38 60 AD 22 0D 0A 8D
$0CF8: 25 0D AE 23 0D 20 6A 0A
$0D00: B0 06 A5 D8 C9 0F F0 07
$0D08: CE 25 0D D0 ED 38 60 A2
$0D10: D6 20 0E 0D 20 0E 0D 24
$0D18: 00 CA D0 F5 4C 68 0C 10
$0D20: 1B 03 10 60 80 00 00 00
$0D28: 00 00 00 00 00 00 00 00
$0D30: 00 00 00 00 00 00 00 60
$0D38: A0 B1 0C 00 00 00 00 00
$0D40: 00 00 00 00 00 00 00 00
$0D48: 00 00 00 00 00 00 00 00
$0D50: 00 00 00 00 00 00 00 00
$0D58: 00 00 00 00 00 00 00 00
$0D60: 00 00 00 00 00 00 00 00
$0D68: 00 00 00 00 00 00 00 00
$0D70: 00 00 00 00 00 00 00 00
$0D78: 00 00 00 00 00 00 00 00
$0D80: 00 00 00 00 00 00 00 00
$0D88: 00 00 00 00 00 00 00 00
$0D90: 00 00 00 00 00 00 00 00
$0D98: 00 00 00 00 00 00 00 00
$0DA0: 00 00 00 00 00 00 00 00
$0DA8: 00 00 00 00 00 00 00 00
$0DB0: 00 00 00 00 00 00 00 00
$0DB8: 00 00 00 00 00 00 00 00
$0DC0: 00 00 00 00 00 00 00 00
$0DC8: 00 00 00 00 00 00 00 00
$0DD0: 00 00 00 00 00 00 00 00
$0DD8: 00 00 00 00 00 00 00 00
$0DE0: 00 00 00 00 00 00 00 00
$0DE8: 00 00 00 00 00 00 00 00
$0DF0: 00 00 00 00 00 00 00 00
$0DF8: 00 00 00 00 00 00 00 00
$0E00: 00 00 00 00 00 00 00 00
$0E08: 00 00 00 00 00 00 00 00
$0E10: 00 00 00 00 00 00 00 00
$0E18: 00 00 00 00 00 00 00 00
$0E20: 00 00 00 00 00 00 00 00
$0E28: 00 00 00 00 00 00 00 00
$0E30: 00 00 00 00 00 00 00 00
$0E38: 00 00 2C 10 C0 A9 00 8D
$0E40: BA 08 AD B8 08 8D 0A 00
$0E48: AD B9 08 8D 0B 00 20 0C
$0E50: FD C9 CA F0 0F C9 EA F0
$0E58: 0B C9 CE F0 06 C9 EE F0
$0E60: 02 D0 EB 60 20 03 0F 8D
$0E68: 05 C0 A2 00 BD 00 BF 9D
$0E70: 00 BF E8 D0 F7 8D 04 C0
$0E78: 20 2C 0F AD B5 08 8D 6F
$0E80: 09 AD B7 08 8D F4 09 AD
$0E88: B1 08 AE B2 08 20 21 0F
$0E90: 8D BB 08 AD B3 08 AE B4
$0E98: 08 20 21 0F 8D BC 08 A9
$0EA0: 00 8D A0 00 8D A1 00 8D
$0EA8: AD 00 8D AE 00 A9 14 8D
$0EB0: 9F 00 AD BC 08 8D 9D 00
$0EB8: 20 95 00 B0 05 AD B6 08
$0EC0: F0 2A AD BC 08 20 00 09
$0EC8: 8D BA 08 F0 1F 20 15 0F
$0ED0: 20 58 FC 20 2D FF A9 BA
$0ED8: 20 ED FD AD BA 08 20 DA
$0EE0: FD A9 A0 20 ED FD 20 0C
$0EE8: FD 4C BD 08 AD BB 08 8D
$0EF0: 9D 00 AD BC 08 8D AA 00
$0EF8: 20 00 00 B0 D0 20 15 0F
$0F00: 4C BD 08 A2 00 BD 00 00
$0F08: 9D 3A 0D BD 58 0F 9D 00
$0F10: 00 E8 D0 F1 60 A2 00 BD
$0F18: 3A 0D 9D 00 00 E8 D0 F7
$0F20: 60 0A 0A 0A 0A E0 01 F0
$0F28: 02 09 80 60 AD 83 C0 AD
$0F30: 83 C0 A9 18 8D 09 F8 A9
$0F38: 90 8D 0A F8 A9 04 8D 0B
$0F40: F8 A9 4C 8D 18 FF A9 3B
$0F48: 8D 19 FF A9 FF 8D 1A FF
$0F50: AD 81 C0 AD 81 C0 60 EA
$0F58: 20 0C 00 B0 4B C6 0A D0
$0F60: F7 60 02 9A 20 5C 00 A9
$0F68: 08 20 63 00 B0 3A 20 50
$0F70: 00 A9 14 20 63 00 B0 30
$0F78: 20 5C 00 A9 08 20 7C 00
$0F80: B0 26 20 50 00 A9 14 20
$0F88: 7C 00 B0 1C 60 00 00 00
$0F90: 00 00 00 00 00 00 00 00
$0F98: 00 00 00 00 00 00 00 00
$0FA0: 00 00 00 00 00 00 00 00
$0FA8: 8D 02 C0 8D 04 C0 90 03
$0FB0: 8D BA 08 60 8D 03 C0 8D
$0FB8: 05 C0 60 85 9F 20 95 00
$0FC0: B0 11 E6 9F E6 9F E6 A0
$0FC8: D0 02 E6 A1 A5 9F C5 0B
$0FD0: 90 EB 18 60 85 AC 20 A2
$0FD8: 00 B0 11 E6 AC E6 AC E6
$0FE0: AD D0 02 E6 AE A5 AC C5
$0FE8: 0B 90 EB 18 60 20 00 BF
$0FF0: 80 9C 00 60 03 60 00 16
$0FF8: 00 00 20 00 BF 81 A9 00
$1000: 60 03 E0 00 16 00 00 00
```

PRODOS.COPYOBJ Hex-Dump

BSAVE PRODOS.COPYOBJ, A3000, L1502
Start mit BRUN PRODOS.COPYOBJ
Neustart mit CALL 3000

```
$0BB8: 4C C6 0B 06 01 06 02 23
$0BC0: 18 01 00 01 60 E0 20 58
$0BC8: FC A2 00 BD D9 0B F0 06
$0BD0: 20 ED FD E8 D0 F5 4C 4A
$0BD8: 10 10 12 0F 04 0F 13 2E
$0BE0: 03 0F 10 19 01 8D 15 2E
$0BE8: 20 13 14 09 05 08 0C 20
$0BF0: 38 34 8D 8D D3 D4 C1 D2
$0BF8: D4 A0 CA AF CE A0 00 00
$0C00: 08 78 20 3A 0C 28 C9 00
$0C08: D0 02 18 60 C9 02 D0 05
$0C10: A9 2B 4C 21 0C C9 01 D0
$0C18: 05 A9 27 4C 21 0C 18 69
$0C20: 30 38 60 0A 0E 24 10 8D
$0C28: 36 10 8A 4A 4A 4A 4A A8
$0C30: AD 36 10 20 C6 0D 4E 24
$0C38: 10 60 AA 29 70 8D 23 10
$0C40: 8A AE 23 10 2A A9 00 2A
$0C48: D0 06 BD 8A C0 4C 53 0C
$0C50: BD 8B C0 BD 89 C0 A9 D7
$0C58: 85 DA A9 50 8D 24 10 A9
$0C60: 00 20 23 0C A5 DA F0 06
$0C68: 20 3A 0F 4C 64 0C A9 01
$0C70: 85 D3 A9 AA 85 D0 AD 20
$0C78: 10 18 69 02 85 D4 A9 00
$0C80: 85 D1 A5 D1 AE 23 10 20
$0C88: 23 0C AE 23 10 BD 8D C0
$0C90: BD 8E C0 A8 BD 8E C0 BD
$0C98: 8C C0 98 10 05 A9 02 4C
$0CA0: F9 0C 20 63 0F 90 0E A9
$0CA8: 01 A4 D4 CC 1F 10 B0 02
$0CB0: A9 04 4C F9 0C A4 D4 CC
$0CB8: 1F 10 B0 05 A9 04 4C F9
$0CC0: 0C CC 20 10 90 05 A9 03
$0CC8: 4C F9 0C AD 22 10 8D 25
$0CD0: 10 CE 25 10 D0 05 A9 01
$0CD8: 4C F9 0C AE 23 10 20 6A
$0CE0: 0D B0 EE A5 D8 D0 EA AE
$0CE8: 23 10 20 07 0D B0 E2 E6
$0CF0: D1 A5 D1 C9 23 90 8B A9
$0CF8: 00 48 AE 23 10 BD 88 C0
$0D00: A9 00 20 23 0C 68 60 A0
$0D08: 20 88 F0 5C BD 8C C0 10
$0D10: FB 49 D5 D0 F4 EA BD 8C
$0D18: C0 10 FB C9 AA D0 F2 A0
$0D20: 56 BD 8C C0 10 FB C9 AD
$0D28: D0 E7 A9 00 88 84 D5 BD
$0D30: 8C C0 10 FB C9 96 D0 30
$0D38: A4 D5 D0 F0 84 D5 BD 8C
$0D40: C0 10 FB C9 96 D0 21 A4
$0D48: D5 C8 D0 F0 BD 8C C0 10
$0D50: FB C9 96 D0 13 BD 8C C0
$0D58: 10 FB C9 DE D0 0A EA BD
$0D60: 8C C0 10 FB C9 AA F0 5C
$0D68: 38 60 A0 FC 84 DC C8 D0
$0D70: 04 E6 DC F0 F3 BD 8C C0
$0D78: 10 FB C9 D5 D0 F0 EA BD
$0D80: 8C C0 10 FB C9 AA D0 F2
$0D88: A0 03 BD 8C C0 10 FB C9
$0D90: 96 D0 E7 A9 00 85 DB BD
$0D98: 8C C0 10 FB 2A 85 DD BD
$0DA0: 8C C0 10 FB 25 DD 99 D7
$0DA8: 00 45 DB 88 10 E7 A8 D0
$0DB0: B7 BD 8C C0 10 FB C9 DE
$0DB8: D0 AE EA BD 8C C0 10 FB
$0DC0: C9 AA D0 A4 18 60 8E 37
$0DC8: 10 8D 36 10 CD 24 10 F0
$0DD0: 5C A9 00 8D 38 10 AD 24
$0DD8: 10 8D 39 10 38 ED 36 10
$0DE0: F0 37 B0 07 49 FF EE 24
$0DE8: 10 90 05 69 FE CE 24 10
$0DF0: CD 38 10 90 03 AD 38 10
$0DF8: C9 0C B0 01 A8 38 20 1D
$0E00: 0E B9 4B 0F 20 3A 0F AD
$0E08: 39 10 18 20 20 0E B9 57
$0E10: 0F 20 3A 0F EE 38 10 D0
$0E18: BD 20 3A 0F 18 AD 24 10
$0E20: 29 03 2A 0D 37 10 AA BD
$0E28: 80 C0 AE 37 10 60 20 0E
$0E30: 10 BD 8D C0 BD 8E C0 A9
$0E38: FF 9D 8F C0 DD 8C C0 48
$0E40: 68 EA A0 04 48 68 20 A5
$0E48: 0E 88 D0 F8 A9 D5 20 A4
$0E50: 0E A9 AA 20 A4 0E A9 AD
$0E58: 20 A4 0E A0 56 EA EA EA
$0E60: D0 03 20 0E 10 EA EA A9
$0E68: 96 9D 8D C0 DD 8C C0 88
$0E70: D0 F0 24 00 EA 20 0E 10
$0E78: A9 96 9D 8D C0 DD 8C C0
$0E80: A9 96 EA C8 D0 EF 20 A4
$0E88: 0E A9 DE 20 A4 0E A9 AA
$0E90: 20 A4 0E A9 EB 20 A4 0E
$0E98: A9 FF 20 A4 0E BD 8E C0
$0EA0: BD 8C C0 60 EA 48 68 9D
$0EA8: 8D C0 DD 8C C0 60 38 BD
$0EB0: 8D C0 BD 8E C0 30 5E A9
$0EB8: FF 9D 8F C0 DD 8C C0 48
$0EC0: 68 20 1B 0F 20 1B 0F 9D
$0EC8: 8D C0 DD 8C C0 EA 88 D0
$0ED0: F0 A9 D5 20 2D 0F A9 AA
$0ED8: 20 2D 0F A9 96 20 2D 0F
$0EE0: A5 D3 20 1C 0F A5 D1 20
$0EE8: 1C 0F A5 D2 20 1C 0F A5
$0EF0: D3 45 D1 45 D2 48 4A 05
$0EF8: D0 9D 8D C0 BD 8C C0 68
$0F00: 09 AA 20 2C 0F A9 DE 20
$0F08: 2D 0F A9 AA 20 2D 0F A9
$0F10: EB 20 2D 0F 18 BD 8E C0
$0F18: BD 8C C0 60 48 4A 05 D0
$0F20: 9D 8D C0 DD 8C C0 68 EA
$0F28: EA EA 09 AA EA EA 48 68
$0F30: 9D 8D C0 DD 8C C0 60 00
$0F38: 00 00 A2 11 CA D0 FD E6
$0F40: D9 D0 02 E6 DA 38 E9 01
$0F48: D0 F0 60 01 30 28 24 20
$0F50: 1E 1D 1C 1C 1C 1C 1C 70
$0F58: 2C 26 22 1F 1E 1D 1C 1C
$0F60: 1C 1C 1C AD 21 10 85 D6
$0F68: A0 80 A9 00 85 D2 4C 73
$0F70: 0F A4 D4 AE 23 10 20 AE
$0F78: 0E 90 03 4C 0E 10 AE 23
$0F80: 10 20 2E 0E E6 D2 A5 D2
$0F88: C9 10 90 E5 A0 0F 84 D2
$0F90: AD 22 10 8D 25 10 99 26
$0F98: 10 88 10 FA A5 D4 38 E9
$0FA0: 05 A8 20 0E 10 20 0E 10
$0FA8: 48 68 EA EA 88 D0 F3 AE
$0FB0: 23 10 20 6A 0D B0 3C A5
$0FB8: D8 F0 13 C6 D4 A5 D4 CD
$0FC0: 1F 10 B0 2F 38 60 AE 23
$0FC8: 10 20 6A 0D B0 1A AE 23
$0FD0: 10 20 07 0D B0 12 A4 D8
$0FD8: B9 26 10 30 0B A9 FF 99
$0FE0: 26 10 C6 D2 10 E0 18 60
$0FE8: CE 25 10 D0 D9 C6 D6 D0
$0FF0: 02 38 60 AD 22 10 0A 8D
$0FF8: 25 10 AE 23 10 20 6A 0D
$1000: B0 06 A5 D8 C9 0F F0 07
$1008: CE 25 10 D0 ED 38 60 A2
$1010: D6 20 0E 10 20 0E 10 24
$1018: 00 CA D0 F5 4C 68 0F 10
$1020: 1B 03 10 60 80 00 00 00
$1028: 00 00 00 00 00 00 00 00
$1030: 00 00 00 00 00 00 00 60
$1038: A0 B1 00 00 00 00 00 00
$1040: 00 00 00 00 00 00 00 00
$1048: 00 00 2C 10 C0 A9 00 8D
$1050: C2 0B 20 0C FD C9 CA F0
$1058: 0F C9 EA F0 0B C9 CE F0
$1060: 06 C9 EE F0 02 D0 EB 60
$1068: AD C3 0B 8D 6F 0C AD BF
$1070: 0B 8D F4 0C 20 56 11 AD
$1078: BB 0B AE BC 0B 20 4B 11
$1080: 8D C4 0B AD BD 0B AE BE
$1088: 0B 20 4B 11 8D C5 0B A9
$1090: 00 8D 82 11 8D 83 11 8D
$1098: 94 11 8D 95 11 A9 16 8D
$10A0: 81 11 AD C5 0B 8D 7F 11
$10A8: 20 72 11 20 35 11 AD C5
$10B0: 0B 20 00 0C 8D C2 0B 20
$10B8: 40 11 AD C2 0B D0 0F AD
$10C0: C4 0B 8D 7F 11 AD C5 0B
$10C8: 8D 91 11 4C CF 10 60 A9
$10D0: 16 8D 81 11 20 72 11 B0
$10D8: F5 EE 81 11 EE 81 11 EE
$10E0: 82 11 D0 03 EE 83 11 38
$10E8: AD C0 0B E9 01 ED 82 11
$10F0: AD C1 0B ED 83 11 90 07
$10F8: AD 81 11 C9 96 90 D5 A9
$1100: 16 8D 93 11 20 84 11 B0
$1108: C5 EE 93 11 EE 93 11 EE
$1110: 94 11 D0 03 EE 95 11 38
$1118: AD C0 0B E9 01 ED 94 11
$1120: AD C1 0B ED 95 11 90 0A
$1128: AD 93 11 C9 96 90 D5 4C
$1130: CF 10 4C C6 0B A2 0F B5
$1138: D0 9D 3A 10 CA 10 F8 60
$1140: A2 0F BD 3A 10 95 D0 CA
$1148: 10 F8 60 0A 0A 0A 0A E0
$1150: 01 F0 02 09 80 60 AD 83
$1158: C0 AD 83 C0 A9 18 8D 09
$1160: F8 A9 90 8D 0A F8 A9 04
$1168: 8D 0B F8 AD 81 C0 AD 81
$1170: C0 60 20 00 BF 80 7E 11
$1178: 90 03 8D C2 0B 60 03 60
$1180: 00 16 00 00 20 00 BF 81
$1188: 90 11 90 03 8D C2 0B 60
$1190: 03 E0 00 16 00 00 00 00
```

Softbreaker 1.0

Eine softwaremäßige Interrupt-Utility für die Apple IIe 64K-Karte

von U. Stiehl

1984, Diskette und Manual, DM 48,–
ISBN 3-7785-1022-3

Softbreaker ist ein Assemblerprogramm, mit dessen Hilfe Programme, die sich von der 64K-Karte (= Extended 80 Column Card für den Apple IIe) starten lassen, unterbrochen, gespeichert, geladen und exakt an der Stelle der Unterbrechung fortgeführt werden können. Dadurch ist es auch möglich, Sicherungskopien von sogenannten kopiergeschützten Programmen herzustellen.

Mit Softbreaker unterbrochene Programme werden komplett, d. h. die ganzen 64K einschließlich Language Card, in nur ca. 11 Sekunden auf einer formatierten Diskette gesichert.

Gerätevoraussetzung: Apple IIe mit 64K-Karte

**Hüthig Software Service,
Postfach 10 28 69, D-6900 Heidelberg**

ProDOS und „Kompatible“

Ein allgemeines Patch-Programm

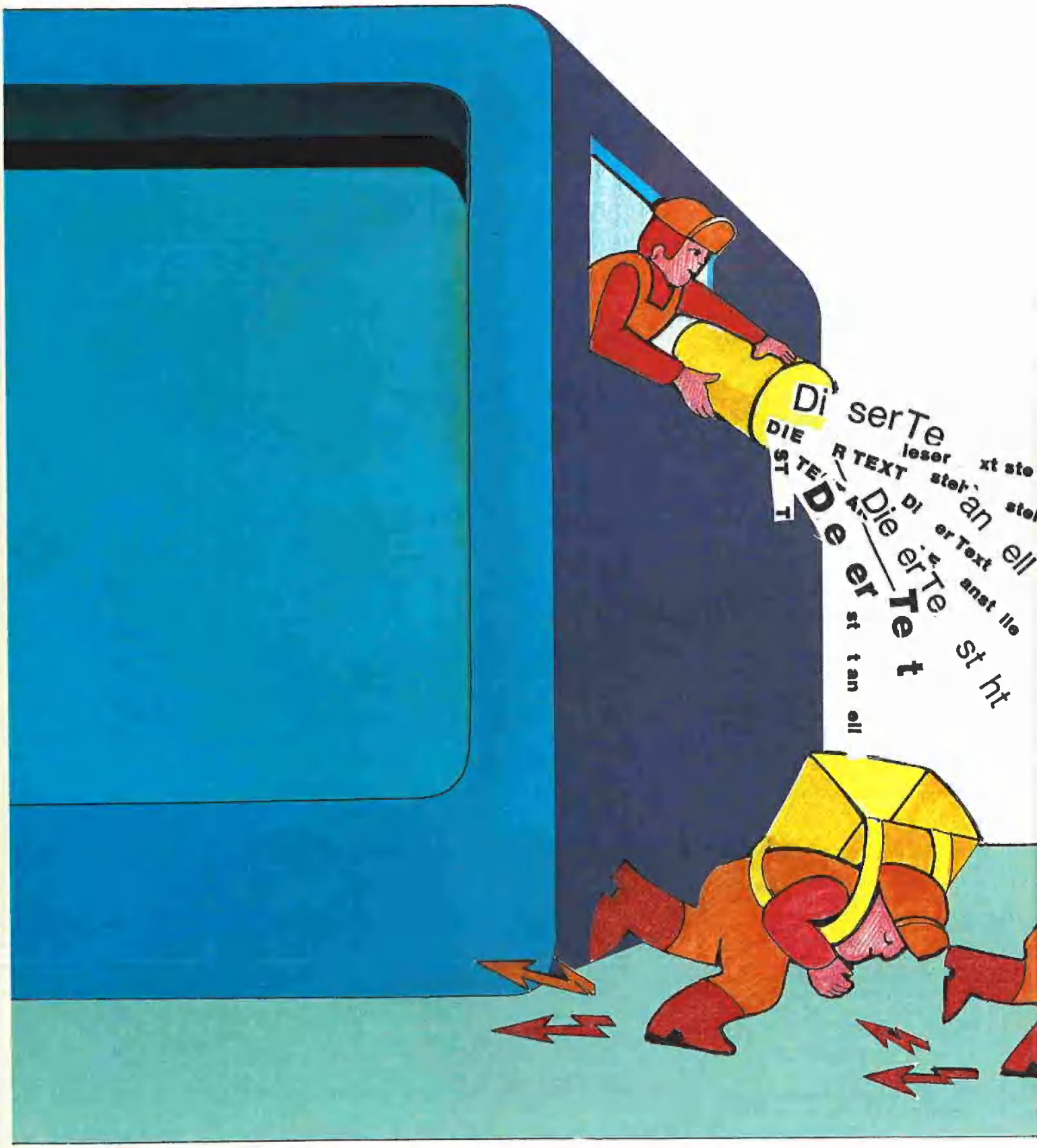
von Arne Schäpers

Mit Hilfe der nachfolgenden Patch-Utility PRODOS.PATCH dürfte man ProDOS auf den meisten Kompatiblen zum Laufen bringen. Da der (neugeschriebene) Kommentar zu diesem Programm erst nach Redaktionsschluß einging, haben wir uns entschlossen, in diesem Heft entgegen unseren sonstigen Gepflogenheiten nur das nackte Programm abzudrucken, da bereits viele Besitzer auf eine solche Utility, die über den F8-Kleinschreibumrüstsatz-Patch für Apple-II-Plus-Geräte aus „Peeker“, Heft 1/84 hinausgeht, gewartet haben.

Hinweis zum Programm

1. Speichern Sie die Utility PRODOS.-PATCH, nachdem Sie diese eingegeben haben, auf einer DOS-Diskette.
2. Erstellen Sie jetzt eine exakte Kopie der Original-USERS.DISK – ProDOS Version 1.0.1 vom 1. Jan. 84 – mit Hilfe eines normalen DOS-Kopierprogramms, z.B. COPYA.
3. Dann starten Sie das Programm PRODOS.PATCH von der DOS-Diskette. Wenn das Menü erscheint, legen Sie die Kopie der USERS.DISK in dasselbe Laufwerk ein und drücken Sie nun eine beliebige Taste. Der Patch-Vorgang ist nach wenigen Sekunden abgeschlossen.
4. Booten Sie nunmehr neu mit PR#6 o.ä.

```
PRODOS.PATCH

 10 HIMEM: 8192: TEXT : HOME
 20 REM
 30 REM RWTS-Aufruf; setzt 767 <> 0 bei Fehler
 40 FOR X = 0 TO 9: READ A: POKE 768 + X,A: NEXT
 50 DATA 32,227,3,32,217,3,110,255,2,96
 60 REM
 70 IOB = 256 * PEEK (987) + 14 * 16 + 8: REM $B7E8 für
    48/64K
 80 BUFFER = 8192: REM $2000
 90 REM
100 PRINT "Bitte 1:1 kopierte ProDOS 'USERS.DISK'"
105 PRINT
110 PRINT "in den Drive, von dem dieses Programm"
115 PRINT
120 PRINT "aus gestartet wurde, einlegen"
125 PRINT
130 PRINT "und <Return> drücken: ";
135 INPUT "";X$: PRINT
199 REM TEST FBC0
200 IF PEEK (15 * 256 * 16 + 11 * 256 + 12 * 16) > 3
    THEN 300
205 PRINT " - Patch auf $2055: JMP $2076": PRINT
210 TRACK = 0:SECTR = 1:CMD = 1: GOSUB 800
220 IF PEEK (BUF) < > 165 OR PEEK (BUF + 1) < > 67
    THEN 900
230 POKE BUF + 85,76: POKE BUF + 86,118: POKE BUF +
    87,32
240 CMD = 2: GOSUB 800: REM Zurückschreiben
299 REM TEST FBB3
300 IF PEEK (15 * 256 * 16 + 11 * 256 + 11 * 16 + 3) =
    14 * 16 + 10 THEN 400
305 PRINT " - Patch auf $23C3: BNE $23DA"
310 TRACK = 1:SECTR = 12:CMD = 1: GOSUB 800
320 IF PEEK (BUF) < > 168 OR PEEK (BUF + 1) < > 7
    THEN 900
330 POKE BUF + 195,208: POKE BUF + 196,21
340 CMD = 2: GOSUB 800: REM Zurückschreiben
399 REM PATCH "Apple II"
400 REM immer ausgeführt
405 PRINT " - PATCH AUF $265B: SEC / NOP"
410 TRACK = 1:SECTR = 9:CMD = 1: GOSUB 800
420 IF PEEK (BUF) < > 169 OR PEEK (BUF + 1) < > 9
    GOTO 900
430 POKE BUF + 91,56: POKE BUF + 92,234
440 CMD = 2: GOSUB 800: REM zurück
499 REM Test CXFF
500 SL = PEEK (IOB + 1) / 16
510 CXLOC = 12 * 16 * 256 + 256 * SL + 255
520 IF PEEK (CX) = 0 GOTO 700
530 PRINT " - Patch im Boot: $081D: LDA #$00"
540 TR = 0:SEC = 0:CMD = 1: GOSUB 800
550 IF PEEK (BUF) < > 1 OR PEEK (BUF + 1) < > 56
    GOTO 900
560 POKE BUF + 29,169: POKE BUF + 30,0
570 CMD = 1: GOSUB 800: REM zurück
580 REM
605 PRINT " - Patch im Relocator: $2507  LDA #$00"
610 TRACK = 1:SECTR = 10:CMD = 1: GOSUB 800
620 IF PEEK (BUF) < > 56 OR PEEK (BUF + 1) < > 208
    GOTO 900
630 POKE BUF + 7,169: POKE BUF + 8,0
640 CMD = 2: GOSUB 800
650 REM
700 PRINT : PRINT "Modifikationen beendet."
710 END
799 REM CALL RWTS: Daten auf $2000
800 POKE IOB + 3,0: REM Volume Number
805 POKE IOB + 8,0: POKE IOB + 9,BUF / 256
810 POKE IOB + 4,TRACK: POKE IOB + 5,SECTR: POKE IOB +
    12,CMD
820 POKE 767,0: CALL 768
830 IF PEEK (767) = 0 THEN RETURN
840 ERR = PEEK (IOB + 13)
842 PRINT : PRINT CHR$ (7)"Fehler: ";
844 IF ERR < > 16 THEN PRINT "I/O"
846 IF ERR = 16 THEN PRINT "Write protected"
860 END
899 REM ERR Exit für Datenprüfung
900 PRINT : PRINT CHR$ (7)"Fehler: nicht die erwarteten
    Daten"
905 PRINT
910 PRINT "         auf der Diskette."
920 END
```


Di serTe
leser xt ste
R TEXT stel an ell
Die erTe st ht
De er Te t
st t an ell

Der FRE-Befehl ist die langsamste Funktion des Applesoft-Interpreters mit einer Ausführungszeit von oft mehreren Minuten. Dem disassemblierten Listing dieser sog. Garbage-Collection, von Microsoft „String Space Compaction" genannt, (eigentlich: Müllabfuhr; in Basic: Beseitigung von nicht mehr benötigten Strings) wird eine auf dem BASIC.SYSTEM-FRE-Befehl basierende DOS-3.3-FRE-Version gegenübergestellt, die alle Rekorde schlägt und deshalb sprichwörtlich den String-Müll wie ein Blitz beseitigt.

Harald Grumser

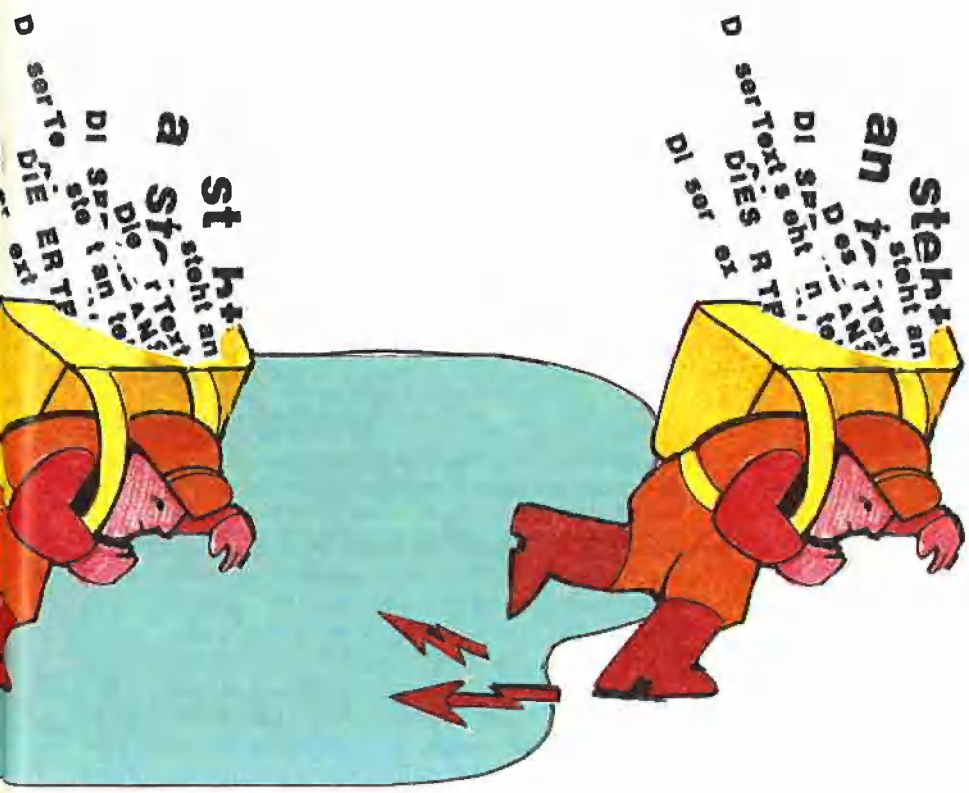
Müllabfuhr wie ein Blitz

Eine ultraschnelle Garbage-Collection-Routine

1. Statische und dynamische Stringverwaltung

Fast alle problemorientierten Programmiersprachen verfügen über eine mehr oder weniger große Anzahl von Befehlen zur Verwaltung von Strings (= Zeichenketten). Hinsichtlich der Verwaltung von Stringvariablen lassen sich zwei Arten unterscheiden: statische Verwaltung von Strings mit konstanter Länge und dynamische Verwaltung von Strings mit variabler Länge. Im folgenden soll kurz auf die Unterschiede dieser Verfahren eingegangen und ein Aspekt der dynamischen Stringverwaltung, die Garbage-Collection (G/C), näher beleuchtet werden.

In den meisten Programmiersprachen wird zur Verwaltung von Stringvariablen ein statisches Verfahren angewandt. Dabei muß der Benutzer jeden String zunächst bezüglich seiner Länge definieren, was normalerweise am Anfang des Programms geschieht. Der Rechner ist dann in der Lage, jeder Stringvariablen einen ihrer Länge entsprechenden Speicherbereich zuzuordnen. In der Variablenreferenzliste (Variablenliste, Variablentabelle, VARTAB) wird dann statt des eigentlichen Strings dessen Adresse im Speicher und seine Länge abgelegt. Erfolgt nun eine Zuweisung, so wird die Zeichenkette in den entsprechenden Stringbereich übertragen und die Länge im Deskriptor (= Stringbeschreiber) der Variablenliste aktualisiert. Hier zeigt sich ein entscheidender Nachteil dieses Verfahrens. Abgesehen vom Mehraufwand des Programmierers, der jede einzelne Stringvariable deklarieren muß, wird die Stringlänge ein einziges Mal festgelegt und kann danach nicht mehr geändert werden. Um nun allen Eventualitäten während des Programmlaufs vorzubeugen, ist man gezwungen, die einzelnen Stringlängen so groß wie möglich festzulegen, was zu einer enorm ineffizienten Speicherausnutzung führt. Da der Implementierungsaufwand dieses Verfahrens – gemessen an der dynamischen Stringverwaltung – bedeutend geringer ist, hat sie u.a auch im Integer-Basic des Apple II Anwendung gefunden.



Die dynamische Handhabung von Stringvariablen offenbart sich dem Anwender in erster Linie durch den Wegfall der Längendeklaration. Wird die Variable zum ersten Mal benutzt, so erfolgt auch hier ein Eintrag im entsprechenden Deskriptor, und die Zeichenkette selbst wird in der Regel oberhalb des freien Speicherbereichs abgelegt. Bei einer erneuten Zuweisung wird der neue String unter die bereits vorhandenen geschrieben und der Deskriptor

entsprechend geändert. Erscheint also in einem Programm zunächst die Anweisung A$ = „OTTO“ und später die Zuordnung A$ = „FRITZ“, so steht sowohl „FRITZ“ als auch „OTTO“ im Speicher, obwohl „OTTO“ jetzt nicht mehr benötigt wird, also Garbage = Müll darstellt. „OTTO“ ist damit eine Stringleiche, die nur dazu geeignet ist, den verfügbaren Speicherraum zu schmälern. Im Laufe der Zeit würden sich immer mehr solcher Stringleichen ansammeln, und der Rechner müßte sich wohl dann mit der Meldung „OUT OF MEMORY“ verabschieden, gäbe es nicht die Garbage-Collection oder Beseitigung der Stringleichen.

Bei der Garbage-Collection werden alle noch aktuellen Strings soweit wie möglich nach oben verschoben, d.h. alle (passiven) Stringleichen werden entfernt und die verbleibenden (aktiven) Strings ab HIMEM, d.h. ab der Speicherobergrenze, abwärts zusammengepackt.

Der Applesoft-Interpreter führt diese G/C immer dann aus, wenn kein Speicherplatz mehr zur Verfügung steht. Durch Benutzung der FRE-Funktion, die die Anzahl noch verbleibender freier Speicherstellen ermittelt, kann man diese G/C auch erzwingen, etwa durch I = FRE (0). Eine weitere Möglichkeit besteht im direkten Aufruf der G/C im Applesoft-Interpreter durch CALL 58500 oder CALL -7036 (entspricht $E484).

2. Die Stringvariablentabelle

Zum Verständnis der hier vorgestellten Programme ist die Kenntnis der Variablenverwaltung des Applesoft-Interpreters wichtig. Es empfiehlt sich in diesem Zusammenhang die Lektüre der Applesoft-Basic-Programmieranleitung, aus der hier noch einmal kurz zusammengefaßt werden soll. Beim Programmlauf legt der Interpreter direkt hinter dem Programm die Variablentabelle an, wobei zuerst alle einfachen Variablen und dann alle indizierten oder dimensionierten (Feld-)Variablen folgen. Es dürfte bekannt sein, daß nur die ersten beiden Buchstaben des Variablennamens signifikant sind, d.h. XMIN und XMAX werden nicht unterschieden. Somit enthält jeder Variableneintrag zunächst zwei Namensbytes. Bit 7 jedes dieser beiden Bytes wird zur Unterscheidung des Variablentyps benutzt (Integer, Real, String). Stringvariablen werden durch Bit 7 = 1 = on im ersten sowie Bit 7 = 0 = off im zweiten Byte gekennzeichnet. Bei den einfachen Variablen folgen nun noch je 5 Bytes, die den Wert beinhalten. Im Falle von Strings werden indessen nur die ersten 3 der 5 Bytes als Deskriptor genutzt. Feldvariablen (= dimensionierte Variablen) enthalten einige Einträge mehr. Auch hier erscheinen zuerst zwei Namensbytes, gefolgt von einer 2-Byte-Offsetadresse zum nächsten Feld, danach die Anzahl der Dimensionen (1 Byte), gefolgt von je 2 Bytes zur Bestimmung der Größe der einzelnen Dimensionen. Im Anschluß daran erscheinen die Werte der einzelnen Elemente, bei Strings wieder in Form von Deskriptoren. Eine kleine Lehre läßt sich aus diesen Erkenntnissen ziehen: Bei der Bearbeitung großer Felder erweist es sich als sinnvoll, alle einfachen Variablen vor der Dimensionierung des Feldes zu benennen, da ansonsten bei jeder neuen einfachen Variablen der gesamte Feldblock nach oben verschoben wird.
Der Applesoft-Interpreter richtet sich für seine Verwaltungstätigkeit Pointer in der Zero-Page ein, die in **Diagramm 1** veranschaulicht sind. Dabei entsprechen LOMEM und HIMEM den Default-Werten.

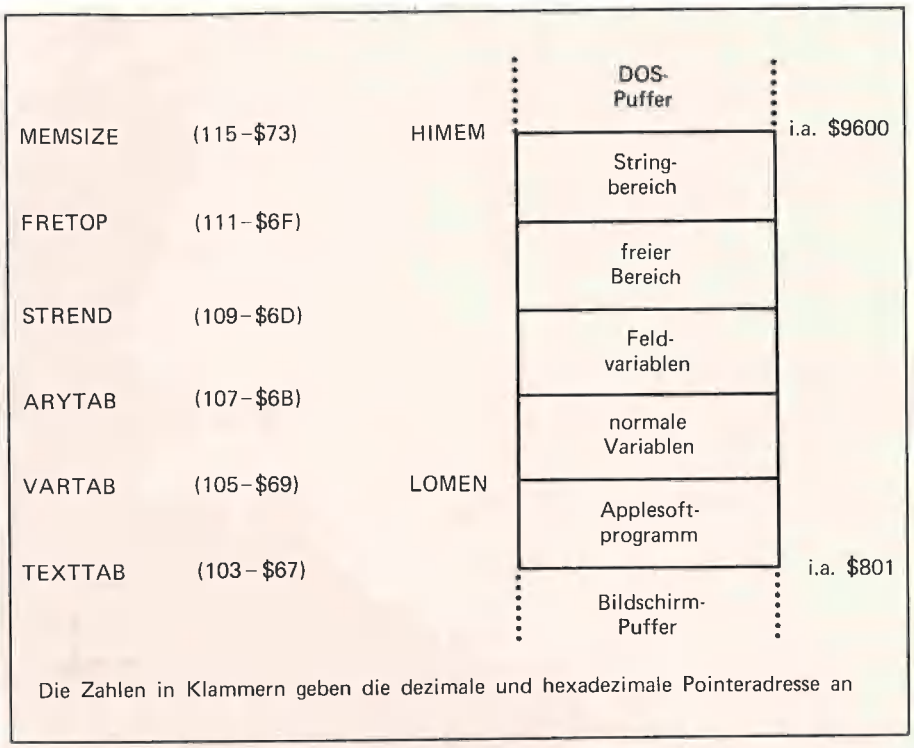


Diagramm 1: Auszug aus der Speicherbelegung

Stringzahl	100	500	1000	2000	5000
LC-version	,032	,11	,20	,52	1,3
48K-version	,020	,14	,43	1,4	7,6
Applesoft-version	,94	19,4	74,4	292,0	1800,0

Genauigkeit +/- 5% (ungeeichte Zeitbasis) Zeit in sec

Diagramm 2: Laufzeittabelle

IBM®-Gehäuse **229,-**

ZUSATZ-KARTEN:

V-24-Schnittstelle ..... **199,-** **Z-80-Karte** **139,-**
80-Zeichen-Karte m.Softswitch **236,-** **16 K-Language-Karte** **138,-**
Centronics-Karte von Epson für Graphik .... **210,-** für Text **145,-**
Eprommer incl. Software **198,-**

Floppy-Controler

FDC 4 für alle Laufwerke ..... **230,-** Bausatz wie links **199,-**
Leerplatine wie oben incl. Prom u. Eprom **130,-**
Autopatch-Controler 1 x 35 bis 2 x 80 Tr.-Disk, keine Patch-Disk notwendig, Gemischtbetrieb möglich ..... **298,-** **(auch für SONY 3½" Laufwerke)**

Joy Stick ..... **69,-** **Netzteil 5A** **149,-**

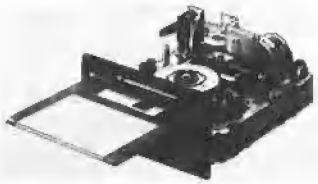
Halbschalen-Gehäuse für 2 Slimline-Laufwerke **65,-**
Floppy-Kabel 34pol. für 2 Laufwerke mit Shugart-Bus **42,-**

Preh Commander Keyboards Wir bieten Ihnen die **Preh-Qualität** auch für Apple. AK 88 spez. mit Gehäuse, Anschlußkabel, Zehner-Tastenfeld, dt. Zeichensatz, Sondertasten für Ctrl-Codes und Rechenfunktionen **339,-**

TEAC FD 55 A 1 x 40 Track ..... **550,-** **TEAC FD 55 B** 2 x 40 Track **631,-**
TEAC FD 55 E 1 x 80 Track ..... **595,-** **TEAC FD 55 F** 2 x 80 Track **695,-**

Patch-Diskette für **SONY 3½" Laufwerke** – ermöglicht die Anpassung an II/IIe und kompatible Computer ..... **80,-** Manual vorab 15,-DM (wird beim Kauf der Patch-Diskette angerechnet).
dto. für 5¼" Laufwerke **80,-**
Disketten Döb&Böd SS/DD 10St. **62,-** **10 Disketten** f.Sony 3½" Laufw. **150,-**

SONY 3½" Laufwerk .. nur 849,-



10 MB Winchester incl. Controler, Software, Gehäuse **4490,-**

Akustikkoppler AK 300 mit FTZ-Nr. incl. Netzteil **549,-**

Die Microfloppy mit Zukunft: (o.Abb.) Speicherkapazität: 2 x 1 MByte formatiert: 2 x 640 kByte. Anschlußfertig mit PROM-residenter Patchsoftware für CP/M 2,2, Apple DOS 3,3 und Apple Pascal 1.1, Pro-DOS zum Preis von **2499,-**

Gesamt-Preisliste anfordern! Bitte beachten Sie auch unsere Anzeige in Nr. 2/84

Ueding electronics

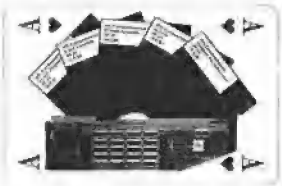
Holtewiese 2
5750 Menden 1

DFÜ 02373/66877
Tel. 02373/63159

Die besten Karten für Ihren Apple®

Alle Trümpfe aus einer Hand

Ein Textsystem wie im Film mit:
AP 17 256K RAM als Pseudodisk
Basic, Pascal, CP/M DM 1690,–
AP 22 Z80B CPU 6 MHz mit 64k RAM DM 895,–
Das Textprogramm Wordstar DM 1310,–



Bestechende Leistung!

AP 20: 6800 CPU mit 128K RAM	DM 1459,–
AP 20 mit Assembler und Pseudodisk	DM 1595,–
AP 20 mit CP/M 68K	DM 2268,–
AP 26-A-256K RAM für AP 20	DM 1695,–
AP 26-B-1 Megabyte für AP 20	DM 6450,–
AP 20 + AP 26 + CP/M 68K	DM 3642,–
CP/M 68K mit C-Compiler	DM 1356,–
Fortran, Pascal, Basic u.a	auf Anfrage

Disketten mit
Innenlochverstärkung ss, sd
10er Pack DM 55,–
100er Pack DM 500,–

Diese Kombination bestimmt Ihre Computerzukunft:

16K RAM Karte	DM 165,–
80Z Karte mit Softswitch	DM 295,–
Z80A Karte	DM 165,–
zusammen nur	DM 595,–

alle Preise incl. MWSt.

iBS COMPUTERTECHNIK

Olper Str. 10
4800 Bielefeld 14
Tel. 0521/444032
West-Germany

1011 Rose Marie Lane 16
Stockton. CA 95207
Tel. (209) 473-7473
USA

Nutzen Sie Ihren Apple-Computer auch zum Lernen

INTUS-Lernprogramme sind attraktiv und motivierend für vergnügliches Lernen mit hohem Lernerfolg.
Alle Programme in deutscher Sprache für Apple IIe und IIc.

- Maschinenschreiben wie der Blitz. Didaktisch ausgezeichnet. Garantiert erfolgreich in 20 Lektionen. DM 188,—
- Basic-Lernprogramm. Bestes Lernprogramm 1982 in USA. DM 295,—
- Computer-Simulator. Mit Lehrgang in 5 Lektionen. DM 146,—
- Lesen wie der Blitz. Sehr wirkungsvolles Lesetraining. DM 115,—
- Deutsche Grammatik mit Spaß. Für Sekretärinnen, Schüler und "Jedermann". Beste Beurteilungen durch Verwender.
  - Rechtschreibetraining mit Vortest/Nachtest, 5 Bereiche, 4 Disk. je Disk. DM 165,—
  - Zeichensetzung, Wortarten, Verb, Zeitenfolge, Adjektive/Adverbien, Fälle, Satzglieder, 7 Disk. (weitere folgen) je Disk. DM 165,—
- Wortschatztrainer, Franz./Deutsch, 800 Wörter DM 140,—
  Englisch/Deutsch, 800 Wörter DM 140,—
  Business English, 1200 Wörter DM 163,—
- Unternehmens-Planspiel "Kartellbrüder und Halsabschneider". Professionelles, lehrreiches Strategiespiel. Vorkenntnisse notwendig. Umfangreiche Dok. DM 495,—
- Schulprogramme wie Rechnen/Mathe, Sprachen, Physik, Chemie, Informatik, Vorschule. Bestens für die Nachhilfe geeignet. Vermietung von Programmen.

★★★★ Mit Ihrer ersten Bestellung erhalten Sie gratis das Programm "Mein Freund der Apple" mit interessanten Denk-Strategie- und Entspannungsspielen.

INTUS LERN-SYSTEME AG

Kaiserstraße 21 · 7890 Waldshut-Tiengen · Telefon 07751-7920

cvb computer

Repsoldstraße 49 2000 Hamburg 1, Tel 040-230685

Komplettsystem SUPER II

+ 48 KB Mikrocomputer, Kleinschreibung, deutsche Tastatur, Zehnerblock, Auto-Repeat, APPLE kompatibel, 8 Slots
+ 16 KB Erweiterung auf 64 KB Hauptspeicher (Sprachkarte), PASCAL-, VISICALC- und CP/M-kompatibel, busgepuffert
+ 5.25"-Diskettenlaufwerk TEAC FD 55-A, voll kompatibel zu APPLE-Laufwerken, 143 KB, mit Controller-Karte
+ 12"-Datenmonitor SANYO DM 2112 CX, grüner Bildschirm, 15 MHz Auflösung, geätzte Bildröhre

anschlußfertig **DM 2450.00**

+ TEAC 5.25"-Laufwerk FD 55-A
+ Z 80-Mikroprozessorkarte
+ 80 Zeichen-Superkarte VIDEX kompatibel, deutscher und ASCII-Zeichensatz, Grafik, Softswitches, busgepuffert

DM 1100.00

TEAC 55-F

Doppellaufwerk 2x 640 KB mit Gehäuse, Controller, Kabel und Software, anschlußfertig für Äpfel

DM 2198.00

SPEEDY 100

80 cps Matrixdrucker A4, Einzelblatt und Traktor, Grafik, EPSON-kompatibel Centronics Parallel

DM 799.00

JUKI 6100

Profi-Typenraddrucker A4 quer, 22 cps, 2K Puffer, WordStar kompatibel, Centronics Parallel

DM 1699.00

Apple · NCR · Epson · JUKI · Speedy · C.Itoh · NEC · Olympia · Sanyo
Taxan · Teac · Verbatim · Scotch 3M · M•T Zubehör-Bücher
Barpreise incl. Mehrwertsteuer volle Garantie
Abholung oder Versand · Infoheft anfordern

3. Die Interpreter-Garbage-Collection

Zum Verständnis der im Applesoft-Interpreter eingebauten G/C-Routine bedarf es einiger Erläuterungen. Die Block-Transfer-Utillity ist eine universelle Blockverschieberoutine, die immer dann aufgerufen wird, wenn ein bestimmter Speicherbereich (LOWTR-HIGHTR) verschoben werden muß. Dabei wird das Ende des Zielbereichs (HIGHDS) als Parameter übergeben. Diese Routine ist nicht geeignet für Verschiebungen nach unten mit Überschneidungen von Ziel- und Quellbereich. Der Deskriptorenstack ist ein kurzer Speicherbereich in der Zero-Page (9 Bytes), in dem bei verschiedenen Stringoperationen, wie z.B. LEFT$, und Stringkonkatenationen (Stringverkettungen, z.B. A$ + B$) Deskriptoren vorübergehend abgelegt werden. Da die G/C zu jeder Zeit aufgerufen werden kann, müssen auch diese Zeichenketten mit berücksichtigt werden.
Bei dem hier angewandten Algorithmus wird in einer großen Schleife bei jedem Durchgang der bisher noch nicht berücksichtigte String mit der höchsten Adresse ermittelt, und dann unter die bereits bearbeiteten Zeichenketten geschrieben. Somit werden alle Strings, von oben nach unten, neu abgelegt und die entsprechenden Deskriptoren in der Variablenliste aktualisiert. Bei der Suche nach dem höchsten String, dem Top-String, wird folgendermaßen verfahren: Zunächst befinden sich alle potentiellen Top-Strings zwischen STREND und FRETOP, d.h. zwischen der unteren Grenze des zur Verfügung stehenden Stringbereichs und dem zuletzt aktualisierten Top-String. Bei der Durchforstung aller Variablen, wobei die numerischen Variablen natürlich ignoriert werden, rückt die Untergrenze dieses Bereichs durch den zuletzt gefundenen Top-String immer weiter nach oben. Dabei fungiert LOWTR als Untergrenze, TOPSTR als Zeiger auf den letzthöchsten Eintrag und INDEX als Laufvariable innerhalb der Variablenliste. Am Ende aller Variablen steht somit der höchste String fest und kann übernommen werden. Dies wird solange wiederholt, bis kein Top-String mehr gefunden wird und somit alle Strings verschoben sind. Die Variable TOPSTR wird dabei am Schleifenkopf als Flag auf Null gesetzt; wurde kein Top-String mehr angetroffen, so bleibt sie als Abruchkriterium zurückgesetzt.

Das besondere Merkmal dieses Programms ist der Verzicht auf die Benutzung weiterer Speicherplätze. Einerseits ist der

APPLESOFT.FRE

```
                        1  *-------------------------------------*
                        2  *                                     *
                        3  *  GARBAGE-COLLECTION-Routine im Applesoft-Interpreter  *
                        4  *                                     *
                        5  *                $E484-$E596          *
                        6  *-------------------------------------*
                        7
                        8  TEMPPT   EQU  $52        ;Pointer für Deskriptorenstack
                        9  DSCSTK   EQU  $55        ;Deskriptorenstack (3 * 3 Bytes)
                       10  INDEX    EQU  $5E        ;Suchpointer
                       11  VARTAB   EQU  $69        ;Start der Variablen = LOMEM
                       12  ARYTAB   EQU  $6B        ;Start der Feldvariablen
                       13  STREND   EQU  $6D        ;Ende der numerischen Variablen
                       14  FRETOP   EQU  $6F        ;Ende der Stringvariablen
                       15  MEMSIZE  EQU  $73        ;= HIMEM
                       16  TOPSTR   EQU  $8A        ;Pointer zum letzten Stringdeskr.
                       17  VARLEN   EQU  $8F        ;Anzahl von Variablenbytes
                       18  LSTLEN   EQU  $91        ;Zwischenspeicher für VARLEN
                       19  ARYPNT   EQU  $94        ;Pointer innerhalb der Feldv.
                       20                           ;Parameter für Block-Transfer
                       21  HIGHDS   EQU  $94        ;Zielbereich, Ende
                       22  HIGHTR   EQU  $96        ;Quellbereich, Ende
                       23  LOWTR    EQU  $9B        ;Quellbereich, Start
                       24
                       25  BLTRU1   EQU  $D39A      ;Block-Transfer-Utility (Entry)
                       26
                       27           ORG  $E484
                       28
                       29  * Beim Aufruf von GARBAG muß VARLEN auf 3 gesetzt sein
                       30
E484: A6 73            31  GARBAG   LDX  MEMSIZE    ;bei HIMEM beginnen
E486: A5 74            32           LDA  MEMSIZE+1
E488: 86 6F            33  GBLOOP   STX  FRETOP     ;FRETOP aktualisieren
E48A: 85 70            34           STA  FRETOP+1
E48C: A0 00            35           LDY  #$0        ;Variablenindex
E48E: 84 8B            36           STY  TOPSTR+1   ;0 als Flag
E490: A5 6D            37           LDA  STREND     ;maximal kleinste Adresse
E492: A6 6E            38           LDX  STREND+1   ; für Stringvariable
E494: 85 9B            39           STA  LOWTR      ; als zuletzt geänderten
E496: 86 9C            40           STX  LOWTR+1    ; String eintragen
                       41
                       42  * Deskriptorenstack bearbeiten
                       43
E498: A9 55            44           LDA  #<DSCSTK   ;Deskriptorstack mit-
E49A: A2 00            45           LDX  #>DSCSTK   ; berücksichtigen
E49C: 85 5E            46           STA  INDEX      ;Suchpointer auf ersten
E49E: 86 5F            47           STX  INDEX+1    ; Stringdeskriptor
E4A0: C5 52            48  DESCSTCK CMP  TEMPPT     ;Deskriptorstack beendet ?
E4A2: F0 05            49           BEQ  NORVAR     ; ja, dann weiter
E4A4: 20 23 E5         50           JSR  SETTOP1    ;prüfen, ob neuer Top-String
E4A7: F0 F7            51           BEQ  DESCSTCK   ;unbedingt
                       52
                       53  * Normale Variablen bearbeiten
                       54
E4A9: A9 07            55  NORVAR   LDA  #7         ;7 Bytes pro Variableneintrag
E4AB: 85 8F            56           STA  VARLEN     ; in Var.-Referenzliste
E4AD: A5 69            57           LDA  VARTAB     ;Suchpointer auf erste
E4AF: A6 6A            58           LDX  VARTAB+1   ; normale Variable
E4B1: 85 5E            59           STA  INDEX
E4B3: 86 5F            60           STX  INDEX+1
E4B5: E4 6C            61  NORVAR1  CPX  ARYTAB+1   ;Feldvariablen erreicht ?
E4B7: D0 04            62           BNE  NORVAR2    ; nein, dann weiter
E4B9: C5 6B            63           CMP  ARYTAB     ; LSB prüfen
E4BB: F0 05            64           BEQ  ARYVAR     ; ja, dann zu den Feldvariablen
E4BD: 20 19 E5         65  NORVAR2  JSR  SETTOP     ;prüfen, ob neuer Top-String
E4C0: F0 F3            66           BEQ  NORVAR1    ;unbedingt
                       67
                       68  * Feldvariablen bearbeiten
                       69
E4C2: 85 94            70  ARYVAR   STA  ARYPNT     ;Feldpointer auf
E4C4: 86 95            71           STX  ARYPNT+1   ; erstes Feld
E4C6: A9 03            72           LDA  #3         ;3 Bytes pro Eintag in der
E4C8: 85 8F            73           STA  VARLEN     ; Feldvariablenreferenzliste
E4CA: A5 94            74  ARYLOOP  LDA  ARYPNT
E4CC: A6 95            75           LDX  ARYPNT+1
E4CE: E4 6E            76  ARYLOOP1 CPX  STREND+1   ;Letztes Feld erreicht ?
E4D0: D0 07            77           BNE  ARYVAR1    ; nein, dann weiter
E4D2: C5 6D            78           CMP  STREND     ; LSB prüfen
E4D4: D0 03            79           BNE  ARYVAR1    ; nein, dann weiter
E4D6: 4C 62 E5         80           JMP  MVESTR     ; ja, dann Top-St. eintragen
E4D9: 85 5E            81  ARYVAR1  STA  INDEX      ;Index auf erstes
E4DB: 86 5F            82           STX  INDEX+1    ; Feldelement
E4DD: A0 00            83           LDY  #$0        ;Feldindex
E4DF: B1 5E            84           LDA  (INDEX),Y  ;1. Feldnamenbyte
E4E1: AA               85           TAX             ; -> X
E4E2: C8               86           INY
```

Algorithmus relativ unkompliziert, was sich in der kurzen Code-Länge widerspiegelt. Andererseits ist die Laufzeit sehr beträchtlich, zumal sie in Abhängigkeit von der Anzahl der Stringvariablen etwa quadratisch steigt, da zur Bearbeitung von n Variablen n-mal alle n Strings untersucht werden müssen. Es ist somit nicht verwunderlich, wenn sich der Apple bei Bearbeitung von großen Stringfeldern in minutenlanges Schweigen hüllt, was besonders bei interaktiven Programmen ein mittleres Ärgernis darstellt.

4. Die neue Garbage-Collection: LC.FRE und RAM.FRE

All denen, die auf eine schnelle Garbage-Collection Wert legen, ist das nun folgende Programm **LC.FRE** gewidmet. Das neue Diskettenbetriebssystem ProDOS bzw. genauer gesagt das Modul BASIC.SYSTEM ist in der Lage, Variablen abzuspeichern. Um die Diskette nicht mit unötigen Stringleichen zu füllen, wird vor Abspeicherung von Stringvariablen eine G/C durchgeführt. Sehr wohl um die Problematik der Applesoft-G/C wissend, haben die Programmierer von ProDOS eine eigene Routine implementiert, die bedeutend schneller läuft. Die Geschwindigkeitserhöhung wurde durch die Einrichtung eines Puffers erreicht, der den gesamten freien Bereich zwischen Variablen und Strings umspannt. Stehen dort weniger als 4 Seiten zur Verfügung, so wird auf die PRODOS-I/O-Puffer zurückgegriffen.
Dieses Programm läßt sich nun mit einigen Modifikationen auch unter dem alten DOS ausnutzen. Um den freien Speicherplatz nicht zu belasten, residiert das Programm in der Language-Card Bank 1 (die von einem ggf. in die LC gemovten DOS 3.3 belegte Bank 2 bleibt unberührt). Der Puffer wird, um Schwierigkeiten mit eventuell geöffneten Files und den HGR-Puffern zu vermeiden, ebenfalls in die Bank 1 verlegt, was auch einen beim PRODOS-FRE-Befehl möglicherweise auftretenden Fehler (OUT OF MEMORY) vermeidet, zumal 14 Seiten Buffer (ca. 3,5K) bei einer großen Anzahl von Strings auch nicht mehr im freien Bereich vorhanden wären. (Das BASIC.SYSTEM bricht exakt dann zusammen, wenn der momentan stringfreie Speicherraum beim Eröffnen eines Textfiles weniger als 1280 Bytes, d.h. weniger als 5 Pages, beträgt, so daß HIMEM theoretisch sicherheitshalber stets bei $9100 liegen müßte. Anm. d. Red.)

Der Programmablauf sieht etwa folgendermaßen aus: Zunächst wird der obere Be-

```
E4E3: B1 5E     87           LDA  (INDEX),Y  ;2. Feldnamenbyte
E4E5: 08        88           PHP             ;Minusflag retten
E4E6: C8        89           INY
E4E7: B1 5E     90           LDA  (INDEX),Y  ;OffsetLSB zum nächsten Feld
E4E9: 65 94     91           ADC  ARYPNT     ;Feldpointer auf nächstes
E4EB: 85 94     92           STA  ARYPNT     ; Feld setzen
E4ED: C8        93           INY
E4EE: B1 5E     94           LDA  (INDEX),Y  ;OffsetMSB zum nächsten Feld
E4F0: 65 95     95           ADC  ARYPNT+1
E4F2: 85 95     96           STA  ARYPNT+1
E4F4: 28        97           PLP             ;Minusflag holen
E4F5: 10 D3     98           BPL  ARYLOOP    ;Bit7 = 0, d.h. kein String
E4F7: 8A        99           TXA
E4F8: 30 D0     100          BMI  ARYLOOP    ;Bit7 = 1, d.h. kein String
E4FA: C8        101          INY
E4FB: B1 5E     102          LDA  (INDEX),Y  ;Anzahl von Dimensionen
E4FD: A0 00     103          LDY  #$0
E4FF: 0A        104          ASL             ;2 Bytes pro Dimension
E500: 69 05     105          ADC  #5         ; + 5 Bytes überspringen
E502: 65 5E     106          ADC  INDEX
E504: 85 5E     107          STA  INDEX      ;LSB -> A
E506: 90 02     108          BCC  ARYVAR2
E508: E6 5F     109          INC  INDEX+1
E50A: A6 5F     110 ARYVAR2  LDX  INDEX+1    ;MSB -> X
E50C: E4 95     111 ARYLOP2  CPX  ARYPNT+1   ;Ende des Feldes erreicht ?
E50E: D0 04     112          BNE  ARYLOP21   ; nein, dann weiter
E510: C5 94     113          CMP  ARYPNT
E512: F0 BA     114          BEQ  ARYLOOP1   ; ja, dann nächstes Feld
E514: 20 23 E5  115 ARYLOP21 JSR  SETTOP1    ;prüfen, ob neuer Top-String
E517: F0 F3     116          BEQ  ARYLOP2    ;unbedingt
                117
                118 * Prüfen, ob neuer Top-String vorliegt
                119
E519: B1 5E     120 SETTOP   LDA  (INDEX),Y  ;1. Namensbyte
E51B: 30 35     121          BMI  NXTVAR     ;Bit7 = 1, d.h. kein String
E51D: C8        122          INY
E51E: B1 5E     123          LDA  (INDEX),Y  ;2. Namensbyte
E520: 10 30     124          BPL  NXTVAR     ;Bit7 = 0, d.h. kein String
E522: C8        125          INY
E523: B1 5E     126 SETTOP1  LDA  (INDEX),Y  ;Stringlänge
E525: F0 2B     127          BEQ  NXTVAR     ;wenn 0, dann übergehen
E527: C8        128          INY
E528: B1 5E     129          LDA  (INDEX),Y  ;Stringadresse LSB -> X
E52A: AA        130          TAX
E52B: C8        131          INY
E52C: B1 5E     132          LDA  (INDEX),Y  ;Stringadresse MSB -> A
E52E: C5 70     133          CMP  FRETOP+1   ;größer als FRETOP ?
E530: 90 06     134          BCC  SETTOP2    ; nein, dann weiter
E532: D0 1E     135          BNE  NXTVAR     ; ja, dann bereits erledigt
E534: E4 6F     136          CPX  FRETOP     ;LSB prüfen
E536: B0 1A     137          BCS  NXTVAR     ;größer, d.h. erledigt
E538: C5 9C     138 SETTOP2  CMP  LOWTR+1    ;größer als letzter String ?
E53A: 90 16     139          BCC  NXTVAR     ; nein, dann übergehen
E53C: D0 04     140          BNE  TOPFND     ; ja, dann Top-String
E53E: E4 9B     141          CPX  LOWTR      ;LSB überprüfen
E540: 90 10     142          BCC  NXTVAR     ;kleiner, d.h. erledigt
                143
                144 * als Top-String eintragen; INDEX auf nächste Variable
                145
E542: 86 9B     146 TOPFND   STX  LOWTR      ;neue Stringadresse eintragen
E544: 85 9C     147          STA  LOWTR+1
E546: A5 5E     148          LDA  INDEX      ;und derzeitigen
E548: A6 5F     149          LDX  INDEX+1    ; Stringpointer als neuen
E54A: 85 8A     150          STA  TOPSTR     ; Top-String eintragen
E54C: 86 8B     151          STX  TOPSTR+1
E54E: A5 8F     152          LDA  VARLEN     ;Variablenlänge retten
E550: 85 91     153          STA  LSTLEN
E552: A5 8F     154 NXTVAR   LDA  VARLEN     ;Suchpointer auf nächste
E554: 18        155          CLC             ; Variable setzen
E555: 65 5E     156          ADC  INDEX
E557: 85 5E     157          STA  INDEX      ;LSB -> A
E559: 90 02     158          BCC  NXTVAR1
E55B: E6 5F     159          INC  INDEX+1
E55D: A6 5F     160 NXTVAR1  LDX  INDEX+1    ;MSB -> X
E55F: A0 00     161          LDY  #$0        ;Variablenindex
E561: 60        162          RTS
                163
                164 * Top-String verschieben, falls gefunden
                165
E562: A6 8B     166 MVESTR   LDX  TOPSTR+1   ;0 ?, d.h. kein Top-String
E564: F0 F7     167          BEQ  NXTVAR1    ; gefunden -> ENDE
E566: A5 91     168          LDA  LSTLEN     ;Variablenlänge des T.-Strings
E568: 29 04     169          AND  #%00000100 ; aus 7 wird 2
E56A: 4A        170          LSR             ; aus 3 wird 0
E56B: A8        171          TAY             ;als Index
E56C: 85 91     172          STA  LSTLEN     ;merken
```

reich der Strings in den Puffer übertragen, und dann werden alle Variablen, die in diesem Puffer liegen, nach oben verschoben, wobei die Reihenfolge hierbei keine Rolle spielt, da ausschließlich gepufferte Strings überschrieben werden. Im Anschluß daran wird der nächste Teil der Strings gepuffert und analog verfahren, bis auf diese Weise alle String bearbeitet wurden. Eine Verfeinerung dieses Verfahrens ergibt sich durch die Ausnutzung der durch die G/C bereits gewonnenen Seiten, die nicht gepuffert werden müssen und dennoch im gleichen Zug verarbeitet werden können. Dies führt zur Unterscheidung eines gepufferten Bereichs, der bis auf den letzten Durchgang stets die gleiche Länge besitzt, und eines Savebereichs, der durch den Speicherplatzgewinn immer größer wird. Der Algorithmus gilt als beendet, wenn der Savebereich die Untergrenze des alten FRETOP-Wertes erreicht hat.

Dem Programm ist ein Initialisierungteil und eine Treiberroutine vorangestellt, um es direkt von der Diskette BRUNen zu können, ohne selbst Umschaltungen vornehmen zu müssen. Die Treiberroutine, die in die Page 3 ab $0300 (dezimal 768) verschoben wird, enthält noch eine kleine Zusatzoption. In der Speicherstelle FREePaGES ($00E3 = dezimal 227) kann die Anzahl der gewünschten freien Seiten im Bereich 1-255 übergeben werden. Sind diese vor der G/C noch vorhanden, so wird sie nicht ausgeführt. Auf die Benutzung des Ampersandvektors wurde zugunsten anderer Erweiterungen verzichtet, der Programmaufruf lautet somit CALL 768:

Die Garbage-Collection wird erzwungen mit:
POKE 227, 255
CALL 768

Dagegen erfolgt die Garbage-Collection nach z.B.
POKE 227, 3
CALL 768
nur dann, wenn z.Zt. des Aufrufs weniger als 3 Pages = weniger als 768 Bytes im Stringpool frei sind.

Eine Besonderheit des Programms ist die ausschließliche Bearbeitung von ganzen Seiten. Dies führt zwar am Start und am Ende zu kleineren Redundanzen, erspart jedoch die ständige Überprüfung des Low Bytes, was eine Erhöhung der Geschwindigkeit und zugleich ein kürzeres Programm bedingt.

```
E56E: B1 8A     173          LDA  (TOPSTR),Y ;Länge des Top-Strings
E570: 65 9B     174          ADC  LOWTR      ; zum Quellbereichsstart
E572: 85 96     175          STA  HIGHTR     ; ergibt
E574: A5 9C     176          LDA  LOWTR+1    ; Quellbereichsende
E576: 69 00     177          ADC  #$0
E578: 85 97     178          STA  HIGHTR+1
E57A: A5 6F     179          LDA  FRETOP     ;FRETOP = Zielbereichsende
E57C: A6 70     180          LDX  FRETOP+1
E57E: 85 94     181          STA  HIGHDS
E580: 86 95     182          STX  HIGHDS+1
E582: 20 9A D3  183          JSR  BLTRU1     ;String verschieben
E585: A4 91     184          LDY  LSTLEN     ;Index wieder herstellen
E587: C8        185          INY             ;neue Adresse in
E588: A5 94     186          LDA  HIGHDS     ; Variable (Deskriptor)
E58A: 91 8A     187          STA  (TOPSTR),Y ; eintragen
E58C: AA        188          TAX             ;FRETOP LSB -> X
E58D: E6 95     189          INC  HIGHDS+1   ;wegen BLTU
E58F: A5 95     190          LDA  HIGHDS+1   ;FRETOP MSB -> A
E591: C8        191          INY
E592: 91 8A     192          STA  (TOPSTR),Y
E594: 4C 88 E4  193          JMP  GBLOOP     ;Nächsten Top-String suchen

275 bytes
```

LC.FRE

```
                1  *-------------------------------------------*
                2  *                                           *
                3  *              ProDOS-FRE-Routine           *
                4  *       modifiziert für Applesoft unter DOS 3.3
                5  *              mit Language-Karte           *
                6  *                                           *
                7  *-------------------------------------------*
                8
                9  *  a) Initialisieren mit BRUN LC.FRE
               10  *  b) FRE-Befehl aufrufen mit:
               11  *     POKE 227, F  wobei F für freie Pages steht
               12  *     (bei POKE 227, 255 erfolgt stets G/C)
               13  *     CALL 768
               14  *
               15  SOURCE   EQU  $3A        ;Quellbereich beim Blocktransfer
               16  DEST     EQU  $3C        ;Zielbereich beim Blocktransfer
               17  INDEX    EQU  $3E        ;Laufvariable,Deskriptorenpointer
               18  VARTAB   EQU  $69        ;Start der Variablen
               19  ARYTAB   EQU  $6B        ;Start der Feldvariablen
               20  STREND   EQU  $6D        ;Ende der Variablen
               21  FRETOP   EQU  $6F        ;Start des Stringbereichs
               22  MEMSIZE  EQU  $73        ;<-> HIMEM
               23
               24  *              Initialisierung              *
               25  *-------------------------------------------*
               26
               27           ORG  $9000
               28
               29  FREPGES  EQU  $E3        ;dezimal 227: POKE 227, FREPGES
               30  PAGE3    EQU  $300
               31  MVEBLK   EQU  $917C
               32
9000: 2C 89 C0 33           BIT  $C089      ;RAM-Karte schreiben
9003: 2C 89 C0 34           BIT  $C089      ; Bank 1
9006: A9 90    35           LDA  #>FREIMGE  ;Parameter für
9008: 85 3B    36           STA  SOURCE+1   ; Block-Transfer setzen
900A: A9 48    37           LDA  #<FREIMGE
900C: 85 3A    38           STA  SOURCE
900E: A9 D0    39           LDA  #>$D000
9010: 85 3D    40           STA  DEST+1
9012: A0 00    41           LDY  #0
9014: 84 3C    42           STY  DEST
9016: A2 02    43           LDX  #2         ;2 Blöcke
9018: 20 7C 91 44           JSR  MVEBLK     ;BLTR-Entry im Fre-Image
901B: 2C 8A C0 45           BIT  $C08A      ;Bankl Schreibschutz
901E: A9 FF    46           LDA  #255       ;G/C immer ausführen
9020: 85 E3    47           STA  FREPGES    ; da stets weniger als 255 Seiten
9022: A0 19    48           LDY  #DRVEND-DRIVER
9024: B9 2E 90 49  MVEDRVE  LDA  DRIVER,Y   ;Treiber-Routine
9027: 99 00 03 50           STA  PAGE3,Y    ; in Seite 3 übertragen
902A: 88       51           DEY
902B: 10 F7    52           BPL  MVEDRVE
902D: 60       53           RTS
               54
               55  *              Treiber-Routine              *
               56  *-------------------------------------------*
               57
```

Für Applebenutzer, die über keine Language Card verfügen, ist am Ende dieses Artikels der Hexdump einer nochmals überarbeiteten Version namens **RAM.FRE** angefügt, die als Puffer ausschließlich den Eingabepuffer ab $0200 benutzt und unterhalb der DOS-Puffer residiert. Sie wurde zugunsten des Speicherplatzes einiger Intelligenz beraubt, zeigt jedoch gegenüber der Original-G/C erheblich kürzere Laufzeiten. Ein ausführliches Listing würde den Rahmen dieses Artikels sprengen, mit der LC-Version müßte die gekürzte Fassung jedoch verständlich werden. (Der Quell-Code von RAM.FRE befindet sich auf der Peeker-Sammeldiskette.) Auch RAM.FRE kann direkt von der Diskette gestartet werden und wird über die Treiberroutine in der Page 3 mit CALL 768 aufgerufen. Nach BRUN RAM.FRE wird HIMEM automatisch auf $94ED gesetzt, denn RAM.FRE belegt den Bereich $94ED-$95FF.

5. Zeitvergleiche

Zur Messung der Laufzeiten der einzelnen Routinen entstand das kleine Applesoft-Testprogramm namens **FRE.TEST**, das nichts anderes tut als eine Menge Stringmüll zu erzeugen. Dies geschieht einfach dadurch, daß jedem Feldelement zweimal eine Zeichenkette (hier nur 1 Zeichen) zugeordnet wird. Der Applesoft-Interpreter legt selbst bei gleicher Länge und gleichem Inhalt stets einen neuen String an. Die Zuordnung A$(I) = „A“ wäre sinnlos, da in diesem Fall der Deskriptor auf die Stelle im Programm selbst verweisen würde, statt auf einen neu angelegten String.

Das Programm gibt vor der Dimensionierung, nach der Zuweisung der Strings und nach der G/C jeweils die Anzahl der durch Strings belegten und der noch freien Speicherplätze an. Vor und nach der G/C erfolgt ein Piepston, um die Zeit stoppen zu können. Es empfiehlt sich, das Programm selbst einmal mit verschiedenen Mengen an Strings zu testen und das Verhältnis der Speicherbereiche zu beobachten. Bei der Benutzung der Original-G/C-Routine ist in der Zeile mit dem CALL 768 statt dessen I = FRE(0) einzusetzen.

Das **Diagramm 2** gibt die Laufzeiten der einzelnen Programme bei verschiedenen Stringzahlen an. Ein Kommentar hierzu erübrigt sich. Die Ungenauigkeit rührt daher, daß nur ein frei schwingender Oszillator als Zeitbasis zur Verfügung stand.

```
902E: 38          58  DRIVER   SEC
902F: A5 6F       59           LDA  FRETOP      ;FRETOP - STREND =
9031: E5 6D       60           SBC  STREND      ; Anzahl freier Seiten
9033: A5 70       61           LDA  FRETOP+1
9035: E5 6E       62           SBC  STREND+1
9037: C5 E3       63           CMP  FREPGES     ;größer wie FREePaGES ?
9039: B0 0C       64           BCS  DRVEND      ; ja, dann fertig
903B: 2C 8B C0    65           BIT  $C08B       ;RAM-Karte lesen
903E: 2C 8B C0    66           BIT  $C08B       ;          schreiben
9041: 20 00 D0    67           JSR  FRE         ;G/C aufrufen
9044: 2C 8A C0    68           BIT  $C08A       ;ROM lesen, RAM Schreibschutz
9047: 60          69  DRVEND   RTS
                  70
                  71  *          modifizierte ProDOS-Routine          *
                  72  *-------------------------------------------------*
                  73
                  74  FREIMGE  ORG  *
                  75           ORG  $D000       ;RAM-Karte Bank1
                  76
                  77  *
                  78  * Speicherverteilung: $D000-$D1FF: Programm
                  79  *                     $D200-$DFFF: Puffer
                  80  *
D000: A9 0D       81  FRE      LDA  #13         ;14 verbleibende Seiten in Bank1
D002: 8D 7B D1    82           STA  PAGES       ; für Puffer einrichten
D005: A9 D2       83           LDA  #>$D200     ;Puffer auf nächste Seite nach
D007: 8D 7A D1    84           STA  BUFFER      ; Fre-Routine setzen
D00A: A5 70       85           LDA  FRETOP+1
D00C: 8D 80 D1    86           STA  OLDTOP      ;alten Stringanfang retten
D00F: A5 74       87           LDA  MEMSIZE+1   ;passen alle Strings in den
D011: E5 70       88           SBC  FRETOP+1    ; Puffer ?
D013: 69 01       89           ADC  #1          ; (Seitenzahl aufrunden)
D015: CD 7B D1    90           CMP  PAGES
D018: B0 03       91           BCS  NOALL       ; ja, dann den Puffer ent-
D01A: 8D 7B D1    92           STA  PAGES       ; sprechend verkleinern
D01D: A5 73       93  NOALL    LDA  MEMSIZE     ;Stringanfang auf HIMEM
D01F: 85 6F       94           STA  FRETOP      ;somit von neuem auffüllen
D021: 18          95           CLC              ; C = 0 wenn Seitenanfang
D022: F0 01       96           BEQ  GOODPGE
D024: 38          97           SEC              ; C = 1 wenn mitten in Seite
D025: A5 74       98  GOODPGE  LDA  MEMSIZE+1
D027: 85 70       99           STA  FRETOP+1    ;   C = 1   ;   C = 0   ;
D029: E9 00       100          SBC  #$00        ;-----------;-----------;
D02B: 8D 7C D1    101          STA  SVESTRT     ;   HIMEM   ;  HIMEM-1  ;
D02E: 69 00       102          ADC  #$00        ;           ;           ;
D030: 8D 7D D1    103          STA  SAVEND      ;  HIMEM+1  ;   HIMEM   ;
                  104
D033: 38          105 LOOP     SEC
D034: A5 69       106          LDA  VARTAB      ;Laufvariable auf Variablen-
D036: E9 07       107          SBC  #7          ; start setzen und 7 Bytes
D038: 85 3E       108          STA  INDEX       ; Offset abziehen
D03A: A5 6A       109          LDA  VARTAB+1    ;
D03C: E9 00       110          SBC  #$00        ; wird später
D03E: 85 3F       111          STA  INDEX+1     ; wieder hinzuaddiert !
D040: A5 6B       112          LDA  ARYTAB      ;Feldindex setzen
D042: 8D 7F D1    113          STA  ARYINDX
D045: AD 7C D1    114          LDA  SVESTRT     ;Untergrenze des alten String-
D048: CD 80 D1    115          CMP  OLDTOP      ; bereiches erreicht ?
D04B: 90 3A       116          BCC  RETURN      ; ja, dann fertig
D04D: ED 7B D1    117          SBC  PAGES       ;Pufferbereich um Anzahl Puffer-
D050: 8D 7E D1    118          STA  BUFSTRT     ; seiten tiefer setzen
D053: A5 70       119          LDA  FRETOP+1    ;kann um die durch die G/C gewon-
D055: ED 7C D1    120          SBC  SVESTRT     ; nenen Seiten verringert werden
D058: ED 7E D1    121          SBC  BUFSTRT     ; (erfolgt durch Betragsbildung
D05B: 49 FF       122          EOR  #$FF        ; mittels Zweierkomplement)
D05D: 69 02       123          ADC  #$02        ;aufrunden
D05F: CD 7E D1    124          CMP  BUFSTRT     ;neuer BUFstart unterschritten ?
D062: 90 03       125          BCC  SETBUF      ; ja, dann weiter
D064: AD 7E D1    126          LDA  BUFSTRT     ; nein, dann wieder abrunden
D067: CD 80 D1    127 SETBUF   CMP  OLDTOP      ;letzter Durchgang ?
D06A: B0 05       128          BCS  SETSAV      ; nein, dann weiter
D06C: AD 80 D1    129          LDA  OLDTOP      ; ja, dann Savebereich
D06F: E9 00       130          SBC  #$00        ; korrigieren (abgerundet)
D071: 8D 7C D1    131 SETSAV   STA  SVESTRT
D074: 20 20 D1    132          JSR  BLTR        ;Savebereich puffern
D077: A6 6C       133          LDX  ARYTAB+1
D079: 20 88 D0    134          JSR  NORVAR      ;einfache Variablen bearbeiten
D07C: 20 BB D0    135          JSR  ARYVAR      ;Feldvariablen bearbeiten
D07F: AD 7C D1    136          LDA  SVESTRT     ;alter Save-Start =
D082: 8D 7D D1    137          STA  SAVEND      ; neues Save-Ende
D085: D0 AC       138          BNE  LOOP        ;nächsten Block (unbedingt)
D087: 60          139 RETURN   RTS              ; E N D E
                  140
                  141 * Einfach Variablen bearbeiten
                  142
D088: 18          143 NORVAR   CLC
```

```
LC.FRE

BSAVE LC.FRE, A$9000, L$01C9

$9000: 2C 89 C0 2C 89 C0 A9 90
$9008: 85 3B A9 48 85 3A A9 D0
$9010: 85 3D A0 00 84 3C A2 02
$9018: 20 7C 91 2C 8A C0 A9 FF
$9020: 85 E3 A0 19 B9 2E 90 99
$9028: 00 03 88 10 F7 60 38 A5
$9030: 6F E5 6D A5 70 E5 6E C5
$9038: E3 B0 0C 2C 8B C0 2C 8B
$9040: C0 20 00 D0 2C 8A C0 60
$9048: A9 0D 8D 7B D1 A9 D2 8D
$9050: 7A D1 A5 70 8D 80 D1 A5
$9058: 74 E5 70 69 01 CD 7B D1
$9060: B0 03 8D 7B D1 A5 73 85
$9068: 6F 18 F0 01 38 A5 74 85
$9070: 70 E9 00 8D 7C D1 69 00
$9078: 8D 7D D1 38 A5 69 E9 07
$9080: 85 3E A5 6A E9 00 85 3F
$9088: A5 6B 8D 7F D1 AD 7C D1
$9090: CD 80 D1 90 3A ED 7B D1
$9098: 8D 7E D1 A5 70 ED 7C D1
$90A0: ED 7E D1 49 FF 69 02 CD
$90A8: 7E D1 90 03 AD 7E D1 CD
$90B0: 80 D1 B0 05 AD 80 D1 E9
$90B8: 00 8D 7C D1 20 20 D1 A6
$90C0: 6C 20 88 D0 20 BB D0 AD
$90C8: 7C D1 8D 7D D1 D0 AC 60
$90D0: 18 A5 3E 69 07 85 3E 90
$90D8: 02 E6 3F 45 6B D0 04 E4
$90E0: 3F F0 EC A0 00 B1 3E C8
$90E8: 51 3E 10 E4 B1 3E 10 E0
$90F0: A0 04 B1 3E CD 7C D1 90
$90F8: D8 CD 7D D1 B0 D2 20 43
$9100: D1 F0 CD 20 E7 D0 B0 5F
$9108: A0 02 B1 3E CD 7C D1 90
$9110: 08 CD 7D D1 B0 03 20 43
$9118: D1 18 A9 03 65 3E 85 3E
$9120: 90 02 E6 3F CD 7F D1 D0
$9128: DF E4 3F D0 DB F0 D4 18
$9130: AD 7F D1 85 3E 86 3F 45
$9138: 6D D0 04 E4 6E F0 28 A0
$9140: 02 B1 3E 65 3E 8D 7F D1
$9148: C8 B1 3E 65 3F AA A0 00
$9150: B1 3E C8 51 3E 10 D8 A0
$9158: 04 B1 3E 0A 69 05 65 3E
$9160: 85 3E 90 03 E6 3F 18 60
$9168: AD 7E D1 85 3B AD 7A D1
$9170: 85 3D A0 00 84 3A 84 3C
$9178: AE 7B D1 E8 B1 3A 91 3C
$9180: C8 D0 F9 E6 3D E6 3B CA
$9188: D0 F2 60 CD 7E D1 90 07
$9190: ED 7E D1 18 6D 7A D1 85
$9198: 3B 88 B1 3E 85 3A 88 38
$91A0: A5 6F F1 3E 85 6F C8 91
$91A8: 3E A5 70 E9 00 85 70 C8
$91B0: 91 3E 88 88 B1 3E F0 09
$91B8: A8 88 B1 3A 91 6F 98 D0
$91C0: F8 60 00 00 00 00 00 00


RAM.FRE

BSAVE RAM.FRE, A$94C0, L$0140

$94C0: A0 10 B9 DC 94 99 00 03
$94C8: 88 10 F7 A9 FF 85 E3 A9
$94D0: 94 A0 ED 85 74 85 70 84
$94D8: 73 84 6F 60 38 A5 6F E5
$94E0: 6D A5 70 E5 6E C5 E3 B0
$94E8: 03 4C ED 94 60 A5 70 8D
$94F0: FF 95 A5 73 85 6F 18 F0
$94F8: 01 38 A5 74 85 70 E9 00
$9500: 8D FD 95 38 A5 69 E9 07
$9508: 85 3E A5 6A E9 00 85 3F
$9510: A5 6B 8D FE 95 20 BE 95
$9518: A6 6C 20 2C 95 20 5C 95
$9520: CE FD 95 AD FD 95 CD FF
$9528: 95 B0 D9 60 18 A5 3E 69
$9530: 07 85 3E 90 02 E6 3F 45
$9538: 6B D0 04 E4 3F F0 EC A0
$9540: 00 B1 3E C8 51 3E 10 E4
$9548: B1 3E 10 E0 A0 04 B1 3E
$9550: CD FD 95 90 D8 D0 D5 20
```

```
D089: A5 3E     144 NORVAR1 LDA  INDEX      ;Laufvariable
D08B: 69 07     145         ADC  #7         ; um Offset zur nächsten
D08D: 85 3E     146         STA  INDEX      ; Variablen erhöhen
D08F: 90 02     147         BCC  NORVAR2
D091: E6 3F     148         INC  INDEX+1
D093: 45 6B     149 NORVAR2 EOR  ARYTAB     ;Feldvariablen erreicht ?
D095: D0 04     150         BNE  NORVAR3    ; nein, dann weiter
D097: E4 3F     151         CPX  INDEX+1    ;MSB prüfen
D099: F0 EC     152         BEQ  RETURN     ; ja, dann zurück
D09B: A0 00     153 NORVAR3 LDY  #$0
D09D: B1 3E     154         LDA  (INDEX),Y  ;1. Namensbyte
D09F: C8        155         INY
D0A0: 51 3E     156         EOR  (INDEX),Y  ;2. Namensbyte
D0A2: 10 E4     157         BPL  NORVAR     ;kein String, wenn Bit7 = Bit7
D0A4: B1 3E     158         LDA  (INDEX),Y
D0A6: 10 E0     159         BPL  NORVAR     ;kein String, wenn Bit7 = 0
D0A8: A0 04     160         LDY  #$4        ;4. Eintrag =
D0AA: B1 3E     161         LDA  (INDEX),Y  ; Stringadresse MSB
D0AC: CD 7C D1  162         CMP  SVESTRT    ;liegt Variable im gepufferten
D0AF: 90 D8     163         BCC  NORVAR1    ; Bereich ?
D0B1: CD 7D D1  164         CMP  SAVEND
D0B4: B0 D2     165         BCS  NORVAR     ; nein, dann nächste Variable
D0B6: 20 43 D1  166         JSR  MVESTR     ; ja, dann verschieben
D0B9: F0 CD     167         BEQ  NORVAR     ;unbedingt
                168
                169 * Array-Variablen bearbeiten
                170
D0BB: 20 E7 D0  171 ARYVAR  JSR  NXTARY     ;Array-Pointer erhöhen
D0BE: B0 5F     172         BCS  RTNARY     ;Ende aller Variablen erreicht
D0C0: A0 02     173 ARYVAR1 LDY  #$02       ;2. Eintrag = MSB
D0C2: B1 3E     174         LDA  (INDEX),Y  ;liegt Variable im gepufferten
D0C4: CD 7C D1  175         CMP  SVESTRT    ; Bereich ?
D0C7: 90 08     176         BCC  NXTVAR
D0C9: CD 7D D1  177         CMP  SAVEND
D0CC: B0 03     178         BCS  NXTVAR     ; nein, dann gleich weiter
D0CE: 20 43 D1  179         JSR  MVESTR     ; ja, dann verschieben
                180
D0D1: 18        181 NXTVAR  CLC
D0D2: A9 03     182         LDA  #3         ;Offset zum nächsten Deskriptor
D0D4: 65 3E     183         ADC  INDEX      ;Laufvariable auf nächsten
D0D6: 85 3E     184         STA  INDEX      ; Deskriptor setzen
D0D8: 90 02     185         BCC  NXTVAR1
D0DA: E6 3F     186         INC  INDEX+1
D0DC: CD 7F D1  187 NXTVAR1 CMP  ARYINDX    ;Feldende erreicht ?
D0DF: D0 DF     188         BNE  ARYVAR1    ; nein, dann normal weiter
D0E1: E4 3F     189         CPX  INDEX+1    ;MSB prüfen
D0E3: D0 DB     190         BNE  ARYVAR1
D0E5: F0 D4     191         BEQ  ARYVAR     ; ja, dann erst NXTARY aufrufen
                192
                193 * Laufvariable auf nächstes Stringfeld setzen
                194
D0E7: 18        195 NXTARY  CLC             ;als Flag und für Addition
D0E8: AD 7F D1  196         LDA  ARYINDX    ;Laufvariable auf
D0EB: 85 3E     197         STA  INDEX      ; nächstes Feld setzen
D0ED: 86 3F     198         STX  INDEX+1
D0EF: 45 6D     199         EOR  STREND     ;Stringende erreicht ?
D0F1: D0 04     200         BNE  NXTARY1    ; nein, dann weiter
D0F3: E4 6E     201         CPX  STREND+1   ;MSB prüfen
D0F5: F0 28     202         BEQ  RTNARY     ; ja, dann mit C = 1 zurück
D0F7: A0 02     203 NXTARY1 LDY  #$2        ;2. Eintrag = Relativpionter
D0F9: B1 3E     204         LDA  (INDEX),Y  ; zum nächsten Feld
D0FB: 65 3E     205         ADC  INDEX      ;zu INDEX addieren
D0FD: 8D 7F D1  206         STA  ARYINDX    ;und als neuen Feldpointer
D100: C8        207         INY             ; merken
D101: B1 3E     208         LDA  (INDEX),Y
D103: 65 3F     209         ADC  INDEX+1
D105: AA        210         TAX             ; X -> MSB
D106: A0 00     211         LDY  #$0        ;NÄCHSTES FELD PRÜFEN
D108: B1 3E     212         LDA  (INDEX),Y  ;1. Namensbyte
D10A: C8        213         INY
D10B: 51 3E     214         EOR  (INDEX),Y  ;2. Namensbyte
D10D: 10 D8     215         BPL  NXTARY     ;kein String, wenn Bit7 = Bit7
D10F: A0 04     216         LDY  #$4        ;4. Eintrag = Anzahl DIM's
D111: B1 3E     217         LDA  (INDEX),Y
D113: 0A        218         ASL             ;2 Bytes pro Dimension
D114: 69 05     219         ADC  #5         ; + 5 Bytes überspringen
D116: 65 3E     220         ADC  INDEX      ; (wegen Tabelle der Größen
D118: 85 3E     221         STA  INDEX      ;  der einzelnen Dimensionen)
D11A: 90 03     222         BCC  RTNARY
D11C: E6 3F     223         INC  INDEX+1
D11E: 18        224         CLC             ;C = 0 -> kein Stringende
D11F: 60        225 RTNARY  RTS             ;C = 1 -> Stringende erreicht
                226
                227 * Savebereich in Puffer übertragen
                228
D120: AD 7E D1  229 BLTR    LDA  BUFSTRT    ;Startadresse des zu puffernden
```

```
$9558: D0 95 F0 D0 20 85 95 B0
$9560: 5C A0 02 B1 3E CD FD 95
$9568: 90 05 D0 03 20 D0 95 18
$9570: A9 03 65 3E 85 3E 90 02
$9578: E6 3F CD FE 95 D0 E2 E4
$9580: 3F D0 DE F0 D7 18 AD FE
$9588: 95 85 3E 86 3F 45 6D D0
$9590: 04 E4 6E F0 28 A0 02 B1
$9598: 3E 65 3E 8D FE 95 C8 B1
$95A0: 3E 65 3F AA A0 00 B1 3E
$95A8: C8 51 3E 10 D8 A0 04 B1
$95B0: 3E 0A 69 05 65 3E 85 3E
$95B8: 90 03 E6 3F 18 60 A0 00
$95C0: AD FD 95 84 3A 85 3B B1
$95C8: 3A 99 00 02 C8 D0 F8 60
$95D0: A9 02 85 3B 88 B1 3E 85
$95D8: 3A 88 38 A5 6F F1 3E 85
$95E0: 6F C8 91 3E A5 70 E9 00
$95E8: 85 70 C8 91 3E 88 88 B1
$95F0: 3E F0 09 A8 88 B1 3A 91
$95F8: 6F 98 D0 F8 60 00 00 00
```

```
FRE.TEST

100  HOME : PRINT "FRE.TEST"
110  PRINT  CHR$ (4)"BRUN LC.FRE"
120  A = 1000: REM Stringanzahl
130  B$ =  CHR$ (7): REM Bell
140  GOSUB 220: DIM A$(A)
150  FOR I = 0 TO A
160  A$(I) = B$:A$(I) = B$
170  NEXT
180  GOSUB 220: PRINT B$;
190  POKE 227, 255: CALL 768:
     REM FRE-Befehl
200  PRINT B$;: GOSUB 220
210  END
220  PRINT "STRINGS= "; PEEK (115) +
     PEEK (116) * 256 - PEEK (111) -
     PEEK (112) * 256,
230  PRINT "FRE= "; PEEK (111) +
     PEEK (112) * 256 - PEEK (109) -
     PEEK (110) * 256
240  RETURN
```

```
D123: 85 3B     230           STA  SOURCE+1   ; Bereichs
D125: AD 7A D1  231           LDA  BUFFER     ;Pufferadressse als Zielbereich
D128: 85 3D     232           STA  DEST+1     ; eintragen
D12A: A0 00     233           LDY  #0         ;nur ganze Seiten
D12C: 84 3A     234           STY  SOURCE
D12E: 84 3C     235           STY  DEST
D130: AE 7B D1  236           LDX  PAGES      ;Seitenzähler
D133: E8        237           INX
D134: B1 3A     238 BLTRLOOP  LDA  (SOURCE),Y ;verschieben
D136: 91 3C     239           STA  (DEST),Y
D138: C8        240           INY
D139: D0 F9     241           BNE  BLTRLOOP
D13B: E6 3D     242           INC  DEST+1     ;seitenweise
D13D: E6 3B     243           INC  SOURCE+1
D13F: CA        244           DEX
D140: D0 F2     245           BNE  BLTRLOOP
D142: 60        246           RTS
                247
                248 * String unter FRETOP eintragen
                249
D143: CD 7E D1  250 MVESTR    CMP  BUFSTRT    ;befindet sich MSB innerhalb
D146: 90 07     251           BCC  MVESTR1    ; des Puffers ?
D148: ED 7E D1  252           SBC  BUFSTRT    ; ja, dann auf entsprechende
D14B: 18        253           CLC             ; Adresse innerhalb des
D14C: 6D 7A D1  254           ADC  BUFFER     ; Puffers setzen
D14F: 85 3B     255 MVESTR1   STA  SOURCE+1   ;MSB des String-Zeigers
D151: 88        256           DEY
D152: B1 3E     257           LDA  (INDEX),Y  ;LSB des String-Zeigers
D154: 85 3A     258           STA  SOURCE     ;als Quellbereich
D156: 88        259           DEY
D157: 38        260           SEC
D158: A5 6F     261           LDA  FRETOP     ;Stringende um Länge
D15A: F1 3E     262           SBC  (INDEX),Y  ; des Strings herabsetzen
D15C: 85 6F     263           STA  FRETOP
D15E: C8        264           INY
D15F: 91 3E     265           STA  (INDEX),Y  ;Deskriptor LSB aktualisieren
D161: A5 70     266           LDA  FRETOP+1
D163: E9 00     267           SBC  #0         ;falls Seite unterschritten
D165: 85 70     268           STA  FRETOP+1
D167: C8        269           INY
D168: 91 3E     270           STA  (INDEX),Y  ;Deskriptor MSB
D16A: 88        271           DEY
D16B: 88        272           DEY
D16C: B1 3E     273           LDA  (INDEX),Y  ;Stringlänge
D16E: F0 09     274           BEQ  RTNMVST    ;wenn 0, dann fertig
D170: A8        275           TAY             ;Länge als Index
D171: 88        276 MVSTLOOP  DEY
D172: B1 3A     277           LDA  (SOURCE),Y ;alten String
D174: 91 6F     278           STA  (FRETOP),Y ; verschieben
D176: 98        279           TYA             ;Zero-Flag ?
D177: D0 F8     280           BNE  MVSTLOOP
D179: 60        281 RTNMVST   RTS
                282
                283 BUFFER    DS   1          ;Adresse des Puffers
                284 PAGES     DS   1          ;Anzahl der Pufferseiten
                285 SVESTRT   DS   1          ;Start des Savebereiches
                286 SAVEND    DS   1          ; Ende des Savebereiches
                287 BUFSTRT   DS   1          ;tatsächlich gepufferter Bereich
                288 ARYINDX   DS   1          ;Feldindex (letzter Feldanfang)
                289 OLDTOP    DS   1          ;<-> FRETOP vor G/C

457 bytes
```

Speichern Sie den Bildschirm ab!

Organisation des 40-Z/Z-Apple- und des 80-Z/Z-Videx-Bildschirms

von Dr. Jürgen B. Kehrel

Den Inhalt einer Bildschirmseite in gedruckter Form vorliegen zu haben kann zu Dokumentationszwecken nützlich sein. Eine Reihe von „intelligenten" Druckerinterfaces ist in der Lage, auf einen einfachen Befehl hin eine sogenannte „Hardcopy" auf einem Drucker zu erzeugen. Was aber, wenn Sie diesen Ausdruck gern mitten in einem längeren Text stehen hätten oder weiterbearbeiten wollen? Hier ergibt sich die Notwendigkeit, eine Bildschirmseite als einen Textfile auf Diskette abzuspeichern, der dann in ein Textverarbeitungsprogramm wie z.B. Applewriter II eingelesen und von beliebigem Text eingerahmt werden kann. Die im folgenden beschriebenen Programme „SCHIRMDISK" und „VIDEXT" erledigen genau diese Aufgabe.

Das SCHIRMDISK-Programm

SCHIRMDISK kann auf zwei Arten benutzt werden: Die eine eignet sich zur Einbindung in Programme, aus denen heraus es aufgerufen werden kann. Die zweite Art ist für solche (Fremd-)Programme gedacht, bei denen man einen CALL-Befehl nicht einbauen kann. Einzige Voraussetzung hier: Mit Reset muß man das Programm verlassen können, und beim Laden darf das SCHIRMDISK-Programm nicht überschrieben werden. Die Diskettenkopie Ihres Bildschirms wird ausgeführt und als Textfile „SCHIRMTEXT" auf das aktuelle Laufwerk geschrieben, wenn Sie die Reset-Taste drücken (Ctrl + Reset auf Apple IIe).

Nun zum Programm selbst. Vor der Anwendung muß es einmal mit „BRUN SCHIRMDISK" initialisiert werden. Dadurch werden die Vektoren in $03F2-$03F4 auf die zweite Einsprungstelle gelegt, denn der Apple schaut nach dem Drücken der Reset-Taste hier nach, wohin er springen soll. $03F2 und $03F3 stellen die Adresse dar und $03F4 eine Prüfzahl, das sogenannte „Power-up-Byte". Es muß den Wert von $03F3 Exklusiv-Oder verknüpft mit der Hexadezimalzahl $A5 besitzen. Falls das nicht der Fall ist, macht Ihr Apple immer einen Kaltstart, benimmt sich also so, als hätten Sie ihn gerade eingeschaltet. Kaum jemand wird im Kopf EOR $A5 ausrechnen können, doch gibt es glücklicherweise im Monitor eine Routine bei $FB6F, die das für uns erledigt und den richtigen Wert nach $03F4 schreibt.

Damit ist die Initialisierungsphase abgeschlossen. Wenn Sie SCHIRMDISK aus einem Programm starten wollen, können Sie dies, auch ohne Reset drücken zu müssen, mit einem **CALL 798** direkt tun.

Der Apple-Textbildschirm

Der Apple kann auf seinem Textbildschirm 24 Zeilen (0-23) mit je 40 Zeichen (0-39) darstellen, also insgesamt 960 Zeichen. Es existieren zwei Textbildschirmseiten, von denen normalerweise nur die Seite 1 angezeigt und benutzt wird. Jedem Zeichen auf dem Bildschirm entspricht eine Speicherposition im RAM des Apple. Die Seite 1 benutzt Speicherstellen von $0400 bis $07FF, die Seite 2 von $0800 bis $0BFF. Letztere werden auch von Applesoft zur Speicherung des Programms verwendet, so daß die Benutzung der zweiten Textseite einige Änderungen erfordert, die in diesem Zusammenhang nicht erläutert werden sollen. Befassen wir uns also nur näher mit der Textseite 1.

Jedes Zeichen wird repräsentiert durch seinen ASCII (American Standard Code for Information Interchange)-Wert. Dem Apple II Reference Manual entnehmen wir, wie jeder Code von $00 bis $FF auf dem Bildschirm dargestellt wird. Jeder Großbuchstabe ist viermal vorhanden, entsprechend NORMAL, INVERSE und FLASH sowie als nicht-sichtbares Ctrl-Zeichen. Zahlen existieren drei- oder viermal, je nachdem ob der Apple Kleinbuchstaben anzeigen kann oder nicht, denn die Codes von $E0 bis $FE sind im Apple IIe oder im umgerüsteten II Plus mit dem kleinen normalen Alphabet belegt. So ist es auch verständlich, warum wir keine blinkenden oder inversen Kleinbuchstaben haben.

(Diese Angaben beziehen sich vornehmlich auf den Apple II Plus mit Kleinschreibumrüstsatz. Beim IIe und IIc gibt es erstens gegenüber dem Apple II einen zusätzlichen, „alternativen" Zeichensatz ALTCHAR und zweitens sind diese alternativen Zeichensätze beim IIe und IIc nicht identisch wegen der sog. Maus-Sonderzeichen beim IIc. Auf die diesbezüglichen Unterschiede wird in einem gesonderten Screen-Dump-Aufsatz eingegangen. Anm. d. Red.)

Aufeinanderfolgende Zeilen stehen nun leider nicht auch aufeinanderfolgend im Speicher. Vielmehr wird folgendes Muster verwendet:

Zeile	Adresse	Zeile	Adresse	Zeile	Adresse
0	$0400-427	8	$0428-44F	16	$0450-477
1	$0480-4A7	9	$04A8-4CF	17	$04D0-4F7
2	$0500-527	10	$0528-54F	18	$0550-577
3	$0580-5A7	11	$05A8-5CF	19	$05D0-5F7
4	$0600-627	12	$0628-64F	20	$0650-677
5	$0680-6A7	13	$06A8-6CF	21	$06D0-6F7
6	$0700-727	14	$0728-74F	22	$0750-777
7	$0780-7A7	15	$07A8-7CF	23	$07D0-7F7

Wenn wir horizontal durch diese Tabelle schauen, so sehen wir, daß immer 7 Zeilen übersprungen werden. $0400-$047F stellt also die Zeilen 0, 8 und 16 dar. Es bleiben 8 Bytes übrig, die nicht gesehen werden können und die als sogenanntes „Scratchpad"-Stellen oder „Screen-Holes" den Erweiterungskarten zugeordnet sind.

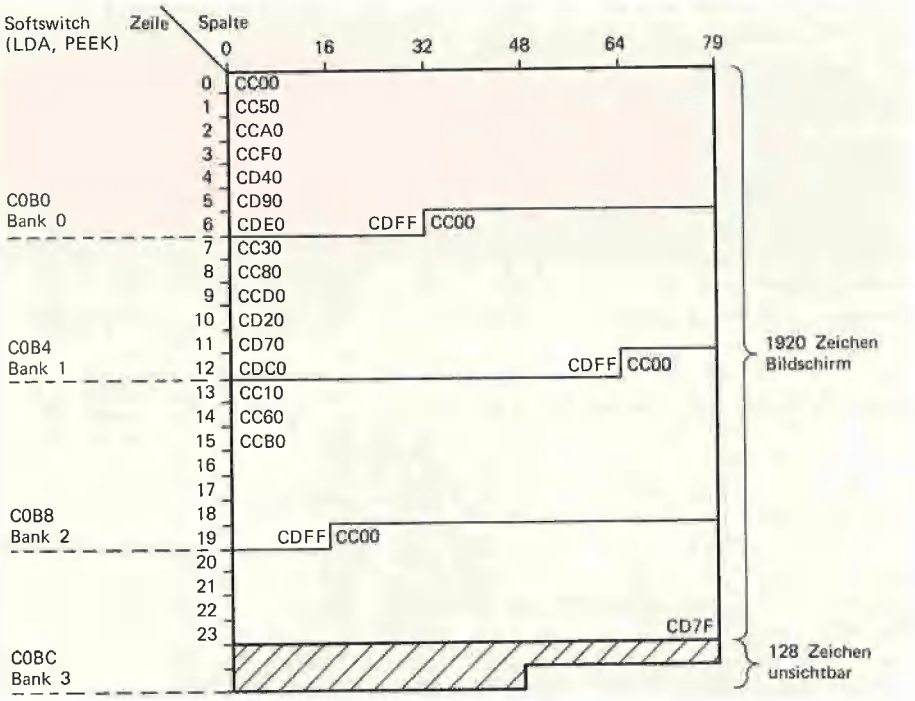


Diagramm 1: Organisation des Videx-Bildschirms

Speichern des Bildschirms

Wollen wir den Bildschirm-Inhalt als Textfile speichern, so müssen wir diesem Muster folgen, da sonst die Zeilen durcheinandergeraten. Für jede Zeile muß das Maschinenprogramm die Anfangsadresse errechnen. Zum Glück besorgt das für uns wieder eine Routine im Apple-Monitor namens BASCALC bei $FBC1, die ihr Ergebnis in BASL, BASH ($0028, $0029) ablegt. Um BASCALC mitzuteilen, welche Zeile wir lesen wollen, benutzen wir die Routine VTABZ ($FC24). Die Zeilennummer muß beim Aufruf im Akkumulator stehen (deshalb der TXA-Befehl). Die Startadresse wird dann nach $0028-$0029 geschrieben, von wo wir durch indirekt indizierte Adressierung mit dem Y-Register die folgenden 40 Zeichen lesen und ausgeben. Dabei werden alle Zeichen in NORMAL umgewandelt, Ctrl-Zeichen werden zu Leerzeichen gemacht.

DOS und DIRECT MODE

Soweit ist unser Programm jetzt fertig, wir müssen nur noch die Diskettenbefehle geben. Wir tun dies genauso, wie wir es in BASIC auch gemacht hätten, und zwar mit
Ctrl-D OPEN SCHIRMTEXT <Return>
Ctrl-D WRITE SCHIRMTEXT <Return>
Ctrl-D CLOSE <Return> am Ende.
Wir drucken die Befehle in einer Schleife über COUT aus, wodurch sie von DOS erkannt werden können. Dieses Verfahren funktioniert leider nur mit DOS 3.3 und nicht mit ProDOS. (Ein Screen-Dump-Programm für ProDOS ist ebenfalls in Vorbereitung. Anm. d. Red.)
Würden wir unser Programm jetzt starten, so erhielten wir nicht den gewünschten Textfile, weil noch drei Fehler vorliegen. Der erste hat damit zu tun, daß wir unser Programm auch über Reset starten wollen. Dadurch wird DOS abgekoppelt, so daß keine Diskettenbefehle mehr möglich sind. Die Routine DOSHOOK oder CONNECT ($03EA) verbindet DOS wieder mit dem Informationsfluß. Trotzdem bekämen wir eine Fehlermeldung, nämlich „NOT DIRECT COMMAND". OPEN und WRITE läßt DOS nämlich nur aus einem laufenden BASIC-Programm zu und nicht als direkte Tastatureingabe. Ein Maschinenprogramm ist für DOS aber wie eine Tastatureingabe. Also müssen wir ein laufendes Programm vortäuschen. Mit den Befehlen in Zeile 38 bis 44 wird dies erreicht. In einem BASIC-Programm enthält $AAB6 eine Codezahl für das verwendete BASIC ($40 = Applesoft ROM), CURLIN und CURLIN+1 die laufende Zeilennummer. Befinden wir uns nicht in einem Programm, ist CURLIN+1 gleich $FF. Das normale Promptzeichen Ü = $DD wird in $0033 abgespeichert und ändert sich bei einem laufenden Programm in $06. Schließlich ist Bit 7 bei $00D9 = RUNMODE in einem Programm gesetzt. Bringen wir diese kleinen Änderungen an, glaubt DOS, ein BASIC-Programm auszuführen, und alles läuft so, wie von uns geplant.
Einen kleinen Schönheitsfehler hat das Programm immer noch. Wenn wir Reset drücken oder es mit einem CALL 798 starten, springt der Bildschirm um eine Zeile hoch, so daß uns die obersten 40 Zeichen verlorengehen. Dies liegt nicht an der Reset-Routine selbst, sondern an der Tatsache, daß wir vor einem Diskettenbefehl ein Return (= $8D) ausgeben müssen. Wieder hilft ein Trick weiter. Mit den Befehlen in Zeile 34 bis 37 koppeln wir schlicht und einfach den Bildschirm ab. Wenn der Apple ein Zeichen über die Routine COUT (= $FDED) sendet, schaut er in den Speicherstellen $0036 und $0037 (= CSWL, CSWH) nach, wohin er es weitergeben soll. Ohne DOS steht hier $FDF0 (= COUT1). DOS schaltet sich nun in den Informationsfluß ein, indem es den Zeiger in sich selber hinein nach $9EBD verweisen läßt. Hier werden die DOS-Befehle ausgefiltert und alles übrige normalerweise an COUT1 übergeben. Diese Weiterleitungsadresse (Vektor-Adresse) steht in $AA53 und $AA54. Schreiben wir hier einfach die Adresse einer Speicherstelle hinein, die eine RTS-Instruktion enthält, bleibt der Bildschirm unangetastet, DOS erhält aber noch immer alle Informationen.
SCHIRMDISK wurde für den Bereich von $0300 an assembliert und kann mit dem Befehl
BSAVE SCHIRMDISK, A$0300, L$00B8
abgespeichert werden.

Der Videx-Videoterm-Bildschirm

Wenden wir uns nun der zweiten Aufgabe zu, den Schirm einer Videx-80-Zeichen-

karte komplett abzuspeichern. Wieder ist es unser erstes Ziel, den Aufbau des Bildes zu verstehen.
Der Videoterm-Bildschirm ist grundsätzlich anders organisiert als der Apple-40-Zeichenschirm. Der komplette Bildschirmspeicher liegt auf der Videoterm-Karte in einem RAM von 2K Umfang. Er ist in die vier Bänke 0-3 aufgeteilt, die jeweils den Adreßraum von $CC00-$CDFF belegen. $C800-$CBFF wird von der Videoterm-Firmware benutzt, $CE00-$CFFF ist frei. Alle Zeichen stehen sequentiell im Speicher, d.h. aufeinanderfolgende Zeichen liegen hintereinander, die Zeilen folgen einander. Von den 2048 Bytes (= 2K) sind immer nur 80 · 24 = 1920 Bytes sichtbar. Die vier Bänke werden aktiviert durch eine Leseoperation (LDA oder PEEK) der Speicherstellen $C0B0 (Bank 0), $C0B4 (Bank 1), $C0B8 (Bank 2) oder $C0BC (Bank 3). Insoweit ist die Videx-Karte einfacher zu verstehen als der Apple-Bildschirm. Kompliziert wird alles erst durch die Tatsache, daß den Bildschirmpositionen *keine* festen Speicherplätze zugeordnet sind, sondern der Bildanfang irgendwo in einer der vier Speicherbänke liegt. Diese Methode wurde gewählt, um ein einfaches Auf- und Abwärts-Scrollen zu ermöglichen. Die variable Bildstartadresse, der Angelpunkt des Bildes, wird an einer definierten Stelle abgelegt ($06FB), um für alle Positionsberechnungen zur Verfügung zu stehen. Ganz beliebig liegt der Start nun aber auch wieder nicht, denn er kann sich immer nur in Sprüngen von 80 Zeichen (= 1 Zeile) fortbewegen. Damit erhalten wir als mögliche Startpositionen $0000, $0050, $00A0, $00F0, $0140, $0190 usw. Wie Sie leicht sehen können, ändern sich nur die beiden mittleren Ziffern, und genau diese werden in das eine Byte bei $06FB gepackt.

Die Cursorposition zählt horizontal (CHORZ) von 0 bis 79 ($00-$49) und vertikal (CVERT) von 0 bis 23 ($00-$17). Die Startadresse der *augenblicklich* bearbeiteten Zeile, die im Bereich von $0000 bis $07FF einschließlich liegt, ist in BASEL und BASEH abgelegt. Befindet sich der Cursor in der ersten Position der ersten Zeile (HOME Cursor), so sind Bildstart und Zeilenstart identisch. Wir werden diese Tatsache benutzen, um eine einfache Methode zum Abspeichern des Bildschirms zu entwickeln.
Fassen wir noch einmal die wichtigsten Adressen zusammen, wie sie für eine Videx-Videoterm in Slot 3 gelten:

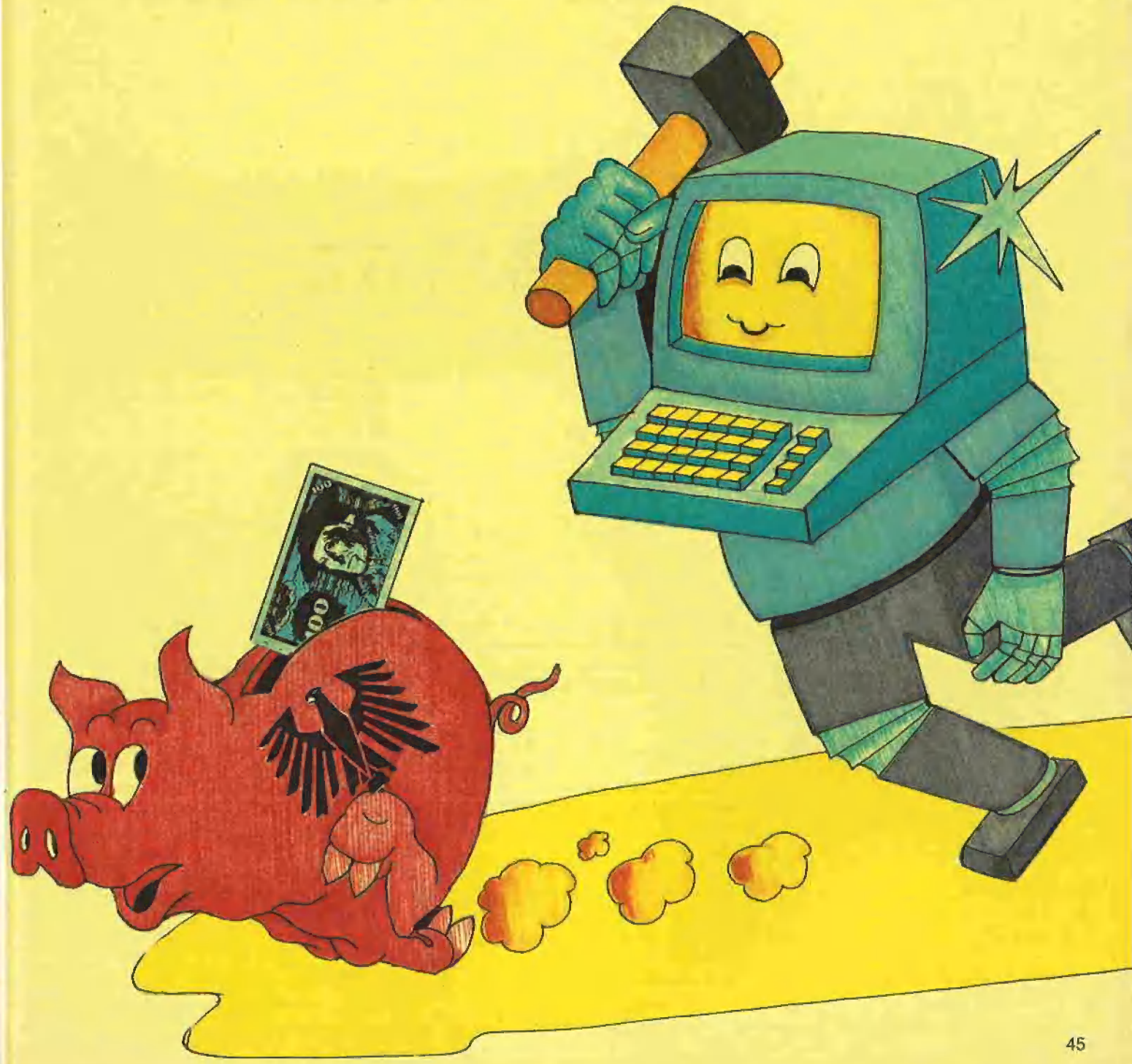
BASEL	$047B	Low Byte aktuelle Zeile
BASEH	$04FB	High Byte aktuelle Zeile

```
SCHIRMDISK

BSAVE SCHIRMDISK, A$0300, L$00B8
Mit BRUN SCHIRMDISK initialisieren
Mit Ctrl-Reset oder CALL 798 starten

                      1  * SCHIRMDISK
                      2  *
                      3  * Apple 40-Z/Z-Schirm -> Textfile
                      4  *
AA53                  5  DOUT     EQU $AA53
FC22                  6  VTAB     EQU $FC22
FC24                  7  VTABZ    EQU $FC24
FDED                  8  COUT     EQU $FDED
FE93                  9  SETVID   EQU $FE93
FB6F                 10  SETPWRC  EQU $FB6F
AAB6                 11  LANGUA   EQU $AAB6
FF58                 12  RETURN   EQU $FF58
0075                 13  CURLIN   EPZ $75
00D9                 14  RUNMODE  EPZ $D9
0033                 15  CURSOR   EPZ $33
03F2                 16  SOFTEV   EQU $3F2
03D0                 17  DOS      EQU $3D0
03EA                 18  DOSHOOK  EQU $3EA
0028                 19  BASL     EPZ $28
                     20  *
0300                 21           ORG $300          ;Läuft bei $300
0300                 22           OBJ $800
0300                 23  *
0300 D8              24  START    CLD               ;Binärmodus
0301 A9 1E           25           LDA #ENTRY        ;Reset-Vektor auf
0303 8D F2 03        26           STA SOFTEV
0306 A9 03           27           LDA /ENTRY        ;Entry setzen
0308 8D F3 03        28           STA SOFTEV+1
030B 20 6F FB        29           JSR SETPWRC       ;Power-up-Byte setzen
030E 60              30           RTS
030F B1 B9 B8        31           ASC "1984 DR. KEHREL"
0312 B4 A0 C4
0315 D2 AE A0
0318 CB C5 C8
031B D2 C5 CC
031E                 32  *
031E 20 EA 03        33  ENTRY    JSR DOSHOOK
0321 A9 58           34           LDA #RETURN       ;Bildschirm
0323 8D 53 AA        35           STA DOUT          ;abkoppeln
0326 A9 FF           36           LDA /RETURN
0328 8D 54 AA        37           STA DOUT+1
032B A9 40           38           LDA #$40
032D 8D B6 AA        39           STA LANGUA        ;ROM Applesoft
0330 85 76           40           STA CURLIN+1      ;laufendes Programm
0332 A9 06           41           LDA #6
0334 85 33           42           STA CURSOR
0336 A9 80           43           LDA #$80
0338 85 D9           44           STA RUNMODE
033A A2 00           45           LDX #0
033C BD 8C 03        46  DISK     LDA TEXT,X        ;Disk-Befehle
033F F0 06           47           BEQ DRUCK
0341 20 ED FD        48           JSR COUT
0344 E8              49           INX
0345 D0 F5           50           BNE DISK
0347                 51  *
0347 A2 00           52  DRUCK    LDX #0
0349 8A              53  TAB      TXA
034A 20 24 FC        54           JSR VTABZ         ;Berechne Adresse
034D A0 00           55           LDY #0
034F B1 28           56  LADEN    LDA (BASL),Y
0351 C9 20           57           CMP #$20
0353 B0 02           58           BCS P1
0355 09 C0           59           ORA #$C0          ;$00-1F -> $C0-DF
0357 C9 60           60  P1       CMP #$60
0359 B0 02           61           BCS P2
035B 09 80           62           ORA #$80          ;$20-5F -> $A0-DF
035D C9 80           63  P2       CMP #$80
035F B0 02           64           BCS P3
0361 49 C0           65           EOR #$C0          ;$60-7F -> $A0-BF
0363 C9 A0           66  P3       CMP #$A0
0365 B0 02           67           BCS SENDEN
0367 A9 A0           68           LDA #$A0          ;$80-9F -> $A0
0369 20 ED FD        69  SENDEN   JSR COUT
036C C8              70           INY
036D C0 28           71           CPY #40           ;Ganze Zeile fertig?
036F D0 DE           72           BNE LADEN
0371 E8              73           INX
0372 E0 18           74           CPX #24           ;alle Zeilen fertig?
0374 D0 D3           75           BNE TAB
0376 A2 00           76           LDX #0
0378 BD AF 03        77  CLOSE    LDA TEXT1,X       ;Disk-Befehl
```

Fortsetzung auf Seite 61

Lohn- und Einkommensteuer 1984

Ein umfassendes CP/M-MBASIC-Programm



Lohn- und Einkommensteuer 1984

Ein umfassendes CP/M-MBASIC-Programm

von Volker Duske

Unter dem Motto – alle Jahre wieder, da schlachten wir ein Schwein – ist es auch jetzt wieder an der Zeit, vom Vater Staat die Steuern zurückzuverlangen. Abertausende von Arbeitnehmern schenken dem Staat jährlich Millionen Deutsche Mark, indem sie gesetzlich verankerte Steuervorteile nicht ausschöpfen. Sie nehmen weder die ihnen zustehenden Freibeträge für das laufende Jahr noch den Lohn/Einkommensteuer-Jahresausgleich in Anspruch, oder sie nutzen mangels entsprechender Kenntnisse und Aufklärung die Ihnen zustehenden Vergünstigungen nicht voll aus.

Noch schwieriger wird es mit der Berechnung der Steuer. Dieses Programm soll Ihnen eine Hilfestellung geben. Das Programm gibt keine Ratschläge in Rechtsfragen, sondern ermöglicht nur dem Laien eine schnelle Berechnung des zu versteuernden Einkommens. Wer also dieses Programm benutzt, muß sich im klaren darüber sein, welchen Betrag er wo ansetzen kann oder nicht.

Das Programm ist in Microsoft Basic (MBASIC) geschrieben und erfordert das Betriebssystem CP/M 56 und folgende Hardware:

Apple II+/e
Monitor
Z80-Karte
80-Zeichen-Karte
1 Diskettenlaufwerk

Um einige Programme schneller zur Verfügung zu haben, könnte man eine 128K/256K-Karte als Pseudo-Floppy benutzen, was jedoch nicht unbedingt erforderlich ist.

Wenn Sie das abgebildete Listing eintippen möchten, hier einige wichtige Hinweise:

Da das Programm sehr lang ist und MBASIC allein 24K in Anspruch nimmt, ist der Speicherplatz für Programm und Daten recht gering. Dies wurde durch etliche Unterprogramme umgangen, die mit A, B, C, ... N, PASS, MENUE und HELP bezeichnet wurden. Die Übergabe der Variablen erfolgt durch die Anweisungen COMMON und CHAIN. Die Variablen sind im Programm PASS durch die DIM-Anweisungen dimensioniert, d.h. dies ist zugleich das spätere Startprogramm.
Der Code-Name „PEEKERDUSKE" muß eingegeben werden, da sonst alle Programme außer MENUE und MBASIC gelöscht werden. Falls Sie sich die Mühe machen wollen und das Programm abtippen, so können Sie einige Unterprogramme weglassen, wenn diese nicht bei Ihrem Antrag benötigt werden. Ein kleines Beispiel:

Sie haben keine Einkünfte aus Gewerbebetrieb oder aus Kapitalvermögen. Damit brauchen Sie die entsprechenden Unterprogramme nicht einzutippen. Das gilt auch für andere Unterprogramme, z.B. HELP. Allerdings ist es ratsam, sich zunächst einmal diese Help-Menüs durchzulesen, da hier sämtliche Instruktionen zum Arbeiten mit dem Steuerprogramm zu ersehen sind. (Das HELP-Modul als Programm bzw. Listing befindet sich nur auf der Peeker-Sammeldiskette. Statt dessen haben wir hier der besseren Übersicht halber die Bildschirm-Menüs selbst reproduziert. Anm. d. Red.)

Abschließend noch ein wichtiger Hinweis: Das vorliegende Programm kann trotz mehrfacher Überprüfung Fehler enthalten. Dies sei nicht nur zum Schutz des Autors und der Redaktion erwähnt, sondern auch zum Schutz vor übereiltem Handeln. Bitte erklären Sie bei Abweichungen der Rechenergebnisse dieses Programmes von denen des Finanzamtes den Steuerbescheid nicht gleich für falsch, da es sich auch um rechtliche Differenzen handeln kann. Dieses Programm kann keine rechtlichen Entscheidungen treffen.

Das Programm kann als gesonderte Peeker-Sammeldiskette im CP/M-Format zum Einzelpreis von DM 28,- (bzw. DM 20,- für Peeker-Sammeldiskette-Abonnenten) erworben werden. Diese Sammeldiskette enthält alle Programmdateien, nicht jedoch MBASIC, das zuvor von der eigenen Systemdiskette geladen werden muß. Danach wird das Steuerprogramm gestartet mit:

A> PASS

Programm Lohn- und Einkommensteuer

PASS.BAS: Password

```
1000 HOME
1005 COMMON A(), B(), C(), D(), E(), M(), N()
1010 VTAB 8:HTAB 35:PRINT"STEUER '84"
1015 VTAB 10:HTAB 33:PRINT"PEEKER-SOFTWARE"
1020 HTAB 38:PRINT"1984"
1025 HTAB 34:PRINT"Volker Duske"
1030 HTAB 32:PRINT"Kieler Str.309 B"
1035 HTAB 32:PRINT"2350 Neumünster"
1040 HTAB 32:PRINT" --------- "
1045 HTAB 32:PRINT"Dr. Hüthig Verlag"
1050 HTAB 32:PRINT"Tel.: 06221/489-1"
1055 HTAB 32:PRINT"Serial #.: d00002"
1060 HTAB 32:PRINT"Passwort : ";
1065 CODE$=""
1070 FOR I = 1 TO 11
1075 T$=INKEY$
1080 IF T$="" THEN 1075
1085 CODE$=CODE$+T$
1090 NEXT I
1095 IF CODE$="PEEKERDUSKE" THEN 1125
1100 HOME
1105 VTAB 3:HTAB 10:PRINT"... bitte warten !!!"
1110 KILL "N.BAS": KILL "PASS.BAS": KILL "A.BAS": KILL
     "B.BAS": KILL "C.BAS"
1115 KILL "D.BAS": KILL "E.BAS": KILL "F.BAS": KILL
     "G.BAS": KILL "H.BAS"
1120 KILL "I.BAS": KILL "J.BAS": KILL "K.BAS": KILL
     "L.BAS": KILL "M.BAS"
1125 DIM A(120), B(120), C(70), D(70), E(20), M(10), N(5)
1130 CHAIN "MENUE"
```

MENUE.BAS: Hauptmenü

```
1135 REM MENUE
1140 COMMON A(), B(), C(), D(), E(), M(), N()
1145 HOME:PRINT"Hauptmenue :":FOR I = 1 TO
     79:PRINT"-";:NEXT I:PRINT
1150 VTAB 4:PRINT" <A> Grundangaben"
1155 PRINT" <B> Einkünfte aus Land- und Forstwirtschaft"
1160 PRINT" <C> Einkünfte aus Gewerbebetrieb"
1165 PRINT" <D> Einkünfte aus selbständiger Arbeit"
1170 PRINT" <E> Einkünfte aus nichtselbständiger Arbeit"
1175 PRINT" <F> Einkünfte aus Kapitalvermögen"
1180 PRINT" <G> Einkünfte aus Vermietung und Verpachtung"
1185 PRINT" <H> Sonstige Einkünfte"
1190 PRINT" <I> Doppelbesteuerung u. außerordentliche
     Einkünfte"
1195 PRINT" <J> Sonderausgaben"
1200 PRINT" <K> Außergewöhnliche Belastungen"
1205 PRINT" <L> Besondere Abzugsbeträge u. ausländische
     Steuern"
1210 PRINT" <M> Einkünfte zur An- u. Abrechnung"
1215 PRINT" <N> Berechnungen durchführen"
1220 PRINT:PRINT" <?> HELP
1225 PRINT" <X> NEUER DURCHLAUF"
1230 PRINT" <Z> ENDE - zurück ins Betriebssystem"
1235 IF M(1) = 1 THEN VTAB 4:HTAB 11:PRINT"*"
1240 IF M(2) = 1 THEN VTAB 5:HTAB 11:PRINT"*"
1245 IF M(3) = 1 THEN VTAB 6:HTAB 11:PRINT"*"
1250 IF M(4) = 1 THEN VTAB 7:HTAB 11:PRINT"*"
1255 IF M(5) = 1 THEN VTAB 8:HTAB 11:PRINT"*"
1260 IF M(6) = 1 THEN VTAB 9:HTAB 11:PRINT"*"
1265 IF M(7) = 1 THEN VTAB 10:HTAB 11:PRINT"*"
1270 IF M(8) = 1 THEN VTAB 11:HTAB 11:PRINT"*"
1275 IF M(9) = 1 THEN VTAB 12:HTAB 11:PRINT"*"
1280 IF M(10) = 1 THEN VTAB 13:HTAB 11:PRINT"*"
1285 IF N(1) = 1 THEN VTAB 14:HTAB 11:PRINT"*"
1290 IF N(2) = 1 THEN VTAB 15:HTAB 11:PRINT"*"
1295 IF N(3) = 1 THEN VTAB 16:HTAB 11:PRINT"*"
1300 IF N(4) = 1 THEN VTAB 17:HTAB 11:PRINT"*"
1305 VTAB 23:HTAB 20:PRINT"Bitte wählen Sie <A - M, ?, X,
     Z> -----> < >"
1310 WTAGE$="ABCDEFGHIJKLMN?XZ"
1315 VTAB 23:HTAB 67:GET WO$
1320 VTAB 23:HTAB 67:PRINT WO$
1325 P=INSTR(WTAGE$,WO$)
1330 IF P=0 THEN 1315
1335 P1=P
1340 ON P1 GOTO
     1350,1350,1350,1350,1350,1350,1350,1350,1350,1350,135
     0,1350,1350,1350,1350,1350,1435
1345 GOTO 1145
1350 FOR I = 1 TO 3:VTAB 1:HTAB 60:PRINT SPC(16);:FOR S =
     1 TO 100:NEXT S:VTAB 1:HTAB 60:PRINT"Bitte warten
     ...":FOR S1 = 1 TO 100:NEXT S1:NEXT I
1355 IF P1 = 1 THEN M(1) = 1:CHAIN "A"
1360 IF P1 = 2 THEN M(2) = 1:CHAIN "B"
1365 IF P1 = 3 THEN M(3) = 1:CHAIN "C"
1370 IF P1 = 4 THEN M(4) = 1:CHAIN "D"
1375 IF P1 = 5 THEN M(5) = 1:CHAIN "E"
1380 IF P1 = 6 THEN M(6) = 1:CHAIN "F"
1385 IF P1 = 7 THEN M(7) = 1:CHAIN "G"
1390 IF P1 = 8 THEN M(8) = 1:CHAIN "H"
1395 IF P1 = 9 THEN M(9) = 1:CHAIN "I"
1400 IF P1 = 10 THEN M(10) = 1:CHAIN "J"
1405 IF P1 = 11 THEN N(1) = 1:CHAIN "K"
1410 IF P1 = 12 THEN N(2) = 1:CHAIN "L"
1415 IF P1 = 13 THEN N(3) = 1:CHAIN "M"
1420 IF P1 = 14 THEN N(4) = 1:CHAIN "N"
1425 IF P1 = 15 THEN CHAIN "HELP"
1430 IF P1 = 16 THEN CLEAR:RUN "PASS"
1435 IF P1 = 17 THEN HOME:SYSTEM
```

A.BAS: Grundangaben

```
1440 REM GRUNDANGABEN
1445 COMMON A(), B(), C(), D(), E(), M(), N()
1450 HOME:PRINT"Grundangaben :"
1455 FOR I = 1 TO 79:PRINT"-";:NEXT I:PRINT
1460 VTAB 5:PRINT"Familienstand : <1> ledig, <2>
     verheiratet ..........."
1465 PRINT"Geburtsdatum (TTMMJJ) : steuerpflichtiger
     Ehemann ..........."
1470 PRINT"Geburtsdatum (TTMMJJ) : steuerpflichtige
     Ehefrau ..........."
1475 PRINT"Religion : <0> keine, <1> ev., <2> rk. Ehemann
     ..."
1480 PRINT"Religion : <0> keine, <1> ev., <2> rk. Ehefrau
     ..."
1485 PRINT"Kinder : Anzahl (1, 2, 3, 4, ...)
     ..............."
1490 PRINT"Zahlkinder : Anzahl (1, 2, 3, 4, ...)
     ..............."
1495 PRINT"Witwensplitting : <J> ja, <N> nein
     ....................."
1500 PRINT:PRINT"Bestehen für die Ehefrau Einkünfte oder
     Ausgaben, die in diesem"
1505 PRINT"Antrag berücksichtigt werden müssen ?
     ............... <J/N> ..."
1510 VTAB 5:HTAB 65:INPUT"",A(1)
1515 IF A(1) = 1 OR A(1) = 2 THEN VTAB 5:HTAB 65:PRINT"
     ";A(1):ELSE 1510
1520 VTAB 6:HTAB 65:INPUT"",A$
1525
     AS1=VAL(RIGHT$(A$,2)):AS2=VAL(MID$(A$,3,2)):AS3=VAL(I
     EFT$(A$,2))
1530 IF LEN(A$) = 6 AND AS1 < 99 AND AS2 < 13 AND AS3 <
     32 THEN VTAB 6:HTAB 65:PRINT" ";A$:ELSE 1520
1535 A=VAL(RIGHT$(A$,2)+MID$(A$,3,2)+LEFT$(A$,2))
1540 A(2)=1
1545 IF A < 350102! THEN A(2)=5
1550 IF A < 200102! THEN A(2)=7
1555 IF A(1) = 1 THEN VTAB 7:HTAB 74:PRINT"------";:GOTO
     1595
1560 VTAB 7:HTAB 65:INPUT"",A$
1565
     AS1=VAL(RIGHT$(A$,2)):AS2=VAL(MID$(A$,3,2)):AS3=VAL(I
     EFT$(A$,2))
1570 IF LEN(A$) = 6 AND AS1 < 99 AND AS2 < 13 AND AS3 <
     32 THEN VTAB 7:HTAB 65:PRINT" ";A$:ELSE 1560
1575 A=VAL(RIGHT$(A$,2)+MID$(A$,3,2)+LEFT$(A$,2))
1580 B(2)=1
1585 IF A < 350102! THEN B(2)=5
1590 IF A < 200102! THEN B(2)=7
1595 VTAB 8:HTAB 65:INPUT"",A(3)
1600 IF A(3) = 0 OR A(3) = 1 OR A(3) = 2 THEN VTAB 8:HTAB
     65:PRINT" ";A(3):ELSE 1595
1605 IF A(1) = 1 THEN VTAB 9:HTAB 79:PRINT"-";:GOTO 1620
1610 VTAB 9:HTAB 65:INPUT"",B(3)
1615 IF B(3) = 0 OR B(3) = 1 OR B(3) = 2 THEN VTAB 9:HTAB
     65:PRINT" ";B(3):ELSE 1610
1620 VTAB 10:HTAB 65:INPUT"",A(4)
1625 VTAB 10:HTAB 65:PRINT" ";A(4)
1630 VTAB 11:HTAB 65:INPUT"",A(5)
1635 VTAB 11:HTAB 65:PRINT" ";A(5)
1640 IF A(1) = 1 THEN 1680
1645 VTAB 12:HTAB 65:GET Y$
1650 IF Y$ = "J" OR Y$ = "N" THEN VTAB 12:HTAB 65:PRINT"
     ";Y$:ELSE 1645
1655 IF Y$ = "J" THEN A(6) = 1:IF A(6) = 1 THEN E(1) = 1
1660 VTAB 15:HTAB 65:GET Y$
1665 IF Y$ = "J" OR Y$ = "N" THEN VTAB 15:HTAB 65:PRINT"
     ";Y$:ELSE 1660
1670 IF Y$ = "N" THEN A(118) = 1
1675 IF Y$ = "N" THEN A(1) = 1
1680 VTAB 23:PRINT"Eingabefehler : sind alle Angaben
     richtig ? <J/N> ...."
1685 VTAB 23:HTAB 65:GET E$
1690 IF E$ = "J" OR E$ = "N" THEN VTAB 23:HTAB 65:PRINT"
     ";E$:ELSE 1685
1695 IF E$ = "N" THEN VTAB 3:HTAB 65:INVERSE:PRINT"->
     KORREKTUR <-":NORMAL
1700 IF E$ = "N" THEN VTAB 23:HTAB 79:PRINT" ";:GOTO 1510
1705 VTAB 3:HTAB 65:PRINT" "
1710 FOR I = 1 TO 3:VTAB 1:HTAB 60:PRINT SPC(16);:FOR S =
```

```
    1 TO 100:NEXT S:VTAB 1:HTAB 60:PRINT"Bitte warten
    ...":FOR S1 = 1 TO 100:NEXT S1:NEXT I
1715 CHAIN "MENUE"
```

B.BAS: Einkünfte aus Land- und Forstwirtschaft

```
1720 REM EINKUENFTE AUS LAND- UND FORSTWIRTWIRTSCHAFT
1725 COMMON A(), B(), C(), D(), E(), M(), N()
1730 HOME:PRINT"Einkünfte aus Land- und Forstwirtschaft
     :"
1735 FOR I = 1 TO 79:PRINT"-";:NEXT I:PRINT
1740 VTAB 7:PRINT"Gewinne : sämtliche Gewinne .........
     <in DM> ..."
1745 PRINT"darin Veräußerungen : nach Par.14.14a ESTG
     ..... <in DM> ..."
1750 PRINT"Angabe der Ermäßigung : 1. Betr.: Gewinn 83/84
     .... <in DM> ..."
1755 PRINT" - dito - : 1. Betr.: Anteil am Betrag. <in
     DM> ..."
1760 PRINT" - dito - : 2. Betr.: Gewinn 83/84 .... <in
     DM> ..."
1765 PRINT" - dito - : 2. Betr.: Anteil am Betrag. <in
     DM> ..."
1770 VTAB 4:HTAB 65:INVERSE:PRINT"--> EHEMANN <--":NORMAL
1775 VTAB 7:HTAB 65:INPUT"",A(15)
1780 VTAB 7:HTAB 65:PRINT" ":VTAB 7:HTAB 65:PRINT USING
     "################";A(15)
1785 HTAB 65:INPUT"",A(16)
1790 VTAB 8:HTAB 65:PRINT" ":VTAB 8:HTAB 65:PRINT USING
     "################";A(16)
1795 IF A(15) = 0 THEN 1840
1800 HTAB 65:INPUT"",A(103)
1805 VTAB 9:HTAB 65:PRINT" ":VTAB 9:HTAB 65:PRINT USING
     "################";A(103)
1810 HTAB 65:INPUT"",A(104)
1815 VTAB 10:HTAB 65:PRINT" ":VTAB 10:HTAB 65:PRINT USING
     "################";A(104)
1820 HTAB 65:INPUT"",A(105)
1825 VTAB 11:HTAB 65:PRINT" ":VTAB 11:HTAB 65:PRINT USING
     "################";A(105)
1830 HTAB 65:INPUT"",A(106)
1835 VTAB 12:HTAB 65:PRINT" ":VTAB 12:HTAB 65:PRINT USING
     "################";A(106)
1840 VTAB 23:PRINT"Eingabefehler : sind alle Angaben
     richtig ? <J/N> .... ";:GET E$
1845 IF E$ = "J" OR E$ = "N" THEN VTAB 23:HTAB 65:PRINT"
     ";E$:ELSE 1840
1850 IF E$ = "N" THEN VTAB 3:HTAB 65:INVERSE:PRINT"->
     KORREKTUR <-":NORMAL
1855 IF E$ = "N" THEN VTAB 23:HTAB 79:PRINT " ";:GOTO
     1775
1860 VTAB 3:HTAB 65:PRINT SPC(15)
1865 IF A(1) = 1 THEN 1985:
1870 IF A(118) = 1 THEN 1985
1875 VTAB 22:PRINT SPC(79):HTAB 79:PRINT" "
1880 VTAB 4:HTAB 65:INVERSE:PRINT"--> EHEFRAU <--":NORMAL
1885 VTAB 7:HTAB 70:PRINT SPC(10):VTAB 8:HTAB 70:PRINT
     SPC(10):VTAB 9:HTAB 70:PRINT SPC(10)
1890 VTAB 10:HTAB 70:PRINT SPC(10):VTAB 11:HTAB 70:PRINT
     SPC(10):VTAB 12:HTAB 70:PRINT SPC(10)
1895 VTAB 7:HTAB 65:INPUT"",B(15)
1900 VTAB 7:HTAB 65:PRINT" ":VTAB 7:HTAB 65:PRINT USING
     "################";B(15)
1905 HTAB 65:INPUT"",B(16)
1910 VTAB 8:HTAB 65:PRINT" ":VTAB 8:HTAB 65:PRINT USING
     "################";B(16)
1915 IF B(15) = 0 THEN 1960
1920 HTAB 65:INPUT"",B(103)
1925 VTAB 9:HTAB 65:PRINT" ":VTAB 9:HTAB 65:PRINT USING
     "################";B(103)
1930 HTAB 65:INPUT"",B(104)
1935 VTAB 10:HTAB 65:PRINT" ":VTAB 10:HTAB 65:PRINT USING
     "################";B(104)
1940 HTAB 65:INPUT"",B(105)
1945 VTAB 11:HTAB 65:PRINT" ":VTAB 11:HTAB 65:PRINT USING
     "################";B(105)
1950 HTAB 65:INPUT"",B(106)
1955 VTAB 12:HTAB 65:PRINT" ":VTAB 12:HTAB 65:PRINT USING
     "################";B(106)
1960 VTAB 23:PRINT"Eingabefehler : sind alle Angaben
     richtig ? <J/N> .... ";:GET E$
1965 IF E$ = "J" OR E$ = "N" THEN VTAB 23:HTAB 65:PRINT"
     ";E$:ELSE 1960
1970 IF E$ = "N" THEN VTAB 3:HTAB 65:INVERSE:PRINT"->
     KORREKTUR <-":NORMAL
1975 IF E$ = "N" THEN VTAB 23:HTAB 79:PRINT " ";:GOTO
     1895
1980 VTAB 3:HTAB 65:PRINT" "
1985 FOR I = 1 TO 3:VTAB 1:HTAB 60:PRINT SPC(16);:FOR S =
     1 TO 100:NEXT S:VTAB 1:HTAB 60:PRINT"Bitte warten
     ...":FOR S1 = 1 TO 100:NEXT S1:NEXT I
1990 CHAIN "MENUE"
```

C.BAS: Einkünfte aus Gewerbebetrieb

```
1995 REM EINKUENFTE AUS GEWERBEBETRIEB
2000 COMMON A(), B(), C(), D(), E(), M(), N()
2005 HOME:PRINT"Einkünfte aus dem Gewerbebetrieb :"
2010 FOR I = 1 TO 79:PRINT"-";:NEXT I:PRINT
2015 VTAB 7:PRINT"Gewinne : sämtliche Gewinne .........
     <in DM> ..."
2020 PRINT"darin Veräußerungen : nach Par.16 u. 17 ESTG
     .... <in DM> ..."
2025 VTAB 4:HTAB 65:INVERSE:PRINT"--> EHEMANN <--":NORMAL
2030 VTAB 7:HTAB 65:INPUT"",A(20)
2035 VTAB 7:HTAB 65:PRINT" ":VTAB 7:HTAB 65:PRINT USING
     "################";A(20)
2040 HTAB 65:INPUT"",A(21)
2045 VTAB 8:HTAB 65:PRINT" ":VTAB 8:HTAB 65:PRINT USING
     "################";A(21)
2050 VTAB 23:PRINT"Eingabefehler : sind alle Angaben
     richtig ? <J/N> .... ";:GET E$
2055 IF E$ = "J" OR E$ = "N" THEN VTAB 23:HTAB 65:PRINT"
     ";E$:ELSE 2050
2060 IF E$ = "N" THEN VTAB 3:HTAB 65:INVERSE:PRINT"->
     KORREKTUR <-":NORMAL
2065 IF E$ = "N" THEN VTAB 23:HTAB 79:PRINT " ";:GOTO 2030
2070 VTAB 3:HTAB 65:PRINT SPC(15)
2075 IF A(1) = 1 THEN 2140
2080 VTAB 22:PRINT SPC(79):HTAB 79:PRINT" "
2085 VTAB 4:HTAB 65:INVERSE:PRINT"--> EHEFRAU <--":NORMAL
2090 VTAB 7:HTAB 70:PRINT SPC(10):VTAB 8:HTAB 70:PRINT
     SPC(10)
2095 VTAB 7:HTAB 65:INPUT"",B(20)
2100 VTAB 7:HTAB 65:PRINT" ":VTAB 7:HTAB 65:PRINT USING
     "################";B(20)
2105 VTAB 8:HTAB 65:INPUT"",B(21)
2110 VTAB 8:HTAB 65:PRINT" ":VTAB 8:HTAB 65:PRINT USING
     "################";B(21)
2115 VTAB 23:PRINT"Eingabefehler : sind alle Angaben
     richtig ? <J/N> .... ";:GET E$
2120 IF E$ = "J" OR E$ = "N" THEN VTAB 23:HTAB 65:PRINT"
     ";E$:ELSE 2115
2125 IF E$ = "N" THEN VTAB 3:HTAB 65:INVERSE:PRINT"->
     KORREKTUR <-":NORMAL
2130 IF E$ = "N" THEN VTAB 23:HTAB 79:PRINT " ";:GOTO
     2095
2135 VTAB 3:HTAB 65:PRINT SPC(15)
2140 FOR I = 1 TO 3:VTAB 1:HTAB 60:PRINT SPC(16);:FOR S =
     1 TO 100:NEXT S:VTAB 1:HTAB 60:PRINT"Bitte warten
     ...":FOR S1 = 1 TO 100:NEXT S1:NEXT I
2145 CHAIN "MENUE"
```

D.BAS: Einkünfte aus selbständiger Arbeit

```
2150 REM EINKUENFTE AUS SELBSTAENDIGER ARBEIT
2155 COMMON A(), B(), C(), D(), E(), M(), N()
2160 HOME:PRINT"Einkünfte aus selbständiger Arbeit :"
2165 FOR I = 1 TO 79:PRINT"-";:NEXT I:PRINT
2170 VTAB 7:PRINT"Gewinn : aus freiberuflicher Arbeit <in
     DM> ..."
2175 PRINT"Gewinn : aus sonstiger Tätigkeit ... <in DM>
     ..."
2180 PRINT"Werbungskosten : angefallene Werbungskosten
     <in DM> ..."
2185 PRINT"dar.Veräußerungs-Gew. : nach Par.18 Abs.3 ESTG
     .... <in DM> ..."
2190 VTAB 4:HTAB 65:INVERSE:PRINT"--> EHEMANN <--":NORMAL
2195 VTAB 7:HTAB 65:INPUT"",A(25)
2200 VTAB 7:HTAB 65:PRINT" ":VTAB 7:HTAB 65:PRINT USING
     "################";A(25)
2205 HTAB 65:INPUT"",A(26)
2210 VTAB 8:HTAB 65:PRINT" ":VTAB 8:HTAB 65:PRINT USING
     "################";A(26)
2215 HTAB 65:INPUT"",A(27)
2220 VTAB 9:HTAB 65:PRINT" ":VTAB 9:HTAB 65:PRINT USING
     "################";A(27)
2225 HTAB 65:INPUT"",A(28)
2230 VTAB 10:HTAB 65:PRINT" ":VTAB 10:HTAB 65:PRINT USING
     "################";A(28)
2235 VTAB 23:PRINT"Eingabefehler : sind alle Angaben
     richtig ? <J/N> .... ";:GET E$
2240 IF E$ = "J" OR E$ = "N" THEN VTAB 23:HTAB 65:PRINT"
     ";E$:ELSE 2235
2245 IF E$ = "N" THEN VTAB 3:HTAB 65:INVERSE:PRINT"->
     KORREKTUR <-":NORMAL
2250 IF E$ = "N" THEN VTAB 23:HTAB 79:PRINT " ";:GOTO
     2195
2255 VTAB 3:HTAB 65:PRINT SPC(15)
2260 IF A(1) = 1 THEN 2345
2265 VTAB 22:PRINT SPC(79):HTAB 79:PRINT" "
2270 VTAB 4:HTAB 65:INVERSE:PRINT"--> EHEFRAU <--":NORMAL
2275 VTAB 7:HTAB 70:PRINT SPC(10):VTAB 8:HTAB 70:PRINT
     SPC(10):VTAB 9:HTAB 70:PRINT SPC(10):VTAB 10:HTAB
     70:PRINT SPC(10)
```

```
2280 VTAB 7:HTAB 65:INPUT"",B(25)
2285 VTAB 7:HTAB 65:PRINT ":VTAB 7:HTAB 65:PRINT USING
     "################";B(25)
2290 HTAB 65:INPUT"",B(26)
2295 VTAB 8:HTAB 65:PRINT ":VTAB 8:HTAB 65:PRINT USING
     "################";B(26)
2300 HTAB 65:INPUT"",B(27)
2305 VTAB 9:HTAB 65:PRINT ":VTAB 9:HTAB 65:PRINT USING
     "################";B(27)
2310 HTAB 65:INPUT"",B(28)
2315 VTAB 10:HTAB 65:PRINT ":VTAB 10:HTAB 65:PRINT USING
     "################";B(28)
2320 VTAB 23:PRINT"Eingabefehler : sind alle Angaben
     richtig ? <J/N> .... ";:GET E$
2325 IF E$ = "J" OR E$ = "N" THEN VTAB 23:HTAB 65:PRINT"
     ";E$:ELSE 2320
2330 IF E$ = "N" THEN VTAB 3:HTAB 65:INVERSE:PRINT"->
     KORREKTUR <-":NORMAL
2335 IF E$ = "N" THEN VTAB 23:HTAB 79:PRINT " ";:GOTO
     2280
2340 VTAB 3:HTAB 65:PRINT SPC(15)
2345 FOR I = 1 TO 3:VTAB 1:HTAB 60:PRINT SPC(16);:FOR S =
     1 TO 100:NEXT S:VTAB 1:HTAB 60:PRINT"Bitte warten
     ...":FOR S1 = 1 TO 100:NEXT S1:NEXT I
2350 CHAIN "MENUE"
```

E.BAS: Einkünfte aus nichtselbständiger Tätigkeit

```
2355 REM EINKUENFTE AUS NICHTSELBSTAENDIGER TAETIGKEIT
2360 COMMON A(), B(), C(), D(), E(), M(), N()
2365 HOME:PRINT"Einkünfte aus nichtselbständiger Arbeit
     :"
2370 FOR I = 1 TO 79:PRINT"-";:NEXT I:PRINT
2375 VTAB 7:PRINT"Brutto-Arbeitslohn : lfd. Arbeitslohn
     ......... <in DM> ..."
2380 PRINT"Bezüge : Versorgungsbezüge ......... <in DM>
     ..."
2385 PRINT"Kosten : Werbungskosten ........... <in DM>
     ..."
2390 PRINT"Beträge für : ALG, SWG, KUG oder ALB .... <in
     DM> ..."
2395 VTAB 4:HTAB 65:INVERSE:PRINT"--> EHEMANN <--":NORMAL
2400 VTAB 7:HTAB 65:INPUT"",A(30)
2405 VTAB 7:HTAB 65:PRINT" ":VTAB 7:HTAB 65:PRINT USING
     "################";A(30)
2410 VTAB 8:HTAB 65:INPUT"",A(31)
2415 VTAB 8:HTAB 65:PRINT" ":VTAB 8:HTAB 65:PRINT USING
     "################";A(31)
2420 VTAB 9:HTAB 65:INPUT"",A(32)
2425 VTAB 9:HTAB 65:PRINT" ":VTAB 9:HTAB 65:PRINT USING
     "################";A(32)
2430 VTAB 10:HTAB 65:INPUT"",A(67)
2435 VTAB 10:HTAB 65:PRINT" ":VTAB 10:HTAB 65:PRINT USING
     "################";A(67)
2440 VTAB 23:PRINT"Eingabefehler : sind alle Angaben
     richtig ? <J/N> .... ";:GET E$
2445 IF E$ = "J" OR E$ = "N" THEN VTAB 23:HTAB 65:PRINT"
     ";E$:ELSE 2440
2450 IF E$ = "N" THEN VTAB 3:HTAB 65:INVERSE:PRINT"->
     KORREKTUR <-":NORMAL
2455 IF E$ = "N" THEN VTAB 23:HTAB 79:PRINT" ";:GOTO 2400
2460 VTAB 3:HTAB 65:PRINT SPC(15)
2465 IF A(1) = 1 THEN 2555
2470 VTAB 22:PRINT SPC(79):HTAB 79:PRINT" "
2475 VTAB 4:HTAB 65:INVERSE:PRINT"--> EHEFRAU <--":NORMAL
2480 VTAB 7:HTAB 70:PRINT SPC(10):VTAB 8:HTAB 70:PRINT
     SPC(10)
2485 VTAB 9:HTAB 70:PRINT SPC(10):VTAB 10:HTAB 70:PRINT
     SPC(10)
2490 VTAB 7:HTAB 65:INPUT"",B(30)
2495 VTAB 7:HTAB 65:PRINT" ":VTAB 7:HTAB 65:PRINT USING
     "################";B(30)
2500 VTAB 8:HTAB 65:INPUT"",B(31)
2505 VTAB 8:HTAB 65:PRINT" ":VTAB 8:HTAB 65:PRINT USING
     "################";B(31)
2510 VTAB 9:HTAB 65:INPUT"",B(32)
2515 VTAB 9:HTAB 65:PRINT" ":VTAB 9:HTAB 65:PRINT USING
     "################";B(32)
2520 VTAB 10:HTAB 65:INPUT"",B(67)
2525 VTAB 10:HTAB 65:PRINT" ":VTAB 10:HTAB 65:PRINT USING
     "################";B(67)
2530 VTAB 23:PRINT"Eingabefehler : sind alle Angaben
     richtig ? <J/N> .... ";:GET E$
2535 IF E$ = "J" OR E$ = "N" THEN VTAB 23:HTAB 65:PRINT"
     ";E$:ELSE 2530
2540 IF E$ = "N" THEN VTAB 3:HTAB 65:INVERSE:PRINT"->
     KORREKTUR <-":NORMAL
2545 IF E$ = "N" THEN VTAB 23:HTAB 79:PRINT " ";:GOTO
     2490
2550 VTAB 3:HTAB 65:PRINT SPC(15)
2555 FOR I = 1 TO 3:VTAB 1:HTAB 60:PRINT SPC(16);:FOR S =
     1 TO 100:NEXT S:VTAB 1:HTAB 60:PRINT"Bitte warten
     ...":FOR S1 = 1 TO 100:NEXT S1:NEXT I
2560 CHAIN "MENUE"
```

F.BAS: Einkünfte aus Kapitalvermögen

```
2565 REM EINKUENFTE AUS KAPITALVERMOEGEN
2570 COMMON A(), B(), C(), D(), E(), M(), N()
2575 HOME:PRINT"Einkünfte aus Kapitalvermögen :"
2580 FOR I = 1 TO 79:PRINT"-";:NEXT I:PRINT
2585 VTAB 7:PRINT"Einnahmen : aus dem Kapitalvermögen ...
     <in DM> ..."
2590 PRINT"Werbungskosten : durch das Kapitalvermögen .
     <in DM> ..."
2595 VTAB 4:HTAB 65:INVERSE:PRINT"--> EHEMANN <--":NORMAL
2600 VTAB 7:HTAB 65:INPUT"",A(35)
2605 VTAB 7:HTAB 65:PRINT" ":VTAB 7:HTAB 65:PRINT USING
     "################";A(35)
2610 HTAB 65:INPUT"",A(36)
2615 VTAB 8:HTAB 65:PRINT" ":VTAB 8:HTAB 65:PRINT USING
     "################";A(36)
2620 VTAB 23:PRINT"Eingabefehler : sind alle Angaben
     richtig ? <J/N> .... ";:GET E$
2625 IF E$ = "J" OR E$ = "N" THEN VTAB 23:HTAB 65:PRINT"
     ";E$:ELSE 2620
2630 IF E$ = "N" THEN VTAB 3:HTAB 65:INVERSE:PRINT"->
     KORREKTUR <-":NORMAL
2635 IF E$ = "N" THEN VTAB 23:HTAB 79:PRINT" ";:GOTO 2600
2640 VTAB 3:HTAB 65:PRINT SPC(15)
2645 IF A(1) = 1 THEN 2710
2650 VTAB 22:PRINT SPC(79):HTAB 79:PRINT" "
2655 VTAB 4:HTAB 65:INVERSE:PRINT"--> EHEFRAU <--":NORMAL
2660 VTAB 7:HTAB 70:PRINT SPC(10):VTAB 8:HTAB 70:PRINT
     SPC(10)
2665 VTAB 7:HTAB 65:INPUT"",B(35)
2670 VTAB 7:HTAB 65:PRINT" ":VTAB 7:HTAB 65:PRINT USING
     "################";B(35)
2675 HTAB 65:INPUT"",B(36)
2680 VTAB 8:HTAB 65:PRINT" ":VTAB 8:HTAB 65:PRINT USING
     "################";B(36)
2685 VTAB 23:PRINT"Eingabefehler : sind alle Angaben
     richtig ? <J/N> .... ";:GET E$
2690 IF E$ = "J" OR E$ = "N" THEN VTAB 23:HTAB 65:PRINT"
     ";E$:ELSE 2685
2695 IF E$ = "N" THEN VTAB 3:HTAB 65:INVERSE:PRINT"->
     KORREKTUR <-":NORMAL
2700 IF E$ = "N" THEN VTAB 23:HTAB 79:PRINT " ";:GOTO
     2665
2705 VTAB 3:HTAB 65:PRINT SPC(15)
2710 FOR I = 1 TO 3:VTAB 1:HTAB 60:PRINT SPC(16);:FOR S =
     1 TO 100:NEXT S:VTAB 1:HTAB 60:PRINT"Bitte warten
     ...":FOR S1 = 1 TO 100:NEXT S1:NEXT I
2715 CHAIN "MENUE"
```

G.BAS: Einkünfte aus Vermietung und Verpachtung

```
2720 REM EINKUENFTE AUS VERMIETUNG UND VERPACHTUNG
2725 COMMON A(), B(), C(), D(), E(), M(), N()
2730 HOME:PRINT"Einkünfte aus Vermietung und Verpachtung
     : ";:HTAB 69:INVERSE:PRINT"SEITE 1":NORMAL
2735 FOR I = 1 TO 79:PRINT"-";:NEXT I:PRINT
2740 VTAB 7:PRINT"Einheitswert :
     selbstgen.Haus/Eigentumsw. <in DM> ..."
2745 PRINT"Wohnzwecke : Wohnnutzung ................ <in %
     > ..."
2750 PRINT"Selbstnutzung : Wohnnutzung in Monaten ....
     <in MM> ..."
2755 PRINT"Schuldzinsen : vor der Selbstnutzung ..... <in
     DM> ..."
2760 PRINT"Schuldzinsen : bei Selbstnutzung ......... <in
     DM> ..."
2765 PRINT"A F A (insgesamt) : Absetzung für Abnutzung
     ... <in DM> ..."
2770 PRINT:PRINT"Wurde der Antrag auf Baugenehmigung nach
     dem 30.09.82 einge-"
2775 PRINT"reicht u. das Gebäude bis 01.01.83
     fertiggestellt . <J/N> ..."
2780 PRINT:PRINT"sonstige Mieteinnahmen: andere
     Häuser/Wohnungen ... <in DM> ..."
2785 PRINT"Werbungskosten : insgesamte Werbungskosten .
     <in DM> ..."
2790 VTAB 4:HTAB 65:INVERSE:PRINT"--> EHEMANN <--":NORMAL
2795 VTAB 7:HTAB 65:INPUT"",A(40)
2800 VTAB 7:HTAB 65:PRINT" ":VTAB 7:HTAB 65:PRINT USING
     "################";A(40)
2805 VTAB 8:HTAB 65:INPUT"",A(41)
2810 IF A(41) > 100 THEN 2805
2815 VTAB 8:HTAB 65:PRINT" ":VTAB 8:HTAB 65:PRINT USING
     "################";A(41)
2820 VTAB 9:HTAB 65:INPUT"",A(42)
2825 IF A(42) > 12 THEN 2820
2830 VTAB 9:HTAB 65:PRINT" ":VTAB 9:HTAB 65:PRINT USING
     "################";A(42)
2835 IF A(42) > 12 THEN 2820
2840 HTAB 65:INPUT"",A(43)
2845 VTAB 10:HTAB 65:PRINT" ":VTAB 10:HTAB 65:PRINT USING
     "################";A(43)
2850 HTAB 65:INPUT"",A(44)
```

```
2855 VTAB 11:HTAB 65:PRINT" ":VTAB 11:HTAB 65:PRINT USING
     "################";A(44)
2860 HTAB 65:INPUT"",A(45)
2865 VTAB 12:HTAB 65:PRINT" ":VTAB 12:HTAB 65:PRINT USING
     "################";A(45)
2870 VTAB 15:HTAB 65:GET Y$
2875 IF Y$ = "J" OR Y$ = "N" THEN VTAB 15:HTAB 65:PRINT"
     ";Y$:ELSE 2870
2880 IF Y$ = "J" THEN A(49) = 1
2885 VTAB 17:HTAB 65:INPUT"",A(46)
2890 VTAB 17:HTAB 65:PRINT" ":VTAB 17:HTAB 65:PRINT USING
     "################";A(46)
2895 HTAB 65:INPUT"",A(47)
2900 VTAB 18:HTAB 65:PRINT" ":VTAB 18:HTAB 65:PRINT USING
     "################";A(47)
2905 VTAB 23:PRINT"Eingabefehler : sind alle Angaben
     richtig ? <J/N> .... ";:GET E$
2910 IF E$ = "J" OR E$ = "N" THEN VTAB 23:HTAB 65:PRINT"
     ";E$:ELSE 2905
2915 IF E$ = "N" THEN VTAB 3:HTAB 65:INVERSE:PRINT"->
     KORREKTUR <-":NORMAL
2920 IF E$ = "N" THEN VTAB 23:HTAB 79:PRINT " ";:GOTO
     2795
2925 VTAB 3:HTAB 65:PRINT SPC(15)
2930 IF A(1) = 1 THEN 3090
2935 VTAB 22:PRINT SPC(79):HTAB 79:PRINT" "
2940 VTAB 4:HTAB 65:INVERSE:PRINT"--> EHEFRAU <--":NORMAL
2945 VTAB 7:HTAB 70:PRINT SPC(10):VTAB 8:HTAB 70:PRINT
     SPC(10):VTAB 9:HTAB 70:PRINT SPC(10)
2950 VTAB 10:HTAB 70:PRINT SPC(10):VTAB 11:HTAB 70:PRINT
     SPC(10):VTAB 12:HTAB 70:PRINT SPC(10)
2955 VTAB 15:HTAB 70:PRINT SPC(10):VTAB 17:HTAB 70:PRINT
     SPC(10):VTAB 18:HTAB 70:PRINT SPC(10)
2960 VTAB 7:HTAB 65:INPUT"",B(40)
2965 VTAB 7:HTAB 65:PRINT" ":VTAB 7:HTAB 65:PRINT USING
     "################";B(40)
2970 VTAB 8:HTAB 65:INPUT"",B(41)
2975 IF B(41) > 100 THEN 2970
2980 VTAB 8:HTAB 65:PRINT" ":VTAB 8:HTAB 65:PRINT USING
     "################";B(41)
2985 VTAB 9:HTAB 65:INPUT"",B(42)
2990 IF B(42) > 12 THEN 2985
2995 VTAB 9:HTAB 65:PRINT" ":VTAB 9:HTAB 65:PRINT USING
     "################";B(42)
3000 HTAB 65:INPUT"",B(43)
3005 VTAB 10:HTAB 65:PRINT" ":VTAB 10:HTAB 65:PRINT USING
     "################";B(43)
3010 HTAB 65:INPUT"",B(44)
3015 VTAB 11:HTAB 65:PRINT" ":VTAB 11:HTAB 65:PRINT USING
     "################";B(44)
3020 HTAB 65:INPUT"",B(45)
3025 VTAB 12:HTAB 65:PRINT" ":VTAB 12:HTAB 65:PRINT USING
     "################";B(45)
3030 VTAB 15:HTAB 65:GET Y$
3035 IF Y$ = "J" OR Y$ = "N" THEN VTAB 15:HTAB 65:PRINT"
     ";Y$:ELSE 3030
3040 IF Y$ = "J" THEN B(49) = 1
3045 VTAB 17:HTAB 65:INPUT"",B(46)
3050 VTAB 17:HTAB 65:PRINT" ":VTAB 17:HTAB 65:PRINT USING
     "################";B(46)
3055 HTAB 65:INPUT"",B(47)
3060 VTAB 18:HTAB 65:PRINT" ":VTAB 18:HTAB 65:PRINT USING
     "################";B(47)
3065 VTAB 23:PRINT"Eingabefehler : sind alle Angaben
     richtig ? <J/N> .... ";:GET E$
3070 IF E$ = "J" OR E$ = "N" THEN VTAB 23:HTAB 65:PRINT"
     ";E$:ELSE 3065
3075 IF E$ = "N" THEN VTAB 3:HTAB 65:INVERSE:PRINT"->
     KORREKTUR <-":NORMAL
3080 IF E$ = "N" THEN VTAB 23:HTAB 79:PRINT " ";:GOTO
     2960
3085 VTAB 3:HTAB 65:PRINT SPC(15)
3090 HOME:PRINT"Einkünfte aus Vermietung und Verpachtung
     : ";:HTAB 69:INVERSE:PRINT"SEITE 2":NORMAL
3095 FOR I = 1 TO 79:PRINT"-";:NEXT I:PRINT
3100 VTAB 7:PRINT"Erhöhte Absetzungen : nach Par. 7b ESTG
     ......... <J/N> ...."
3105 PRINT:PRINT"Erfolgte der Bauantrag oder der Beginn
     der Bauarbeiten oder"
3110 PRINT"der Erwerb des Vertrages nach dem 29.07.1981
     ....... <J/N> ...."
3115 VTAB 4:HTAB 65:INVERSE:PRINT"--> EHEMANN <--"
3120 VTAB 5:HTAB 65:PRINT"--> EHEFRAU <--":NORMAL
3125 VTAB 7:HTAB 65:GET Y$
3130 IF Y$ = "J" OR Y$ = "N" THEN VTAB 7:HTAB 65:PRINT"
     ";Y$:ELSE 3125
3135 IF Y$ = "N" THEN 3155
3140 VTAB 10:HTAB 65:GET Y$
3145 IF Y$ = "J" OR Y$ = "N" THEN VTAB 10:HTAB 65:PRINT"
     ";Y$:ELSE 3140
3150 IF Y$ = "J" THEN A(48) = A(4) - 1:IF A(48) < = 0
     THEN A(48) = 0
3155 VTAB 23:PRINT"Eingabefehler : sind alle Angaben
     richtig ? <J/N> .... ";:GET E$
3160 IF E$ = "J" OR E$ = "N" THEN VTAB 23:HTAB 65:PRINT"
     ";E$:ELSE 3155
3165 IF E$ = "N" THEN VTAB 3:HTAB 65:INVERSE:PRINT"->
     KORREKTUR <-":NORMAL
3170 IF E$ = "N" THEN VTAB 23:HTAB 79:PRINT " ";:GOTO
     3125
3175 FOR I = 1 TO 3:VTAB 1:HTAB 60:PRINT SPC(16);:FOR S =
     1 TO 100:NEXT S:VTAB 1:HTAB 60:PRINT"Bitte warten
     ...":FOR S1 = 1 TO 100:NEXT S1:NEXT I
3180 CHAIN "MENUE"
```

H.BAS: Sonstige Einkünfte

```
3185 REM SONSTIGE EINKUENFTE
3190 COMMON A(), B(), C(), D(), E(), M(), N()
3195 HOME:PRINT"Sonstige Einkünfte :"
3200 FOR I = 1 TO 79:PRINT"-";:NEXT I:PRINT
3205 VTAB 7:PRINT"Einnahmen : nach Par.22 Nr.1 u. 1a <in
     DM> ..."
3210 PRINT"Werbungskosten : auf die Einnahmen ..... <in
     DM> ..."
3215 PRINT"Einnahmen : sonstige Einnahmen .... <in DM>
     ..."
3220 PRINT"Werbungskosten : auf sonstige Einnahmen <in
     DM> ..."
3225 VTAB 4:HTAB 65:INVERSE:PRINT"--> EHEMANN <--":NORMAL
3230 VTAB 7:HTAB 65:INPUT"",A(50)
3235 VTAB 7:HTAB 65:PRINT" ":VTAB 7:HTAB 65:PRINT USING
     "################";A(50)
3240 HTAB 65:INPUT"",A(51)
3245 VTAB 8:HTAB 65:PRINT" ":VTAB 8:HTAB 65:PRINT USING
     "################";A(51)
3250 HTAB 65:INPUT"",A(52)
3255 VTAB 9:HTAB 65:PRINT" ":VTAB 9:HTAB 65:PRINT USING
     "################";A(52)
3260 HTAB 65:INPUT"",A(53)
3265 VTAB 10:HTAB 65:PRINT" ":VTAB 10:HTAB 65:PRINT USING
     "################";A(53)
3270 VTAB 23:PRINT"Eingabefehler : sind alle Angaben
     richtig ? <J/N> .... ";:GET E$
3275 IF E$ = "J" OR E$ = "N" THEN VTAB 23:HTAB 65:PRINT"
     ";E$:ELSE 3270
3280 IF E$ = "N" THEN VTAB 3:HTAB 65:INVERSE:PRINT"->
     KORREKTUR <-":NORMAL
3285 IF E$ = "N" THEN VTAB 23:HTAB 79:PRINT" ";:GOTO 3230
3290 VTAB 3:HTAB 65:PRINT SPC(15)
3295 IF A(1) = 1 THEN 3380
3300 VTAB 22:PRINT SPC(79):HTAB 79:PRINT" "
3305 VTAB 4:HTAB 65:INVERSE:PRINT"--> EHEFRAU <--":NORMAL
3310 VTAB 7:HTAB 70:PRINT SPC(10):VTAB 8:HTAB 70:PRINT
     SPC(10):VTAB 9:HTAB 70:PRINT SPC(10):VTAB 10:HTAB
     70:PRINT SPC(10)
3315 VTAB 7:HTAB 65:INPUT"",B(50)
3320 VTAB 7:HTAB 65:PRINT" ":VTAB 7:HTAB 65:PRINT USING
     "################";B(50)
3325 HTAB 65:INPUT"",B(51)
3330 VTAB 8:HTAB 65:PRINT" ":VTAB 8:HTAB 65:PRINT USING
     "################";B(51)
3335 HTAB 65:INPUT"",B(52)
3340 VTAB 9:HTAB 65:PRINT" ":VTAB 9:HTAB 65:PRINT USING
     "################";B(52)
3345 HTAB 65:INPUT"",B(53)
3350 VTAB 10:HTAB 65:PRINT" ":VTAB 10:HTAB 65:PRINT USING
     "################";B(53)
3355 VTAB 23:PRINT"Eingabefehler : sind alle Angaben
     richtig ? <J/N> .... ";:GET E$
3360 IF E$ = "J" OR E$ = "N" THEN VTAB 23:HTAB 65:PRINT"
     ";E$:ELSE 3355
3365 IF E$ = "N" THEN VTAB 3:HTAB 65:INVERSE:PRINT"->
     KORREKTUR <-":NORMAL
3370 IF E$ = "N" THEN VTAB 23:HTAB 79:PRINT" ";:GOTO 3315
3375 VTAB 3:HTAB 65:PRINT SPC(15)
3380 FOR I = 1 TO 3:VTAB 1:HTAB 60:PRINT SPC(16);:FOR S =
     1 TO 100:NEXT S:VTAB 1:HTAB 60:PRINT"Bitte warten
     ...":FOR S1 = 1 TO 100:NEXT S1:NEXT I
3385 CHAIN "MENUE"
```

I.BAS: Doppelbesteuerungsabkommen und außerordentliche Einkünfte

```
3390 REM DOPPELBESTEUERUNGSABKOMMEN UND AUSSERORDENTLICHE
     EINKUENFTE
3395 COMMON A(), B(), C(), D(), E(), M(), N()
3400 HOME:PRINT"Doppelbesteuerungsabkommen u.
     außerordentliche Einkünfte :"
3405 FOR I = 1 TO 79:PRINT"-";:NEXT I:PRINT
3410 VTAB 7:PRINT"Steuerfrei : nach Doppelbesteurungsabk.
     <in DM> ..."
3415 PRINT"darin Einkünfte : außerordentliche Einkünfte
     <in DM> ..."
3420 PRINT"Entschädigungen : nach Par.24 Nr.1 ESTG .....
     <in DM> ..."
3425 PRINT"Nutzungsvergütungen : nach Par.24 Nr.3 ESTG
     ..... <in DM> ..."
3430 VTAB 4:HTAB 65:INVERSE:PRINT"--> EHEMANN <--":NORMAL
3435 VTAB 7:HTAB 65:INPUT"",A(65)
```

```
3440 VTAB 7:HTAB 65:PRINT" ":VTAB 7:HTAB 65:PRINT USING
     "################";A(65)
3445 HTAB 65:INPUT"",A(66)
3450 VTAB 8:HTAB 65:PRINT" ":VTAB 8:HTAB 65:PRINT USING
     "################";A(66)
3455 HTAB 65:INPUT"",A(60)
3460 VTAB 9:HTAB 65:PRINT" ":VTAB 9:HTAB 65:PRINT USING
     "################";A(60)
3465 HTAB 65:INPUT"",A(61)
3470 VTAB 10:HTAB 65:PRINT" ":VTAB 10:HTAB 65:PRINT USING
     "################";A(61)
3475 VTAB 23:PRINT"Eingabefehler : sind alle Angaben
     richtig ? <J/N> .... ";:GET E$
3480 IF E$ = "J" OR E$ = "N" THEN VTAB 23:HTAB 65:PRINT"
     ";E$:ELSE 3475
3485 IF E$ = "N" THEN VTAB 3:HTAB 65:INVERSE:PRINT"->
     KORREKTUR <-":NORMAL
3490 IF E$ = "N" THEN VTAB 23:HTAB 79:PRINT" ";:GOTO 3435
3495 VTAB 3:HTAB 65:PRINT SPC(15)
3500 IF A(1) = 1 THEN 3585
3505 VTAB 22:PRINT SPC(79):HTAB 79:PRINT" "
3510 VTAB 4:HTAB 65:INVERSE:PRINT"--> EHEFRAU <--":NORMAL
3515 VTAB 7:HTAB 70:PRINT SPC(10):VTAB 8:HTAB 70:PRINT
     SPC(10):VTAB 9:HTAB 70:PRINT SPC(10):VTAB 10:HTAB
     70:PRINT SPC(10)
3520 VTAB 7:HTAB 65:INPUT"",B(65)
3525 VTAB 7:HTAB 65:PRINT" ":VTAB 7:HTAB 65:PRINT USING
     "################";B(65)
3530 HTAB 65:INPUT"",B(66)
3535 VTAB 8:HTAB 65:PRINT" ":VTAB 8:HTAB 65:PRINT USING
     "################";B(66)
3540 HTAB 65:INPUT"",B(60)
3545 VTAB 9:HTAB 65:PRINT" ":VTAB 9:HTAB 65:PRINT USING
     "################";B(60)
3550 HTAB 65:INPUT"",B(61)
3555 VTAB 10:HTAB 65:PRINT" ":VTAB 10:HTAB 65:PRINT USING
     "################";B(61)
3560 VTAB 23:PRINT"Eingabefehler : sind alle Angaben
     richtig ? <J/N> .... ";:GET E$
3565 IF E$ = "J" OR E$ = "N" THEN VTAB 23:HTAB 65:PRINT"
     ";E$:ELSE 3560
3570 IF E$ = "N" THEN VTAB 3:HTAB 65:INVERSE:PRINT"->
     KORREKTUR <-":NORMAL
3575 IF E$ = "N" THEN VTAB 23:HTAB 79:PRINT" ";:GOTO 3520
3580 VTAB 3:HTAB 65:PRINT SPC(15)
3585 FOR I = 1 TO 3:VTAB 1:HTAB 60:PRINT SPC(16);:FOR S =
     1 TO 100:NEXT S:VTAB 1:HTAB 60:PRINT"Bitte warten
     ...":FOR S1 = 1 TO 100:NEXT S1:NEXT I
3590 CHAIN "MENUE"
```

J.BAS: Sonderausgaben

```
3595 REM SONDERAUSGABEN
3600 COMMON A(), B(), C(), D(), E(), M(), N()
3605 HOME:PRINT"Sonderausgaben : ";:HTAB
     69:INVERSE:PRINT"SEITE 1":NORMAL
3610 FOR I = 1 TO 79:PRINT"-";:NEXT I:PRINT
3615 VTAB 6:HTAB 65:INVERSE:PRINT"--> EHEMANN <--":NORMAL
3620 VTAB 7:PRINT"Arbeitnehmer-Anteil : zur ges.
     Rentenversicherung <in DM> ..."
3625 PRINT"Arbeitgeber -Anteil : zur ges.
     Rentenversicherung <in DM> ..."
3630 PRINT:HTAB 65:INVERSE:PRINT"--> EHEFRAU <--":NORMAL
3635 PRINT"Arbeitnehmer-Anteil : zur ges.
     Rentenversicherung <in DM> ..."
3640 PRINT"Arbeitgeber -Anteil : zur ges.
     Rentenversicherung <in DM> ..."
3645 PRINT:HTAB 65:INVERSE:PRINT"--> EHEMANN <--"
3650 HTAB 65:PRINT"--> EHEFRAU <--":NORMAL
3655 PRINT"Sonderausgaben : unbeschränkt, ohne Spenden
     <in DM> ..."
3660 PRINT"Vorsorgeaufwendungen : abzugsfähige
     Versicherungen <in DM> ..."
3665 PRINT"Bausparbeiträge : lt. Bausparvertrag <in DM>
     ..."
3670 IF (A(30) + A(31)) = 0 THEN 3695
3675 VTAB 7:HTAB 65:INPUT"",A(72)
3680 VTAB 7:HTAB 65:PRINT" ":VTAB 7:HTAB 65:PRINT USING
     "################";A(72)
3685 HTAB 65:INPUT"",A(73)
3690 VTAB 8:HTAB 65:PRINT" ":VTAB 8:HTAB 65:PRINT USING
     "################";A(73)
3695 IF A(1) = 1 THEN 3725
3700 IF (B(30) + B(31)) = 0 THEN 3725
3705 VTAB 11:HTAB 65:INPUT"",B(72)
3710 VTAB 11:HTAB 65:PRINT" ":VTAB 11:HTAB 65:PRINT USING
     "################";B(72)
3715 HTAB 65:INPUT"",B(73)
3720 VTAB 12:HTAB 65:PRINT" ":VTAB 12:HTAB 65:PRINT USING
     "################";B(73)
3725 VTAB 16:HTAB 65:INPUT"",A(70)
3730 VTAB 16:HTAB 65:PRINT" ":VTAB 16:HTAB 65:PRINT USING
     "################";A(70)
3735 HTAB 65:INPUT"",A(71)
3740 VTAB 17:HTAB 65:PRINT" ":VTAB 17:HTAB 65:PRINT USING
     "################";A(71)
3745 HTAB 65:INPUT"",A(74)
3750 VTAB 18:HTAB 65:PRINT" ":VTAB 18:HTAB 65:PRINT USING
     "################";A(74)
3755 VTAB 23:PRINT"Eingabefehler : sind alle Angaben
     richtig ? <J/N> .... ";:GET E$
3760 IF E$ = "J" OR E$ = "N" THEN VTAB 23:HTAB 65:PRINT"
     ";E$:ELSE 3755
3765 IF E$ = "N" THEN VTAB 5:HTAB 65:INVERSE:PRINT"->
     KORREKTUR <-":NORMAL
3770 IF E$ = "N" THEN VTAB 23:HTAB 79:PRINT" ";: GOTO
     3670
3775 HOME:PRINT"Sonderausgaben : ";:HTAB
     69:INVERSE:PRINT"SEITE 2":NORMAL
3780 FOR I = 1 TO 79:PRINT"-";:NEXT I:PRINT
3785 VTAB 4:HTAB 65:INVERSE:PRINT"--> EHEMANN <--"
3790 HTAB 65:PRINT"--> EHEFRAU <--":NORMAL
3795 VTAB 6:PRINT"Spenden : an politische Parteien ....
     <in DM> ..."
3800 PRINT"sonstige Spenden : an andere Institutionen ...
     <in DM> ..."
3805 PRINT:PRINT"Sind in den sonstigen Spenden auch
     Spenden für wissenschaft-"
3810 PRINT"liche, staatspolitische u. als besonders
     förderungswürdig"
3815 PRINT"anerkannte Zwecke ? Bitte die Höhe angeben
     ..... <in DM> ..."
3820 PRINT:HTAB 30:INVERSE:PRINT"WAHLRECHT":NORMAL
3825 PRINT"Sie können die Begrenzung des Spendenabzugs
     wählen :"
3830 PRINT" <A> 5 % des Gesamtbetrages der Einkünfte"
3835 PRINT" <B> 2 v.T. der Summe der gesamten Löhne u.
     Gehälter"
3840 PRINT"--> <B> gilt nur für Arbeitgeber"
3845 PRINT:PRINT"Bitte wählen Sie <A> oder <B>
     ..... <A/B> ..."
3850 PRINT:PRINT"Es folgt nun die Gesamtsumme der Löhne
     u. Gehälter <in DM> ..."
3855 VTAB 6:HTAB 65:INPUT"",A(76)
3860 VTAB 6:HTAB 65:PRINT" ":VTAB 6:HTAB 65:PRINT USING
     "################";A(76)
3865 HTAB 65:INPUT"",A(75)
3870 VTAB 7:HTAB 65:PRINT" ":VTAB 7:HTAB 65:PRINT USING
     "################";A(75)
3875 IF A(75) = 0 THEN 3915
3880 VTAB 11:HTAB 65:INPUT"",A(78)
3885 VTAB 11:HTAB 65:PRINT" ":VTAB 11:HTAB 65:PRINT USING
     "################";A(78)
3890 VTAB 19:HTAB 65:GET Y$
3895 IF Y$ = "A" OR Y$ = "B" THEN VTAB 19:HTAB 65:PRINT"
     ";Y$:ELSE 3890
3900 IF Y$ = "A" THEN 3915
3905 VTAB 21:HTAB 65:INPUT"",A(77)
3910 VTAB 21:HTAB 65:PRINT" ":VTAB 21:HTAB 65:PRINT USING
     "################";A(77)
3915 VTAB 23:PRINT"Eingabefehler : sind alle Angaben
     richtig ? <J/N> .... ";:GET E$
3920 IF E$ = "J" OR E$ = "N" THEN VTAB 23:HTAB 65:PRINT"
     ";E$:ELSE 3915
3925 IF E$ = "N" THEN VTAB 3:HTAB 65:INVERSE:PRINT"->
     KORREKTUR <-":NORMAL
3930 IF E$ = "N" THEN VTAB 23:HTAB 79:PRINT" ";:GOTO 3855
3935 HOME:PRINT"Sonderausgaben : ";:HTAB
     69:INVERSE:PRINT"SEITE 3":NORMAL
3940 FOR I = 1 TO 79:PRINT"-";:NEXT I:PRINT
3945 VTAB 5:HTAB 65:INVERSE:PRINT"--> EHEMANN <--"
3950 HTAB 65:PRINT"--> EHEFRAU <--":NORMAL
3955 VTAB 8:PRINT"Verlustabzug : Verlust der Einkünfte
     ..... <J/N> ...."
3960 PRINT"Verlust '80 : für das Jahr 1980 ..... <in
     DM> ..."
3965 PRINT"Verlust '81 : für das Jahr 1981 ..... <in
     DM> ..."
3970 PRINT"Verlust '82 : für das Jahr 1982 ..... <in
     DM> ..."
3975 PRINT"Verlust '83 : für das Jahr 1983 ..... <in
     DM> ..."
3980 PRINT"Verlust '85 : für das Jahr 1985 ..... <in
     DM> ..."
3985 PRINT"Verlust '86 : für das Jahr 1986 ..... <in
     DM> ..."
3990 VTAB 8:HTAB 65:GET Y$
3995 IF Y$ = "J" OR Y$ = "N" THEN VTAB 8:HTAB 65:PRINT"
     ";Y$:ELSE 3990
4000 IF Y$ = "N" THEN 4065
4005 HTAB 65:INPUT"",A(80)
4010 VTAB 9:HTAB 65:PRINT" ":VTAB 9:HTAB 65:PRINT USING
     "################";A(80)
4015 HTAB 65:INPUT"",A(81)
4020 VTAB 10:HTAB 65:PRINT" ":VTAB 10:HTAB 65:PRINT USING
     "################";A(81)
4025 HTAB 65:INPUT"",A(82)
4030 VTAB 11:HTAB 65:PRINT" ":VTAB 11:HTAB 65:PRINT USING
     "################";A(82)
4035 HTAB 65:INPUT"",B(79)
```

```
4040 VTAB 12:HTAB 65:PRINT" ":VTAB 12:HTAB 65:PRINT USING
     "################";B(79)
4045 HTAB 65:INPUT"",B(80)
4050 VTAB 13:HTAB 65:PRINT" ":VTAB 13:HTAB 65:PRINT USING
     "################";B(80)
4055 HTAB 65:INPUT"",B(81)
4060 VTAB 14:HTAB 65:PRINT" ":VTAB 14:HTAB 65:PRINT USING
     "################";B(81)
4065 VTAB 23:PRINT"Eingabefehler : sind alle Angaben
     richtig ? <J/N> .... ";:GET E$
4070 IF E$ = "J" OR E$ = "N" THEN VTAB 23:HTAB 65:PRINT"
     ";E$:ELSE 4065
4075 IF E$ = "N" THEN VTAB 4:HTAB 65:INVERSE:PRINT"->
     KORREKTUR <-":NORMAL
4080 IF E$ = "N" THEN VTAB 23:HTAB 79:PRINT" ";: GOTO
     3990
4085 VTAB 3:HTAB 65:PRINT SPC(15)
4090 FOR I = 1 TO 3:VTAB 1:HTAB 60:PRINT SPC(16);:FOR S =
     1 TO 100:NEXT S:VTAB 1:HTAB 60:PRINT"Bitte warten
     ...":FOR S1 = 1 TO 100:NEXT S1:NEXT I
4095 CHAIN "MENUE"
```

K.BAS: Außergewöhnliche Belastungen

```
4100 REM AUSSERGEWOEHNLICHE BELASTUNGEN
4105 COMMON A(), B(), C(), D(), E(), M(), N()
4110 HOME:PRINT"Außergewöhnliche Belastungen :";:HTAB
     69:INVERSE:PRINT"SEITE 2":NORMAL
4115 FOR I = 1 TO 79:PRINT"-";:NEXT I:PRINT
4120 VTAB 7:PRINT"Aufwendungen : nach Par.33
     ..... <in DM> ..."
4125 PRINT"Aufwendung / DDR : Pakete, Päckchen, Besuch ..
     <in DM> ..."
4130 PRINT"Aufwendung / Haushalt : Gehilfin, Heim, Pflege
     .... <in DM> ..."
4135 PRINT"Anzahl der Kinder : mit Unterhaltszahlungen .
     <0,1,2,...> ."
4140 VTAB 4:HTAB 65:INVERSE:PRINT"--> EHEMANN <--"
4145 HTAB 65:PRINT"--> EHEMANN <--":NORMAL
4150 VTAB 7:HTAB 65:INPUT"",A(85)
4155 VTAB 7:HTAB 65:PRINT" ":VTAB 7:HTAB 65:PRINT USING
     "################";A(85)
4160 HTAB 65:INPUT"",A(86)
4165 VTAB 8:HTAB 65:PRINT" ":VTAB 8:HTAB 65:PRINT USING
     "################";A(86)
4170 HTAB 65:INPUT"",A(88)
4175 VTAB 9:HTAB 65:PRINT" ":VTAB 9:HTAB 65:PRINT USING
     "################";A(88)
4180 HTAB 65:INPUT"",A(89)
4185 VTAB 10:HTAB 65:PRINT" ":VTAB 10:HTAB 65:PRINT USING
     "################";A(89)
4190 VTAB 23:PRINT"Eingabefehler : sind alle Angaben
     richtig ? <J/N> .... ";:GET E$
4195 IF E$ = "J" OR E$ = "N" THEN VTAB 23:HTAB 65:PRINT"
     ";E$:ELSE 4190
4200 IF E$ = "N" THEN VTAB 3:HTAB 65:INVERSE:PRINT"->
     KORREKTUR <-":NORMAL
4205 IF E$ = "N" THEN VTAB 23:HTAB 79:PRINT" ";: GOTO
     4150
4210 HOME:PRINT"Außergewöhnliche Belastungen : ";:HTAB
     69:INVERSE:PRINT"SEITE 2":NORMAL
4215 FOR I = 1 TO 79:PRINT"-";:NEXT I:PRINT
4220 U = 0
4225 VTAB 7:PRINT"Sind Aufwendungen für Unterhalt oder
     Berufsausbildung für"
4230 PRINT"Personen entstanden, für die kein Anspruch
     auf"
4235 PRINT"Kindergeld bestand ? :
     ..... <J/N> ...."
4240 PRINT:PRINT"Personen - Nr. : automatische Anzeige
     ..... <Nr.> ...."
4245 PRINT"Aufwendung : Höhe des Unterhalts ..... <in
     DM> ..."
4250 PRINT"Unterhaltsaufwendung : dritter Personen
     ..... <in DM> ..."
4255 PRINT"Einkünfte u. Löhne : für die unterstützte
     Person <in DM> ..."
4260 PRINT"Zeitraum : Unterstützungszeit (Mon.) . <in MM>
     ..."
4265 VTAB 4:HTAB 65:INVERSE:PRINT"--> EHEMANN <--"
4270 HTAB 65:PRINT"--> EHEFRAU <--":NORMAL
4275 VTAB 9:HTAB 65:GET Y$
4280 IF Y$ = "J" OR Y$ = "N" THEN VTAB 9:HTAB 65:PRINT"
     ";Y$:ELSE 4275
4285 U = 0
4290 IF Y$ = "N" THEN 4475
4295 U = U + 1
4300 VTAB 11:HTAB 78:PRINT U
4305 VTAB 12:HTAB 65:INPUT"",U1
4310 VTAB 12:HTAB 65:PRINT" ":VTAB 12:HTAB 65:PRINT USING
     "################";U1
4315 HTAB 65:INPUT"",U2
4320 VTAB 13:HTAB 65:PRINT" ":VTAB 13:HTAB 65:PRINT USING
     "################";U2
4325 HTAB 65:INPUT"",U3
4330 VTAB 14:HTAB 65:PRINT" ":VTAB 14:HTAB 65:PRINT USING
     "################";U3
4335 VTAB 15:HTAB 65:INPUT"",U4
4340 IF U4 > 12 THEN 4335
4345 VTAB 15:HTAB 65:PRINT" ":VTAB 15:HTAB 65:PRINT USING
     "################";U4
4350 IF (U1 + U2 = 0) THEN 4380
4355 UU = 3600: IF U1 < 3600 THEN UU = U1
4360 U5 = UU * U1 / (U1 + U2)
4365 U6 = U3 - 4200:IF U6 < 0 THEN U6 = 0
4370 U7 = INT ((U5 - U6) * U4/12):IF U7 < 0 THEN U7 = 0
4375 A(90) = A(90) + U7
4380 VTAB 17:PRINT"Abzugsfähig für : Berechnung für
     Person ";U;" .....";:HTAB 65:PRINT USING
     "################";U7
4385 VTAB 18:PRINT"Abzugsfähig insgesamt : Berechnung für
     alle Personen .....";:HTAB 65:PRINT USING
     "################";A(90)
4390 U1 = 0 : U2 = 0 : U3 = 0 : U4 = 0 : U5 = 0 : U6 = 0
     : U7 = 0 : UU = 0
4395 VTAB 23:PRINT"Eingabefehler : sind alle Angaben
     richtig ? <J/N> .... ";:GET E$
4400 IF E$ = "J" OR E$ = "N" THEN VTAB 23:HTAB 65:PRINT"
     ";E$:ELSE 4395
4405 IF E$ = "N" THEN A(90) = 0
4410 IF E$ = "N" THEN VTAB 3:HTAB 65:INVERSE:PRINT"->
     KORREKTUR <-":NORMAL
4415 IF E$ = "N" THEN VTAB 23:HTAB 79:PRINT" ";:GOTO 4275
4420 VTAB 3:HTAB 65:PRINT SPC(15)
4425 VTAB 23:HTAB 79:PRINT" "
4430 VTAB 23:PRINT"Unterstützung : weiterer Personen ?
     ..... <J/N> .... ";:GET E$
4435 IF E$ = "N" THEN 4475
4440 VTAB 22:PRINT SPC(79)
4445 VTAB 12:HTAB 70:PRINT SPC(10)
4450 VTAB 13: HTAB 70:PRINT SPC(10)
4455 VTAB 14:HTAB 70:PRINT SPC(10)
4460 VTAB 15:HTAB 70:PRINT SPC(10)
4465 VTAB 17:HTAB 70:PRINT SPC(10)
4470 GOTO 4295
4475 HOME:PRINT"Außergewöhnliche Belastungen : ";:HTAB
     69:INVERSE:PRINT"SEITE 3":NORMAL
4480 FOR I = 1 TO 79:PRINT"-";:NEXT I:PRINT
4485 VTAB 7:PRINT"Ausbildungsfreibetrag : wurde einer
     beantragt ..... <J/N> ..."
4490 PRINT"Personen - Nr. : automatische Anzeige .....
     <Nr.> ..."
4495 PRINT"Anspruch Kindergeld : für diese Person
     ..... <J/N> ..."
4500 PRINT"Alter des Kindes : ist das Kind 18 Jahre alt ?
     <J/N> ..."
4505 PRINT"Wohnung d. Kindes : <A> Eltern, <B> auswärtig
     ... <A/B> ..."
4510 PRINT"Einkünfte d. Kindes : Bezüge, Lohn u. Gehalt
     ..... <in DM> .."
4515 PRINT"BAFOEG : BAFOEG Beträge ..... <in DM>
     .."
4520 PRINT"Kindergeldanspruch : im 18. Lebensjahr
     (Monate) . <in MM> .."
4525 VTAB 4:HTAB 65:INVERSE:PRINT"--> EHEMANN <--"
4530 HTAB 65:PRINT"--> EHEFRAU <--":NORMAL
4535 VTAB 7:HTAB 65:GET Y$
4540 IF Y$ = "J" OR Y$ = "N" THEN VTAB 7:HTAB 65:PRINT"
     ";Y$:ELSE 4520
4545 U = 0
4550 IF Y$ = "N" THEN 4780
4555 U = U + 1
4560 VTAB 8:HTAB 78:PRINT U
4565 VTAB 9:HTAB 65:GET U$
4570 IF U$ = "J" OR U$ = "N" THEN VTAB 9:HTAB 65:PRINT"
     ";U$:ELSE 4565
4575 IF U$ = "N" THEN 4685
4580 VTAB 10:HTAB 65:GET Y$
4585 IF Y$ = "J" OR Y$ = "N" THEN VTAB 10:HTAB 65:PRINT"
     ";Y$:ELSE 4580
4590 VTAB 11:HTAB 65:GET U$
4595 IF U$ = "A" OR U$ = "B" THEN VTAB 11:HTAB 65:PRINT"
     ";U$:ELSE 4590
4600 VTAB 12:HTAB 65:INPUT"",U1
4605 VTAB 12:HTAB 65:PRINT" ":VTAB 12:HTAB 65:PRINT USING
     "################";U1
4610 HTAB 65:INPUT"",U2
4615 VTAB 13:HTAB 65:PRINT" ":VTAB 13:HTAB 65:PRINT USING
     "################";U2
4620 VTAB 14:HTAB 65:INPUT"",U3
4625 IF U3 > 12 THEN 4620
4630 VTAB 14:HTAB 65:PRINT" ":VTAB 14:HTAB 65:PRINT USING
     "################";U3
4635 IF Y$ = "N" AND U$ = "B" THEN UU = 900
4640 IF Y$ = "J" AND U$ = "A" THEN UU = 1200
4645 IF Y$ = "J" AND U$ = "B" THEN UU = 2100
4650 U4 = U1 - 1200: IF U4 < 0 THEN U4 = 0
4655 U5 = UU - U4 - U2: IF U5 < 0 THEN U5 = 0
4660 U5 = INT (U5 * U3 / 12)
4665 A(91) = A(91) + U5
```

```
4670 VTAB 17:PRINT"Abzugsfähig für : Berechnung für
     Person ";U;" .............";:HTAB 65:PRINT USING
     "################";U5
4675 VTAB 18:PRINT"Abzugsfähig insgesamt : Berechnung für
     alle Personen .........";:HTAB 65:PRINT USING
     "################";A(91)
4680 UU = 0 : U1 = 0 : U2 = 0 : U3 = 0 : U4 = 0 : U5 = 0
4685 VTAB 23:PRINT"Eingabefehler : sind alle Angaben
     richtig ? <J/N> .... ";:GET E$
4690 IF E$ = "J" OR E$ = "N" THEN VTAB 23:HTAB 65:PRINT"
     ";E$:ELSE 4685
4695 IF E$ = "N" THEN A(91) = 0
4700 IF E$ = "N" THEN VTAB 3:HTAB 65:INVERSE:PRINT"->
     KORREKTUR <-":NORMAL
4705 IF E$ = "N" THEN VTAB 23:HTAB 79:PRINT" ";:GOTO 4535
4710 VTAB 3:HTAB 65:PRINT SPC(15)
4715 VTAB 23:HTAB 79:PRINT" "
4720 VTAB 23:PRINT"Ausbildungsfreibetrag : weitere
     Freibeträge ? ..... <J/N> .... ";:GET E$
4725 IF E$ = "N" THEN 4780
4730 VTAB 22:PRINT SPC(79)
4735 VTAB 8:HTAB 70:PRINT SPC(10)
4740 VTAB 9:HTAB 70:PRINT SPC(10)
4745 VTAB 10:HTAB 70:PRINT SPC(10)
4750 VTAB 11:HTAB 70:PRINT SPC(10)
4755 VTAB 12:HTAB 70:PRINT SPC(10)
4760 VTAB 13:HTAB 70:PRINT SPC(10)
4765 VTAB 14:HTAB 70:PRINT SPC(10)
4770 VTAB 17:HTAB 70:PRINT SPC(10)
4775 GOTO 4555
4780 HOME:PRINT"Außergewöhnliche Belastungen : ";:HTAB
     69:INVERSE:PRINT"SEITE 4":NORMAL
4785 FOR I = 1 TO 79:PRINT"-";:NEXT I:PRINT
4790 VTAB 7:PRINT"Körperbehinderte : Pauschbetrag n.
     Par.33b .... <J/N> ...."
4795 VTAB 10:PRINT"Erwerbsunfähigkeit : Minderung
     ................ <in % > ..."
4800 PRINT"Erwerbsunfähigkeit : durch
     Blindheit/Hilfslosigk. <J/N> ..."
4805 PRINT"Hinterbliebenenbezug : laufender Bezug
     ........... <J/N> ...."
4810 VTAB 15:PRINT"Erwerbsunfähigkeit : Minderung
     ................ <in % > ..."
4815 PRINT"Erwerbsunfähigkeit : durch
     Blindheit/Hilfslosigk. <J/N> ..."
4820 PRINT"Hinterbliebenenbezug : laufender Bezug
     ........... <J/N> ...."
4825 VTAB 4:HTAB 65:INVERSE:PRINT"--> EHEMANN <--"
4830 VTAB 5:HTAB 65:PRINT"--> EHEFRAU <--"
4835 VTAB 9:HTAB 65:PRINT"--> EHEMANN <--"
4840 VTAB 14:HTAB 65:PRINT"--> EHEFRAU <--":NORMAL
4845 VTAB 7:HTAB 65:GET Y$
4850 IF Y$ = "J" OR Y$ = "N" THEN VTAB 7:HTAB 65:PRINT"
     ";Y$:ELSE 4845
4855 IF Y$ ="N" THEN 4945
4860 VTAB 10:HTAB 65:INPUT"",A(93)
4865 VTAB 10:HTAB 65:PRINT" ":VTAB 10:HTAB 65:PRINT USING
     "################";A(93)
4870 VTAB 11:HTAB 65:GET E$
4875 IF E$ = "J" OR E$ = "N" THEN VTAB 11:HTAB 65:PRINT"
     ";E$:ELSE 4870
4880 IF E$ = "J" THEN A(93) = 300
4885 VTAB 12:HTAB 65:GET Y$
4890 IF Y$ = "J" OR Y$ = "N" THEN VTAB 12:HTAB 65:PRINT"
     ";Y$:ELSE 4885
4895 IF Y$ = "J" THEN A(94) = 1
4900 IF A(1) = 1 THEN 4945
4905 VTAB 15:HTAB 65:INPUT"",B(93)
4910 VTAB 15:HTAB 65:PRINT" ":VTAB 15:HTAB 65 PRINT USING
     "################";B(93)
4915 VTAB 16:HTAB 65:GET E$
4920 IF E$ = "J" OR E$ = "N" THEN VTAB 16:HTAB 65:PRINT"
     ";E$:ELSE 4915
4925 IF E$ = "J" THEN B(93) = 300
4930 VTAB 17:HTAB 65:GET Y$
4935 IF Y$ = "J" OR Y$ = "N" THEN VTAB 17:HTAB 65:PRINT"
     ";Y$:ELSE 4930
4940 IF Y$ = "J" THEN A(94) = 1
4945 VTAB 23:PRINT"Eingabefehler : sind alle Angaben
     richtig ? <J/N> .... ";:GET E$
4950 IF E$ = "J" OR E$ = "N" THEN VTAB 23:HTAB 65:PRINT"
     ";E$:ELSE 4945
4955 IF E$ = "N" THEN VTAB 3:HTAB 65:INVERSE:PRINT"->
     KORREKTUR <-":NORMAL
4960 IF E$ = "N" THEN VTAB 23:HTAB 79:PRINT" ";: GOTO
     4845
4965 VTAB 3:HTAB 65:PRINT SPC(15)
4970 FOR I = 1 TO 3:VTAB 1:HTAB 60:PRINT SPC(16);:FOR S =
     1 TO 100:NEXT S:VTAB 1:HTAB 60:PRINT"Bitte warten
     ...":FOR S1 = 1 TO 100:NEXT S1:NEXT I
4975 CHAIN "MENUE"
```

L.BAS: Besondere Abzugsbeträge und ausländische Steuer

```
4980 REM BESONDERE ABZUGSBETRAEGE UND AUSLAENDISCHE
     STEUER
4985 COMMON A(), B(), C(), D(), E(), M(), N()
4990 HOME:PRINT"Besondere Abzugsbeträge u. ausländische
     Steuer"
4995 FOR I = 1 TO 79:PRINT"-";:NEXT I:PRINT
5000 VTAB 7:PRINT"Ermäßigungsbetrag : nach Par.16,17
     Berlin FG .. <in DM> ..."
5005 PRINT"Ermäßigungsbetrag : als
     Arbeitgeber/Verm.-Leist.<in DM> ..."
5010 PRINT"Ausl. Einkünfte : nach Par.34c, Abs.I,II,V ..
     <in DM> ..."
5015 PRINT"Ausl. Steuern : nach Par.34c. Abs.I,II,V ..
     <in DM> ..."
5020 PRINT"Ausl. Steuern : nach Par.34c. Abs.III .....
     <in DM> ..."
5025 PRINT"Begünstigste Einkünfte: für Erfinder
     ............... <in DM> ..."
5030 PRINT"Ausbildungsplatzeink. : nach Par. 24b ESTG
     ......... <J/N> ...."
5035 PRINT:PRINT"Einkünfte aus der :
     Ausbildungsplatzschaffung . <in DM> ..."
5040 PRINT:PRINT"Einkünfte aus der :
     Ausbildungsplatzschaffung . <in DM> ..."
5045 HTAB 30:INVERSE:PRINT"WAHLRECHT":NORMAL
5050 PRINT" <A> Abzug von der EST nach Par. 34c I"
5055 PRINT" <B> Abzug vom Gesamtbetrag d. Einkünfte nach
     Par.34c II"
5060 PRINT"Bitte hier eingeben : Ihre Wahl <A> oder <B>
     ..... <A/B> ...."
5065 VTAB 4:HTAB 65:INVERSE:PRINT"--> EHEMANN <--"
5070 HTAB 65:PRINT"--> EHEFRAU <--"
5075 VTAB 14:HTAB 65:PRINT"--> EHEMANN <--"
5080 VTAB 16:HTAB 65:PRINT"--> EHEFRAU <--":NORMAL
5085 VTAB 7:HTAB 65:INPUT"",A(95)
5090 VTAB 7:HTAB 65:PRINT" ":VTAB 7:HTAB 65:PRINT USING
     "################";A(95)
5095 HTAB 65:INPUT"",A(96)
5100 VTAB 8:HTAB 65:PRINT" ":VTAB 8:HTAB 65:PRINT USING
     "################";A(96)
5105 HTAB 65:INPUT"",A(97)
5110 VTAB 9:HTAB 65:PRINT" ":VTAB 9:HTAB 65:PRINT USING
     "################";A(97)
5115 HTAB 65:INPUT"",A(98)
5120 VTAB 10:HTAB 65:PRINT" ":VTAB 10:HTAB 65:PRINT USING
     "################";A(98)
5125 HTAB 65:INPUT"",A(100)
5130 VTAB 11:HTAB 65:PRINT" ":VTAB 11:HTAB 65:PRINT USING
     "################";A(100)
5135 HTAB 65:INPUT"",A(102)
5140 VTAB 12:HTAB 65:PRINT" ":VTAB 12:HTAB 65:PRINT USING
     "################";A(102)
5145 VTAB 13:HTAB 65:GET Y$
5150 IF Y$ = "J" OR Y$ = "N" THEN VTAB 13:HTAB 65:PRINT"
     ";Y$:ELSE 5145
5155 IF Y$ = "N" THEN 5185
5160 VTAB 15:HTAB 65:INPUT"",A(101)
5165 VTAB 15:HTAB 65:PRINT" ":VTAB 15:HTAB 65:PRINT USING
     "################";A(101)
5170 IF A(1) = 1 THEN 5185
5175 VTAB 17:HTAB 65:INPUT"",B(101)
5180 VTAB 17:HTAB 65:PRINT" ":VTAB 17:HTAB 65:PRINT USING
     "################";B(101)
5185 IF A(97) = 0 AND A(98) = 0 THEN 5205
5190 VTAB 21:HTAB 65:GET Y$
5195 IF Y$ = "A" OR Y$ = "B" THEN VTAB 21:HTAB 65:PRINT"
     ";Y$:ELSE 5185
5200 A(99) = 1: IF Y$ = "B" THEN A(99) = 2
5205 VTAB 23:PRINT"Eingabefehler : sind alle Angaben
     richtig ? <J/N> .... ";:GET E$
5210 IF E$ = "N" THEN VTAB 3:HTAB 65:INVERSE:PRINT"->
     KORREKTUR <-":NORMAL
5215 IF E$ = "N" THEN VTAB 23:HTAB 79:PRINT" ";:GOTO 5085
5220 IF E$ = "J" OR E$ = "N" THEN VTAB 23:HTAB 65:PRINT"
     ";E$:ELSE 5205
5225 VTAB 3:HTAB 65:PRINT SPC(15)
5230 FOR I = 1 TO 3:VTAB 1:HTAB 60:PRINT SPC(16);:FOR S =
     1 TO 100:NEXT S:VTAB 1:HTAB 60:PRINT"Bitte warten
     ...":FOR S1 = 1 TO 100:NEXT S1:NEXT I
5235 CHAIN "MENUE"
```

M.BAS: Angaben für die An- und Abrechnung

```
5240 REM ANGABEN FUER DIE AN- U. ABRECHNUNG
5245 COMMON A(), B(), C(), D(), E(), M(), N()
5250 HOME:PRINT"Einkünfte für An- u. Abrechnung :":NORMAL
5255 FOR I = 1 TO 79:PRINT"-";:NEXT I:PRINT
5260 VTAB 7:PRINT"Angaben : An- u. Abrechnung .........
     <J/N> ...."
5265 PRINT"Lohnsteuer : Gesamt-Lohnsteuer ......... <in
     DM> ..."
```

```
5270 PRINT"Kirchensteuer : evangelische Kirche .....
     <in DM> ..."
5275 PRINT"Kirchensteuer : katholische Kirche ..... <in
     DM> ..."
5280 PRINT"Kapitalertragssteuer : insgesamt
     ..... <in DM> ..."
5285 PRINT"Körperschaftssteuer : anrechenbare Steuer
     ..... <in DM> ..."
5290 PRINT"Einkommenssteuer : vorausgezahlte Beträge ....
     <in DM> ..."
5295 PRINT"Kirchensteuer : vorausgezahlte Beträge ev. <in
     DM> ..."
5300 PRINT"Kirchensteuer : vorausgezahlte Beträge rk. <in
     DM> ..."
5305 VTAB 4:HTAB 65:INVERSE:PRINT"--> EHEMANN <--"
5310 HTAB 65:PRINT"--> EHEFRAU <-
-":NORMAL
5315 VTAB 7:HTAB 65:GET Y$
5320 IF Y$ = "J" OR Y$ = "N" THEN VTAB 7:HTAB 65:PRINT"
     ";Y$:ELSE 5315
5325 IF Y$ = "N" THEN 5440
5330 IF A(30) = 0 AND A(31) = 0 AND B(30) = 0 AND B(31) =
     0 THEN 5375
5335 VTAB 8:HTAB 65:INPUT"",A(110)
5340 VTAB 8:HTAB 65:PRINT" ":VTAB 8:HTAB 65:PRINT USING
     "################";A(110)
5345 IF A(3) < > 1 AND B(3) < > 1 THEN 5360
5350 VTAB 9:HTAB 65:INPUT"",A(111)
5355 VTAB 9:HTAB 65:PRINT" ":VTAB 9:HTAB 65:PRINT USING
     "################";A(111)
5360 IF A(3) < > 2 AND B(3) < > 2 THEN 5375
5365 VTAB 10:HTAB 65:INPUT"",A(112)
5370 VTAB 10:HTAB 65:PRINT" ":VTAB 10:HTAB 65:PRINT USING
     "################";A(112)
5375 IF A(35) + B(35) = 0 AND A(36) + B(36) = 0 THEN 5400
5380 VTAB 11:HTAB 65:INPUT"",A(113)
5385 VTAB 11:HTAB 65:PRINT" ":VTAB 11:HTAB 65:PRINT USING
     "################";A(113)
5390 VTAB 12:HTAB 65:INPUT"",A(114)
5395 VTAB 12:HTAB 65:PRINT" ":VTAB 12:HTAB 65:PRINT USING
     "################";A(114)
5400 VTAB 13:HTAB 65:INPUT"",A(115)
5405 VTAB 13:HTAB 65:PRINT" ":VTAB 13:HTAB 65:PRINT USING
     "################";A(115)
5410 IF A(3) < > 1 AND B(3) < > 1 THEN 5425
5415 VTAB 14:HTAB 65:INPUT"",A(116)
5420 VTAB 14:HTAB 65:PRINT" ":VTAB 14:HTAB 65:PRINT USING
     "################";A(116)
5425 IF A(3) < > 2 AND B(3) < > 2 THEN 5440
5430 VTAB 15:HTAB 65:INPUT"",A(117)
5435 VTAB 15:HTAB 65:PRINT" ":VTAB 15:HTAB 65:PRINT USING
     "################";A(117)
5440 VTAB 23:PRINT"Eingabefehler : sind alle Angaben
     richtig ? <J/N> .... ";:GET E$
5445 IF E$ = "J" OR E$ = "N" THEN VTAB 23:HTAB 65:PRINT"
     ";E$:ELSE 5440
5450 IF E$ = "N" THEN VTAB 3:HTAB 65:INVERSE:PRINT"->
     KORREKTUR <-":NORMAL
5455 IF E$ = "N" THEN VTAB 23:HTAB 79:PRINT" ";:GOTO 5315
5460 VTAB 3:HTAB 65:PRINT SPC(15)
5465 FOR I = 1 TO 3:VTAB 1:HTAB 60:PRINT SPC(16);:FOR S =
     1 TO 100:NEXT S:VTAB 1:HTAB 60:PRINT"Bitte warten
     ...":FOR S1 = 1 TO 100:NEXT S1:NEXT I
5470 CHAIN "MENUE"
```

N.BAS: Rechenteil

```
5475 REM RECHENTEIL
5480 COMMON A(), B(), C(), D(), E(), M(), N()
5485 IF A(118) = 1 THEN A(1) = A(1) + A(118)
5490 REM EINKUENFTE AUS LAND- U. FORSTWIRTSCHAFT
5495 C(1) = INT(A(15)): D(1) = INT(B(15))
5500 REM EINKUENFTE AUS DEM GEWERBEBETRIEB
5505 C(2) = INT(A(20)): D(2) = INT(B(20))
5510 REM EINKUENFTE AUS SELBSTAENDIGER ARBEIT
5515 C(3) = INT(A(25) + A(26) - A(27)): D(3) = INT(B(25)
     + B(26) - B(27))
5520 REM EINKUENFTE AUS NICHTSELBSTAENDIGER ARBEIT
5525 C(4) = INT(A(30) + A(31)): D(4) = INT(B(30) + B(31))
5530 V = INT(40 * A(31) / 100 + .5): IF V > 4800 THEN V =
     4800
5535 V1 = INT(40 * B(31) / 100 + .5): IF V1 > 4800 THEN
     V1 = 4800
5540 C(4) = C(4) - V - 600 - 480: D(4) = D(4) - V1 - 600
     - 480
5545 IF C(4) < 0 THEN C(4) = 0
5550 IF D(4) < 0 THEN D(4) = 0
5555 IF A(32) < 564 AND A(32) > C(4) THEN 5575
5560 IF A(32) < 564 THEN C(4) = C(4) - 564: MM = 1: IF
     C(4) < 0 THEN C(4) = 0
5565 IF MM < > 1 THEN 5575
5570 GOTO 5580
5575 C(4) = INT(C(4) - A(32))
5580 IF B(32) < 564 AND B(32) > D(4) THEN 5600
5585 IF B(32) < 564 THEN D(4) = D(4) - 564: NN = 1: IF
     D(4) < 0 THEN D(4) = 0
5590 IF NN < > 1 THEN 5600
5595 GOTO 5605
5600 D(4) = INT(D(4) - B(32))
5605 REM EINKUENFTE AUS KAPITALVERMOEGEN
5610 IF A(35) = 0 AND B(35) = 0 AND A(36) = 0 AND B(36) =
     0 THEN 5805
5615 P = 100: IF A(1) = 2 THEN P = 200
5620 IF ((A(36) + B(36)) < = P) AND ((A(36) + B(36)) >
     (A(35) + B(35))) THEN KK = 1: GOTO 5655
5625 IF P < (A(36) + B(36)) THEN 5655
5630 C(5) = INT(A(35) - (P * A(35) / (A(35) + B(35))) +
     .5)
5635 IF C(5) < 0 THEN C(5) = 0
5640 D(5) = INT(B(35) - (P * B(35) / (A(35) + B(35))) +
     .5)
5645 IF D(5) < 0 THEN D(5) = 0
5650 GOTO 5665
5655 C(5) = INT(A(35) - A(36)): D(5) = INT(B(35) - B(36))
5660 IF KK = 1 THEN 5805
5665 IF A(1) = 1 THEN 5690
5670 IF C(5) < = 0 AND D(5) < = 0 THEN 5715
5675 IF C(5) < = 0 AND D(5) > 0 THEN 5725
5680 IF C(5) > 0 AND D(5) < = 0 THEN 5745
5685 GOTO 5765
5690 S0 = 300
5695 IF C(5) < 0 THEN S0 = 0: GOTO 5705
5700 IF C(5) < 300 THEN S0 = C(5)
5705 C(5) = C(5) - S0
5710 GOTO 5790
5715 S0 = 0: SP = 0
5720 GOTO 5790
5725 S0 = 0: SP = 600
5730 D(5) = D(5) - 600
5735 IF D(5) < 0 THEN D(5) = 0: GOTO 5790
5740 GOTO 5790
5745 S0 = 600: SP = 0
5750 C(5) = C(5) - 600
5755 IF C(5) < 0 THEN C(5) = 0: GOTO 5790
5760 GOTO 5790
5765 S0 = 300: SP = 300
5770 IF C(5) > 300 AND D(5) > 300 THEN C(5) = C(5) - 300:
     D(5) = D(5) - 300: GOTO 5790
5775 IF C(5) < 300 AND D(5) < 300 THEN C(5) = 0: D(5) =
     0: GOTO 5790
5780 IF C(5) < 300 THEN SP = 300 - C(5): C(5) = 0: D(5) =
     D(5) - 300 - SP: GOTO 5790
5785 S0 = 300 - D(5): D(5) = 0: C(5) = C(5) - 300 - S0:
     GOTO 5790
5790 IF (C(5) < 0) AND (S0 > 0) THEN C(5) = 0
5795 IF (D(5) < 0) AND (SP > 0) THEN D(5) = 0
5800 GOTO 5805
5805 REM EINKUENFTE AUS VERMIETUNG U. VERPACHTUNG
5810 A = A(40) * 1.4 / 100 * A(41) / 100 * A(42) / 12
5815 B = B(40) * 1.4 / 100 * B(41) / 100 * B(42) / 12
5820 C = A(44): IF (A(44) > A) THEN C = A: IF A(49) = 1
     THEN C = A(44): IF C > 10000 THEN C = 10000
5825 D = B(44): IF (B(44) > B) THEN D = B: IF B(49) = 1
     THEN D = B(44): IF D > 10000 THEN D = 10000
5830 A = A - C: B = B - D
5835 C(6) = INT(A - A(43) - A(45) + A(46) - A(47) + .5)
5840 D(6) = INT(B - B(43) - B(45) + B(46) - B(47) + .5)
5845 REM SONSTIGE EINKUENFTE
5850 P = 200: P1 = 200
5855 IF A(51) < = P AND A(51) > A(50) THEN 5875
5860 IF A(51) > P THEN 5875
5865 C(7) = A(50) - P: IF C(7) < 0 THEN C(7) = 0
5870 GOTO 5880
5875 C(7) = A(50) - A(51)
5880 IF B(51) < = P1 AND B(51) > B(50) THEN 5900
5885 IF B(51) > P1 THEN 5900
5890 D(7) = B(50) - P1: IF D(7) < 0 THEN D(7) = 0
5895 GOTO 5905
5900 D(7) = B(50) - B(51)
5905 C(7) = INT(C(7) + A(52) - A(53) + .5)
5910 D(7) = INT(D(7) + B(52) - B(53) + .5)
5915 REM GESAMTSUMME DER EINKUENFTE
5920 C(8) = 0: FOR I = 1 TO 7: C(8) = C(8) + C(I): NEXT I
5925 D(8) = 0: FOR I = 1 TO 7: D(8) = D(8) + D(I): NEXT I
5930 REM BETRAG DER ALTERSENTLASTUNG
5935 IF A(2) < > 7 THEN 5965
5940 A = 40 * A(30) / 100
5945 B = C(1) + C(2) + C(3) + C(4) + C(5) + C(6)
5950 C(13) = A: IF B > 0 THEN C(13) = C(13) + B * 40 /
     100
5955 IF C(13) > 3000 THEN C(13) = 3000
5960 C(13) = INT(C(13))
5965 IF B(2) < > 7 THEN 5995
5970 A = 40 * B(30) / 100
5975 B = D(1) + D(2) + D(3) + D(5) + D(6)
5980 D(13) = A: IF B > 0 THEN D(13) = D(13) + B * 40 /
     100
5985 IF D(13) > 3000 THEN D(13) = 3000
5990 D(13) = INT(D(13))
5995 REM AUZUGSBETRAG FUER AUSBILDUNGSPLATZSCHAFFUNG
```

```
6000 C(14) = INT(A(101) + .5)
6005 D(14) = INT(B(101) + .5)
6010 REM FREIBETRAG FUER LAND- U. FORSTWIRTSCHAFT
6015 IF A(15) = 0 AND B(15) = 0 THEN 6070
6020 A = 2000: IF A(1) = 2 THEN A = 4000
6025 IF (C(1) < 0) OR (D(1) < 0) THEN 6040
6030 B = A * C(1) / (C(1) + D(1))
6035 C = A * D(1) / (C(1) + D(1))
6040 IF D(1) < 0 THEN B = A
6045 IF B > C(1) THEN B = C(1)
6050 IF B < 0 THEN B = 0
6055 IF C > D(1) THEN C = D(1)
6060 IF C < 0 THEN C = 0
6065 C(15) = INT(B + .5): D(15) = INT(C + .5)
6070 REM AUSLAENDISCHE STEUERN
6075 C(12) = INT(A(100))
6080 IF A(99) = 2 THEN C(12) = C(12) + INT(A(98))
6085 IF A(1) = 2 THEN D(12) = INT(C(12) / 2): C(12) =
     INT(C(12) / 2 + .9)
6090 REM GESAMTBETRAG DER EINKUENFTE
6095 C(16) = C(8) - C(12) - C(13) - C(14) - C(15)
6100 D(16) = D(8) - D(12) - D(13) - D(14) - D(15)
6105 C(16) = C(16) + D(16)
6110 REM SONDERAUSGABEN OHNE VORSORGEPAUSCHALE
6115 REM SPENDEN
6120 SP = 0
6125 IF A(77) < > 0 THEN 6155
6130 A = 5 / 100 * C(16)
6135 IF A < 0 THEN A = 0
6140 B = A(78): IF (A(78) > 2 * A) THEN B = 2 * A
6145 C = A(75) - A(78): IF (C > A) THEN C = A
6150 SP = INT(B + C + .5): GOTO 6170
6155 A = 2 / 1000 * A(77)
6160 B = A(75): IF (B > A) THEN B = A
6165 SP = INT(B + .5)
6170 A = A(76): P = 1800: IF A(1) = 2 THEN P = 3600
6175 IF A > P THEN A = P
6180 SP = INT(SP + A + .5)
6185 C(18) = INT(SP + A(70) + .5)
6190 P = 270: IF A(1) = 2 THEN P = 540
6195 IF C(18) < P THEN C(18) = P
6200 REM HOECHSTBETRAG
6205 A = 0: B = 0: B1 = 0: P = 0: H = 0
6210 N = 0: M = 0: N = A(73): M = B(73)
6215 A = A(71) + A(72) + B(72)
6220 B = 9 * A(30) / 100: B1 = 9 * B(30) / 100
6225 IF A(30) > 62400! THEN B = 62400! * 9 / 100
6230 IF B(30) > 62400! THEN B1 = 62400! * 9 / 100
6235 IF A(73) = -1 THEN B = 0: N = 0
6240 IF B(73) = -1 THEN B1 = 0: M = 0
6245 P = 3000: IF A(1) = 2 THEN P = 6000
6250 P = P - N - M
6255 IF A(72) = 0 THEN P = P - B
6260 IF B(72) = 0 THEN P = P - B1
6265 IF P < 0 THEN P = 0
6270 IF (P > A) THEN P = A
6275 H = P: A = A - P: B = 0: B1 = 0
6280 A = A + A(74)
6285 P = 2340: IF A(1) = 2 THEN P = 4680
6290 P = P + 600 * A(4) + 300 * A(5)
6295 IF (P > A) THEN H = H + A: GOTO 6320
6300 H = H + P: A = A - P
6305 P = P / 2: A = A / 2
6310 IF A < P THEN H = H + A
6315 IF (P < = A) THEN H = H + P
6320 REM VORSORGEPAUSCHALE
6325 E(2) = H: REM HOECHSTBETRAG
6330 A = 40 * A(30) / 100: IF A > 3000 THEN A = 3000
6335 A1 = 40 * B(30) / 100: IF A1 > 3000 THEN A1 = 3000
6340 IF A(2) < > 7 THEN A = 0
6345 IF B(2) < > 7 THEN A1 = 0
6350 B = A(30) + A(31): B1 = B(30) + B(31)
6355 IF B = 0 AND B1 = 0 THEN 6570
6360 B = B - V - 600 - A: IF B < 0 THEN B = 0
6365 B1 = B1 - V1 - 600 - A1: IF B1 < 0 THEN B1 = 0
6370 IF B > 62400! THEN B = 62400!
6375 IF B1 > 62400! THEN B1 = 62400!
6380 K1 = 9 / 100 * B: K2 = 9 / 100 * B1: K3 = 18 / 100 *
     B: K4 = 18 / 100 * B1
6385 M = 0: N = 0
6390 IF A(73) < = 0 THEN M = 1
6395 IF B(73) < = 0 THEN N = 1
6400 IF A(1) = 2 AND ((M + N = 0) OR (M + N = 2)) THEN
     6445
6405 IF A(1) = 2 AND M = 1 AND N = 0 THEN 6475
6410 IF A(1) = 2 AND M = 0 AND N = 1 THEN 6520
6415 REM VORSORGEPAUSCHALE FUER LEDIGE
6420 P = 2340 + 600 * A(4) + 300 * A(5): IF M = 1 THEN P
     = 1000 + 600 * A(4) + 300 * A(5)
6425 P1 = 1170 + 300 * A(4) + 150 * A(5): IF M = 1 THEN
     P1 = 1000 + 300 * A(4) + 150 * A(5)
6430 IF K1 < P THEN P = K1
6435 IF K1 < P1 THEN P1 = K1
6440 GOTO 6560
6445 REM VORSORGEPAUSCHALE FUER BEIDE EHEGATTEN
6450 P = 4680 + 600 * A(4) + 300 * A(5): IF M + N = 2
     THEN P = 2000 + 600 * A(4) + 300 * A(5)
6455 P1 = 2340 + 300 * A(4) + 150 * A(5): IF M + N = 2
     THEN P1 = 2000 + 300 * A(4) + 150 * A(5)
6460 IF (K1 + K2) < P THEN P = K1 + K2
6465 IF (K1 + K2) < P1 THEN P1 = K1 + K2
6470 GOTO 6560
6475 REM VORSORGEPAUSCHALE FUER DEN STEUERPFLICHTIGEN
     EHEGATTEN
6480 P1 = 0: A1 = K1: IF K1 > (1000 + 600 * A(4) + 300 *
     A(5)) THEN A1 = 1000 + 600 * A(4) + 300 * A(5)
6485 A2 = K1: IF K1 > (1000 + 300 * A(4) + 150 * A(5))
     THEN A2 = 1000 + 300 * A(4) + 150 * A(5)
6490 T1 = A1 + A2 + K4: IF T1 > (4680 + 600 * A(4) + 300
     * A(5)) THEN T1 = 4680 + 600 * A(4) + 300 * A(5)
6495 T2 = (A1 + A2 + K4 - 4680 - 600 * A(4) - 300 * A(5))
     / 2: IF T2 > (4680 + 600 * A(4) + 300 * A(5)) / 2
     THEN T2 = (4680 + 600 * A(4) + 300 * A(5)) / 2
6500 T1 = T1 + T2: A1 = K1: IF K1 > (2000 + 600 * A(4) +
     300 * A(5)) THEN A1 = 2000 + 600 * A(4) + 300 * A(5)
6505 A2 = K1: IF K1 > (2000 + 300 * A(4) + 150 * A(5))
     THEN A2 = 2000 + 300 * A(4) + 150 * A(5)
6510 A1 = A1 + A2: P = T1: IF A1 > T1 THEN P = A1
6515 GOTO 6560
6520 REM VORSORGEPAUSCHALE FUER DEN STEUERPFLICHTIGEN
     EHEGATTEN
6525 P1 = 0: A1 = K2: IF K2 > (1000 + 600 * A(4) + 300 *
     A(5)) THEN A1 = 1000 + 600 * A(4) + 300 * A(5)
6530 A2 = K2: IF K2 > (1000 + 300 * A(4) + 150 * A(5))
     THEN A2 = 1000 + 300 * A(4) + 150 * A(5)
6535 T1 = A1 + A2 + K3: IF T1 > (4680 + 600 * A(4) + 300
     * A(5)) THEN T1 = 4680 + 600 * A(4) + 300 * A(5)
6540 T2 = (A1 + A2 + K3 - 4680 - 600 * A(4) - 300 * A(5))
     / 2: IF T2 > (4680 + 600 * A(4) + 300 * A(5)) / 2
     THEN T2 = (4680 + 600 * A(4) + 300 * A(5)) / 2
6545 T1 = T1 + T2: A1 = K1: IF K2 > (2000 + 600 * A(4) +
     300 * A(5)) THEN A1 = 2000 + 600 * A(4) + 300 * A(5)
6550 A2 = K2: IF K2 > (2000 + 300 * A(4) + 150 * A(5))
     THEN A2 = 2000 + 300 * A(4) + 150 * A(5))
6555 A1 = A1 + A2: P = T1: IF A1 > T1 THEN P = A1
6560 P = P + P1
6565 VP = INT(P / 54) * 54: IF VP < 300 * A(1) THEN VP =
     300 *A(1)
6570 IF (A(30) + A(31) + B(30) + B(31)) = 0 THEN VP = 300
     * A(1)
6575 E(3) = VP: REM VORSORGEPAUSCHALE
6580 C(19) = H: IF VP > H THEN C(19) = VP
6585 C(19) = INT(C(19) + .5)
6590 REM FREIBETRAG FUER FREIBERUFLICHE
6595 A = 0: B = 0
6600 IF A(25) < = C(18) - C(3) THEN 6615
6605 A = INT(A(25) * 5 / 100 + .5)
6610 IF A > 1200 THEN A = 1200
6615 IF B(25) < = D(8) - D(3) THEN 6630
6620 B = INT(B(25) * 5 / 100 + .5)
6625 IF B > 1200 THEN B = 1200
6630 IF A < 0 THEN A = 0
6635 IF B < 0 THEN B = 0
6640 C(21) = A + B
6645 REM AUSSERGEWOEHNLICHE BELASTUNGEN
6650 A = A(4) + A(5): B = A(1): C = C(16)
6655 IF B = 1 AND A = 0 AND C < = 30000 THEN D = 5
6660 IF B = 1 AND A = 0 AND C > 30000 THEN D = 6
6665 IF B = 1 AND A = 0 AND C > 100000! THEN D = 7
6670 IF B = 2 AND A = 0 AND C < = 30000 THEN D = 4
6675 IF B = 2 AND A = 0 AND C > 30000 THEN D = 5
6680 IF B = 2 AND A = 0 AND C > 100000! THEN D = 6
6685 IF A < = 2 AND A > 0 AND C < = 30000 THEN D = 2
6690 IF A < = 2 AND A > 0 AND C > 30000 THEN D = 3
6695 IF A < = 2 AND A > 0 AND C > 100000! THEN D = 4
6700 IF A > = 3 AND C < = 100000! THEN D = 1
6705 IF A > = 3 AND C > 100000! THEN D = 2
6710 A = D * C(16) / 100
6715 IF A(85) = 0 THEN A = 0
6720 IF (A(85) < = A) THEN A = A(85)
6725 E(5) = A
6730 REM AUSSERGEWOEHNLICHE BELASTUNGEN
     (DDR/HAUSHALTSHILFEN)
6735 C(22) = INT(A(85) - A + A(86) + A(88) + A(89) * 620
     + A(90) + A(91) + .5)
6740 A = A(93): GOSUB 6745: A = B(93): GOSUB 6745: GOTO
     6805
6745 B = 0
6750 IF A > = 25 AND A < = 34 THEN B = 600
6755 IF A > = 35 AND A < = 44 THEN B = 840
6760 IF A > = 45 AND A < = 54 THEN B = 1110
6765 IF A > = 55 AND A < = 64 THEN B = 1410
6770 IF A > = 65 AND A < = 74 THEN B = 1740
6775 IF A > = 75 AND A < = 84 THEN B = 2070
6780 IF A > = 85 AND A < = 90 THEN B = 2400
6785 IF A > = 91 AND A < = 100 THEN B = 2760
6790 IF A = 300 THEN B = 7200
6795 IF A = 1 THEN B = B + 720
6800 C(22) = C(22) + B:RETURN
6805 A = 0: B = 0
6810 C(24) = C(16) - C(18) - C(19) - C(21) - C(22): REM
     EINKOMMEN
```

```
6815 E(11) = A(80): E(12) = A(81): E(13) = A(82): E(14) =
     B(79): E(15) = B(80): E(16) = B(81)
6820 IF C(24) < = 0 THEN 6900
6825 C(23) = C(24)
6830 C(24) = C(24) - A(79): IF C(24) < = 0 THEN C(24) =
     0:GOTO 6900
6835 C(24) = C(24) - B(80): IF C(24) < = 0 THEN E(15) =
     E(15) + C(24): C(24) = 0: GOTO 6900
6840 E(15) = 0
6845 C(24) = C(24) - A(80): IF C(24) < = 0 THEN E(11) =
     E(11) + C(24): C(24) = 0: GOTO 6900
6850 E(11) = 0
6855 C(24) = C(24) - A(81): IF C(24) < = 0 THEN E(12) =
     E(12) + C(24): C(24) = 0: GOTO 6900
6860 E(12) = 0
6865 C(24) = C(24) - A(82): IF C(24) < = 0 THEN E(13) =
     E(13) + C(24): C(24) = 0: GOTO 6900
6870 E(13) = 0
6875 C(24) = C(24) - B(79): IF C(24) < = 0 THEN E(14) =
     E(14) + C(24): C(24) = 0: GOTO 6900
6880 E(14) = 0
6885 C(24) = C(24) - B(81): IF C(24) < = 0 THEN E(16) =
     E(16) + C(24): C(24) = 0: GOTO 6900
6890 E(16) = 0
6895 C(23) = C(23) - C(24)
6900 REM ALTERSFREIBETRAG
6905 IF A(2) = 7 THEN C(25) = 720
6910 IF B(2) = 7 THEN C(25) = C(25) + 720
6915 REM HAUSHALTSFREIBETRAG
6920 IF A(1) = 1 AND (A(4) + A(5)) > 0 THEN C(26) = 4212
6925 REM KINDERFREIBETRAG
6930 C(27) = A(4) * 432 + A(5) * 216
6935 REM ZU VERSTEUERNDES EINKOMMEN
6940 C(29) = C(24) - C(25) - C(26) - C(27): IF C(29) < 0
     THEN C(29) = 0
6945 REM ABZUGSBETRAG NACH PARAGRAPH 34F
6950 C(47) = A(48) * 600: C(40) = INT(A(95) + .5): C(41)
     = INT(A(96) + .5)
6955 REM EINKOMMENSTEUER-BERECHNUNG
6960 IF A(1) = 1 THEN C(30) = 1
6965 IF A(1) = 2 THEN C(30) = 2
6970 IF A(6) = 1 THEN C(30) = 2
6975 AU = INT(A(16) + B(16) + A(21) + B(21) + A(28) +
     B(28) + A(60) + B(60) + A(61) + B(61)): AN = 0: AV =
     AU
6980 IF AU > C(29) THEN AU = C(29): AN = 4212
6985 IF C(29) < = 0 THEN AN = 0
6990 FOR I = AN TO AU: REM KUERZUNGSBETRAG = I
6995 T1 = 0: T2 = 0: AV = AU - I
7000 S1 = C(29) - AV + A(65) + A(67) + B(67) - A(66): E =
     S1: CP = C(30)
7005 GOSUB 7075: IF S1 > 0 THEN T1 = INT((C(29) - AV) *
     ST / S1)
7010 S2 = C(29) + A(65) + A(67) + B(67): E = S2: CP =
     C(30)
7015 GOSUB 7075: IF S2 > 0 THEN T2 = AV * ST / S2 / 2
7020 T = T1 + T2: IF (T > = TV) AND (I > AN) THEN 7030
7025 TV = T: T3 = T1: T4 = T2: S3 = S1: S4 = S2: NEXT I
7030 E(17) = C(29) - AV
7035 IF (A(65) > 0) OR (A(66) > 0) OR (A(67) > 0) OR
     (B(67) > 0) THEN E(18) = E(17): E(17) = 0
7040 E(19) = AV: C(31) = INT (T3 * 100) / 100
7045 IF (C(29) - AV) < = 0 THEN 7055
7050 IF S3 > 0 THEN E(6) = INT (T3 * 100 / (C(29) - AV) *
     100) / 100
7055 IF AU > 0 THEN E(8) = INT (T4 * 100 / AV * 100) /
     100
7060 E(9) = I - 1: IF (A(65) > 0) OR (A(66) > 0) OR
     (A(67) > 0) OR (B(67) > 0) THEN C(32) = C(31): C(31)
     = 0: E(7) = E(6): E(6) = 0
7065 C(33) = INT (T4): C(34) = C(31) + C(33): IF C(31) =
     0 THEN C(34) = C(32) + C(33)
7070 GOTO 7155
7075 REM UNTERPROGRAMM - TABELLE
7080 IF CP = 2 THEN E = E / 2
7085 E = INT(E / 54) * 54
7090 IF E < = 4212 THEN ST = 0: GOTO 7140
7095 IF E > 18000 THEN 7105
7100 ST = INT(.22 * E - 926): GOTO 7140
7105 IF E > = 60000! THEN 7120
7110 Y = (E - 18000) / 10000
7115 ST = INT((((3.05 * Y - 73.76) * Y + 695) * Y + 2200)
     * Y + 3034): GOTO 7140
7120 IF E > 130000! THEN 7135
7125 Z = (E - 60000!) / 10000
7130 ST = INT((((.09 * Z - 5.45) * Z + 88.13) * Z + 5040)
     * Z + 20018): GOTO 7140
7135 ST = INT(.56 * E - 14837)
7140 IF ST < 0 THEN ST = 0
7145 IF CP = 2 THEN ST = ST * 2
7150 RETURN
7155 REM AUSLAENDISCHE STEUER
7160 IF C(16) < = 0 THEN 7185
7165 AS = INT(A(97) * C(34) / C(16))
7170 C(35) = INT(A(98))
7175 IF AS < = A(98) THEN C(35) = AS
7180 IF A(99) = 2 THEN C(35) = 0
7185 REM ERMAESS. F. ERFINDER
7190 IF C(16) < = 0 THEN 7205
7195 C(37) = INT(A(102) * C(34) / C(16) / 2 + .5)
7200 IF C(16) < A(102) THEN C(37) = INT(C(16) * C(34) /
     C(16) / 2 + .5)
7205 REM ERMAESSIGUNG FUER LAND- U. FORSTWIRTSCHAFT
7210 IF C(29) < = 0 THEN 7290
7215 A = C(1): B = C(15): C = A(103): D = A(104): GOSUB
     7240: V1 = U: IF V1 < 0 THEN V1 = 0
7220 A = D(1): B = D(15): C = B(103): D = B(104): GOSUB
     7240: V2 = U: IF V2 < 0 THEN V2 = 0
7225 A = C(1): B = C(15): C = A(105): D = A(106): GOSUB
     7240: V3 = U: IF V3 < 0 THEN V3 = 0
7230 A = D(1): B = D(15): C = B(105): D = B(106): GOSUB
     7240: V4 = U: IF V4 < 0 THEN V4 = 0
7235 GOTO 7260
7240 U = C(34) / C(29) / 2 * (A - B): V = 1000 - ((C -
     50000!) / 10)
7245 IF D > 0 THEN V = D
7250 IF U > V THEN U = V
7255 RETURN
7260 IF (A(105) = 0) AND (A(106) = 0) THEN V3 = 0
7265 IF (B(105) = 0) AND (B(106) = 0) THEN V4 = 0
7270 AA = 2000: IF (A(1) = 2) AND (C(1) > 0) AND (D(1) >
     0) THEN AA = 1000
7275 C(36) = V1 + V3: IF (C(36) > AA) THEN C(36) = AA
7280 D(36) = V2 + V4: IF (D(36) > AA) THEN D(36) = AA
7285 C(36) = INT(C(36) + D(36) + .5): D(36) = 0
7290 REM FESTZUSETZENDE EINKOMMENSTEUER
7295 C(44) = C(34) - C(35) - C(36) - C(37) - C(38) +
     C(39) - C(40) - C(41) - C(42) + C(43)
7300 IF C(44) < 0 THEN C(44) = 0
7305 REM KIRCHENSTEUER-BERECHNUNG
7310 T = C(44)
7315 IF A(4) = 1 THEN T = C(44) - 600
7320 IF A(4) = 2 THEN T = C(44) - 600 - 960
7325 IF A(4) > 2 THEN T = C(44) - 600 - 960 - ((A(4) - 2)
     * 1800)
7330 IF T < = 0 THEN T = 0
7335 REM STEUER AUS S.D.E.
7340 CP = 1: E = C(8): GOSUB 7080: T3 = ST
7345 CP = 1: E = D(8): GOSUB 7080: T4 = ST
7350 IF A(1) = 1 THEN 7385
7355 IF (A(3) > 0) AND (B(3) > 0) THEN 7385
7360 IF A(3) = 0 THEN 7370
7365 KI = T * T3 / (T3 + T4): KI = 9 * KI / 100
7370 IF B(3) = 0 THEN 7390
7375 KI = T * T4 / (T3 + T4): KI = 9 * KI / 100
7380 GOTO 7390
7385 KI = 9 * T / 100
7390 REM KIRCHENSTEUER FUER LEDIGE
7395 KI = INT(KI * 100) / 100
7400 IF A(3) = 1 THEN C(45) = KI
7405 IF A(3) = 2 THEN C(46) = KI
7410 IF A(1) = 1 THEN 7455
7415 REM KIRCHENSTEUER FUER EHEGATTEN
7420 IF A(3) = 1 THEN C(45) = .5
7425 IF A(3) = 2 THEN C(46) = .5
7430 IF B(3) = 1 THEN C(45) = C(45) + .5
7435 IF B(3) = 2 THEN C(46) = C(46) + .5
7440 IF C(45) = .5 AND C(46) = 0 THEN C(45) = 1
7445 IF C(46) = .5 AND C(45) = 0 THEN C(46) = 1
7450 C(45) = C(45) * KI: C(46) = C(46) * KI
7455 REM AN- U. ABRECHNUNG
7460 C(50) = INT(A(113) + .9): C(51) = INT(A(114) + .9)
7465 C(53) = INT(A(110) + .9): C(54) = INT(A(115) + .9)
7470 C(56) = INT(A(111) + .9): C(57) = INT(A(112) + .9)
7475 C(58) = INT(A(116) + .9): C(59) = INT(A(117) + .9)
7480 IF C(47) > C(44) THEN C(47) = C(44)
7485 C(52) = C(44) - C(50) - C(51) - C(47): C(55) = C(52)
     - C(53) - C(54)
7490 C(60) = C(45) - C(56) - C(58): C(61) = C(46) - C(57)
     - C(59)
7495 C(62) = C(60) + C(61): C(63) = C(62) + C(55)
7500 REM AUSDRUCK AUF DEM BILDSCHIRM
7505 HOME:PRINT"An- u. Abrechnung : ";:HTAB
     69:INVERSE:PRINT"SEITE 1":NORMAL
7510 FOR I = 1 TO 79:PRINT"-";:NEXT I:PRINT
7515 VTAB 4:PRINT"E i n k ü n f t e : Ehemann: Ehefrau:
7520 FOR I = 1 TO 79:PRINT"*";: NEXT I
7525 PRINT"Land- u. Forstwirtschaft";:HTAB 44:PRINT USING
     "########";C(1);:HTAB 62:PRINT USING "########";D(1)
7530 PRINT"Gewerbebetrieb";:HTAB 44:PRINT USING
     "########";C(2);:HTAB 62:PRINT USING "########";D(2)
7535 PRINT"Selbständiger Arbeit";:HTAB 44:PRINT USING
     "########";C(3);:HTAB 62:PRINT USING "########";D(3)
7540 PRINT"Nichtselbständiger Arbeit";:HTAB 44:PRINT
     USING "########";C(4);:HTAB 62:PRINT USING
     "########";D(4)
7545 PRINT"Kapitalvermögen";:HTAB 44:PRINT USING
     "########";C(5);:HTAB 62:PRINT USING "########";D(5)
7570 PRINT:PRINT"Alterentlastungsbetrag";:HTAB 44:PRINT
     USING "########";C(13);:HTAB 62:PRINT USING
     "########";D(13)
7575 PRINT"Ausbildungsplatz-Abzugsbetrag";:HTAB 44:PRINT
```

```
    USING "########";C(14);:HTAB 62:PRINT USING
    "########";D(14)
7580 PRINT"Freibetrag für Land- u. Forstwirtschaft";:HTAB
    44:PRINT USING "########";C(15);:HTAB 62:PRINT USING
    "########";D(15)
7585 PRINT"Ausländische Steuer n. Par.34c";:HTAB 44:PRINT
    USING "########";C(12);:HTAB 62:PRINT USING
    "########";D(12)
7590 FOR I = 1 TO 79:PRINT"=";:NEXT I
7595 PRINT"Gesamtbetrag der Einkünfte für
    Ehemann/frau";:HTAB 55:PRINT USING "########";C(16)
7600 VTAB 23:HTAB 25:PRINT"--> ";:INVERSE:PRINT"Drücken
    Sie eine Taste";:NORMAL:PRINT" <--":GET Y$
7605 HOME:PRINT"An- u. Abrechnung : ";:HTAB 69:INVERSE:
    PRINT"SEITE 2":NORMAL
7610 FOR I = 1 TO 79:PRINT"-";:NEXT I:PRINT
7615 VTAB 6:PRINT"Ausgaben, Aufwendungen, Freibeträge
    Ehemann/Ehefrau"
7620 FOR I = 1 TO 79:PRINT"*";:NEXT I
7625 PRINT"Sonderausgaben";:HTAB 60:PRINT USING
    "########";C(18)
7630 PRINT"Vorsorgeaufwendungen";:HTAB 60:PRINT USING
    "########";C(19)
7635 PRINT"Freibetrag für Freiberufler";:HTAB 60:PRINT
    USING "########";C(21)
7640 PRINT"Außergewöhnliche Belastungen";:HTAB 60:PRINT
    USING "########";C(22)
7645 PRINT"Verlustabzüg";:HTAB 60:PRINT USING
    "########";C(23)
7650 FOR I = 1 TO 79:PRINT"=";:NEXT I
7655 PRINT"Einkommen";:HTAB 60:PRINT USING
    "########";C(24)
7660 PRINT:PRINT"Altersfreibetrag";:HTAB 60:PRINT USING
    "########";C(25)
7665 PRINT"Haushaltsfreibetrag";:HTAB 60:PRINT USING
    "########";C(26)
7670 PRINT"Kinderfreibetrag";:HTAB 60:PRINT USING
    "########";C(27)
7675 FOR I = 1 TO 79:PRINT"=";:NEXT I
7680 PRINT"Zu versteuerndes Einkommen";:HTAB 60:PRINT
    USING "########";C(29)
7685 VTAB 23:HTAB 23:PRINT"--> ";:INVERSE:PRINT"<W>eiter,
    <Z>urück ... <W/Z>";:NORMAL:PRINT" <--":GET Y$
7690 IF Y$ = "W" OR Y$ = "Z" THEN 7695:ELSE 7685
7695 IF Y$ = "Z" THEN 7505
7700 HOME:PRINT"An- u. Abrechnung : ";:HTAB
    69:INVERSE:PRINT"SEITE 3":NORMAL
7705 FOR I = 1 TO 79:PRINT"-";:NEXT I:PRINT
7710 VTAB 5:PRINT"Einkommen/Lohn-Steuer Ehemann/Ehefrau"
7715 FOR I = 1 TO 79:PRINT"*";:NEXT I
7720 PRINT"laut ";
7725 IF C(30) = 1 THEN PRINT"Grundtabelle ";
7730 IF C(30) = 2 THEN PRINT"Splitting ";
7735 PRINT"<";E(17);" * ";E(6);" %>";:HTAB 60:PRINT USING
    "########";C(31)
7740 PRINT"laut Progressionsvorbehalt <";E(18);" *
    ";E(7)" %>";:HTAB 60:PRINT USING "########";C(32)
7745 PRINT"Ermäßigte Steuer n.Par.34 <";E(19);" * ";E(8)"
    %>";:HTAB 60:PRINT USING "########";C(33)
7750 FOR I = 1 TO 79:PRINT"=";:NEXT I
7755 PRINT"Tarifliche Einkommen/Lohn-Steuer";:HTAB
    60:PRINT USING "########";C(34)
7760 PRINT:PRINT"Ausländische Steuer n. Par.34c";:HTAB
    60:PRINT USING "########";C(35)
7765 PRINT"Ermäßigung aus Land- u. Forstwirtschaft";:HTAB
    60:PRINT USING "########";C(36)
7770 PRINT"Ermäßigung für Erfinder";:HTAB 60:PRINT USING
    "########";C(37)
7775 PRINT"Ermäßigung Berlin FG n. Par. 16, 17";:HTAB
    60:PRINT USING "########";C(40)
7780 PRINT"Ermäßigung Vermögensbildungsgestz n. Par.
    14";:HTAB 60:PRINT USING "########";C(41)
7785 FOR I = 1 TO 79:PRINT"=";:NEXT I
7790 PRINT"Festzusetzende Einkommen/Lohn-Steuer";:HTAB
    60:PRINT USING "########";C(44)
7795 PRINT"Festzusetzende Kirchensteuer ev.";:HTAB
    60:PRINT USING "########";C(45)
7800 PRINT"Festzusetzende Kirchensteuer rk.";:HTAB
    60:PRINT USING "########";C(46)
7805 VTAB 23:HTAB 23:PRINT"--> ";:INVERSE:PRINT"<W>eiter,
    <Z>urück ... <W/Z>";:NORMAL:PRINT" <--":GET Y$
7810 IF Y$ = "W" OR Y$ = "Z" THEN 7815:ELSE 7805
7815 IF Y$ = "Z" THEN 7605
7820 HOME:PRINT"An- u. Abrechnung : ";:HTAB
    69:INVERSE:PRINT"SEITE 4":NORMAL
7825 FOR I = 1 TO 79:PRINT"-";:NEXT I:PRINT
7830 VTAB 6:PRINT"Einkommen/Lohn-Steuer Ehemann/Ehefrau"
7835 FOR I = 1 TO 79:PRINT"*";:NEXT I
7840 PRINT"Festzusetzende Einkommen/Lohn-Steuer";:HTAB
    60:PRINT USING "########";C(44)
7845 PRINT"Abzug n. Par. 34f";:HTAB 60:PRINT USING
    "########";C(47)
7850 PRINT"Kapitalertragssteuer";:HTAB 60:PRINT USING
    "########";C(50)
7855 PRINT"Anrechnung Körperschaftssteuer";:HTAB 60:PRINT
    USING "########";C(51)
7860 FOR I = 1 TO 79:PRINT"=";:NEXT I
7865 PRINT"Verbleiben";:HTAB 60:PRINT USING
    "########";C(52)
7870 PRINT:PRINT"Lohnsteuer";:HTAB 60:PRINT USING
    "########";C(53)
7875 PRINT"Vorausgezahlte Einkommen/Lohn-Steuer";:HTAB
    60:PRINT USING "########";C(54)
7880 FOR I = 1 TO 79:PRINT"=";:NEXT I
7885 PRINT"Einkommen/Lohn-Steuer";:HTAB 60:PRINT USING
    "########";C(55)
7890 VTAB 23:HTAB 23:PRINT"--> ";:INVERSE:PRINT"<W>eiter,
    <Z>urück ... <W/Z>";:NORMAL:PRINT" <--":GET Y$
7895 IF Y$ = "W" OR Y$ = "Z" THEN 7900:ELSE 7890
7900 IF Y$ = "Z" THEN 7700
7905 HOME:PRINT"An- u. Abrechnung : ";:HTAB
    70:INVERSE:PRINT"SEITE 5":NORMAL
7910 FOR I = 1 TO 79:PRINT"-";:NEXT I:PRINT
7915 VTAB 6:PRINT"Einkommen/Lohn-Steuer Ehemann/Ehefrau"
7920 FOR I = 1 TO 79:PRINT"*";:NEXT I
7925 PRINT"Einkommen/Lohn-Steuer";:HTAB 60:PRINT USING
    "########";C(55)
7930 PRINT"Festgesetzte Kirchensteuer ev.";:HTAB 60:PRINT
    USING "########";C(45)
7935 PRINT"Festgesetzte Kirchensteuer rk.";:HTAB 60:PRINT
    USING "########";C(46)
7940 PRINT"Lohn-Kirchensteuer ev.";:HTAB 60:PRINT USING
    "########";C(56)
7945 PRINT"Lohn-Kirchensteuer rk.";:HTAB 60:PRINT USING
    "########";C(57)
7950 PRINT"Vorausgezahlte Kirchensteuer ev.";:HTAB
    60:PRINT USING "########";C(58)
7955 PRINT"Vorausgezahlte Kirchensteuer rk.";:HTAB
    60:PRINT USING "########";C(59)
7960 FOR I = 1 TO 79:PRINT"=";:NEXT I
7965 PRINT"Restbetrag Kirchensteuer ev.";:HTAB 60:PRINT
    USING "########";C(60)
7970 PRINT"Restbetrag Kirchensteuer rk.";:HTAB 60:PRINT
    USING "########";C(61)
7975 PRINT"Restbetrag Gesamt-Kirchensteuer ev. u.
    rk.";:HTAB 60:PRINT USING "########";C(62)
7980 FOR I = 1 TO 79:PRINT"=";:NEXT I
7985 PRINT"Gesamt-Einkommen/Lohn-Steuer";:HTAB 60:PRINT
    USING "########";C(63)
7990 HTAB 60:PRINT"========"
7995 VTAB 23:HTAB 23:PRINT"--> ";:INVERSE:PRINT"<W>eiter,
    <Z>urück ... <W/Z>";:NORMAL:PRINT" <--":GET Y$
8000 IF Y$ = "W" OR Y$ = "Z" THEN 8005:ELSE 7995
8005 IF Y$ = "Z" THEN 7820
8010 HOME:PRINT"An- u. Abrechnung : ";:HTAB
    70:INVERSE:PRINT"SEITE 6":NORMAL
8015 FOR I = 1 TO 79:PRINT"-";:NEXT I:PRINT
8020 VTAB 7:PRINT"Lohn/Einkommensteuer Erstattung oder
    Erhebung, sowie Erläuterungen"
8025 FOR I = 1 TO 79:PRINT"*";:NEXT I
8030 IF C(63) = 0 THEN 8045
8035 IF C(63) > 0 THEN PRINT"Restbetrag";:HTAB 60:PRINT
    USING "########";C(63);:PRINT" DM"
8040 IF C(63) < 0 THEN PRINT"Erstattungsbetrag";:HTAB
    60:PRINT USING "########";C(63) * -1;:PRINT" DM"
8045 PRINT:IF E(1) = 1 THEN PRINT"Einzelveranlagung mit
    Witwensplitting Par. 32a VI ESTG.":PRINT
8050 IF A(71) + A(72) + B(72) + A(74) = 0 THEN 8110
8055 PRINT"An Vorsorgeaufwendungen plus Bausparbeiträge
    wurden ";A(71) + A(72) + B(72) + A(76);" DM
    berücksichtigt"
8060 PRINT"Höchstbetrag : ";E(2);" DM"
8065 PRINT"Vorsorgepauschale/Pauschbetrag : ";E(3);" DM"
8070 IF (A(4) + A(5)) > 0 THEN PRINT"Es wurde(n) ";A(4);"
    Kind(er) und ";A(5);" Zahlkind(er) berücksichtigt."
8075 IF (A(79) + A(80) + A(81) + A(82) + B(79) + B(80) +
    B(81)) = 0 THEN 8110
8080 PRINT"In 1984 noch abzugsfähige Verluste:"
8085 PRINT" aus 1979 : 0";:HTAB 30:PRINT"aus 1980 :
    ";E(11)
8090 PRINT" aus 1981 : ";E(12);:HTAB 30:PRINT"aus 1982 :
    ";E(13)
8095 PRINT" aus 1983 : ";E(14);:HTAB 30:PRINT"aus 1983 :
    Verlustrücktrag"
8100 PRINT" aus 1985 : ";E(15);:HTAB 30:PRINT"aus 1987 :
    Verlustvortrag"
8105 PRINT" aus 1986 : ";E(16)
8110 IF C(16) = 0 THEN 8125
8115 PRINT:PRINT"Verlustrücktrag des negativen
    Gesamtbetrags der Einkünfte in 1983 + 1982"
8120 PRINT"in Höhe von ";ABS (C(16));" DM"
8125 IF A(85) > 0 THEN PRINT:PRINT"Von den
    außergewöhnlichen Belastungen n. Par.33 ESTG wurde
    eine zumutbare":PRINT"Eigenleistung I.H.V. ";E(5);"
    DM abgezogen."
8130 IF E(9) > 0 THEN PRINT"Die begünstigten Einkünfte
    wurden zu Gunsten um ";E(9);" DM gekürzt"
8135 VTAB 23:HTAB 23:PRINT"--> ";:INVERSE:PRINT"<M>enue,
    <Z>urück ... <M/Z>";:NORMAL:PRINT" <--":GET Y$
8140 IF Y$ = "M" OR Y$ = "Z" THEN 8145:ELSE 8135
8145 IF Y$ = "Z" THEN 7905
8150 CHAIN "MENUE"
```

Helpmenue :

<A> Anleitung
<B> Kurzerklärungen der Lohn/Einkommensteuer
<C> Abkürzungen

<Z> MENUE – zurück zum Hauptmenue

Bitte wählen Sie < A – D, Z > ———> < >

Anleitung : SEITE 1

Dieses Programm läuft vollständig automatisch ab. Das heißt, Sie brauchen nur bei der Eingabe von Zahlen die entsprechenden Zahlen eintippen und die Return-Taste drücken. Bei Eingaben, die einen Buchstaben erwarten, braucht nur der Buchstabe angetippt werden.

Um Ihnen die Eingaben weiter zu erleichtern, steht rechts neben der geforderten Angabe in Klammern die Dimension, z.B. <DM>. Hier muß also eine Zahl eingetippt werden. Anschließend die Return-Taste drücken.

Angaben, die einen bestimmten Wert erwarten, z.B. die Monatsangabe <MM>, kann nicht falsch eingegeben werden, da dieses Programm die Eingabe dann automatisch überwacht. Ist die Zeile abgeschlossen, so wird der Wert an den Rand geschoben. Wenn Sie keinen Wert zum Eingeben haben, brauchen Sie anstatt der Eingabe "0" nur die RETURN-Taste drücken.

Bitte wählen Sie ... <W>eiter, <M>enue —> < >

Anleitung : SEITE 2

Falls Sie sich trotzdem einmal vetippt haben, die Zeile aber schon abgeschlossen ist, so arbeiten Sie das Programm weiter durch, denn am Ende des Bildschirms erscheint jedesmal die Frage:

.... : sind alle Angaben richtig ? <J/N>

Tippen Sie nun "N" ein, so springt der Cursor auf die erste Eingabezeile zurück. Zur Kontrolle steht rechts der vorher eingegebene Wert. Sie können nun den Wert korrigieren. Alle richtigen Werte müssen allerdings dann erneut eingegeben werden.

Ist ein Programmabschnitt abgeschlossen und Sie kehren zum Hauptmenue zurück, so erscheint ein Stern <*> in der abgearbeiteten Zeile, z.B.

<A> * <—— Grundangaben

Hier haben Sie also eine Kontrolle, welcher Abschnitt schon bearbeitet worden ist. Wollen Sie einen Abschnitt noch einmal bearbeiten, so ist es jederzeit möglich, auch wenn Sie die Abrechnung schon angesehen haben. Diese Variante erlaubt es Ihnen Ihr Geld im Rahmen des Erlaubten richtig einzusetzen.

Bitte wählen Sie ... <W>eiter, <M>enue —> < >

Anleitung : SEITE 3

Um einen neuen Durchlauf zu starten, drücken Sie die Taste <X> im Hauptmenue. Sie gelangen nun zum Passwort-Schutz zurück. Geben erneut das Passwort ein und die Variablen sind auf "0" gesetzt. Aus systemtechnischen Gründen mußte dieser etwas umständliche Weg gewählt werden.

ACHTUNG !!!

Achten Sie auf die Reihenfolge im Hauptmenue. D.h., um einen einwandfreien Durchlauf im Programm zu gewährleisten, ist es erforderlich, daß Sie das Hauptmenue von oben nach unten durcharbeiten. Sie fangen also mit dem Programm "<A> Grundangaben" an und arbeiten sich nach unten durch. Haben Sie z.B.

keine <B> Einkünfte aus Land- u. Forstwirtschaft,
keine <C> Einkünfte aus Gewerbebetrieb,
keine <D> Einkünfte aus selbständiger Arbeit,
aber <E> Einkünfte aus nichtselbständiger Arbeit,

so übergehen Sie die Abschnitte <B>, <C>, und <D> und tippen <E> ein. Dementsprechend fahren Sie weiter von oben nach unten fort.

Bitte wählen Sie ..... <M>enue —> < >

Kurzerklärungsmenue :

Erklärungen aus folgenden Bereichen sind möglich:

<A> Grundangaben
<B> Einkünfte aus der Land- u. Forstwirtschaft
<C> Einkünfte aus Gewerbebetrieb
<D> Einkünfte aus selbständiger Arbeit
<E> Einkünfte aus nichtselbständiger Arbeit
<F> Einkünfte aus Kapitalvermögen
<G> Einkünfte aus Vermietung u. Verpachtung
<H> Sonstige Einkünfte
<I> Doppelbesteuerungsabkommen u. außerordentl. Einkünfte
<J> Sonderausgaben
<K> Außerordentliche Belastungen
<L> Besondere Abzugsbeträge u. ausländische Steuern
<M> Einkünfte zur An- und Abrechnung

<Z> MENUE – zurück zum Helpmenue

Bitte wählen Sie < A – M, Z > ———> < >

Kurzerklärung : Grundangaben SEITE 1

Als Grundangaben benötigt das Programm den Familienstand, das Geburtsdatum, die Religion, die Anzahl der Kinder und Zahlkinder.

Folgende Werte können Sie im Lohn/Eink.-Steuer-Antrag entnehmen:
Familienstand ..... Zeile 9/15
Geburtsdatum ..... Zeile 4/72 u. 11/73
Religion ..... Zeile 4/72 u. 11/73
Anzahl d. Kinder .. Seite 2, Zeile 29 – 34
Zahlkinder ..... Seite 2, Zeile 42 u. 43

Die Angabe des <Familienstandes> dient der Entscheidung, ob die Einkommensteuer oder die Einkommensteuer-Splittingtabelle anzuwenden ist. Sie hat ferner die Bedeutung für die Berechnung der Vorsorgepauschale und der höchstens abzugsfähigen Vorsorgeaufwendungen, ferner für die Berechnung ganz allgemein der Sonderausgaben und der außergewöhnlichen Belastungen.

Bitte wählen Sie ... <W>eiter, <M>enue —> < >

Kurzerklärung : Grundangaben SEITE 2

Aus der Angabe des <Geburtsdatums> kann das Programm erkennen, ob der Altersfreibetrag von 720 DM für 64jährige oder ob der Versorgungs-Freibetrag für 62jährige oder schwerbehinderte 60jährige oder der Altersentlastungsbetrag für 64jährige in Betracht kommt.

Falls der Arbeitnehmer/geber oder sein Ehegatte oder beide einer <Religion> angehören, ist dies unmißverständlich anzugeben.

Die Anzahl der <Kinder> ist für die Berechnung besonders wichtig, da hier etliche Berechnungen auf die Anzahl der Kinder berührt, z.B. Kinderfreibetrag, Prämienspargesetze, Einkommensgrenze usw.

Unter <Zahlkinder> versteht man die Kinder, die dem Elterteil zugeordnet sind, also in der Regel die Kinder, für die der Ehemann Unterhalt zahlen muß.

Bitte wählen Sie ... <W>eiter, <M>enue —> < >

Kurzerklärung : Grundangaben SEITE 3

Das Programm ermöglicht zwei verschiedene Veranlagungsformen:
1. die Einzelveranlagung, d.h., ein lediger, geschiedener oder verwitweter Steuerpflichtiger wird allein veranlagt
2. die Zusammenveranlagung, d.h., die Veranlagung von beiden Ehegatten

Das Programm entscheidet anhand der Eingabe <ledig/verheiratet>.

Bei einer Einzelveranlagung wird zusätzlich abgefragt, ob <Witwensplitting zu beachten ist. In den folgenden Fällen ist dies zu beachten:
1. der Steuerpflichtige ist 1983 verwitwet und hat nicht wieder geheiratet.
   Folge: im Jahr 1984 Einzelveranlagung <ledig> mit Witwensplitting
2. der Steuerpflichtige wurde 1984 geschieden, oder die Ehe wurde anderweitig aufgelöst, und der Ehegatte hat wieder geheiratet und erfüllt die Voraussetzungen für eine Zusammenveranlagung bzw. getrennte Veranlagung.
   Folge: Eine Einzelveranlagung mit Witwensplitting

Bitte wählen Sie ..... <M>enue —> < >

Kurzerklärung : Einkünfte aus Land- u. Forstwirtschaft

Angegeben wird hier der <Gewinn> aus der Land- u. Forstwirtschaft nach Paragraph 13 ESTG. Außerdem ist der <Veräußerungsgewinn> einzugeben, der in diesem Gewinn steckt. Der Freibetrag für die Land- u. Forstwirtschaft wird automatisch ermittelt.

Für den <Ermäßigungsbetrag> nach Par. 34e ESTG wird der <Gewinn 84/85> aus den verschiedenen Land- u. Forstwirtschaftsbetrieben angegeben, bzw. der durch den Feststellungsbescheid festgestellten Höchstbetrag.

Die Angaben für die Einkünfte aus Land- u. Forstwirtschaft können Sie der Anlage L entnehmen, die Sie beim FA erhalten.

Bitte wählen Sie ..... <M>enue —> < >

Kurzerklärung : Einkünfte aus dem Gewerbebetrieb

Anzugeben ist hier der <Gewinn> nach Par. 15 ESTG und außerdem auch der im Gewinn enthaltene <Veräußerungsgewinn>.

Die Angaben für die Einkünfte aus dem Gewerbebetrieb können Sie der der Anlage GSE entnehmen, die Sie beim FA erhalten.

Bitte wählen Sie ..... <M>enue —> < >

Kurzerklärung : Einkünfte aus selbständiger Arbeit

Eingegeben wird der <Gewinn> aus der <freiberuflichen Arbeit> und aus der <sonstigen Tätigkeit>. Diese beiden Unterschiede werden in der Berechnung berücksichtigt, da der Freibetrag ebenfalls unterschiedlich ist.

Weiterhin sollen die <Werbungskosten> sowie der <Veräußerungsgewinn> nach Paragraph 18, Abs. 3 angegeben werden.

Für die <Veräußerungsgewinne> gibt es einen ermäßigten Steuersatz nach Paragraph 34e ESTG. Bei Angabe des <Veräußerungsgewinns> wird in jedem Fall nur der steuerpflichtige Teil eingegeben, d.h., nach Abzug eines Freibetrages.

Die Angaben für die Einkünfte aus selbständiger Arbeit können Sie der Anlage GSE entnehmen, die Sie beim FA erhalten.

Bitte wählen Sie ..... <M>enue —> < >

Kurzerklärung : Einkünfte aus nichtselbständiger Arbeit SEITE 1

Die Werte für Bruttolohn, Bezüge, Werbungskosten und die Beträge von Arbeitslosengeld <ALG>, Schlechtwettergeld <SWG>, Kurzarbeitergeld <KUG> und Arbeitslosenbeihilfe <ALB> werden hier eingegeben.

Folgende Werte können Sie im Lohn/Eink.-Steuer-Antrag entnehmen:

Arbeitslohn	Anlage N, Zeile 3/10 u. 3/11
Versorgungsbezüge .	Anlage N, Zeile 21/32 u. 21/33
Werbungskosten	Anlage N, Seite 2, Zeilen 37 - 63
ALG/SWG/KUG/ALB ...	Anlage N, Zeilen 14 - 17

Der <Bruttoarbeitslohn> ergibt sich aus dem lfd. Lohn aus der aktiven Tätigkeit. <Versorgungsbezüge> sind z.B. Beamtenbezüge aus ehemaliger Tätigkeit.

Für <Werbungskosten> erhält jeder Arbeitnehmer einen Pauschbetrag von 564 DM. pro Jahr. Wird dieser Betrag durch tatsächliche Aufwendungen nicht erreicht, erübrigt sich der Eintrag auf Anlage N. Beim LST-Jahresausgleich muß in der Regel ein einwandfreier Nachweis mit Belegvorlage erbracht werden.

Bitte wählen Sie ... <W>eiter, <M>enue —> < >

Kurzerklärung : Einkünfte aus nichtselbständiger Arbeit SEITE 2

Die <Werbungskosten> beinhalten :

Fahrten Wohnung - Arbeitsstätte	Bewerbungskosten	Fortbildungskosten
Beiträge zu Berufsverbänden	Kilometergeld	Telefongebühren
Aufwendungen für Arbeitsmittel	Arbeitszimmer	Umzugskosten
Mehraufwendg. f. Verpflegung	Berufskrankheit	
Mehraufwendg. f. dopp. Haushalt	Pauschsätze für bestimmte Berufsgruppen	

Ab 1982 unterliegen das <Arbeitslosengeld>, das <Kurzarbeitergeld>, das <Schlechtwettergeld> und die <Arbeitslosenhilfe> dem Progressionsvorbehalt, der ausschließlich durch das FA zu berücksichtigen ist. Die genannten Lohnersatzleistungen bleiben dabei steuerfrei. Auf das zu versteuernde Einkommen wird der höhere Steuersatz angewandt, der sich ergibt, wenn die Lohnersatzleistung mit Ihrem Bruttobetrag, wie Arbeitslohn, in die Einkommensberechnung mit einbezogen wird.

Bitte wählen Sie ..... <M>enue —> < >

Kurzerklärung : Einkünfte aus Kapitalvermögen

Bei den Angaben über Ihr Kapitalvermögen werden die Werte <Einnahmen> und <Werbungskosten> erwartet. Zu den Einnahmen gehören z.B. Sparzinsen und Dividenden. Bei den Dividenden ist zu beachten, daß die zu versteuernden Einnahmen über dem Auszahlungsbetrag liegen. Das bedeutet:

Auszahlungsbetrag + Kapitalertragssteuer + Körperschaftssteuer = Einkommen

Die Angaben für die Einkünfte aus dem Kapitalvermögen können Sie der Anlage KSO entnehmen, die Sie beim FA erhalten.

Bitte wählen Sie ..... <M>enue —> < >

Kurzerklärung : Einkünfte aus Vermietung u. Verpachtung SEITE 1

Die Einkünfte aus Vermietung u. Verpachtung unterteilen sich in zwei Gruppen:
1) Die Werte für <selbstgenutztes Einfamilienhaus>, <selbstgenutzte Eigentumswohnung> und das völlig selbstgenutzte Haus.

Hierbei wird der <Einheitswert>, bzw. Anteil, laut Einheitswertbescheid, der <Wohnzweck> mit seiner Prozentangabe über Nutzung des Hauses oder der Eigentumswohnung, der Angabe der <Selbstnutzung> in Monaten, die <Schuldzinsen> vor der Selbstnutzung (diese sind übrigens voll als Werbungskosten abzugsfähig) und <Schuldzinsen> danach (diese sind nur bis zur Höhe des Grundbetrages abzugsfähig), eingegeben.

Sollte ein Bauantrag nach dem 30.09.82 gestellt worden sein und ist das Gebäude bis zum 01.01.87 fertiggestellt, so können bis zu 10000 DM pro Jahr an Schuldzinsen für drei Jahre abgezogen werden.

Außerdem dürfen Sie die Absetzung für die Abnutzung <AFA> für sämtliche zulässigen Abschreibungen nach Par. 7b IV/V ESTG und Par. 82a - g ESTDV geltend machen.

Bitte wählen Sie ... <W>eiter, <M>enue —> < >

Kurzerklärung : Einkünfte aus Vermietung u. Verpachtung SEITE 2

2) Unter diesen Abschnitt fallen die <Mieteinnahmen> und die <Werbungskosten>

Hierbei sind die <Mieteinnahmen> für vermietete Häuser und die darauf entfallenden <Werbungskosten> einzugeben. Unter <Einnahmen> fällt auch der Nutzungswert der selbstgenutzten Wohnung und die Nutzungsrechte. Zu den <Werbungskosten> gehört wiederum die AFA oder die AFA nach Par. 7b ESTG für ein Haus, bei dem der Bauantrag oder der Baubeginn oder der Erwerbsantrag nach dem 29.07.81 gestellt, bzw. datiert sind.

Die Angaben für die Einkünfte aus Vermietung u. Verpachtung können Sie der Anlage V entnehmen, die Sie beim FA erhalten.

Bitte wählen Sie ..... <M>enue —> < >

Kurzerklärung : Sonstige Einkünfte

Zu den sonstigen Einkünften gehören, die <Einnahmen>, die <Werbungskosten>, <sonstige Einnahmen> und die <Werbungskosten> für die sonstigen Einnahmen.

Bei den <Einnahmen> handelt es sich um Einnahmen nach Par.22 1 ESTG. Das sind Renten und wiederkehrende Bezüge. Die Renten werden mit ihrem Betragsanteil und die Bezüge werden voll geltend gemacht. Der Ertragsanteil ist dem ESTG zu entnehmen. Dazu kommen die Einnahmen nach Par.22 1a, die sich aus Unterhaltsleistungen an den Ehegatten nach Par.10 Abs.1 Nr.1 als Sonderausgaben im gegenseitigen Einverständnis ergeben - maximal 900 DM-. Die <Werbungskosten>, die in diesem Fall entstehen, werden ebenfalls eingegeben.

Bei den <sonstigen Eingaben> handelt es sich um Einnahmen, wenn sie im Par.22 Nr.2 ff ESTG genannt sind, unter anderem Einkünfte aus Spekulationsgeschäften. Auch hier können Sie <Werbungskosten> geltend machen.

Angaben für die sonstigen Einkünfte können Sie der Anlage KSO entnahmen, die beim FA erhalten.

Bitte wählen Sie ..... <M>enue —> < >

Kurzerklärung : Doppelbesteuerung u. außerordentl. Einkünfte

Einkünfte, die unter ein <Doppelbesteuerungsabkommen> fallen und dadurch von der Einkommensteuer befreit werden, sind hier anzugeben. D.h., sie werden zwar nicht besteuert, aber für die Ermittlung des Steuersatzes herangezogen. Die **daran enthaltenen <außerordentlichen Einkünfte> sind gesondert anzugeben.**

Ebenfalls anzugeben sind <Entschädigungen> nach Par.24 Nr.1 ESTG und <Nutzungsvergütungen> nach Par.24 Nr.3 ESTG <Entschädigungen> und <Nutzungsvergütungen müssen jedoch bereits in einer der sieben Einkunftsarten berücksichtigt sein, z.B. im Gewinn aus dem Gewerbebetrieb.

Daraus ergibt sich folgende Aufstellung:

Veräußerungsgewinn + Entschädigung + Nutzungsvergütung = außerordentl. Eink.

Angaben finden Sie in der Anlage N, Zeile 18/36 u. 39

Bitte wählen Sie ..... **<M>enue —>** < >

Kurzerklärung : Sonderausgaben SEITE 1

Als Sonderausgaben bezeichnet das ESTG bestimmte Kosten der Lebenshaltung, die förderungswürdig erscheinen. Dazu gehören z.B., <Kirchensteuer>, <Steuerberatungskosten>, <Aufwendungen> für die <Berufsausbildung>, <Spenden>, Beiträge an <Bausparkassen>.

Zu den <Vorsorgeaufwendungen> zählen abzugsfähige Versicherungen, wie Kranken-, Unfall-, Haftpflicht- und Lebensversicherungen. <Bausparbeiträge> sind nur dann einzugeben, wenn keine Sparprämie beantragt wurde. <Spenden an politische Parteien> können nur bei Vorlage der Spendenbescheinigung geltend gemacht werden. Dasselbe gilt bei den <sonstigen Spenden>. Hierzu gehören Spenden an wissenschaftliche, staatspolitische und als besonders förderungswürdige anerkannte Institutionen. Zu den <unbeschränkten Sonderausgaben> gehören die Kirchensteuern, die Steuerberatungskosten u. die Ausbildung für die Berufsausbildung.

Bitte wählen Sie ... <W>eiter, <M>enue —> < >

Kurzerklärung : Sonderausgaben SEITE 2

Zu den Sonderausgaben gehört weiterhin der <Arbeitnehmer/geber-Anteil> und der <Verlustausgleich>.

Der <Arbeitnehmeranteil> beinhaltet den Arbeitnehmer-Anteil zur gesetzlichen Rentenversicherung. Der <Arbeitgeberanteil> dementsprechend den Arbeitgeberanteil zur ges. Rentenversicherung. Der <Verlustausgleich> beinhaltet den negativen Gesamtbetrag der Einkünfte des jeweiligen Jahres.

Angaben können Sie dem EST-Antrag auf Seite 3 entnehmen. Hier sind die Zeilen 62 - 85 einzusehen.

Bitte wählen Sie ..... <M>enue —> < >

Kurzerklärung : Außergewöhnliche Belastungen

Außergewöhnliche Belastungen sind: <Aufwendungen> n. Par.33, Pakete, Päckchen in die DDR, Besuch aus der DDR, <Unterhaltszahlungen> an Personen ohne Kindergeldanspruch und <Ausbildungsfreibeträge>.

Unter <Aufwendungen n. Par. 33> fallen: Krankheitskosten -soweit nicht erstattet-, Ehescheidungskosten, Beerdigungskosten, Kurkosten, Heim- u. Pflegekosten usw.

Für die <Unterhaltszahlungen> an Personen sollten Sie folgendes beachten: Höhe der Unterhaltsaufwendunden sind tatsächlich gezahlte Beiträge. Unterhaltszahlungen Dritter sind geleistete Zahlungen von Dritten an dieselbe Person. Unter Einkünfte und Bezüge versteht man die Einkünfte und Bezüge nach Steuerrecht, d.h., Einkünfte = Einnahmen - Werbungskosten - Freibeträge/Pauschalen.
Bezüge = Einnahmen - Unkostenpauschale (360 DM)

Für den <Ausbildungsfreibetrag gilt dasselbe wie vorgenannt.

Die Angaben für die außergewöhnlichen Belastungen können Sie dem LST-Antrag **aus den Zeilen 86 - 94 entnehmen.**

Bitte wählen Sie ..... <M>enue —> < >

Kurzerklärung : Besondere Ermäßigungen

Zu den besonderen Abzugsbeträgen gehört die Anrechnung ausländischer Steuerbeträge, die auf ausländische Einkünfte entfällt. Bei Staaten ohne Doppelbesteuerungsabkommen gilt Par. 34c ESTG>. Weiterhin gehört die Ermäßigung für Darlehen zum Berlin-Förderungsgesetz, Ermäßigung für vermögenswirksame Leistungen des Arbeitgebers, sowie die Ermäßigung für die Schaffung von Ausbildungsplätzen.

Die Angaben können Sie der Anlage N, Zeile 23 - 30 entnehmen.

Bitte wählen Sie ..... **<M>enue —>** < >

Kurzerklärung : An- u. Abrechnung

Einzugeben sind die Eintragungen der <Lohnsteuer>, <Kirchensteuer>, <Kapitalertragssteuer> und <Körperschaftssteuer> laut Dividendenbescheinigung.

Die Angaben der Lohn/Kirchen-Steuer kann der Anlage N, Zeile 1 - 5 entnommen werden

Bitte wählen Sie ..... **<M>enue —>** < >

Abkürzungen :

Die wichtigsten Abkürzungen in alphabetischer Reihenfolge:

Abs.	Absatz
AFA	Absetzung für Abnutzung
ALB	Arbeitslosenbeihilfe
ALG	Arbeitslosengeld
BAFOEG	Bundesausbildungsföderungsgesetz
Betr.	Betrieb
ESTG	Einkommensteuergesetz
EST	Einkommensteuer
FA	Finanzamt
FG	Förderungsgesetz
KUG	Kurzarbeitergeld
Par.	Paragraph
SWG	Schlechtwettergeld
v.T.	von Tausend

Bitte wählen Sie ..... <M>enue —> < >

Fortsetzung von Seite 44

CHORZ	$057B	Cursor horizontal
CVERT	$05FB	Cursor vertikal
BYTE	$067B	I/O-Byte für PASCAL
START	$06FB	Bildschirmanfang
POFF	$077B	Kaltstart, GOTOXY, Ctrl-Z
FLAGS	$07FB	Video-Schalter u. -Flaggen

BYTE, POFF und FLAGS sollen uns in diesem Zusammenhang nicht weiter interessieren. Fassen wir außerdem den Bildschirmaufbau grafisch zusammen, wobei wir vereinfachend annehmen, daß sich der Bildanfang bei $CC00 in Bank 0 befindet (siehe **Diagramm 1**).

Wir betrachten also aus dem 2K-RAM immer nur ein großes Fenster, das jetzt beliebig herauf- und heruntergeschoben werden kann, denn intern schließt sich an die Bank 3 wieder die Bank 0 an. Wird für uns der Bildschirm z.B. um eine Zeile nach *oben* geschoben, verändert die Videx-Karte den tatsächlichen Speicherinhalt nicht. Lediglich das Fenster wandert eine Zeile nach *unten*, indem die Zeiger neu gesetzt werden. Das geht bedeutend schneller, als fast 2K um jeweils 80 Bytes zu verschieben!

Das Auslesen des Bildschirms

Nachdem wir nun das Prinzip des Bildschirmaufbaus verstanden haben, können wir beginnen, ihn auszulesen und abzuspeichern. Eigentlich funktioniert das genauso wie beim 40-Zeichen-Schirm, nur daß wir hier eine andere Art der Positionsberechnung durchführen müssen. Der Apple-Monitor hat leider keine Routine bereit, aber wir können einen Teil des Videoterm-ROMs adaptieren. Wir benötigen folgende Informationen:
1) Start des Bildes
2) Start der jeweiligen Zeile
3) RAM-Bank
4) Seite $CC00 oder $CD00.
Wenn wir den Cursor in die erste Zeile stellen, sind Zeilenanfang und Bildstart identisch, so daß die Punkte 1) und 2) zu einer Aufgabe verschmelzen. Ctrl-Y bewirkt dieses „HOME CURSOR", ohne den Bildschirm zu löschen. BASEL und BASEH liefern dann die Adresse, aus der wir nur noch RAM-Bank und Speicherseite extrahieren müssen. Dies geschieht durch „Bitschieben" und ist erst auf den zweiten Blick durchsichtig.
Der Zeilenbeginn liegt irgendwo zwischen $0000 und $07FF. Betrachten wir hiervon nur das höherwertige Byte, so gibt es acht Möglichkeiten, die folgende Tabelle zusammenfaßt:

```
037B F0 06       78            BEQ ENDE
037D 20 ED FD    79            JSR COUT
0380 E8          80            INX
0381 D0 F5       81            BNE CLOSE
0383 20 22 FC    82  ENDE      JSR VTAB            ;normalisiere Cursor
0386 20 93 FE    83            JSR SETVID
0389 4C D0 03    84            JMP DOS             ;Warmstart
038C             85  *
038C 8D 84       86  TEXT      HEX 8D84
038E CF D0 C5    87            ASC "OPENSCHIRMTEXT"
0391 CE D3 C3
0394 C8 C9 D2
0397 CD D4 C5
039A D8 D4
039C 8D 84       88            HEX 8D84
039E D7 D2 C9    89            ASC "WRITESCHIRMTEXT"
03A1 D4 C5 D3
03A4 C3 C8 C9
03A7 D2 CD D4
03AA C5 D8 D4
03AD 8D 00       90            HEX 8D00
03AF 8D 84       91  TEXT1     HEX 8D84
03B1 C3 CC CF    92            ASC "CLOSE"
03B4 D3 C5
03B6 8D 00       93            HEX 8D00
00B8             94  ZEND      EQU *-START


VIDEXT

BSAVE VIDEXT, A$0300, L$00AD
Initialisieren mit BLOAD VIDEXT
Aufrufen mit CALL 768

                  1  * VIDEXT
                  2  *
                  3  * Videx 80 Z/Z -> Diskette
                  4  *
FDED              5  COUT      EQU $FDED
03D0              6  DOS       EQU $3D0
03EA              7  DOSHOOK   EQU $3EA
00D9              8  RUNMODE   EPZ $D9
0033              9  PROMPT    EPZ $33
0075             10  CURLIN    EPZ $75
AAB6             11  LANGUA    EQU $AAB6
CFFF             12  COROM     EQU $CFFF
C300             13  INIT      EQU $C300
FF58             14  RETURN    EQU $FF58
AA53             15  DOUT      EQU $AA53
                 16  *
                 17  *
0300             18            ORG $300
0300             19            OBJ $800
0300             20  *
0300             21  ** Videx in Slot 3 !! **
0300             22  *
047B             23  BASEL     EQU $47B            ;LL * Adresse im Bild-
04FB             24  BASEH     EQU $4FB            ;HH * Schirmspeicher
C0B0             25  DEV0      EQU $C0B0           ;Bank Auswahl
CC00             26  DISP0     EQU $CC00           ;1. Speicherseite
CD00             27  DISP1     EQU $CD00           ;2. Speicherseite
0300             28  *
0300             29  *
0300             30  *
0300 20 EA 03    31  START     JSR DOSHOOK         ;DOS Anhängen
0303 8D FF CF    32            STA COROM           ;Koresidente ROMs abschalten
0306 8D 00 C3    33            STA INIT            ;Videx-ROM einschalten
0309 A9 99       34            LDA #$99            ;Ctrl-Y
030B 20 ED FD    35            JSR COUT            ;HOME Cursor
030E A9 80       36            LDA #$80            ;Bit 7 setzen
0310 85 33       37            STA PROMPT          ;ein laufendes
0312 85 76       38            STA CURLIN+1        ;Applesoftprogramm
0314 85 D9       39            STA RUNMODE         ;vortäuschen
0316 A9 58       40            LDA #RETURN         ;direkte Bildschirm-
0318 8D 53 AA    41            STA DOUT            ;ausgabe unterdrücken
031B A9 FF       42            LDA /RETURN
031D 8D 54 AA    43            STA DOUT+1
0320 A2 00       44            LDX #0
0322 BD 83 03    45  DISK      LDA TEXT,X
0325 F0 06       46            BEQ DRUCK
0327 20 ED FD    47            JSR COUT
032A E8          48            INX
032B D0 F5       49            BNE DISK
032D             50  *
032D A9 18       51  DRUCK     LDA #24             ;Zeilenzahl
032F A2 50       52  VIDEX1    LDX #80             ;Zeichenzahl
```

High-Byte	Dualzahl	Seite	Bank
00	0000	$CC00	0
01	0001	$CD00	0
02	0010	$CC00	1
03	0011	$CD00	1
04	0100	$CC00	2
05	0101	$CD00	2
06	0110	$CC00	3
07	0111	$CD00	3

Schauen wir uns die Dualzahl genauer an, so stellen wir fest, daß immer eine 0 in Bit 0 und Seite $CC00 bzw. eine 1 in Bit 0 und Seite $CD00 zusammenfallen. Um das niederwertige Bit zu testen, schieben wir es mit der LSR-Instruktion ins Carry-Bit, das mit PHP „gerettet" wird. Es dient in Zeile 65 des Listings zur Auswahl der Seite.
Bleibt die Frage nach der Bank. Unsere Dualzahl hatte vor dem Schieben Einsen in den Bits 0, 1 und 2. Nach dem Rechtsschieben befinden sie sich nur noch in den Bits 0 und 1. Mit dem Befehl AND #$03 blenden wir den Rest des Bytes (Bit 7-2) weg. Jetzt wird zweimal nach links geschoben und hinten jeweils eine 0 angefügt (= ASL). Dadurch erhalten wir für die High-Bytes $00 und $01 eine $00, für $02 und $03 eine $04, für $04 und $05 eine $08 und schließlich für $06 und $07 eine $0C. Für High-Byte = $06 noch einmal die Einzelschritte:

$06 = 0110 Carry ?
LSR → 0011 Carry 0 → Seite $CC00
ASL ← 0110
ASL ← 1100 = $0C

Den erhaltenen Wert tauschen wir ins Y-Register und sprechen damit indiziert den richtigen Bank-Schalter an.
Unser Ausleseprogramm ist noch in eine innere Schleife für 80 Zeichen und eine äußere Schleife für 24 Zeilen eingebettet. Die Diskettenbefehle entsprechen dem Programm SCHIRMDISK, nur wurde eine Variante zur Vortäuschung eines laufenden BASIC-Programms verwendet. VIDEXT (= VIDex zu tEXTfile) wurde für den Speicherbereich ab $0300 assembliert und kann mit
BSAVE VIDEXT, A$0300, L$00AD
abgespeichert werden. Von BASIC aus wird es mit einem **CALL 768** gestartet und legt dann einen Textfile namens „VIDEXTEXT" auf Diskette ab. Ein Programmstart mittels Reset wurde nicht realisiert, weil dabei einige Register in Unordnung geraten.

```
0331 48          53            PHA                 ;Retten
0332             54  ** Routine verÄndert nach Videx-ROM **
0332 AD FB 04    55  VIDEX2    LDA BASEH           ;Hi-Byte-Position
0335 4A          56            LSR
0336 08          57            PHP
0337 29 03       58            AND #3              ;%0000 0011
0339 0A          59            ASL
033A 0A          60            ASL
033B A8          61            TAY
033C 99 B0 C0    62            STA DEV0,Y          ;Softswitch für Bank
033F AC 7B 04    63            LDY BASEL           ;Low-Byte-Position
0342 28          64            PLP
0343 B0 05       65            BCS VIDEX3
0345 B9 00 CC    66            LDA DISP0,Y         ;1. Seite
0348 90 03       67            BCC VIDEX4
034A B9 00 CD    68  VIDEX3    LDA DISP1,Y         ;2. Seite
034D EE 7B 04    69  VIDEX4    INC BASEL           ;erhöhe Low-Bytes
0350 D0 03       70            BNE VIDEX5          ;Überlauf?
0352 EE FB 04    71            INC BASEH           ;dann auch Hi-Byte
0355             72  ** Ende verändertes Videx-ROM **
0355 20 ED FD    73  VIDEX5    JSR COUT            ;Zeichen ausgeben
0358 CA          74            DEX
0359 D0 D7       75            BNE VIDEX2          ;bis Zeile fertig ist
035B 68          76            PLA                 ;hole Zeilenzähler zurück
035C 38          77            SEC
035D E9 01       78            SBC #1              ;nächste Zeile
035F D0 CE       79            BNE VIDEX1          ;bis alle fertig sind
0361             80  *
0361 A2 00       81            LDX #0
0363 BD A4 03    82  CLOSE     LDA TEXT1,X
0366 F0 06       83            BEQ ENDE
0368 20 ED FD    84            JSR COUT
036B E8          85            INX
036C D0 F5       86            BNE CLOSE
036E 20 00 C3    87  ENDE      JSR $C300           ;Bildschirm ankoppeln
0371 4C D0 03    88            JMP DOS             ;Warmstart
0374             89  *
0374 B1 B9 B8    90            ASC "1984 DR. KEHREL"
0377 B4 A0 C4
037A D2 AE A0
037D CB C5 C8
0380 D2 C5 CC
0383             91  *
0383 8D 84       92  TEXT      HEX 8D84
0385 CF D0 C5    93            ASC "OPENVIDEXTEXT"
0388 CE D6 C9
038B C4 C5 D8
038E D4 C5 D8
0391 D4
0392 8D 84       94            HEX 8D84
0394 D7 D2 C9    95            ASC "WRITEVIDEXTEXT"
0397 D4 C5 D6
039A C9 C4 C5
039D D8 D4 C5
03A0 D8 D4
03A2 8D 00       96            HEX 8D00
03A4             97  *
03A4 8D 84       98  TEXT1     HEX 8D84
03A6 C3 CC CF    99            ASC "CLOSE"
03A9 D3 C5
03AB 8D 00      100            HEX 8D00
03AD            101  *
00AD            102  ZEND      EQU *-START
```

Hinweis: Diese Programm wurde mit dem LISA-Assembler erstellt.

Programmierzeit auf die Hälfte gekürzt!
Speicherbedarf auf ein Drittel reduziert!

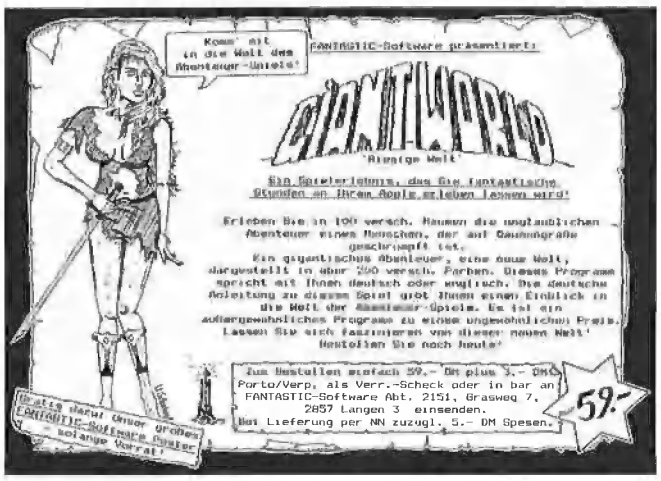
EXBASIC® LEVEL II™

Mit EXBASIC LEVEL II können Sie Ihren Computer um ein Vielfaches einfacher und effizienter programmieren. Sie haben über 75 neue, äußerst leistungsfähige Funktionen, die den Erfordernissen moderner Software-Erstellung entsprechen. Für Computer von

apple® commodore®

EXBASIC LEVEL II ist bereits weltweit im Einsatz. Schreiben Sie sofort „Schicken Sie kostenlos ausführliche Informationen zu EXBASIC LEVEL II für..... (genauen Computertyp angeben!)"

INTERFACE AGE
Verlag GmbH
Josefsburgstraße 6
8000 München 80
Tel.: 089/43 40 89



ELECTRONIC ARTS ORIGIN SYSTEMS DATASOFT
PENGUIN SIERRA ON-LINE STRATEGIC SIMULATIONS

In Amerika schon Standard...
hier groß im Kommen:

Mockingboard

Sprach- und Tongenerator für Apple II, //e
6-stimmiger Stereoton mit den neuen
Programmen der großen Softwarehäuser:
*Music Construction Set *One On One
*Skyfox *Ultima III *Zaxxon u.v.m.

Appleland Walliser & Co.
Mönchseestr. 99 7100 Heilbronn Tel.07131/60048

WS 2000 WORLD STANDARD MODEM

Die neue Version dieses weltweit benutzten professionellen Modems – immer noch zum unschlagbaren Preis von DM 798,–!

- ☆ Datenaustausch und Kommunkation mit praktisch jedem Computer weltweit möglich
- ☆ Zugriff zu Datenbanken, Mailboxen, Btx, Btx rückwärts usw.
- ☆ Telex für alle durch einen neuen Dienst – mit Ihrem Computer und dem WS 2000 (fragen Sie uns)
- ☆ Alle gängigen Baudraten (75, 300/300, 600, 1200, 1200/75, 75/1200) und international üblichen Übertragung-Standards (CCITT, BELL) – umschaltbar per Hand oder per Computer (IC-Satz SK1 hierfür DM 96,90; Anschlußkabel UPL DM 48,–)
- ☆ Automatisches Wählen mit Zusatzplatine AD2 (DM 199,50) und Kabel UPL
- ☆ Automatisches Annehmen von Anrufen mit Zusatzplatine AA2 (DM 199,50)
- ☆ Einfacher Anschluß (parallel zur Telefonleitung); eingebautes Netzteil; deutsche Anleitung; 1 Jahr Garantie
- ☆ Viele Interfaces lieferbar; z. B.
  CBM I für C64/VIC20 einschl. Listing DM 136,80
  AC Kommunikations-Karte für APPLE DM 330,60
  SPEC für SPECTRUM einschl. Software (auf ROM) DM 256,50
- ☆ Anschlußkabel zwischen Computer und Modem (bitte benötigten Steckertyp angeben) DM 57,–
- ☆ Liefermöglichkeit: sofort ab Lager Hamburg
- ☆ Alle Preise einschließlich MwSt. zuzüglich Verpackung, Porto und Nachnahme (Bei Vorauszahlung durch V-Scheck/Überweisung Porto und Verpackung frei)

Claus F. Erbrecht
Computer Related Products
Lappenbergsallee 37 · 2000 Hamburg 20
Telefon 040/850 52 55
Bankverbindung: Bank für Gemeinwirtschaft
BLZ 200 101 11, Konto-Nr. 1 241 223 700

Achtung: Nur für hausinterne Telefon-Anlagen und nicht amtsberechtigte Nebenstellen – in der BRD ist der Anschluß an das öffentliche Telefonnetz nicht gestattet!

Pascal-Directory unter der Lupe

von Dieter Geiß

Viele Berichte und Aufsätze befassen sich mit dem Aufbau des Inhaltsverzeichnisses (Directory) auf pascalformatierten Disketten. In den meisten Darstellungen werden jedoch nicht alle Komponenten des Directory erfaßt, in keinem Artikel wirklich alle Komponenten genau erklärt. Deshalb bringen wir nunmehr eine umfassende Analyse des Directory, die auch als theoretische Basis für das GETPAS-Programm dient.

Jede Diskette, die unter einem UCSD-Pascal-Betriebssystem für Apple-Computer formatiert wurde, ist in logische Blöcke unterteilt. Ein Block beherbergt 512 Bytes, und auf einer normalen 140K-Diskette haben 280 Blöcke Platz. Auf Block 0 und 1 steht das Maschinenprogramm für den Boot-Vorgang. Ab Block 2-5 folgt das Directory, das genau 4 Blöcke umfaßt. Die letzten 20 Bytes bleiben unbenutzt. Ab Block 6 folgt dann entweder ein zweites Directory, das sog. „Duplicate-Directory", oder der freie Diskettenplatz zum Abspeichern der Files (siehe **Abb. 1**).

Das „Duplicate-Directory" wird vom Pascalsystem wie das Original-Directory geführt, und sollte letzteres einmal defekt sein, was nicht selten vorkommt, so kann man mittels eines kleinen Programms (s. **COPYDUPDIR**) das „Duplicate-Directory" auf das Original-Directory überspielen. Voraussetzung ist allerdings, daß das „Duplicate-Directory" angelegt wurde. Dies kann man im FILER mit Hilfe des Zero-Kommandos bewerkstelligen. Für Datendisketten, auf denen sich viele Pascal-Programme befinden, ist das durchaus anzuraten.

In Listing 1 ist die vollständige Definition des Directory enthalten

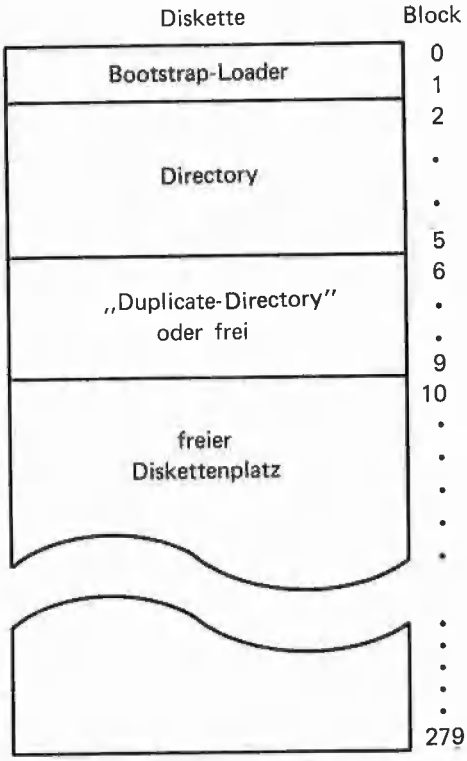


Diagramm 1

Die Konstanten

Maxdir (Maximum Number of Directory Entries) gibt die maximale Anzahl der Einträge im Directory an. Der erste Eintrag ist mit Null indiziert und definiert die Art des Directory. Dann kommen die Angaben für insgesamt 77 Files. Mehr Files können niemals im Directory stehen, auch wenn man die Anzahl der Directory-Blöcke künstlich erhöht.

In der Anleitung, die zu den Teac-Laufwerken mitgeliefert wird, ist zu lesen, daß die Kapazität des Directory erhöht wurde, da die Laufwerke bis zu 1280 Blöcke pro Diskette ansprechen können. Da alle Systemkomponenten wie SYSTEM.PASCAL, SYSTEM.FILER etc. immer nur 4 Blöcke einlesen können, kann die Modifikation, die die Teac-Techniker ansprechen, niemals funktionieren, es sei denn, daß das Betriebssytem selbst völlig umgeschrieben wurde.

Vidleng (Volume Identification Length) begrenzt die Anzahl der Buchstaben in einem „Volume-Namen" auf 7.

Tidleng (Title Identification Length) gibt an, aus wievielen Buchstaben sich ein File-Name maximal zusammensetzt. Diese 15 Buchstaben dürfen keine Leerzeichen oder Ctrl-Zeichen enthalten.

Fblksize (File Block Size) bedeutet die Anzahl der Bytes pro Diskettenblock (Block Size = Blockgröße).

Dirblk (Directory Block) definiert die Stelle auf einer Diskette, auf der das Directory steht (erster Block).

Dupdirblk (Duplicate Directory Block) zeigt auf die Stelle des „Duplicate-Directory" (erster Block).

Die Typen

Daterec (Date Record) ist die Definition des Datums im Pascal-Betriebssystem. Ein Datum ist, da es in einem gepackten Record steht, immer 16 Bit groß. Beim Jahr muß man sich ein 19.. hinzudenken, was einen Bereich von 1900 – 1999 abdeckt. Steht im Directory-Eintrag eines Files das Jahr des dazugehörigen Datums auf 100, so ist dieser File nicht-permanent. Einen solchen File kann man in einem Programm mittels des Rewrite-Befehls erzeugen. Schließt man den File nicht mit der Lock-Option im Close-Befehl, sondern nur mit der Normal-Option, dann wird der File nicht im Directory gespeichert. Normalerweise kann man diesen File dann nicht mehr sehen, wenn man ein L(ist der Diskette ausgibt. Ein normaler Close-Befehl wird vom Compiler immer in den Code eingefügt, wenn irgendwelche Files benutzt werden. Deswegen dürfte eigentlich der Fall eines falschen Jahrs im Datum niemals auftreten.
Manchmal steigt der Compiler aber auch wegen zu geringen Speicherplatzes mit Stack Overflow aus. Dann kann dieses Datum nicht mehr verbessert werden, weil das System neu gebootet werden muß. Man kann dann diesen File auch nicht mehr im FILER mit R(emove ‚File-Name' löschen, weil der File nicht mehr gefunden werden kann. Mit einem R(emove und XXX=, falls der Name z. B. XXX lautet, erreicht man dann aber doch ein Löschen des Files. Manchmal wird ein solcher File auch vom System gelöscht; darauf kann man sich aber leider nicht verlassen.
Dirrange (Directory Range) ist der zulässige Bereich für Indizierungen in das Directory.
Vid (Volume Identification) definiert den zu Vidleng gehörigen String, während
Tid (Title Identification) den zu Tidleng gehörigen String definiert.
Filekind ist ein Aufzählungstyp, der alle Arten von Files benennt, die im Directory vorkommen können. Dazu gehören:
Untypedfile: Von diesem Typ ist der Directory-Eintrag selbst und steht zuerst in Block 2; entspricht der Indizierung des Directory-Arrays mit dem Index 0.
Xdskfile (Examined Disk File) ist Bad File mit defekten Blöcken, der mit dem FILER-Kommando X(amine markiert werden kann. Daher ergibt sich die seltsame Abkürzung für ein markiertes d(i)skfile.

Codefile, Textfile, Infofile, Datafile, Graffile, Fotofile sind als File-Typen erzeugbar, wenn eine Datei mit dem entsprechenden Suffix .CODE, .TEXT usw. geöffnet wird. Ob ein Codefile dann tatsächlich Code, ein Graffile tatsächlich ein Grafikbild enthält, ist eine Sache der Interpretation. Auch eine zufällige Anordnung von Bits kann man als – wenn auch nicht gerade ästhetische – Grafik ansehen.

Securedir (Secure Directory) ist in keinem Handbuch dokumentiert, was aber nicht bedeutet, daß es hier nicht erklärt werden kann. Untersucht man nämlich den Pcode des Pascalsystems genauer, so entdeckt man in Segment 5 (GETCMD) und Segment 0 (PASCALSYSTEM), wann ein Securedir abgefragt wird. Um zu verstehen, was dort vor sich geht, muß man sich auch den Informations-File SYSTEM.MISCINFO genauer ansehen. Der File SYSTEM.MISCINFO wird beim Booten des Systems und bei dessen Initialisierung eingelesen. Darin befinden sich die Informationen über die Konfiguration des Systems wie Höhe und Breite des Bildschirms, ob eine Uhr angeschlossen ist usw. In diesem File befindet sich aber auch eine Variable namens ‚Userkind'. Diese Variable ist vom Aufzählungstyp und kann folgende Werte annehmen: Normal, Aquiz, Booker und Pquiz. Beim regulären System wird diese Variable auf Normal initialisiert, und da diese Variable nicht ohne einen Trick im File SYSTEM.MISCINFO verändert werden kann, bleibt in ihr auch der Wert erhalten. Das Pascalsystem versucht bei Zugriff auf ein Directory zu erkennen, ob dieses gültig ist. Ob dies der Fall ist, hängt von der Art des Directory (Untypedfile, Securedir) sowie vom Userkind ab. Ein Directory ist genau dann ein gültiges Pascal-Directory, wenn
a) der Dfirstblk (s.u.) gleich Null und der Userkind gleich ‚Booker' ist,
b) der Typ des Directory gleich Securedir und der Userkind entweder ‚Aquiz' oder ‚Pquiz' ist (Die ausgeschriebenen amerikanischen Ausdrücke für Aquiz und Pquiz konnten noch nicht ermittelt werden.),
c) der Typ des Directory gleich Untypedfile und der Userkind gleich ‚Normal' ist.
Fall c) ist der Normalfall.
Nachdem die Einzelfälle abgetestet wurden, versucht das System zu erkennen, ob die Directory-Einträge in Ordnung sind. Dazu wird die Länge der Dvid und die Zahl der Files auf der Diskette (Dnumfiles) überprüft. Da der Dfirstblk auf allen Disketten immer auf Null gesetzt ist, kann man, wenn man sich im Modus ‚Booker' befin-

INPUT 2.0

Ein Bildschirm-Maskengenerator für DOS 3.3 und ProDOS
von U. Stiehl

1984, Diskette und Manual, DM 98,–
ISBN 3-7785-1021-5

Der für den Apple II bestimmte Maskengenerator „Input 2.0" basiert auf den früheren Programmen „Input 1.0" und „Input 80 1.0" (von denen noch Restbestände lieferbar sind) und ist sowohl unter DOS 3.3 wie auch unter dem neuen ProDOS lauffähig. Der Maskengenerator setzt einen Apple II Plus mit Language Card oder einen Apple IIe voraus. Im 40 Z/Z-Modus funktioniert er auf beiden Gerätetypen, im 80 Z/Z-Modus dagegen nur auf dem Apple IIe mit 80-Zeichen-Karte. (Die alte Videx-Karte für den Apple II wird nicht unterstützt!)

„Input 2.0" liegt wahlweise in der Bank 1 oder Bank 2 der Language Card und wird durch einen kurzen Driver in den unteren 48K aufgerufen. „Input 2.0" läßt sich problemlos in nicht-compilierte und compilierte Applesoft- sowie in Assemblerprogramme einbinden. Die Übergabe der Feldinhalte an das Anwenderprogramm erfolgt durch ein einfaches Verfahren, das auch bei Compilern funktioniert.

Für jedes Feld der Bildschirmmaske lassen sich u. a. definieren: Feldlänge (bis zu 255 Zeichen) – Vtab – Htab – Datentyp (insgesamt 8 Typen) – Scrollflag (starre oder dynamische Maske) – Ctrlflag – Füllflag – Löschflag – Bildschirmflag (40- oder 80 Zeichendarstellung). Innerhalb eines Eingabefeldes besteht jeder denkbare Redigierkomfort (Insert, Delete, Rubout, Restore usw.).

Bei der neuen Version des Maskengenerators können jetzt auch Ctrl-Zeichen beim Datentyp String eingegeben werden. Ferner sind – das gilt nur für IIe – die Apfeltasten als schnelle Cursortasten definiert.

Gerätevoraussetzung: Apple IIe oder IIc; ferner Apple II+ im 40-Zeichenmodus

Hüthig Software Service,
Postfach 10 28 69,
D-6900 Heidelberg

det, die Directories aller Disketten lesen, ob Securedir oder Untypedfile. Steht im Userkind der Wert für ‚Normal', so kann man zwar normale, nicht jedoch Secure-Directories lesen. Dies erklärt auch den Namen des „sicheren Directory". Sie sind vor jedem normalen Benutzer sicher. Im Userkind ‚Aquiz' und ‚Pquiz' kann nur ein Secure-Directory gelesen werden, keines vom Typ ‚Untypedfile'. Während man im ‚Pquiz'-Modus normal arbeiten kann, also editieren, compilieren usw., geschieht im ‚Aquiz'-Modus folgendes:

Nach dem Booten der (Secure-Directory-) Diskette wird wie beim normalen Booten zuerst der File SYSTEM.ATTACH gestartet, mit dem man Disketten- oder Druckertreiber anschließen kann. Danach wird wie gewöhnlich SYSTEM.STARTUP geladen und ausgeführt. Das entspricht beim DOS dem sogenannten HELLO-Programm, beim ProDOS dem STARTUP-Programm. Nach dessen Ablauf geht das System aber nicht in die Kommandozeile, sondern sucht den File SYSTEM.WRK.CODE, um ihn zu starten. Nach dessen Ablauf bootet das System neu. Ist nur der SYSTEM.WRK.TEXT allein vorhanden, wird er zuerst compiliert, bevor er ausgeführt werden kann. Ohne SYSTEM.WRK.TEXT wird ein noch zu spezifizierender File compiliert und ausgeführt. Das Secure-Directory ist also dazu da, um im Userkind ‚Aquiz' das Programm im File SYSTEM.WRK.CODE auszuführen und dann wieder neu zu starten. Da kein normaler Benutzer das Directory lesen kann, kann auch keiner ohne weiteres die einzelnen Files kopieren. Die ganze Diskette hingegen kann ganz normal im FILER oder z. B. mit COPYA kopiert werden. Selbst kann man solche Disketten allerdings nicht herstellen, da man sonst im Initialisierungssegment des Pascalsystems etwas ändern müßte. Selbst wenn man mit einem Disk-Read-Write-Programm das Directory auf Securedir und auch im SYSTEM.MISCINFO den Userkind auf ‚Aquiz' verstellt, so kann das System die Diskette nicht lesen, denn der Userkind wird beim Booten auf ‚Normal' gestellt, bevor SYSTEM.MISCINFO eingelesen werden kann. Man müßte also entweder SYSTEM.APPLE oder SYSTEM.PASCAL patchen. So scheint das Secure-Directory nur für System-Programmierer gedacht zu sein.

Man sollte im übrigen Secure-Directories nicht mit Duplicate-Directories verwechseln, denn beide haben nichts miteinander zu tun.

Subsvol (Subsidiary Volumes) sind Unterdirectories, ähnlich wie beim ProDOS, aber sie sind erst in Pascal IV.1 implementiert und nur der Vollständigkeit halber im Filekind mit aufgenommen, da Pascal-IV.1-Disketten den gleichen Aufbau des Directory haben. Ein Subsidiary Volume sieht nach außen aus wie ein File. In ihm befindet sich aber in den ersten sechs Blöcken wieder ein vollständiges Directory, das dann über weitere Unitnummern angesprochen werden kann.

Direntry (Directory Entry) ist der eigentliche Eintrag im Directory. In diesem Record wird unterschieden, ob es sich um Directory-Informationen oder um File-Informationen handeln soll. Gleich sind allerdings die beiden ersten Variablen des Records. Zum folgenden s. **Abb. 2a** und **Abb. 2b.**

Dfirstblk (First Block of Disk File) zeigt auf den ersten Block des Files. Ist der Eintrag ein Directory, also vom Typ Untypedfile oder Securedir, dann enthält Dfirstblk immer die Zahl 0, obwohl sich die erste gültige Information auf Block 2 befindet. Dies ist die einzige Ausnahme. Bei allen anderen Einträgen enthält Dfirstblk immer den korrekten Wert für den Anfang des Files.

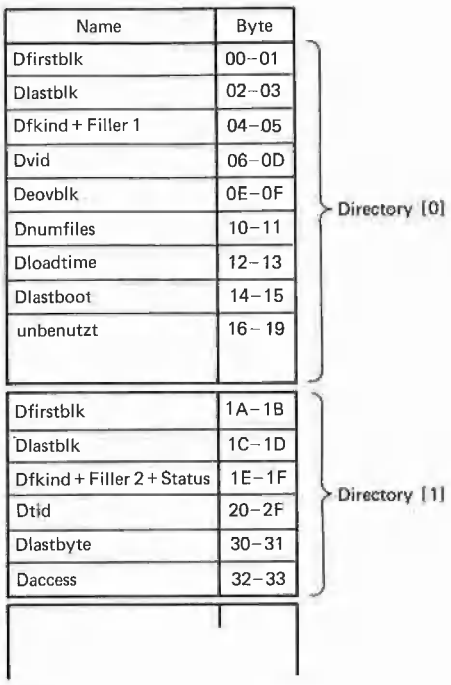


Diagramm 2a: Directory [0] und [1]

Dlastblk (Last Block of Disk File) zeigt nicht etwa auf den letzten Block des Files, sondern auf den Block, der diesem folgt. Die Variable müßte eigentlich Dnextblk heißen. Dlastblk zeigt beim Eintrag des Directory selbst auf Block 6 oder 10, in Abhängigkeit davon, ob ein „Duplicate-Directory" angelegt wurde oder nicht.

Dfkind (Disk File Kind) unterscheidet nun, um welchen Typ es sich bei dem jeweiligen File handelt. Im Falle eines Directory, also des Typs Untypedfile oder Securedir, verzweigt die Definition in den ersten Ast mit seinen Variablen.

Filler1 ist ein 13-Bit-Füller für Aufwärtskompatibilität.

Dvid (Disk Volume Identification) enthält den Namen der Diskette und darf sieben Buchstaben nicht überschreiten.

Deovblk (Disk End of Volume Block) enthält die Anzahl der Blöcke, die auf einer Diskette vorhanden sind. Diese Zahl muß also geändert werden, wenn man größere Laufwerke anschließen will.

```
00  00 00 06 00 00 00 05 42 4C 41 4E 4B  .......BLANK
0C  00 00 18 01 01 00 00 00 E8 A0 00 00  ........H ..
18  00 00 06 00 26 00 05 00 00 53 59 53  ....&....SYS
24  54 45 4D 2E 41 50 50 4C 45 00 90 B2  TEM.APPLE..2
30  00 02 9C A7
```

Diagramm 2b: Directory [0] und [1] als Sektor-Dump

Dnumfiles (Number of Files in Directory) ist die aktuelle Anzahl von Files, die sich auf der Diskette befinden.

Dloadtime (Disk Load Time) gibt die Zeit an, wann das Directory zum letzten Mal gelesen oder geschrieben wurde. Nur wenn eine Uhr im Rechner eingebaut ist, wird diese Zahl benutzt, um festzustellen, ob ein Directory aktualisiert werden muß.

Dlastboot (Last Boot of Disk) schließlich ist das Datum, das man mit Hilfe des D(ate-Kommandos im FILER setzen kann und das beim Booten in das System-Datum übernommen wird.

Ist die Art des Eintrags ein File, so verzweigt die Definition in den anderen Ast.

Filler2 ist ein 12-Bit-Füller für Aufwärtskompatibilität.

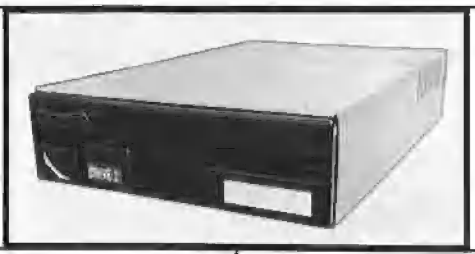
Status wird vom FILER benutzt, um die Files zu markieren, die mit einem „Wildcard" angesprochen wurden.

Dtid (Disk Title Identification) ist der Name

DISTAR für APPLE-II

★ APPLE-II kompatibel
★ halbspurfähig
★ 40 Spuren, 163 KB
★ Spur-0-Erkennung
★ Direktantrieb
★ Stahlband-Positionierung
★ Kabel für DISK-II-Controller
★ kann DOS 3.3, Pascal, CP/M

II, II + IIe 498,– IIc 527,–



DISTAR für IBM u.ä.

★ IBM PC kompatibel
★ double sided, DD, 48 TPI
★ Track to Track 6 ms
★ MSDOS 1.1.: 320 kB/2.1: 360 kB
★ Direktantrieb
★ Stahlbandpositionierung

539,–

Wir sind DISTAR-Distributor

APPLE-II kompatibel
COMPUTER

fert. aufgebauti. Gehäuse, 8 Slots, verst. Schaltnetzt., Zeichnungen. deutsch od. ASCII, Ziffernbl., incl. Handbuch, ohne Firmware

PC-48: 48K, 6502 Proz. **999,–**
PC-64: 64K, 6502 Proz. **1099,–**
PC-64z: 6502 + Z80 Proz. **1199,–**
PC-64zE: wie PC-64z, Geh. IBM-look, abgesetzte deutsche Tastatur **1399,–**

Aufpreis Monitor grün entsp. 279,–
Aufpreis Monitor bernst. entsp. 314,–
Aufpreis Disk (slimline) + Geh. + Contr 585,–
IBM-look-Tastatur m. **Apple-Adapter** 349,–
Joysticks 49,– bis 69,–

Drives 5,25" f. **APPLE II** Contr: SIEMENS, Spur-0-Sensor, **II+, IIe** 599,–

Doppeldiskstation 1,2 MByte **2399,–**
2 80-Spur-Laufwerke (2x640K) m. Spezial-Controller, eigenes Gehäuse u. Netzteil

SENTINEL-Disketten m. Verstärk.-Ring double density 10/50/100 49,–/230,–/450,–

VIDEX ULTRATERM
max. **4096** Z/Screen 1170,–
80 Zeich. verbessert. Darstellung 184,–
ext. **Softswitch** f. 80-Zeich.-Karte 39,90
80 Zeich. dto. Softsw. **integriert** 269,–
16K-, Z80-, Disk-Controller, par. Interf. (graf.) f. NEC m. Kabel **129,–**
RS232C async. (wie Apple high speed) 149,–
Super-Serial-Interface 298,–
Grappler + Print. Interf. m. Kabel 169,–
Grappler + incl. 64K Buffer 599,–
128k Karte incl. Pseudodisk-Softw. 499,–
Parallel Karte (2x6522) 149,–
Experimentier-Karte 39,–
Firmware in EPROMs 57,–
dto. mit BASIC-Lehrbuch (Sybex) 87,–

EPROM-Burner plus + mit Nullkraft-TEXTOOL-Sockel einschl. Software f. 2704/8/58/16/32, 2516/32 u. 2764, 27128, 27256, zeitoptimiert jetzt mit Schnellprogramm.-Modus Zusatz zur Progrmg. des 8748/49/55 373,–
218,–

16-Bit Computer **IBM-PC** kompatibel:
IBM-look Gehäuse, 128K, 2 Drives DS. 4780,–
Nt. 100 W, deutsche Tastatur, Monochrom-Karte, Motherboard mit 1 par. + 2 ser. + Disk-Interf., auf **1MByte** ausbaubar

Monitor spez. f. **IBM-TTL**-Anschluß **519,–**
dto. **amber** anstatt grünem Bildsch. **539,–**

Baugruppen:
Motherboard wie oben 2053,–
Monochrom-Display Karte 390,–
Color Graphics Karte 529,–
Netzteil 100W 434,–
Netzteil 130W 559,–
512KByte RAM-Karte (OKRAM) **328,–**
Gehäuse f. Computer IBm-look 279,–
Harddisk 10MByte mit Host-Adapt 4990,–
Drive 5,25" 48tpi DS, DD 539,–
Tastatur US ASCII 399,–
Tastatur deutsch 429,–

SENTINEL-Disketten ds, dd 10 Stck. **70,–**

Matrixdrucker KDC FT-5002 **1198,–**
ASCII- + **IBM-Mode** wählbar, 1K Buffer, 40 eig. Zeich. per Softw. definierbar **120 Z/s,** 8 Schriftarten (mischbar), 11 internat. Zeich.-sätze, hoch- + tiefstellen f. math. Anw., **Traktor + Friktion** qualit. hochwertig. **Schönschriftmodus**

Typenraddrucker KDC WP-550 **1498,–**
Typrad m. 100 Zeichen, 14 Z/s, Traktor + Friktion, ser. + **Centr. par.**-Eingänge seriem., Pap.-Breite 15", **WORDSTAR**-kompatibel, Option: Einzelblatteinzug

4-Farb-Plotter KDC FPL-2000 **2190,–**
200 mm/s, 0,1 mm Genauigkeit, 0,05 mm Schritt, Farben per Spftw. anwählbar, **ser. + Centr. par.**-Eingang serienm., 500 Byte Buffer, **HP-7470-kompatibel** Papierbreite ca. 300 mm (**DIN A 3**)

Preise incl. MwSt., 6 Monate Garantie. Lieferung per NN. Ausland Vorkasse. Neue ausführliche Info gegen DM 1,40 in Briefmarken.

Ingenieurbüro Dipl.-Ing. R. Springmann
Stöckener Straße 199 · 3000 Hannover 21

TOMBSTONE-MICRO

Th. Tank & G. Körber

MESON II, 48 K	1100,– DM
MESON II, 64 K (16 K + Z 80-Card)	1250,– DM
beide Rechner mit 10er-Block, 1 Jahr Garantie	
Z 80-Card	125,– DM
16 K-Card	125,– DM
Floppy-Disk 5¼"	480,– DM
Controller-Card	150,– DM
PAL-Card + Modulator	170,– DM
MESON II, Leerplatine	85,– DM
Tastaturen und Gehäuse auf Anfrage	

Joyport, zum Anschluß von zwei ATARI-Joysticks	40,– DM
ADD2 B, (VIA 6522) mit RAM + Backup	170,– DM
ADD2 L, Leerplatine	55,– DM
ADD4 B, (PIA 6821) mit RAM + Backup	150,– DM
ADD4 L, Leerplatine	45,– DM
PIA-Card mit Wrap-Feld ausgerüstet	

Entwicklungen auf Anfrage

Gardeschützenweg 72, 1000 Berlin 45
☎ 030/8 33 13 03 (SHOP), Q 7 46 57 28 (BÜRO)

apple® & CP/M®

SOFTWARE HARDWARE LITERATUR

SpeeDemon von M·c·T

DM 1368.-

Machen Sie Ihren APPLE II+/e bis 3,5 x schneller !!
mit CPU 65C02C - 64 K RAM - belegt einen Slot -
Nur ein Beispiel unserer Leistungsfähigkeit! Verlangen Sie Qualität - wir bieten sie - Ihr Import-Spezialist!

Karl-Heinz Weiß
Am Wiesenhof 17 · 2940 Wilhelmshaven
☎ 04421/83179

Katalog gegen eine Schutzgebühr von DM 3,- in Bfm.

Telefon (02 41) 3 49 62
Noppiusstraße 19, 5100 Aachen

RÖCKRATH MICROCOMPUTER

Das deutsche Apple-II-ROM-Listing ist da!

- Einleitung zum prinzipiellen Ablauf des Applesoftinterpreters:
  - Aufbau und Verarbeitung der/des
    - Programmtextes – Variablentabelle – Stringspace
    - Fließkommaformate – Basicstacks (GDSUB, FOR-NEXT, ...)
  - Beschreibung der wichtigsten Unterprogramme, z. B. Variablensuche, Garbage collection, Ausdrucksauswertung, CHRGET, ...
- Vollständig disassemblierte und sehr ausführlich deutsch kommentierte Auflistung des Applesoft-BASIC-Interpreters
- Übersichtliche Auflistung aller vom Interpreter benutzten RAM-Zellen mit allen Verwendungszwecken
- Über 150 ausführlich dokumentierte Unterprogramme:
  - Funktion – Ein-/Ausgabeparameter

Auch für Apple-IIe und c und Kompatible!

Außerdem folgende ROM-Listings vorrätig:

TRS-80 Model I, Genie I+II	**69,50 DM**
TRS-80 Model III	**79,– DM**
Colour-Genie	**59,– DM**

Preis inkl. MwSt./Händleranfragen erwünscht!

M. Buck: Apple-II-ROM-Listing (116 S. DIN A4, kartoniert. 59 DM, Best.-Nr.: 06-010-8)

des Files, der eine Länge von maximal 15 Buchstaben haben kann.
Dlastbyte (Last Byte of Disk File) gibt an, welches das letzte Byte im letzten Block ist, das beim Lesen benutzt werden soll. Das System erkennt an dieser Variablen, wann ein File zu Ende ist, wenn es sich bereits im letzten Block befindet. Nur bei Data-Files, z. B. einem File of Integer, wird diese Zahl benutzt. Text- oder Codefiles haben immer eine 512 in Dlastblk.
Daccess (Disk Access) ist das Datum, das immer gesetzt wird, wenn der File modifiziert worden ist. Dies wird intern durch den Close-Befehl bewirkt. Das Jahr im Datum wird – wie schon oben erwähnt – benutzt, um nicht-permanente Files zu markieren.

Directory ist nun das Array von Directory-Einträgen, also das die gesamten 4 Blocks umfassende Directory. Das Element, das mit 0 indiziert wird, also Directory [0], ist vom Typ Untypedfile oder Securedir und enthält die Information für das Directory selbst. Die anderen Einträge enthalten die Information für die Files.

Mit der Directory-Definition in einem Programm kann nun ein Benutzer Diskettenin-formationen verwerten und auch modifizieren. In dem Beispielprogramm (COPY-DUPDIR) wird das „Duplicate-Directory" einer Diskette auf das normale Directory überspielt, wenn erkannt wird, ob dieses defekt ist. Das Programm fragt nach der Unitnummer, die eine der Zahlen 4, 5, 9, 10, 11 oder 12 sein kann. Dann wird das Directory gelesen. Ist das Original-Directory in Ordnung, wird das Programm verlassen. Andernfalls wird es auf das Original-Directory übertragen. Wenn auf der Diskette die Blöcke 2–6 schlecht sind, müssen diese zuerst repariert werden, was z. B. im FILER mit dem X(amine-Kommando gemacht werden kann. Nach Ablauf des Programms ist das Directory der Diskette wiederhergestellt.

GETPAS: Konvertierung von Pascal- in DOS-Textfiles

von Ulrich Stiehl

Unsere „Peeker" wird komplett auf dem Apple IIe erfaßt, wobei die Druckerei kodierte DOS-Textfile-Disketten zur Übertragung in die Lichtsatzanlage erhält. Zu diesem Zweck wurde ein superschnelles Transmitterprogramm entwickelt, das mit einer Übertragungsrate von bis zu 1000 Bytes/Sekunde (einschließlich des Einlesens der Dateien von der Apple-Diskette!) die Daten online über eine V.24-Schnittstelle an den Großrechner der Satzanlage schickt. Neben dem Transmitterprogramm mußten mehrere Spezialprogramme zur Vorkodierung und zur Konvertierung in DOS-Textfile-Format geschrieben werden. In diesem Peeker-Heft sowie in den folgenden Ausgaben werden wir einige der interessanteren Spezialprogramme vorstellen.

GETPAS

Das nachstehend gelistete Programm GETPAS zur Konvertierung von Pascal- in DOS-Textfiles stammte ursprünglich von Dana J. Schwartz vom Washington Apple Pie. Es hatte jedoch einen Bug, der bei größeren Dateien zu einer Zerstörung des DOS-Catalogs führte, und außerdem war es für unsere Zwecke viel zu langsam. Deshalb wurde es gestrafft, völlig umgeschrieben und durch ein Assemblerprogramm für alle zeitkritischen Routinen erweitert, so daß es jetzt eine Datenübertragungsrate von bis zu 480 Bytes/Sekunde (bei Diversi-DOS und NOMON) für Einlesen *und* Speichern vorweisen kann.

Konvertierungsvorgang

Aus Gründen der Geschwindigkeit und der Speicherkapazität ist das GETPAS-Programm „gekruncht" und damit weitgehend undurchschaubar. Deshalb geben wir hier nur einige generelle Anmerkungen:
1. GETPAS erwartet die DOS-Diskette in S6, D1 und die Pascal-Diskette in S6, D2, ist also nur für 2-Drive-Besitzer gedacht. Nach RUN GETPAS muß man lediglich den Pascal-Textfile-Namen eingeben. Das ist alles. Ist der Dateiname unbekannt, so kann man sich durch Eingabe von Return allein zunächst das Pascal-Directory ansehen.
2. Die DATA-Statements poken in den Bereich $1800-$18FF ein Maschinenprogramm, das die RWTS, die Tab-Konvertierung usw. umfaßt. Der Quellcode hierzu ist auf der Peeker-Sammeldiskette enthalten.
3. Der Speicherbereich $1900-$20FF nimmt zunächst die 4 Pascal-Directory-Blocks 2–5 auf, aus denen der Name sowie die erste und letzte Blocknummer der Datei extrahiert werden. $1900-$20FF wird damit nur vorübergehend benötigt.
4. Danach wird – beginnend mit dem zweiten Dateiblock – der gesamte Pascal-Textfile in 2-Block-Schüben in den Bereich $1900-$1CFF eingelesen und von dort in den eigentlichen DOS-Textfile-Puffer $1D00-$99FF (dies sind maximal 32000 Bytes) übertragen. Dabei wird die pascaltypische Tabulatorkodierung $10 $XX (= Tab $XX), wobei $XX – $20 die Anzahl der Tab-Spaces impliziert, in echte Leertasten konvertiert. Eventuelle Ctrl-Zeichen werden eliminiert.
5. Schließlich wird der konvertierte Textfile aus dem Puffer auf die DOS-Diskette geschrieben.

Konvertierungsfehler

Zu den häufigeren Fehlern gehören:
1. Es sind nicht zwei Laufwerke an Slot 6 angeschlossen.
2. Die zu übertragende Pascaldatei ist kein Textfile.
3. Der erzeugte DOS-Textfile ist größer als 32000 Bytes.

Abschließend sei darauf hingewiesen, daß HUFFIN mit jedem DOS läuft (DOS 3.3, Diversi-DOS usw.), jedoch *nicht* für ProDOS gedacht ist. Eine ProDOS-Spezialversion wird zu einem späteren Zeitpunkt publiziert. Diese ist übrigens erheblich unkomplizierter, da Pascal und ProDOS beide eine gleichartige Diskettenblockorganisation haben.

BUCH-SHOP

Apple DOS 3.3
von Ulrich Stiehl
2. Aufl. 1984, 203 S., kart.,
DM 28,–

Dies ist die erste deutschsprachige Darstellung des Diskettenbetriebssystems DOS 3.3 für den Apple II/II Plus/IIe, die sich sowohl an Applesoft- als auch an Assembler-Programmierer wendet. Sinngemäß ist das Buch zweigeteilt:
Der erste Teil behandelt ausführlich die dem Applesoft-Programmierer zur Verfügung stehenden DOS-Befehle, wobei die Textfiles wegen ihrer großen Bedeutung und der vergleichsweise komplizierten Handhabung besonders dargestellt werden. Viele Textfile-Tricks werden hier zum erstenmal geschildert.
Aber auch im zweiten Teil findet der reine Applesoft-Programmierer insbesondere in dem Kapitel „Vermischte Tips, Tricks und Patches" zahlreiche Anregungen. Im übrigen ist der zweite Teil für Assembler-Programmierer gedacht. Neben einer detaillierten Beschreibung der DOS-Interna enthält dieser Teil elf vollständige RWTS-Anwenderprogramme – z. B. CPM-Refiner, DOS-lose Datendisk, TSL-Maker, File-Reader, Pseudo-Disk-Driver und Fastbrun-Routine –, die Techniken enthüllen, die bislang noch niemals publiziert worden sind. Dieses DOS-Buch ist deshalb der unentbehrliche Begleiter für jeden Apple-Programmierer.

Apple II Basic Handbuch
von Douglas Hergert
304 Seiten, 116 Abb.
DM 32,–

Das Buch ist als Nachschlagewerk konzipiert, daß seinen Platz neben jedem APPLE II, II+ und IIe haben sollte. Es richtet sich an Anfänger und fortgeschrittene Programmierer.

Aus der Praxis heraus präsentiert der Autor Tips und Vorschläge, die das Programmieren leichter und zugleich effizienter machen. Alle Applesoft- und Integer-BASIC-Begriffe sind alphabetisch aufgelistet und werden eingehend erklärt.

Dazu werden alle DOS-Befehle (neben vielen Begriffen der Computerterminologie) vorgestellt.
Beispielprogramme zeigen dem Nutzer, wie jeder Befehl funktioniert und helfen, die richtige Anwendung zu üben. Unter anderem lernt der Leser den besten Weg, um FOR/NEXT-Schleifen und IF/THEN-Entscheidungen für seine Zwecke einzusetzen.
Durch die präzise und leicht verständliche Sprache des Autors werden auch schwierige Befehle einfach in der Anwendung.

Apple Maschinensprache
von Don und Kurt Inman
1984, 208 S., zahlr. Abb. und Tabellen, DM 49,–

Dieses Buch ist wahrscheinlich die beste Einführung in die 6502-Programmierung für denjenigen Assembler-Anfänger, der zuvor noch nie ein Maschinenprogramm geschrieben hat.
Aus dem Inhalt:
Applesoft II BASIC – kurzgefaßt – Alles über Zeichen – Alles über Speicher – Alles über Maschinenbefehle – Maschinenprogramme mit BASIC eingeben – Graphik – Text – Ton – Arithmetik – Was tun mit den Maschinenprogrammen

Apple II leicht gemacht
von Joseph Kascmer
1984, 185 S., zahlr. Abb., kart.,
DM 28,–

Dies ist ein Buch, wie es sich jeder Apple-Anfänger nur wünschen kann: Schrittweise, leichtverständliche Anleitung zum Umgang mit dem Apple mit einigen durchsichtigen, unkomplizierten Beispielen in Applesoft, die ihn nicht Abschrekken, sondern ermutigen sollen, sich mit dem Gerät näher vertraut zu machen. Damit ist „Apple II leicht gemacht" das ideale Einsteigerbuch für den reinen Anwender, der nicht nur „auf den Knopf drücken", sondern zumindest einige Details aus der Black Box namens Apple erfahren will.

Aus dem Inhalt:
Kontrolle des Geräts – Schreiben und Zeichnen auf dem Bildschirm – Geheimnisvolle Abläufe: Programme – Verschiedene Eingriffsmöglichkeiten – Mobile Speicher: Disketten – Kontrollmöglichkeiten – Das Innenleben

Apple Assembler
Tips und Tricks
von Ulrich Stiehl
1984, 226 S., 3 Abb., kart.,
DM 34,–

„Apple Assembler" wendet sich an alle, die bereits Anfängerkenntnisse der 6502-Programmierung haben – z. B. aufgrund des Buches „Apple Maschinensprache" – und nunmehr ein Nachschlagewerk für ihren Apple II Plus/IIe/IIc suchen, in dem alle wichtigen ROM-Routinen sowie eine Vielzahl sonstiger Hilfsprogramme in einer systematischen Form zusammengestellt werden. Insgesamt umfaßt dieses Buch über 40 Utilities, darunter mehrere völlig neuartige Programme wie Double-Lores, Double Hires, Screen-Format u. a.
Der erste Teil enthält ein Repetitorium der wichtigsten Befehle, Adressierungsarten und sonstigen Besonderheiten des 6502.
Im zweiten Teil werden alle Adressen des Monitors zusammengestellt, die für Assembler-Programmierer von Nutzen sein können. Darüber hinaus findet der Leser Unterroutinen für hexadezimale Addition/Subtraktion/Multiplikation/Division, Binär-Hex-ASCII-Umwandlung usw.
Der dritte Teil befaßt sich mit der Speicherverwaltung der Language Card und der IIe-64K-Karte und enthält Move-Programme zum Verschieben von Daten in die und aus der Language Card sowie der 64K-Karte.

Der vierte Teil ist dem Applesoft-ROM gewidmet und listet eine große Anzahl nützlicher Interpreter-Adressen. Bei den Utility-Programmen liegt das Schwergewicht auf Fließkommamathematik einschließlich Print Using.
Der letzte Teil behandelt den Text- und Graphikspeicher. Neben einem professionellen Maskengeneratorprogramm werden auch Routinen zur Double-Lores- und Double-Hires-Grafik vorgestellt.

Arbeiten mit dem Macintosh
von N. Hesselmann
416 Seiten, 320 Abb. DM 54,–

Das Buch erklärt den Umgang mit dem Macintosh von Grund auf, wobei auch auf elementare Dinge eingegangen wird, wie z. B. die Benutzung der Tastatur und der Maus, das Einlegen von Disketten und den Systemstart. Ganz besonderes Augenmerk wird auf die Erklärung der speziellen Software-Umgebung des Macintosh gelegt, wobei das Menü- und Fensterkonzept sowie das Anwählen durch Piktogramme gekennzeichneter Funktionen klar dargestellt wird.

Der Umgang mit den Programmen MacPaint und MacWrite wird erläutert; dies geschieht teilweise anhand von Beispielen, die leicht nachvollzogen werden können. Ein umfangreiches Kapitel ist dem für den Macintosh erhältlichen Microsoft-BASIC gewidmet.

BASIC Übungen für den Apple
von J. P. Lamoitier
1983, 252 S., zahlr. Abb., kart.,
DM 38,–

Das Buch ist konzipiert, allen Apple-Anwendern Applesoft-BASIC durch praktische Übungen an Hand von reellen Programmen beizubringen. Datenverarbeitung, Statistik, kommerzielle Programme, Spiele und vieles mehr. Jede Übung beinhaltet eine Beschreibung der Problemstellung, eine Analyse der Lösungsmöglichkeiten, ein Flußdiagramm und ein fertiges Programm samt Probelauf.

Aus dem Inhalt:
Ihr erstes BASIC-Programm – Flußdiagramme – Übungen mit Integerzahlen – Elementare Beispiele aus der Geometrie – Allgemeine Übungen aus der Datenverarbeitung – Mathematische Berechnungen – Kaufmännische Berechnungen – Spiele – Operations Research – Statistik

Apple ProDOS für Aufsteiger
Band 1
von Ulrich Stiehl
1984, 202 S., kart., DM 28,–

ProDOS ist das neue „professionelle DOS" (Professional Disk Operating System) für den Apple IIe sowie den mit einer Language Card ausgestatteten Apple II Plus. Band 1 befaßt sich mit den theoretischen Grundlagen von ProDOS, der internen und externen Speicherorganisation und enthält grundlegende Beispielprogramme für Assembler-Programmierer sowie generelle Untersuchungen zum BASIC-SYSTEM. Da ProDOS über erheblich vielfältigere und leistungsfähigere, zugleich jedoch erheblich kompliziertere Dateistrukturen verfügt, sind theoretische Kenntnisse von ProDOS unabdingbar, wenn man die Features von ProDOS voll ausschöpfen will.
Aus dem Inhalt:
Ein erster Überblick – ProDOS und DOS 3.3 – Interne Speicherorganisation – Externe Speicherorganisation – MLI (Machine Language Interface) – ProDOS für Applesoft-Programmierer

Bestellkarte am Heftende

```
10  PRINT  CHR$ (4)"PR#3": PRINT : INVERSE : PRINT "GETPAS:
    DOS,D1 <--- PASCAL,D2": NORMAL : PRINT  CHR$ (4)"UNLOCK
    GETPAS,S6,D1"
11  DATA 169,24,160,18,32,217,3,176,1,96
12  DATA 173,31,24,141,17,24,96,0,1,96
13  DATA 2,0,0,0,36,24,0,0,0,0
14  DATA 1,0,0,96,1,234,0,1,239,21
15  DATA 76,49,24,76,86,24,76,210,24,169
16  DATA 0,133,206,169,29,133,207,169,0,168
17  DATA 145,206,200,208,251,230,207,166,207,224
18  DATA 154,144,243,169,0,141,84,24,169,29
19  DATA 141,85,24,96,0,0,169,0,133,254
20  DATA 169,25,133,255,173,84,24,133,206,173
21  DATA 85,24,133,207,160,0,177,254,201,0
22  DATA 240,87,201,13,240,20,201,32,176,16
23  DATA 201,16,208,17,160,1,177,254,160,0
24  DATA 201,33,144,13,176,20,145,206,32,171
25  DATA 24,32,185,24,76,104,24,32,185,24
26  DATA 32,185,24,76,104,24,170,169,32,145
27  DATA 206,32,171,24,202,224,32,208,244,240
28  DATA 232,230,206,208,9,230,207,165,207,201
29  DATA 154,144,1,0,96,230,254,208,20,230
30  DATA 255,165,255,201,29,144,12,104,104,165
31  DATA 206,141,84,24,165,207,141,85,24,96
32  DATA 169,0,133,206,169,29,133,207,160,0
33  DATA 177,206,240,16,9,128,32,237,253,200
34  DATA 208,244,230,207,165,207,201,154,144,236,96
35  RESTORE
36  FOR X = 6144 TO 6384: READ Y: POKE X,Y: NEXT
37  PRINT  CHR$ (4)"MAXFILES1"
38  HIMEM: 6144: CLEAR :A = 6144:B = 6400:C = A:D = A + 17:
    E = A + 18:F = E + 4:G = E + 5:H = E + 9
39  PRINT : INPUT "TEXTFILE <CR = DIR>:";A$: IF LEN (A$) = 0
    THEN  GOSUB 58: GOTO 39
40  GOSUB 46: IF K <  > 3 OR L - M < 4 THEN  PRINT  CHR$
    (7);"TEXTFILE NICHT GEFUNDEN": GOTO 39
41  CALL 6184:N =  INT (B / 256):O = N + 1:P = O + 1:Q = P +
    1:R = M + 2: GOSUB 52
42  CALL 6184 + 3:R = R + 2: IF R = L THEN 44
43  GOSUB 52: GOTO 42
```

```
44  HOME : PRINT  CHR$ (4)"UNLOCK GETPAS,D1": PRINT  CHR$
    (4)"MON O": PRINT  CHR$ (4);"OPEN"A$: PRINT  CHR$
    (4);"WRITE"A$: CALL 6184 + 6: PRINT : PRINT  CHR$
    (4);"CLOSE"
45  PRINT  CHR$ (4)"NOMON O": PRINT  CHR$ (4)"MAXFILES3":
    END
46  S =  INT (B / 256):T = O: FOR U = 11 TO 4 STEP  - 1:
    GOSUB 56:S = S + 1: NEXT U:V =  PEEK (B + 16):W = B +
    32:X =  LEN (A$)
47  IF  PEEK (W) <  > X THEN 50
48  FOR J = 1 TO X: IF  PEEK (W + J) <  >  ASC ( MID$
    (A$,J,1)) THEN 50
49  NEXT J:M =  PEEK (W - 6) +  PEEK (W - 5) * 256:L =  PEEK
    (W - 4) +  PEEK (W - 3) * 256:K =  PEEK (W - 2): RETURN
50  W = W + 26:V = V - 1: IF V > 0 THEN 47
51  K =  - 1: RETURN
52  Y = R: GOSUB 53:S = N:U = Z: GOSUB 56:S = O:U = A1: GOSUB
    56:Y = R + 1: GOSUB 53:S = P:U = Z: GOSUB 56:S = Q:U =
    A1: GOSUB 56: RETURN
53  T =  INT (Y / 8):B1 = (Y / 8 - T) * 8:A1 = 2 * (7 - B1):Z
    = A1 + 1: IF B1 = 0 THEN Z = 0
54  IF B1 = 7 THEN A1 = 15
55  RETURN
56  POKE F,T: POKE G,U: POKE H,S: POKE D,0: CALL C: IF  PEEK
    (D) = 0 THEN  RETURN
57  TEXT : PRINT  CHR$ (7);"RWTS-FEHLER "; PEEK (D): POP :
    POP : END
58  S =  INT (B / 256):T = O: FOR U = 11 TO 4 STEP  - 1:
    GOSUB 56:S = S + 1: NEXT U:B$ = "":C1 = B + 6: FOR I = 1
    TO  PEEK (C1):B$ = B$ +  CHR$ ( PEEK (C1 + I)): NEXT I
59  PRINT B$;":":C$ = "":X = 1:D1 =  PEEK (B + 16): IF D1 =
    0 THEN  INPUT "<LEER> ";D$: RETURN
60  FOR I = 1 TO D1:E1 = B + I * 26 + 6:C1 =  PEEK (E1): IF
    C1 = 0 THEN  GOTO 62
61  FOR J = 1 TO C1:C$ = C$ +  CHR$ ( PEEK (E1 + J)): NEXT
    J: PRINT " ";C$:C$ = "":X = X + 1
62  IF X > 20 OR I = D1 THEN  PRINT : INPUT "";D$:X = 1
63  NEXT I: RETURN
64  PRINT "Urversion von D.J.Schwartz 1981, durch
    Maschinenprogramm getrimmt von U.Stiehl 1984"
```

GETDOS: Konvertierung von DOS- in Pascal-Textfiles

von Jürgen Geiß

Mit Hilfe des Programms GETDOS lassen sich DOS- in Pascal-Textfiles konvertieren. Dabei kann eine Textdatei maximal 93 Sektoren (23.552 Bytes) umfassen. GETDOS wurde in Pascal geschrieben und wird deshalb von Pascal aus gestartet. Es ist das Gegenstück zu GETPAS, mit dem sich Pascal- in DOS-Textdateien umwandeln lassen.

Nach dem Start des Programms erwartet GETDOS die Eingabe einer Source-Unit (= DOS-Diskette). Als Source-Unit sind alle gültigen Pascal-Unit-Nummern von 4-12 erlaubt, z. B. #5 für DOS-Diskette in Drive 2. Als Destination-Units können neben der Pascal-Diskette (z. B. #4 für Pascal-Diskette in Drive 1) noch zusätzlich die Unit-Nummern 1-3 angegeben werden. Damit ist es möglich, den DOS-Textfile direkt auf den Bildschirm (Unit #1) auszugeben. Meistens wird aber das Schreiben auf eine Datei bevorzugt. Nach dem Kopieren kann dann der Text z. B. mit dem UCSD-Editor weiterverarbeitet werden. Man beachte, daß man den Dateinamen des zu konvertierenden DOS-Textfiles bereits kennen muß, da GETDOS im Gegensatz zu GETPAS nicht den DOS-Catalog anzeigen kann.

Eine Anwendung für das Konvertieren einer DOS-Textdatei könnte etwa das Entwickeln eines Assemblerprogramms für Pascal sein. Das Assemblerprogramm kann dann unter DOS mittels Single Step (Einzelschrittbetrieb) getestet werden. Nach dem Konvertieren kann die Textdatei mit dem UCSD-Editor in UCSD-Assemblersyntax umgeschrieben werden. Da es in Pascal keine guten Debugging-Möglichkeiten für Assemblerprogramme gibt, ist dies fast die einzige Alternative, um längere Assemblerprogramme fehlerfrei in Pascal zu implementieren.

Anmerkung der Redaktion: GETDOS dient ferner dem Zweck, die auf den Peeker-Sammeldisketten als DOS-Textfiles abgespeicherten Pascal-Programme in das Pascal-Betriebssystem zu übernehmen. Wegen der geschweiften Klammern usw. verwenden wir bei den DOS-konvertierten Pascal-Files auf der Sammeldiskette nur den amerikanischen Zeichensatz. In den abgedruckten Listings kodieren wir jedoch aus ästhetischen Gründen meist die Umlaute und Eszett in den Menüanweisungen. Es gilt folgende Zuordnung für deutsche und amerikanische ASCII-Sonderzeichen:

$5B = Ä = [
$5C = Ö = \
$5D = Ü =]
$7B = ä = {
$7C = ö = |
$7D = ü = }
$7E = ß = ~

```
program Getdos (input, output);

{************************************}
{                                    }
{ Dieses Programm konvertiert DOS 3.3 }
{                                    }
{ Textfiles in Pascal-Textformat     }
{                                    }
{************************************}

const Max          = 23551;
      Page         =   256;
      ClearEOL     =    29;
      Clearscreen  =    12;

type Byte          = 0..255;
     Units         = 1..12;
     Untypedfile   = file;
     Sector        = record
                       Block : 0..7;
                       Half  : (First, Second)
                     end;

var Map            : packed array [0..15]  of Sector;
    Buf            : packed array [0..511] of Byte;
    Text           : packed array [0..Max] of Byte;
    Sourcefile     : string;
    Destfile       : string;
    Sourceunit     : Units;
    Destunit       : Units;
    Drives         : set of Units;
    Block          : integer;
    Track          : integer;
    Sec            : integer;
    Blockcount     : integer;
    I              : integer;
    Ptr            : integer;
    done           : boolean;
    found          : boolean;
    Ch             : char;
    F              : file of char;
    Diskfile       : Untypedfile;

{*********************************************************}

procedure Initialize;

begin {Initialize}
  Map [00].Block := 0;
  Map [00].Half  := First;
  Map [01].Block := 7;
  Map [01].Half  := First;
  Map [02].Block := 6;
  Map [02].Half  := Second;
  Map [03].Block := 6;
  Map [03].Half  := First;
  Map [04].Block := 5;
  Map [04].Half  := Second;
  Map [05].Block := 5;
  Map [05].Half  := First;
  Map [06].Block := 4;
  Map [06].Half  := Second;
  Map [07].Block := 4;
  Map [07].Half  := First;
  Map [08].Block := 3;
  Map [08].Half  := Second;
  Map [09].Block := 3;
  Map [09].Half  := First;
  Map [10].Block := 2;
  Map [10].Half  := Second;
  Map [11].Block := 2;
  Map [11].Half  := First;
  Map [12].Block := 1;
  Map [12].Half  := Second;
  Map [13].Block := 1;
  Map [13].Half  := First;
  Map [14].Block := 0;
  Map [14].Half  := Second;
  Map [15].Block := 7;
  Map [15].Half  := Second
end;  {Initialize}

{*********************************************************}

procedure Printat (Y : integer; S : string);

begin {Printat}
  gotoxy (0, Y);
  write (chr (ClearEOL), S)
end; {Printat}

{*********************************************************}

procedure Uppercase (var S : string);

var I : integer;

begin {Uppercase}
  for I := 1 to length (S) do
  if S [I] in ['a'..'z'] then S [I] := chr
                                (ord (S[I]) - ord (' '))
end; {Uppercase}

{*********************************************************}

procedure Getnames;

var OK   : boolean;
    Dest : string;

begin {Getnames}
  Drives := [4..5, 9..12];
  repeat
    write (chr (Clearscreen));

    Printat (1, '*** GETDOS ***');

    Printat (3, ' 1. CONSOLE:   2. SYSTERM:   3. GRAPHIC:');
    Printat (4, ' 4. DISK #4:   5. DISK #5:   6. PRINTER:');
    Printat (5, ' 7. REMIN:     8. REMOUT:    9. DISK #9: ');
    Printat (6, '10. DISK #10: 11. DISK #11: 12. DISK #12:');

    Printat (8, 'DOS 3.3 Disk: (4, 5, 9..12): ');
    readln (Sourceunit);
    Printat (10, 'Pascal Ausgabegerät (1-12): ');
    readln (Destunit);
    Printat (12, 'DOS 3.3 Filename: ');
    readln (Sourcefile);
    Uppercase (Sourcefile);
    Destfile := Sourcefile;
    if pos ('.TEXT', Sourcefile) = 0 then
      Destfile := concat (Sourcefile, '.TEXT');
    str (Destunit, Dest);

    case Destunit of
      1 : Destfile := 'CONSOLE:';
      2 : Destfile := 'SYSTERM:';
      3 : Destfile := 'GRAPHIC:';
      4,
      5,
      9,
     10,
     11,
     12 : Destfile := concat ('#', Dest, ':', Destfile);
      6 : Destfile := 'PRINTER:';
      7 : Destfile := 'REMIN';
      8 : Destfile := 'REMOUT'
    end; {case}

    Printat (14, concat ('Pascal Filename: ', Destfile));
    Printat (16, '<RETURN> startet, <ESCAPE> beendet,
                  <SPACE> wiederholt');
    read (keyboard, Ch);
    OK := eoln (keyboard);
    if Ch = chr (27) then exit (program)
  until OK;
end; {Getnames}

{*********************************************************}

procedure Readsec;

begin {Readsec}
  Block := (Track * 8) + Map [Sec].Block;

  unitread (Sourceunit, Buf, 512, Block);
  if IOresult <> 0 then exit (program);

case Map [Sec].Half of
  First  : moveleft (Buf [0],    Text [Ptr], Page);
  Second : moveleft (Buf [Page], Text [Ptr], Page);
end; {case}
```

```
  Ptr := Ptr + Page
end; {Readsec}

{*******************************************************}

procedure Readsource;

const Bytesperentry = 35;
      Header        = 11;
      Blanks        = '                                   ';
                      {25 blanks}

var   Name          : string [30];
      Filetype      : Byte;
      Limit         : integer;
      Sectorcount   : integer;
      Entry         : integer;
      List          : packed array [1..80, 0..1] of Byte;

begin {Readsource}
  {liest DOS 3.3 Katalog von Spur 17, Sektoren 15-1}
  Printat (18, 'Lese Katalog');
  Track := 17;
  Ptr   := 0; {Nimm den Anfang des Textes}

  for Sec := 15 downto 1 do
  begin
    Readsec;
    write ('.')
  end;

  {Suche DOS 3.3 Eintritt}
  Limit := Ptr - 1;
  Ptr   := Header;
  Entry := 0;
  Sourcefile := concat (Sourcefile, ' ');

  repeat
    Name  := copy (Blanks, 1, length (Sourcefile));
    Entry := Entry + 1;
    for I := 1 to length (Sourcefile) do
      Name [I] := chr (Text [Ptr + I + 2] - 128);
    found := Name = Sourcefile;

    if not found then {erhoehe Ptr}
    begin
      if (Entry mod 7) = 0 then Ptr := Ptr + Header;
      Ptr := Ptr + Bytesperentry
    end;
  until found or (Ptr > Limit);

  if not found then
  begin
    write (chr (7));
    Printat (18, 'Fehler: File nicht gefunden');
    exit (Readsource)
  end;

  Track       := Text [Ptr];
  Sec         := Text [Ptr + 1]; {Track/Sector Liste}
  Filetype    := Text [Ptr + 2];
  Sectorcount := Text [Ptr + 33] - 1;

  if ((Filetype <> 0) and (Filetype <> 128)) or
     (Sectorcount > Max div 256) then
  begin
    Found := false;
    write (chr (7));
    Printat (18, concat ('Fehler: ',
     Sourcefile, 'ist kein Textfile'));
    exit (Readsource)
  end;

  Ptr := 0;
  Readsec;
  Ptr := 12;

  for I := 1 to Sectorcount do
  begin
    List [I ,0] := Text [Ptr];
    Ptr := Ptr + 1;
    List [I, 1] := Text [Ptr];
    Ptr := Ptr+1
  end;

  if odd (Sectorcount)
  then Blockcount := (Sectorcount + 1) div 2
  else Blockcount := Sectorcount div 2;

  fillchar (Text, size_of (Text), 0);
  Printat (18, 'Lese file');
  Ptr := 0;

  for I :=  1 to Sectorcount do
  begin
    write ('.');
    Track := List [I, 0];
    Sec   := List [I, 1];
    Readsec
  end
end; {Readsource}

{*******************************************************}

procedure Writedest;

var Nextbyte: Byte;

begin {Writedest}
  Ptr := 0;
  if Sourceunit <> Destunit then
  begin
    Printat (18, concat ('Bitte Diskette mit ', Destfile,
                         ' einlegen und <RETURN> drücken'));
    readln (keyboard)
  end; {if}

  Printat (18, 'Einen Augenblick bitte');
  fillchar (Buf, size_of (Buf), 0);

  for I := 0 to Blockcount * 512 - 1 do
  begin
    if I mod 512 = 0 then write ('.');
    if Text [I] > 127 then Text [I] := Text [I] - 128;
  end;

  Printat (18, concat ('Schreibe jetzt ', Destfile));
  write ('.');
  rewrite (Diskfile, Destfile);
  write ('.');
  I := blockwrite (Diskfile, Buf, 1, 0);
  write ('.');
  I := blockwrite (Diskfile, Buf, 1, 1);
  write ('.');
  I := blockwrite (Diskfile, Text [0], Blockcount, 2);
  close (Diskfile, lock)
end; {Writedest}

{*******************************************************}

begin {Getdos}
  Initialize; {Sektors/Blocks}

  repeat
    Getnames;
    Readsource;
    if found then Writedest;
    writeln;
    writeln;
    write ('Nochmal ? [j/n]');
    read (keyboard, Ch);
    Done := Ch in ['N','n']
  until done
end. {Getdos}
```

COPYDUPDIR zu Pascal-Directory unter der Lupe

```
program CopyDupDir (input, output);

const Maxdir      =    77;
      Vidleng     =     7;
      Tidleng     =    15;
      Fblksize    =   512;
      Dirblk      =     2;
      DupDirblk   =     6;

type  Daterec     = packed record
                      Month : 0..12;
                      Day   : 0..31;
                      Year  : 0..100
                    end; {Daterec}
```

```
      Dirrange    = 0..Maxdir;
      Vid         = string [Vidleng];
      Tid         = string [Tidleng];

      Filekind    = (Untypedfile, Xdskfile, Codefile,
                     Textfile, Infofile, Datafile, Graffile,
                     Fotofile, Securedir, Subsvol);

      Direntry    = packed record
                      Dfirstblk : integer;
                      Dlastblk  : integer;
                      case Dfkind : Filekind of
                        Securedir,
                        Untypedfile : (Fillerl   : 0..4095;
                                       Dvid      : Vid;
                                       Deovblk   : integer;
                                       Dnumfiles : Dirrange;
                                       Dloadtime : integer;
                                       Dlastboot : Daterec);
                        Xdskfile,
                        Codefile,
                        Textfile,
                        Infofile,
                        Datafile,
                        Graffile,
                        Fotofile,
                        Subsvol     : (Filler2   : 0..2047;
                                       Status    : boolean;
                                       Dtid      : Tid;
                                       Dlastbyte : 1..Fblksize;
                                       Daccess   : Daterec)
                    end; {Direntry}

      Directory   = array [Dirrange] of Direntry;

      ValidValues = (NotValid, ValidMain, ValidDupl);

var   Unitnum     : integer;
      MainDir     : Directory;
      DuplDir     : Directory;

{-------------------------------------------------------------}

procedure ReadDir;

var Ok : ValidValues;

{-------------------------------------------------------------}

  function FetchDir (Unitnum : integer) : ValidValues;

{-------------------------------------------------------------}

  function Valid (var Dir : Directory; Block : integer) : boolean;

  var Ok : boolean;

  begin {Valid}
    Valid := false;
    unitread (Unitnum, Dir, size_of (Dir), Block);
    if IOresult = 0 then
    with Dir [0] do
    Valid := (Dfirstblk = 0) and (Dfkind in
              [Untypedfile, Securedir])
              and (length (Dvid) > 0) and
              (length (Dvid) <= Vidleng)
              and (Dnumfiles >= 0) and (Dnumfiles <= Maxdir)
  end; {Valid}

{-------------------------------------------------------------}

  begin {FetchDir}
    if Valid (MainDir, Dirblk)
    then FetchDir := ValidMain
    else
      if Valid (DuplDir, DupDirblk)
      then FetchDir := ValidDupl
      else FetchDir := NotValid
  end; {FetchDir}

{-------------------------------------------------------------}

begin {ReadDir}
  page (output);
  writeln;
  writeln ('Duplicate-Directory Verschieber
            von Dieter Geiß, 30. 11. 1984');
  repeat
    writeln;
    write ('Welche Directory soll repariert werden?
            Unitnummer: ');
    readln (Unitnum);
  until Unitnum in [4, 5, 9..12];
  Ok := FetchDir (Unitnum);
  case Ok of
    NotValid  : begin
                  writeln (chr (7));
                  writeln ('Entweder ist kein Laufwerk
                            angeschlossen oder');
                  writeln ('es ist keine Diskette
                            im Laufwerk oder');
                  writeln ('die Diskette ist keine
                            Pascal-Diskette oder');
                  writeln ('sowohl Haupt- als auch
                            Duplicate-Directory');
                  writeln ('sind zerstört.');
                  exit (CopyDupDir)
                end; {NotValid}
    ValidMain : begin
                  writeln;
                  writeln ('Das Haupt-Directory ist
                            in Ordnung.');
                  writeln ('Kein Reparieren notwendig.');
                  exit (CopyDupDir)
                end; {ValidMain}
    ValidDupl : begin
                  writeln;
                  writeln ('Das Haupt-Directory ist zerstört,
                            aber das');
                  writeln ('Duplicate-Directory ist in Ordnung.')
                end {ValidDupl}
  end {case}
end; {ReadDir}

{-------------------------------------------------------------}

procedure WriteDir;

begin {WriteDir}
  unitwrite (Unitnum, DuplDir, size_of (DuplDir), Dirblk);
  writeln;
  if IOresult = 0
  then writeln ('Die Diskette hat wieder ein gültiges Directory')
  else
    begin
      write (chr (7));
      writeln ('Die Blöcke 2 - 6 sind physikalisch
                nicht in Ordnung.');
      writeln ('Bitte reparieren Sie diese und
                versuchen es dann nochmals.')
    end {else}
end; {Write Dir}

{-------------------------------------------------------------}

begin {CopyDupDir}
  ReadDir;
  WriteDir
end {CopyDupDir}.
```

**Hätten Sie jemals geglaubt,
daß es einen 8-MHz-68000-Mikro gibt,
der für den Einzeiler
FOR X = 1 TO 10000: Y = SIN (X): NEXT
mehr Zeit als der 1-MHz-6502-Apple benötigt?
Dann lesen Sie jetzt,
wie die Mac-Leute den 68000 zur Schnecke
und den Anwender zum Mecki gemacht haben.**

Der Macintosh und die Lisa gehören zu derjenigen Klasse von Mikrocomputern, mit denen eine ganz neuartige Philosophie propagiert wird, über deren Sinn und Unsinn an dieser Stelle diskutiert werden soll, wobei auf technische Details – wie sonst im „Peeker" üblich – bewußt verzichtet wird, um den philosophischen Hintergrund besser erhellen zu können.

Ikonen und Deixis oder die Philosophie des Macintosh

von Ulrich Stiehl

Folgendes soll vorab klargestellt werden: Der Macintosh ist ein überlegenes Gerät, soweit es für Applikationen eingesetzt werden soll, die hochauflösende Schwarzweißgrafik benötigen. Der Macintosh wird jedoch von der Firma Apple nicht als ein spezieller Grafikcomputer propagiert, sondern als ein in Büros schlechthin einsetzbarer genereller Mikrocomputer. Zumindest im kommerziellen Büroeinsatz (Textverarbeitung, Buchhaltung, Adreßverwaltung, Tabellenkalkulation usw.) wäre ein vornehmlich auf Grafik ausgelegter Computer fehl am Platze, so daß nachstehend geprüft werden soll, welche Effizienz der Macintosh zusätzlich für die weitaus mehr gefragten nicht-grafischen Applikationen mitbringt.

Wenn man mit dem Macintosh spielt, kommen einem zwei Begriffe in den Sinn, nämlich Ikonisation und Deixis. In einer „Einführung in die allgemeine Semantik", die ich vor 15 Jahren veröffentlicht habe, schrieb ich u. a. über das Wesen des Zeichens. Da die damaligen Erörterungen auch heute noch nicht ihre Gültigkeit verloren haben, seien einige kurze Passagen aus dem Zeichen-Kapitel jenes Buches zitiert:

Die Gesamtheit aller Zeichen teilt man vorteilhaft in sprachliche und nichtsprachliche Zeichen auf, da die sprachlichen Zeichen wegen ihrer überragenden Bedeutung eine eigenständige Zeichenklasse bilden, hinter der die nichtsprachlichen Zeichen an Relevanz für den Informationsverkehr stark zurücktreten...
Zeichen lassen sich in natürliche und in künstliche Zeichen aufteilen. Künstliche Zeichen liegen dann vor, wenn der Signorezipient (Zeichenempfänger) diesen künstlichen Zeichen nicht direkt ihre Bedeutungen entnehmen kann, womit also auch keine entsprechende Vorstellung in seinem Bewußtsein auftauchen kann. Künstliche nichtsprachliche Zeichen werden zum reibungslosen Ablauf des interpersonalen Informationsverkehrs stets da

eingeführt, wo natürliche Zeichen, denen man unmittelbar ihre Bedeutungen entnehmen kann, nicht mehr gefunden werden können und andererseits eine sprachliche Vermittlung zu umständlich wäre...
Wenn ein Zeichen in vielen charakteristischen Eigenschaften mit dem Designat der von ihm intendierten Vorstellung übereinstimmt und dieses Designat quasi abbildet, kann der Signorezipient die Bedeutung des Zeichens verstehen. Je weniger Ähnlichkeit das ikonisierende Zeichen mit dem Designat der entsprechenden Vorstellung aufweist, um so mehr wird es zu einem künstlichen Zeichen...
Das Wort „Zeichen" läßt sich auf das Wort „zeigen" zurückführen und damit wird zugleich die ursprüngliche Bedeutung des Bezeichnens angedeutet, nämlich die des direkten Hindeutens... Dieses Hindeuten nennt man das deiktische Zeichen...

Wenn man den Macintosh mit anderen Computern vergleicht, auch mit dem Apple II usw., dann fällt ins Auge, daß zwei klassische kulturelle Fähigkeiten, nämlich das Lesen und das Schreiben, hier stark zurückgedrängt werden. An die Stelle des Lesens tritt das Betrachten von Ikonen, wie die Macintosh-Bildchen in der amerikanischen Literatur bezeichnet werden, und an die Stelle der dem Schreiben dienenden Tastatur tritt die Maus als deiktisches Instrument. Der oberste Manager des Apple-Unternehmens, John Sculley, der bekanntlich früher bei Pepsi Cola war, mag zwar von Mikrocomputern aus technischer Sicht wenig verstehen. Dafür beherrscht er aber das Marketing aus dem Effeff, und so ist es ihm nicht entgangen, daß die Analphabetenquote nach einer UNESCO-Erhebung aus dem Jahre 1977 in Nord-Amerika über 10% beträgt und die Fähigkeit des Lesens und Schreibens (= „Alphabetentum") einen Abwärtstrend zeigt. Wie soll man jedoch eine Marktpenetration erreichen können, wenn es so viele Analphabeten gibt? Ergo wurden die Ikonen und die Maus erfunden, denn wenn jemand schon nicht lesen kann, Bildchen wird er wohl noch deuten können, und wenn jemand schon nicht schreiben kann, mit einer Maus auf Bildchen zeigen wird wohl noch möglich sein. Nehmen wir als Beispiel die Papierkorb-Ikone. Wenn man eine Datei namens XYZ auf der Diskette löschen will, so muß man bei konventionellen Mikrocomputern aus einem Anwenderprogramm heraus nach der Frage „Datei XYZ löschen J/N" auf die J-Taste tippen oder man muß (wie etwa bei vielen Datenbankprogrammen) einen Befehl in der Art „Lösche XYZ" über das Keyboard eingeben. Anders beim Macintosh. Hier muß man das Bildchen der Datei in das Bildchen des Papierkorbs stecken, indem man mit der Maus zunächst auf die Datei-Ikone und dann auf die Papierkorb-Ikone zeigt. Um darüber hinaus den Leseunkundigen behutsam an das geschriebene Wort heranzuführen, ist zu allem Überfluß die Papierkorb-Ikone mit dem Schild „Papierkorb" beschriftet.

Halten wir zunächst einmal fest: Computer benutzerfreundlicher zu machen, sollte oberstes Anliegen jedes Computerproduzenten sein, denn nicht jeder will, kann und soll programmieren lernen. Aber drückt sich Benutzerfreundlichkeit darin aus, daß abstrakte Vorgänge wie das Löschen einer Datei mit aller Gewalt bildhaft veranschaulicht werden? Wir Deutsche sind etwas empfindlich gegenüber einer zu trivialen Simplifikation. Dies hängt mit der philosophischen Tradition und dem Schulsystem unseres Landes zusammen. Haben doch viele Philosophen wie Kant und Hegel ihre Zuflucht in dunklen Formulierungen gesucht (Brevis esse laboro et obscurus fio...). Mit dieser Tradition im Hintergrund ist man etwas konsterniert, wenn man beispielsweise nunmehr auf dem Bildschirm ohne jeden erläuternden Text einen „Hasen" sowie eine „Schildkröte" präsentiert bekommt. Was könnte es bewirken, wenn ich mit der „Maus" (auch wieder eine bildhafte Umschreibung eines sonst Rollball genannten Instruments) auf den Hasen „klicke"? Daß ich jetzt vom Macintosh ein Wildbret anstelle einer Schildkrötensuppe erhalte? Oder daß – was zutrifft – die Repeat-Funktion der Tasten des Macintosh eingestellt wird? Warum nicht einfach und eindeutig die schriftliche Frage: „Tastendruck schneller J/N?" So gesehen muß man sich fragen, ob Bildchen wirklich zur Benutzerfreundlichkeit führen. Hier zeigt sich die oben aus dem Semantik-Buch zitierte Problematik ikonographischer Zeichen, die oft äquivok sind und deshalb einer Konvention (= Festlegung der Bedeutung) bedürfen. Wenn man jedoch die Symbolbedeutung des „Hasen" erst erlernen muß, dann fragt es sich, ob einem mit einer schriftlichen Formulierung nicht besser gedient wäre. *Nicht*sprachliche Zeichen sind stets dann erforderlich, wenn eine sprachliche (schriftliche) Kommunikation zwischen Mensch und Computer nicht möglich ist. Hier sind drei Fälle zu nennen:

Ausländer: Aus der Sicht der Amerikaner sind die Deutschen Ausländer. Betriebs-

Scrollen: Meckis Macke

Daß selbst ein so hervorragender Prozessor wie der MC68000 bei zu aufwendiger Pixel-Grafik in die Knie geht, läßt sich anhand der Scroll-Geschwindigkeit beweisen. Zu diesem Zweck wurde je ein kurzes Basic-Programm (FOR X = 1 TO 1000: PRINT „aaaaa...usw.": NEXT) erstellt und einerseits auf dem Macintosh und andererseits auf dem Apple IIe mit 80-Zeichenkarte getestet. Dabei ergaben sich folgende Geschwindigkeitsunterschiede des Apple IIe gegenüber dem Macintosh:

1. Der normale Apple IIe mit 1MHz-6502 und normaler 80-Zeichenkarte-Routine ist immerhin schon 2mal so schnell wie der Macintosh.

2. Der normale Apple IIe mit einer selbst geschriebenen 80-Zeichenkarte-Fastscroll-Routine ist bereits 4mal so schnell wie der Macintosh.

3. Und schließlich ist der Apple IIe mit 3,5MHz-65C02C (Acclerator-Karte) und der Fastscroll-Routine sage und schreibe 10mal so schnell wie der Macintosh.

Dieses Ergebnis ist für den Macintosh angesichts des 8MHz-MC68000-Prozessors ungewöhnlich schwach.

Um den Basic-Test zu erhärten, wurde die Scrollgeschwindigkeit des Macwrite- mit der des Applewriter-Programms verglichen. Zunächst wurde von der Macwrite-Diskette der kurze Demotext namens MEMORANDUM eingelesen. Am Anfang dieses Textes wurden exakt 10 Bildschirmzeilen *ohne Returns und ohne Leertasten* („-----usw.") eingegeben, um auf diese Weise das Scrollen zu erzwingen. Dann wurde der Maus-Zeiger zum Beginn der 10 Endloszeilen gesetzt und innerhalb von 5 Sekunden exakt 10mal auf die Return- und 10mal auf die Delete-Taste gedrückt. Es dauerte dann genau weitere 10 Sekunden, bis sich das 10malige Abwärts- und Aufwärts-Scrollen erholt hatte. Beim Applewriter IIe mit Acclerator-Karte gibt es demgegenüber überhaupt keine Verzögerung, d.h. das Scrollen wird sofort ausgeführt. Was kann man nun daraus folgern? Eine mäßige Schreibkraft erzielt 180, eine gute 280 Anschläge/Minute. Wenn in 5s 20 Tasten gedrückt, aber effektiv erst in 5s + 10s = 15s ausgeführt werden, so entspricht dies einer Tastgeschwindigkeit von 80 Anschlägen pro Minute. Niemand würde eine Schreibkraft mit dieser Leistung einstellen.

systeme in die jeweilige Landessprache zu übersetzen kostet Zeit und Geld. Durch die Einführung von Symbolen könnte man sich die Arbeit erheblich erleichtern. Dann müßten es allerdings international geregelte Symbole sein (ähnlich wie die international genormten Straßenverkehrsschilder) und nicht willkürlich festgelegte „Hasen" und „Schildkröten".

Kinder: Kinder sind erst im Begriff, lesen und schreiben zu lernen. Ob der Macintosh indessen auch für Kinder konzipiert wurde, ist zu bezweifeln, denn dem steht der zu hohe Anschaffungspreis entgegen.

Analphabeten: Für Lesebehinderte ist die ikonographisch-deiktische Sprache das Mittel zur Wahl. Doch scheint mir zumindest der berufliche Einsatz eines Mikrocomputers ohne diese elementaren Fähigkeiten mehr als bedenklich.

Manager: Die Bedienung einer Tastatur wird von Führungskräften in der Regel abgelehnt. Dafür gibt es Sekretärinnen, so heißt es. Um den Macintosh in der Geschäftsetage allererst „salonfähig" zu machen, bot sich die Maus an. Die Tastatur ist beim Macintosh scheinbar ein nebensächliches Anhängsel, das nur noch von Fall zu Fall bedient werden muß. In Wahrheit ist jedoch die Tastatur beispielsweise bei der Textverarbeitung eine conditio sine qua non. Da man jedoch beim Macintosh *gezwungen* ist, die Maus zu benutzen, muß das permanente Hin und Her zwischen Maus und Tastatur als wenig ergonomisch bezeichnet werden, ganz zu schweigen von der Tatsache, daß dadurch Schnellschreiber ihr gewohntes Soll nicht mehr erfüllen können. Oder sollte der Macintosh für Leute, die schnell arbeiten müssen, gar nicht gedacht sein?

Läßt man unsere philosophischen Erörterungen Revue passieren, so könnte der Eindruck entstehen, daß der Macintosh – von der Zielgruppe der Manager einmal abgesehen – eigentlich nur für die von Mutter Natur weniger gut Bedachten konzipiert worden ist. („Dada Hase klickiklick und Mecki ganz schnell".) Dem ist jedoch nicht so. Hinter dem Macintosh steht das ernsthafte Bemühen, ein Gerät zu konzipieren, daß über optimale Benutzerfreundlichkeit verfügt, was mit Hilfe von Ikonen, Maus und sonstigen Grafiken zu realisieren versucht wird. Ganz an erster Stelle steht beim Macintosh das Komfortprinzip und erst unter „ferner liefen" das Effizienzprinzip. Ohne Grafik geht beim Macintosh gar nichts. Dies hat zwar einerseits den Vorteil, daß Texte und Bilder beliebig gemischt werden können, andererseits wird die Effizienz ungewöhnlich stark beeinträchtigt. Überspitzt könnte man sagen, daß der Macintosh im wesentlichen damit beschäftigt ist, pausenlos Ikonen und Grafiken zwischen internem und externem Speicher hin- und herzuschaufeln und daß deshalb für das eigentliche Anwenderprogramm nur noch wenig „CPU-Zeit" und nur noch wenig Speicherraum zur Verfügung steht.

– Tatsache ist, daß wegen des ca. 20K-Bildschirmspeichers die Scrollgeschwindigkeit so gering ist, daß bei einer flotten 10-Finger-Schreibkraft Textverarbeitungsprogramme in der Art des Macwrite – unabhängig von dem permanenten Wechsel Maus–Tastatur – nicht mithalten können.

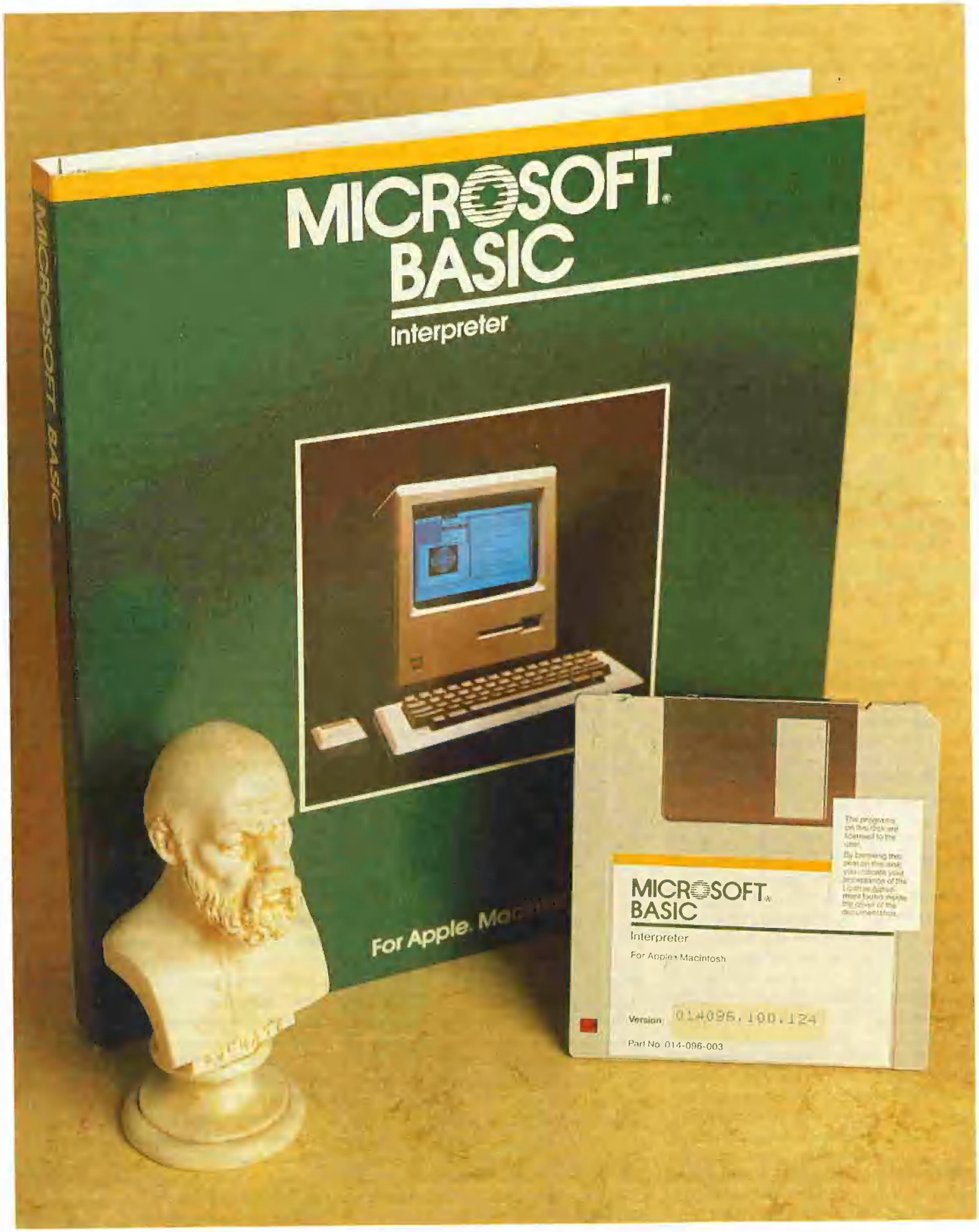
– Tatsache ist auch, daß sowohl der interne RAM- wie auch der externe Disketten-Speicher durch das infolge der aufwendigen Grafikroutinen ungewöhnlich aufgeblähte Betriebssystem keinen nennenswerten Platz für eigene Programme und Daten läßt, so daß insbesondere 1-Drive-Besitzer ungebührliche Einschränkungen in Kauf nehmen müssen. Auf einer 1-Drive-Macintosh-Diskette befinden sich in der Regel weit mehr Systemprogramme, Zeichensätze und Bildchen als sinnvolle Daten des Anwenders. Übrigens dauert das Duplizieren einer Diskette beim 1-Drive-Besitzer 6,5 Minuten.

Als konkretes Beispiel zitieren wir hier aus der amerikanischen Zeitschrift „A+", Heft 10/1984, S. 175, die Anfrage eines verzweifelten Macintosh-Benutzers, der wegen der genannten speichertechnischen Probleme eine 5-Megabyte-Festplatte erworben hatte und nunmehr feststellen mußte, daß er bei der Verwendung langer Dateinamen nicht mehr als 100 Dateien auf der Harddisk abspeichern konnte, obgleich de facto noch mehr als 1,5 Megabyte frei waren. Bei jedem wiederholten Versuch stürzte das System mit „can't find workspace" erneut ab. Was war passiert? Das Kopierprogramm FILER hat einen Directory-Puffer (Dateinamenpuffer), der nur ca. 100 längere Dateinamen aufnehmen kann. Andererseits ist es jedoch auch nicht in der Lage, die Summe aller Dateinamen in Blöcken einzulesen, wie dies bei anderen Betriebssystemen wie ProDOS usw. der Fall ist. Hierzu muß man wissen, daß der RAM-Bereich des Macintosh 128K umfaßt, während der Directory-Puffer wahrscheinlich nicht mehr als ca. 4K einnimmt, weil die restlichen ca. 124K durch Systemprogramm und Ikonen belegt sind. Hierzu muß man ferner wissen, daß sich in dem Speicherraum des Macintosh-Grafikbildschirms, der ca. 20K umfaßt, mehr als 1000 Dateinamen unterbringen lassen könnten.

Symptomatisch ist die Antwort, die die appletreue „A+" dem Anfrager gibt: „Sie können wenige große Datenbank-Dateien auf Ihrer Harddisk unterbringen, nicht jedoch viele kleine Macwrite/Macpaint-Dateien. Sie sollten einige der Dateien von der Harddisk auf eine Diskette übertragen, womit Ihr Problem einstweilen gemildert wird." Wirklich gemildert? Wenn das Effizienzprinzip dem Komfortprinzip geopfert wird, wenn also wegen der „Bildchen" eine Festplatte nicht mehr optimal genutzt werden kann, dann kann man nur noch auf die Schildkröte deuten: „Dada klickiklick und Mecki wieder brav".

Ich möchte folgendes Fazit ziehen: Der Macintosh ist im Grunde ein zukunftweisendes Gerät, doch ist selbst ein so hervorragender Prozessor wie der M68000 nicht in der Lage, exzessive Grafiken schnell und speicherökonomisch zu verwalten. Da müssen dann schon echte 32-Bit-Prozessoren herhalten. Hinzu kommt, daß insbesondere bei kommerziellen Applikationen Grafik oft entbehrlich ist. Wer wollte schon 5000 Kundenadressen in 12p Chicago drucken? Deshalb ist es in meinen Augen nicht nur wünschenswert, sondern notwendig, daß die Firma Apple neben dem Grafik-Modus beim Macintosh auch den normalen Text-Modus einführt, weil sonst die Gefahr besteht, daß der Macintosh zu einem Spezialgerät abgestempelt wird, das nur denjenigen Benutzern attraktiv erscheint, die von Berufs wegen mit Grafik zu tun haben, also die Grafiker selbst. Will man jedoch *weder* auf die Grafik *noch* auf die Effizienz verzichten, dann muß man tief in die Tasche greifen. Laut Apple-Preisliste vom November 1984 kostet der Macintosh mit der nunmehr erhältlichen 512K-Speichererweiterung 12.250,- DM. Nimmt man noch die Kosten für Drucker, Festplattenlaufwerk oder externes Drive sowie für Anwendersoftware hinzu, ist man schnell in einer Preiskategorie, die nur diejenigen Firmen nicht abschreckt, für die Geld keine Rolle spielt.

(Zur Mac-Sprache s. S. 94.)


MICROSOFT BASIC
Interpreter
MICROSOFT BASIC
For Apple Macintosh
MICROSOFT BASIC
Interpreter
For Apple Macintosh
Version 014096.100.124
Part No. 014-096-003

Microsoft Basic leicht geMACht

Teil 2: Funktionen und Programmsteuerbefehle

von Pit Capitain

4. FUNKTIONEN

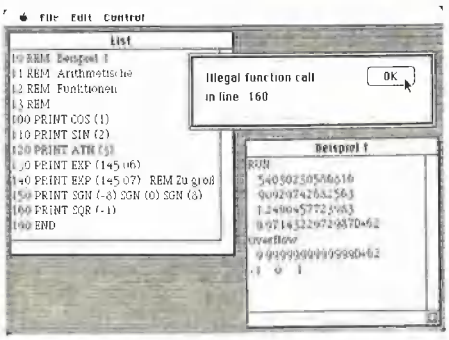
Mit diesem zweiten Teil wird unsere Serie über das Microsoft Basic, Version 1.0, für den Macintosh fortgesetzt. Es sei an dieser Stelle darauf hingewiesen, daß eine verbesserte und und teilweise erweiterte Version dieses Basic-Dialekts in Kürze erscheint, so daß wir wahrscheinlich bereits im abschließenden Teil dieser Serie auf die Neuerungen hinweisen können.

4.1. Arithmetische Funktionen

Beim Microsoft-Basic für den Macintosh gibt es dieselben arithmetischen Funktionen wie beim Applesoft-Basic. Sie werden jedoch mit einer höheren Genauigkeit berechnet, nämlich mit der (von den numerischen Variablen her bekannten) doppelten Genauigkeit von bis zu 14 gültigen Ziffern. Im einzelnen gibt es die folgenden arithmetischen Funktionen:

COS (X) – Cosinus von X
SIN (X) – Sinus von X
TAN (X) – Tangens von X
ATN (X) – Arcustangens von X
EXP (X) – Exponentialfunktion
LOG (X) – natürlicher Logarithmus
SQR (X) – Quadratwurzel aus X
ABS (X) – Betrag von X
SGN (X) – Signum-(Vorzeichen-)Funktion

Neben der genaueren Berechnung gibt es beim Microsoft-Basic noch einen weiteren Unterschied zum Applesoft-Basic. Wenn bei einer arithmetischen Funktion ein Überlauf auftritt, so wird zwar eine Fehlermeldung ausgegeben, das Programm hält aber nicht an, sondern läuft nach der Fehlermeldung weiter. Das Funktionsergebnis ist dann die dem Betrage nach größte darstellbare Zahl, nämlich 9.9999999999999 * 10 ↑ + 62, natürlich mit dem richtigen Vorzeichen versehen.
Anders verhält es sich selbstverständlich bei dem Fall, daß eine Funktion mit einem Argument aufgerufen wird, das nicht in dem Definitionsbereich der Funktion liegt, z.B. „SQR (-1)". In diesem Fall wird die Fehlermeldung „ILLEGAL FUNCTION CALL" ausgegeben, und das Programm wird wie beim Applesoft-Basic unterbrochen.
Beispiele für die arithmetischen Funktionen und für die eben erwähnte Fehlerbehandlung sieht man in **Abb. 1**.



4.2. String-Funktionen

Alle String-Funktionen des Applesoft-Basic sind vorhanden:

LEFT$ (X$,I) – Linker Teilstring von X$ mit der Länge I
MID$ (X$,I) – Rechter Teilstring von X$ ab dem I-ten Zeichen
MID$ (X$,I,J) – Mittlerer Teilstring von X$ ab dem I-ten Zeichen mit der Länge J
RIGHT$ (X$,I) – Rechter Teilstring von X$ mit der Länge I
LEN (X$) – Länge des Strings X$

Im Unterschied zum Applesoft-Basic kann bei der „LEFT$"- und „RIGHT$"-Funktion der Längenparameter 0 sein. In diesem Fall wird der leere String "" erzeugt.
Neben diesen Funktionen existieren noch drei weitere String-Funktionen:

SPACE$ (I) – ein String aus I Leerzeichen.
STRING$ (I,J) – ein String aus I Zeichen mit dem ASCII-Code J. Beispiel: STRING$ (5,42) = "*****".
STRING$ (I,X$) – ein String, der aus I-mal dem ersten Zeichen von X$ besteht. Beispiel: STRING$ (4,"Test") = "TTTT".
INSTR (I,A$,B$) – sucht das erste Vorkommen des Strings B$ in dem String A$ ab dem I-ten Zeichen. Wenn B$ in A$ gefunden wird, so wird die Position von B$ in A$ zurückgegeben, ansonsten eine Null. Der Parameter I kann auch weggelassen werden; dann fängt die Suche am Anfang von A$ an.

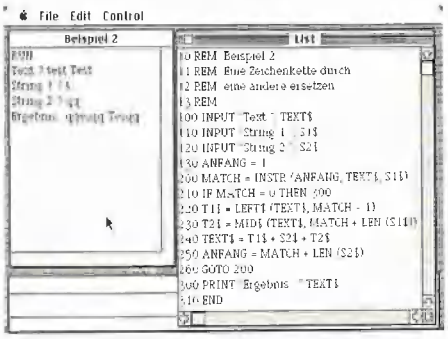


Abb. 2 zeigt ein Beispiel für die Verwendung von String-Funktionen. Es wird in einem String eine bestimmte Zeichenfolge durch eine andere ersetzt.

4.3. Typ-Umwandlung

Beim Microsoft-Basic gibt es verschiedene Arten von Operanden (vgl. 1. Teil unserer Serie). Mittels einer ganzen Anzahl von Funktionen läßt sich ein bestimmter Typ in einen anderen umwandeln.

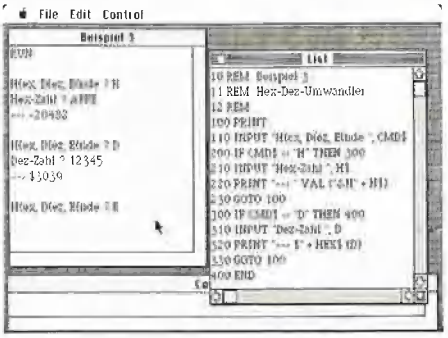
FIX (X) – schneidet alle Nachkommastellen von X ab. Das Ergebnis ist eine Integerzahl.
INT (X) – liefert die größte Integerzahl, die kleiner oder gleich X ist.

Der Unterschied zwischen „FIX" und „INT" wird bei negativen Argumenten deutlich: FIX (-1.5) = -1, aber INT (-1.5) = -2. Es gilt:
FIX (X) = SGN (X) * INT (ABS (X)).

CDBL (X) – die Zahl X wird im doppelt genauen Format zurückgegeben.
CSNG (X) – die Zahl X wird im einfach genauen Format zurückgegeben. Dabei wird X auf 6 Ziffern gerundet, falls dies nötig ist (z.B. CSNG (1.23456789) = 1.23457).
CINT (X) – die Zahl X wird zu einer Integerzahl gerundet (z.B. CINT (1.5) = 2).

Neben diesen Funktionen, die einen numerischen Ausdruck in eine numerische Zahl in anderem Format umformen, gibt es noch Funktionen, die aus einer Zahl einen String machen und umgekehrt:

ASC (X$) – ASCII-Code des ersten Zeichens von X$.
CHR$ (I) – ein String, der aus dem Zeichen mit ASCII-Code I besteht.
VAL (X$) – interpretiert den String X$ als Darstellung einer Zahl und liefert diese zurück.
STR$ (X) – Darstellung der Zahl X als String.
HEX$ (I) – wie „STR$", die Integerzahl I wird jedoch in hexadezimale Form umgewandelt.
OCT$ (I) – wie „STR$", die Integerzahl I wird jedoch in oktale Form umgewandelt.



Als kleines Beispiel ist in **Abb. 3** ein Programm aufgelistet, das Dezimalzahlen in Hexadezimalzahlen umwandelt und umgekehrt.
Zu den Typ-Umwandlungs-Funktionen gehören noch sechs weitere Funktionen, die zum Abspeichern von Zahlen in sogenannten „Random-Access"-Dateien benötigt werden. Diese Funktionen wandeln eine Zahl in einen String um und umgekehrt, aber anders als die Funktionen „STR$" und „VAL":

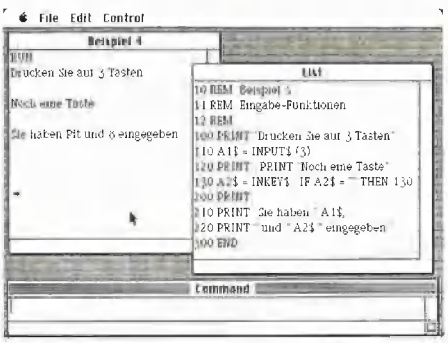
MKI$ (I) – die (2 Byte lange) Integerzahl I wird in einen String umgewandelt, der 2 Zeichen lang ist. Dabei entspricht jedes Zeichen einem Byte der internen Darstellung von I. Zum Beispiel ergibt MKI$ (65 * 256 + 80) den String „AP", da 65 der ASCII-Code von „A" und 80 der von „P" ist.
MKS$ (X) – analog wie „MKI$", nur wird die einfach genaue rationale Zahl X (intern 4 Bytes lang) in einen 4 Zeichen langen String umgewandelt.
MKD$ (X) – analog wie „MKI$", nur wird die doppelt genaue rationale Zahl X (intern 8 Bytes lang) in einen 8 Zeichen langen String umgewandelt.
CVI (X$) – der 2 Zeichen lange String X$ wird in eine Integerzahl umgewandelt (umgekehrt wie bei „MKI$"). Z.B. ist CVI ("AP") = 65 * 256 + 80.
CVS (X$) – der 4 Zeichen lange String X$ wird in eine einfach genaue rationale Zahl umgewandelt (umgekehrt wie bei „MKS$").
CVD (X$) – der 8 Zeichen lange String X$ wird in eine doppelt genaue rationale Zahl umgewandelt (umgekehrt wie bei „MKD$").

Beispiele zu diesen sechs Funktionen folgen im nächsten Teil der Serie bei den Ein-/Ausgabe-Befehlen.

4.4. Ein-/Ausgabefunktionen

Es gibt einige Funktionen, die mit der Ein- bzw. Ausgabe im Zusammenhang stehen. Eine dieser Funktionen ist auch beim Applesoft-Basic vorhanden, die anderen sind neu.

**INKEY$** – Falls eine Taste des Macintosh gedrückt wurde, die noch nicht vom Programm eingelesen worden ist (man kann ja z. B. im voraus tippen), so liefert die Funktion einen String, der aus diesem Zeichen besteht. Ansonsten liefert „INKEY$" den leeren String "". Keines der so eingelesenen Zeichen wird auf dem Bildschirm ausgegeben.
INPUT$ (I,#N) – gibt einen String zurück, der aus I Zeichen besteht. Diese Zeichen werden alle von der Datei mit der Nummer N eingelesen. Sie werden nicht auf dem Bildschirm ausgegeben. Man kann den zweiten Parameter weglassen; in diesem Fall werden I Zeichen von der Tastatur eingelesen.



Beispiele für diese Funktionen sieht man in **Abb. 4**.
Die nächsten Funktionen machen eine Aussage über den Zustand eines Ein-/Ausgabe-Gerätes:

POS (I) – gibt die augenblickliche horizontale Position der Bildschirmausgabe an. Auf dieser Position erfolgt die nächste Ausgabe.
LPOS (I) – wie „POS", nur wird die Position des Schreibkopfes beim Drucker zurückgemeldet.
EOF (N) – hat den Wert -1 (wahr), falls das Ende der Datei mit der Nummer N erreicht wurde. Wenn dies nicht der Fall ist, so hat EOF den Wert 0 (falsch).
LOF (N) – gibt die Länge der Datei mit der Nummer N in Bytes an. Für den Bild-

schirm, die Tastatur und den Drucker wird eine Null zurückgegeben.
LOC (N) – bestimmt die Position der letzten Ein-/Ausgabe bei der Datei mit der Nummer N, z. B. bei sequentiellen Dateien die Zahl der bisher gelesenen bzw. geschriebenen Zeichen. Bei der Tastatur liefert LOC eine 1, falls ein Zeichen eingelesen werden kann, sonst eine 0. Bei dem Text-Zwischenspeicher und bei der seriellen Schnittstelle ("CLIP:" bzw. "COM1:", vgl. 1.3.) gibt LOC die Zahl der Zeichen an, die eingelesen werden können.
POINT (X,Y) – die Farbe eines Punktes im Ausgabefenster mit den Koordinaten (X,Y). Der gesamte Macintosh-Bildschirm hat eine Auflösung von 512 Punkten horizontal und 342 Punkten vertikal. Punkt (0,0) ist die linke obere Ecke des Ausgabefensters. Ein weißer Punkt hat die „Farbe" Nummer 30, ein schwarzer Punkt die Nummer 33. Falls die Koordinaten (X,Y) außerhalb des Ausgabefensters liegen, so ist POINT (X,Y) = -1. Obwohl zur Zeit nur 2 Farben (weiß und schwarz) zur Verfügung stehen, sind in den ROM-Routinen und auch im Basic alle Vorkehrungen für eine spätere Farberweiterung vorhanden.

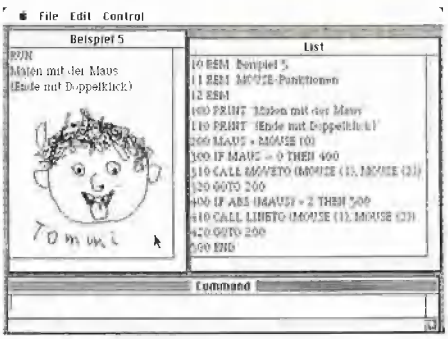
Eine weitere interessante Eingabefunktion gibt den Zustand der Maus an:

MOUSE (I) – kann mit Parametern von 0 bis 6 aufgerufen werden. Je nach dem Wert von I hat diese Funktion eine andere Bedeutung:

MOUSE (0) – (Status) gibt den Zustand vom K(n)opf der Maus an. Eine Null bedeutet, daß der Knopf seit dem letzten Aufruf von „MOUSE (0)" nicht gedrückt worden ist. Falls der Knopf doch gedrückt wurde, gibt „MOUSE (0)" an, wie oft das der Fall war, einmal, zweimal („Doppelklick") oder gar dreimal. Das Vorzeichen von „MOUSE (0)" gibt an, ob der Knopf der Maus noch gedrückt ist (negativ) oder ob er schon wieder losgelassen wurde (positiv). Beispiele:

Knopf wurde zweimal gedrückt und dann losgelassen: MOUSE (0) = 2.
Knopf wurde einmal gedrückt und ist noch unten: MOUSE (0) = -1.
Knopf wurde nicht gedrückt:
MOUSE (0) = 0.

MOUSE (1) – (aktuelle X-Koordinate): die horizontale Position der Maus zu dem Zeitpunkt, als „MOUSE (0)" zuletzt aufgerufen wurde.
MOUSE (2) – (aktuelle Y-Koordinate): die vertikale Position der Maus zu dem Zeitpunkt, als „MOUSE (0)" zuletzt aufgerufen wurde.
MOUSE (3) – (Anfangs-X-Koordinate): die horizontale Position der Maus zu dem Zeitpunkt, als der Knopf der Maus gedrückt wurde.
MOUSE (4) – (Anfangs-Y-Koordinate): die vertikale Position der Maus zu dem Zeitpunkt, als der Knopf der Maus gedrückt wurde.
MOUSE (5) – (End-X-Koordinate): die horizontale Position der Maus zu dem Zeitpunkt, als der Knopf der Maus losgelassen wurde.
MOUSE (6) – (End-Y-Koordinate): die vertikale Position der Maus zu dem Zeitpunkt, als der Knopf der Maus losgelassen wurde.



Ein Beispiel zur Benutzung der „MOUSE"-Funktion zeigt **Abb. 5**. Dieses Programm erlaubt es, ein Bild mit der Maus zu malen. Immer wenn der Knopf der Maus gedrückt ist, wird eine Linie der Maus folgend gezeichnet. Es werden die Grafik-ROM-Routinen „LINETO" und „MOVETO" benutzt, die im 4. Teil der Serie näher erläutert werden.

4.5. Sonstige Funktionen

Bei den sonst noch vorhandenen Funktionen handelt es sich zunächst um zwei Funktionen für die Programmierung in Maschinensprache:

PEEK (A) – Inhalt des Bytes mit der Adresse A. A muß im Bereich 0 bis 16777215 (2 ↑ 24 - 1) liegen.
VARPTR (Variablen-Name) – die Adresse des ersten Bytes der Variablen mit dem angegebenen Namen. Dies wird vor allem beim Aufruf von Maschinensprache-Programmen benötigt. Beispiele zu diesem Befehl folgen in dem Abschnitt über die ROM-Routinen in Teil 4, aber auch in diesem Teil in Abb. 9.

Für die Fehlerbehandlung gibt es zwei sehr nützliche Funktionen:

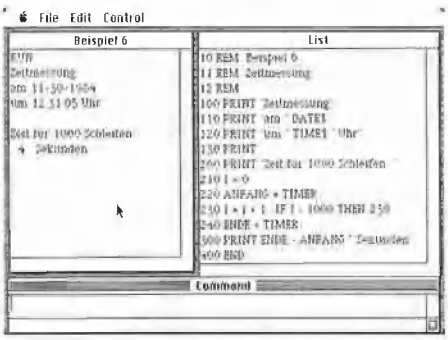
ERR – gibt nach einem Fehler die Nummer des aufgetretenen Fehlers an. Die Fehlernummern sind im Anhang des Basic-Handbuchs aufgeführt.
ERL – die Nummer der Programmzeile, in der der Fehler auftrat.

Diese beiden Funktionen sind in einer Fehlerbehandlungsroutine sehr nützlich. Eine solche Routine läßt sich wie beim Applesoft-Basic durch den Befehl „ON ERR GOTO ..." programmieren (wird weiter unten erklärt).
Als Zufallszahlengenerator steht die Funktion „RND (I)" zur Verfügung. Je nach dem Wert des Integer-Parameters „I" wird eine andere Zufallszahl erzeugt:

RND (I) mit I *negativ* – beginnt eine neue Reihe von Zufallszahlen. Für jedes negative I wird eine eigene Zufallszahlenreihe begonnen.
RND oder **RND (I)** mit *positivem* I – die nächste Zufallszahl der Reihe.
RND (0) – liefert die zuletzt erzeugte Zufallszahl noch einmal.

Die erzeugten Zufallszahlen sind von doppelter Genauigkeit aus dem Bereich von 0 bis 1.
Daneben gibt es noch einige Systemfunktionen:

FRE (-1) – gibt die Zahl der freien Bytes auf der Halde des Macintosh an.
FRE (N) mit N > = 0 – die Zahl der freien Bytes im Basic-Speicher.
FRE ("") – hat dieselbe Bedeutung wie z. B. „FRE (0)".



Dank der eingebauten Batterie verfügt der Macintosh über eine fortlaufende Uhr mit Kalender. Die nächsten Funktionen benutzen diese Uhr:

DATE$ – liefert einen String, der das aktuelle Datum darstellt. Der String hat das

Format "mm-dd-yyyy", gibt das Datum also in amerikanischer Notation an.
TIME$ – gibt einen String mit der augenblicklichen Tageszeit zurück. Der String hat das Format "hh:mm:ss" (Stunde, Minute, Sekunde).
TIMER – gibt die Zahl der Sekunden an, die seit Mitternacht verstrichen sind. Dies läßt sich z. B. zur Zeitmessung verwenden, wie **Abb. 6** zeigt.

5. PROGRAMMSTEUERBEFEHLE

Hier werden solche Befehle vorgestellt, die den normalen Programmablauf verändern. Es gibt zwei verschiedene Gruppen von Programmsteuerbefehlen: solche, die den Programmablauf innerhalb *eines* Programms steuern, und solche, die von einem Programm aus ein anderes Programm oder das Basic-System aufrufen.

5.1. Steuerung in einem Programm

Diese Befehle sind fast alle auch beim Applesoft-Basic vorhanden. Es gibt noch einige neue Befehle, die das strukturierte Programmieren, das manche vielleicht von Pascal gewöhnt sind, etwas besser unterstützen.

FOR I=A TO E (STEP S) – die normale FOR-Schleife. Als Laufvariable (hier I) kann jeder numerische Typ verwendet werden (anders als beim Applesoft-Basic, wo keine Integervariable I% erlaubt ist).
NEXT – bezeichnet das Ende einer FOR-Schleife. Wie beim Applesoft-Basic können der oder die Namen der Laufvariablen angegeben werden.
GOTO nnn bzw. **GOSUB nnn** – der normale GOTO- bzw. GOSUB-Befehl. Es gibt leider auch hier noch keine symbolischen Adressen.
RETURN – bezeichnet das Ende eines Unterprogramms, das mit „GOSUB" aufgerufen wurde. Anders als beim Applesoft-Basic kann nach dem „RETURN"-Befehl eine Zeilennummer angegeben werden, zu der gesprungen werden soll, jedenfalls steht es so im Handbuch. In Wirklichkeit wird dadurch ein „SYNTAX ERROR" erzeugt. Durch diesen Befehl könnte der Applesoft-Befehl „POP" simuliert werden, der beim Microsoft-Basic nicht mehr vorhanden ist, z. B. so:

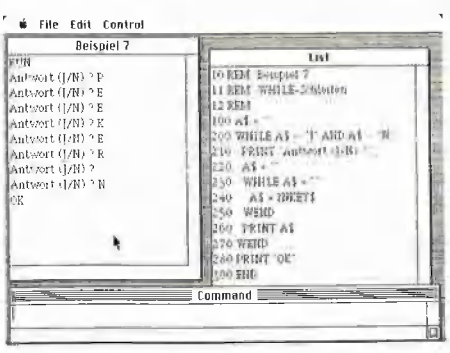
```
9600 POP
9610 .... (Applesoft)
wird zu
9600 RETURN 9610
9610 .... (Macintosh)
```

Wahrscheinlich (oder besser: hoffentlich) wird dieser Fehler bei der in Kürze erscheinenden neuen Version des Microsoft-Basic behoben sein.

ON I GOTO n1,n2,... und
ON I GOSUB n1,n2,... – berechneter Sprung bzw. berechneter Aufruf. Diese Befehle haben dieselbe Wirkung wie beim Applesoft-Basic.

IF X THEN ... ELSE ... – der IF-Befehl kann, wie hier gezeigt, noch einen ELSE-Teil besitzen. Dieser Teil wird dann ausgeführt, wenn die Bedingung (hier X) nicht erfüllt ist (also wenn der Ausdruck X den Wert 0 = falsch hat). Anstelle des Wortes „THEN" kann auch das Wort „GOTO" eingesetzt werden. Falls mehrere IF-THEN-ELSE-Befehle verschachtelt werden, bezieht sich ein „ELSE" immer auf das nächstgelegene „THEN" (wie beim UCSD-Pascal), z. B. erzeugt der Ausdruck IF A > B THEN IF B > C THEN ?"X" ELSE ?"Y"
die folgende Ausgabe:
falls A > B > C dann „X",
falls A > C > = B dann „Y",
falls B > = A dann keine Ausgabe.
Wie in Pascal gibt es beim Microsoft-Basic außer der FOR-NEXT-Schleife noch einen anderen Schleifenbefehl:

WHILE X: ... : WEND – entspricht der WHILE-Schleife bei Pascal. Die Befehle zwischen „WHILE" und „WEND" werden solange ausgeführt, wie die Bedingung X erfüllt ist (also einen Wert < > 0 liefert). Diese Schleifen können verschachtelt werden. Auch hier gilt: jedes „WEND" bezieht sich auf das nächstgelegene „WHILE", wie man in **Abb. 7** sieht.



Wie beim Applesoft-Basic bietet auch das Microsoft-Basic des Macintosh die Möglichkeit der Fehlerbehandlung:

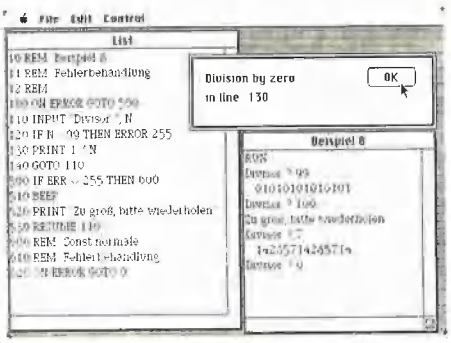
ON ERROR GOTO nnn – hat dieselbe Wirkung wie der Applesoft-Befehl „ONERR GOTO nnn". Nachdem diese Zeile ausgeführt wurde, wird bei einem auftretenden Fehler keine Fehlermeldung ausgegeben, sondern zu der angegebenen Zeilennummer gesprungen.

In der Fehlerbehandlungsroutine kann dann die Art des Fehlers und die Zeile, in der er aufgetreten ist, durch die Funktionen „ERR" und „ERL" bestimmt werden (s.o.). Beim Applesoft-Basic mußte man noch ein „PEEK (222)" ausführen, um wenigstens an die Fehlernummer heranzukommen.
Die Wirkung eines „ON ERROR GOTO"-Befehls kann man wie folgt aufheben:

ON ERROR GOTO 0 – bewirkt, daß bei einem Fehler wieder eine Fehlermeldung ausgegeben und das Programm abgebrochen wird.

Zur Beendigung einer Fehlerbehandlungsroutine gibt es den Befehl „RESUME". Dieser Befehl existiert in mehreren Varianten:

RESUME oder **RESUME 0** – beginnt den weiteren Programmablauf bei dem Befehl, der den Fehler ausgelöst hat (wie beim Applesoft-Basic).
RESUME NEXT – beginnt mit dem Befehl, der direkt *nach* dem Befehl folgt, welcher den Fehler ausgelöst hat.
RESUME nnn – setzt die Programmausführung bei der Zeile mit der Nummer nnn fort.



Mit diesen Befehlen läßt sich eine sehr komfortable Fehlerbehandlung programmieren. Zum Testen einer solchen Routine (und natürlich für weitere Anwendungen) gibt es noch einen Befehl:

ERROR I – simuliert das Auftreten des Fehlers mit der Nummer I. I kann die Werte 1 bis 255 annehmen. Da beim Microsoft-Basic längst nicht alle dieser Fehlernummern verwendet werden, lassen sich

auf diese Art eigene Fehler definieren. Ein Beispiel für den „ERROR"-Befehl gibt **Abb. 8**.

5.2. Wechsel von Programmen

Zu dieser Gruppe gehören Befehle, die von einem Programm aus ein anderes Programm aufrufen, aber auch solche Befehle, die ein Programm vom Basic-System aus starten oder ein Programm anhalten.

RUN nnn – startet das im Speicher befindliche Programm bei der angegebenen Zeile. Falls keine Zeilennummer angegeben ist, beginnt das Programm mit der ersten Programmzeile.
RUN X$,R – lädt das Programm mit dem Namen X$ von der Diskette ein und startet es. Wenn man kein ",R" an den Dateinamen anhängt, dann werden alle geöffneten Dateien geschlossen, bevor das neue Programm geladen wird. Beispiele:

RUN "Pit's Disk: Test" startet das Programm, das auf der Diskette "Pit's Disk" unter dem Namen "Test" abgespeichert ist. Dabei werden alle noch offenen Dateien geschlossen.
RUN "FileIO.3",R startet das Programm mit dem Namen "FileIO.3" und läßt dabei alle bisher geöffneten Dateien offen.

CONT,
END und
STOP – haben dieselbe Bedeutung wie beim Applesoft-Basic.

Dieselbe Bedeutung wie der Befehl „RUN X$,R" hat auch ein anderer Befehl:

LOAD X$,R – wie „RUN X$,R".
**LOAD X$** – lädt das Programm mit Namen X$ in den Speicher, das Programm wird aber nicht ausgeführt. Man befindet sich danach im Basic-Betriebssytem.
LOAD ohne Dateinamen – lädt ein Programm, dessen Name der Benutzer in einem Dialogfenster festlegen kann (vgl. Abb. 3 im 1. Teil der Serie).
**MERGE X$** – mischt das im Speicher befindliche Programm mit einem Programm von der Diskette. Das Programm mit dem Namen X$ muß im ASCII-Format auf der Diskette gespeichert worden sein (vgl. 1. Teil). Das „Mischen" geht so vor sich: Alle Zeilen des neuen Programms werden zu dem Programmtext des alten Programms im Speicher hinzugefügt. Dabei werden Zeilen des alten Programms, die dieselbe Zeilennummer haben wie Zeilen des neuen Programms, durch die neuen Zeilen überschrieben.

Durch diesen Befehl werden die Variablen gelöscht. Nach dem Befehl läuft ein Programm auch nicht weiter, sondern man befindet sich dann wieder im Basic-Betriebssystem. Mit dem nächsten Befehl kann man das aber vermeiden:

CHAIN X$,nnn bzw.
CHAIN MERGE X$,nnn – lädt ein neues Programm von der Diskette und startet es. Dabei können Variablen des alten Programms an das neue Programm übergeben werden. Alle offenen Dateien bleiben geöffnet.

Mit dem Zusatz „MERGE" wird das neue Programm, das in diesem Fall eine ASCII-Datei sein muß, mit dem alten Programm vermischt, wie es oben bei dem Befehl „MERGE" beschrieben wurde.
Man kann mit „nnn" eine Zeilennummer angeben, bei der das neue Programm gestartet werden soll. Falls nnn weggelassen wird, beginnt das neue Programm mit der ersten Zeile.
An den „CHAIN"-Befehl kann man noch bis zu zwei Zusätze anfügen:

,ALL – bedeutet, daß alle Variablen an das neue Programm übergeben werden. Falls dieser Zusatz nicht vorhanden ist, muß über einen besonderen Basic-Befehl („COMMON", s. 5. Teil) angegeben werden, welche Variablen übergeben werden sollen.
,DELETE n1-n2 – bedeutet, daß vor dem Laden bzw. Einfügen des neuen Programms die Zeilen mit den Nummern von n1 bis n2 gelöscht werden. Dies kann dazu benutzt werden, um Platz für ein neues Unterprogramm zu schaffen, das mit „CHAIN MERGE" in einen bestimmten Zeilenbereich (z.B. n1-n2) geladen werden soll.

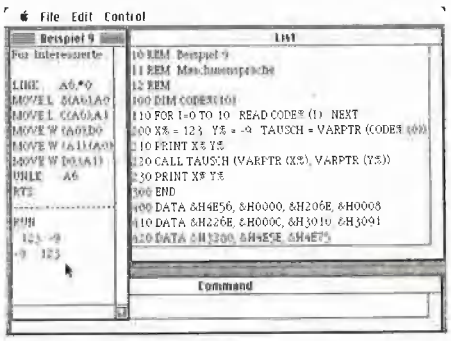
Wenn einer dieser Zusätze benutzt wird, dann muß das Komma zwischen „X$" und „nnn" (s. Syntax oben) auch dann angegeben werden, wenn gar keine Anfangszeilennummer angegeben ist:

CHAIN "NextProgram",ALL ist falsch,
CHAIN "NextProgram",,ALL ist richtig!

Beispiele für den „CHAIN"-Befehl:

CHAIN "Prog.3",1000,ALL – lädt das Programm "Prog.3" in den Speicher und startet es bei Zeile 1000. Es werden alle Variablen des alten Programms an das neue Programm übergeben.
CHAIN MERGE "Proc.New",,ALL,DELETE 5000-9999 – löscht bei dem im Speicher befindlichen Programm die Zeilen 5000 bis 9999 und fügt dann das Unterprogramm "Proc.New" zu dem alten Programm hinzu. Alle Variablen bleiben erhalten.
Nach diesen doch relativ komplizierten Befehlen gibt es noch zwei einfachere Steuerbefehle:

SYSTEM – beendet das Basic-System und kehrt zum normalen Macintosh-Betriebssystem zurück.
CALL Variablenname – ruft eine in Maschinensprache geschriebene Routine auf. Als Variablenname kann der Name einer der ROM-Routinen verwendet werden, die im Teil 4 beschrieben werden. Ansonsten ist „Variablenname" der Name einer numerischen Variable, die die Anfangsadresse der Maschinensprache-Routine angibt.



Hinter diesem Befehl kann in Klammern („(",,)") eine Liste von Argumenten angegeben werden, die vor dem Aufruf der Routine auf den Stack geschoben werden. Auf diese Art lassen sich Parameter an die Maschinensprache-Routine übergeben.
Ein Beispiel dazu zeigt **Abb. 9**. Das Maschinensprache-Programm wird aus den DATA-Zeilen in das Array „CODE%" eingelesen. Es vertauscht den Inhalt von zwei Integer-Variablen. Hier sieht man auch die Benutzung der Funktion „VARPTR" (s.o.). Sie liefert zum einen die Anfangsadresse des Maschinenprogramms, zum anderen die Adressen der beiden Variablen. Falls jemand an dem Maschinen-„Prögrämmchen" selbst interessiert ist, ist im Ausgabefenster der Abb. 9 das Programm in mnemonischer (lesbarer) Form angegeben; die Ausgabe des aufgelisteten Programms steht unter dem Strich.


POWER
Tx DATA
Rx DATA
CARRIER
ON LINE
1200Rx-75Tx
VIEWTEXT
300/300
75Rx-1200Tx
EXTERNAL
CONTROL
ON LINE
LOCAL TEST
CCITT
300 FULL
DUPLEX
ORIG
ANS
BELL
1200
EQU
1200
1200
1200
EQU
ANS
600
300 FULL
ORIG
V23
TEST
MODE
MIRACLE
Technology
Modem WS 2000

Das Modem WS 2000 ermöglicht über das Telefonnetz die Daten-Kommunikation zwischen zwei Computern an beliebigen Orten der Welt. Durch neuentwickelte, hochintegrierte Schaltungen kann es sowohl Signale nach den Standards der CCITT als auch von BELL verarbeiten. Damit sind praktisch alle Länder der Erde abgedeckt.

Weltweite Datenübertragung mit dem WS 2000

von Matthias Pohl

Ein Universalmodem für alle Normen

In einem schwarzen ABS-Kunststoffgehäuse vereint, bietet das World Standard Modem WS 2000 alle gebräuchlichen Übertragungs-Standards und -Geschwindigkeiten an und macht seinem Besitzer Ferndatenbanken, Bulletin Boards und BTX zugänglich.
Das Modem wird mit dem Rechner über eine V24-Schnittstelle verbunden, die bei Apple, außer im Modell IIc, nicht serienmäßig vorhanden ist und deshalb extra gekauft werden muß. Die Verbindung des Modems mit dem Telefon erfolgt über eine auf der Rückseite des Gerätes herausgeführte Telefon-Zuleitung, die bedauerlicherweise nur der englischen Postnorm entspricht. Daher ist es fast unumgänglich, den Stecker abzutrennen und die angeschlossenen Zuleitungen einzeln mit der Telefonleitung oder dem zwischengeschalteten Akustikkoppler zu verbinden.
Auf der Frontseite des WS 2000 (s. **Bild**) sind die Bedienungselemente übersichtlich angeordnet. Der größere der drei Drehschalter dient zur Wahl der Betriebsart. Hier stehen folgende Möglichkeiten zur Verfügung:

USA Bell 103, 300 Baud, Voll Duplex Empfangen
USA Bell 103, 300 Baud, Voll Duplex Senden
USA Bell 202, 1200 Baud, Halb Duplex
CCITT V23, 1200 Baud, Halb Duplex
CCITT V23, 600 Baud, Halb Duplex
CCITT V23, 300 Baud, Voll Duplex Empfangen
CCITT V23, 300 Baud, Voll Duplex Senden

Beide 1200-Baud-Betriebsarten besitzen eine zusätzliche, mit EQU bezeichnete Stellung, durch die ein Entzerrer-Filter aktiviert wird. Allerdings sollte dieses nur bei gestörten Telefonleitungen benutzt werden, da es sonst seinerseits wieder zu Störungen führen kann.
Mit dem oberen der beiden kleinen Drehschalter wird festgelegt, welchem der beiden Übertragungs-Kanäle die höhere Baud-Rate zugeordnet werden soll. Für BTX und die verwandten Dienste Prestel bzw. Micronet muß dieser Umschalter in der Position 1200 RX - 75 TX stehen, d.h. Daten werden mit 1200 Baud empfangen, während gleichzeitig mit 75 Baud gesendet wird.
Der dritte Drehschalter schließlich verfügt über zwei Stellungen: ON LINE und LOCAL. In der ON LINE-Stellung ist das Modem mit der Telefonleitung verbunden. In der LOCAL-Stellung werden die vom Computer kommenden Signale bis zum Ausgang des Modems verarbeitet und gleich wieder durch das Modem zurückgeführt. Die Stellung LOCAL wird also immer dann zu wählen sein, wenn keine Verbindung aufgebaut ist. Außerdem kann hier ein komplettes Selbsttest-Programm aktiviert werden.
Zur Anzeige und Überwachung des Betriebszustandes dienen fünf Leuchtdioden. Eine mit POWER bezeichnete Diode zeigt an, daß das Modem eingeschaltet ist und mit Strom versorgt wird. TX DATA leuchtet, wenn das Modem Daten vom Computer erhält und aussendet, RX Data leuchtet, wenn Daten empfangen werden, und CARRIER weist auf das Vorhandensein der Trägerfrequenz hin. ON LINE schließlich zeigt an, daß das Modem auf die Telefonleitung geschaltet ist.
Wie nun gestaltet sich die Benutzung des Modems? Zuerst muß das Modem mit der V24-Schnittstelle des Computers verbunden werden. Sehr bewährt hat sich hier als Interface die Super Serial Card von Apple, die über einen speziellen Kommunikations-Modus verfügt. Dieser bietet eine Vielzahl von Möglichkeiten zur Einstellung des Datenformats und der Datenübertragungsgeschwindigkeit. Die Baud-Raten von Modem und Interface müssen selbstverständlich übereinstimmen. Für die Benutzung von BTX und verwandten Diensten bedarf es umfangreicher Software. Sollen hingegen Mailboxen oder Bulletin Boards in Anspruch genommen werden, so reicht die Kommunikations-Software

der Super Serial Card vollständig aus. Diese wird durch die Tastenfolge IN#n (n = Slot, in der die SSC eingesteckt ist) Ctrl-A T aktiviert. Als nächstes ist über Telefon oder mit der Zusatzplatine AD-2 über den Computer der gewünschte Rechner anzuwählen. Hat dies Erfolg und ist die Trägerfrequenz als Pfeifton zu hören, so muß nur noch das Modem ON LINE geschaltet werden; der weitere Ablauf der Kommunikation hängt von der Art des angerufenen Systems ab.

Die an sich schon sehr umfangreichen Möglichkeiten des WS 2000 können durch getrennt erhältliches Zubehör noch wesentlich erweitert werden. Als wichtigste sind an dieser Stelle die Auto Dial/Auto Answer-Platinen AA-2 und AD-2 für automatisches Wählen und automatische Anrufbeantwortung zu nennen. Alle Betriebsart-Umschaltungen und die Zusatzplatinen sind nach dem Einsatz weiterer IC's auch direkt vom Rechner aus über ein Steuerkabel zu bedienen. Die Anschlußbelegung entspricht allerdings dem ACORN B Computer.

Das Modem WS 2000 besitzt z. Zt. *keine* allgemeine fernmeldetechnische Zulassung (FTZ-Nummer) und teilt dieses Schicksal mit praktisch allen erhältlichen Modems. Die Deutsche Bundespost gestattet deshalb keinen Anschluß des WS 2000 an das Telefonnetz.

Fazit: Das WS 2000 stellt aufgrund seiner vielfältigen Betriebsarten ein universell einsetzbares Modem dar. Die Verwendung mit einem Apple-Rechner ist unter Einsatz der Super Serial Card unproblematisch. Das erhältliche Zubehör erlaubt einen den individuellen Erfordernissen entsprechenden Ausbau der Leistungsmerkmale des Modems.

Bezugsquelle: Miracle Technology (UK) Ltd., Ipswich IP1 1XB. Importeur: Claus F. Erbrecht, Computer Related Products, Lappenbergsallee 37, 2000 Hamburg 20, Tel. 040/8505255. WS 2000 DM 798,-, Zusatzplatinen je DM 199,50. Weiteres Zubehör wie Kabel, Interface, Software usw. ist erhältlich, ein Akustikkoppler AC-1 in Vorbereitung.

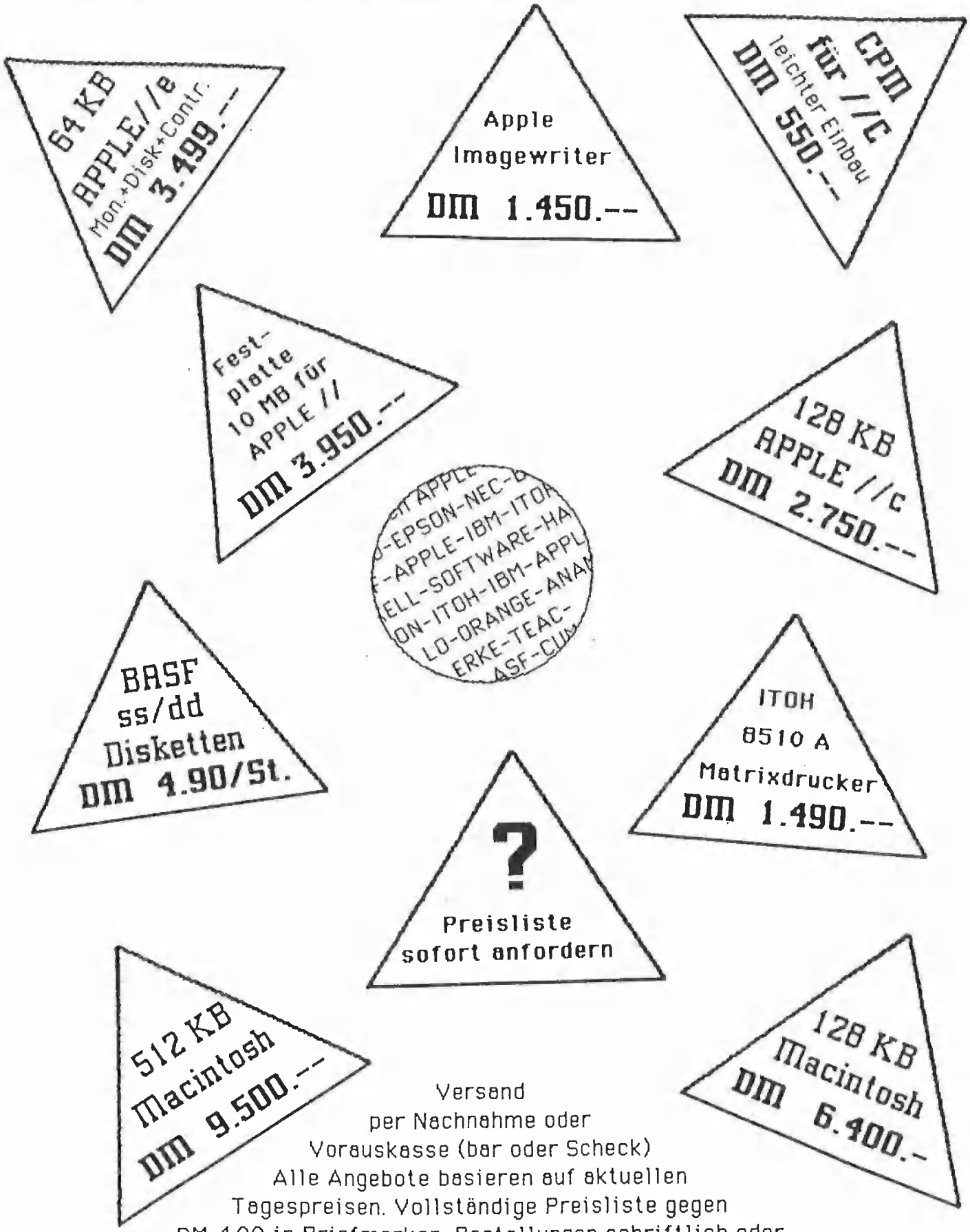
Applesoft-Editor-Macros

Ein Quickie für den PRODOS.EDITOR

Der speziell für das Betriebssystem ProDOS gedachte Applesoft-Editor namens PRODOS.EDITOR (Hüthig Software Service) läßt u. a. die Tasten Q, W, E, R, T und Z als frei definierbare Tastatur-Macrobefehle mit einer Länge von je 31 Zeichen zu. Da der Editor jedoch in einem von ProDOS geschützten Speicherbereich liegt, können selbstdefinierte Macros nicht direkt mit BSAVE und BLOAD gespeichert bzw. geladen werden. Deshalb wurde das folgende Applesoft-Programm PRODOS.-EDITOR.MACROS entwickelt, mit dem beliebig viele Macrotabellen angelegt werden können. Damit bei Macros Ctrl-Zeichen wie etwa Return (Ctrl-M) zulässig sind, muß man die Definition eines Macros entweder mit der Del-Taste (Apple IIe/IIc) oder mit Ctrl-Null (Apple II Plus) abschließen. Diese zwei Tasten sind die einzigen Ctrl-Zeichen, die nicht Bestandteil eines Macros (und im übrigen auch nicht eines Strings) sein können. Ctrl-Zeichen werden am Bildschirm invers angezeigt. Da auch Links- und Rechtspfeil zu den zulässigen Ctrl-Zeichen gehören, ist ein Editieren der Macrozeile nicht möglich.

Die Unterroutine in dem Applesoft-Programm Zeilen 350–430 ist auch für andere Programme verwendbar, bei denen in einen String beliebige Ctrl-Zeichen eingegeben werden sollen. Man beachte jedoch, daß die Routine nicht mit der 80-Zeichenkarte des Apple IIe funktioniert, da diese z. B. kein ESC als Ctrl-Zeichen zuläßt. us

```
100  TEXT : HOME : CLEAR : DIM M$(6), S$(6), L(6)
110  INVERSE : PRINT "PRODOS.EDITOR-MACROS": NORMAL
120  PRINT : PRINT "1 MACROS EINLESEN"
130  PRINT : PRINT "2 MACROS ANLEGEN"
140  PRINT : PRINT "3 ENDE"
150  PRINT
160  GET X$: ON X$ <  > "1" AND X$ <  > "2" AND X$ <  >
     "3" GOTO 160: IF X$ = "3" THEN  END
170  REM *** MACROS EINLESEN ***
180  IF X$ = "1" THEN  HOME : PRINT : PRINT  CHR$
     (4)"CAT": INPUT "NAME: ";N$: PRINT  CHR$ (4)
     "RESTORE";N$: GOSUB 290: END
190  REM *** MACROS ANLEGEN ***
200  HOME : PRINT "DEL = ENDE MACRO-EINGABE"
210  DATA Q,W,E,R,T,Z
220  RESTORE : FOR Y = 1 TO 6: READ M$(Y): NEXT
230  FOR Y = 1 TO 6: PRINT : INVERSE : PRINT M$(Y);:
     NORMAL : PRINT ": ";: GOSUB 360:S$(Y) = A$:
     L(Y) = L: NEXT
240  PRINT : PRINT "OKAY J/N ";
250  GET X$: IF X$ = "N" THEN 100
260  IF X$ <  > "J" THEN 250
270  HOME : GOSUB 290: PRINT  CHR$ (4)"CAT": INPUT "NAME:
     ";N$: PRINT  CHR$ (4)"STORE";N$: END
280  REM *** MACROS POKEN ***
290  FOR Y = 1 TO 6
300  H = 36182 + (Y - 1) * 32
310  POKE H,L(Y)
320  FOR X = 1 TO L(Y):H = H + 1: POKE H, ASC ( MID$
     (S$(Y),X,1)) + 128: NEXT X
330  NEXT Y
340  RETURN
350  REM *** INPUT MACROS ***
360  L = 1:A$ = ""
370  GET X$: ON X$ <  > "" GOTO 380: ON L = 1 GOTO 360:L
     = L - 1: GOTO 430
380  A =  ASC (X$): ON A <  > 127 GOTO 390: ON L = 1 GOTO
     360:L = L - 1: GOTO 430
390  A$ = A$ + X$
400  IF A < 32 THEN  INVERSE : PRINT  CHR$ (A + 64);:
     NORMAL : GOTO 420
410  PRINT X$;
420  IF L < 31 THEN L = L + 1: GOTO 370
430  PRINT " "L: RETURN
```



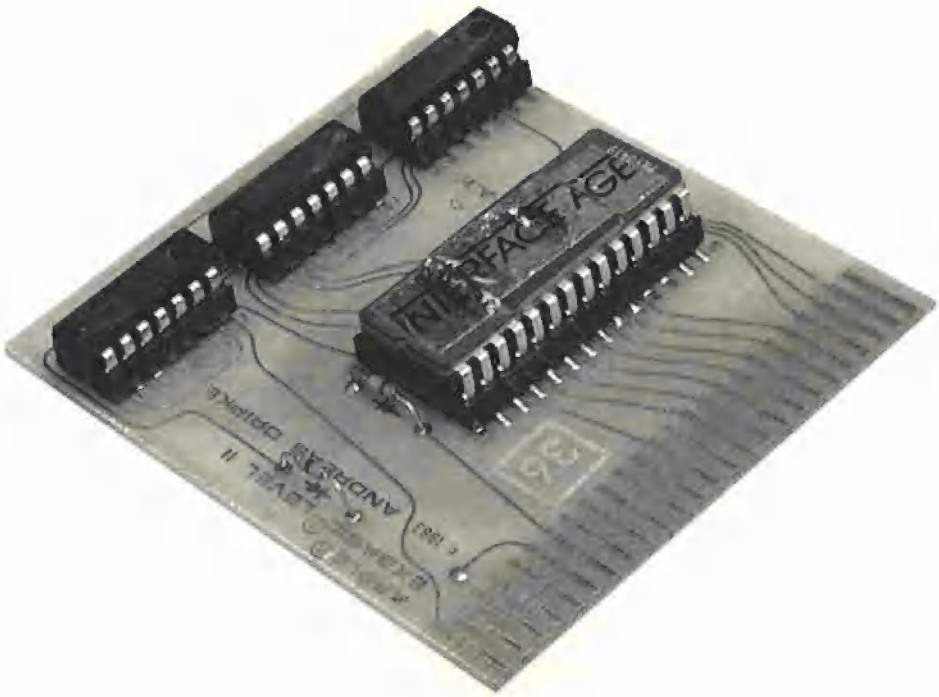
Versand
per Nachnahme oder
Vorauskasse (bar oder Scheck)
Alle Angebote basieren auf aktuellen
Tagespreisen. Vollständige Preisliste gegen
DM 4.00 in Briefmarken. Bestellungen schriftlich oder
telefonisch. Lieferung unfrei, ab Lager innerhalb 24 Stunden.

>>>>>> pyramid versand koslik <<<<<<<<

>>>>>>> 7800 freiburg steinhalde 109 <<<<<<<

>>>>>>>>>>>>>> telefon 0761 / 63675 <<<<<<<<<<<<<<<<<<<<

Exba



Eigentlich wollte ich einen nüchternen Bericht schreiben, aber jetzt wird es doch ein ganz persönlicher. Den Anstoß zu diesem Umschwung gab ein Test über Exbasic Level II im neuen Magazin Apple's (CW-Publikationen Verlag). Tester Andreas Dripke stimmte mich dort so positiv ein, daß ich nicht umhin konnte, einmal selber alles ganz genau nachzuprüfen.
„So etwas wie eine Verpackung scheint der Hersteller nicht zu kennen, das Ganze kommt reichlich lose, aber immerhin bruchsicher eingewickelt ins Haus. Hier hat man offensichtlich der Güte des Produkts die Präferenz gegenüber bunter Verpackung gegeben." schreibt Tester Dripke. Ersteres konnte ich bestätigen, letzteres nur hoffen.

Die Handbücher

Exbasic Level II wurde 1981 für Commodore-Rechner entwickelt und 1983 dann für den Apple angepaßt. Diese Tradition schlägt sich in zwei Handbüchern nieder, die Sie geliefert bekommen. Ein dickes (ca. 110 Seiten) für Commodore und ein Nachtrag (ca. 25 Seiten) für Apple. *„Die Bücher sind gut zu lesen"* schreibt Tester Dripke. Motiviert stürzte ich mich in das Apple-Addendum, das mir bald sagte, ich solle jetzt einige Seiten im Commodore-Heft weiterlesen. So getan, gelangte ich bald wieder in den Nachtrag zurück, wo mir dann erklärt wurde, was alles von dem eben Gelesenen bei Apple nicht gilt und wie es dort statt dessen heißen muß. Das Spielchen ging einige Male hin und her, vor und zurück. Wahrhaft gut zu lesen! Für fast 400,- DM sollte der Käufer eigentlich erwarten können, eine vollständige Anleitung zu erhalten. Aber messen wir Exbasic Level II lieber an seinen Taten als an seiner Beschreibung.

Die Platine

Tester Dripke ermutigte mich: *„Der Einsatz der Platine in den Computer ist problemlos. Allerdings: Im Unterschied zu der Mehrzahl der Apple-Karten müssen die Bauteile von vorne gesehen nach links zeigen."* Also frisch ans Werk, Netzteil aus, den Deckel ab und die Karte in den empfohlenen Slot 2. Die erste Überraschung: Sie paßt nicht! Das EPROM 2764 mit Sockel ist gewaltig dick und macht den Bausteinen auf meinem Druckerinterface in Slot 1 den Platz streitig, so daß beide Karten nicht gleichzeitig im Gerät stecken können. Die Apple-Slots stehen nun einmal sehr eng, und nur wenn sich alle Hersteller an das ungeschriebene Gesetz halten, ihre Karten nach rechts zu orientieren, wird es zwar eng, aber es geht. Exbasic Level II ist meines Wissens die erste Karte, die nach links zeigt. Ihr Hersteller sollte schleunigst den Aufbau ändern, was bei nur 6 Bauteilen nicht unlösbar sein dürfte.

Die Versionen

Exbasic Level II gibt es in zwei Versionen für den II Plus und den IIe. Diese Tatsache ist weder im Handbuch erwähnt noch auf der Karte selbst vermerkt. Die Unterschiede liegen ausschließlich in der Software, die in der Version für den IIe auch die Cursortasten, DEL und TAB auswertet sowie den 80-Zeichen Bildschirm unterstützt. Bei der Initialisierung der Karte wird der Zustand des Rechners ebenso überprüft wie das Vorhandensein von Steckkarten. Scheitern diese Tests, weil z.B. ein Apple II Plus vorliegt, aber eine IIe-Karte verwendet wird, stürzt das System ab. Eine Fehlermeldung wäre freundlicher gewesen. Also achten Sie auf die richtige Karte! Für den Apple IIc ist eine Diskettenversion in Vorbereitung.

Das Software-Modul

Exbasic Level II ist in 6502-Assembler geschrieben und belegt den Adreßbereich von $D000 bis $EFFF parallel zum Applesoft-Interpreter. Damit es hier keine Probleme gibt, werden bei der Initialisierung entsprechende Treiberroutinen nach

sic Level II im Test

Eine Basic-Erweiterung bringt Sie ins Staunen, nicht nur positiv.

von Dr. Jürgen B. Kehrel

Den Versuch, das antiquierte Applesoft-Basic zu erweitern, haben schon viele unternommen. Ganze Sammlungen von nachladbaren Routinen sind dabei entstanden, die zumeist über den Ampersand-Vektor & an Applesoft angekoppelt werden. Exbasic Level II geht hier einen anderen Weg. Es ist ständig auf einer kleinen Karte vorhanden, die Sie bei Bedarf mit einem POKE aktivieren können.

$300-$3CF geschrieben. Dieser Speicherbereich steht Ihnen also nicht mehr zur Verfügung, z.B. für eigene kleine Assemblerprogramme. Die gesamte Software ist in einem 8K-EPROM abgelegt und mit 3 weiteren integrierten Schaltungen auf der kleinen Platine untergebracht. Die Typenbezeichnungen dieser Bausteine sind heruntergekratzt, eine um sich greifende Unsitte bei Herstellern, um Nachbauten zu erschweren. Ein Schaltbild erhalten Sie selbstredend dann auch nicht. Wenn einmal etwas kaputt gehen sollte, sind Sie immer auf den Service des Händlers oder des Herstellers angewiesen.

Der Full-Screen-Editor

Der Apple ist nur mit einem Minimaleditor ausgestattet, der von Exbasic Level II zu einer vollwertigen Programmierumgebung ausgebaut wird. Der neue Editor erlaubt es, Programmzeilen zu ändern, ohne die ganze Zeile mit dem Cursor überstreichen zu müssen. Bei einem RETURN wird die ganze Zeile so übernommen, wie sie auf dem Bildschirm steht, unabhängig von der Position des Cursors. EINFÜGEN und LÖSCHEN werden über ESC und Pfeiltasten (II Plus) oder über DEL und TAB (IIe) gesteuert. Das funktioniert auch im Direktmodus, wenn Sie z. B. aus einem CATALOG ein Programm aufrufen wollen. Sie schreiben nur RUN in der richtigen Höhe und drücken Return. Der Rest der Zeile wird mit übernommen. Die Cursortasten rollen Ihr gesamtes Programm, das nach oben oder unten über den Bildschirm geschoben wird, ohne daß Sie auch nur einmal LIST sagen mußten. Ausgewachsene Editoren bringen sicherlich noch etwas mehr, der Exbasic-Editor ist aber für die meisten Fälle schon voll ausreichend.

Die Hilfsfunktionen

Die Hilfsfunktionen unterstützen die Arbeit des Editors. Es sind im einzelnen:
HELP Zeigt alle Befehlsworte von Exbasic Level II an.
MEM Zeigt die Größe und Belegung des Apple-Speichers an.
MATRIX/DUMP Gibt alle einfachen und alle Feldvariablen aus.
MATRIXDIM Zeigt die Dimensionierung aller Feldvariablen.
SYSTEM Identisch mit CALL -151.
BASIC Schaltet Exbasic Level II ab.
MERGE Fügt ein Programm von Diskette oder Kassette in oder an ein im Speicher stehendes Programm.
AUTO Automatische Zeilennummerngenerierung mit wahlweisem Anfang und bestimmbarer Schrittweite.
RENUM Numeriert ein Programm neu, leider immer nur ganz.
FIND Sucht nach einem Befehl oder Text im Speicher. Listet entsprechende Zeilen.
Die angenehmste Überraschung war für mich die Funktion **ON/OFF**. Beim Austesten von Programmen wird nicht nur „SYNTAX ERROR" ausgegeben, sondern statt dessen die ganze Zeile gelistet und der Cursor über die fehlerhafte Stelle gesetzt, die so in Windeseile zu berichtigen ist, da auch der Editor natürlich sofort aktiv ist. OFF schaltet zum normalen Applemodus zurück. Das Exbasic **TRACE** zeigt nicht nur die Zeilennummern wie Applesoft, sondern listet die entsprechende Zeile und stellt einen Leuchtzeiger über den gerade bearbeiteten Befehl. Mit SPEED= 100 oder ähnlich können Sie dann genüßlich Ihrem Programm „auf die Finger sehen".

Mathematische Funktionen

MIN/MAX Sucht in einer Liste von Zahlen oder Variablen den kleinsten/größten Wert.
ODD Prüfung auf gerade oder ungerade Zahl.
HEX$/DEC Umrechnung Hexadezimal/Dezimal.
RND Erzeugt ganzzahlige Zufallszahlen > = 1.
ROUND Rundet Dezimalzahlen auf eine vorgegebene Zahl von Nachkommastellen.
FRAC Gibt den Nachkommateil einer Zahl aus.

Bildschirmbefehle

SCREEN Definiert das Bildschirmfenster. Ersetzt POKE 32, 33, ...
DELLINE Löscht eine Zeile, rückt den Rest von unten nach.
INSTLINE Fügt eine Zeile ein, schiebt den Rest nach unten.
BEGINLINE Löscht vom Cursor zum Zeilenanfang.
ENDLINE Löscht vom Cursor zum Zeilenende = CALL -868.
BEGINPAGE Löscht vom Cursor zum Seitenanfang.
ENDPAGE Löscht vom Cursor zum Seitenende = CALL -958.
SCREEN UP/DOWN Schiebt den ganzen Bildschirm eine Zeile hoch oder runter.

Basicbefehle

IF.THEN.ELSE Ermöglicht Mehrfachverzweigungen für den Fall, daß die IF-Bedingung nicht zutrifft.
CLEAR Array Löscht ein Variablenfeld mit Inhalt und Dimensionierung.

DEEK/DOKE Doppelbyte-PEEK bzw. -POKE, um Zahlen bis 65535 in einem Schritt in zwei aufeinanderfolgenden Speicherstellen zu bearbeiten.
EVAL Wertet im Gegensatz zu VAL auch einen numerischen Ausdruck aus.
GOTO/GOSUB Erlauben auch errechnete Sprünge, die aber jedes Programm sehr unübersichtlich machen.
RESTORE Erlaubt mittels Parameter, den DATA-Zeiger auf eine ganz bestimmte Zeile zu setzen, als ON .. RESTORE sogar mit Parameterliste.
RESUME/.NEXT Springt nach einem Fehler die angegebene oder die nächste Zeile an.
SEC SEC X hält das Programm X Sekunden an.
POP NEXT Aussprung aus einer Schleife mit Korrektur des Stacks.
POP CLEAR Löschen des ganzen Stacks.
INPUTFORM Verbessertes INPUT, das auch „:" akzeptiert. Der Editor ist bei der Eingabe aktiv.
DEF USR=/DEF PERFORM Dienen zum Einbinden von Maschinenprogrammen, indem der USR-Vektor festgelegt wird bzw. die Adresse eines Programmes, das dann mit PERFORM aufgerufen werden kann. Eine Sammlung solcher Programme (SOFTMODULE) ist in Vorbereitung.
INSTR Instring-Routine, um einen String in einem anderen zu suchen.
SWAP Tauscht zwei Variablen gleichen Typs gegeneinander aus.
STRING$/SPACE Gibt ein Zeichen sooft aus, wie es dem Parameter entspricht bzw. bis der angegebene Raum gefüllt ist.
PRINT USING/USING$ Formatierte Text- und Zahlenausgabe.
VARPTR Ermittelt die Adresse einer Variablen im Speicher.
PRINT AT Positioniert den Cursor auf einer der 960 möglichen Bildschirmpositionen (nur 40 Z/Z).
HARDCOPY Gibt den Bildschirm auf den Drucker aus.
MUSIC Erzeugt Töne von angegebener Dauer und Tonhöhe.

Wie arbeitet Exbasic

Im Prinzip arbeitet Exbasic Level II mit der sogenannten CHRGET-Manipulation. Kurz gesagt, es wird dabei jede Eingabe erst einmal daraufhin überprüft, ob sie ein Exbasic-Wort enthält. Wenn ja, arbeiten die Exbasic-Routinen, andernfalls wird das Kommando an Applesoft weitergegeben. Dieser Umweg bedingt, daß auch reine Applesoftprogramme langsamer laufen (bis 10%), wenn Exbasic aktiv ist. Applesoft speichert seine Befehlsworte platzsparend in einem Byte (= Token) ab; nicht so Exbasic. Hier steht das volle Wort im Speicher und frißt den durch die Mächtigkeit der Befehle gewonnenen Raum wieder auf. Ein Exbasic-Programm ist weder compilierbar noch läuft es auf einem Rechner ohne die Erweiterungskarte. Auch unter ProDOS gelang es mir nicht, Exbasic zum Leben zu erwecken, da es versucht, einige dort nicht mehr vorhandene Speicherstellen zu lesen. INTEGER-BASIC dürfen Sie mit INT nicht aufrufen, da sich das System dann „aufhängt". Auch wer gewohnt ist, mit FP den Speicher zu löschen, muß sich umstellen, da ebenfalls ein Systemabsturz erfolgt, aus dem Sie selbst RESET nicht befreit. Entsprechende Hinweise fehlen leider im Handbuch.

Gesamteindruck

Mein Gesamturteil bleibt zwiespältig: Einer Reihe von wirklich sinnvollen neuen Befehlen und einem recht guten Editor stehen die „verbaute" Platine und ein unübersichtliches Handbuchgespann gegenüber. Wenn der Hersteller sich zu einer Neuauflage von Karte und Anleitung durchringt, kann ich Ihnen Exbasic Level II empfehlen, im jetzigen Zustand leider nur sehr eingeschränkt.

Lassen wir doch noch einmal Tester Dripke zu Wort kommen: *„Fazit: Mit Exbasic erhält man eine solide, technisch ausgereifte Basic-Erweiterung, die ihre 392 Mark wert ist."*

Bezugsquelle: Exbasic Level II, Versionen für Apple II Plus, IIe (IIc in Vorbereitung) und Basis 108, Preis DM 392,- inkl. Mwst., Hersteller: Unternehmensberatung Andreas Dripke, Wiesbaden, Vertrieb: INTERFACE AGE Verlag GmbH, Josefsburgstr. 6, 8000 München 80

Erscheinungs- und Anzeigenschlußtermine für **peeker**

Ausgabe	Erscheinungstermin	Anzeigenschluß
3	25. 2. 85	25. 1. 85
4	25. 3. 85	22. 2. 85
5	22. 4. 85	22. 3. 85
6	20. 5. 85	19. 4. 85
7	24. 6. 85	24. 5. 85
8	22. 7. 85	21. 6. 85
9	26. 8. 85	26. 7. 85
10	23. 9. 85	23. 8. 85
11	21. 10. 85	20. 9. 85
12	25. 11. 85	25. 10. 85

Hinweise für Autoren

Wir suchen laufend Beiträge: Erfahrungsberichte über kommerzielle Hardware- und Software-Produkte, Tips und Tricks für Anwender und Programmierer, gut dokumentierte Programme für Anfänger und Fortgeschrittene, hardwaretechnische Beiträge usw. Wir zahlen DM 200,- pro Druckseite, gleichviel ob Text oder Programm, wobei ein Beitrag mehrere Druckseiten oder gar mehrere Aufsatzfolgen umfassen kann. Es lohnt sich also, für uns zu schreiben.
Alle Textbeiträge einschließlich der Programmlistings werden von einem Apple IIe direkt in die Lichtsatzanlage unserer Druckerei übertragen. Im einzelnen sind deshalb folgende Richtlinien zu beachten:

Beiträge müssen sowohl als Papierausdruck wie auch als Diskette eingereicht werden. Zur Zeit können ausschließlich die klassischen 51/4 Zoll Disketten für Apple II/IIe/IIc angenommen werden. Berücksichtigt werden grundsätzlich nur Originalaufsätze und Originalprogramme. Es ist jedoch urheberrechtlich zulässig, aus Werken anderer Urheber zu zitieren. So können Sie z.B. Teile aus einem fremden Programm in Ihr eigenes übernehmen. In diesem Fall muß jedoch an geeigneter Stelle ein Zitathinweis erfolgen, etwa in der Form „Zeilen X-Y enthalten den Teil Soundso aus dem Programm ABC des Autors X, erschienen bei Firma Y, Ort und Jahr".
Wie im Verlagswesen üblich, übertragen Sie uns mit den eingereichten Aufsätzen und/oder Programmen das Recht der Vervielfältigung und Verbreitung in jeder Form. Speziell in bezug auf die Programme besagt dies, daß sie von uns nicht nur in Druckform, sondern auch in Diskettenform vertrieben werden, und zwar als Peeker-Sammeldisketten, die in unregelmäßiger Folge erscheinen und eine Auswahl der größeren Programme als Quell- und Objekt-Code enthalten. Versehen Sie Ihre Programme nicht mit dem amerikanischen Copyright-Vermerk, sondern schreiben Sie statt dessen nur „Von Name, Ort und Jahr".

1. Textbeiträge

a) Die Aufsatztexte sollten im Idealfall mit einem DOS-3.3-Textprogramm erstellt werden, das normale ASCII-Textfiles erzeugt, z.B. Applewriter II/IIe usw. Wordstar-Dateien im CP/M-Format sind ebenfalls möglich, doch müssen in diesem Fall die Texte endlos (ohne Silbentrennung) erfaßt werden.
b) Die Texte dürfen keinerlei Steuerzeichen für Matrixdrucker usw. enthalten. Überschriften dürfen nicht unterstrichen werden. Vielmehr brauchen lediglich im ausgedruckten Manuskript die Überschriften sowie darüber hinaus im Text selbst Hervorhebungen mit einem Farbstift o.ä. markiert zu werden. Die entsprechende Kodierung für die Satzanlage wird dann von uns übernommen.
c) Der Text muß grundsätzlich endlos erfaßt werden, d.h. ein Return darf nur am Absatzende und niemals am Zeilenende stehen, da der Satzrechner sonst keinen automatischen Umbruch mehr vornehmen kann. Vermeiden Sie darüber hinaus Einzüge zu Beginn eines Absatzes. Auch dies besorgt nämlich die Satzanlage automatisch.
d) Ausgangspunkt unserer Kodierung ist der deutsche ASCII-Zeichensatz, also mit Umlauten, ß usw. anstelle von eckigen und geschweiften Klammern. Wenn Sie – z.B. beim alten Apple II – keine Umlaute usw. eingeben können, rufen Sie uns an. Bei Texten, die *sowohl* die deutschen *als auch* die amerikanischen ASCII-Sonderzeichen enthalten sollen, z.B. bei Pascal-Aufsätzen, müssen die letzteren kodiert werden. Kodierung bedeutet, daß eine Buchstabenkombination, z.B. „§p" für griechisches π durch unser Transmitter-Programm in den entsprechenden Kode der Satzanlage konvertiert wird.
e) Vermeiden Sie unnötige Einrückungen und Kurztabellen. Ferner mischen Sie bitte nicht Aufsatztexte mit Programmtexten, da letztere gesondert in die Satzanlage übertragen werden. Größere Tabellen, die von uns konventionell gesetzt werden, sind auf gesonderten Textfiles zu speichern. Es genügt, wenn Sie senkrechte Trennlinien mit Kuli o.ä. andeuten. Schaubilder usw. sind stets willkommen. Hierzu genügen Bleistiftskizzen als Vorlagen für unsere Reinzeichner.
f) Achten Sie bitte auf Orthographie und Interpunktion. Deutsch ist immer noch eine wichtigere Sprache als Basic und Assembler!

2. Programmlistings

Beachten Sie, daß wir im Gegensatz zu anderen Computer-Zeitschriften Programmlistings nicht von den meist sehr häßlichen Printouts abphotographieren, sondern wie die Textbeiträge von Diskette in die Satzanlage übertragen, wobei eine besondere Schreibmaschinen-Lichtsatzschrift verwendet wird. Nur bei exotischen Programmiersprachen wählen wir z. Zt. noch den konventionellen Weg.

2.1. Basic-Programme

a) Applesoft-Programme usw. liefern Sie bitte als normale Programm-Files, die von uns für die Satzanlage aufbereitet werden.
b) Zeilennummern sollten stets dieselbe Stellenzahl haben, z.B. nur 3stellige Zeilennummern 100, 110, 120 o.ä., damit diese Zahlen von uns korrekt freigestellt werden können (RENUMBER anwenden).
c) REM's sollten der besseren Lesbarkeit halber möglichst in Groß-Klein-Buchstaben geschrieben werden.
d) Menü-Texte, REM's usw. müssen unbedingt in Deutsch sein. Englischsprachige Programme werden nicht angenommen. Englische Computer-Fachausdrücke sind demgegenüber selbstverständlich zulässig.

2.2. Assembler-Programme

a) Assembler-Programme sind als Source-Code und als Objekt-Code auf Diskette einzureichen. Wir bevorzugen den Assembler namens Merlin (= Big Mac) von G. Bredon. Verwendbar sind ferner Lisa 2.5, Toolkit-Assembler, ProDOS-Assembler sowie jeder andere Assembler, bei dem die Möglichkeit besteht, anstelle auf Papier auf Diskette zu assemblieren.
b) Der Source-Code sollte möglichst im 40-Z/Z-Modus geschrieben werden, damit die assemblierten Listings in der Zeitschrift 2spaltig gesetzt werden können. In der Kommentarspalte des Source-Codes sollten nur ganz kurze Bemerkungen stehen. Längere Kommentare schreibe man *über* die Befehle in reine Kommentarzeilen, wobei auch diese 40 Zeichen pro Zeile nicht überschreiten sollten.
c) Vermeiden Sie Makros und seltene Pseudo-Op-Codes, damit der Quell-Code ohne großen Aufwand in einen anderen Assembler konvertiert werden kann.

2.3. Sonstige Programme

Programme in selteneren Programmiersprachen sind willkommen, erfordern jedoch eine vorherige Absprache wegen der Konvertierungsmöglichkeit. Ggf. muß ein Schönschreibdrucker-Printout erstellt werden.

NEUE BÜCHER

Das Buch zum Apple II

von Erich Esders
1984, 210 S., geb., DM 54,-
ISBN 3-7723-7641-X
Franzis Verlag, München

Für den fortgeschrittenen Apple II Benutzter hat gute Literatur eher Seltenheitswert, besonders wenn sie auch noch in Deutsch geschrieben ist. Esders hat hier die Ergebnisse vieler Stunden am Rechner zu Papier gebracht. Applesoftkenntnisse und zumindest praktische Grundfertigkeiten in 6502-Assembler sollte der Leser schon mitbringen, um mit den vielen Einblicken in das Innenleben des Apple etwas anfangen zu können. Einer kurzen Einführung in den Prozessor und in den Speicheraufbau folgt als Hauptteil des Buches die Darstellung des Basic-Interpreters, der zu diesem Zweck teilweise disassembliert und kommentiert wurde. Alle wichtigen Einsprungstellen von Basic-Befehlen, Floating-Point-Routinen, Stringoperationen bis hin zur hochauflösenden Grafik sind benannt. In übersichtlicher Tabellenform wurden die Einsprungbedingungen (was muß wo stehen), die benutzten Register und Speicher sowie die Ausgabe dokumentiert. Ein Applikationsteil zeigt die Anwendung von Interpreter- und Monitorroutinen in eigenen Maschinenprogrammen mit jeweils einem ausgeführten Beispiel für jede Kategorie. Der Apple-Monitor kommt nur unter „ferner liefen" vor, dagegen aber auf 20 Seiten ein für die meisten Apple-Benutzer sicherlich nutzloser Interpretervergleich Apple versus Commodore 64.

Im Untertitel verweist das Buch auf den IIe, geht jedoch dann an keiner Stelle auf dessen Besonderheiten ein. Alle beschriebenen „Softswitches" z.B. gelten nur für den IIPlus. Der IIe hat bedeutend mehr davon und unterscheidet auch zwischen Lese- und Schreibbefehlen im Bereich von $C000 an aufwärts. Die 80-Zeichenkarte des IIe ersetzt eine Reihe von Monitorroutinen, so daß besonders Ein- und Ausgabeoperationen mit 80 Z/Z anders verlaufen als beschrieben. Beide Rechner verfügen aber über den gleichen Interpreter. Bleibt eigentlich nur noch zu sagen, daß das Buch sauber und fehlerfrei gesetzt ist, die Matrixdrucker-Listings gut lesbar sind und daß alles in einem festen Einband steckt, es rundum also eine (fast) runde Sache ist.

Programmierung des 6502

von Rodnay Zaks
2. Aufl. 1982, 368 S., kart., DM 44,-
ISBN 3-88745-011-6
SYBEX-Verlag, Düsseldorf

Dieses Buch ist als Einführung in die Assemblerprogrammierung gedacht. Behutsam und verständlich führt R. Zaks zuerst in die verschiedenen Zahlensysteme und den internen Aufbau des Prozessors ein, bevor die einzelnen Befehle des 6502, gut mit Grafiken dokumentiert, auf rund 80 Seiten erläutert werden. Ihr Gebrauch wird an vielen Beispielen geübt, die alle nach einem klaren Schema aufgebaut und z.T. mit einem Flußdiagramm versehen sind. Der Schwierigkeitsgrad steigert sich langsam und paßt sich den wachsenden Kenntnissen des Lesers an. Binäre Datenbäume werden ebenso abgehandelt wie Hashing, Bubble-Sort und Mischen. In einem letzten Teil werden Computersysteme mit 6502 Prozessor und Grundzüge von Assemblern vorgestellt.
R. Zaks wendet sich nicht an einen speziellen Rechner. Die Programme funktionieren auch auf einem Apple. Für den reinen Apple-Programmierer ist das Buch aber gerade deshalb keine erschöpfende Einführung, da alle Besonderheiten der Apple-Architektur nicht berücksichtigt wurden. Das betrifft insbesondere die Ein- und Ausgaberoutinen. Für ein vollständiges Programm ist es aber erforderlich, direkt formatiert auf den Bildschirm schreiben sowie Disketten lesen und schreiben zu können. Das alles wird nicht behandelt, von den vielen Besonderheiten der Apple-Grafik ganz zu schweigen. Über kurz oder lang wird sich der Assembler-Neuling also noch ein weiteres Buch zulegen müssen, das direkt für den Apple geschrieben wurde. Dabei werden ihm die Grundkenntnisse aber von Nutzen sein, die er in diesem Buch erhalten hat.

6502 Anwendungen

von Rodnay Zaks
2. Aufl. 1983, 288 S., kart., DM 38,-
ISBN 3-88745-014-0
SYBEX-Verlag, Düsseldorf

Laut Werbung soll dieses Buch umfassend die Ein- und Ausgabe Ihres 6502 beschreiben. In dieser Hinsicht werden Ihre Erwartungen nur dann nicht enttäuscht, wenn Sie dieses sehr wörtlich nehmen. Bildschirmausgabe, Diskettenverwaltung oder Grafik kommen in diesem Buch nicht vor. Ausführlich wird dagegen die Steuerung einer Zusatzkarte mit 6522 Prozessor als Ein- und Ausgabeeinheit behandelt. Die Bauanleitung dazu befindet sich im Buch. Daneben werden 6520, 6530 und 6532 vorgestellt. Mit dem Erweiterungsbaustein ist es möglich, kleine Programme und Anwendungen wie Relais, Morseprogramm, Sirenenton, Impulsmessung, Verkehrsampel, Alarmanlage oder eine einfache A/D-Wandlung (Thermometer) zu üben. Das Beispiel des automatischen Telefonwählers kann in Deutschland nicht funktionieren, da die Deutsche Bundespost ein ganz anderes Wählverfahren benutzt. Da ist der Hinweis des Übersetzers, das beschriebene Programm lasse sich „derzeit" noch nicht einsetzen, ganz schön geschmeichelt.
Das Buch ist didaktisch gut aufgebaut und sauber hergestellt, es enthält aber auch eine Menge „Luft". So sind alle größeren Listings zweimal abgedruckt: einmal im laufenden Text und ein zweites Mal im Anhang. Auf 15 Seiten werden antiquierte Systeme wie KIM-1, SYM-1, AIM65 oder PET vorgestellt, und weitere 19 Seiten nimmt die Beschreibung und das Listing für einen Minimal-Assembler ein, geschrieben in HP2000F TSS Basic.
Wer sich für die speziellen Anwendungen interessiert oder die Programmierung von I/O-Bausteinen erlernen möchte, für den lohnt sich die Anschaffung dennoch.

Apple II ROM Listing

von Matthias Buck
1984, 116 S., kart., DM 59,-
Röckrath Microcomputer

Apple selber hat nie den Quellcode zum Applesoft-Interpreter veröffentlicht, so daß unabhängige Autoren sich an seine Entschlüsselung gemacht haben. Begonnen hat alles 1980 mit J. Crossley, es folgten komplette Versionen von

M.A. Carpick, Bob Sander-Cederlof, Randy Hyde und Glen Bredon. M. Buck legt jetzt die erste deutsche Fassung vor, d.h. mit deutschsprachigen Kommentaren. Von $D000 bis $F7FF ist alles dokumentiert. Die Anmerkungen sind zahlreicher als z.B. bei Bredon, bringen jedoch keine Tatsachen ans Licht, die nicht schon in den englischen Versionen erwähnt wären. Aufgeführt sind jeweils die Adresse, Assembler-Quellcode und Kommentar, der Objectcode fehlt dagegen bei Buck. Übersichtstabellen und Kurzeinführungen in wichtige Interpreterroutinen runden den Inhalt ab.
Das ganze Buch ist auf einem Matrixdrucker erstellt und im Copy-Schnelldruck vervielfältigt, worunter die Lesbarkeit und Übersichtlichkeit sehr leidet. Es macht insgesamt einen „billigen" Eindruck, der nicht zu einem Preis von DM 59,- paßt. Der Inhalt ist eigentlich solide genug, um in einer besseren äußeren Ausstattung präsentiert zu werden. Wer das ROM-Listing noch nicht in Englisch hat, für den ist Bucks Buch eine Bereicherung, wenn auch eine teure.

Fortgeschrittene 6502 Programmierung

von Rodnay Zaks
1984, 288 S., kart., DM 42,-
ISBN 3-88745-047-7
SYBEX-Verlag, Düsseldorf

Trüge dieses Buch den Titel „6502 Anwendungen Teil 2", so würden meiner Meinung nach Umschlag und Inhalt besser zusammenpassen. Auf dem Umschlag heißt es vielversprechend: „Bald können Sie den geeigneten Algorithmus auswählen und ihn schrittweise definieren, eine leistungsfähige Datenstruktur entwickeln und diese bewerten, Echtzeitlösungen realisieren ...". Gearbeitet wird dann aber praktisch nur mit einem „Hobbyboard" als Hardware, das vom Leser selbst gebaut werden muß und dessen Anschluß an das kleine Entwicklungssystem SYM-1 beschrieben ist. (Appleanschluß im Anhang). 15 LED's werden darauf angesteuert nebst einer Hexadezimaltastatur (Sedezimal für die Puristen) und einem Lautsprecher, mit dem Musik (Jingle Bells) gemacht wird. Pseudozufallszahlen werden erzeugt, die dann zu „visuellen Mustern" aus blinkenden LED's führen. Bei einer Realzeit-Simulation umkreisen Lichtsignale ein Quadrat aus LED's. Sie müssen Licht- und Tonsignale behalten und nachvollziehen. Schließlich wird als „Künstliche Intelligenz" (S. 207) ein simples TIC-TAC-TOE-Spiel vorgestellt, bei dem gewinnt, wer als erster eine waagerechte, senkrechte oder diagonale Dreierreihe in einem Quadrat mit 3 · 3 Feldern zustande bringt.
Sicherlich werden einige interessante Programmiertechniken behandelt, deren Kenntnis allgemein nützlich ist, die praktische Umsetzung erfolgt jedoch an Trivialitäten. Mit einem Home-Computer oder einem Apple IIe kann man aber bei weitem mehr machen, als hier als Fortschritt verkauft wird. Rodnay Zaks ist in der Lage, auch schwierige Sachverhalte nachvollziehbar und einprägsam zu erklären. Er sollte sein Talent nicht an allzu simplen Spielchen vergeuden.

Macintosh Benutzerhandbuch

von Carol Kaehler
1983, 174 S., zahlr. Abb., Spiralbindung
Ins Deutsche übersetzt von Apple Computer, München
(Wird beim Kauf des Macintosh mitgeliefert)

Dieses Handbuch erläutert in 4 Kapiteln die Benutzung des Macintosh:
1. Macintosh kennenlernen
2. Mehr über den Macintosh erfahren
3. Mit dem Finder arbeiten
4. Nachschlageteil
Die Typografie und die farbigen Abbildungen sind – wie bei allen Apple-Handbüchern – ganz ausgezeichnet, während der Text selbst teilweise völlig unverständlich ist, weil nunmehr die „Ikonen" (= Verbildlichungen abstrakter Begriffe) durch Hypostasen (= Verdinglichungen abstrakter Begriffe) ersetzt werden, was zu grotesken semantischen Konstruktionen führt, wie man sie sonst nur vom magischen Denken der Urmenschen her kennt. Einige Beispiele mit Seitenangaben:
S. 21: „Wählen Sie eine Information aus, dann suchen Sie eine Handlung für sie aus": Können Informationen handeln? Vgl. hierzu S. 60: „Setzen Sie den Zeiger dorthin, wo die Handlung stattfinden soll" = Zeigen Sie auf den Handlungsort (?).
S. 36: „Die Abbilder wurden dadurch ausgewählt, indem Sie in diese geklickt haben": Klein-Mecki in der Vernissage. Fragt der Aussteller: „Schon ein Bild ausgewählt?" – „Nee, hab' noch in keins geklickt" – „Dacht' ich mir".
S. 40: „Verschieben Sie das neue Abbild von dem Ordnerabbild, das Sie dupliziert haben": Einst gab es einen konkreten, physisch greifbaren (Leitz-)Ordner. Daraus wurde eine Mac-Entität namens „Ordner". Dann kam das Abbild des „Ordners" und schließlich das Abbild des Abbildes des „Ordners", das dann „verschoben" wurde. Ontologisch gesprochen, gab es einst ein „Seiendes", dann kam die „Seinsheit" und schließlich die „Seinsheitlichkeit".
S. 40: „Jedesmal, wenn Text ausgewählt ist, wird dieser Text durch den Text, den Sie schreiben, ersetzt.": Wieviele Texte gibt es hier?
S. 41: „Dokumente und Anwenderprogramme können Sie entweder auf Ihrem Schreibtisch lassen oder sie in Ordner oder auf Disketten unterbringen": Denksportaufgabe: Nennen Sie bei diesem Satz die konkreten Begriffe sowie die Verdinglichungen abstrakter Begriffe!
S. 43: „Sie können auch einen Befehl aussuchen und mit ihm auf eine Gruppe ausgewählter Abbilder einwirken": Was ist gemeint?
S. 65: „Verschieben Sie über den Text bis zum Ende des ausgewählten Textes": „Verschieben" ist ein transitives Verb, erfordert also ein Akkusativobjekt. Wenn man an diesem unglücklich gewählten Begriff festhalten will, so muß es heißen: etwas über etwas schieben, etwas an etwas entlang schieben, etwas über etwas hinweg schieben usw. *und nicht* (S. 160) „Zum Auswählen einer Information wird der Zeiger über sie verschoben" *und auch nicht* (S. 163) „Wenn durch das Menü verschoben wird, während eine Menüfunktion hervorgehoben ist, wird diese ausgesucht". Klein-Mecki im Restaurant. Fragt der Ober: „Schon ein Menü gewählt?" – „Nee, hab' noch durch keins verschoben" – „Dacht' ich mir".
S. 95: „Sie können durch gleichzeitiges und dem Punkt den Druckvorgang abbrechen": Wirre Mecki-Sprache? Wer klärt mich auf?
Sprechen Sie „Mecki"? Wenn ja, Frage: Wie nennt man „die Information, die durch den nächsten Befehl beeinflußt wird" (Lösung auf S. 160)?
Der Philosoph Schopenhauer schrieb einmal über Hegel: „Das Publikum war genötigt worden einzusehen, daß das Dunkle nicht immer sinnlos ist. Sogleich flüchtete sich das Sinnlose hinter den dunklen Vortrag. Jedoch die größte Frechheit im Auftischen baren Unsinns, im Zusammenschmieren sinnleerer, rasender Wortgeflechte, wie man sie bislang nur in Tollhäusern vernommen hatte, trat endlich in Hegel auf." Es hätte mich interessiert, was Schopenhauer über den Autor/Übersetzer des Macintosh-Handbuches geschrieben hätte.
us

Hüthig-FACHBUCH-TIP

„Apple Assembler" wendet sich an alle, die bereits Anfängerkenntnisse der 6502-Programmierung haben und nunmehr ein Nachschlagewerk für ihren Apple II Plus/IIe/IIc suchen, in dem alle wichtigen ROM-Routinen sowie eine Vielzahl sonstiger Hilfsprogramme in einer systematischen Form zusammengestellt werden. Insgesamt umfaßt dieses Buch über 40 Utilities, darunter mehrere völlig neuartige Programme wie Double-Lores, Double-Hires, Screen-Format u. a.

Der erste Teil enthält ein Repetitorium der wichtigsten Befehle, Adressierungsarten und sonstigen Besonderheiten des 6502 sowie Angaben zu den apple-spezifischen Zahlenformaten (Integer- und Fließkommazahlen). Im zweiten Teil werden neben einer Kurzwiederholung der Monitor-Befehle alle Routinen und Adressen des Monitors zusammengestellt, die für Assemblerprogrammierer von Nutzen sein können. Darüber hinaus findet der Leser Unterroutinen für Vorwärts- und Rückwärts-Moven, hexadezimale Addition/Subtraktion/Multiplikation/Division, Binär-, Hex-und ASCII-Umwandlung, Dumpen/Disassemblieren, Aufwärts-Scrollen, Reset u. a.
Der dritte Teil befaßt sich mit der Speicherverwaltung der Language-Card und der IIe-64K-Karte und enthält Test- und Move-Programme zum Verschieben von Daten in die und aus der Language Card sowie der 64K-Karte, wobei besonders ausführlich auf die Softswitches eingegangen wird.

Der vierte Teil ist dem Applesoft-ROM gewidmet und beschreibt die interne Struktur von Applesoft-Programmen, die Methoden der Parameterübergabe mittels CALL, USR, &, PEEK und POKE und listet dann eine große Anzahl nützlicher Interpreter-Adressen. Bei den Utility-Programm liegt das Schwergewicht auf Fließkommamathematik einschließlich Print Using.

Der letzte Teil behandelt den Text- und Grafikspeicher. Neben einem professionellen Maskengeneratorprogramm findet der Leser hier auch erstmals Routinen zur Double-Lores- und Double-Hires-Grafik des Apple IIe.

Apple Assembler - Tips und Tricks -

von U. Stiehl

232 S., 40 Programm-Listings, 3 Abb., kart., DM 34,—
ISBN 3-7785-1047-9

Begleitdiskette zum Buch DM 28,—
ISBN 3-7785-1048-7

BESTELLCOUPON

Bitte ausfüllen und an Hüthig Vertriebsservice, Postfach 102869 · 6900 Heidelberg schicken.

Buchtitel

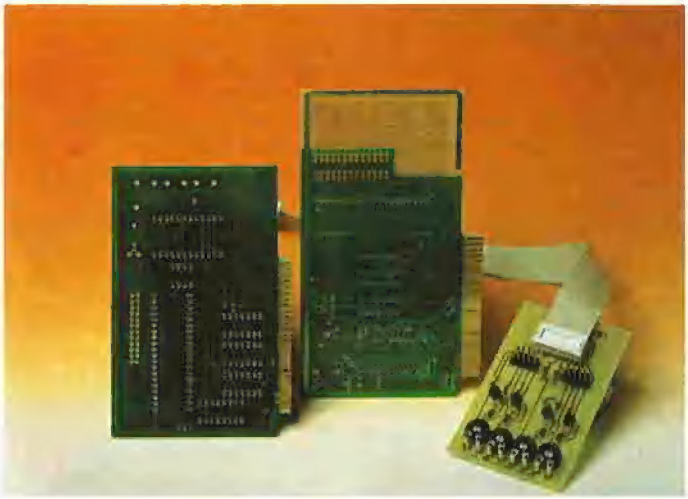
Name

Straße

Ort

Unterschrift

Doppel-Parallelinterface mit RAM oder EPROM



Die Firma Tombstone-Micro stellt zwei Versionen dieser Doppel-Parallelschnittstelle her. Es handelt sich dabei um eine Steckkarte für Applecomputer, die ganz nach den Anwenderbedürfnissen aufgebaut werden kann.
Die erste Version des Interface trägt den Namen ADD 2 und enthält einen VIA 6522 (Versatile Interface-Adapter) mit 2 mal 8-Bit-Input/Output plus 2 Handshakeleitungen. Daneben existieren ein 8-Bit-Schieberegister zur seriellen Datenübertragung und 2 Interrupt-Timer, die auch als Zähler verwendet werden können.
Das Interface ADD 4, die zweite Version, ist mit einem PIA 6520 (periphere Interface-Adapter) bestückt. Dieser enthält nur die 2 mal 8-Bit-Input/Output plus 2 Handshakeleitungen.
Beide Karten können mit einem 2716 EPROM oder 6116 RAM bestückt werden. Der Speicherbereich ist dann folgender:

$0000 – $03FF Expansion
Memory $C800 – $CBFF
$0400 – $06FF nicht adressierbar
$0700 – $07FF Slotbereich
$Cs00 – $CsFF
(In der provisorischen Betriebsanleitung waren die obigen Angaben falsch, was inzwischen korrigiert wurde.)

Es ist ebenfalls Platz für einen Akku, der das RAM für mindestens eine Woche mit Strom versorgt. Leider kann die Karte dann nicht aus dem Apple genommen werden, denn es besteht die Möglichkeit, daß die Daten zerstört werden.
An beiden Schnittstellen können entweder Drucker, Tastaturen, EPROM Burner oder ähnliche Geräte angeschlossen werden. Es sind auch 4 Dip-Schalter vorhanden, mit denen das Treiberprogramm gegen Schreibzugriffe geschützt werden kann. Die Stromversorgung des Akku kann auch mit einem dieser Schalter unterbrochen werden.
Die ADD 2 wird mit einer DOS 3.3 Diskette geliefert. Auf ihr sind Testprogramme in Applesoft und Assembler vorhanden. Diese Programme fehlen bei der ADD 4, jedoch gibt es auf der Platine ein Wrap-Feld mit 229 Bohrungen für eigene Schaltungen.
Zur Programmierung des VIA wird im Begleitmaterial, das nach unserer Meinung zu kurz gefaßt ist, auf das Buch „Mostech 65xx Prozessorreihe" hingewiesen. Hier soll nur eine grobe Übersicht über den PIA gegeben werden.
Der PIA 6520 (6821) enthält zwei 8-Bit-Ports. Port A dient gewöhnlich als Eingangsport und Port B zur Ausgabe. Jeder Port enthält:
– Ein Daten- oder Peripherieregister, das entweder Eingangs- oder Ausgangsdaten aufbewahrt.
– Ein Datenrichtungsregister. Die Bits in diesem Register bestimmen, ob die entsprechenden Datenregisterbits (und Anschlüsse) Eingänge (0) oder Ausgänge (1) sind.
– Ein Steuerregister, welches die Statussignale aufbewahrt, die für den Quittierungsbetrieb erforderlich sind, sowie andere Bits, die die Logikverbindungen innerhalb des PIA auswählen.
– Zwei Steuerleitungen, die durch die Steuerregister konfiguriert werden. Diese Leitungen können für die Quittierungssignale verwendet werden.
Die Bedeutung der Bits im Datenrichtungsregister und den Steuerregistern bezieht sich auf die entsprechende Hardware und ist völlig willkürlich, soweit es den Programmierer für die Assemblersprache betrifft.
Jeder PIA belegt vier Speicheradressen:
Basis = C080 + n0, n = Slotnummer
Basis + 0 ORA/DDRA Portregister A
Basis + 1 CRA Controllregister A
Basis + 2 ORB/DDRB Portregister B
Basis + 3 CRB Controllregister B
Die RS (Register Select = Registerauswahl)-Leitungen wählen eines der vier Register aus. Da es in jedem PIA sechs Register gibt (zwei periphere, zwei Datenrichtungsregister und zwei Steuerregister), ist ein weiteres Bit für die Adressierung erforderlich. Bit 2 jedes Steuerregisters bestimmt, ob sich die andere Adresse auf dieser Seite auf das Datenrichtungsregister (0) oder das Peripherieregister (1) bezieht. Die Preise für das voll bestückte Modell mit Akku sind (Die Abbildung zeigt die Leerplatinen):
ADD 2 mit Diskette und Beschreibung DM 180,-
ADD 4 mit Beschreibung DM 150,-
Bezugsquelle: Firma Tombstone-Micro, Gardeschützenweg 72, 1000 Berlin 45, Tel. 030/8331303

Modula-2 für den Macintosh

Modula-2, das neueste Kind des Pascal-Erfinders Nikolaus Wirth, ist nun auch für den Macintosh verfügbar. Modula-2 ist eine konsequente Weiterentwicklung von Pascal und bietet neben den bekannten Pascalfunktionen folgende Neuerungen:
– Ein Modulkonzept, das es ermöglicht, auf bequeme Weise ein Programm in kleinere Einzelteile zu zerlegen, wobei zwischen der Schnittstellenbeschreibung und dem eigentlichen Programmteil unterschieden wird. Zusätzlich werden über Modulgrenzen Typprüfungen durchgeführt.
– Eine systematische Syntax, die das Erlernen der Sprache vereinfacht.
– Möglichkeiten für die systemnahe Programmierung (Adreßrechnung, Prozedurvariablen usw.).
– Dynamische Felder als Prozedurparameter (z.B. für Strings unterschiedlicher Länge).
Modula-2 auf dem Macintosh enthält den kompletten Sprachumfang. Darüber hinaus bietet es den Zugriff auf sämtliche Macintosh-Routinen wie Bildschirmmanipulation, Maus-Kontrolle, Grafikroutinen, Speicherplatzverwaltung, Ausgabe von Tönen, Dateizugriffe usw.
Der Übersetzer wird zusammen mit einem vollständigen Entwicklungssystem und einem englischen Manual für DM 598,50 inkl. MwSt angeboten.
Bezugsquelle: pl Gesellschaft für Informatik mbH, Gotthardstr. 99, 8000 München 21, Tel. 089/580609

64K-Karte für IIe

Über die Firmen Wagner Datentechnik in Uhldingen und ProSoft in Koblenz sind preiswerte erweiterte 80-Zeichenkarten für den Apple IIe erhältlich, die funktionsmäßig mit der Originalkarte von Apple identisch sind. Übrigens sind auch die deutschen Anleitungen der Firmen Wagner und ProSoft identisch, d.h. gleicher Textwortlaut im Innenteil bei verschiedenem Umschlag.

Bezugsquelle: Wagner Datentechnik, Im Öschle 21, 7772 Uhldingen-Mühlhofen 1

Staub und Schmutz auf Computer und Tastatur?

Computer und Peripheriegeräte stauben schnell zu. Gerade während der Nacht und des Wochenendes, wenn Ruhe im Büro herrscht, setzt der tagsüber aufgewirbelte Staub sich auf die wertvollen Bürogeräte nieder. Dabei ziehen die Kunststoffgehäuse und insbesondere die Bildschirme nach dem Abschalten den Staub wie Magnete an. Schutzhauben verhindern den Staubansatz. Die Staubschutzhauben sind aus kräftigem und reißfestem Material gefertigt. Es liegen Schutzhauben vor für den Apple II/II+, Apple IIc, Apple III und den Macintosh.
Bezugsquelle: Schonenberg Schutzhauben, Drosselweg 20, 4782 Erwitte, Tel. 02943/7613

TempoHexe „SpeeDemon" 65C02C-Prozessorkarte

Die TempoHexe („SpeeDemon") macht den Apple fast so schnell wie die populären 16-Bit-Rechner.
– Es ist absolut keine Software-Änderungen notwendig.
– Sie kann im Apple II,II+ und IIe verwendet werden, wobei die speziellen Features vom IIe unterstützt werden sollen.
– Funktionsprinzip: Neuer Mikroprozessor 65C02C mit 3.6 MHz tritt an die Stelle des 1.02 MHz Originalprozessors 6502.
Der „SpeeDemon" ist eine Konkurrenzkarte zur bislang teureren Accelerator IIe. Ob sie dasselbe leistet, wird ein Testbericht in einer der nächsten „Peeker"-Ausgaben zeigen.
Bezugsquelle: Softline Rut Alverdes, Schwarzwaldstraße 8 A, 7802 Oberkirch, Tel. 07802/3707

Lohn- und Gehaltsabrechnung

Dieses Programmpaket bietet eine moderne, ausgefeilte Methode, die Lohn- und Gehaltsabrechnung im Betrieb im Dialogverkehr mit dem Rechner ohne besondere Vorkenntnisse sowohl des Rechners als auch der kleinen Details der Gesetzgebung durchzuführen.
Dabei kann die monatliche Abrechnung getrennt nach Arbeitern und Angestellten erstellt werden, wie dies in manchen Branchen üblich ist. Das Programm ist ausgelegt für bis zu 70 Mitarbeiter bei Einsatz von zwei 5-Zoll-Laufwerken oder bis zu 200 Mitarbeiter bei Einsatz von zwei 8-Zoll-Laufwerken.
Im einzelnen werden folgende Daten erfaßt, die für die Brutto- und/oder Nettolohnabrechnung sowie für die Ausdrucke und Berichte erforderlich sind: Firmenstammdaten, Personalstammdaten, Daten von bis zu 5 Hausbanken und bis zu 50 Personalbanken, Daten des Finanzamtes.
Bezugsquelle: Orgasoft GmbH, Werner-von-Siemens-Straße 3, 7730 VS-Villingen, Tel. 07721/72223

Lotus – Jazz

„Jazz" ist ein multifunktionales Software-Paket mit fünf Anwendungsbereichen: Textverarbeitung, Erstellen von Arbeitsblättern, Datenbankverwaltung, Datenaustausch und Erstellen von Grafiken.
Die fünf Funktionen von „Jazz" sind miteinander integrierbar, so daß der Anwender Dokumente wie Analysen, Budgets, Memos, Empfehlungen und Berichte leicht erarbeiten kann.
Dem Paket beigefügt sind zwei Handbücher mit detaillierten Funktions- und Bedienungsanleitungen, „Jazz" wird in englischer Version ab März 1985 und in komplett deutscher Version ab Mitte 1985 zum Preis von DM 1.895,– ohne MwSt erhältlich sein.
„Jazz" läuft auf dem Macintosh 512K mit einem externen Disketten-Laufwerk.

Bezugsquelle: HHCC GmbH, Mainzer Landstraße 46, 6000 Frankfurt am Main 1, Tel. 069/720711

Branchenprogramm Reisebüro

Ziel dieses Programmpakets ist die Abwicklung der innerhalb eines Reisebüros anfallenden Arbeiten wie z. B. Kunden- und Adreßverwaltung, Fakturierung, Werbeschreiben-Erstellung, Preislistenausdruck und Änderung, Auswertungen der Kunden- und Reisedaten.
Das in Modularbauweise aufgebaute Programmpaket führt den Anwender im Bildschirmdialogverfahren durch alle notwendigen Arbeitsschritte. Bei der Datenerfassung schlägt das System definierte Eingaben vor, die zu bestätigen sind. Mit diesem Verfahren können Angebote, Anmeldebestätigungen und Rechnungen geschrieben werden. Darüber hinaus können mit der Kundenadresse spezifische Daten abgespeichert werden.
Im angegliederten Textverarbeitungsprogramm werden in Verbindung mit der Adreßverwaltung Reiseankündigungsbriefe, Werbeschreiben, Tagespost und Preislisten geschrieben. Das Programmpaket umfaßt auch ein Scheckschreibungsprogramm, mit dem die eingehenden Rechnungen schnell und problemlos beglichen werden können. Neben diesen Tagesroutinen können auch dispositive und statistische Auswertungen über Listenausdrucke vorgenommen werden. Diese Transaktionen, z. B. Kundenadreßlisten, lassen sich durch ein Kennwort vor unbefugtem Zugriff schützen.
Bezugsquelle: Orgasoft GmbH, Werner-von-Siemens-Straße 3, 7730 VS-Villingen, Tel. 07721/72223

UCSD p-System für den Macintosh

Das hannoversche Systemhaus FOCUS Computer bietet die Version IV.13 des UCSD p-Systems in Deutschland ab sofort für den Macintosh an. Damit ist zum ersten Mal ein vollständiges Software-Entwicklungssystem verfügbar, das auf dem Macintosh lauffähig ist.
Mit dem p-System sind Compiler für die Programmiersprachen UCSD Pascal und FORTRAN-77 sowie ein 68000-Assembler lieferbar. Neben p-Code kann mit einem Native Code Generator auch Maschinencode erzeugt werden. Das p-System bietet vollen Zugriff auf die Macintosh ROM-Routinen, so daß Grafik, verschiedene Schrifttypen und die Maus leicht von den Programmiersprachen aus angesprochen werden können.
Außerdem gibt es für das p-System viele weitere Zusatzprodukte sowie insgesamt etwa 4500 Seiten an Handbüchern.
Das p-System ist ein universelles, sehr komfortables Betriebssystem. Derzeit liegt sein Marktanteil mit weltweit 12% an dritter Stelle hinter MS-DOS und CP/M. Da das p-System prozessor- und hardwareunabhängig ist, ist es auf allen bekannten Mikrocomputern lauffähig. Damit kann jetzt eine große Auswahl vorhandener Software von Rechnern wie Apple II oder IBM PC auf den Macintosh übertragen werden. Das p-System bietet eine dynamische Speicherverwaltung und wird daher eingesetzt, um sehr große Programme auf kleinen Rechnern zu ermöglichen. Der mit 128K relativ bescheidene Macintosh-Arbeitsspeicher bildet so kein Hindernis mehr.

Bezugsquelle: FOCUS Computer GmbH, Friesenstraße 14, 3000 Hannover 1, Tel. 0511/345461

STATEX für Bildschirme

Bildschirme und Kunststoffe laden sich statisch auf und ziehen Staub und Schmutz an – sie bedürfen daher besonderer Pflege. STATEX ist ein speziell entwickeltes Tuch für Computerbildschirme. Es reinigt nicht nur wirkungsvoll Bildschirme, Gehäuse, Tastaturen usw., sondern schützt auch zuverlässig – bei regelmäßiger Anwendung – vor Wiederaufladung durch einen antistatischen Wirkstoff.

Bezugsquelle: PROSANA GmbH, Zeisselstr. 11 A, 6000 Frankfurt am Main 1, Tel. 069/590571

Hüthig-FACHBUCH-TIP

Apple ProDOS für Aufsteiger

Band 1

von Ulrich Stiehl
1984, 202 S., kart., DM 28,-
ISBN 3-7785-1027-4
Hüthig Verlag, Heidelberg

ProDOS ist das neue „professionelle DOS" (Professional Disk Operating System) für den Apple IIe sowie den mit einer Language Card ausgestatteten Apple II Plus. Reine Applesoft-Programmierer, die bereits unter DOS 3.3 programmiert haben, werden sich schnell an ProDOS gewöhnen, da die diesbezüglichen Unterschiede zwischen DOS 3.3 und ProDOS weniger gravierend sind.
Sinngemäß liegt das Schwergewicht in dem ersten Band von „ProDOS für Aufsteiger" auf der Assembler-Programmierung, da Assembler-Programmierer unter ProDOS völlig umdenken müssen. Insbesondere sind alle früheren Assemblerprogramme unter ProDOS nicht mehr lauffähig und bedürfen einer intensiven Überarbeitung. Band 1 befaßt sich überwiegend mit den theoretischen Grundlagen von ProDOS, der internen und externen Speicherorganisation und enthält grundlegende Beispielprogramme für Assembler-Programmierer sowie generelle Untersuchungen zum BASIC.SYSTEM. Da ProDOS über erheblich vielfältigere und leistungsfähigere, zugleich jedoch erheblich kompliziertere Dateistrukturen verfügt, sind theoretische Kenntnisse von ProDOS unabdingbar, wenn man die Features von ProDOS voll ausschöpfen will.

Aus dem Inhalt:
Ein erster Überblick – ProDOS und DOS 3.3 – Interne Speicherorganisation – Externe Speicherorganisation – MLI (Machine Language Interface) – ProDOS für Applesoft-Programmierer

BESTELLCOUPON

Bitte ausfüllen und an Hüthig Vertriebsservice, Postfach 102869, 6900 Heidelberg schicken.

Buchtitel

Name

Straße

Ort

Unterschrift

Computerunterstütztes Lernen

Die INTUS-Lernprogramme werden hauptsächlich schulbegleitend für die Nachhilfe und Übung eingesetzt. Sie haben sich vielfach bewährt und sind bei Lehrern und Schülern beliebt.
Lernen mit computerunterstützten Lernprogrammen ist für viele Schüler auf die Dauer eine neue, ergänzende Lernform. Gute Lernprogramme sind dialogfähige, geduldige „Privatlehrer". Sie passen sich dem Verständnis des Schülers laufend an, so daß das Lernen nie langweilig wird.
Alle INTUS-Lernprogramme sind selbststartend und menügesteuert.

Es sind keine Computerkenntnisse notwendig. Es liegen Programme für folgende Gebiete vor:
– Rechnen, Mathematik
– Physik
– Chemie
– Biologie
– Deutsche Sprache
– Fremdsprachen
– Geographie
– Vorschule
– Informatik
– Kaufmännische Fächer
– u. a. m.

Bezugsquelle: INTUS Software, Kaiserstraße 21, 7890 Waldshut-Tiengen, Tel. 0 77 51 / 79 20

Mailmerger für AppleWorks

AppleWorks umfaßt bekanntlich Textbearbeitung, Datenbank und Tabellenkalkulation auf einer Diskette. Teile aus diesen drei Anwenderprogrammen können miteinander verknüpft werden. Es können jedoch keine Daten aus einem der Anwenderprogramme in einem anderen verwendet werden. So können beispielsweise Adressen aus der Datenbank nicht mit Briefen aus der Textverarbeitung gemischt werden, um Mitteilungen zu individualisieren.
Dafür gibt es den „Mailmerger für AppleWorks". Es ist ein leicht zu verwendendes Programm, das den Einsatz von AppleWorks erheblich erweitert. Die Handhabung erfolgt wie bei Appleworks. Zusätzlich können Global- oder Einzelinformationen vor dem Ausdrucken eingegeben sowie Zahlenkombinationen zusammengezählt werden. Mailmerger wird mit einem ausführlichen Handbuch geliefert, das allerdings kaum benötigt wird. Der Preis ist DM 257,–.

Bezugsquelle: INTUS Software, Kaiserstraße 21, 7890 Waldshut-Tiengen, Tel. 0 77 51/79 20

Verschiedenes

Apple User Club Austria

Der Club umfaßt rund 300 Mitglieder in Österreich. Geboten wird ein eigenes Mitteilungsblatt, gemeinsame Einkaufsaktionen, ein Mitgliederverzeichnis, ein Diskettenabonnement, eine telefonische Sprechstunde, Clubtreffen an mehreren Orten in Österreich und vieles anderes. Aktuelle Mitteilungen werden über die Wiener Telefonnummer 00 43 – 2 22 – 47 82 16 verlautbart. Interessenten mögen sich mit dem Club, Postfach 51, A-1181 Wien, in Verbindung setzen.

Apple-Diebe am Werk

In der Nacht zum 30. 10. 1984 wurden im Goethe-Gymnasium in Bensheim eine Reihe von Apple-Computern entwendet, die von der Fa. Schöpp aus Bensheim stammten, welche dann in der Nacht vom 12. 11. 1984 ebenfalls heimgesucht wurde. Glücklicherweise konnten jedoch die Täter, die bei der Fa. Schöpp eingebrochen hatten, bereits am 6. 12. 1984 gefaßt werden, während die anderen noch auf freiem Fuß sind.

Peeker-Sammeldisketten

Einzelbezug DM 28,–
Fortsetzungsbezug DM 20,–
(Mindestbezug 6 Disketten)
(* = nur auf Diskette, nicht im Peeker gelistet!)

Sammeldiskette Heft 1 + 2, 1984
DOS-Format

T.DISASSEMBLER.65C02
DISASSEMBLER.65C02
T.ACCEL.WAIT
ACCEL.WAIT
T.ACCEL.BOOT
ACCEL.BOOT
ACCEL.LC.KOPIERER
T.ACCEL.LC.KOPIE
ACCEL.LC.KOPIE
T.ACCEL.ROM.KOPIE1
ACCEL.ROM.KOPIE1
T.ACCEL.ROM.KOPIE2
ACCEL.ROM.KOPIE2

TURTLE.GRAFIK.MIT.REMS
TURTLE.GRAFIK.OHNE.-REMS

DOUBLE.LORES.SOFT-SWITCH.DEMO
DOUBLE.LORES.APPLE-SOFT.DEMO
AMPER.DOUBLE.LORES.-DEMO
T.AMPER.DOUBLE.LORES
AMPER.DOUBLE.LORES
T.DOUBLE.LORES
DOUBLE.LORES

HIRES
T.PRINTHIRES
PRINTHIRES

DHGR.APSOFT.DEMO
AMPER.DOUBLE.HIRES.BAS
AMPER.DOUBLE.HIRES
T.AMPER.DOUBLE.HIRES
DHGR.LINEPLOTTER

INSTRING.TEST
INSTRING.OBJ
T.INSTRING.OBJ
INSTRING.LISA.SOURCE

LOESCHEN.EINES.ARRAYS
ULTRATERM.ENGLISCH*
ULTRATERM.DEUTSCH*

PRIMZAHLEN.OVERMEYER*
PRIM.OBJ0*
PRIM.OBJ1*
PRIM.TEST*
PRIM.TOOLKIT.SOURCE*

Sammeldiskette Heft 1–2, 1985
DOS-Format

T.RAMDISKLC
RAMDISKLC
T.IBS.RAMDISKDRIVER
IBS.RAMDISKDRIVER
T.AP20.RAMDISKTEST
AP20.RAMDISKTEST

T.QUICKCOPY
QUICKCOPY
QUICKCOPY.PUFFER
PRODOS.COPYA
T.PRODOS.COPYOBJ*
PRODOS.COPYOBJ
PRODOS.PATCH

T.APPLESOFT.FRE
APPLESOFT.FRE
T.LC.FRE
LC.FRE
FRE.TEST
T.RAM.FRE*
RAM.FRE

T.SCHIRMDISK
SCHIRMDISK.LISA.SOURCE
SCHIRMDISK
T.VIDEXT
VIDEXT.LISA.SOURCE
VIDEXT

GETPAS
GETDOS.PASCAL.SOURCE
COPYDUPDIR.PASCAL.-SOURCE

Sammeldiskette, Heft 1–2, 1985
CP/M-Format

STEUER.84
PASS.BAS
MENUE.BAS
HELP.BAS*
A.BAS
B.BAS
C.BAS
D.BAS
E.BAS
F.BAS
G.BAS
H.BAS
I.BAS
J.BAS
K.BAS
L.BAS
M.BAS
N.BAS

Hüthig Software Service
Postfach 10 28 69 · 6900 Heidelberg 1

PEEKER

Vorläufige Themenübersicht
Heft 3/85 erscheint am 25.2.85

Hardware

Der Apple als persönliches Ingenieurwerkzeug

Controller- und Laufwerk-Funktionsprüfung

Masterclock von Hoco im Test

Balfer-Interface

ProDOS

ProDOS für Anfänger, Teil 2

Neuer Format-Befehl für ProDOS

Pascal

Pascal 1.2: Evolution statt Revolution

Apples Maus lernt jetzt auch Pascal

CP/M

Konvertierung von CP/M – in DOS-Textfiles

Apple IIc lernt CP/M

Applesoft

Testgenerator für Legastheniker

Höhere Präzision bei den vier Grundrechenarten

Die CHRGET-Manipulation

Assembler

Macros für die neuen 65C02-Befehle

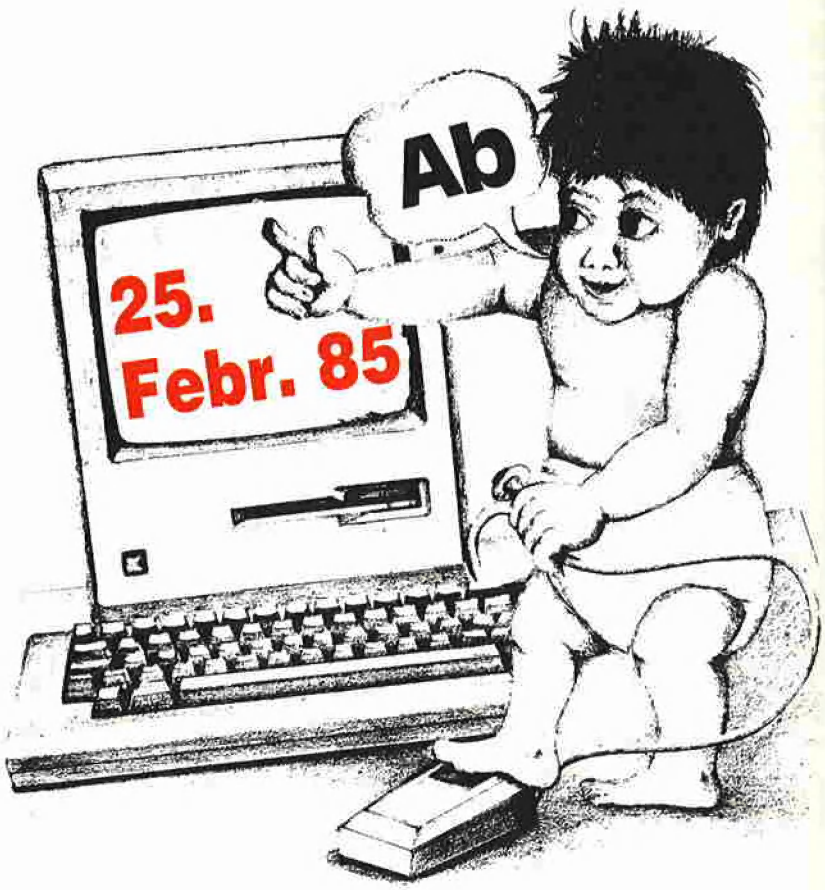
Flag-Monitor: Geschwindigkeitsoptimierung bei Assemblerprogrammen

Grafik

Double-Lores-Utilities

Macintosh

Microsoft Basic leicht geMACht, Teil 3



PRO METRIC® IST DA!

B2-Motherboard mit 6502, 65C02C, Z80B, V24, Parallel-Schnittstelle, 80-Zeichenkarte, RGB-Ausgang, 192 KB RAM	**DM 2299,--**
B3-Motherboard, wie B2, zusätzlich 256 KB Pseudofloppy	**DM 2899,--**
Prometric® B2 Tischgerät	**ab DM 3349,--**
Prometric® B3 Tischgerät	**ab DM 3999,--**
Prometric® Portable B2, Monitor 9"	**ab DM 4449,--**
Prometric® Portable B3, Monitor 9"	**ab DM 4999,--**

Neu von DRV: C-DOS, das 80-Track DOS für B2 und Kompatible, verarbeitet Integer, DOS 3.3 und Prodos-Files DM 149,--

DEUTSCHES BASIC, ROM-residient, übersetzt Ihre Apple-Soft Files ins Deutsche und umgekehrt DM 149,--

CALVADOS, deutsche Textverarbeitung DM 299,--

Wir suchen Stützpunkt-Händler mit entsprechendem finanziellen Hintergrund. Anfragen bitte nur schriftlich.

TEAC FD55F DM 674,--

FDC4-Controler	DM 179,--
Shugart-Bus-Kabel	DM 45,--
PREH-Commander AK87	DM 329,--
MONITOR 12", 22 MHZ	DM 349,--
M100-DRUCKER	DM 825,--
Graphik-Par.-Interface	DM 125,--

YE-DATA 480 DM 694,--

ERPHI-Controler	DM 298,--
IBM-look Gehäuse	DM 218,--
Monitor 15"	DM 528,--
32K-PRINTER-BUFFER	DM 389,--
MOTHERBOARD 64 KB + Z80	DM 699,--
PSEUDO-FLOPPY 256 KB	DM 899,--

Alle Preise inklusive 14% MwSt. Weiteres Zubehör auf Anfrage.

E. Böhmer, DRV, Am Kellersbusch, 6072 Dreieich, 0 61 03 / 8 46 47