

CLUB 80

Club info
der

TANDY -

GENIE -

und KOMTEK -

ANWENDER

21. AUSGABE



KONTAKTADRESSE : CLUB 80 / HARTMUT OBERMANN / SCHWALBACHER STRASSE 6 / 6209 HEIDENROD 1
POSTFACH 27 / TEL.: 06124 / 3913

Seite:
und Autor:

Seite:
und Autor:

Clubinternes

Neues vom Vorstand	1 - 3	Hartmut Obermann
Gebraikers Vereinigung	4	Joop Groenendijk
Auf zu neuen Ufern	5	Paul-Jürgen Schmitz
Termine / Messen	6	Redaktion
Club-Emblem	7 - 8	Gerald Schröder

Software

Merkzeuge eines Assembler-Prog. ...	9 - 10	Gerald Schröder
BASIC ruft Assembler	11 - 13	Artikel aus ELCOMP
Floppy auch ohne Interrupt	14	
Dreispaltig drucken	15 - 21	Arnulf Sopp
COM -> COM ohne CPMac	23 - 24	
Mie baue ich mir ein CP/M?	25 - 29	Gerald Schröder
MS-DOS Ecke	30	
Model 4 - Info	30	Klaus Hermann
Turbo Tip	31	
Turbo Patch	32	Hartmut Obermann
dBase II	33 - 35	Harald Mand
Apfelmännchen auf der GOP64	37 - 42	Helmut Bernhardt

Hardware

Genie, CP/M und 80 Zeichen und HRG	43 - 50	Manfred Heid
Portbelegung	44 - 50	Eckehard Kuhn

Börse

Ner hat was -- wer will was	51
-----------------------------------	----

sonstiges

Arbeitstier mit gesundem Herz	52	Heinrich Betz
Was habt Ihr gegen CP/M	53 - 54	Helmut Bernhardt
LS-DOS 6.3 - Kopierschutz	55	Klaus Hermann
Das TSCRIPS-Drama	56 - 61	Klaus-Jürgen Mühlenbein

Programm-Bibliothek

Neue Eingänge in der Diskettothek ..	63 - 66	Werner Förster, Andreas Rychlik
--------------------------------------	---------	---------------------------------

Club Bibliothek

Club-80 - Bücherei	65 - 66	Hartmut Obermann
Die Bücherliste / Buchvorstellungen	67 - 81	Gerald Schröder

Die letzten Seiten

Impressum	82	
Schluß	83	Redaktion
Hofacker-Verlag (Brief)	84	Hofacker-Verlag
Clubmitgliederadressen	am INFO-Ende	Redaktion
Sonderausgaben	am INFO-Ende	Gerald Schröder, Klaus-Jürgen Mühlenbein

Neues vom Vorstand

Diesmal könnte ich mit dem Material, welches sich in letzter Zeit bei mir angesammelt hat ein ganzes Info nur mit Neuigkeiten vom Vorstand füllen. Da ich sicher bin, daß dies nicht in euerem Interesse ist, werde ich mich möglichst kurz fassen und trotzdem versuchen nichts wesentliches auszulassen!

ELCOMP

Zumindest den TRS 80-Freunden, die sich schon länger mit ihrem Tandy-Computer herumschlagen, wird der Name ELCOMP ein Begriff sein. Diese vom Hofacker-Verlag herausgegebene Computer-Zeitschrift (die übrigens eine der ersten ihrer Art war!) beschäftigte sich in ihren Anfangsjahren recht häufig mit dem TRS 80-Computer. Da auch die Firma Hofacker mit der Zeit gehen muß, ist natürlich aus der heutigen ELCOMP die Tandylinie praktisch ganz verschwunden.

Unser Mitglied Ulrich Böckling hat sich nun einmal die Mühe gemacht und alle seine ELCOMP-Ausgaben (und er besitzt praktisch alle!) nach TRS 80-spezifischen Artikeln zu durchforsten. Die reiche Ausbeute, die er dabei gemacht hat, wird nun von Zeit zu Zeit bei uns im Info erscheinen und so auch den TRS 80-Usern zur Verfügung gestellt, denen die ELCOMP und die darin erschienenen Veröffentlichungen bisher kein Begriff waren!

In diesem Zusammenhang möchte ich dem Hofacker-Verlag für die freundliche Genehmigung danken, die uns erst ermöglicht, die ELCOMP-Artikel abzudrucken. Bitte beachtet dazu den Brief des Hofacker-Verlags, der aus postalischen Gründen leider auf der letzten Umschlagseite abgedruckt werden muß!

Mein ganz besonderer Dank gilt unserem Freund Ulrich Böckling, der uns durch seine Arbeit ein riesiges Reservoir interessanter Beiträge für das Clubinfo erschlossen hat!

Anschaffungen

Während des Vorstandstreffens wurde beschlossen ein Reinigungsset für Diskettenlaufwerke zu beschaffen. Zuständig dafür ist Werner Förster, bei dem das Set auch ausgeliehen werden kann. Genauerer dazu wird er euch sicher bald mitteilen! Weiterhin wird die Büchersammlung des CLUB 80 aufgestockt. Dazu wird Gerald Schröder eine Auswahl von Büchern zusammenstellen, über deren Ankauf der Vorstand dann entscheiden wird. Die Bücher sollen vor allem aktuell sein und sich mit CP/M sowie mit Hochsprachen beschäftigen. Weiterhin wird der Ankauf von Public-Domain-Software aus USA, sowohl für TRS/NewDOS als auch für CP/M erwogen!

CP/M Programmbibliothek

Wie schon im letzten Info berichtet, hat der CLUB 80 nun auch eine CP/M-Programmsammlung! Als Verwalter hat sich Andreas Rychlik zur Verfügung gestellt, dessen Vorstellung und Konzept ihr in diesem Info lesen könnt. Ich wünsche ihm mit der übernommenen Arbeit viel Glück und Spaß und hoffe, daß er sich bald vor Bestellungen nicht mehr retten kann!

Rückporto

Im Zusammenhang mit der CP/M-Programmbibliothek wurde auf dem Vorstandstreffen entschieden, daß ab sofort Besteller von Programmen aus einer der Sammlungen, genau wie Buchentleiher, die Portokosten tragen müssen! Es werden also nur noch dann TRS/NewDOS- und CP/M-Programme verschickt, wenn der Bestellung entsprechendes Rückporto beigelegt wurde. Ich bitte dies bei Bestellungen zu beachten!

Werbung

Es steht zwar nicht zu befürchten, daß unser Club demnächst an Mitgliedermangel zu Grunde geht, trotzdem soll mehr Werbung gemacht werden! Ein großer Teil dieser Werbung wird vom Vorstand durch eine Beilage zu einer Werbesendung der Firma Hermsstedt realisiert. Weiterhin wird ein neues Werbeplakat erstellt, daß dann bei Messen, Ausstellungen und Flohmärkten sowie in Computerläden ausgehängt werden soll. Hier sind nun die Mitglieder gefordert!

Jedes Mitglied sollte sich nach guten Plätzen für die Plakate umsehen. Wenn eine Ausstellung oder ähnliches in eurem Bereich läuft, solltet ihr dort ein Plakat aufhängen und auch eurem Computerhändler könnt ihr sicher eines unterjubeln. Mancher ahnt gar nicht, daß ganz in seiner Nähe noch ein Tandy-Besitzer einen recht einsamen Kampf mit seinem TRS 80 ausficht. Durch solche Aktionen können Verbindungen geschaffen werden, die sehr nützlich sein können!

Weiterhin sollten die Mitglieder mehr Mundpropaganda für den CLUB 80 machen! Sicher kennen viele unserer Mitglieder Tandy-Benutzer, die noch nicht dem Club angehören aber am Austausch von Informationen interessiert sind. Solche Leute sollten verstärkt angesprochen und für den Club geworben werden!

Material, wie Probeinfos und Werbeplakate, könnt ihr bei mir bekommen!

MS-DOS - Ecke

Auch dieses Problem wurde auf dem Vorstandstreffen diskutiert. Als Ergebnis kam heraus, daß vorerst keine spezielle MS-DOS-Ecke oder -Gruppe geschaffen wird. Es werden aber selbstverständlich Berichte und Artikel unserer Mitglieder, die sich mit MS-DOS beschäftigen veröffentlicht (siehe z.B. Info 20 Seite 99/100). Die Entscheidung, ob sich solche Artikel häufen, wird damit praktisch den Schreibern und Lesern des Infos, also allen Mitgliedern des CLUB 80 überlassen!

Ich hoffe auch weiterhin auf rege Diskussionsbeiträge zu diesem Thema und möchte in diesem Zusammenhang auf einen Brief von Paul-Jürgen Schmitz zu diesem Thema hinweisen, der in diesem Info abgedruckt ist!

TRS 80 Gebruikers Vereniging

Schon seit einiger Zeit hat der CLUB 80 durch Gerald Dreyer Kontakt mit einem holländischen TRS 80-Club. Nun hat der Vorstand der "TRS 80 Gebruikers Vereniging" eine kurze Vorstellung für unser Info geschrieben, die im Anschluß an diese Rubrik abgedruckt wird. Leider haben wir immer noch das Problem, daß wir niemand haben, der uns die interessanten Artikel aus der Zeitschrift des holländischen Clubs übersetzt. Dadurch gehen uns einige wirklich interessante Veröffentlichungen verloren! Sollte jemand im Club der holländischen Sprache mächtig sein, sollte er sich auf alle Fälle bei mir melden, damit wir die Sprachbarriere durchbrechen können!

Clubtreffen 88

Es erscheint zwar noch nicht sehr dringend, trotzdem hat der Vorstand schon das Datum für das Clubtreffen 1988 festgelegt. Das Treffen findet im Zeitraum zwischen dem 12. und 15. Mai statt. Der 12. ist ein Donners- und Feiertag, so daß wir diesmal für das Treffen etwas mehr Zeit ansetzen konnten. Ihr solltet euch diesen Termin schon jetzt in eurem Kalender vormerken und freihalten! Genaueres über Ort und Ablauf folgt in den nächsten Infos.

Zehnzeilerwettbewerb

Jens hat ja schon in seinem Schlußwort zum Info 20 von der Pleite bei der Abstimmung über die fünf eingesandten Zehnzeiler berichtet. Mangels Abstimmungsergebnis mußte der Vorstand über die Vergabe der Preise entscheiden und hat dies wie folgt getan. Jeder der vier Einsender bekommt als Anerkennung für seine Teilnahme eine Zehnerpackung Disketten. Es erfolgt damit keine Bewertung der Beiträge, da sich der Vorstand nicht in der Lage sah, ohne Hilfe der Mehrzahl der Mitglieder eine solche zu erstellen. Ich bedanke mich recht herzlich bei denen die mitgemacht haben und die nichts dafür können, daß die ganze Veranstaltung mehr oder weniger ins Wasser gefallen ist!

Damit genug für diesmal! Ich wünsche euch ein paar schöne Stunden bei der Lektüre des Infos und viel Spaß beim Computern, Euer

Karimut Obermann

GEBRUIKERS VERENIGING
 POSTBUS 551
 2070 AN SANTPOORT-NOORD
 TELEFOON 023-384135

Liebe Freunde,

Es ist mir eine grosse Ehre einen Beitrage zu Ihrem Blatt leisten zu dürfen.

Ueber Herrn Van de Sar, unsere Kontaktperson, habe ich schon einige Exemplare von Ihrem Blatt empfangen und jedes Mal habe ich darüber gestaunt was Sie mit einer so kleinen Gruppe leisten.

Herr Van de Sar hat mich, als Sekretär des holländisches TRS-80 Gebraucher Vereins gebeten, einen kurzen Bericht über unseren Verein zu schreiben.

Unser Verein wurde im Oktober 1978 errichtet und richtete sich ausschliesslig auf dem Gebrauch von TRS-80 Computern, wie die Modelle I, III und 4, sowie auch auf Computer die damit Compatibel waren.

Selbstverständlich hat das Aufkommen des PC auch für unseren Verein Verlust von Mitgliedern bedeutet, aber es hat sich gezeigt, dass viele Leute, obwohl sie einen PC gekauft hatten, doch eine so starke Vereinstreue hatten, dass sie Mitglied blieben.

Aus diesem Grund haben wir letzten Mai auf einer Generalversammlung entschieden, dass wir uns demnächst auch auf den Gebrauch von PC richten werden.

Im Augenblick hat unser Verein noch circa 2000 Mitglieder, diese Zahl war leider grösser gewesen, ist aber wieder am steigen seit wir den eben gemeldete Entschluss gefasst haben.

Diese Mitglieder können sich mindestens einmal monatlich in Versammlungen der neun Abteilungen treffen aus denen unser Verein besteht.

Bei diesen Versammlungen gibt es die Möglichkeit Vorträgen bei zu wohnen, oder Löt- und Programmierhilfe zu bekommen.

Weiter hat unser Verein eine Softwarebibliothek, die ungefähr 60 Disketten mit eigener Software umfasst, daneben verbreiten wir auch noch Public-Domain Software so wie Sie die auch kennen.

Auch hat unser Verein eine sehr aktieve Hardwaregruppe, die schon sehr viel geleistet hat, wie zum Beispiel : ein Modem, ein Druckerschalter, einen Ramdisk usw.

Wie auch Ihren Verein, bringen wir jede zwei Monate ein Heft, Remarks, in dem wir versuchen unsere Mitglieder von den neuesten Entwicklungen auf dem laufenden zu halten Vereinsnachrichten veröffentlichen und Programme und Hardware-Verbesserungen verbreiten.

Ich hoffe, Sie können sich mit diesen Informationen ein Bild von unser Verein machen und ich wünsche Ihnen noch viele schöne Stunden am Pult Ihren Maschinen.

Joop Groenendijk,
 Sekretär TRS-80 Gebruikers Vereniging

Paul-Jürgen Schmitz

Dipl.-Ökonom

Bremer Str. 9
6236 Eschborn
T.06196/46832

P. J. Schmitz • Bremer Str. 9 • 6236 Eschborn

Herrn
Hartmut Obermann
Schwalbacher Str.6

6209 Heidenrod 1

25. August 1987

Auf zu neuen Ufern

Lieber Hartmut,

Deinen Artikel habe ich mit großem Interesse gelesen. Es freut mich sehr, daß nun auch mal CP/M angesprochen wird. Daran bin ich selbst interessiert, denn in der Tat findet dieses Betriebssystem immer mehr neue Anwender, da bekannte Hardware-Anbieter, wie Schneider auch in neuerer Zeit noch Rechner mit diesem Betriebssystem neu auf den Markt bringen.

Ich bin an einem Erfahrungsaustausch dazu sehr interessiert, da ich auf meinem GENTE I mit Banker CP/M 2.2 fahren kann. Ich habe mich damit beschäftigt in der Vorstellung die vielen professionell nutzbaren Programme unter CP/M verwenden zu können, z.B. um Serienbriefe zu schreiben. Leider waren meine Bemühungen noch wenig erfolgreich. Wenn Du mir dabei weiterhelfen könntest, würde ich mich sehr freuen.

MS-DOS: Wie auch schon mein Artikel dazu ausdrückt, arbeite ich beruflich mit einem PC und suche Erfahrungsaustausch zu dieser Maschine. Dabei muß es doch kein Widerspruch sein, wenn sich ein Anhang dazu im Info fände. Natürlich finden sich zu diesem System ohnehin mehr Veröffentlichungen auf dem Markt, daß man mit Informationen eher überschwemmt zu werden droht; der Wert des Info's wäre es demnach, auf die Unterschiede, Besonderheiten, Möglichkeiten und Probleme sowie deren Lösungen aus Sicht des Aufsteigers hinzuweisen. Ich finde es deshalb unterstützenswert dieses Thema separat aufzunehmen. Es wird praktisch ja nur zu fragen sein, ob dies im Rahmen des Info-Umfanges noch zu leisten sein wird. Wir müßten mal fragen, wieviele Leser daran interessiert sind. Übrigens schreibt mir selbst Arnulf Sopp, daß er sich evtl. einen PC hinstellen wird, also warum dann nicht auch darüber "hacken".

Herzliche Grüße

Jürgen

-- Termine -- Termine -- Termine --

Nächster Redaktionsschluß25. November 1987

Süddeutsches Regionaltreffen 8. November 1987

Wir sehen uns auf der Hobbytronik in Stuttgart.
Treffpunkt ist der INFO-Stand der Hobbytronik im Eingangsbereich.

Zeitpunkt: ab Eröffnung zu jeder vollen Stunde
bis 5 Minuten danach

Norddeutsches Regionaltreffen20. - 22. November 1987

A c h t u n g :

Anmeldungen zwecks Zimmerreservierungen:

Hotel Zum Lindenhof
Bremer Straße 55 2107 Rosengarten-Neundorf
Tel.: 04108/ 7148

Was sonst noch auf dem Programm steht erfährt Ihr
beim Hartmut Obermann oder beim Gerald Schröder

-- Messen '87 --

HOBBY ELEKTRONIK 87Stuttgart 5. - 8. November 1987

PRODUCTRONICAMünchen 10. - 14. November 1987

HEFT
21
September
1987

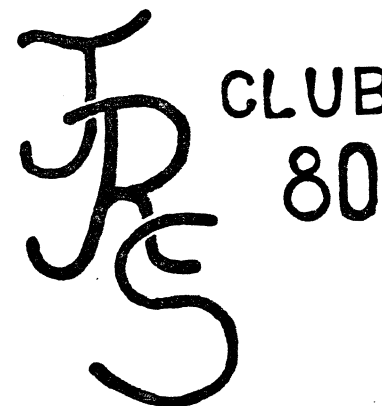
Das Club-Emblem - die Resonanz

Leider ist der Titel falsch, denn die Resonanz hat eigentlich gefehlt. Zwei Clubmitglieder (Hans-Martin und Jens, bei denen ich mich hier ganz herzlich bedanken möchte) haben mir ihre Vorschläge geschickt. Die Aktion ist trotzdem noch nicht beendet! Ich hoffe immer noch, daß sich unter Euch ein paar verkannte Künstler befinden, die sich beteiligen möchten. Schickt also ruhig Eure Vorschläge ein!

Als kleine Anregung werden nach diesem Artikel die bisher eingegangenen Vorschläge veröffentlicht. Es sind zwei von Jens und einer von Hans-Martin. Wenn Euch irgendeiner besonders gut gefällt, teilt es mir irgendwie mit (Telefon, Postkarte, Brief, Flaschenpost, persönliches Erscheinen,...).

Wie gesagt, erwarte ich noch Vorschläge und Stellungnahmen von Euch. Irgendwann wird dann der Vorstand endgültig darüber entscheiden, welches Emblem wir nehmen. Mehrheitsentscheidungen der Club-Mitglieder sind nach Jens' Erfahrungen nicht möglich, also ist es wohl so am besten.

Gerald Schröder



Jens Neueder (1)



Hans-Martin Stephan



User's of
TANDY's
and
Compatible's

Jens Neueder (2)



Eckehard Kuhn

Die Werkzeuge eines Assembler-Programmierers

Vor kurzem sollte ich "außer Haus" unerwartet ein Problem in Assembler lösen. Dabei merkte ich erst, wie hilflos ich ohne die elementaren Hilfsmittel (zum Assemblern) bin. Ich möchte sie Euch hier kurz vorstellen, damit Ihr 1. als Anfänger ungefähr wißt, was Ihr Euch besorgen solltet und 2. als Fortgeschrittener mal seht, wie andere Leute arbeiten. Natürlich habe ich nicht der Wahrheit letzten Schluß für mich gepachtet, und es kann sein, daß Ihr ganz andere Werkzeuge für besser geeignet haltet. Dann schreibt gefälligst einen Artikel darüber!

1. Die Paperware

1.1. Ein Quick-Reference-Guide

Ich benutze ein selbstgemachtes, denn ich habe noch kein geeignetes käufliches gefunden. Es sollte enthalten:

- alle Z80-Opcodes, alphabetisch geordnet mit den zugehörigen Hex-Codes (Binär muß immer erst umgerechnet werden); sinnvoll beim schnellen Codieren von wenigen Befehlen per DEBUG
- zu allen Codes die Art, wie sie die Flags beeinflussen und bei einigen auch, was sie genau machen (Rotier-/Schiebe-Befehle); dauernd beim Programmieren zugebrauchen
- alle Z80-Opcodes, aufsteigend nach Hex-Code sortiert; zum Disassemblieren (Decodieren) von wenigen Befehlen per DEBUG

1.2 Eine Umwandlungstabelle

Die Tabelle sollte 255 Einträge enthalten, jeweils mit ASCII-Zeichen und der dazugehörigen Zahl in dezimaler, hexadezimaler und binärer Schreibweise. S. auch ASCII-TAB aus Info 14. Diese Tabelle hilft bei kleinen Umrechnungen enorm.

1.3 Ein Buch mit einigen Assembler-Routinen

Dies ist sinnvoll, wenn man einige Algorithmen oft benötigt. Ein Beispiel ist die Umwandlung von ASCII-Eingaben in Zahlen oder umgekehrt von Zahlen (Registerinhalten) in darstellbare ASCII-Codes (Ziffern). Eine gute Sammlung von Standard-Routinen ist auch gut. Das Z80-Buch von Zaks bietet einige Mathe-Algorithmen; ansonsten kann ich es leider nicht empfehlen und kenne auch kein anderes Buch, das sich dafür eignet. Also: selbstmachen.

1.4 Das DOS-Buch von Grosser

Unverzichtbar beim Programmieren des Floppy-Controllers und beim intensiven Arbeiten mit den Innereien des DOS'.

1.5 Die Seiten 53-64 aus dem Newdos-Handbuch

Hier werden alle zum Diskettenzugriff wichtigen Routinen hervorragend beschrieben. Im Gegensatz zum DOS-Buch allerdings nicht die Routine selbst, sondern wie man sie benutzt. So sind schnell und effektiv Programme mit Diskettenzugriff schreibbar.

1.6 Das ROM-Listing von Röckrath

Das ROM selber ist nicht so interessant, obwohl man einige Routinen (bei mir vor allem 33h, 49h, 5d9h, RST 18h) gut gebrauchen und hier etwas über sie erfahren kann. Vor allem beim Disassemblieren von Programmen, die das ROM nutzen, ist es aber unverzichtbar. Schön ist auch die Belegung der Tastatur-Adressen dargestellt, so daß man leicht direkt darauf zugreifen (obwohl nicht unbedingt empfehlenswert).

2. Die Software

2.1 ZEUS

Zweifelloch das Werkzeug für den Assembler-Programmierer. Allein der automatische Syntax-Check ist goldwert. Wenn Ihr Arnulf und mir nicht glauben wollt, sondern lieber den Edtasm oder sonstwas benutzt, ist das auch OK.

2.2 DSMBLR

Der Misosys-Disassembler kann eine verdaunte Menge, vor allem, wenn man ihm sagt, wo DBs usw. stehen. Dies kann man mit FED feststellen (s. 2.3). Die Outputs können mit ZEUS bearbeitet werden. Durch diese Reihenfolge (FED, DSMBLR, ZEUS) habe ich schon viele Programme erforscht, kommentiert und schließlich geändert.

2.3 FED

Der File-Editor bietet mehrere Funktionen, wegen derer er sich hervorragend zur Erkundung von Programmen eignet. Um nur einige zu nennen: Suchen nach einer Adresse (L); Anzeigen, an welcher Adresse man sich befindet (M); Blättern durch die Ladecodes (Z). Super ist das Ding, um sich anzusehen, wo Dms/Dws/DBs in einem File auftauchen könnten (ASCII-Texte einfach zu lokalisieren, Tabellen oft durch stetig wiederkehrendes Zeichen (jedes zweite, nämlich den MSB) gekennzeichnet), um dies dann DSMBLR mitzuteilen. Dadurch kann man sich oft den Ärger ersparen, die Funktion von vollkommen unsinnig scheinenden Programteilen begreifen zu wollen, die in Wirklichkeit nur die Copyright-Meldung darstellen.

2.4 DEBUG

Dürfte jeder kennen (sonst "123" drücken und das Newdos-Manual nochmal ganz genau durchlesen). Der Vorteil ist offensichtlich: Das Teil verbraucht nur DOS-Speicher und kein freies RAM. Damit (vor allen den Stepper-Funktionen I und C) lassen sich hervorragend Fehler in eigenen Programmen aufspüren oder das Verhalten freader Programme nachvollziehen.

2.5 Routinen-Sammlung

Oft benötigte Routinen (wie oben angesprochen) sollte man sich dort abspeichern, wo sie schnell zu finden sind. Bei mir ist (neben den angesprochenen Ein-Ausgabe-Umwandlungs-routinen) dauernd ein Prgramteil in Benutzung, der nach einem File-Namen fragt, diesen File dann byteweise einliest und wieder auf Diskette schreibt (evtl. unter einem anderen Namen). Zwischen dem Lesen und Schreiben füge ich dann verschiedene Routinen ein, die den File umwandeln (z.B. die Ladecodes ändern o. analysieren oder bestimmte Bytes entfernen).

3. Die Hardware

Hier ist nur ein Teil zu nennen: ein Taschenrechner, der auch mit Hexadezimal- und Binärzahlen rechnen bzw. diese in dezimale und umgekehrt umwandeln kann. Bei leistet dies der Casio fx-115.

Das war es schon, ohne jeden Anspruch auf Vollständigkeit. Ich hoffe, daß wenigstens die Assembler-Anfänger nun einen ungefähren Überblick haben, welche Werkzeuge in Ihren Erste-Hilfe-Kasten gehören, damit sie nicht so hilflos sind wie ich, wenn sie unerwartet um die Vorführung Ihrer Fähigkeiten gebeten werden. Übrigens werden unter CP/M bestimmt ganz andere Werkzeuge benutzt. Ich hoffe, daß bald mal jemand darüber berichtet.

Gerald Schröder

HEFT
21
September
1987

BASIC ruft Assembler

Laden und Aufrufen von Routinen in Maschinensprache

6 Verfahren für TRS-80 und Video Genie

PET-Besitzer haben es gut. Sie können kleine Maschinenprogramme (ab jetzt nur noch MP's genannt) in den 2. Kassettenpuffer "poken", wo sie (die MP's) vor BASIC geschützt sind.

```
1 * MP-CALL. VERSION 1: MP IM RESERVIERTEN RAM
2 *
10 DATA 205,127,10,125,33,0,60,119,17,1,60,1,255,3,237,176,201
20 A=32750:FORX=A TO A+16:READB:POKEX,B:NEXT:DEFUSR=A
30 INPUT"MIT WELCHEM ASCII-CODE VIDEO FUELLEN";C
40 C=USR(C)
50 GOTO 50
```

Version 1

Version 1 im Listing ist die klassische Standardlösung. Das MP wird in den reservierten RAM "gepocket". Für nicht TRS-Fans: Der Rechner fragt nach dem Einschalten bzw. Booten im DOS "Memory Size?". Wenn man die Frage nicht mit (ENTER) ignoriert, sondern eine Adresse eingibt, "denkt" BASIC, daß sein RAM bei dieser Adresse - 1 endet.

Das kleine MP, das hier zur Demonstration benutzt wird, füllt eine Bildschirmseite mit einem vom User wählbaren ASCII- oder Grafikzeichen. Geben Sie z. B. 65 ein, sehen Sie schlagartig 1024 A's oder bei 191 wird der Schirm weiß.

DEFUSR= ist ein Disk-BASIC-Befehl und eine praktische Abkürzung von POKE 16526, . . . , POKE 16527, . . . in Level 2. Mehr sollte man über Version 1 nicht schreiben, gehen wir zur 2.

```
1 * MP-CALL. VERSION 2: MP IN STRING. STRING IN HIGH RAM
2 *
10 DATA 205,127,10,125,33,0,60,119,17,1,60,1,255,3,237,176,201
20 FOR X=1 TO 17:READ B:A*=A*+CHR*(B):NEXT
21 X=VARPTR(A*):DEFUSR=PEEK(X+1)+256*PEEK(X+2)-65536
30 INPUT"MIT WELCHEM ASCII-CODE VIDEO FUELLEN";C
40 C=USR(C)
50 GOTO 50
```

Version 2

TRS-80-Besitzer haben es besser. Sie dürfen nämlich selbst bestimmen, was sie mit ihrem RAM anfangen, allerdings auch selbst darüber nachdenken, wie sie MP's und BASIC auseinander halten.

Nun, das Nachdenken des Autors erbrachte 6 Varianten. Sicherlich gibt es noch mehr, aber dieser Artikel soll ja kein Buch werden.

Das MP muß lediglich als eine sinnige Folge von Bytes irgendwo im RAM stehen. Ein String darf fast den gesamten Zeichensatz also CHR\$(0) bis CHR\$(255) enthalten. Fast!

Steht eine 34 in dem String, weil sie da hinein "gepocket" wurde, wird das vom Computer als Anführungszeichen und somit als Stringende erkannt. LD HL,(nn), was 34 bedeutet, sollte man also im MP vermeiden. OK, lädt man also eine String mit den Bytes des MP's, wie in Zeile 20 gezeigt, und sagt dann via VARPTR, daß das MP da beginnt, wo die String im RAM startet, ist alles klar.

Zeile 21 lädt den USR-Vektor. Hier wird ein System mit >16k unterstellt, was in L3 wohl der Normalfall sein dürfte.

Wenn Sie in Level 2 arbeiten und (=16k haben, können Sie die beiden Bytes PEEK(X+1)/(X+2) direkt nach 16526/27 poken.

In Version 2 steht die String mit dem MP im mit CLEAR reservierten Stringbereich, also im high RAM. Das hat den Nachteil, daß es nach jedem Programmstart erst dahin gebracht werden muß.

In Version 2.1 wird zuerst eine Stringkonstante definiert (Zeile 15). Die Zeichen darin sind beliebig. Der Autor hat die Ziffern einge-

```
1 * MP-CALL. VERSION 2.1: MP IN STRING. STRING IM PROGRAMMTEXT
2 *
10 DATA 205,127,10,125,33,1,60,119,17,2,60,1,254,3,237,176,201
15 A*="12345678901234567"
16 A=A:X=VARPTR(A*):A=PEEK(X+1)+256*PEEK(X+2)
20 FORX=A TO A+16:READB:POKEX,B:NEXT:DEFUSR=A
30 INPUT"MIT WELCHEM ASCII-CODE VIDEO FUELLEN";C
40 C=USR(C)
50 GOTO30
```

Version 2.1

Vergleichen Sie einmal die DATA-Zeilen. In Version 2.1 sind einige Bytes geändert worden. Es gibt keine Null-Bytes mehr, und da man in einem MP nicht einfach alle Nullen in Einsen ändern kann, noch einige Changes. Warum?

Nun, in einer BASIC-Zeile darf ein Nullbyte nur an einer Stelle stehen und zwar am Ende. Es ist das amtliche Schlußzeichen, das wir nicht einfach mitten in eine Zeile poken können.

Der seltsame Befehl A=A in Zeile 16 hat die Aufgabe A zu definieren. Man könnte auch A=4711 schreiben. A wird nach der VARPTR-Funktion benutzt. Würde man nach VARPTR A als neue Variable definieren, würde die Variablenliste länger, A\$ verschoben werden, und die Routine mit einer falschen Adresse arbeiten.

```
1 * MP-CALL. VERSION 3: MP IN ARRAY (SINGLE PREC.)
2 *
10 DATA 205,127,10,125,33,0,60,119,17,1,60,1,255,3,237,176,201
20 DIM MP(4)
21 B=B:X=X:A=VARPTR(MP(0))
22 FOR X=A TO A+16:READ B:POKE X,B:NEXT:DEFUSR=A
30 INPUT"MIT WELCHEM ASCII-CODE VIDEO FUELLEN";C
40 DEFUSR=VARPTR(MP(0)):C=USR(C)
50 GOTO 50
```

Version 3

Die schlechteste Lösung dafür steht im Level-2-Handbuch. Die einfachste ist 65536 zu subtrahieren.

geschrieben, um leichter 17 Zeichen (MP-Länge) abzählen zu können. Eine im Programm definierte String steht bekanntlich im Programmtext. Wenn man also in diese das MP schreibt, kann man anschließend die DATA-Zeilen und die POKE-Schleife löschen. Das Programm selbst hält dann das MP. Den Nachteil der Version 2 haben wir somit beseitigt und prompt einen neuen eingefangen.

Also Version 2 hat Nachteile, 2.1 auch, beide gemeinsam haben noch den Nachteil, daß das MP nicht länger als 255 Bytes (Version 2) bzw. 233 (Version 2.1) sein darf, ergo doch die Mem-Size-Lösung?

Version 3 ist die Array-Lösung. Durch DIM MP(4) werden 5 Elemente dimensioniert, ergibt in single precision 20 Bytes, genug unsere 17 aufzunehmen.

Falls Ihr Rechner nicht mehr als 16k hat, können Sie die Routine verwenden. Darüber wird es mit dem POKE schwierig, denn bekanntlich muß man ja sobald 32767 überschritten werden, das MSB setzen, sprich ein negatives Vorzeichen einführen.

Wie dem auch sei, der Array kann hier oder dort im RAM stehen, mal muß man positive, mal negative Zahlen poken, warum soll man den

Array nicht gleich mit den Zahlen laden (so als A(X)=B) ?

Wir sind leider in BASIC und nicht in APL. D. h. ein Array-Element belegt im Integer-Format immer noch 2 Bytes. Würden wir da eine Zahl (256 hineinladen, bliebe ein Byte auf 0, sprich jedes 2. Byte des "MP's".

Trotzdem ist Version 3.1 machbar. Man muß halt jeweils 2 Bytes in ein Element packen. Wie die Zeilen 21 - 25 zeigen, ist dies ganz einfach. Bei seinem ersten Versuch bekam der Autor prompt eine Error-Meldung, die mit Zeile 23 behoben wurde. Diese prüft, ob die gepackte Zahl außerhalb der Integer-Grenze liegt (32767) und verfährt dann wie bei POKE.

```

1 * MP-CALL. VERSION 3.1: MP IN INTEGER-ARRAY
2 *
10 DATA 205,127,10,125,33,0,60,119,17,1,60,1,255,3,237,176,201,0
20 DEFINT M:DIM MP(8)
21 FOR X=0 TO 8
22 READ L,H: B=L+256*H
23 IF B>32767 THEN B=B-65536
24 MP(X)=B
25 NEXT
30 INPUT"MIT WELCHEM ASCII-CODE VIDEO FUELLEN";C
40 DEFUSR=VARPTR(MP(0)):C=USR(C)
50 GOTO 50

```

Version 3.1

Die DATA-Zeile ist ein Byte länger geworden. Das MP an sich hat 17 Bytes. Da aber immer 2 Bytes gelesen werden, brauchen wir 18, weshalb ein Nullbyte angehängt wurde. Dieser NOP wird nie ausgeführt, weil davor 201, im Klartext RET steht.

Sie möchten gerne, daß jemand, der Ihr Programm mit BREAK anhält und es dann listet oder wieder mit RUN startet, eine böse Überraschung erlebt ?

Dann nehmen Sie doch die Version 4 mit der DATA-Zeile von Version 1. In Version 4 ist die erste Zeile (0) dafür vorgesehen, anstelle der Punkte des MP aufzunehmen.

Wo Zeile 0 im RAM steht, wird in Zeile 15 festgestellt. Ansonsten ist im Programm nichts

Neues. Mit der DATA-Zeile wie im Listing, d. h. ohne Nullbytes, läuft das Programm beliebig oft. Mit Nullbytes (DATA-Zeile lt. Version 1) läuft es auch in der Schleife 30-50-30. Wenn Sie allerdings mit BREAK die Schleife verlassen und dann wieder RUN geben, ernten Sie nur Syntax-Error in Line 273.

Ein LIST zeigt dann nur wilde Zeichen in 0 und keine Zeile 273. Ursache sind wieder die Nullbytes mitten in der Zeile. Den gleichen Effekt können Sie übrigens auch in der Version 2.1 erzielen.

Wenn es Ihnen nur auf den Effekt als solchen ankommt, brauchen Sie kein MP. Poken Sie einfach im Laufe des Programms in Zeilen, die nicht mehr durchlaufen werden, eine Null.

```

0 REM.....
1 * MP-CALL. VERSION 4: MP IN ERSTER BASIC-ZEILE
2 *
10 DATA 205,127,10,125,33,1,60,119,17,2,60,1,254,3,237,176,201
15 A=PEEK(16548)+256*PEEK(16549)+5
20 FORX=A TO A+16:READB:POKE X,B:NEXT:DEFUSR=A
30 INPUT"MIT WELCHEM ASCII-CODE VIDEO FUELLEN";C
40 C=USR(C)
50 GOTO 30

```

Version 4

P. Wollschlaeger, Hildesheim

Floppy auch ohne Interrupt

Im Info 20/87 stellt Andreas Rychlik unter der Überschrift "Hardcopy auch ohne Interrupt" eine Methode vor, mit der Boot-Falle einen Ausdruck des Bildschirms anzufertigen. Vor Jahren wurde einmal in einer Computerzeitschrift eine ähnliche Idee veröffentlicht: Das Programm im Boot-Sektor fertigte eine Hardcopy gleich selber an.

Bei den TRS-80 und den älteren Genies ohne Banker fällt mir dazu auch nicht mehr ein. Hat der User jedoch die Möglichkeit, den unteren Speicherbereich direkt oder als Kopie (mit dem EG 64 MBA oder so) zu modifizieren, dann eröffnet sich eine einfachere, zudem flexiblere Lösung:

Programme, die die maskierbaren Interrupts (INT) ausknipsen, werden sich damit abfinden müssen, daß der nicht maskierbare Interrupt (NMI) nach wie vor ausgegeben werden kann. Bei den älteren Modellen unserer Computerfamilie geht das mit dem "Reset"-Knopf (der gar keinen Reset, sondern eben jenen NMI triggert). Beim Genie 3s können die beiden unbelegten Tasten leicht für diesen Zweck benutzt werden. Bei sonstigen Typen tut's ein Klingelknopf, der den NMI-Pin der CPU auf GND legt.

Der NMI springt bei 0066h ein. Dort steht in unserem Microsoft-Interpreter eine Kaltstart-Routine. Sie wird, falls der Interpreter überschrieben werden kann (s. o.), heimtückisch für unseren sinistren Zweck verbogen:

Der Library-Befehl DUMP soll den Bildschirminhalt als CMD-File auf Diskette schreiben, von wo er jederzeit wieder abgerufen werden kann. Die DUMP-Parameter enthalten außer dem Programmnamen (hier BILD/CMD, aber das ist Geschmacksache) natürlich auch die Bildschirmanfungs- und -endadresse sowie als Einsprungsadresse 0071h. An 0071h (Label wait) steht nämlich der Befehl JP 0049h. Er bewirkt, daß die Tastatur so lange interviewt wird, bis der User irgendeine Taste drückt. Am Ende der Tastatur-Routine steht RET, so daß es anschließend nahtlos im DOS, im BASIC oder sonstwo weitergeht.

Das hier vorgestellte Programm selbst bietet gegenüber der (bei ROM-Computern vermutlich unausweichlichen) Unbequemlichkeit einen weiteren Vorteil: Es endet nämlich ganz schlicht mit RET, so daß es in PACMAN gleich weitergeht, sobald der traumhafte High Score für die neidische Nachwelt auf Disk dokumentiert ist. RETN (2 Bytes lang) nach dem NMI ist nicht nötig, denn es restauriert lediglich den Zustand der INT-Akzeptanz (EI oder DI). Da bei Verwendung dieser Methode die INTs schon maskiert waren, können sie es nach dem NMI auch bleiben.

Arnulf Sopp

0066	00001	ORG	0066h	;Entry des NMI	
	00002				
0066	E5	PUSH	HL	;Register retten	
0067	F5	PUSH	AF		
0068	217400	LD	HL,dump	;dort steht der DUMP-Befehl	
006B	CD1944	CALL	4419h	;DUMP-Befehl unter DOS-CALL ausführen	
006E	F1	POP	AF	;Register restaurieren	
006F	E1	POP	HL		
0070	C9	00009 start	RET	;weiter in Programm	
	00010				
0071	C34900	00011 wait	JP	0049h ;auf Tastendruck warten	
	00012				
0074	44	00013 dump	DM	'DUMP,BILD/CMD,3C00H,3FFFH,0071H',0dh ;DOS-Befehl DUMP	
	00014				
0070	00015	END	start	;dort steht RET (gleich wieder ins DOS)	
00000	Fehler				
dump	0074	start	0070	wait	0071

HEFT
21
September
1987

Arnulf Sopp

Das Programm, mit dem dieser Text ausgedruckt wurde, ist so lang, daß ich es nur in sehr groben Zügen erläutern möchte. Es interessiert wohl eh' keinen, wie es arbeitet. Deshalb will ich außer einigen Hinweisen zum Ablauf nur erklären, wie es an andere Hardware angepaßt werden kann, und welche Mängel es hat (mit denen man aber gut leben kann).

Fangen wir mit letzterem an. Jedes OAh-Byte (LF) wird brutal in ODh (CR) umgewandelt. Und jedes ODh wird als Zeilenende interpretiert. Deshalb sind beispielsweise Steuercodes unzulässig oder zumindest sehr schwierig zu handhaben, die OAh (10d) oder ODh (13d) enthalten (z. B. linken Rand auf die 10. oder 13. Stelle setzen).

Leere Zellen (in TSCRIPS z. B. bei PF=2) am Kopf oder Fuß einer Seite werden nicht unterdrückt. So kann dort optisch eine Achterbahn entstehen. Es empfiehlt sich daher, PF=1 zu wählen (Absatz ohne Leerzeile).

TSCRIPS zählt die Seiten anders als dieses Hilfsprogramm. Daher muß im Text TM=1 und BM=72 (bei Endlospapier) gesetzt werden. Der Seitenvorschub wird vom Drucker erledigt.

Es ist den meisten von uns auch aus schmerzlicher Erfahrung bekannt, daß mehrere Drucker-Steuercodes im TSCRIPS-Text das Zeilen-Layout verhunzen können. Bei den sehr kurzen Drittelzeilen macht sich das umso auffälliger bemerkbar.

Es ist auch ziemlich schwierig, Steuercodes zu programmieren, die nur zu Beginn einer Zeile vom Drucker akzeptiert werden.

In der Überschrift hätte ich z. B. gern auf den linken und rechten Randausgleich zugunsten einer Zentrierung verzichtet. TSCRIPS warf das Handtuch angesichts des Chaos von \$- und &-Sequenzen.

Schon der Fettdruck im Titel dieses Artikels war ein echtes Problem. Die Ausgabe dieser Codes gilt nicht etwa nur für eine Textzeile - was weiß der Drucker von meiner Muttersprache und meinen Versuchen, sie zu schreiben? Er kennt nur Zeilen von ganz links bis ganz nach rechts. Das Fett-Kommando mußte daher nach jedem Wort wieder aufgelöst werden, sonst wäre die ganze Druckzeile fett gedruckt worden, also auch die mittlere und rechte Spalte.

Noch so ein Beispiel ist der Vorschlag für das Banking mit dem G3s, der später folgt. Wegen des Wechsels von Proportionalanschrift auf LQ mußte auch nach jeder Zeile auf PS zurückgeschaltet werden, sonst wäre alles Folgende in diesen Zeilen, auch in den beiden anderen Spalten, unangenehm aufgefallen.

Und hätte ich jetzt, als ich dieses schreibe, gewußt, in welcher Spalte der Programmvorschlag erscheinen wird, hätte ich mich präziser ausdrücken können.

Es ist also so richtig etwas für Leute, die einfach nur einen Text tippen wollen und auf die äußere Form keinen gestelgerten Wert legen. Oder für totale Freaks, die den noch immer, wenn auch erschwert, möglichen TSCRIPS-Fähigkeiten ein Bonbon hinzufügen möchten.

Die Anpassung an andere Drucker als den NEC P6/7 ist einfach: Die Steuer-codes müssen entsprechend umgeschrieben werden. In den Kommentaren des Listings sind sie erklärt.

Codes für Features, die der eigene Drucker nicht hat, können schlicht gelöscht werden. Unabdingbar sind nur die linke (und mögl. rechte) Randeinstellung und die Möglichkeit, einen CR ohne Papiertransport zu programmieren.

Notfalls geht es sogar ohne das. Wenn bei der Grundeinstellung des Druckers (Label lp-init) Tabulatormarken gesetzt werden (die sollte der Drucker aber nun wirklich beherrschen), dann kann man zur nächsten Drittelzeile mit einem TAB (09h) gelangen. Für den beidseitigen Randausgleich sorgt notfalls das Textprogramm (bei TSCRIPS: J=Y). Falls man auf ihn überhaupt Wert legt.

Die Anpassung an einen anderen Computer als das G3s ist schon problematischer. Das Programm lebt davon, daß ein Text nicht sofort ausgedruckt, sondern zunächst gepuffert wird. Notfalls kann der Puffer in der oberen Hälfte des Textspeichers liegen, so daß es auch mit einem TRS-80 geht.

Banking ist eleganter und spart Speicherplatz. Hier wird der Megabyte-Blocker von Helmut Bernhardt angewendet. Es geht auch mit seinem 256-kB-Blocker. Elnige Computertypen in unserem Club haben von Hause aus mehr als 64 kB. Mit denen funktioniert es auch. Gegf. muß dann auf die RAM-Floppy verzichtet werden.

Beim G3s ohne Umbauten sieht das etwa so aus: Alle Bankerei mit irgendwelchen Megabytes (hier über den Port F3h) wird ersetzt durch das Banking, das dem G3s angeboren ist: Mit der Codefolge

```
IN  A,(OF9h)
XOR  CF0h
OUT  (F9h),A
```

wird zwischen dem User-RAM in Bank 1 und einem

möglichen Textpuffer in Bank 2 gewechselt.

Und schließlich das Problem von immer: Das Programm selbst, der Stack und der Textpuffer müssen irgendwo Platz haben. Wenn alle Stricke reißen, paßt alles ins Himem (das dann geschützt werden sollte). Wer aber mit Banking arbeitet, muß unbedingt darauf achten, daß das Programm und der eigene Stack in einem Common-Bereich liegen.

In der hier vorgestellten Version liegt das Programm hinter einigen TSCRIPS-Umleitungen ab 3260h. Da es einigermaßen lang ist und beim G3s bereits ab 3400h wieder Lebenswichtiges steht, mußte der Stack in die Cassetten-Routine des Interpreters verlegt werden. Das kann sich jeder Leser nach seinen eigenen Gegebenheiten einrichten.

Diese Routine arbeitet mit jedem Textprogramm zusammen, das über die Druckroutine des Interpreters printet. TSCRIPS tut das auch, wenn man eine kleine Änderung durchführt: Bei meiner Version (Rel. 5.4, aber vielleicht auch nicht mehr ganz sauber, als ich es bekam) wird nur an der Stelle 86E3h etwas auf den Drucker ausgegeben. Hier kann CALL 05BBh (CD-BB-05) gepatcht werden.

Bei der Anwendung zusammen mit TSCRIPS muß der linke Rand mit LM=0 ganz nach links gesetzt werden. Die RM-Einstellung ist vom Pitch abhängig, also etwa 20 - 40. Außer bei Proportionalanschrift muß dieser Wert unbedingt derselbe sein wie der Druckercode für den rechten Rand.

Wer Lust hat, kann das Programm zum Bestandteil letzten Zeile aufgesucht. TSCRIPS und andere Programme haben nämlich die @G-Option (DIR anzeigen) Untugend, mit einer Folge von Zeilenvorschüben das Blatt auszustoßen. Da das als CMD-File mit @C oder

auch schon vor TSCRIPS aufgerufen.

Es bleibt auch nach dem Ausdrucken aktiv, d. h. die Umleitungen des Druckers und der Interrupts werden nicht rückgängig gemacht. Es könnte ja sein, daß der Benutzer mit einem Text noch nicht zufrieden ist und er ihn verändert noch einmal ausdrucken will. Der Nachteil: Alle folgenden Ausgaben auf den Drucker, auch nach der Arbeit mit dem Textprogramm (z. B. DIR,P) werden dreispaltig gedruckt. Man sollte also zuvor neu booten.

Oder man ruft das Programm erneut auf, diesmal mit dem Zusatz .N. Gemäß der G-DOS-Gewohnheit wird dadurch der normale Zustand der Drucker- und der INT-Routine restauriert. Dieser Teil des Programms läuft im Sektorpuffer des DOS, wo er nicht stört und keinen anderweitig benötigten Speicherplatz verbraucht.

Jetzt in aller gebotenen Kürze die Arbeitsweise des Programms. Durch die Drucker-Umleitung wird der ASCII-Klartext nicht sogleich ausgedruckt, sondern in einem Puffer zwischengespeichert (UP buff). Dabei werden ein Zeiger auf das bisherige Textende und ein Zeilenzähler unterhalten.

Inzwischen wartet eine Umleitung der Interrupt-Serviceroutine darauf, daß die PRINT-Taste des G3s gedrückt wird. Wer die nicht hat, kann stattdessen jede andere Taste zu diesem Zweck nehmen.

Wenn PRINT gedrückt wurde, beginnt der wirkliche Ausdruck mit einigen Berechnungen: Zunächst wird der erste CR nach der letzten Zeile aufgesucht. TSCRIPS und andere Programme haben nämlich die @G-Option (DIR anzeigen) Untugend, mit einer Folge von Zeilenvorschüben das Blatt auszustoßen. Da das bei die Perforation nie-

mals an der Abreißkante landet, ist das überflüssiger Schnickschnack.

Nicht nur das; in unserem Falle würden die abschließenden Zeilenvorschübe dem Programm einreden, es gäbe noch allerhand leeres Papier anfüllen. Das muß nicht sein.

Die Stelle des stattdessen ermittelten wirklich letzten CR wird nun als Zeiger auf das tatsächliche Textende geladen. Bis dorthin werden nun die CRs, also die Zeilen des Textes abgezählt. Da bei dem hier sichtbaren Format 195 Textzeilen auf eine Seite passen, wird die Zeilenzahl zunächst durch diesen Wert dividiert, um die Anzahl der Seiten zu berechnen.

Für jede neue Seite muß das Zeilenlayout neu ermittelt werden. Das geht so: Je nach dem Ergebnis einer Division durch 3 (drei Drittelzeilen) wird der Beginn der mittleren und der rechten Textspalte festgestellt. Beim Ausgeben einer Druckzeile, die ja drei auseinanderliegende Textzeilen enthält, wird nun abwechselnd eine Textzeile aus dem linken, mittleren und rechten Bereich ausgegeben. Dazwischen liegen jeweils CRs,

damit der Randausgleich des Druckers hinhaut. Durch die Einstellung von 0/180' Zeilenabstand (= überhaupt kein Papervorschub) bleibt der Drucker in der Zeile. Erst vor der nächsten linken Spalte wird ein normaler Zeilenvorschub ausgegeben.

Irgendwann später ist schließlich der ganze Text ausgedruckt. Die Drucker-Routine, die ja nun nicht mehr drucken, sondern puffern soll, wird wieder verbogen. Mit RET wird am Ende nicht etwa gleich zum Textprogramm, sondern in

die Fortsetzung der INT-Routine verzweigt. Diese kehrt zurück, wenn sie die Register wieder restauriert hat, die sie zuvor rettete

Stützt da jemand? Wenn ein INT auftritt, werden

```

00001 ;*****
00002 ;
00003 ;      Programm zum dreispaltigen Ausdruck von Texten
00004 ;      Wird vor dem Ausdruck über DOS-CALL aufgerufen
00005 ;
00006 ;      Arbeitet mit einem modifizierten TSCRIPS, wahrschein-
00007 ;      lich jedoch mit jedem ähnlichen Textprogramm zusammen
00008 ;
00009 ;      Modifikationen des Interpreters können, falls er in einem ROM
00010 ;      steht, durch Abfangen der Routinen in RAM-CALLs ersetzt werden.
00011 ;      Die ORGs können beliebig verlegt werden, dürfen jedoch nicht mit
00012 ;      dem Text des Editors in Konflikt geraten (geg. HIMEM ändern).
00013 ;      Das Banking über Port F3 kann je nach Hardware auf eine andere
00014 ;      Technik der Pufferung des ASCII-Druckertextes geändert werden.
00015 ;      Die Steuercodes für den NEC P6/7 müssen je nach Drucker angepaßt,
00016 ;      gelöscht oder ergänzt werden.
00017 ;
00018 ;      (C) 1987 by The HACKTORY Arnulf Sopp
00019 ;
00020 ;*****
00021 ;
4000 00022 txt_buf EQU 4000h ;in Bank 1 oberhalb Com. (geg. verleg.)
00023 ;
4601 00024 ORG 4e01h ;in der Interrupt-Serviceroutine
4601 00025 DW int ;Umleitung auf PRINT-Tastentest
00026 ;
00027 ;-----
00028 ;
3260 00029 ORG 3260h ;hinter ein paar TSCRIPS-Umleitungen
00030 ;
00031 ;die Druckdaten abfangen und stattdessen im RAM ablegen
00032 buff PUSH HL ;Register retten
00033 PUSH AF
00034 DI ;INTs aus, sonst passiert ein Unglück
00035 LD H,A ;Zeichen retten
00036 LD A,01h ;auf Bank 1
00037 OUT (0f3h),A ;umschalten
00038 LD A,H ;Drucker-Byte
00039 CP 0ah ;war es ein LF?
00040 JR NZ,byte_ok ;falls nein
00041 LD A,0ah ;evtl. LF in CR umwandeln
00042 byte_ok LD HL,txt_buf ;Zeiger auf das aktuelle Drucker-Byte
00043 point EQU $-2 ;verändert sich
00044 LD (HL),A ;puffern anstatt zu drucken
00045 INC HL ;Zeiger erhöhen
00046 LD (HL),00h ;nächste Pufferstelle vorsichtsh. löschen
00047 LD (point),HL ;und wieder merken
00048 CP 0ah ;war oder wurde es ein CR?
00049 JR NZ,cont_ok ;falls nein
00050 LD HL,(lin_cnt) ;Zeilenzähler laden
00051 INC HL ;erhöhen
00052 LD (lin_cnt),HL ;wieder merken
00053 cont_ok XOR A ;A ← 00: auf Bank 0

```

```

3285 D3F3 00054 OUT (0f3h),A ;zurückschalten
3287 F1 00055 POP AF ;Register restaurieren
3288 E1 00056 POP HL
3289 FB 00057 EI ;Interrupts wieder zulassen
328A C9 00058 RET ;weiter in TSCRIPS
00059
00060 ;-----
00061 ;
00062 ;Umleitung der INT-Serviceroutine: prüfen, ob die PRINT-Taste gedr. ist
328E 3AE038 00063 int LD A,(36e0h) ;Tastaturzeile mit PFINT-Taste
328E E680 00064 AND 80h ;wurde sie gedrückt?
3290 CABE45 00065 JF 2,45beh ;INT-Routine fortsetzen, falls nein
00066 ;-----
00067 ;
00068 ;
00069 ;die PRINT-Taste wurde gedrückt, es geht los:
00070 ;neuen Stack einrichten, Druckerumleitung rückgängig machen
3293 ED73EB32 00071 LD (sp_buff),SP ;Stackpointer wegen des Bankings retten
3297 316002 00072 LD SP,0260h ;Stack in der Überfl. Cassetten-Routine
329A 3E32 00073 LD A,32h ;Opcode LD (nn),A
329C 21E837 00074 LD HL,37e8h ;dort ist der Drucker memory mapped
329F CDF832 00075 CALL restore ;Befehl LD (37E2),A in die Druckroutine
32A2 21C233 00076 LD HL,lp_init ;Initialisierungsdaten des Druckers
32A5 CDA233 00077 CALL print ;Drucker soft resetten usw.
00078
00079 ;den Zeilenzähler auf die tatsächliche Textlänge einstellen
32A8 3ED1 00080 LD A,01h ;auf Bank 1
32AA D3F3 00081 OUT (0f3h),A ;umschalten
32AC 2A7032 00082 LD HL,(point) ;Zeiger hinter Textende
32AF 3E0D 00083 LD A,0dh ;CR als Suchbyte
32B1 47 00084 LD B,A ;BC (- mind. 0000, langt als Suchstrecke
32B2 ED89 00085 CPDR ;HL vor letzten CR des Textes stellen
32B4 23 00086 INC HL ;auf letzten CR vorstellen
32B5 ED5BE433 00087 LD DE,(lin_cnt) ;dort Anzahl Textzeilen
32E9 13 00088 INC DE ;weil die Suche mit DEC DE beginnt
32BA 01000C 00089 LD BC,0000h ;Suchstrecke für Schluß-CR vorbereiten
32BD 1E 00090 end_loop DEC DE ;Zeilenzähler erniedrigen
32BE 0C 00091 INC C ;BC (- 0001 als Suchstrecke
32BF ED89 00092 CPDR ;CR 1 Byte zuvor aufsuchen
32C1 2BFA 00093 JCR Z,end_loop ;so lange, bis anderes Byte gefunden
32C3 23 00094 INC HL ;auf ersten CR des CR-Strings stellen
32C4 13 00095 INC DE ;letzten CR noch mitzählen
32C5 EB 00096 EX DE,HL ;HL (- Anzahl Textzeilen
32C6 AF 00097 XOR A ;A ← 00: auf Bank 0
32C7 D3F3 00098 OUT (0f3h),A ;umschalten
00099
00100 ;Anzahl der Seiten berechnen, vollständige Seiten ausdrucken
32C9 3EC3 00101 LD A,195 ;= 65 Druckzeilen + 3 Textzeilen
32CB CDB44C 00102 CALL 4cb4h ;HL (- volle Seiten, A (- Restzeilen
32CE 4F 00103 LD C,A ;retten
32CF 7C 00104 LD A,H ;prüfen, ob nur eine angebr. Seite
32D0 B5 00105 OR L ;dann HL = 0000
32D1 79 00106 LD A,C ;Akku restaurieren
32D2 F5 00107 PUSH AF ;Zeilenzähler des letzten Blattes retten
32D3 44 00108 LD B,H ;BC als Seitenzähler
32D4 2B0E 00109 JR Z,1st_pag ;falls nur eine angebrochene Seite
32D6 4D 00110 LD C,L ;Seitenzähler vervollständigen
32D7 C5 00111 pag_loop PUSH BC ;retten
32DE 21C300 00112 LD HL,195 ;Zeilenzahl einer Seite
32DB CDF832 00113 CALL one_pag ;eine Seite ausdrucken
32DE C1 00114 POP BC ;Seitenzähler
32DF 0B 00115 DEC BC ;prüfen, ob alle ganzen Seiten ausgedr.
32E0 7E 00116 LD A,B
32E1 B1 00117 CP C

```

HEFT
21
September
1987

```

32E2 20F3 00118 JR NZ,pgs_lop :falls noch nicht
          00119
          00120 :letzte angebrochene Seite ausdrucken
32E4 F1 00121 lst_psg POP AF :Zeilenzahl des letzten Blattes
32E5 60 00122 LD H,B :H (- 00
32E6 6F 00123 LD L,A :HL (- Zeilenzahl des letzten Blattes
32E7 C0FF32 00124 CALL one_psg :letzte Seite ausdrucken
          00125
          00126 :Ausgang: alten Zustand restaurieren, über INT-Forts. nach TSCRIPS zurück
32EA 310C00 00127 LD SP,0000h :alten Stack restaurieren
32EB 00128 sp_buff EQU $-2 :($P bis jetzt hier gepuffert)
32ED 210040 00129 start LD HL,txt_buf :Anfangsadresse des Textpuffers
32FD 227032 00130 LD (point),HL :den Zeiger wieder auf 0 stellen
32F3 3EC3 00131 LD A,0c3h :0c0code JP nn
32F5 216C32 00132 LD HL,buff :Sprungziel
32F8 328E35 00133 restore LD (05bh),A :JP buff dort patchen
32FB 228C05 00134 LD (05bh),HL
32FE C9 00135 RET :weiter im TSCRIPS-Editor (nach INT-Ende)
          00136
          00137 :Anzahl und Layout der Druckzeilen einer Seite berechnen
32FF 23 00138 one_psg INC HL :Zähler ab 1, nicht ab 0
3300 228433 00139 LD (lin_cnt),HL :neu vermerken
3303 2B 00140 DEC HL :alter Wert
3304 3E03 00141 ED A,3 :Divisor für 3 Textzeilen pro Druckzeile
3306 C0E44C 00142 CALL 4cb4h :HL (- HL/A, A (- Rest (Textzeilen /3)
3309 45 00143 LD B,L :Anzahl der mittleren Drittelzeilen
330A 2805 00144 JR Z,div_don :falls ja
330C 23 00145 INC HL :sonst eine linke u./o. mittl. Zeile mehr
330D 3D 00146 DEC A :Divisionsrest 1? (10/3, 13/3, 16/3 usw.)
330E 2801 00147 JR Z,div_don :falls Rest 1 (nur linke Spalte länger)
3310 45 00148 LD B,L :falls Rest 2 (linke u. mittl. Sp. länger)
3311 7D 00149 div_don LD A,L :Anzahl der linken Drittelzeilen
3312 323E33 00150 LD (ln_cnt1),A :als Suchzähler patchen
3315 78 00151 LD A,B :dto. Anzahl mittlere Drittelzeilen
3316 326C33 00152 LD (ln_cnt2),A
          00153
          00154 :eventuellen Drucker-Reset in Text lösen
3319 3E01 00155 LD A,01h :auf Bank 1
331E D3F3 00156 OUT (0f3h),A :umschalten
331D 010040 00157 LD BC,txt_buf :Startadresse des ASCII-Textes in Bank 1
331E 00158 tx_beg1 EQU $-2 :wird ab 2. Seite geändert
3320 2A0040 00159 LD HL,(txt_buf) :erste 2 Drucker-ASCII-Bytes in Puffer
3321 00160 tx_beg2 EQU $-2 :verändert ab 2. Seite
3323 11C046 00161 LD DE,1bh+256*'S' :DE (- 'ESC-S' (Halb-Reset des Druckers)
3326 E5 00162 PUSH HL :retten
3327 ED52 00163 SRC HL,DE :sind die ersten beiden Bytes ein Reset?
3329 E1 00164 POP HL
332A 2E0E 00165 JR Z,ign_res :lösch. falls ja
332C 1C 00166 INC E :DE (- 'FS-S' (auch Reset, aber brutaler)
332D E7 00167 OR A :Cy löschen
332E E5 00168 PUSH HL
332F ED52 00169 SRC HL,DE :war es der Totalreset des Druckers?
3331 E1 00170 POP HL
3332 2002 00171 JR NZ,strt_ok :falls kein Reset als Textanfang
3334 03 00172 ign_res INC BC :sonst Textzeiger hinter 'Reset' stellen
3335 03 00173 INC BC :($2 Bytes für 'FS-S' bzw. 'ESC-S')
          00174
          00175 :Zeiger auf die drei Drittelzeilen errechnen.
3336 60 00176 strt_ok LD H,B :HL (- BC (Zeiger auf Textanfang.
3337 69 00177 LD L,C :evtl. hinter Reset-Sequenz)
3338 225721 00178 LD (lft_col),HL :als neuen Textanfang laden
3339 3E00 00179 LD A,00h :CR als Signal für Zeilenende
333C 0400 00180 LD B,00h :Drittelzeilen-Zähler links
333E 00181 ln_cnt1 EQU $-1 :je nach dem

```

```

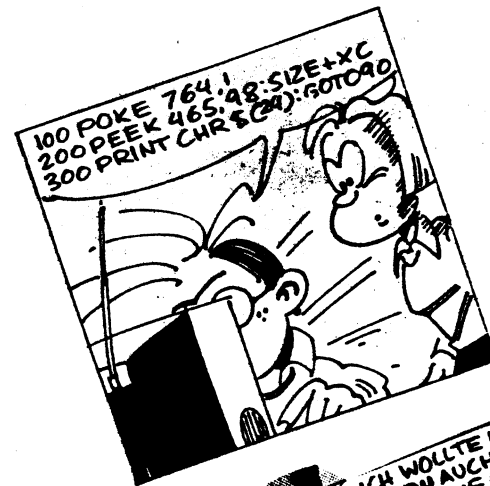
333F C0BA33 00182 CALL seek_ln :1. mittlere Drittelzeile suchen
3342 226333 00183 LD (mid_col),HL :ihren Anfang dort patchen
3345 0600 00184 LD B,00h :Drittelzeilen-Zähler Mitte
3346 00185 ln_cnt2 EQU $-1 :je nach dem
3347 C0BA33 00186 CALL seek_ln :1. rechte Drittelzeile suchen
334A 226F33 00187 LD (rgt_col),HL :dort patchen
          00188
          00189 :Text dreispaltig ausdrucken
334D 21CD33 00190 prt_lop LD HL,no_lf :Codestring für CR ohne LF
3350 C0A233 00191 CALL print :ausgeben
3353 11DE33 00192 LD DE,margins :Zeiger auf Randeinstellungen
3356 210000 00193 LD HL,0000h :Anfang der 1. Drittelzeile
3357 00194 lft_col EQU $-2 :evtl. hinter der Reset-Sequenz
3359 C0E233 00195 CALL prt_lin :eine Drittelzeile ausdrucken
335C 225733 00196 LD (lft_col),HL :Anfang der nächsten 1. Drittelzeile
335F 228133 00197 LD (sto_end),HL := evtl. Ende der Seite
3362 210000 00198 LD HL,0000h :Anfang der 2. Drittelzeile
3363 00199 mid_col EQU $-2 :je nach dem
3365 C0E233 00200 CALL prt_lin :nächste Drittelzeile ausdrucken
3368 226333 00201 LD (mid_col),HL :Zeiger nachstellen
336B 228133 00202 LD (sto_end),HL :s. o.
336E 210000 00203 LD HL,0000h :Anfang der 3. Drittelzeile
336F 00204 rgt_col EQU $-2 :je nach dem
3371 C0E233 00205 CALL prt_lin :ausdrucken
3374 226F33 00206 LD (rgt_col),HL :Zeiger nachstellen
3377 228133 00207 LD (sto_end),HL :s. o.
337A 21D133 00208 LD HL,lf :Codestring für CR mit LF
337D C0A233 00209 CALL print :ausgeben (auf nächste Druckzeile)
3380 18CB 00210 JR prt_lop :weiterdrucken
          00211
          00212 :eine Drittelzeile mit SteuerCodes ausdrucken
3382 E5 00213 prt_lin PUSH HL :Textzeiger retten
3383 010000 00214 LD BC,0000h :Zeilen-Zähler
3384 00215 lin_cnt EQU $-2 :je nach dem
3386 0E 00216 DEC BC :herunterzählen
3387 ED438433 00217 LD (lin_cnt),BC :neu laden
3388 78 00218 LD A,E :ist das Ende erreicht?
338C B1 00219 OR C
338D E1 00220 POP HL
338E 281C 00221 JR Z,end_lin :falls ja
3390 1A 00222 LD A,(DE) :Stelle des linken Randes
3391 32DC33 00223 LD (l_marg),A :in den Codestring der linken Randeinst.
3394 13 00224 INC DE :Zeiger weiterstellen
3395 1A 00225 LD A,(DE) :Stelle des rechten Randes
3396 32D933 00226 LD (r_marg),A :in den Codestring der rechten Randeinst.
3399 13 00227 INC DE :Zeiger weiterstellen
339A E5 00228 PUSH HL :Textzeiger retten
339B 21D733 00229 LD HL,rgs_str :Codestring für beide Ränder
339E C0A233 00230 CALL print :ausgeben
33A1 E1 00231 POP HL :Textzeiger
          00232
          00233 :einen String bis einschl. zum nächsten CR ausdrucken
33A2 7E 00234 print LD A,(HL) :Druck-Byte laden
33A3 23 00235 INC HL :Zeiger nachstellen
33A4 C0B405 00236 CALL 05B4h :Zeichen auf den Drucker ausgeben
33A7 FE00 00237 CF 00h :war es ein CR?
33A9 CE 00238 RET Z :fertig, falls ja
33AA 18F6 00239 JR print :sonst String weiter ausdrucken
          00240
          00241 :Rückkehr aus der Druckschleife, wenn der Zeilen-Zähler abgelaufen ist
33AC AF 00242 end_lin XOR A :A (- BC: auf Bank 0
33AD D3F3 00243 OUT (0f3h),A :umschalten
33AF F1 00244 POP AF :Stack bereinigen (CALL-Ebene löschen)
33B0 210000 00245 LD HL,0000h :Textzeiger auf Seitenende

```

```

32E1      00246 sto_end EQU $-2          ;wird jeweils gepatcht
33E3 001E33 00247 LD (tx_beg),HL        ;moment. Textzeiger als neuen Start laden
33E6 002133 00248 LD (tx_beg2),HL
33E9 C9 00249 RET
          00250
          00251 ;der Beginn einer Drittelzeile aufsuchen
33EA C5 00252 seek_in PUSH BC           ;Zeilenzähler retten
33EE 47 00253 LD B,A                  ;BC (- Suchstrecke min. 0000 (lang genug))
33EC EDE1 00254 CFIR                  ;CR aufsuchen
33EE C1 00255 POF BC                  ;Zeilenzähler
33EF 1DF9 00256 DJNZ seek_in          ;bis der Zeiger 8 Zeilen weiter steht
33C1 C9 00257 RET
          00258
          00259 -----
          00260
          00261 ;verschiedene Drucker-Steuersequenzen und Randeinstellungen
          00262
          00263 ;Druckerinitialisierung mit Reset, volles Randausgleich, skip over perf.,
          00264 ;Proportionalschrift, CR ohne LF (beim Label no_lf)
33C2 1C 00265 lo_init DB 1ch,'s',1bh,'a'.03h,1bh,'N',07h,1bh,'p',01h
          00266 ;Ann. für NEC P7: Proportionalschrift wird mit ' ', ' ', 00h angewählt
          00267
          00268 ;weitere Drucker-Steuersequenzen und linke und rechte Randspalten
33CD 1E 00269 no_lf DB 1bh,'3',00h,0dh ; CR ohne LF
33D1 1B 00270 lf DB 1bh,'1',00h,1bh,'2',0dh ;Codestr. für l. Rand = 0 & 6 lpi
33D7 1B 00271 msg_str DB 1bh,'0',00h ; rechte Randeinst.
33D9 00272 r_marg EQU $-1 ;hier Zeichenspalte des rechten Randes
33DA 1B 00273 DE 1bh,'1',00h,0dh ;Codestring für linke Randeinstellung
33DC 00274 l_marg EQU $-2 ;hier Zeichenspalte des linken Randes
33DE 07 00275 margins DB 7,29,30,52,53,75 ;li. und re. Ränder der Drittelzeilen
          00276 ;
          00277 ;
          00278 -----
          00279
10BB 00280 ORG 42ED ;im Sektorpuffer, wo es nicht stört
          00281
          00282 ;beim Einsprung: entweder Kuckucksei legen oder Programm deaktivieren
10BE CD954C 00283 entry CALL 4cd5h ;prüfen, ob das Programmnamen etwas folgt
10BB CAED32 00284 JF Z,start ;ab die Post, falls nein
10BE 21BE45 00285 LD HL,45beh ;sonst Fortsetzungsadr. der INT-Routine
10C1 22D146 00286 LD (4601h),HL ;diese restaurieren
10C4 3E32 00287 LD A,32h ;opcode LD (nn),A
10C6 21E837 00288 LD HL,37e3h ;memory mapped Adresse des Druckers
10C9 C3FE32 00289 JP restore ;Dr.-Umleitung rückgängig machen, zurÜck
          00290
          00291 -----
          00292
10BB 00293 END entry ;dort Einsprung
          00294
00000 Fehler
buff 3260 byte_ok 326F cnt_ok 3284 div_don 3311 end_lin 33AC end_lop 32BD
entry 10BE ign_res 3334 int 3286 l_marg 33DC l' 33D1 lft_col 3357
lin_cnt 3384 ln_cnt1 333E ln_cnt2 3346 lo_init 33C2 lst_pag 32E4 margins 33DE
mid_col 3363 msg_str 33D7 no_lf 33C0 one_pag 32FF pag_lop 32D7 point 327D
print 33A2 prt_lin 33E2 prt_lop 334D r_marg 33D9 restore 32F8 rst_col 336F
seek_in 338A sr_buff 32EB start 32ED sto_end 33E1 strt_ok 3336 tx_beg1 331E
tx_beg2 3321 tat_buf 46C0

```



HEFT
21
September
1987

Nachdem mein eigenes CP/M fertig war, wollte ich auch drei Programme, die ich mit ZEUS erstellt hatte, ohne große Änderungen ins CP/M übernehmen. Dazu schrieb ich mir ein Programm, das ich hier vorstellen möchte. Es ist in der Diskothek erhältlich.

Eines muß ich noch vorweg sagen: ich habe nichts gegen CPMac. Das Programm scheint echt top zu sein, aber mir ist alles suspekt, was ich bezahlen soll. Wenn ich mein Geld sparen kann, tue ich das auch.

Bei der Übertragung tun sich mehrere Probleme auf. Das erste lautet, daß ein für CP/M erstelltes Programm bei 100h beginnen muß, während im Newdos erst ab 5200h freies RAM zu finden ist. Da das Programm vor der Übertragung in CP/M ins RAM per "LOAD" geladen werden muß, darf es nicht unterhalb 5200h stehen. Ich habe mir 6000h als Basisadresse ausgesucht.

Nun kann man im Source die ersten zwei Zeilen so schreiben:

```
ORG      6000h
offset  equ  $-100h
```

und dann bei jedem Auftauchen einer absoluten (nicht relativen wie bei JR) Adresse, dieses Label "offset" wieder davon abziehen. Das sieht dann zum Beispiel so aus:

```
...      JP      start-offset
...
start    LD      HL,meldung-offset
...
```

Allerdings vergesse ich leicht mal ein Offset, was der Assembler nicht als Fehler meldet, aber meist zum Programmabsturz führt.

Also ziehe ich eine andere Möglichkeit vor: die erste Zeile meines Programmlautet:

```
ORG      100h
```

also ganz normal für ein CP/M-Programm. Nur kann dieses Programm nach dem Assemblieren nicht geladen werden, sonst kommt irgendeine Fehlermeldung oder ein Super-Absturz.

Mein Programm "shift/cmd" arbeitet nun in mehreren Phasen. Phase 1 lädt mein CP/M (das als GDOS-CMD-File vorliegt) in das obere RAM. Eigentlich brauche ich nur das BIOS, wie Ihr gleich sehen werdet, aber das hatte ich gerade nicht einzeln rumliegen.

Am Anfang von Phase 2 fragt "shift" nach dem zu übertragenden Programm. Dieses wird nun in einen Buffer bei 6000h geladen. Dabei tut "shift" einfach so, als wäre 6000h gleichbedeutend mit 100h, denn so lautet die Ladeadresse des Programms (sonst ist es nicht unter CP/M lauffähig).

In Phase 3 wartet "shift" nun darauf, daß eine CP/M-Systemdiskette in Laufwerk 0 gepackt wird. Dann schaltet es um auf durchgehendes RAM und schiebt das bei 6000h stehende Programm in seinen endgültigen Speicherbereich bei 100h (wo es auch eigentlich stehen/laufen sollte). Dann springt "shift" das BIOS an und führt somit einen Cold Start durch. Dabei wird der Speicherbereich von 100h bis zum Anfang des CCP nicht gelöscht und somit steht nach dem Cold Start das Programm vollständig in der TPA. Es kann nun mit "SAVE xx test.com" gesichert werden. xx ist dabei die Länge des Programms in 256-Byte-Sektoren, die "shift" auch nochmal meldet, damit man möglichst nichts falsch machen kann.

Natürlich hat dieses System gegenüber CPMac mehrere Schwächen:

1. Ihr habt kein cpm/cmd. Das braucht Ihr auch nicht, wenn Ihr nur ein BIOS habt (reicht, wenn dieses beim Cold Start sowohl CCP als auch BDOS lädt). Wenn das zu übertragende Programm nicht zu lang ist, kann man "shift" auch in einer Endlosschleife enden lassen und dann per RESET booten. Allerdings wird dann zumindest der Speicherbereich ab 4200h verändert (das Programm darf nicht länger als 65 Sektoren bzw. 16,25 KByte sein).

2. Das zu übertragende Programm ist zu lang. Da kann ich auch nicht helfen. Wendet Euch an CPMac. Allerdings ist es sowieso nicht sinnvoll, große Programme für CP/M unter ZEUS zu erstellen. Das lohnt sich nur bei kleinen Sachen, bei denen es zu aufwendig ist, erst Wordstar und den ganzen Kram zu bemühen.

Für mich hat sich das Programm gelohnt, denn so brauche ich mir nicht CPMac anzuschaffen (um es dann dreimal im Jahr zu benutzen) und kann trotzdem schnell mal was mit ZEUS programmieren und unter CP/M laufen lassen.

Gerald Schröder



Für den Heimwerker: Wie baue ich mir ein CP/M?

Dieser Artikel ist nicht ...

... eine Einführung in CP/M

... eine Beschreibung von CP/M

... eine konkrete Bauanleitung für ein CP/M

sondern eine allgemeine Beschreibung, wie man sich ein CP/M aus Newdos heraus bauen kann bzw. wie ich dieses gemacht habe. Wenn ich Euch genau schildere, wie es aussieht, bringt Euch das fast nichts, denn mein Genie IIs läuft anders als Eure Rechner. Interessanter ist m.E. viel mehr der Weg zu einem eigenen CP/M. Der Artikel richtet sich an Leute, die kein CP/M haben und solche, denen ihr CP/M nicht ausreicht. Assembler-Kenntnisse sind mitzubringen.

1. Die Werkzeuge

Sie wird in einem anderen Artikel beschrieben worden. Alle Sources liegen unter ZEUS vor. Einige Hilfsprogramme, die ich selbst geschrieben habe, werde ich gesondert vorstellen.

2. Die Theorie des CP/M

Hinter CP/M steht eine ziemlich ausgefeilte Philosophie, die es ermöglicht, daß es auf möglichst vielen Rechnern läuft. Bevor man sich mit dem Bau (der Praxis) beschäftigt, muß erst die Theorie bekannt sein. Dazu gibt es zwei Wege: a) den Lehrer, b) das Lehrbuch.

Ich empfehle die erste Möglichkeit. Sie sieht folgendermaßen aus: Ihr sucht Euch einen Typen, der möglichst in der Nähe wohnt und außerdem etwas Ahnung von CP/M hat. Solche Leute wie Hartmut sind ein gutes Ziel. Nun beginnt die "Löcherphase". Ihr bohrt in (bzw. an) ihm rum, bis er Euch eine kostenlose Einführung in CP/M angedeihen läßt. Oft sind solche Leute auch froh, mal ihr Wissen präsentieren zu können, so daß die Löcher flach sein können. Aber den Bohrer noch nicht absetzen! Nun muß nämlich auch noch das letzte an Substanz aus dem Loch geholt werden. Entweder kann der Typ so gut erklären, daß Du danach besser Bescheid weißt als die Herren von Digital Research oder Du stehst so lange vor seiner Haustür, bis er Dich mit allen Materialien zu CP/M (Büchern, Fotokopien, Zeitschriften, Sources) bewirft, die er sowieso schon lange loswerden wollte. Die sammelst Du schön säuberlich auf und studierst im kleinen Kämmerchen weiter, bis Du überschlau bist.

Die zweite Möglichkeit ist der zweite Teil der ersten: Du studierst von vornherein nur alles, was Du Dir zusammenschmorren oder Dein Geldbeutel sich leisten kann. Sobald Du weißt, was CCP, BDOS, BIOS, Sprungleiste (JP-Vektoren, BIOS-Aufrufe), DPH, DPB, XLT (und was es sonst noch an tollen Abkürzungen gibt) sind, bist Du für den nächsten Schritt gerüstet.

3. Die Ausgangslage

Beim Sichten der verfügbaren (zusammengeschnorren, hinterhergeworfenen) Materialien wirst Du zweierlei feststellen: erstens geht jedermann davon aus, daß Du ein CP/M benutzt, wenn Du eines bauen willst; zweitens sind die meisten Beispiele entweder in 8080-Assembler oder unvollständig oder für ein 8-Zoll-SS/SD-Diskettensystem oder gar nicht vorhanden. Aber was hast Du? Ein Newdos, eine zusammengewürfelte (hochwertige) Hardware, einen Z80-Assembler, kein lauffähiges CP/M (willst Du ja erst bauen), keine Ahnung, aber große Pläne.

Was brauchst Du? Ein CP/M bestehend aus CCP und BDOS unter Newdos (als Source-Code); ein möglichst gut dokumentiertes Beispiel-BIOS (Dein eigenes mußt Du Dir bauen); ein Format-Programm; ein Programm, das Dein CP/M als System auf eine Diskette packt.

4. Wie bekommst Du was?

Fangen wir ganz vorne an. Die Paperware sollten Dir Club-Kameraden zur Verfügung stellen können. Eine gute Adresse wird hoffentlich bald unsere Club-Bibliothek sein.

Das CP/M ist ein Problem, denn Digital Research hat das Copyright. Allerdings gibt es schon Public-Domain-CP/Ms (z.B. für Atari) und ein komplettes Listing (Röckrath), so daß diese Sache wohl nicht mehr so heiß gegessen wird (wenn unser aller Freund Schmidt nicht die Rechte erwirbt). Am besten holst Du es Dir von jemandem, der es Dir unter Newdos geben kann. Bei der Source-Anpassung, damit es relokatable wird (also überall laufen kann), kann ich helfen.

Dann wäre ein kommentiertes BIOS für eine möglichst ähnliche Maschine zu besorgen. Empfehlen kann ich das Montezuma-BIOS für das Modell IV bzw. die Dokumentation dazu, die ein fast vollständiges Listing des BIOS enthält. Die fehlenden Teile kann ich liefern (auch mit Kommentar). Außerdem wird das von mir erstellte BIOS in der Diskothek verfügbar sein.

Das Formatierungs- und das Systemgenerierungs-Programm müssen neu programmiert werden. Beispiele werden in der Diskothek zu finden sein (format/cmd und putsys/cmd).

Natürlich solltet Ihr die Hardware Eures Rechners gut kennen und gute Unterlagen darüber besitzen. Geeignet für das Schreiben der Disketten-Ein-/Ausgabe ist zum Beispiel das DOS-Buch von Grosser.

5. Die Grundgedanken

Das CP/M-System soll vollkommen vom Newdos aus hergestellt werden, denn das ist unsere "normale" Systemoberfläche, deren Werkzeuge wir auch am besten kennen.

Die Reihenfolge der Programmierung sieht folgendermaßen aus: zuerst das BIOS erstellen, dann eine Diskette mit dem eigenen Format-Programm formatieren und zuletzt das ganze CP/M-System (Bootsektor, CCP, BDOS, BIOS) auf diese Diskette packen, die dann eine bootbare CP/M-Systemdiskette sein wird.

Das BIOS ist einigermaßen umfangreich und sollte möglichst sicher funktionieren, denn es ist als Ganzes nur schwer zu testen. Deshalb halte ich die Programmier-Methodik der Montezuma-Hersteller für unerlässlich: das Programmieren in Modulen. Das heißt, daß das BIOS in mehrere Teile (z.B. Bildschirmausgabe) aufgespalten wird, die Ihr nun getrennt programmiert und testet.

Erst wenn alle (oder möglichst viele) Module richtig funktionieren, macht Ihr Euch an die Programmierung des Format-Programms, denn Ihr könnt viel von dem Wissen, daß Ihr vorher gesammelt habt, und auch mindestens eines der Module (Disk-Ein-/Ausgabe) bei diesem Programm verwenden. Ebenso steht es mit dem Systemgenerierungsprogramm.

HEFT

21

September

1987

6. Die Moduln

Von Montezuma und mir sind die Namen vieler Moduln und auch eine Reihe von Labels vorgegeben, die Ihr weiter verwenden solltet, denn damit spart Ihr Euch eine Menge Arbeit. Ein großer Vorteil der Modul-Programmierung ist, daß Ihr langsam vorgehen und die Anforderungen dadurch in Grenzen halten könnt. Es steigert unheimlich die Motivation, wenn das erste Modul funktioniert. Das Druckerausgabe-Modul dürfte das einfachste sein, also fangt damit an. Dann geht Ihr zur Bildschirmausgabe über usw. Als letztes nehmt Ihr Euch die Disketten-Ein-/Ausgabe vor. Sie dürfte am schwersten sein. Allerdings solltet Ihr Euch jetzt überlegen, was Euer CP/M können soll, d.h. welche exotischen Diskettenformate von ihm gelesen werden sollen. Je nachdem ist mehr oder weniger Aufwand an dieser Stelle erforderlich. Mein CP/M-BIOS kann erheblich weniger als das von Montezuma, aber der größte Teil davon ist für mich einfach überflüssig. Zum Beispiel hat mein System nur zwei Diskettenlaufwerke und wird wohl nie eine Festplatte bekommen, warum also mehr als zwei Laufwerke berücksichtigen? An anderer Stelle habe ich ebenso auf die Implementierung des ID-Byte verzichtet, das zwar sehr schön sein mag, mir aber nichts bringt. Viele Sachen sind am Anfang einfach auch unsinnig und können bequem von Euch nachgerüstet werden, wenn die Version 1.0 Eures CP/M einmal läuft.

Bei der Programmierung und vor allem dem Test der einzelnen Moduln müßt Ihr darauf achten, welche Variablen und Routinen von zwei Teilen gemeinsam benutzt werden und diese Labels dann irgendwie dem gerade in Arbeit befindlichen Modul bekanntgeben (z.B. eine Routine, die nur aus einem RET besteht, beim Test einbauen). Jedes Modul wird einzeln getestet. Dazu schreibt Ihr kurze Test-Programme, in die Ihr die Routinen einbindet und dann prüft, ob das richtige herauskommt, wenn Ihr etwas bestimmtes eingibt. So läßt sich zum Beispiel hervorragend testen, ob die Tastatur-Abfrage (conin) die richtigen Werte ergibt oder die Bildschirmsteuercodes auf dem Bildschirm die richtigen Reaktionen hervorrufen (conout).

Und gleich noch ein Vorteil: wenn Ihr schon irgendwo bestimmte Programmteile habt, die sich verwenden lassen, baut Ihr sie einfach als Modul ein! Wiederum könnt Ihr viele der BIOS-Moduln in anderen Programmen verwenden, z.B. die Disketten-Ein-/Ausgabe für format/cmd und putsys/cmd.

7. Das Linken der Moduln

Diese Sache sollte nicht zu leicht genommen werden! ZEUS hat hervorragende Fähigkeiten zum Linken. Ihr schreibt einfach einen Hauptfile (bei mir heißt er master/src), der alle anderen Bestandteile per GET-Kommando holt. Naturgemäß werdet Ihr dabei feststellen, daß irgendwelche Labels nicht definiert wurden und deshalb geändert, eingefügt oder sonstwas werden müssen. Dann müßt Ihr erkunden, wie groß Euer BIOS im Endeffekt ist und je nachdem die Basisadresse des CP/M nach unten verschieben. Auch kein größeres Problem: in master/cmd fügt Ihr einfach eine EQU \$-Variable direkt vor dem END ein, deren Wert FFO0h nicht übersteigen darf (der Rest ist Stack). Den Wert der Variablen könnt Ihr Euch per Kommando "Blabel" ansehen.

Nachdem das System nun vollständig in den Speicher paßt und keine Assemblierungsfehler mehr auftreten, könnt Ihr den File cpm/cmd erzeugen. Das dauert zwar etwas, aber jetzt habt Ihr Euer CP/M!

8. Das Programm format/cmd

Dieses Programm soll eine Diskette für Euch formatieren, und zwar in dem Format, das Ihr für Euer System bevorzugt. Die Format-Routine des Newdos ist nicht geeignet, denn wir brauchen ja kein Directory und außerdem sind 256-Byte-Sektoren nicht unbedingt das Optimum. Und Ihr solltet noch etwas weiter denken: Euer CP/M soll später auch mal Disketten formatieren können, also schreibt Ihr hier ein Programm, das sich später leicht auf CP/M übertragen lassen sollte, um für Euch dort die Disketten zu formatieren.

Beim Formatieren solltet Ihr eins beachten: das Füllbyte sollte (muß?) E5h sein. Ich habe es mit dem DD-Füllbyte 6DB6h versucht, was zur Folge hatte, daß mein CP/M später meldete "No file" (im Directory, ist OK), aber: "Diskette full". Eine volle Diskette ohne einen einzigen File drauf gibt es nicht. Es liegt allein am Füllbyte (schätzungsweise Bit 7 oder so, fragt jemand, der sich mit CP/M auskennt).

9. Das Programm putsys/cmd

Wenn Ihr eine formatierte Diskette und cpm/cmd habt, sollte irgendwie cpm/cmd bootfähig auf die Diskette gepackt werden. Dies erledigt putsys/cmd. Allerdings müßt Ihr dazu noch ein Modul erstellen, und zwar den Bootsektor. Das Ding sitzt auf Track 0, Sektor 0 (manchmal auch 1) und wird von Eurem ROM nach 4200h gezogen, dort gestartet und soll dann Euer CP/M irgendwie laden und zum Laufen bewegen. Es kann sein, daß Ihr dazu Track 0 nochmal neu formatieren müßt, wenn Euer ROM sonst Mist baut. Mein IIS liest jedenfalls alles, was man ihm anbietet, nur muß ein bestimmtes Byte den Wert 00 haben. Bootsektoren gibt's wieder bei Grosser zur Ansicht. Ansonsten wäre eine Newdos-Systemdiskette auch eine gute Informationsquelle.

Bei meinem CP/M lädt der Bootsektor das BIOS an den richtigen Platz und startet dieses. Das BIOS wiederum lädt erst CCP und BDOS und springt dann den CCP an. Das könnt Ihr auch anders lösen, aber bei mir bedeutet das, daß auf der Systemdiskette stehen müssen:

Track 0, Seite 0, Sektor 0: Bootsektor

Track 0, Seite 0, Sektor 1-3: BIOS

Track 0, Seite 0, Sektor 4, und Seite 1, Sektor 0-5: CCP und BDOS.

Dabei solltet Ihr bedenken, daß jeder Sektor 1024 (!) Byte lang ist.

Bei Euch kann das ganz anders aussehen, aber ich habe gute Erfahrungen mit dieser Anordnung gemacht.

putsys/cmd funktioniert bei mir folgendermaßen: zuerst nulle ich den Speicher von 5200h aufwärts (Befehl "0" im Gdos). Dann lade ich das CP/M: LOAD CPM/CMD. Nun wird putsys/cmd gestartet: PUTSYS/CMD. putsys wartet darauf, daß ich die formatierte CP/M-Diskette in Lw. 0 einlege, schreibt dann Bootsektor, BIOS, CCP und BDOS auf die Diskette und startet zuletzt das BIOS (oder wartet auf die Newdos-Diskette oder auf RESET).

Eine abgeänderte Form des putsys/cmd braucht Ihr übrigens wieder unter CP/M, denn Ihr wollt ja nicht alle Systemdisketten vom Newdos aus erzeugen.

10. Die letzten Tests

Die letzten Tests sind eigentlich nie zu Ende, denn irgendwo taucht immer was auf, das Euch nicht gefällt. Bei mir dauerte es drei Tage, bevor ich die erste CP/M-Diskette booten konnte, nachdem das System fertiggestellt war, denn ein Befehl der Cold-Start-Routine war falsch. Also nicht verzagen, wenn nicht alles gleich funktioniert.

Weitere Feinarbeiten werden sich enlos hinziehen, aber wenigstens könnt Ihr sie durchführen, während Ihr Euch bei anderen (gekauften) Systemen machtlos mit den Fehlern rumärgern müßt!

11. Was unter CP/M als erstes zu tun ist

Drei Programme hängen unter CP/M allein von Eurer Hardware ab und müssen deshalb möglichst bald erstellt werden. a) format.com, b) sysgen.com, c) pdrive.com.

format.com leistet dasselbe wie format/cmd, nur unter CP/M. Es formatiert eine Diskette; wenn Ihr Lust habt, auch in den blödesten Formaten (35 Tracks, 8S/SD, 128 Bytes/Sektor).

sysgen.com ähnelt stark dem putsys/cmd, sollte aber etwas bearbeitet werden. Im Grunde könnt Ihr zwar das "laufende" CP/M-System auf die Diskette dumpen (natürlich mit Bootsektor), aber ich halte es für besser, das System frisch von einer Systemdiskette zu laden und dann auf eine neu formatierte Diskette zu feuern. TCS formatiert übrigens von putsys.com aus nochmal die ersten beiden (System-) Tracks in einem Standard-Format. Dann können die exotischsten Disketten mit einem System ausgerüstet werden, aber wer will das? Mir reicht ein Format.

pdrive.com ist vollkommen neu zu erstellen. Es soll Euch helfen, Disketten in anderen Formaten zu lesen. Theoretisch könnt Ihr die Werte auch per DDT oder so ändern, aber viel schöner ist es doch, wenn Ihr nur gefragt werdet, wieviele Tracks usw. die zu lesende Diskette hat und den Rest Euer Programm erledigt. Eine Rumpf-Version findet Ihr in der Diskothek. Sie ist nicht sehr komfortabel, aber wenn ich mal gut drauf bin, werde ich sie verbessern.

Wie Ihr Programme für CP/M mit ZEUS erstellt und dann unter CP/M laufen laßt, erzähle ich Euch in einem anderen Artikel.

12. Schlußwort

Das war ein verdammt langer Artikel, wenn er auch nicht an Arnulfs und Helmut's Produktionen heranreicht (vor allem in der Tiefe). Wenn Ihr Euch bis hierhin durchgekämpft habt und irgendwann mal ein CP/M bauen wollt, stellen sich Euch vielleicht Probleme, die ich nicht vorausgesehen und nicht angesprochen habe. Schreibt mir dann einfach und ich werde versuchen, Euch zu helfen.

Gerald Schröder

MS-DOS Ecke

=====

Nachdem eine rege Diskussion offensichtlich gewünscht wird und im Gange ist, möchte ich auch noch einige Worte sagen. Über MS-DOS selbst braucht man nicht viel sagen, Höhen und Tiefen sind hinreichend bekannt.

Zur Frage, ob eine MS-DOS Ecke im CLUB80-Info erwünscht ist, sollte man einerseits sehen, daß es bereits zahlreiche Zeitschriften mit MS-DOS Veröffentlichungen gibt, von den Sachbüchern ganz zu schweigen. Ist dann der Platz hierfür in unserem Info nicht zu schade, zumal sicherlich viele der Clubmitglieder auch andere Zeitschriften lesen, in denen MS-DOS Probleme angesprochen werden? Wenn nun eine MS-DOS Ecke eingerichtet wird, kommt dann auch bald eine UNIX, OS/2 oder sonstige Ecke, die eigentlich mit TANDY/GENIE nichts gemein hat? Ich finde diesbezüglich eben, daß wir darauf achten sollten, die "80" in unserem Clubnamen nicht zu vergessen. Das Ziel sollte meiner Meinung nach sein, daß es im CLUB80 auch in einigen Jahren noch Anregungen und Tips für TANDY-Anwender geben sollte.

Auf der anderen Seite gibt es zwischenzeitlich doch einige Verbindungen zwischen MS-DOS und TANDY/GENIE-Computer, z.B. das im letzten Info bereits angesprochene TurboPascal, Datentransferprogramme, Model 4-Emulator für MS-DOS etc. Und sicher gibt es auch einige Clubmitglieder die mit beiden Systemen arbeiten.

Eine MS-DOS-Ecke könnte deshalb schon ein Plätzchen im Info finden. Die Beiträge sollten sich dann aber auf Gemeinsamkeiten beziehen und nicht etwa Einführungen oder Tips für MS-DOS beinhalten.

Wir sollten auf jeden Fall darauf achten, daß wir nicht dem Beispiel von Computerzeitschriften folgen, die 80er Anwendungen so nach und nach verdrängen.

Klaus Hermann

HEFT
21
September
1987

Model 4 - Info
=====
Nachstehend ein kleiner Patch, der bewirkt, daß bereits vom Systemstart an der Großbuchstaben-Modus eingeschaltet ist. Eine manuelle Umschaltung per CAPS-Taste entfällt somit.
PATCH SYS0/SYS.LSIDOS (D00,80=20:F00,80=00)
Klaus Hermann

Turbo Tip

Als Fachliteratur für

Viele Grafikkarten (so auch die HRG des TRS 80 Model 4) erlauben es, den Textbildschirm ganz abzuschalten bzw. auszublenken. Tritt bei der Erstellung von Grafikprogrammen unter Turbo Pascal ein Runtime-Error auf nachdem die HRG ein- und der Textbildschirm ausgeschaltet wurde, steht man im wahrsten Sinne des Wortes im Dunkeln, da Turbo den Textbildschirm nicht automatisch wieder einblendet. Mit folgendem Patch kann man Turbo Pascal dazu bewegen, dies doch zu tun!

Ab Adresse 016Bh* steht der Initialisierungsstring, den man bei der Installation von Turbo Pascal angeben kann. Dieser String wird unter anderem jedesmal dann an das Terminal geschickt, wenn Turbo nach einem Laufzeitfehler aus dem laufenden Programm zurückkehrt. Die Routine, die das besorgt, steht ab 030Ah und beginnt mit LD HL,016Bh. Diese Anweisung ersetzt man durch einen CALL xxxx, wobei xxxx die Adresse der eigenen Routine zur Umschaltung auf den Textbildschirm ist. Platz für die Routine findet man z.B. bei 2174h (Copyrighteintrag von Borland), wo auch Jochen Tucht schon seine Patches untergebracht hat (siehe PASCAL international Heft 7/87). Beim Model 4p sieht die Routine wie folgt aus:

```
LD    A,0FCh
OUT   (83h),A
LD    HL,016Bh
RET
```

Wer aufgepaßt hat, dem ist sicher aufgefallen, daß die vorletzte Instruktion genau der entspricht, die man bei 030Ah durch den CALL auf die eigene Routine überschrieben hat. Während das Unterprogramm von Rechner zu Rechner unterschiedlich sein kann, muß die letzte Aktion vor der Rückkehr (RET) ins Hauptprogramm immer die gleiche sein, sonst passieren (je nach dem Wert von HL) unvorhersehbare Dinge!

Daß die Änderung am besten mit einem Debugger durchgeführt wird, mag eingefleischte Hochsprachenprogrammierer vielleicht abschrecken, trotzdem wünsche ich viel Spaß und Erfolg beim Erstellen von Grafiken mit Turbo Pascal!

Rammut Obermann

*Alle gemachten Angaben beziehen sich auf Turbo Pascal 3.0 unter CP/M!

Hier ein kleines Programm, mit dem sich Turbo-PASCAL selbst patcht!

```
const
  a: array [1..8] of byte =
    ($3E,$FC,$D3,$83,$21,$6B,$01,$C9);
  (LD    A,252
   OUT   (131),A
   LD    HL,016Bh
   RET)
```

```
begin
  move(a,mem[$21AB],8);
  mem[$030A]:= $CD; {CALL 21ABh}
  mem[$030B]:= $AB;
  mem[$030C]:= $21;
end.
```

Turbo Patch

In der Zeitschrift PASCAL international Heft 5 und 7/87 stellte ein Herr Tucht zwei Patches vor, die dem Turbo-PASCAL-Compiler zu noch mehr Benutzerfreundlichkeit verhelfen sollen. Leider sind Herrn Tucht ein paar kleine aber fatale Fehler unterlaufen, die ich in dem folgenden Patchprogramm beseitigt habe.

```
{Patch's fuer Turbo-Pascal 2.0 und 3.0 aus PASCAL international 5 und 7/87}
{Patch 1 dient dazu, Turbo-Pascal durch R.COM wieder aufrufbar zu machen,}
{ohne den gerade editierten Text zu verlieren. Patch 2 stellt zusaetzlich}
{noch die Moeglichkeit zur Verfuegung, den Namen des zu editierenden Files}
{schon beim Aufruf von Turbo mit anzugeben (z.B. TURBO TEST.PAS)}
{Vorgehensweise Patch 1: TURBO mit oder ohne Fehlermeldungen starten}
{
  TP-PATCH.PAS laden
  nicht benoetigte Zeilen loeschen oder einklammern}
{ TP-PATCH mit R-Option starten}
{ W-Option aufrufen und auf die Frage nach dem neu-}
{ en Workfile mit ENTER antworten}
{ TURBO mit Q verlassen und wie folgt abspeichern:}
{ TURBO 2.0 ohne Fehlermeldungen - SAVE 122 T2.COM}
{ TURBO 2.0 mit Fehlermeldungen - SAVE 127 T2.COM}
{ TURBO 3.0 ohne Fehlermeldungen - SAVE 123 T3.COM}
{ TURBO 3.0 mit Fehlermeldungen - SAVE 129 T3.COM}
{Vorgehensweise Patch 2: genau wie bei Patch 1! Es ist zu beachten, dass}
{ Patch 2 an einem noch nicht mit Patch 1 behandel-}
{ ten TURBO vorgenommen werden muss. Wurde TURBO}
{ schon mit Patch 2 behandelt, so ist die Funktion}
{ von Patch 1 automatisch schon enthalten!!!}
}
```

```
{Patch 1 fuer TURBO 2.0}
```

```
BEGIN
  MEM [$205D]:= $C3;
  MEM [$205E]:= $38;
  MEM [$205F]:= $21;
END.
```

```
{Patch 1 fuer TURBO 3.0}
```

```
BEGIN
  MEM [$2174]:= $C3;
  MEM [$2175]:= $38;
  MEM [$2176]:= $22;
END.
```

```
{Patch 2 fuer TURBO 2.0}
```

```
const
  a: array [1..45] of byte =
    ($06,$0B,$21,$6B,$00,$2B,$7E,$FE,$3F,
     $28,$1F,$10,$F8,$FE,$20,$2B,$19,$21,
     $65,$00,$7E,$FE,$20,$20,$0B,$3E,$50,
     $77,$23,$3E,$41,$77,$23,$3E,$53,$77,
     $CD,$5B,$2C,$CD,$01,$24,$C3,$35,$21);
```

```
begin
  move(a,mem[$205D],45)
end.
```

```
{Patch 2 fuer TURBO 3.0}
```

```
{Geaendert am 16.07.87 von Rammut Obermann}
{Der Fehler bei der Option X "(.OVR not found" und das Nachladen}
{der schon in TURBO.COM enthaltenen Fehlermeldungen wurde beseitigt!}
```

```
const
  a: array [1..53] of byte =
    ($06,$0B,$21,$6B,$00,$2B,$7E,$FE,$3F,
     $28,$1F,$10,$F8,$FE,$20,$2B,$19,$21,
     $65,$00,$7E,$FE,$20,$20,$0B,$3E,$50,
     $77,$23,$3E,$41,$77,$23,$3E,$53,$77,
     $CD,$3A,$2D,$CD,$F9,$24,$C3,$3B,$22,
     $54,$55,$52,$42,$4F,$20,$56,$33);
```

```
begin
  move(a,mem[$2174],53);
  mem[$2b37]:= $a1;
  mem[$2dab]:= $a1;
  mem[$2da4]:= $c9;
end.
```

dBase II

Eine kurze Einleitung in das Datenbanksystem

v. Harald Nand

Bereits vor 2 Jahren machte ich Bekanntschaft mit diesem Datenbanksystem. In unserer Firma sollte ein Verwaltungssystem für Meßmittel auf einem PC erstellt werden. Ein Informatiker erstellte das entsprechende Programm mit dBase II. Als ich mich näher damit befaßte, stellte ich fest, dieses Datenbanksystem ist äußerst leistungsfähig und eignet sich für viele Verwaltungsaufgaben. So habe ich selbst angefertigte Programme in dBase II zu schreiben und im Laufe der Zeit entstanden eine Menge Programme. Das letzte Programm wurde auf meinem Homecomputer (Tandy Modell 4P) geschrieben und ist eine Vereinsverwaltung für einen Sportverein.

Was ist nun dBase II? Es ist ein rationales Datenbanksystem wo der Benutzer schon mit relativ wenig Anweisungen in der Lage ist, Daten zu erstellen und zu bearbeiten. Zur Erstellung von kompl. dBase Programmen ist es nur ein kleiner Schritt weiter.

1. Erstellung einer Datenbank:

Eine dBase II Datenbank ist leicht durch den Befehl:

```
" CREATE < Dateiname > "
```

zu erstellen. Es müssen dann jedoch noch folgende Dinge eingegeben werden: Name, Typ, Größe und Dezimalstellen.

Zur Bearbeitung wird eine dBase II Datenbank mit dem Befehl
" USE < Dateiname > " geöffnet.

Sehr oft arbeitet man bei dBase mit Indexdateien, da die Datenfelder hier sortiert gespeichert sind z.B. bei Adressendateien nach Nachname, Wohnort, Alter etc. In dBase können bis zu 7 Indexdateien verarbeitet werden, wodurch sich die Verarbeitungsgeschwindigkeit verkürzt. Wichtig bei Indexdateien ist, daß sie vor dem Aufruf reorganisiert sein muß, da es sonst zu Fehlermeldungen (" RECORD OUT OF RANGE ") kommt.

Dateistrukturen können leicht durch den Befehl:

```
" MODIFY STRUCTURE "
```

geändert werden. Jedoch Vorsicht! Dieser Befehl löscht alle bereits gespeicherten Daten. Vor einer Strukturänderung sollten die Daten in eine Hilfsdatei gespeichert werden.

z.B. " USE < Dateiname > "

```
" COPY TO HILF " ==> Hilfsdatei
```

```
" MODIFY STRUCTURE " ==> Ändern der Struktur
```

```
" APPEND FROM HILF " ==> Rücktransfer
```

Eine Einsicht in die Datenbank bekommt man nach dem Aufruf des Dateinamens durch die Befehle:

```
" LIST ALL " (jedoch klappt hier der Bildschirm durch) oder
```

```
" DISPLAY ALL " (es wird nur eine Bildschirmseite eingeblendet)
```

Selbstverständlich können auch Daten in einer Datei sehr einfach geändert werden. Dieses erfolgt nach dem Aufruf des Programmnamens durch den Befehl:

```
" BRONCE "
```

Der Cursor steht jeweils auf dem Datensatz und alle Daten in der ganzen Datenbank können so leicht geändert werden. Der Befehl

```
" EDIT 100 "
```

erlaubt dagegen nur eine Änderung des entsprechenden Datensatzes (hier Satz 100).

Datensätze können mit dem Befehl:

```
" DELETE RECORD < Nr. > "
```

gelöscht werden. Dabei wird der angesprochene Datensatz jedoch noch nicht endgültig gelöscht, sondern nur zum Löschen markiert. Die tatsächliche Löschung erfolgt erst mit dem Befehl:

```
" PACK "
```

Dabei sollten allerdings mit der Datei auch die entsprechenden Indexdateien eröffnet sein, da diese aktualisiert werden müssen. Bei unterschiedlichem Aktualisierungsgrad von Datei und Indexdatei erfolgt sonst die Fehlermeldung:

```
" RECORD OUT OF RANGE "
```

Sollte diese Fehlermeldung auftauchen muß die Datei mit folgenden Befehl neu indiziert werden.

```
" USE < Dateiname > "
```

```
" INDEX ON NACHNAME TO NACHNAME " (Indexdatei ist nach Nachname geordnet)
```

Zum Löschen markierte Datensätze können leicht mit dem Befehl:

```
" RECALL NEXT 1 "
```

wieder aktiviert werden.

Man sollte immer wieder prüfen, ob die Datensätze endgültig gelöscht werden müssen, oder noch an anderer Stelle gebraucht werden (z.B. Beispiel bei der Vereinsverwaltung für eine Kündigungsliste).

Komplette Dateien können mit dem Befehl:

```
" DELETE < dateiname > "
```

gelöscht werden.

2. Sortieren und Indizieren von dBase II - Dateien:

Der Befehl:

```
" INDEX ON NACHNAME TO NACHNAME " indiziert eine Datei nach dem angegebenen Datenfeld in die Datei Nachname und stellt die Datensätze nach nach diesem Feld sortiert zur Verarbeitung bereit.
```

Mit den Befehlen:

```
" GO BOTTOM und GO TOP
```

kann zu dem ersten und letzten Datensatz einer Datei gesprungen werden, anschließend können mit dem Befehl:

```
" SKIP-1 "
```

die Datensätze ab dem letzten Datensatz solange angezeigt werden, bis mit der Recordnummer 0 der Anfang der Datei erreicht ist.

3. Beenden von dBase II:

dBase II wird verlassen durch den Befehl:

```
" QUIT "
```

und kehrt zum Betriebssystem zurück.

Das die erste kurze Einleitung in dBase II. Wir können jetzt eine Datenbank erstellen, verändern, Indexdateien eröffnen und Daten löschen. Für eine erste Einleitung soll es genug sein. Im nächsten Info geht es dann weiter mit der Erstellung von Programmen, Editierbefehlen, Programmschleifen, Bildschirm- und Druckersteuerungen usw.

Falls jemand sich näher damit auseinandersetzen will stehen ich gerne mit Rat und Tat zur Verfügung. Im Moment bin ich dabei mich mit dBase III plus auseinanderzusetzen, da ich in der Firma einen neuen Rechner (Siemens PC 16-20 = AT Kompatibel) habe.

Als Fachliteratur für dBase II kann ich folgende Bücher empfehlen :

1. Das Datenbanksystem dBase II v. Dr. Peter Albrecht
(Verlag Markt und Technik)

und

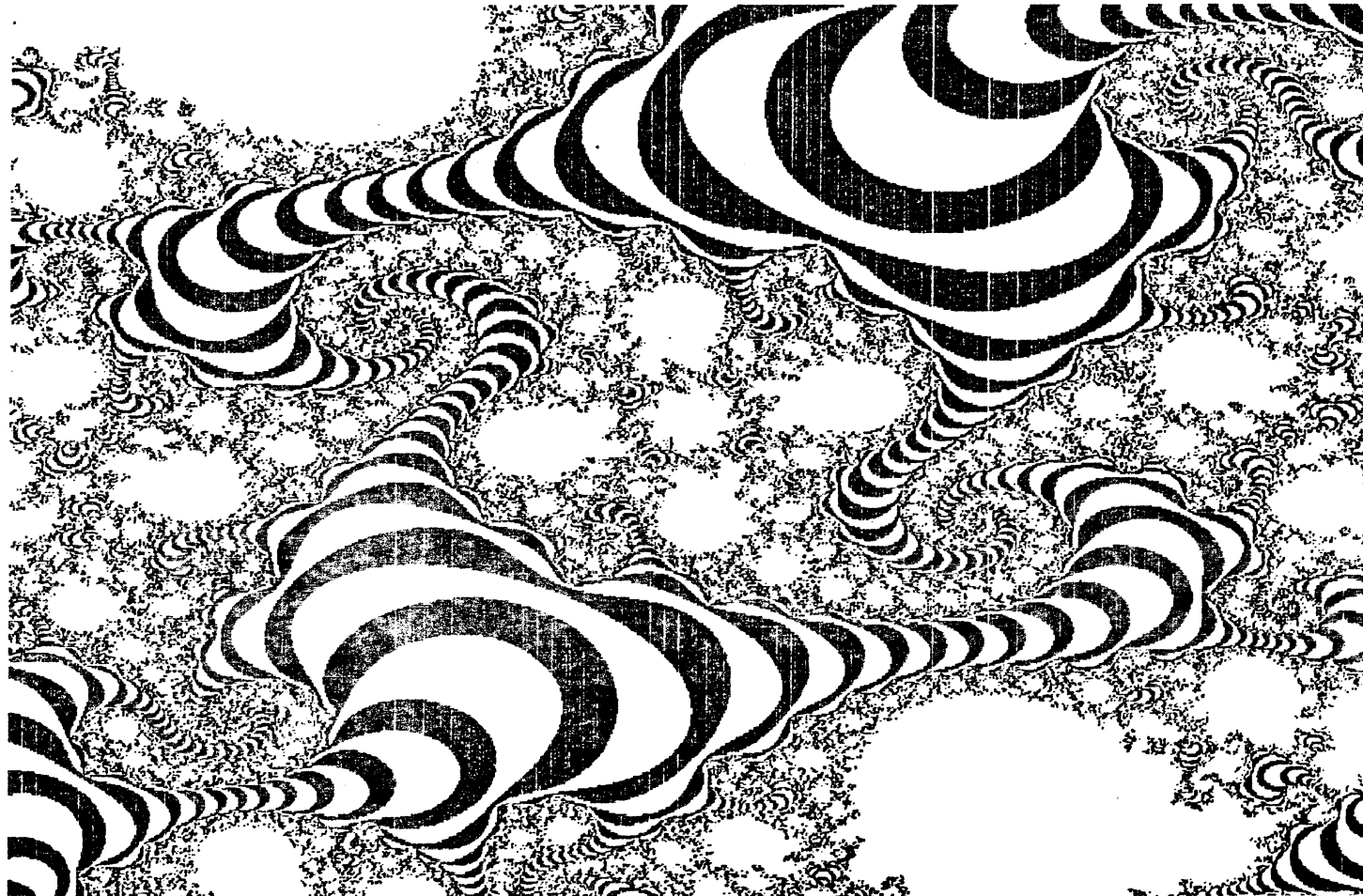
2. dBase II Anwendungen v. M. Apfelbeck
(IWT-Verlag Vaterstetten b. München)

Die Preise liegen bei ca. 58,- DM.

So das war es fürs erste. Also bis zum nächsten Info.

Harald

Harald



Apfelmännchen auf der GDP64

Helmut Bernhardt

Da der Computer für mich zu 80% ein Spielzeug ist, weil ich keine elektrische Eisenbahn habe, an der ich rumbasteln kann, kommt es recht selten vor, daß ich damit ernsthafte Arbeiten erledige. Eine dieser Spielereien war das Aufbauen und auf Auslesbarkeit des Grafikspeichers Erweitern der GDP64-VideoKarte des NDR-Klein-Computers. Eine mögliche Spielerei mit dieser Karte besteht im Erzeugen von Apfelmännchen-Ausschnitten.

Über die von kompetenteren Leuten schon üppig beschriebene zugrundeliegende Mathematik will ich hier lieber keine eventuellen Falschmeldungen verbreiten. Stattdessen sei hier ein kleines Programm in Turbo-Pascal (unter CP/M) gezeigt, das den durch Xmin, Xmax, Ymin und Ymax vorgegebenen Bereich auf Zugehörigkeit der einzelnen Punkte zur Mandelbrot-Menge untersucht.

Ein Punkt gehört dann zur Mandelbrot-Menge, wenn die Iteration in der REPEAT-UNTIL-Schleife des Hauptprogramms Rmax-mal durchlaufen wird, ohne daß die Abbruchbedingung $X^2 + Y^2 > Kmax$ erfüllt wurde. In diesem Fall wird der durch Xi und Yi vorgegebene Punkt nicht gesetzt.

Wenn diese Bedingung aber schon vorher erfüllt wird, soll ein Punkt nur dann gesetzt werden, wenn eine ungeradzahlige Anzahl Iterationen zur Erfüllung der Bedingung $X^2 + Y^2 > Kmax$ geführt hat.

Da solche Berechnungen enorme Zeiten beanspruchen und man gewährleisten haben will, daß ein fertiggestelltes Bild zum Schluß auch gerettet wird, bevor der Kühlschrank den Computer in den Wald schickt und alles umsonst war, wird das Bild automatisch ausgelesen und auf Diskette geschrieben (zweiter Teil des Hauptprogramms).

Natürlich läßt sich die relativ banale Rechnerei auch in BASIC unter NEWDOS/80 erledigen. Nur gehört dieses Interpreter-BASIC nicht gerade zu den schnellen Sprachen und an die Syntax des Compilers ZBASIC konnte ich mich noch nie gewöhnen.

Die für den Umgang mit Reals auf Z80-Rechnern wohl schnellste Sprache Turbo-Pascal benötigte mit einem bei 9,216 MHz laufenden HD64180 dann auch noch volle 4 Tage zur Erzeugung des größer wiedergegebenen Bildes, das mit einem entsprechend abgewandelten Programm auf der GRIP-5 mit 768*560 Bildpunkten errechnet wurde. Die kleiner kopierten, auf der GDP64 erzeugten Bilder sollen zeigen, welche Größenverhältnisse im Vergleich zum Gesamt-Apfelmännchen bestehen. Jedes Bild ist dabei ein verkleinerter Ausschnitt des vorherigen Bildes.

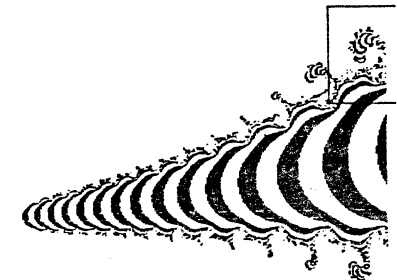
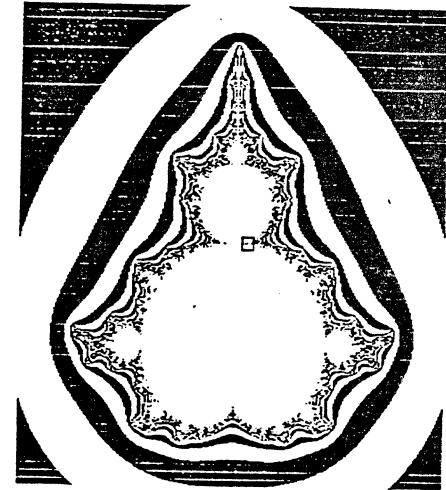
Die Parameter der einzelnen Bilder sind:

Xmin	Xmax	Ymin	Ymax	Kmax (Iterationen)
-2,0	0,5	-1,25	1,25	50
-0,8	-0,7	0,05	0,015	50
-0,795	-0,775	0,135	0,152	200
-0,795	-0,794	0,1434	0,1444	200
-0,79452	-0,79440	0,14432	0,14444	250
-0,79445	-0,79441	0,14434	0,14438	300
-0,79445	-0,79443	0,14435	0,14437	300

```

1: PROGRAM mandel;
2: CONST
3:   (* fuer die Apfelmaenner-Berechnung *)
4:   Xmin = -0.79445;
5:   Xmax = -0.79441;
6:   Ymin = 0.14434;
7:   Ymax = 0.14438;
8:   Kmax = 300;
9:   fname = 'aman18.gra';
10:
11:   ak = 2;
12:   Rmax = 50;
13:
14:   (* fuer die HRG GDP64 *)
15:   XhrG = 512;
16:   YhrG = 256;
17:   cmd = $70;
18:   ctrl1 = $71;
19:   xmsb = $78;
20:   xlsb = $79;
21:   ymsb = $7A;
22:   ylsb = $7B;
23:   hrgrd = $60;
24:
25:   speicher = 256;
26:
27: VAR
28:   puffer : ARRAY A1..speicher OF byte;
29:   ib, A, B, I, J : byte;
30:   f : FILE;
31:   Xn, Yn, Xr, K : integer;
32:   Xi, Yi, dX, dY, p, q, X, Xalt, Y : real;
33:
34: PROCEDURE ready;
35: VAR
36:   status : byte;
37: BEGIN
38:   REPEAT
39:     status := portAcmdÜ;
40:   UNTIL ( status AND 4) =4
41: END;
42:
43: PROCEDURE setx ( Xp : integer);
44: BEGIN
45:   ready;
46:   portAxmsbü := Xp DIV 256;
47:   portAxlsbü := Xp MOD 256;
48: END;
49:
50: PROCEDURE sety ( Yp : integer);
51: BEGIN
52:   ready;
53:   portAymsbü := Yp DIV 256;
54:   portAylsbü := Yp MOD 256;
55: END;
56:
57: PROCEDURE dot ( V : integer );
58: BEGIN
59:   ready;
60:   IF V = 0 THEN portAcmdÜ := 1 ELSE portAcmdÜ :=0;

```

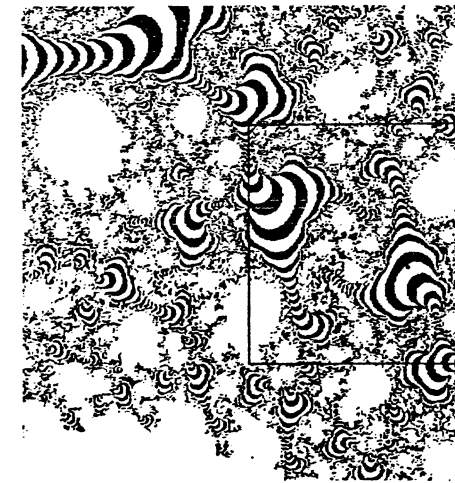
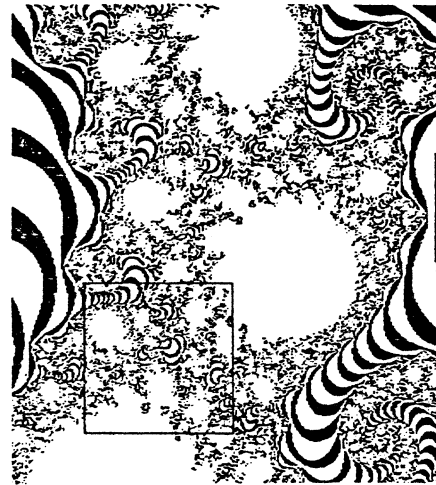
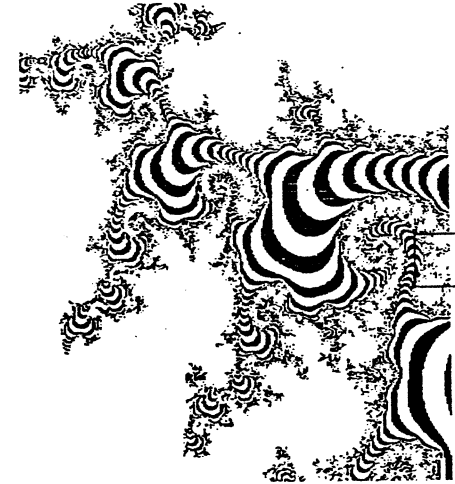
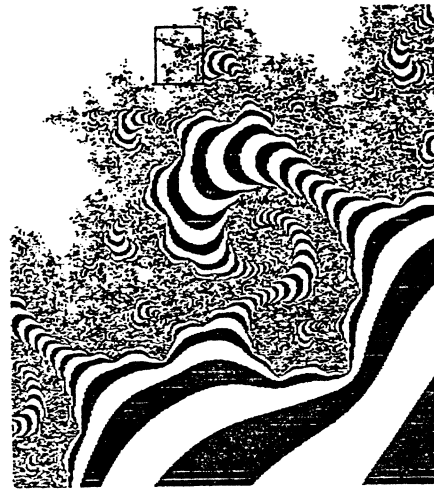


HEFT
21
September
1987

```

61:   ready;
62:   portÄcmdÜ := 128;
63: END;
64:
65: PROCEDURE c1h;
66: BEGIN
67:   ready;
68:   portÄcmdÜ := 7;
69:   ready;
70:   portÄctrl1Ü := 33;
71: END;
72:
73:
74: BEGIN (* Hauptprogramm *)
75:   c1h;
76:   dX := ( Xmax - Xmin ) / XhrG;
77:   dY := ( Ymax - Ymin ) / YhrG;
78:   FOR Xn := 0 TO XhrG - 1 DO
79:     BEGIN (* Apfelmaenner-Berechnung *)
80:       Xi := Xmin + dX * Xn;
81:       setx ( Xn );
82:       FOR Yn := 0 TO YhrG - 1 DO
83:         BEGIN
84:           Yi := Ymin + dY * Yn;
85:           K := 0;
86:           X := 0;
87:           Y := 0;
88:           REPEAT
89:             Xalt := X;
90:             X := X * X - Y * Y + Xi;
91:             Y := 2 * Xalt * Y + Yi;
92:             K := K + 1;
93:             UNTIL ( X * X + Y * Y > Rmax ) OR ( K = Kmax );
94:             IF K = Kmax THEN K := 0;
95:             sety ( Yn );
96:             dot ( K MOD ak );
97:           END;
98:         END;
99:         BEGIN (* Abspeichern der HRG auf Disk *)
100:            assign ( f , fname );
101:            rewrite ( f );
102:            FOR A := 0 TO 63 DO
103:              BEGIN
104:                Xr := A * B;
105:                setx ( Xr );
106:                FOR B := 0 TO 255 DO
107:                  BEGIN
108:                    sety ( B );
109:                    REPEAT
110:                      ready;
111:                      portÄcmdÜ := 15;
112:                      ready;
113:                      I := portÄhrgrdÜ;
114:                      ready;
115:                      portÄcmdÜ := 15;
116:                      J := portÄhrgrdÜ;
117:                      UNTIL I = J;
118:                      puffer Ä B+1 Ü := i;
119:                    END;
120:                    blockwrite ( f,puffer,2);

```



```

121:      END;
122:      close ( f );
123:      writeln ( 'Bild fertig und abgespeichert !' );
124:      END;
125: END.

```

```

;Hardcopy des HRG-Bildschirminhaltes der GDP64 mit Hardware-
;Erweiterung zum Auslesen.
;Ausdrucken über EPSON MX80 mit Typ 3 ROMs
;
CMD      EQU      70H          ;Ports der GDP64
CTRL1   EQU      CMD+1
CTRL2   EQU      CMD+2
CSIZE   EQU      CMD+3
DELX    EQU      CMD+5
DELY    EQU      CMD+7
XMSB    EQU      CMD+8
XLSB    EQU      CMD+9
YMSB    EQU      CMD+10
YLSB    EQU      CMD+11
MFREE   EQU      60H          ;Auslesen nur mit Hardware-Patch
;
;ESC, A, 0BH ( 8/72" Linespacing)
; Zeichen an Drucker
LD      A,1BH
CALL    PRINT
LD      A,41H
CALL    PRINT
LD      A,B
CALL    PRINT
XOR     A
OUT     (YMSB),A          ;YMSB immer 0
LD      B,1              ;Zähler für Bildschirmhälften
B35     LD      A,0FFH     ;zweite Hälfte fertig?
CP      B
JR      NZ,WEITER       ;nein: dann nächste B-Dot-Spalte
LD      A,0CH           ;sonst <CR> an Drucker
CALL    PRINT
JP      0000H           ;und
;WBOOT, wenn fertig
;
;1 oder 0 in B an XMSB
;für zweite oder erste Hälfte
;rechter Rand minus 8 Dots
;ESC, L, 00H, 03H an Drucker
; 768 Graphik-Bytes folgen
B40     LD      A,1BH
CALL    PRINT
LD      A,'L'
CALL    PRINT
XOR     A
CALL    PRINT
LD      A,3
CALL    PRINT
LD      A,C              ;X-Koordinate innerhalb einer
;Bildschirmhälfte
OUT     (XLSB),A        ;Bildschirmhälfte
LD      E,0FFH         ;Abwärtszähler für Y-Koordinate
B90     LD      A,E
OUT     (YLSB),A        ;nächstes Byte lesen
;Y-Wert vorgeben
TEST1   LD      A,0FH     ;Ausgabe anfordern
OUT     (CMD),A
CALL    WAIT

```

```

IN      A,(MFREE)       ;HRG-Byte
LD      D,A            ;zweimal lesen,
LD      A,0FH         ;um eventuelle
OUT     (CMD),A       ;Lesefehler
CALL    WAIT          ;zu erkennen
IN      A,(MFREE)
CP      D
JR      NZ,TEST1      ;wenn keine Übereinstimmung
CPL     ;Byte invertiert drucken
CALL    PRINT         ;dreimal drucken,
CALL    PRINT         ;um Bildschirmformat
CALL    PRINT         ;ungefähr zu treffen
DEC     E             ;nächster Y-Wert
LD      A,0FFH       ;waren das schon alle ?
CP      E
JR      NZ,B90        ;wenn nicht, nächstes Byte
LD      A,C           ;X-Koordinate
SUB     B             ;um ein Byte zurücksetzen
LD      C,A
LD      A,0DH         ;Wagenrücklauf
CALL    PRINT
LD      A,0AH         ;Zeilenvorschub
CALL    PRINT
LD      A,0FBH
CP      C
JR      NZ,B40        ;Bildschirmhälfte fertig ?
DEC     B             ;nein: nächste Zeile
JR      B35           ;sonst nächste Hälfte
;
PRINT   PUSH    DE
;
PUSH    HL
PUSH    BC
PUSH    AF
LD      E,A
LD      C,2
CALL    0005H
POP     AF
POP     BC
POP     HL
POP     DE
RET
;
WAIT    IN      A,(CMD)   ;GDP ready ?
AND     04H
JR      Z,WAIT
END

```

HEFT
21
September
1987

Unser Computer ist schon ein gutes Geraet. So mit NewDOS, G-DOS, H-DOS ect. laesst sich sehr viel machen. Vor allem, haben wir sehr gute Programme und Programmiersprachen.

ABER, die Betriebssysteme (Newdos, G-DOS, ect.) sind zu keinen anderen Systemen (Osborne, Prof80, NDR, usw.) kompatibel. Die Programme und Disketten koennen nur zwischen den Computern TRS 80, GENIE und KOMTEC getauscht werden. Und diese Computer sterben (leider) langsam aus.

Anders bei dem Betriebssystem CP/M. CP/M ist zwar auch schon auf dem absteigenden Ast, aber durch die grosse Zahl der CP/M User wird es noch eine ganze Weile unser System ueberleben. Bei den CP/M Computer ist nur ein kleiner Teil des Betriebssystems, das BIOS, an den Rechner angepasst, der Rest des Betriebssystems ist fuer alle Rechner gleich. Durch diese Software-Schnittstellen (BDOS, CCP) ist CP/M weltweit zu einem Standard-Betriebssystem geworden. CP/M ist bei weitem nicht so komfortabel als Newdos, aber durch diesen Standart sind alle kommerziellen Programme mit 8 Bit CPU nur unter CP/M lauffaehig.

Es gibt nun einige Schaltungen, mit denen man den GENIE oder den TRS 80 auf CP/M aufbohren kann. Viele von Euch haben das auch schon gemacht. Aber genau hier fangen die Probleme an. Nicht mit dem CP/M, das laeuft einwandfrei, sondern mit der Darstellung am Bildschirm. Bekanntlich hat unser Bildschirm 64 Zeichen und 16 Zeilen. Um vernuenftig mit CP/M arbeiten zu koennen, braucht man aber 80 Zeichen und 24 Zeilen. Alle CP/M Programme (zb. WordStar, Dbase, Multiplan, Turbo Pascal, Mbasic, ect.) sind auf dieses Format angewiesen.

Schmitke Elektronik hat zwar vor laengerer Zeit eine 80 Zeichen Karte herausgebracht, aber diese ist inzwischen nicht mehr lieferbar. Ausserdem ist die Karte nicht unbedingt das, was man sich unter einer guten Loesung fuer dieses Problem vorstellt.

Hier kommt nun GRIP ins Spiel. Zu einem Zeitpunkt, als ich CP/M eingebaut hatte, aber (Gott sei dank) noch keine Schmitke 80 Zeichenkarte vorhanden war, wurde GRIP (Grafik-Interface-Prozessor) in der CT beschrieben. Nach der anfaenglichen Begeisterung und den (unvermeidlichen) Schwierigkeiten (CP/M Anpassung Software maessig) ist die GRIP bei mir bis heute voll im Einsatz.

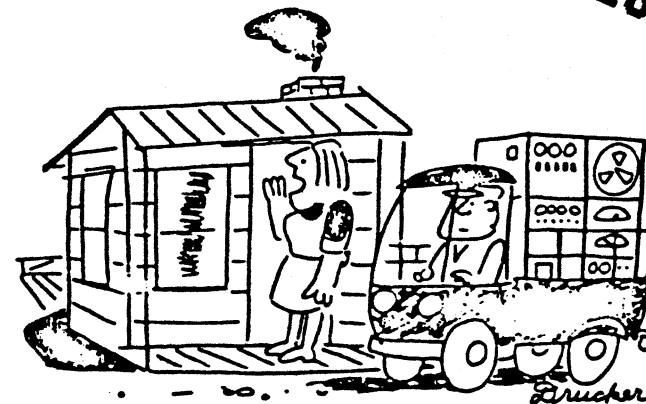
Nun zur GRIP selbst.

GRIP ist nicht nur eine 80 Zeichen Karte, sondern ein ganzes Computer System auf einer Europakarte. Sie laest sich wie ein Terminal ansteuern (TVI 950) und enthaelt ein Videointerface fuer Text und Hochaufloesende Grafik, mehrere serielle und parallele Schnittstellen zur Verbindung mit dem Hostrechner der und Peripherie sowie Anschluesse fuer einen Lichtgriffel und einen Lautsprecher.

Ein eigener Z80A-Prozessor uebernimmt den Datentransfer und die Grafik berechnungen. Die Karte laesst sich auf zwei verschiedene Arten von dem Host ansteuern: ueber eine V24 Schnittstelle oder ueber den ECB-Bus. Bei meinem System wird von letzterer Moeglichkeit Gebrauch gemacht.

Da GRIP 66kByte RAM enthaelt, koennen alle Schnittstellen gepuffert werden, d.h. die eingehenden Daten werden vor der Verarbeitung oder Ausgabe zwischengespeichert.

CLUB 80 HARD --- CLUB 80 HARD



„Charlie! Wo soll der neue Computer hin?“

Portbelegung

Um das Umschreiben von Programmen zu erleichtern habe ich hier von einigen Computerkonfigurationen aus unserem Club die Portadressen zusammengetragen. Jedoch haben sich nur wenige (4) daran beteiligt. So konnte ich nicht feststellen ob sich nicht manche Ports bei den selbstgebastelten Zusatzkarten überschneiden.

Alle Portadressen sind in sedezimaler Schreibweise dargestellt.

Gesteuert wird GRIP mit doppel Escape (1bh 1bh xx yy), Escape (1bh xx) und Controll Zeichen (00 bis 1fh).

Die Karte laesst sich wie ein normales Text-Terminal einsetzen. Sie enthaelt im Text-Modus 9 verschiedene Zeichensaetze, darunter alle griechischen und sonstigen europaeischen Sonderzeichen. Auch ein eigener Zeichensatz laesst sich laden, d.h. alle Zeichen koennen vom Benutzer selbst umdefiniert werden.

Zusaetzlich sind in der Software noch Sonderfunktionen vorgesehen:

- Zwei Statuszeilen wovon nur immer eine sichtbar ist
 - erste Zeile mit Logo, Status, Uhr
 - letzte Zeile kann selbst definiert werden
- eine abfragbare Echtzeituhr
- Monitortestbild
- einstellbares Textformat (max. 96 Zeichen x 30 Zeilen)
- einstellbares Video-Signal (Anpassung des Video-Signals an den Monitor)
- Tasten-Umcodetabelle (fuer jede Taste kann statt des Original-Tastencode ein beliebiges Zeichenstring zum Host gesendet werden.)
 - diese Moeglichkeit ist fuer uns leider unbrauchbar, da unsere Tastatur eine Matrix ist, und keinen direkten ASCII-Code abgibt.
- Tastenzwischenspeicher (max. 64 Zeichen)
- einen Melodiegenerator (Tastenklick)
- Abfrage des Lichtgriffels
- einschaltbaren Drucker Spooler
- und sonst noch einige Sonderfunktionen.

An Schnittstellen stehen zur Verfuegung:

- ECB-Bus
- V24/RS323
- 8 Bit Apsgang (Drucker)
- 8 Bit Eingang (Tastatur)
- BAS,Hsync,Vsync (mussen zusammengeschlossen werden bei normalen Monitoren)
- Sound

Die Kommunikation mit der GRIP erfolgt entweder ueber die V24 Schnittstelle oder ueber den ECB-Bus. Ueber den ECB-Bus werden zwei Portadressen (A0,A1 oder C0,C1 ueber Jumper auswaehlbar) benoetigt. Vom ECB-Bus sieht der Datentransfer so aus:

Auf der Adresse X1 werden die Daten mit I/O-Befehlen eingeschrieben oder ausgelesen; auf X0 der Status abgefragt. Ist Bit 7 des Statusbytes auf '1' gesetzt, so steht am Daten-Port ein Byte von GRIP zum lesen bereit. Statusbit 6 signalisiert mit '1', dass das naechste Datenbyte zum Daten Port geschrieben werden kann.

Die Reaktion auf ein Befehls- oder Datenbyte richtet sich nach dem Modus, in dem sich die Karte gerade befindet.

Im TVI-Modus verhaelt sich GRIP wie ein TVI950-Terminal. Es koennen zusaetzlich zu den Steuer codes (01h - 1fh) noch einige Sonderfunktionen gewaehlt werden:

- Cursorattribute (unsichtbar, blinkender/stehender Block, blinkende/stehende Linie)
- Cursorposition setzen/abfragen
- Zeichenattribute (Index tief/hoch, Kleinschrift, Breitschrift, Unterstreichen, Invertieren, Durchstreichen, unsichtbare Schrift)
- Invertieren ein/aus
- Bildschirm ein (hell)/dunkel
- Strichgrafik ein/aus (zb. fuer Multiplan)

TRS-80 M1 und Sonderausfuhrungen von Manfred Held (X).

Port	
0-5	HRG von RB
0	ausschalten der HRG
1	einschalten der HRG
2	untere Ansteueradresse (LOB)
3	obere Ansteueradresse (HOB)
4	einlesen eines Grafikpunktes
5	setzen eines Grafikpunktes
80-87	RS232 - Schnittstelle von RB
80	Datenport für Kanal A
81	Datenport für Kanal B
82	Controlport für Kanal A
83	Controlport für Kanal B
84-87	CTC - Timer auf RS232
84	Kanal 1 der DART
85	Kanal 2 der DART
86-87	zur freien Verfuegung
D0-D7	80 - Zeichenkarte von Schmidtke
D0	Adressregister CRTC
D1	Datenregister CRTC
D2	Steuerregister
D3-D7	doppelbelegt
9-F	* Prommer
B0-B2	* Palbrenner
A0-A2	* Grip 2
A2-A3	* Tastatur Port
A4-AF	* Uhr
E8	* Drucker
EC-EF	* Floppycontroller
40-7F	* HD 64180 Intern
E0-E3	* Floppyselect
C0	* CPM

HEFT
21
September
1987

Im Tektronix-Modus verhaelt sich GRIP wie ein Tektronix-Terminal mit Speicherbildroehre.

Im Tektronix-Modus gibt es wieder verschiedene Modi:

- Alphamodus: ist wie ein "abgemagertes" Text-Terminal
- Punktmodus: es wird ein Punkt an der angegebenen Koordinate gezeichnet
- Vektormodus: es wird eine Linie zwischen zwei Koordinaten gezeichnet
- Inkrementalmodus: es wird von der letzten Koordinate 1-Punkt-weise weitergezeichnet (wie ein Plotter)

Auch im Tektromix-Modus gibt es Zeichenattribute:

- wie im TVI-Modus
- Zeichne durchgehende Linie
- Zeichne gepunktete Linie
- Zeichne strichpunkt Linie
- Zeichne kurz gestrichelte Linie
- Zeichne lang gestrichelte Linie

Soweit zu GRIP 2.6. Es gibt aber auch noch GRIP 3.1. Die Unterschiede betreffen aber nur die hoehere Aufloesung (768 x 560 Punkte), die man nur mit einem Monitor mit hoher Nachleuchtdauer einsetzen kann.

Dafuer fallen aber der Spooler, die Statuszeile und die Tastenumdecodierung weg.

Um mit GRIP 3.1 zu arbeiten muss man nur den EProm und einen Jumper tauschen (Kleinigkeit, kann jeder).

GRIP 4.1 ist wie GRIP 2, Softumschaltbar auf GRIP 3, hat einen erweiterten Befehlssatz, einen grosseren RAM Speicher und ist schneller im Bildaufbau.

Die letzte Entwicklung von CONITEC ist GRIP 5.43. Sie ist wie GRIP 4, ist jedoch noch schneller, hat 256kByte RAM, mehr Befehle und die Uhr ist Batterie gepuffert.

Zu den Preisen:

Meine GRIP war noch sehr teuer. Die Platine mit EProm und sehr gut beschriebenen Handbuch kostete damals wie heute ca. 250 deutsche Maerker (vor 3 Jahren waren die IC's noch teurer).

Mittlerweile gibt es fertigaufgebaute GRIP's schon zu sehr guenstigen Preisen (z.B. Inserate von CT, MC, usw.)

Fuer die Platine mit EProm und Handbuch von GRIP 5.43 legt man ca. 350 DM's hin. (rentiert sich aber !)

Ausserdem gibt es noch GRIPS (z.Zeit V 1.15 und 2.10). GRIPS ist eine Erweiterung fuer GRIP 2.x. Durch diese Zusatzplatine und einen anderen EProm wird GRIP 2.x schneller und hat einen grosseren Befehlssatz.

Zusätzlich kann man an GRIPS 2.10 eine Maus anschliessen.

Soweit zu GRIP selbst. Nun zum Anschluss an unser System.

In meinem System wird GRIP nur fuer CP/M benutzt (eigentlich eine Verschwendung). Die Karte wird ueber den ECB-Bus angesprochen. Man muss nur aufpassen, wenn ein 96 poliger Bus verwendet, das die B Leiste zum restlichem System unterbrochen ist. Denn GRIP kann auch Farbe. JA, GRIP kann wirklich Farbe, und das sogar in HRG. Man braucht dazu allerdings eine Zusatzplatine (ca.200.- DM). Dann hat man 16 von 4096 Farben gleichzeitig am Bildschirm zur Verfuegung. Mit einer 2 Farbkarte hat man 128 von 4096 Farben zur Verfuegung.

Fuer die Farbkarte(n) braucht GRIP eben diese B Leiste vom ECB Bus.

TRS80 M4/4p

Port

F8-FB	Druckerport Datenausgang Druckerstatus Eingang
F4-F7	Ausgang FDC Configuration
F0-F3	Ausgang FDC Controlregister Eingang FDC Controlregister
E8-EB	RS232 Ausgänge
E8	UART Master Reset
E9	BAUD Rate Register
EA	UART Control Register
EB	UART Transmit Holding Register
E8-EB	RS232 Eingänge
E8	MODEM Status
E9	
EA	UART Status Register
EB	UART Receive Holding Register
FC-FF	Cassettenausgang Cassetteneingang
E4-E7	NMI Mask Reg. Ausgang NMI Mask Status Eingang
E0-E3	INT Mask Reg. Ausgang INT Mask Reg. Eingang
9C-9F	ein- und ausblenden des Boot-ROM
90-93	Tonausgang
84-87	Ausgang zur Umschalten des Video Speicher und Zeichendarstellung

Fuer den Monitoranschluss gibt es mehrere Moeglichkeiten:

1. zwei Monitore - aufwendigste Loesung - brutal method
2. zwei Adapterleitungen und jedesmal umstecken - Holzhacker method
3. einen Umschalter - hauRuck method
4. elektr. Umschalter - gute Loesung - kein Kontaktprellen / Verschleiss
5. elektronischen Umschalter ueber Software Umschaltbar - elegante method

Bei meinem System wird (selbstvertaendlich) Punkt 5 verwendet. Dabei liegt der Umschalter auf Portnr. fbh Bit 4. Der Sound-Anschluss von GRIP und vom Genie sind ueber einen Addierer und einen kleinen Verstaerker zusammen auf den Lautsprecher gelegt. Ansonsten wird nur eine Stromversorgung benoetigt. GRIP braucht ne menge saft (5V ca. 1A), und das schafft das Netzteil des Systems nicht mehr. Will man noch die V24 Schnittstelle benutzen, muessen auch noch +12V zur Verfuegung gestellt werden (ca. 50 mA).

Software seitig, habe ich in den Bootsector von CP/M eine kleine Routine zum umschalten auf GRIP eingepflanzt (ganze 4 Bytes). Im BIOS von CP/M ist die Displayausgabe auf Adresse EA92 (die ich nach langem suchen endlich gefunden hatte). Man suche sich im File SYS/SYS die Routine, und ersetze sie durch die im Handbuch abgedruckte, und beim naechsten Booten erscheint GRIP mit Logo.

Jetzt muessen nur noch die Programme (WordStar, dBase, Turbo, usw) an die GRIP angepasst werden. Das ist aber ein anderes Thema. Vielleicht schreibt irgend Jemand einmal einen Artikel darueber, wenn genuegend GRIP bei uns im Club eingesetzt sind.

Ich, von meiner Seite aus, kann GRIP nur weiterempfehlen. Zur Zeit duerfte es nichts besseres auf diesem Gebiet geben, das dieses Preis/Leistungsverhaeltniss hat. Falls es noch Fragen zu GRIP gibt koennt Ihr mich ja anrufen oder Euch gleich an CONITEC wenden.

Viel spass beim GRIPeln wuenscht, Manfred Held

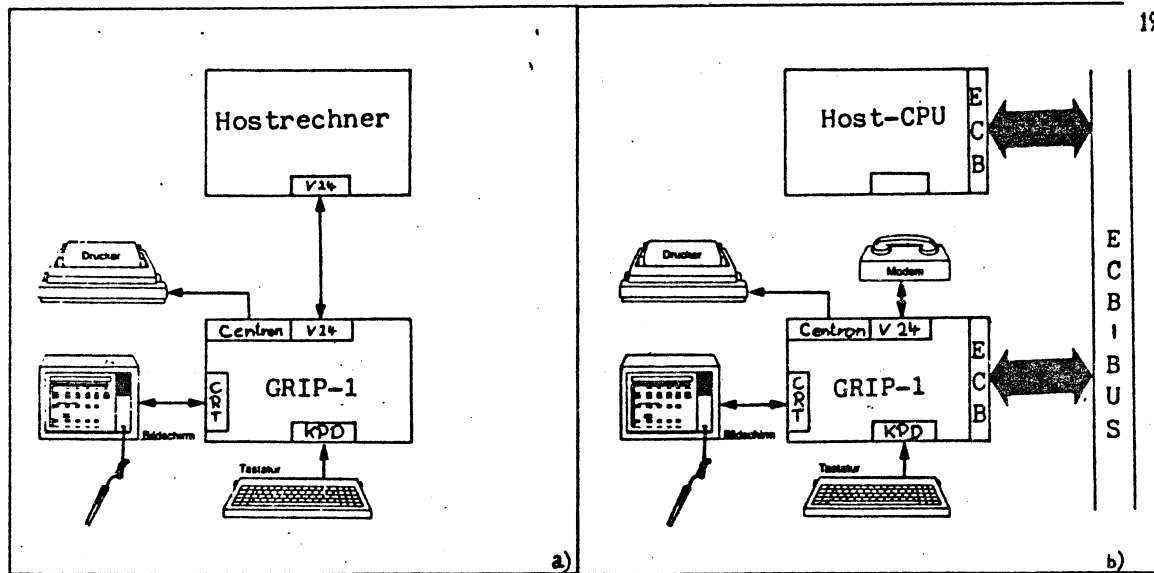
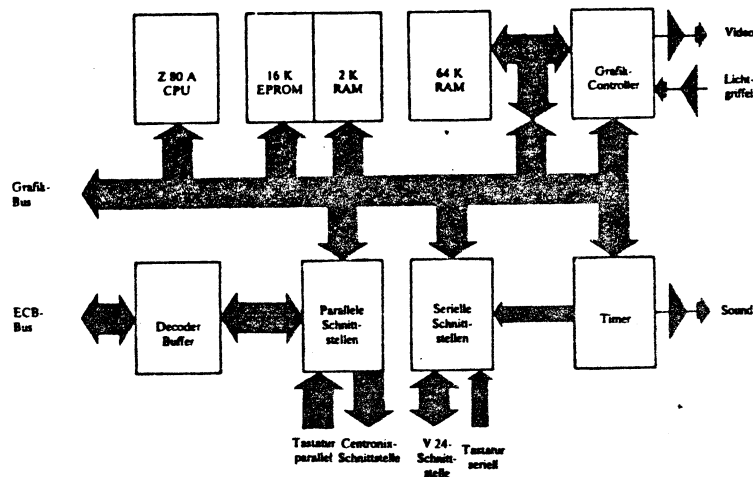
Genie IIs und Zusatzkarten von Gerald Schröder

Port

0-3F	HD 64188 Intern (Überschneidung mit HRG 1b und Host-Karte des IIs)
7E	Zusatz-RAM im IIs
7F	Zusatz-ROM im IIs
80-87	RB-SIO
C0	EG 64 (Erweiterun auf 64 bzw. 96 K RAM)
D0-D7	80-Zeichenkarte von Schmidtke
EC-ED	Banker von Helmut Bernhard
F9	Interrupt-Steuerung im IIs. Diverses im IIIs
FA	Systembyte im IIIs
FD	Druckerport bei alten Genies
FE	Systembyte im Genie IIs
FF	Kassetten E/A
(D0-D7, F1, F2, F9, :	SIO/PIO bei IIs (TCS-Zusatz-Karte)

Eckehard Juhn

HEFT
21
September
1987



BÖRSE -- BÖRSE -- BÖRSE

Suche Programm-Disk für VISICALC, TRS 80 Modell 3 Tastatur QUERTZU-
40 oder 80 Tracks DD/SS. Deutsches Begleitbuch vorhanden.
Angebote mit Preisangabe (Coby für die Schweiz kein Problem)
an W. Piller
Rohnenstr. 8
CH-8835 Feusisberg
Tel. 01/784 74 18

Ich habe auch noch folgendes Problem:
Für meinen TRS 80 Modell 3, habe ich noch verschiedene Handbücher mit
Programmketten die auf dem Computer mit TRS DOS laufen. Da es sich
jedoch um original amerikanische Programme handelt, sind sie auch für die
amerikanische Tastatur ausgelegt. Mein TRS 80 hat jedoch die QUERTZU
Tastatur, deshalb habe ich grosse Schwierigkeiten bei der Dateneingabe.
Und zwar nicht nur weil Z- und Y vertauscht sind, sondern auch die Zusatzzeichen
sind unterschiedlich angeordnet. (werden für "commands" gebraucht)
Wer weiss mir eine Lösung?
Gibt's event. ein Convert-Programm mit dem die Programmdisk umgeschrieben
werden kann?

Herzliche Grüsse

PS. Komme am 8. Nov. 87 zur Hobby Elektronik (Samstag nicht möglich)

Angebot für den Bastler:

2 Cassettendrive's von ADAM

1 Tastatur (ADAM)

sowie 3 Programm-Cassetten mit Programmen

für ADAM-Rechner zu verschenken. (gestiftet von Peter Spies)

Gesamt oder einzeln bei der Redaktion erhältlich.

Eigentlich sind das keine guten Nachrichten für unsere Clubmitglieder, was im letzten Info unter der Überschrift "Tandy in Deutschland tot" stand. Wenn Hartmut Obermann unsere Computer als Dinosaurier der Homecomputerszene bezeichnet, dann hat er in gewisser Weise recht, obwohl es schmerzlich ist, einen erst vor drei Jahren auf den deutschen Markt gebrachten 4p-Computer schon heute zum alten Eisen zu zählen.

Doch überlegen wir mal: Wer von uns fährt schon immer das neueste Automodell? Ist nicht der mehrere Jahre alte Wagen für die meisten von uns ein nützlicher und unentbehrlicher Helfer? Ähnlich beim Computer: Entscheidend ist, daß er seinen Zweck erfüllt, unabhängig vom Baujahr. Wer sich mit Datenbanken, integrierten Systemen oder komplizierten mathematischen Problemen beschäftigt, braucht schnelle Mikroprozessoren mit 16 oder gar 32 Bit Wortbreite und Speichern im Megabytebereich. Wer jedoch bescheidenere Aufgaben mit dem Computer lösen will, kommt mit den "Dinosauriern" noch ganz gut zurecht.

In der industriellen Steuerungstechnik liegen die 8 Bit-Mikroprozessoren nach wie vor an führender Stelle. Wo Platzbedarf, vielfältiges Angebot an Hard- und Software und geringer Leistungsverbrauch eine Rolle spielen, sind die 8 Bit-Typen immer noch die erste Wahl. Gerade das Herz unserer TRS 80-Computer, der Z80, ist - was die verkauften Stückzahlen betrifft - ein ganz großer Renner. Aus der Zeitschrift "Design & Elektronik" habe ich einen kleinen Abschnitt herauskopiert, der eine Bestätigung dafür ist, daß wir mit Zilogs Erstling nicht aufs falsche Pferd gesetzt haben.

Obwahl die CPU-Architektur des Z80 bereits vor mehr als 10 Jahren entstand, ist diese Einheit immer noch der am weitesten verbreitete 8-Bit-Mikroprozessor, von dem — bei steigender Tendenz — jährlich etwa 50 Millionen Stück verkauft werden. Die Langlebigkeit des Z80 auf dem Mikroprozessormarkt ist sicherlich auf die Leistungsfähigkeit und Flexibilität seiner Architektur zurückzuführen.

Was jährlich 50 Millionen mal verkauft wird, muß ein erstklassiges Produkt sein und wird in absehbarer Zeit nicht aussterben. Für Aufgaben im unteren oder mittleren Leistungsbereich ist das Leistungs/Preisverhältnis der neueren TRS-80 und des Z80 ganz hervorragend.

Heinrich Betz

Was habt Ihr gegen CP/M ?

Jedesmal, wenn sich einer der TRS80-/GENIE-Päpste zu CP/M äußert, sträuben sich bei mir die Nackenhaare. Da kommen dann Äußerungen zu Papier, die dem unvoreingenommenen Info-Leser gleich die Vorstellung aufzwingen, daß man von CP/M lieber die Finger läßt, wenn schon ein Herr H. aus E. oder ein Herr S. aus L. u.s.w. damit nicht arbeiten wollen oder vorgeben, damit Schwierigkeiten zu haben. Die gleichen Herren bedienen sich aber gerne der CP/M-Programme, die unter NEWDOS oder GDOS nicht verfügbar sind. Und gerade die Vielzahl guter Programme, die unter CP/M laufen, ist eines der Hauptargumente für dieses Betriebssystem.

Ein weiterer Pluspunkt für CP/M ist seine Hardware-Unabhängigkeit. Voraussetzung auf der Hardwareseite ist lediglich, daß als CPU ein 8080, 280, HD64180 oder Z280 vorliegt und daß ein Umschalten zwischen ROM und RAM (ab 0000H) möglich ist. Nach dem Abschalten des Boot-ROMs muß ein zusammenhängender RAM-Speicher von 0000H an vorliegen. Wie Peripherie angeschlossen ist, spielt keine Rolle. Diese Hardware-Komponenten werden entsprechend im BIOS berücksichtigt. Das BIOS ist der Teil des Betriebssystems, der elementare I/O-Operationen mit der Hardware erledigt (z.B. Zeichen von der Tastatur holen, Zeichen auf den Bildschirm ausgeben, Sektor von Diskette lesen oder auf Diskette schreiben u.s.w.). Da durch Anpassung im BIOS CP/M auf jeder Hardware-Kombination zum Laufen zu kriegen ist, ist es völlig egal, ob der Bildschirm memory mapped CPU-Adreßraum verschlingt oder als seriell angeschlossenes Terminal bereitsteht.

Diese Vielfalt an möglichen Computerarchitekturen ließ in der Blütezeit der 8-Bit-Rechner (und auch heute noch) hervorragende Maschinen entstehen, die sich in der Hardware zwar alle unterschieden, auf denen aber alle CP/M-Programme liefen.

Die neuere Entwicklung durch Aufstieg eines führenden Großrechner-Herstellers in den Microcomputer-Markt mit einem Mächtigeren-16Bit-Prozessor und einem dem CP/M ähnlichen Betriebssystem führte aber ungewollt in eine Sackgasse. Das DOS dieser Geräte sieht zwar auch die Abwicklung der I/O-Operationen über Betriebssystem-Aufrufe vor, was aber wegen der erbärmlichen Geschwindigkeit der Bildschirmausgabe nicht eingehalten worden ist. Die meisten Programme bedienen die Peripherie direkt (ähnlich wie wir das von allen besseren Programmen unter NEWDOS/GDOS kennen).

Dieses Verhalten der Software-Anbieter führte dazu, daß nur der Hardware-Hersteller sein Gerät verkaufen konnte, der es 100% kompatibel zum Vorbild baute. Wenn diese Voraussetzung nicht erfüllt war, liefen viele Programme nicht einwandfrei, derentwegen der Anwender sich das Gerät gekauft hatte. Mit solchen Geräten ist nicht viel zu verdienen. Der Begriff "Industrie-Standard" ist nur eine positive Bezeichnung dieser Misere. Dahinter verbirgt sich ein in der Hardware völlig festgefahrener System, das mit dem TRS80 in dieser Hinsicht durchaus vergleichbar ist.

Aber zurück zum TRS80 und seinen Nachfolgern. Die fest vorgegebene und recht verbaute Hardware und die direkt darauf zugreifenden Programme ließen wenig Spielraum für Verbesserungen, ohne daß das Gerät die TRS80-Kompatibilität einbüßt. Um CP/M fahren zu können, war es nötig, daß Floppy, Drucker, Tastatur und Bildwiederholpeicher nicht mitten im Adreßraum der CPU liegen. Für NEWDOS/80 und GDOS war das aber zwingend. Es wurden Banker entwickelt, die zwischen ROM und memory mapped I/O und RAM umschalten oder nur zwischen ROM und RAM umschalten und memory mapped I/O an das obere Speicherende verschieben. Mit teilweise riesigem Aufwand wurde die gleichzeitige TRS80-Kompatibilität und CP/M-Fähigkeit erreicht. In einem Testbericht der CHIP zum GIIIs liest sich das

so: "Mußte es denn ausgerechnet eine 8Bit-Maschine sein, in die soviel Arbeit und Ideenreichtum gesteckt wurde ?"

Dabei kann die Architektur eines CP/M-Rechners sehr einfach ausfallen, wenn man auf die TRS80-Kompatibilität verzichtet (siehe mc-CP/M-Computer). Es ist also nicht verwunderlich, daß bei dem dafür nötigen Aufwand und den damit verbundenen Einschränkungen heute niemand mehr TRS80-kompatible Computer anbietet.

Aber nicht nur die geringen Anforderungen an die Hardware und die damit verbundene Gerätevielfalt und das riesige Software-Angebot sprechen für CP/M. Auch das Programmieren in Maschinsprache ist unter CP/M sehr viel einfacher. Alle Betriebssystem-Aufrufe und Hardware-Zugriffe erfolgen über CALL 0005H, wobei im Register C immer die gewünschte Funktion ausgegeben wird und in anderen Registern je nach Funktion die Parameter übergeben / übernommen werden. Der Programmierer braucht sich über die Hardware-Architektur überhaupt keine Gedanken zu machen.

Daß nun ausgerechnet unsere Software-Päpste um CP/M einen großen Bogen machen, liegt wahrscheinlich daran, daß sie die enorme Arbeit, die investiert werden muß, um mit NEWDOS/GDOS umgehen zu können, nicht umsonst geleistet haben wollen und auch nicht bei CP/M noch einmal von vorne anfangen wollen. Es kostet tatsächlich einige Überwindung, anstelle von ZEUS nun ein Maschinenprogramm mit Wordstar zu schreiben und mit einem der zahlreichen Assembler zu übersetzen. Man muß gleich mehrere Handbücher wälzen.

Etwas fadenscheinig kommen mir die Argumente wie miese Leistungsfähigkeit und mangelnder Bedienungskomfort vor. Zugegeben, es ist schon etwas lästig, wenn man sich immer erst erinnern muß, in welchem Drive nun die Diskette steckt, auf der die gewünschte Datei vorliegt. NEWDOS sucht mir das selbst heraus. Andererseits ärgert mich bei NEWDOS das dort übliche DIR-Chaos. Ein neu auf die Diskette geschriebener File verschwindet irgendwo mitten zwischen den schon vorher auf der Scheibe vorhandenen Files. Dagegen wird von CP/M das Directory so ausgegeben, wie die Einträge nacheinander erfolgt sind. Und dann gibt es da noch diese herrlichen Wildcards, mit denen eine selektive Anzeige des Directory möglich ist.

Die vorgeworfene geringe Leistungsfähigkeit von CP/M ist eher als Vorteil zu sehen. Dadurch, daß CP/M nur auf reine Betriebssystem-Funktionen beschränkt ist, nimmt es wenig Speicherplatz weg und braucht auch nicht mit Overlay-Technik zu arbeiten. Alle über die Betriebssystem-Funktionen hinausgehenden Leistungen müssen entsprechende Programme liefern, und dafür gibt es sehr gute Utilities. Es ist mir völlig gleichgültig, ob ein Disketten-Kopieren mit einem Betriebssystem-Befehl oder mit einem Programm durchgeführt wird, ausschlaggebend ist der Komfort bei der ganzen Aktion.

Und schließlich habe ich als CP/M-User Anschluß an all die vielen Z80-Rechner, die in den letzten 10 Jahren auf den Markt gebracht wurden. Was bei mir läuft, ist auch dort verwendbar und umgekehrt.

Andererseits ist es bedauerlich, daß NEWDOS/80 nicht einfach auf andere Geräte angepaßt werden kann. Selbst wenn sich irgendjemand diese Mühe machen sollte, alle hardware-abhängigen Routinen herauszusuchen und umzuschreiben, wäre das von zweifelhaftem Wert, weil halt die besseren Programme unter NEWDOS/80 die Treiber-Routinen für Bildschirm und Tastatur umgehen und selbst versuchen, Tasteneingaben in den Speicherzellen ab 3800H zu erkennen.

Erfreulich ist im Zusammenhang mit CP/M, daß sich unser Vorstand über die Unkenrufe der Gurus hinweg für eine Öffnung in Richtung CP/M ausgesprochen hat. Mit zusätzlich CP/M sind wir nicht mehr ganz so weit vom heutigen Geschehen entfernt wie ausschließlich mit NEWDOS/80 und GDOS. Oder weiß sich jemand zu erinnern, in den letzten 2 Jahren in einer Fachzeitschrift etwas über TRS80, GENIE, NEWDOS/80 oder GDOS gelesen zu haben? Von CP/M liest man inmitten des ganzen IBM-, Amiga- und Atari-Gdröhns dann doch gelegentlich noch was.

Helmut Bernhardt

HEFT
21
September
1987

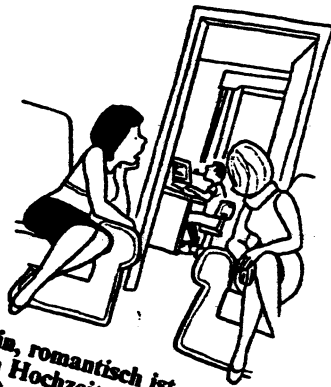
LS-DOS 6.3 - KOPIERSCHUTZ

=====
 Mit seinem neuen LS-DOS 6.3 folgt Logical Systems Inc. dem Beispiel anderer Softwarefirmen und verpasste dem DOS einen Kopierschutz. Allerdings auf eine besondere Art und Weise. Beim erstmaligen Booten einer neuen Systemdiskette werden vom DOS hardwarebezogene Daten ermittelt und innerhalb des Systems gespeichert. Wie dies vor sich geht und um welche Daten es sich handelt, ist bislang noch nicht bekannt. Auf jeden Fall werden diese Daten bei jedem Backup mitkopiert. Bei jedem Booten vergleicht nun LS-DOS 6.3 die innerhalb des DOS gespeicherten Daten mit denen aus dem momentan benutzten Computer. Wird ein Unterschied festgestellt, also ein anderer Computer verwendet, setzt das DOS ein Flag. Wann und wo dies geschieht, ist nicht bekannt. Auch wird dieser Test sehr schnell und nur bei einigen DOS-Befehlen durchgeführt. Ab und zu wird auch bei der Rückkehr von Anwenderprogrammen ins DOS nach diesem Flag geschaut. Wird nun tatsächlich ein anderes LS-DOS 6.3/Computer benutzt, wird man vom System aufgefordert, das richtige LS-DOS zu booten. Es ist nun durchaus möglich bzw. fast sicher, daß diese Routine innerhalb des Systems aufgespürt wird. Der ermittelte Code wird jedoch sicherlich an der einen und anderen Stelle des Systems verschlüsselt gespeichert sein. Es soll auch nicht möglich sein, diesen Code zu entschlüsseln. Teile davon dürften aber in der Seriennummer eines jeden LS-DOS zu finden sein. Versuche, die bislang unternommen wurden, den Schutzcode zu umgehen oder auszuschalten, führten in den meisten Fällen zum Absturz oder Zerstörung des Systems.

Das oben Erwähnte ist eine kurze Zusammenfassung aus einem Artikel der SOMicro. Über weitere Erfahrungen mit diesem Kopierschutz wurde bislang nicht berichtet.

Übrigens: Eine deutsche bzw. eingedeutschte Version des LS-DOS 6.3 soll ab Spätherbst 1987 auch bei uns erhältlich sein.

Klaus Hermann



„Nein, romantisch ist er nicht.
 Zum Hochzeitstag hat er nicht
 statt Rosen einen Gruß auf dem
 Heimcomputer hinterlassen.“

Personen der Handlung:

Alchemist Schelm
 Kopfjäger
 Hoher Offizier
 Stimme (Faust)

Orte der Handlung:

Hackerplätze in Lück, Wim und Miro

Zeit der Handlung:

1987 (das genaue Datum ist bei jedem Akt angeführt)

Irgendwelche Ähnlichkeiten mit lebenden Personen sind nicht zufällig, sondern voll beabsichtigt.

Anmerkung: Um die Druckkopfnadeln zu schonen, wurden die Namen der handelnden Personen durch leicht verständliche Kürzel ersetzt. (Die Überschriften der Aufzüge sind den "Betreffs" der entsprechenden Briefe entnommen.)

20.4. AS: "Beitrag zur Scheibothek"

Hallo, Genosse von der anderen Fakultät! Ich habe da in meiner Giftküche ein Ding für TSCRIPS (ab jetzt: "TS") gebraut, das dem Vielschreiber das Leben erleichtern soll. Es setzt automatisch das Datum und die Anschrift in Deinen Briefkopf ein. Glaubst Du, daß ich das dem Fußvolk anbieten kann, ohne daß ich anschließend mit Rückfragen totgetrampelt werde? Als ehemaliger Diskothekar kennst Du Dich doch mit den Qualitäten unserer Leser aus. Hier: Nimm und lies!

23.4. KJ: "Aus der Stille des Urlaubs"

Nun, was kümmert Dich die Menge?
 Wirf es mitten ins Gedränge,
 was Dein siebter Sinn ersann!
 Glücklich sollt sich jeder schätzen,
 wenn er auch nur einen Fetzen
 Deiner Gab' ergattern kann!
 Fürchte nicht zuviele Fragen!
 Alchemiste, laß Dir sagen:
 "Rückmeldungen" sind sehr knapp!
 Doch gesetzt den Fall, sie kämen,
 sollte Dich das nicht sehr grämen:
 Freude hängt vom Echo ab!
 Glücklich preise sich die Meute,
 daß sie dieses Mal und heute
 nicht "IIIs" von Dir bekommt,
 sondern (was wir selten haben)
 für den Alltag gute Gaben,
 wie es armen Leuten frommt.
 Ich erkenne Dich nicht wieder -
 "leise flehen Deine Lieder" -
 singst mir einen Grabgesang,
 Derweil Du mit TSCRIPS' kecken
 A- und D-Utily-Blöcken
 treibst den Trappschuh in die Ecken,
 weil ihm sowas nicht gelang!
 Die Programme sind 'ne Ware,
 die durch gute Kommentare
 erst den wahren Wert erhält.
 Folglich mußt Du es ertragen,
 falls man Dir mit hundert Fragen
 später auf den Wecker fällt.
 Hach, ich kann es selbst nicht lass'n-

Frage; kann auch "EDTASM"
Deine Source assembliern?
Dann will ich am grünen Holze
(und sogar mit eitlem Stolze)
es auch einmal selbst probiern.
Bin in TSCRIPS gut zu Hause
wie in meiner eignen Klause;
habe die Version "Fünf-Fünf".
Kann ich's hiermit auch probieren?
Oder braucht man die "Fünf-Vieren"?
(Hältst mich jetzt für unvernünftig-
ja, für uneinsichtig gar!?)
Das ist traurig - aber wahr...)

25.4. AS: "Vom Dichten und Trachten"

Habe nun, ach, die Source auch noch für EDTASM umgeschrieben.
Geb zu: Mein Anhängsel in TS einzubauen, ist etwas kompliziert.
Man kann davon leicht einen Tea-Schwips kriegen...

28.4. AS:

Aha, Du bist also ein 5.5er. Her damit! Mit "Sicherheit" lassen
sich die Patches auch dort einbauen (...!!!!...)

4.5. KJ:

Ich leide an Breitspurigkeit. Deshalb kann ich Deine schmalen
BOSpuren nicht lesen. Daß Du aber Dein Spurwerk nicht zu einer
sauberen Überspringprozeßion veranlassen kannst, ist ein Alp-
traum. Außerdem: Wieso schaffst Du noch mit "5.4"?

7.5. AS:

Ich bewältige die PD's jetzt programmatisch. Zur Hölle mit dem
PD-Befehl!

19.5. KJ: "Bohrt der schon wieder?..."

Da von WF (WeinFurt) noch immer nichts kommt, schicke mir doch
bitte, bitte --- mische einfach mal 80 und 40 durcheinander
(auf Vorder- und Rückseite). ZEUS vergällt's Dir!

21.5. AS: "TSCRIPS und überhaupt"

Muß zugeben, daß mein Unding noch Mängel hat, denn es hängt
sich ab + an = auf. Wenn der Fehler ausgemorzen ist, kriegst Du
die Platte. Vorausgesetzt, ich bewältige die verdammten PD's...

2.6. KJ: "TRIED TO LOAD ROM"

Genau das ist die Antwort meines TRS80 jedesmal, wenn ich meine
Version 5.5 mit Deiner Erweiterung (die ich inzwischen endlich
aus WF erhalten habe) boote. Paßt sie doch nur zu 5.4? Jetzt
kriegst Du doch zwei Disks von mir m.d.B., deren Rillen mal mit
feilen Files zu füllen. "Auf Los geht's los! 80+40!"

8.6. AS:

Hallo Alter! Es steht geschrieben (Anmerkung des Kulissen-
schiebers: Wo?), daß bei der Implementation die ORGs ein waches
Auge verdienen. (Aha!)
Du mußt mit ORG GANZOBEIN das Teil neu assemblieren und dann
anhängen. Nimm auf gut Glück HIMEM =FD00h. Mit ORG 4049h DEFW
GANZOBEIN-1 begrenze dann den Textspeicher.- Gute Nacht!

10.6. KJ: "Sehr geehrter Herr..."

Aus lauter Ehrerbietung vor so vielen ORGs in Ihren works müßte
ich Sie eigentlich siezen. Und vor sovielen Ratschlägen, die Du
mir erteilst und die ich nicht verstehe! Wohin denn nun mit dem
DEFW OFD00h? Und welche "Zeile 4049h" soll ich ändern? Die
gibt's ja gar nicht in Deinem opus! Und wieso strempelst Du von
einem ORG zum ändern? 6mal! Das ist ja ORGiastisch!

(Anm.d.Regie: Die weiteren tollen statements von KJ seien dem
Zuschauer hier vorenthalten.)

Kurz und ungut: Hier ist Dein Listing; jetzt male mir mal hin-
ein, wie Du Dir das denkst, bitte! Schreib doch mal ein rich-
tiges Ass-Buch, damit alle mal das Assemblieren begreifen!

12.6. AS: "Was T ich scrips, des Löt ich folb..."

Dummer! Ich taumle nicht "von einem ORG zum andern". Das sind
Patches in TSCRIPS. Das heißt: Ab der Adresse, auf die "ORG"
verweist, wird der Speicher mit den auf "ORG" folgenden Bytes
belegt. Und 4049h ist natürlich nicht in meinem Programm. ES
ist einfach der Speicherplatz, wo der Zahlenwert des HIMEM
steht. Willst Du HIMEM ändern, z.B. in FE00h, so mußt Du diese
neue untere (!) USER-Grenze dort wie folgt eintragen:

```
ORG 4049h DEFW OFE00h
```

Das kann an einer beliebigen Stelle im Programm stehen (natür-
lich nicht mitten in einem Programmablaufteil). 9100h ist
übrigens der Startpunkt von TS 5.4.

Lehrbuch Schreiben? Nee, ein dummer Leser (wie Du) genügt
mir.Tschüss!

13.6. KJ:

So, jetzt hab' ich Dich erwischt: HIMEM ist nicht die untere,
sondern die obere Grenze des ungeschützten User-Speichers!
(Auch der treueste Hund kann mal seinen Herrn beißen...)

17.6. AS: "TS-Erweiterungen"

Also, mein Lieber: In die Versionen 5.5 und 5.6 geht es nicht
rein, ohne gewisse verwanzte Adressen zu ändern.

(Anm.des Kulissenschiebers: Vgl. Aufzug vom 28.4.!)

Jetzt besorge Dir also endlich mal die Version 5.4, mit der
alle anderen anständigen Menschen arbeiten! Außerdem solltest
Du Dir endlich ein 80er-Laufwerk anschaffen.

Denn leicht ist's wirklich nicht, zu schreiben
mit 80er Draiw auf 40er Scheiben!

5.7. KJ:

Gesagt, getan. Version 5.4 ist jetzt da. Sie tut alles, was ich
will (was ich nicht von jeder sagen kann...):

Sie lädt, schreibt, speichert, schweigt und spricht -
nur drucken tut sie leider nicht!

Dauernd behauptet sie: "LINE PRINTER NOT READY!"

Das ist natürlich gelogen. Denn mein Liniendrucker ist mehr als
READY. Er ist ganz scharf wie ein RADI aufs Drucken! Ich, nicht
faul, disassembliere das Teil. Que lac je? Lauter E's und
83H's! Seitenweise! Na sowas! Wer multipliziert da wen? Wieder
mal bin ich vom Stamme Kannitverstan.

7.7. AS: "TS und kein Ende"

TS besteht zur Hälfte aus Bildschirm. Das hättest Du am Auf-
tauchen der Adresse 3C00 erkennen müssen. Der eigentliche Ein-
sprung ins Textsystem ist bei 9100h. Der Code endet deshalb mit
02-02-00-91. Schau mal nach. - Mein nächster Rat (welche Nummer
hat dieser jetzt eigentlich?) lautet: Forche mal nach D3-FD
und DE-FD <= OUT(FD) bzw. IN(FD)>. Da stehen 3 Bytes, aus denen
Du 3A-E8-37 (Druckerstatus abfragen) bzw. 38-E8-37 (Zeichen
ausgeben) machen kannst.

(Der Souffleur: "32-E8-37, nicht 38-...!")

Die Regie zum Souffleur: "Lauauauter!"

10.7. KJ: "Die (doch) endliche TS-Geschichte"

Guten Morgen, lieber AichensSt!
Habe gemacht. Wie Du sagtest. Oje, muß ich dumm sein. Immer
noch sehe/lese ich: LINE PRINTER NOT READY! Sollte ich die 3
Bytes vielleicht doch woanders hinschreiben?

Nein, ich geb's auf...
"..."

17.7. AS: "DEM WAHREN, GUTEN, SCHÖNEN"

Die 3 Bytes 3A-E8-37 bzw. 32-E8-37 (Anm. des Souffleurs: "Merkt ihr was? Am 7.7. hieß es 38-...! - O diese Alchimisten, geben Plumbum für Aurum aus! Da muß ich doch mal Rum-Bum machen!!) muß Du anstelle von D3-FD-00 usw. usw. Der Nuller ist ein NOP (No Operation). Prüfe mit dem Debugger, was an der Stelle 37E8 steht, wenn der Drucker Ready ist. Oder ersetze den ganzen Quark durch ein CALL 05B4. Suche in Deiner Version nach der Abfrage. Checke, ob gleich darauf die Ausgabe folgt..... (hier hat der Darsteller sein Rollenbuch verlegt - o nein, hier hat der Dichter dieses Dramas den Rosa Faden verloren. Die Geister, die er rief, versteht er selbst nicht mehr.)
Notfalls schicke mir einen Kontoauszug des Sektors, wo DB-FD-00 steht. Hier tut sich offenbar eine Unvereinbarkeit zwischen TRS80 und GENIE auf, die infowürdig ist.
(Der Produktionsleiter: In der Tat. Über diese Unvereinbarkeit wurde im INFO schon berichtet, und zwar von einem Hohen Offizier. An den sollte sich der Kopfgänger mal halten.)

19.7. HD: "Hallo..."

Nimm und lies INFO 14, S.43! Die Version, die ich Dir für Deinen TRS80 schickte, 'war für's GENIE (!!) und kann bei Dir nicht laufen (!). Beim GENIE gehen Druckerein- und -ausgabe über den Druckerport, während sie beim TRS80 "memory-mapped" sind. Du mußt also alle (alle!) Port- in Memory-Schreib/Lese-Operationen umsetzen. - Das wär's.

22.7. KJ

Jetzt druckt's. Seit ich mit der Außenwelt nicht mehr per Schiff über den Hafen (port) verkehre, sondern meine Gedanken direkt aus meinem Gedächtnis (memory) durch die Luft zu ihr fliegen lasse - es druckt zwar, aber nur normale Texte; nicht Deinen Adressenautomaten. Hier weigert sich meine TreueRosinante80mal und wiehert: "Formfehler beim Laden!"

23.7. AS: "Von den letzten Dingen"

Das Problem geht seiner Klärung entgegen. (!)
Wo OUT(FD),A steht, kannst Du D3-FD-00 gegen CD-B4-05 eintauschen. Ersetze mal den ganzen Schmus in FRS17 von RB B3 bis B7 durch CD-D1-05-00-00 - das geht sicher! Mehr fällt mir jetzt nicht ein - -
(An dieser Stelle fordert der Theaterdirektor zu einem unangesagten Posaunenstoß auf, da der größere Teil des Publikums soeben eingeschlafen ist!)

24.7. KJ: "Ach du heiliger A..."

Fortsetzung vom 22.7.: "Formfehler beim Laden!" Setzte ich zur Abwechslung mal ORG=OFD00h und ORG 4049h auf OFCFFh (denn es muß doch um 1 niedriger sein!?), so erfuhr ich:

"LADEVERSUCH AUF ROM-SPEICHERPLATZ"

Ich habe Deine komplette Anleitung nochmal Schritt für Schritt durchexerziert.
(Hier folgt die Beschreibung der Schritte. Auch sie sei dem ungeduldigen Zuschauer erspart.)

27.7. AS: "Das Übliche"

Die ORG's innerhalb TS müssen selbverständlich unverändert bleiben. Stelle sicher, daß alle CALL-Adressen, die innerhalb von TS liegen, ein Label haben, bloß keine Zahl. Du kannst auch 02-02-00-91 in 05-xx-egal-egal verwandeln. 05 heißt Kommentar. xx ist die Anzahl der jetzt noch freien Bytes bis rechts unten (im Sektor). Wenn Du dann meine Erweiterung mit APPEND an-

hängst, müßte auf Anhieb alles das richtige Format haben.
Übrigens: Wenn es Dir doch noch gelingen sollte, das Ding auf Deiner Rosinante zum Laufen zu bringen, so wisse, daß unser Hoher Offizier auch daran interessiert ist. Einer von Euch wird's schon schaffen.

28.7. KJ: "Ente jut - (fast) alles jut"

Danke für die Hinweise! Jetzt klappen: Die automatische Datumeingabe, der Klammeraffe und die durchgehende Unterstreichung - bloß die automatische Adresseneingabe funkt nicht. Was soll das heißen in Deinem Erweiterungscode: "Kann bis vor das Label 'address' gelöscht werden? Ein Label 'address' gibt es in Deinem Listing überhaupt nicht! Ich habe jetzt mal TS disassembliert. Du lieber Gott! Bei Blatt 111 schaltete ich den Drucker ab (das sind 33,5m Papier)! Und in meinen Speicher ging das alles schon gar nicht hinein. So konnte ich auch nicht nach sämtlichen CALL-Labels suchen.

31.7. AS: "Von kalten Spalten und kessen Adressen"

Da-Da-Da-Do- - d.h.: Da das Datum doch korrekt bearbeitet wird, kann ich mir
De-Da-Da-Do- - d.h.: denken, daß das DOS Unterschiede aufweist. Sieh mal im GROSSER nach und vergleiche mittels Debug.

5.8. KJ: "Die Große Aufgabe"

Ich geb's auf - - -
Neuerdings druckt mein Drucker mit Deinem TS nur noch Chaos. Obwohl auf dem Bildschirm Sinnvolles steht...

7.8. AS: "-nverstanden!"

Chaos ist die Quelle der Schöpfung. Du mußt halt lernen, Hieroglyphen zu lesen. Dechiffriere sie auf Millimeterpapier. Bei Deiner Patcherei scheint einiges in die Grütze gegangen zu sein. Oder die ORGs in unseren TS's stimmen nicht überein. Ich habe meins ja vielfach vergewöhntätigt! Schick mir doch mal 'nen Hex-Ausdruck von so einem mißratenen Brief.

10.8. KJ: "Schreiben Ohne PINKELPause..."

Hier ist der HEX-Ausdruck des Chaos! Schaut wirklich verHEXT aus.

12.8. AS: "Von den vorletzten Dingen"

Dieser HEX-Ausdruck ist die Grafik für Deinen Briefkopf. Nur gibt Dein Drucker sie nicht als solche aus, weil das Programm die Anzahl Grafikbytes gleich Null setzt. Also ist beim Patchen was in die Grütze gegangen. Wahrscheinlich sind die Anzapf-adressen bei Deiner und meiner Version nicht dieselben. Du mußt am besten jemand mit möglichst ähnlicher Hardware finden und darauf ansetzen.

13.8. KJ: "Nun sind die Ferien bald zu Ende, da muß ich nochmal ganz behende..."

("Halt!" ruft die Regie, "Wir machen hier keine Dichterlesung! Da dieser Akt wieder voll gereimt ist, geht jetzt der Vorhang nieder!")

15.8. AS: "Von den hinterletzten Dingen..."

Mit Sicherheit ist Dein TS nicht OK. TS bedient den Drucker nicht über den DCB (CHROUT), sondern gibt selber den Output auf 37E8 bzw. Port FD aus. An der Stelle setzt der Check auf das Byte 7F (Randausgleich) an.
Letzter Versuch: Ersetze den CALL 05B4h durch LD(37E8),A.
Übrigens: Die Zeichensätze von TS erzeugt der EPSON mithilfe seiner Drucker-HRS. Jeder Deiner Briefe enthält daher eine Menge Grafik. Durch Deine Zapperei ist aber die Angabe der An-

16.B. KJ: "Er- und andere -weiterungen"

Es sei Dir gewährt Deine Bitte: Verbannt sei aus unserer Mitte mit unbarmherzigem Tritte Herr TSCRIPS, weil ohne Fort-Schritte!

21.B. KJ: "Haha-hihi..."

Röckraths DOS-Buch sowie WAGNERS "Machine language Disk I/O & other mysteries" verhalfen dazu, den Adressenautomaten nun endlich dazu zu bringen, das Gewünschte auszuspucken! Wie das? Ich habe mich mal im Debuggen geübt und mit der Einzelschrittausführung ('I') die Veränderungen in den Registern und im Stack beobachtet. Dabei stellte ich fest, daß der Filename "ADRESSEN/LIB" nicht mit dem Schlußbyte 03 oder 0D in den FCB übernommen wurde. Flugs setzte ich dieses dort ein- und plötzlich ging's: Auf das Kommando 'G' des DEBUG wurde Deine Anschrift eingegeben! Du hast in der Source für EDTASM die Zeile DEFM 'ADRESSEN/LIB', ODh geschrieben, was bei EDTASM nicht erlaubt ist. Ich habe daraufhin das ODh gelöscht und als Extrazeile DB 'ODh' eingefügt. Mußte dann aber immer noch bei jedem Brief den Debugger bemühen...

24.B. AS: "Uff!"

Wir kommen der Sache näher. Du mußt nun nur noch das Zählbyte für diesen Record erhöhen!

29.B. KJ: "Fin de Partie!"

Erdal, der Frosch! HEUREKA! Ich hab's!

Wir haben beide am gleichen Fehlerstrang gezogen. Ich hatte Deine fehlerhafte Zeile nur halb richtig korrigiert, indem ich das ODh in Apostrophchen kleidete! Es muß natürlich DEF8 ODh heißen! Hätte ich beim Assemblieren die Option /WE (Wait upon Error) gewählt, wäre ich früher gewarnt worden.

Nun aber kommt das Tollste: Die Entdeckung des Fehlers in Deinem Source-Code für EDTASM weckte mein Mißtrauen gegen diesen und motivierte mich, mich nun endlich einmal näher mit ZEUS zu befassen! So fing ich von vorne an und assemblierte Deine ZEUS-Source mit ZEUS -: Und nun klappt's wirklich!

31.B. AS: "Betr. iff"

Alternativ zum Druckbefehl "BREAK F..." kann man meine Automatik auch mit @K abrufen. Dazu mußt Du nur das zum "Umleitung-bei-P"-ORG gehörige 01-Byte in 05 (=Kommentar) umzapfen, dann wird dieser Teil nicht gelesen.....

* * * * *

DEUS ex Machina (mit FAUSTens Stimme):

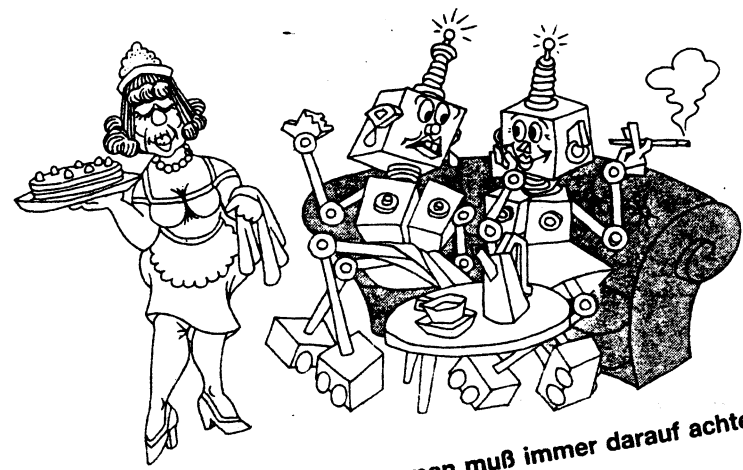
O glücklich, wer noch hoffen kann,
aus diesem Meer des Irrtums aufzutauchen!
Was man nicht weiß, das eben brauchte man,
und was man weiß, kann man nicht (immer) brauchen.
(Der Tragedie erster Teil)

Die letzten Zeilen sind (ihr wißt's) von Wolfgang Goethe. Der ganze Rest: er stammt von Arnulf & KaJoete. Ward diese Niederkunft der hundertdreißig Tage oft den Akteuren und dem Publikum zur Plage und tat so mancher sich bereits beim Lunch entfernen - so konnten, die da blieben, doch so manches lernen!

* * * * *
V O R H A N G
* * * * *



Sie war ja teuer – aber keine Steuern, keine Sozialleistungen und wenn es sein muß, arbeitet sie 24 Stunden.



Sie kostet nicht viel – aber man muß immer darauf achten, daß sie auch etwas tut.

- Wie Hartmut im letzten Info bereits angekündigt hat, besitzt der Club nun ein Reinigungsset für Diskettenlaufwerke. Wenn Ihr also damit Probleme haben sollte, so polliert doch mal Eu're heißgelaufenen Schreib- Leseköpfe und laßt Euch Reinigungsdiskette und Reinigungsflüssigkeit zuschicken. Ihr müßt mir jedoch mitteilen, für welchen Laufwerkstyp (Single Side oder Double Side) es benötigt wird (da gibt's nämlich einen Unterschied).

- In den letzten Wochen sind bei mir die nachstehenden Programme eingegangen.

Nr.	Name	Autor	Kurzbeschreibung
367	DCSDR/CMD	K.J. Mühlenbein	Direkte Ausgabe von Texten auf den Drucker (siehe Info 20, Seite 35/36)
368	TRIGONOM/BAS	K.J. Mühlenbein	Berechnung der trigonometrischen Grundfunktionen (siehe Info 20, Seite 17/18)
369	WURZEL/BAS	K.J. Mühlenbein	Wurzelberechnung (siehe Info 20, Seite 19)
370	CP/M Packet	Gerald Schröder	Alle Source-Files für CP/M-BIOS (nur Genie II) einschließlich Begleittext als ASCII und Textfile (Die Programme wurden von Gerald Schröder selbst programmiert, sind also nicht mit einem Copyright vorbelastet)
371	GDOS-Hilfpgm	Gerald Schröder	format: Programm zum Formatieren einer CP/M Diskette putsys, bootsec: Programm zum Generieren einer CP/M Systemdiskette shift: Programm zum Übertragen von GDOS/CMD-files nach CP/M
372	CP/M-Hilfpgm	Gerald Schröder	format: Formatieren von CP/M Disketten sysgen: Erzeugen einer Systemdiskette pdrive: Einstellung der Laufwerkparameter
359	TRSRAM	Gerald Schröder	neueste Fassung des Programms RANDISK
372	TRSRAMLM	Gerald Schröder	ähnlich TRSRAM, man kann jedoch wählen, auf welchem Laufwerk die RANDISK installiert werden soll
357	TRSSYS	Gerald Schröder	neueste Fassung von TRSSYS Läd alle Systemfiles in zwei Banks. Man kann somit die Systemdiskette aus Laufwerk 0 entfernen.
373	HEFAMD	Gerald Schröder	wie TRSRAM, jedoch für den Prozessor HD64180
374	HEFES	Gerald Schröder	wie TRSSYS, jedoch für den Prozessor HD64180
375	HEFAC56	Gerald Schröder	Utility-Program für HD64180; liefert TRAP-Abfang für Genie IIs
376	ZBOTRE	Gerald Schröder	80-Zeichen-Treiber für Schmidtke-Karte in Genie IIs
377	SYSS	Gerald Schröder	neues SYSS (Debug); nun auch unter 80-Zeichen benutzbar (nur mit ZBOTRE)
378	TREIBER	Gerald Schröder	Grafik-Treiber für Genie IIs mit Graphic 1.5
379	HPETRE	Gerald Schröder	Grafik-Treiber für HRG 1b
380	GRAFTEST1	Gerald Schröder	Testprogramme für Grafik-Treiber
381	GRAFSTB	Gerald Schröder	Assembler-Grafikroutinen-Bibliothek + Demo

Andreas Rychlik
Königsberger Allee 120
4100 Duisburg 1

Duisburg, den 2.8.1987

Hallo liebe Clubkameraden !

Wir haben jetzt neu auch eine CP/M 2.2-Software Bibliothek in unserem Club 80.

Ich habe mich bereit erklärt, diese umfangreiche CP/M 2.2-Software Bibliothek zu verwalten und bin auch nur für diese zuständig. Die NEW/TRSDOS-Software Bibliothek ist natürlich die Sache vom Werner Förster.

Wir haben:

ca. 82 Public-Domain Disketten von der CP/M-UserGroup
ca. 32 Public-Domain Disketten von der SIG/M-UserGoup
und noch einige eigene CP/M 2.2-Programme (z.B. für die HRG 1B)

Die Softwareliste gibt es nur auf Diskette, weil die Liste so lang ist, und nicht im Info abgedruckt werden kann !!!
Als Beilage zu der Katalog-Diskette gibt es eine Aufstellung der Diskettennummern, die wir alle haben.

Trotz der Katalog-Diskette werde ich in jedem Info ein Programm herausstellen, um die CP/M-Software schmackhaft zu machen.

Bezüglich der Diskettenformate ist folgendes zu erwähnen:

Da das Anpassen der CP/M-Diskettenformate meist nur mit viel Problemen vonstatten geht, habe ich mir kein einheitliches Format ausgesucht.

Ich werde die Software in einem gewünschtem Format kopieren !
Dieses Format muß natürlich bei der Bestellung angegeben werden.

Mögliche Formate sind:

Montezuma Sys. 80 DS/DD
Schmidtke S80 DS/DD
Schmidtke S40 SS/DD
Genie III 80/DS/DD
Epson QX-10 40/SS/DD
Osborne 40/SS/SD

Montezuma Sys. 40 SS/DD
Schmidtke S40 DS/DD
Genie III s 80/DS/DD
P 2000 80/DS/DD
IBM CP/M 86 40/SS/DD
Rainbow 80/SS/DD

Ich glaube, daß jeder mit einem dieser Formate zurechtkommt.
Falls nicht, bitte nach meiner PDRIVE-Liste nachfragen! Dort sind ca. 100 Formate angegeben.

382 SCRIPT1
BLOCK1 Gerald Schröder zwei Schriftsätze für GRABIB

363 VCMOD K.J. Mühlenbein hiermit können Zeilen oder Spalten eines in DIF-Format gespeicherten VISICALC-Files alphabetisch oder numerisch sortiert werden (siehe Info 19, Seite 29-31)

384 ZAFTS Arnulf Sopp Adressenmanager und andere neue Features für TSCRIPS 5.4

Das war's von der Club-Bibliothek,

Werner Förster

Kopieraufträge werden nur schriftlich angenommen und man sollte die Katalog-Nummer und die Software-Group angeben.

Am Besten wäre es natürlich, wenn alle Disketten in dem gewünschtem Format vorformatiert eingeschickt werden.

Auch muß der Einsender die vollen Porto- und Verpackungskosten in Briefmarken einsenden! (Ab sofort trägt der CLUB 80 nicht mehr die Portokosten.)

-

Wer seine Programme einschicken will, sollte das Meisterwerk nur mit einer Dokumentation in ASCII als File abschicken. (* /DOC)
Ferner muß kein Copyright vorhanden sein. Urheberrechtlich geschützte Programme werden nicht angenommen.

Natürlich können auch ganze PD-Disketten eingeschickt werden. Bestimmt ist einigen nicht bekannt, daß sie die Disketten, selbst wenn sie bei irgend jemand gekauft haben, jederzeit weitergeben dürfen.

Viel Spass mit der CP/M 2.2 Software !!!

Andreas Kypke

CLUB 80 - Bücherei

Langsam kommt wieder etwas Leben in die beinahe schon tot geglaubte Clubbücherei, wenn auch das Leben mehr aus Zugängen als aus Ausleihungen besteht!

So kann ich erfreulicherweise berichten, daß sich eines der vermissten Bücher wieder eingefunden hat. Die Nr. 7 in unserer Liste, das Buch "Introduction to TRS 80 Graphics" hat sich wieder in die Reihe der auf die Ausleihung wartenden Publikationen eingeordnet!

Zusätzlich hat die Bücherei Zuwachs von einem TRS 80-Aussteiger, nämlich von Josef Konrad bekommen. Josef hat uns neben einer ganzen Sammlung von 80 US-Zeitschriften drei Bücher vermacht (siehe die folgende Liste). Für diese freundliche Spende möchte ich mich an dieser Stelle recht herzlich bedanken!

Noch mehr Bewegung wird es in der Clubbücherei geben, wenn die bei der Vorstandssitzung beschlossenen Neuankäufe von wirklich aktuellen Büchern getätigt werden. Ich hoffe und glaube, daß dann auch die Anzahl der Ausleihungen wieder rapide steigt!

Karlheut Obermann

Nr. 0007: Introduction to TRS-80 Graphics

Don Inman --- dilithium Press (Portland-USA)

In diesem Buch wird gezeigt, was man mit der TRS-80 Graphik machen kann und vor allem wie. Beispiele und Aufgaben veranlassen zum experiment

Nr. 46 : FACAL

David L. Heiserman --- TAB Books Inc.

Einführung in die Programmierung von PASCAL, Tiny PASCAL und Supersoft Tiny PASCAL

Nr. 47 : TRS 80 Assembly Language Programming

William Barden --- Radio Shack

Eine Einführung in die Programmierung des Z80-CPU des TRS 80.

Nr. 48 : 80 US (Zeitschrift)

Amerikanische TRS 80-Zeitschrift Ausgaben Feb. 81 bis Juli 83 (teilweise fehlen Ausgaben)

HEFT

21

September

1987

67 Ich werde versuchen, Euch dieses Thema anhand von Fragen und kurzen Antworten nahe zu bringen. Wie Ihr feststellen werdet, hängen alle Fragen eng zusammen. Ob die Antworten jeweils stimmen, dürft Ihr gestrost anzweifeln. Mir paßten sie jedenfalls gut in den Kram.

1. Frage: Warum benutzen so wenige Leute im Club CP/M?

Antwort: Unser Rechner war nicht für CP/M gedacht, also hat jeder mit Newdos angefangen und kann/will jetzt nicht umsteigen, vor allem, weil es nicht genug Informationen darüber gibt.

2. Frage: Warum programmiert Ihr alle in Basic, wo Pascal doch viel besser ist?

Antwort: Ihr wißt gar nicht, daß Pascal besser ist. Außerdem gibt es nur unter CP/M ordentliche Pascals (Turbo-Pascal) und das müßtet Ihr auch noch neu lernen. Wieder fehlen Informationen.

3. Frage: Warum leiht sich niemand etwas aus der Clubbibliothek aus?

Antwort: Die Bücher sind für Euch nicht interessant oder Ihr wißt nicht, was alles Interessantes darin steht.

4. Frage: Welche interessanten Bücher schaffen wir neu an?

Antwort: Das hängt ganz von Euch ab, aber vorerst hat der Vorstand zwei Themenkomplexe vorgesehen: Einführungen in CP/M und Pascal. Weitere Vorschläge (Themen oder konkrete Bücher) sind von Euch herzlich willkommen.

5. Frage: Wer sucht die Bücher aus?

Antwort: Klein-Gerald war dumm genug, sich die Vorarbeiten dazu aufbürden zu lassen. Im Endeffekt sollt Ihr uns sagen, welche Bücher euch gefälligst kaufen sollten. Also sagt mal was! (Auch negative Kritik wie z.B. "Bloß nicht den Zaks; der schreibt total unverständlich!" können wir gut gebrauchen.)

6. Frage: Wie geben wir den Leuten Tips, die sich unbedingt selbst ein Buch kaufen wollen?

Antwort: Wir machen einige Bücher schlecht und loben andere in den Himmel. Dann könnt Ihr Euch die schlechten kaufen und selbst darüber ärgern, wie mies sie sind. Wiederum seid Ihr gefragt: Jeder, der lesen gelernt und mehr als den Titel eines Buches entziffert hat, schreibt sofort, was er über das Buch denkt. Dann können andere sich dann nach dem Buch die Hacken abwetzen oder mit einem abfälligen Blick bedenken, wenn sie es im Buchladen sehen.

7. Frage: Wie geben wir Euch die Tips konkret weiter?

Antwort: Seht Ihr vor Euch (das Info). Kurzkritiken (von mir aus auch lange) werden hier veröffentlicht und die wichtigsten Daten in eine Liste übernommen, die ich führen werden. In der Liste stehen (nach Sachgebieten geordnet) die Bücher und die Nummer des Infos, in dem die entsprechende Kurzkritik (bzw. auch mehrere Nummern, wenn mehr als einer das Buch gelesen haben sollte) abgedruckt ist. Wenn Ihr Euch dann dazu durchgerungen habt, mal ein Buch zu kaufen, seht Ihr in die Liste (gibt's bei mir oder von Zeit zu Zeit im Info) und sucht Euch nach den Kritiken das billigste und hoffentlich beste Werk aus. Da so ein Buch meist weit über 50 DM kostet, dürften sich diese kurzen Nachforschungen lohnen. Natürlich könnt Ihr auch erst 10 Bücher kaufen, bis Ihr das richtige findet.

8. Frage: Kritiken über welche Bücher sind interessant?

Antwort: Einfach alle. Übrigens reicht es auch, wenn Ihr in einer Buchhandlung herumgestöbert und dabei ein Buch gefunden habt, auf das Ihr andere hinweisen möchtet oder das Euch überhaupt nicht gefiel, auch wenn Ihr es nicht gekauft habt! Ein Profi wie Arnulf braucht sich ein DOS-Buch nur anzusehen und weiß schon, ob es was taugt. Dann kann er das auch ruhig sagen, damit andere, die nicht soviel darüber wissen, nicht einen Fehler begehen (indem sie kaufen oder auch nicht).

9. Frage: Wer macht den Anfang?

Antwort: Ist schon passiert. Auch wenn Ihr es vielleicht nicht bemerkt haben solltet, hat Hartmut schon einiges zum Thema "Bücher über CP/M" gesagt. In diesem Info werdet Ihr außerdem einige Kurzkritiken von meinereiner finden, wobei ich zugeben muß, daß ich nur eins der Bücher wirklich gelesen habe und mich bei den anderen auf den ersten Eindruck nach dem Durchblättern verließ. Ich halte das für legitim. Dazu aber an anderer Stelle mehr.

10. Frage: Wer darf Kritiken schreiben und wie sollen die aussehen?

Antwort: Jeder darf das und was Ihr schreibt, ist vollkommen egal. Wir sind froh, wenn überhaupt jemand sich bequemt, ein paar Zeilen aufs Papier zu werfen. Auch Bücher, die schon besprochen wurden oder die schon in der Clubbibliothek vorhanden sind, sollten von Euch ruhig nochmal kritisch beschrieben werden (wobei kritisch auch heißen kann, daß Ihr total begeistert von einem Buch seit und ohne es nicht mehr leben könnt).

11. Frage: An wen sollen die Kritiken geschickt werden?

Antwort: An Jens, Hartmut oder mich. Denn Jens veröffentlicht die Sachen, Hartmut ist für die Bibliothek zuständig und meine Wenigkeit wird die Listen mit den besprochenen Büchern führen.

12. Frage: Woher könnt Ihr die Listen bekommen?

Antwort: Die jeweils neueste Liste gibt's bei mir (man bemühe die Bundespost).

13. Frage: Wann macht der endlich Schluß mit seinen blöden Fragen?

Antwort: Jetzt.

Gerald Schröder

Bücherliste

besprechende Bücher

(keine Garantie für Preis und ISBN)

Pascal

Verlag	Autor, Titel	Preis	ISBN	Info
<u>Franzis</u>	Plate/Wittstock: Pascal - Einführung - Programmentwurf - Strukturen	58,-	3-7723-6903-0	21
<u>Sybex</u>	Tiberghien: Das Pascal-Handbuch	64,-	3-88745-614-9	21
<u>Vieweg</u>	Krekel/Trier: Die Programmiersprache Pascal	68,-	3-528-03337-1	21
	Kaier/Rudolfs: T.-P. Wegweiser für Microcomputer: Aufbaukurs	58,-	3-528-04443-8	21
	Hering/Scheurer: Fortgeschrittene Programmieretechniken in Turbo-P.	52,-	3-528-04467-5	21
	Fedtke: Pascal	??	3-528-04488-8	21
<u>Heise</u>	Wostrack: Turbo-Tools u. -Utilities unter CP/M	49,80	0123-5	21
	Rauch: Modell der Wirklichkeit - Simulation dynamischer Systeme mit dem Mikrocomputer	29,80	0524-3	21
<u>Markt & Technik</u>	Lüke: Turbo-Pascal	49,-	3-89090-150-6	21
<u>Teubner</u>	Wirth: Systematisches Programmieren	25,80	3-519-42327-8	21
	Wirth: Algorithmen und Datenstrukturen	38,-	3-519-02330-X	21
	" in Modula-2	38,-	3-519-02250-8	21
	Erbs/Stolz: Einführung in die Programmierung mit Pascal	25,80	3-519-12506-4	21
	Dal Cin/Lutz/Risse: Programmierung in Modula-2	19,80	3-519-00100-4	21
	Ottmann/Widayer: Programmierung mit Pascal	18,80	3-519-10084-3	21
	Mittelbach/Weremuth: Turbo-Pascal in der Praxis	24,80	3-519-02544-2	21
<u>Springer</u>	Wirth: Programmieren in Modula-2	42,-	3-540-13301-1	21

Verlag	Autor, Titel	Preis	ISBN	Info
<u>Springer</u>	Marthy: Methodik der Programmierung in Pascal	42,-	3-540-16611-4	21
	Däßler/Sommer: Pascal	??	3-540-12835-2	21
<u>Springer</u>	Remmele: Pascal - systematisch - strukturiert	??	3-540-12250-8	21
	Brosell/Zhou: Modula-2	69,-	3-540-15159-1	21
<u>Addison-Wesley</u>	Anstel: Programmieren: Die Entwicklung von Algorithmen in Pascal	48,-	3-925118-19-5	21
<u>Hanser</u>	Wilson/Addyman: Pascal	28,-	3-446-13953-2	21
	Pomberger: Softwaretechnik in Modula-2	??	3-446-14122-7	21
<u>Vogel</u>	Renner: Turbo-Pascal	43,-	3-8023-0758-5	21
<u>Oldenbourg</u>	Herschel/Pieper: Pascal und Pascal-Systeme	34,80	3-486-23685-7	21
	Herschel: Turbo-Pascal	29,80	3-486-29064-9	21
<u>Akademische Verlagsgesellschaft</u>	Doberkat/Rath/Rupietta: Programmieren in Pascal	??	3-400-00474-X	21
<u>IDEA</u>	Price: Strukturiertes Programmieren	??	3-88793-055-X	21

CP/M

Verlag	Autor, Titel	Preis	ISBN	Info
<u>Franzis</u>	Plate: Betriebssystem CP/M	58,-	3-7723-7522-7	14
	RPB1: CP/M kompakt	9,80	3-7723-7522-7	14
<u>Sybex</u>	Zaks: CP/M- Handbuch	48,-	3-88745-053-1	14
<u>Hüthig</u>	Weiß: CP/M - Ein Betriebssystem stellt sich vor	39,-	3-7785-1266-8	21
<u>Markt & Technik</u>	Dennon: CP/M - Anatomie eines Betriebssystems	68,-		14
	Digital Research: CP/M Operating System Manual	38,-	3-89090-369-X	21
<u>IWI</u>	Pol: Vom Umgang mit CP/M	43,-	3-88322-004-3	21

HEFT
21
September
1987

Die ersten Kurzkritiken

In diesem Info findet Ihr irgendwo einen Artikel über die Bücherliste. Hier möchte ich nun einen Teil davon vorstellen, und zwar die ersten Kurzkritiken, wobei ich hoffe, daß diese Rubrik mit Eurer Hilfe (sprich: schreibt Kurz- oder Langkritiken!) zur regelmäßigen Institution werden kann.

Über welche Themen sollen die Bücher sein, die Ihr besprechen könnt? Das ist vollkommen egal. Sicherlich ist die neueste Ausgabe von Müller's Kochbuch weniger interessant für die meisten Mitglieder (alles Männer + Computerfreaks => jemand anders steht in der Küche). Allerdings wird das Buch schon interessant, wenn zum Kochen ein LötKolben und zum Umrühren ein Schraubenzieher benutzt werden oder die Rezepte in einer dbase-Datei erfaßt sind. Also schreibt einfach über alles.

Gleichmaßen ist die Form schnurzpiepe. Ob ein Text, Stichpunkte, et-was Subjektives ("Zu heiß!") oder Objektives ("Bei 100 Grad heißem Wasser zeigte meine Hand Rötungen."), kurz, mittel oder lang, in hochdeutsch, plattdeutsch oder bayrisch: die Leser sollen froh sein, daß sie überhaupt was zu lesen bekommen!

Um gleich ein negatives Beispiel zu geben, habe ich ein paar Kurzkritiken verfaßt. Dabei war mir der Themenkomplex "alles" zu groß und ich habe mich auf die gezielte Suche nach Büchern gemacht, die sich mit Pascal bzw. mit CP/M beschäftigen. Selbstverständlich sollt Ihr nicht auch verzweifelt nach Büchern suchen, über die Ihr Kritiken schreiben könnt, aber ich hatte hochhoffiziellen Auftrag von unserem Oberguru, äh, Pardon: Obermann, Bücher aus diesen Bereichen anzuschauen.

Meine Methode war zugegeben dilettantisch: ich habe mir einfach überlegt, welche Anforderungen ich an die Bücher stellen würde und versucht, diese zu überprüfen. Meine Kriterien waren bei Pascal darauf abgestimmt, für Anfänger geeignete Bücher zu finden. Wie gesagt, sind die Kriterien stark subjektiv. Ein Buch sollte also (nach meiner unmaßgeblichen Meinung) leicht verständlich formuliert sein, damit auch Hobby-Programmierer es verstehen können. Wichtig ist, daß das Buch gut gegliedert ist, zu jedem Kapitel Übungsaufgaben zur Überprüfung des Gelernten bietet und im Anhang über eine vollständige Liste der Pascal-Syntax in Form von Syntaxdiagrammen verfügt. Außerdem sollte die Aufmachung einigermaßen professionell sein (keine Fotokopien oder schlecht lesbaren Listings). Letztes Kriterium war der besprochene Pascal-Dialekt: vorzuziehen wäre eine allgemeine Besprechung (also über Standard-Pascal) mit mindestens einem Kapitel über die speziellen Eigenschaften von Turbo-Pascal, weil dies der für unseren Rechner wohl die am ehesten verfügbare (vollständige) Pascal-Version ist. Eine Einführung in Turbo-Editor und -Compiler wäre auch nicht schlecht. Ganz raus fallen auf jeden Fall Bücher, die sich mit UCSD-Pascal beschäftigen.

Wie habe ich diese Kriterien angewandt? Da ich kaum alle Bücher durchlesen konnte, habe ich mir jedes nur kurz angeschaut und die folgenden Punkte überprüft:

- Inhaltsverzeichnis übersichtlich?
- Einleitung gut lesbar und verständlich?
- Übungsaufgaben vorhanden?
- Syntaxdiagramme vorhanden?
- Turbo-Pascal erwähnt?
- Druck in Ordnung?
- wie sehen die Programm-Beispiele (Listings) im Buch aus?

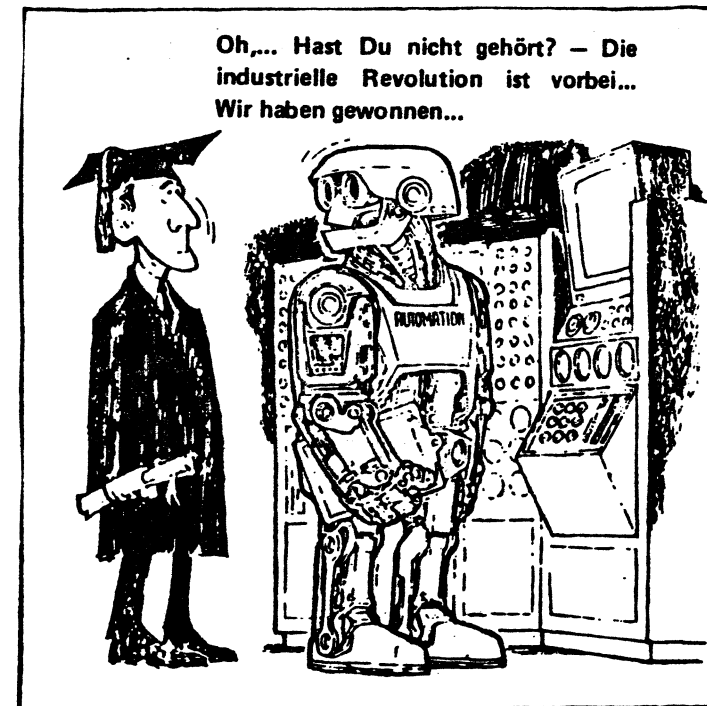
Vielleicht sind für Euch andere Dinge wichtiger, aber der absolute Anfänger bekommt mit dieser Liste vielleicht eine Vorstellung, wie er sein erstes Pascal-Buch auswählen könnte. Genug Auswahl gibt es jedenfalls. Selbst ohne UCSD-Pascal-Bücher habe ich noch ca. 50 Exemplare allein aus Katalogen herausgefischt, wobei man nach den Titeln meist schlecht sagen kann, welche Bücher für Anfänger geeignet sind. Aber vielleicht finden auch fortgeschrittene Programmierer bei den besprochenen Büchern welche, die sie interessieren.

Bei CP/M war es mir dagegen unmöglich, solche Kriterien aufzustellen. Vielleicht ist es wichtig zu wissen, wie die mitgelieferten Programme von Digital Research (wie ED) funktionieren. Aber wer wird sie schon benutzen, wenn er sofort bessere bekommen kann? Die Befehle der DOS-Oberfläche sind natürlich wichtig, aber die sind schnell erklärt. Anders sieht das schon mit der Programmierung aus, aber gehört das wiederum schon in ein Anfänger-Buch?

Hier überlasse ich gern kompetenteren Leuten das Feld (hoffentlich finden sich welche). Mit der Literatur zu CP/M sieht es eher spärlich aus. Meine Kritiken zu CP/M-Büchern sind auf jeden Fall mit noch mehr Vorbehalt als die zu den Pascal-Büchern zu lesen.

Im Anschluß an die Kritiken möchte ich Euch noch eine Liste mit Büchern vorlegen, die ich zwar in einem Katalog gefunden habe, mir aber weder in der Bibliothek noch in einer Buchhandlung anschauen konnte (kaufen wollte ich sie verständlicherweise nicht). Es wäre toll, wenn jemand sich auf das ein oder andere Buch stürzen würde. Vielleicht habt Ihr es ja gelesen oder Euch intensiv irgendwo angeschaut. Schreibt dann doch mal was darüber!

Gerald Schröder



Kurzkritiken

Kaier/Rudolfs: Turbo-Pascal-Wegweiser für Mikrocomputer
(Vieweg, DM 58,-)

Dieses Buch ist auf Turbo-Pascal zugeschnitten; allerdings konnte ich mir den 1. Band (Grundkurs) nicht anschauen. Es ist sehr ausführlich und scheint mir gut verständlich. Der Druck der mir vorliegenden Ausgabe ließ allerdings zu wünschen übrig. Zum Inhalt: Gezeigt wird alles erstes die Bedienung des Turbo-Pascal-Editors, was sehr wichtig ist. Behandelt werden dann (fortgeschrittene) Datenstrukturen (Sets, Records, Files, Pointer) und verschiedene Algorithmen (Rekursion, Sortieren, Baum, Datei-Behandlung). Ohne den ersten oder einen anderen einführenden Band ist dieses Buch nicht zu empfehlen.

Wirth: Systematisches Programmieren
(Teubner, DM 25,80)

Wie alle Werke von Wirth, dem Pascal-"Erfinder", ist dies ein Standardwerk, zumindest für jeden Informatik-Studenten. Die Sprache ist relativ trocken (eben wissenschaftlich), aber Wirth liefert hier eine gute, grundlegende Einführung, die für jeden verständlich ist. Er arbeitet viel mit Flußdiagrammen, was seine Programmbeispiele sehr schön veranschaulicht. Dieses Buch gehört zu den ältesten (vielleicht ist es das älteste) Pascal-Büchern. Deshalb finden sich auch keinerlei Dialekt-Einflüsse. Es handelt sich um "reines" Standard-Pascal.

Wirth: Algorithmen und Datenstrukturen
(Teubner, DM 38,-)

Dieses ist das zweite Standardwerk von Wirth. Er geht hier anhand von Pascal auf verschiedene Programm-Teile und Daten-Arten ein, die oft im normalen Programmierbetrieb benötigt werden und in anderen Sprachen oft schlecht zu realisieren sind. Die großen Themenkomplexe lassen sich unter "Bäume", "Sortieren" und "Arrays" zusammenfassen. Das Buch richtet sich an Leute, die schon mit Pascal programmiert haben. Leider wird im Anhang die Pascal-Syntax (welche Befehle sehen wie aus?) nur in der Backus-Naur-Form angegeben, die ich für schlecht lesbar halte. Positiv aber ist zu vermerken, daß es Übungsaufgaben in dem Buch gibt.

Wirth: Algorithmen und Datenstrukturen in Modula-2
(Teubner, DM 38,-)

Dieses Buch habe ich aufgenommen, weil es die Zukunft bedeutet. Wirth hat nach Pascal eine Pascal-ähnliche Programmiersprache definiert, die besonders den modularen Aufbau eines Programms betont, eben Modula-2 (die "2" steht für zweite Version). Die Themen entsprechen dem oben besprochenen Pascal-Band, die Programme sind fast identisch, nur eben in Modula formuliert, was sie in meinen Augen eleganter erscheinen läßt. Leider wird auch die Syntax von Modula nur in BNF angegeben.

Erbs/Stolz: Einführung in die Programmierung mit Pascal
(Teubner, DM 25,80)

Dieses Buch ist eine echte Einführung, die sich auch an den Computer-Neuling richtet. Es ist locker formuliert und mit vielen Zeichnungen verziert. Leider entstand es auch noch vor der Turbo-Zeit, aber dafür lehrt es eben das "reine" Pascal. Ich würde das Buch für den echten Anfänger sehr empfehlen, da es auch Übungsaufgaben zur eigenen Überprüfung bietet.

Kurzkritik

Als ich unserem Freund Gerald Schröder während des Vorstandstreffens die Arbeit auf's Auge drückte, eine Liste der Bücher zu erstellen, die der CLUB 80 für die Clubbibliothek anschaffen sollte, habe ich nicht damit gerechnet, daß er so gründlich vorgehen würde. Ich möchte mich an dieser Stelle für sein großes Engagement dadurch herzlich bedanken, daß ich eine Kurzkritik über eines der Bücher verfasste, derer er nicht habhaft werden konnte!

Kaier/Rudolfs: Turbo-PASCAL Wegweiser für Mikrocomputer
Grundkurs (Vieweg, 48,-DM)

Das Buch ist für den geschrieben, der sich ohne große fremde Hilfe in die Sprache Turbo PASCAL einarbeiten will. Daher ist auch das erste Kapitel den allgemeinen Grundlagen der Datenverarbeitung und Programmentwicklung vorbehalten. Auf den folgenden 15 Seiten wird dann die Bedienung des Turbo PASCAL-Compilers recht anschaulich beschrieben, wobei aber immer wieder ausdrücklich auf das Handbuch verwiesen wird. In diesem Kapitel ist auch eine Auflistung der im Grundkurs behandelten Befehle zu finden, die allerdings nicht sehr übersichtlich gestaltet ist! In den folgenden Kapiteln wird, beginnend bei "Lineare Programme" bis zu "Array als strukturierter Datentyp", der Leser in gut gegliederten Lernabschnitten langsam in die Programmierung mit PASCAL eingeführt. Dabei werden die einzelnen Schritte jeweils an Beispielen erklärt und durchgearbeitet.

Fazit: Das Buch ersetzt keineswegs das Originalhandbuch, führt aber in sehr anschaulicher und leicht verständlicher Weise in die Programmierung mit Turbo PASCAL ein und ist gerade für Autodidakten genau die richtige Einstiegslektüre. Weiterführend ist der Aufbaukurs aus der gleichen Buchserie sehr zu empfehlen!

Karlsruhe Obermann

HEFT
21
September
1987

Dal Cin/Lutz/Risse: Programmierung in Modula-2
(Teubner, DM 19,80)

Wer schon in Pascal programmiert hat, wird sich vielleicht über Modula-2 gefreut haben, denn es behebt die alten Pascal-Schwächen und bietet neue Möglichkeiten, die in der Pascal-Definition nicht vorgesehen waren und erst von Pascal-Dialekten eingeführt wurden. Dieses Buch will nun dem Umstieg von Pascal auf Modula-2 erleichtern. Es stellt alle Unterschiede klar heraus und arbeitet sich von elementaren Programmstrukturen zu den Feinheiten vor, die systemnahe Programmierung erlauben. Alle neuen Elemente werden anhand von Listings gezeigt. Die Wirth'schen Systemmodul-Definitionen sind ebenso abgedruckt wie die Syntaxdiagramme (!!!) und die BNF zur Modula-2-Syntax. Ein Buch für Umuund Aufsteiger.

Wirth: Programmieren in Modula-2
(Springer, DM 42,-)

Mir lag von diesem Buch nur die englische Ausgabe vor, die ich hiermit beschreiben will. Das Buch scheint gut geschrieben, eben ein echter Wirth, wieder nur mit der BNF zur Syntax-Definition. Das Buch richtet sich nicht nur an den fortgeschrittenen Pascal-Programmierer und wird wohl auch bald als Standardwerk gelten.

Marthy: Methodik der Programmierung in Pascal
(Springer, DM 42,-)

Dieses Buch ist als Einführung gedacht, wenn der Titel auch nicht so klingt. Es ist kurz gehalten, spricht aber alle wichtigen Themen an. Es scheint verständlich, allerdings fehlen Übungsaufgaben.

Herschel/Pieper: Pascal und Pascal-Systeme
(Oldenbourg, DM 34,80)

Dieses Buch ist gute Einführung in Pascal und bietet einen Gesamtüberblick. Erklärt wird das Standard-Pascal, aber am Ende des Buches werden die meisten gebräuchlichen Pascal-Dialekte noch angesprochen. Leider sind keine Übungsaufgaben vorhanden, die das Buch zu einer hervorragenden Einführung werden ließen. Es werden die Bestandteile eines Pascal-Systems (Editor usw.) allgemein und einige Pascal-Implementierungen beschrieben. Sogar die Hauptmerkmale von Modula werden erwähnt.

Krekel/Trier: Die Programmiersprache Pascal
(Vieweg, 68,-)

Das Buch rechtfertigt seinen Preis allein durch den Umfang, der aber für mich eher abschreckend wirkte. Der Druck dagegen ist billig. Für den normalen Menschen ist das Buch viel zu ausführlich. Es enthält unter anderem 400 Querverweise auf andere Literatur, die kaum jemanden interessieren dürften.

Däßler/Sommer: Pascal
(Springer, DM ??)

Dieses Buch besteht aus zwei Teilen. Der erste Teil ist eine kurze, sehr trockene Einführung in Pascal, die aber durchaus für Anfänger geeignet ist. Der zweite Teil enthält einen vollständigen DIN-Entwurf zu Pascal. Dieser Teil ist bestimmt für jeden interessant, der Beamter ist oder werden will.

Remele: Pascal - systematisch - strukturiert
(Springer, DM ??)

Diese Einführung ist zwar trocken geschrieben, aber sehr ausführlich. Gleichzeitig mit Pascal können noch Struktogramme und ihre Anwendung gelernt werden. Damit erleichtert man sich später den Weg von der Idee zum Programm und ist gleichzeitig up to date, denn Flußdiagramme sind out. Es fehlen aber Übungen.

Doberkat/Rath/Rupietta: Programmieren in Pascal
(Akademische Verlagsgesellschaft, DM ??)

Diese Buch hat nur eine Zielgruppe: Lehrer. Es ist billig aufgemacht. Die eingestreuten Syntaxdiagramme werden nicht im Anhang zusammengefaßt und es gibt keine Übungsaufgaben.

Price: Strukturiertes Programmieren
(IDEA, DM ??)

Hier haben wir eine einfach zu lesende, sehr gut strukturierte und übersichtliche Einführung. Sie ist ausführlich und gut für Anfänger geeignet. Leider fehlen Syntax-Diagramme, dafür gibt es aber reichlich Übungsaufgaben. Das Buch geht nicht tief und sollte später durch ein fortführendes Buch (Wirth's Algorithmen u. Datenstrukturen o.ä.) ergänzt werden.

Plate/Wittstock: Pascal: Einführung - Programmentwurf - Strukturen
(Franzis, DM 58,-)

Dieses äußerst dicke Buch ist für den Anfänger gedacht. Es liefert sowohl Syntaxdiagramme als auch Übungsaufgaben. Es sind viele Listings abgedruckt, deren Druckqualität aber nicht so gut ist, weil sie nur fotokopiert und nicht neu gesetzt wurden. Auf Turbo-Pascal wird auch kurz eingegangen.

Tiberghien: Das Pascal-Handbuch
(Sybex, DM 64,-)

Dieses Buch sollte statt "Handbuch" eher "Lexikon" heißen, denn es liefert eine sortierte Stichwortsammlung zu allem, was zu Pascal irgendwie dazugehört. Natürlich fehlen dabei auch nicht dialektsspezifische Sachen. Vieles dürfte zwar geeignet sein, aber einiges ist wohl so weit von unserem Programmieralltag entfernt, daß es für immer ungelesen bleiben würde. Wer nicht intensiv in Pascal einsteigen will (alles lesen, alles eintippen, alles verstehen, für jeden Rechner), kann sich dieses Buch schenken.

Hering/Scheurer: Fortgeschrittene Programmieretechniken in Turbo-Pascal
(Vieweg, DM 52,-)

Dieses Buch hat zwei Anliegen: 1. Den Aufwand beim Entwickeln von Programmen verringern, indem allgemein oft gebrauchte Routinen vorgestellt werden. 2. Den Entwicklungsweg, das Programmieren, einzustudieren und allgemeine Hilfen zu geben. Dadurch soll wiederum der Aufwand verringert werden, aber hierbei liegt die Betonung auf der Überlegung vor dem Eintippen (was kann ich von anderen Programmen verwenden usw.). Es werden auch Übungen angeboten. Viele Sachen beziehen sich allerdings auf MS-DOS. Dieses Buch ist nur für fortgeschrittene Programmierer gedacht!

Fedtko: Pascal: Algebra - Numerik - Computergrafik
(Vieweg, DM ??)

Dieses Buch zeigt nur Anwendungen von Pascal, und zwar in den drei im Titel genannten Gebieten. Es sind viele Zeichnungen vorhanden, aber trotzdem sollte niemand das Buch in die Hand nehmen, der nicht wirklich am Thema interessiert ist, denn die Sachen sind nicht so einfach zu verstehen (von der Theorie her).

Wostrack: Turbo-Tools und -Utilities unter CP/M
(Heise, DM 49,80)

Hier werden kleinere Hilfsprogramme als Listings vorgestellt, die das Programmieren erleichtern sollen. Ich glaube nicht, daß es sich lohnt, dieses Buch zu kaufen, denn die Programme gibt es bestimmt auf Diskette und da gehören sie auch hin (wer tippt das schon alles ab). Die Erläuterungen hätten als Doku-File dazugehört, sollten hier wohl aber als Käuferfang dienen (oder was weiß ich).

Rauch: Modell der Wirklichkeit - Simulation dynamischer Systeme mit dem Mikrocomputer
(Heise, DM 29,80)

Hier haben wir ein sehr interessantes Buch über Modelle und Modellbildung mit vielen Listings. Zwar wurde Turbo-Pascal verwendet, aber auf einem umgebauten Apple und mit vielen Grafik- und sonstigen Erweiterungen in der Sprache. Diese Anpassungen (der Software) sind sorgfältig dokumentiert und müßten nachvollzogen werden, um sie auf einer anderen Maschine lauffähig zu machen. Das bedeutet viel Arbeit und ist nur für den absolut Interessierten zu empfehlen.

Lücke: Turbo-Pascal
(Markt & Technik, DM 49,-)

Dieses Buch bezieht sich vor allem auf die MS-DOS-Version des Turbo-Pascal, könnte aber trotzdem für uns geeignet sein. Es liefert keine Syntaxdiagramme, scheint aber vollständig und ausführlich zu sein. Zwar fehlt (zumindest im Inhaltsverzeichnis) etwas die Strukturierung, aber es werden auch einige Turbo-Tools bzw. Dienstprogramme beschrieben.

Ottmann/Widmayer: Programmierung mit Pascal
(Teubner, DM 18,80)

Dieses Buch ist für den absoluten Anfänger gedacht. Es liefert Syntaxdiagramme, aber keine Übungen. Der Stoff wird nicht sehr vertieft und es werden nur Programmenteile als Beispiele angeführt, was die Sache übersichtlicher, aber nicht gerade leicht nachvollziehbar macht (eintippen und laufen lassen geht nicht!). Der Druck ist vorbildlich und macht das Lesen zur Freude.

Brosell/Zhou: Modula-2
(Springer, DM 69,-)

Dieses Buch ist eine dicke, vollständige Einführung in Modula, die wohl auch von einem Anfänger verstanden werden kann. Es sind viele Listings vorhanden.

Amstel: Programmieren: Die Entwicklung von Algorithmen in Pascal
(Addison-Wesley, DM 48,-)

Eigentlich stimmt der Titel dieses Buches nicht. Das "in Pascal" dürfte bestenfalls in Klammern stehen, denn Amstel geht nur um den Weg der Entwicklung von Algorithmen, wobei Pascal nur zufällig die Sprache ist, in die das Endergebnis der Überlegung übertragen wird. Dementsprechend werden viele Pascal-Eigenschaften wie Dateien, Mengen und Pointer überhaupt nicht angesprochen. Ein irreführender Titel.

Wilson/Addyman: Pascal
(Hanser, DM 28,-)

Dieses Buch bietet meiner Meinung nach äußerlich alles, was ein gutes Pascal-Lehrbuch ausmacht: Übungen, Syntaxdiagramme, verständliche Sprache, Ausführlichkeit, guten Druck. Ob der Inhalt das hält, kann ich nicht sagen, aber die 28,- DM sind ein gutes Argument, es mal zu probieren.

Mittelbach/Wermuth: Turbo-Pascal in der Praxis
(Teubner, DM 24,80)

Dieses Buch ist nicht für den Anfänger geeignet, es ist für Umsteiger gedacht, die sich nach einem anderen Pascal nun in Turbo versuchen wollen. Evtl. könnte man es als Ergänzung zu einem allgemeinen Pascal-Anfängerbuch nehmen. Es wird eine Einführung in den Turbo-Editor angeboten. Das Buch bezieht sich sowohl auf CP/M als auch auf MS-DOS.

Renner: Turbo-Pascal
(Vogel, DM 43)

Dieses Buch ist nicht unbedingt für Anfänger geeignet; der Leser sollte schon vorher programmiert haben, wenn auch nicht unbedingt in Pascal. Der Inhalt bezieht sich auf MS-DOS, aber Editor/Compiler (Optionen) sind gut und ausführlich dokumentiert.

Herschel: Turbo-Pascal
(Oldenbourg, DM 29,80)

Dies ist eine elementare Einführung für Anfänger. Leider fehlen Übungsaufgaben, aber Syntaxdiagramme sind vorhanden. Der Editor wird auch beschrieben, allerdings beziehen sich viele Dinge auf MS-DOS.

Pomberger: Softwaretechnik in Modula-2
(Hanser, DM ??)

Mit diesem Buch sollen Leute angesprochen werden, die viel programmieren und dabei große Programmsysteme herstellen. Methoden und Arbeitsweise werden dabei direkt mit der dafür bestens geeigneten Sprache Modula-2 beschrieben. Ohne zumindest elementare Kenntnisse von Pascal wird es schwer sein, dieses Buch zu verstehen. Dafür wird Modula so gut erklärt, daß es auch als Modula-Einführung für Umsteiger geeignet ist. Das Buch ist für berufliche u. Vollblut-Programmierer.

Digital Research: CP/M-Operating System Manual
(Markt & Technik, DM 38,-)

Bei diesem Buch handelt es sich um einen Nachdruck (bzw. mehr eine Kopie) der englischen Originalausgabe. Die Druckqualität ist schlecht und die Druckfehler wurden nicht korrigiert. Dafür hat man aber die Original-Dokumentation der CP/M-Vertreiber in der Hand, die auch die Struktur des Betriebssystems einigermaßen gut vermitteln. Leider sind die Programmbeispiele unbrauchbar, denn 1. sind sie in 8080-Maschinensprache geschrieben, 2. sind es nur Programmrümpfe (an den wichtigen Stellen: setzen sie hier Ihre Routine ein) und 3. sind sie, falls ausführlich, auf ein vollkommen unbrauchbares (jedenfalls für unsere Zwecke) System zugeschnitten. Empfehlen kann ich es nicht.

Weiß: CP/M - ein Betriebssystem stellt sich vor
(Hüthig, DM 39,-)

Dieses Buch stellt nur die CCP-Befehle und die normalerweise vorhandenen Dienstprogramme vor. Zum Programmieren mit CP/M ungeeignet.

Pol: Vom Umgang mit CP/M
(IWT, DM 43,-)

Dieses Buch ist sehr dick und ausführlich. Es erklärt alle Befehle, Hilfsprogramme und was man sonst noch braucht. Aber der Anhang ist nicht ausreichend, um mal schnell etwas nachschlagen zu können. Außerdem beziehen sich alle Programmbeispiele auf den 8080. Aber das Buch ist für den absoluten Computer-Neuling bestens geeignet, denn es führt elementar in die Benutzung des Computers ein (wie mache ich ein Backup usw.).

HEFT
21
September
1987

Pascal

(Bücher, die evtl. interessant wären)
(keine Garantie für Preis und ISBN)

Verlag	Autor, Titel	Preis	ISBN	Verlag	Autor, Titel	Preis	ISBN
<u>Franzis</u>	Busch: Der sichere Einstieg in P.	48,-	3-7723-7861-7	<u>Teubner</u>	Bielig/Schulz: 3-D-Graphik in P.	25,-	
	Klein: Was ist Pascal?	32,-	3-7723-7001-2		Erbs: 33 Spiele mit Pascal	32,-	
<u>Sybox</u>	Miller: Pascal-Programme (60 Algorithmen)	58,-	3-88745-007-8		Weber: Pascal in Übungsaufgaben	24,80	3-519-02539-6
	Hunter: 50 Pascal-Programme	48,-	3-88745-065-5	<u>Humboldt</u>	Schäuck: Programmiersprache Pascal Schritt für Schritt	9,80	ht 551
	" " (Turbo)	48,-	3-88745-645-9	<u>Springer</u>	Blaschek/Pomberger: Einführung in die Programmierung mit Modula-2	45,-	3-540-15158-3
	Braun: Das Arbeitsbuch zu T.-P.	48,-	3-88745-629-7		Gleaves: Modula-2 für Pascal-Pro- grammierer	39,-	3-540-15157-5
	Rollke: Das Turbo-Pascal-Buch	42,-	3-88745-608-4	<u>Girardet</u>	Weber/Mrowka: Grundkenntnisse P.	5,95	22910
<u>Hüthig</u>	Fischer: Datenverarbeitung mit Turbo-Pascal	45,-	3-7785-1476-8	<u>Taylorix</u>	Hamann/Lewerenz: Datenverarbeitung mit Turbo-Pascal	46,-	3-88640-037-9
	Lange: Turbo-Pascal - Ständig im Griff (Referenz)	21,80	3-7785-1313-3	<u>Addison-Wesley</u>	Angeli: A-W Kompaktführer T.-P.	22,80	3-925118-15-2
<u>Vieweg</u>	Harbeck: Einführung in T.-P. unter CP/M 80	32,-	3-528-04440-3	<u>Hanser</u>	Joepgen: Turbo-Pascal	58,-	3-446-14431-5
	Kaier/Rudolfs: T.-P. Wegweiser für Micro.: Grundkurs	48,-	3-528-04415-2	<u>Vogel</u>	Baumann: Programmieren mit Pascal (Anfänger)	25,-	3-8023-0667-8
	Schneider: Einführung in Pascal	26,-	3-528-04200-1	<u>Oldenbourg</u>	Blaha/Haselberger/Schärf: Program- miertraining Pascal (Bd. 1)	26,80	3-486-20003-8
	Pinke: Strukturiertes Programmieren in Pascal	35,-	3-528-04439-X		Schärf/Vertneg: P. für Kaufleute	29,80	3-486-20000-3
<u>Heise</u>	Feller: Pascal-Datenstrukturen u. prakt. Anwendungen (fortgesch.)	44,-	7020-8		Vahrenkamp: P. auf Mikrocomputern	28,80	3-486-28921-7
	Gilder/Barrus: 112 T.-P.-Programme für math. und elektrotechn. An- wendungen (Anfänger)	49,80	3-88229-102-8	<u>Dümlers</u>	Heidemann: Algorithmen u. Daten- strukturen	8,80	3-427-42771-4
	Claßen: Mathematikprogramme in P.	36,80	3-620-00114-6	<u>Bibl. Institut</u>	Kaucher/Klatte/Ullrich: Pascal	26,80	3-441-05796-3
	Stede: T.-P.-Programme zur künstl. Intelligenz	?	9126-5	<u>rororo</u>	Ledgard/Singer: Pascal auf der Spur	12,80	8109
	Eisenecker: Matcalc - Matrizen- rechnung in UCSD-/T.-P.	34,80	3-88229-143-5	<u>IWI</u>	Rompel: Turbo-Pascal-Anwendungen	48,-	3-88322-167-8
<u>Markt & Technik</u>	Ciric/Thies: Turbo-P. Kursbuch 1.: Einführung	49,-	3-921803-34-9		Bürgel: Modula-2	68,-	3-88322-169-4
	Kassera: Programmieren mit T.-P. (Ver. 3.0, MS-DOS)	59,-	3-89090-159-X	<u>McGraw Hill</u>	Sand: Pascal-Programmiertechnik für Fortgeschrittene	35,-	3-89028-039-0
				<u>Expert</u>	Scheibl: Pascal-Einführung	48,-	3-8169-0093-3
				<u>Vahle</u>	Schönefelder: Turbo-P. Softregister	13,80	3-9800252-2-5

Verlag	Autor, Titel	Preis	ISBN
???	Kwiatowski/Dierig: Pascal-Computerspiele	20,80	3-7724-5374-0
	Welsh/Elder: Einführung in P.	42,-	3-437-40109-2
	Reid-Green: Computerspiele u. Grafik	56,-	3-528-04357-1

CP/M

Verlag	Autor, Titel	Preis	ISBN
<u>Sybox</u>	Miller: Programmieren mit CP/M	52,-	3-88745-077-9
	Steinhoff: SYBEX-Ratgeber CP/M	55,-	3-88745-301-8
<u>Hüthig</u>	Weiß: CP/M - Ständig im Griff (Referenz)	21,80	3-7785-1267-6
<u>Vieweg</u>	Schneider: Einführung in die Anwendung des Betriebssystems CP/M	29,80	3-528-04252-4
<u>Markt & Technik</u>	Digital Research: CP/M Plus (3.0)	38,-	3-89090-371-1
	D. R.: Programmer's Utilities and SID for CP/M 2.2/3.0	49,-	3-89090-372-X
<u>Oldenbourg</u>	Blackburn/Taylor: Terminal-Buch CP/M	24,80	3-486-29331-1
<u>Addison-Wesley</u>	Boon/Pelsaeker: A-W Kompaktführer CP/M	22,80	3-925118-13-6
<u>Vahle</u>	Bürenkämper: CP/M-Softregister	13,80	3-9800252-6-8
<u>Girardet</u>	Pape: Das Mikrocomputer-Betriebssystem CP/M	5,95	3-7736-2359-3
<u>Röckrath</u>	Kämpf: CP/M-Assembler-Listing	59,-	3-925074-11-2

1. Vorsitzende Hartmut OBERMANN
Schwalbacher Straße 6
6289 Heidenrod 1
☎ 06124 /3913

2. Vorsitzende Gerald SCHRÖDER
Am Schützenplatz 14
2105 Seevetal 1
☎ 04105 /2602

Hardwarekoordinator Eckehard KUHN
Im Dorf 14
7443 Frickenhausen 1
☎ 07022 /45417

Diskotheke Werner FÖRSTER
Christoph-Krebs-Straße 9
8720 Schweinfurt
☎ 09721 /21841

Redaktion Jens NEUEDER
Panoramastraße 21
7178 Michelbach /Bilz
☎ 0791 /42877

Autoren Die Redaktion bedankt sich bei den im INHALTSVERZEICHNIS genannten Autoren für die Mitarbeit an der Club-INFO.

Druck Peter Spieß
Trugenhofenstraße 27
8859 Rennertshofen 1
☎ 08434 /454

Bankverbindung des CLUB 80
Postgirokonto Peter STEVENS
Sonderkonto CLUB 80
Konto-Nummer 285 491 - 465
Postgiroamt Dortmund
BLZ 440 100 46

*Das INFO erscheint zweimonatlich.
Es erfolgt keine Zensur oder Kontrolle
der jeweiligen eingesendeten Infobeiträge
durch die Redaktion.*

HEFT
21
September
1987

SCHLUSS

Hallo Club-Mer,

entgegen ~~aller~~ Versprechungen kann ich leider erst jetzt das neueste INFO herausgeben, ich hoffe, Ihr habt trotzdem Eure Freude daran.

Es ist ja nun schon fast (unabsichtlich!!) zur Gewohnheit geworden, daß das INFO immer etwas später bei Euch eintrudelt. Ich kann aber im Moment die von mir selbst gesetzten Wunschtermine für das Erscheinen der INFO's zu meinem Bedauern nicht halten, zuviele private und berufliche Termine stehen ins Haus und müssen bewältigt werden. Ein Ende meines persönlichen "Terminboom's" ist vor dem Frühjahr/Sommer 88 nicht in Sicht, deshalb werde ich Euch also noch ein paar INFO's mit "unregelmäßigem" Erscheinungsdatum zumuten müssen.

Ein Informationsverlust entsteht bekanntlich dadurch nicht, unser INFO wird höchstens dicker, je länger ich "samle". Ich hoffe, Ihr habt Verständnis dafür.

Nun möchte ich mich noch etwas erfreulicherem Dingen zuwenden. Diesem INFO liegen zwei "Extrawerke" bei:

- Zum Einen ist es eine Kurzanleitung die ich auf kartoniertes Papier kopiert habe. Etwas stabilisiert -damit Ihr sie auch richtig nutzen könnt-!
- Zum Anderen gibt es eine "Stichwort-Infothek", die als Unterstützung beim Suchen nach INFO-Artikeln dient.

Das war's auch schon wieder.

Ich wünsch Euch viel Spaß mit Dieser -
und bis zur neuesten INFO grüßt Euch

Jaus

HOFACKER

ING. W. HOFACKER GMBH



Ing. W. Hofacker GmbH - Tegernseer Straße 18 - 8150 Holzkirchen

Herrn
U. Böckling
Gartenstr. 31

5410 Höhr-Grenzhausen

Der große Fachverlag für

- Elektronik
 - Neue Technologien und Medien
 - Home- and Personalcomputer
 - Software für Microcomputer
- Seit 1968

Postanschrift:

HOFACKER-VERLAG

Ing. W. Hofacker GmbH
Tegernseer Straße 18

D-8150 Holzkirchen/Obb.

Telefon: 0 80 24/ 73 31

Telex: 52 69 73

Sitz der Gesellschaft: Holzkirchen

Unser Zeichen Datum

JM

21. 7. 1987

Ihr Zeichen

Ihre Nachricht

Unsere Nachricht

Sehr geehrter Herr Böckling,

vielen Dank für Ihren Brief vom 20. 7. 1987 bezüglich Beiträgen zu TRS-80.

Ich bin damit einverstanden, daß Sie Beiträge bezüglich TRS-80 aus der ELCOMP in Ihrer Clubzeitschrift veröffentlichen. Ich bitte Sie aber, zu jedem Beitrag eine Quellenangabe zu machen (ELCOMP Heft Nr... Hofacker Verlag, Holzkirchen). Bitte schicken Sie mir doch gelegentlich ein Belegheft.

Vielleicht können Sie auch noch auf die folgenden Spiele (Cassette) hinweisen, die bei uns noch für TRS-80 lieferbar sind:

- TRS-80 OPERA
- The Great Race
- Owl Tree

Jede Cassette ist für DM 5,- erhältlich. Lieferung gegen Vorkasse auf unser Postscheckkonto München, Kto.-Nr. 15 994-807, BLZ 700 100 80 oder per Nachnahme (+ DM 6,50 NN-Gebühr).

Mit freundlichen Grüßen
Ing. W. Hofacker GmbH Verlag

J. Maier

PS.: Ich gehe davon aus, daß Ihre Club-Infos nicht kommerziell verbreitet werden, sondern nur an Ihre Clubmitglieder abgegeben werden.

Amstgericht München
HRB 50 295

Geschäftsführung:
Josefa Maier

Bankverbindung:
Dresdner Bank Mchn., Kto.-Nr. 5 390 370
(BLZ 700 800 00)
Postscheckkonto Mchn. 159 94-807

Frachtsanschrift:
Mchn. Ostbahnhof

Expresgutanschrift:
Mchn. Hauptbhf.