CLUBINFO

32. AUSGABE

KONTAKTADRESSE : CLUB 80 / FRITZ CHWOLKA / SAARSTRABE 34 / 5173 ALDENHOVEN
TEL.: 02464 / 8920

Inhaltsverzeichnis

C L U B 80

Fritz Chwolka

5173 Aldenhoven, den 03.12.1990
Saarstr.34
(02464)8920

Liebes Clubmitglied!
Ich hoffe den meisten bin ich inzwischen ausreichend bekannt, auch wenn dies nicht unbedingt in meiner Funktion als erster Vorsitzender des Clubs ist.
Für die anderen möchte ich ein Wenig zu meiner Person erzählen.
Ich bin 32 Jahre alt, von Beruf Betriebsschlosser auf einem Kohlekraftwerk und seit ca. 1979 mit dem Z80 verbunden.
Neben diversen Rechenschiebern habe ich auch ein paar von diesen Kisten, welche sich Computer nennen. Ohne jetzt eine alphabethische Liste erstellen zu wollen zähle ich die *Kisten* mal auf. Da wären ein Tatung TPC-2000, ein Commodore C128, Epson PX-8, ein Morrow MD-3 und mein Lieblingsmodell der Genie IIIs. Im Moment baue ich, wie einige andere auch, den Z280 Rechner vom Dipl.Ing. Tilman Reh auf. Tilman sieht zwar nicht unbedingt gerne die Voranstellung des Berufstitels, aber Ehre wem Ehre gebührt.
Ich hoffe, daß in 1990 wenig Klagen über den Vorstand zu hören waren, außer vielleicht über die manchmal lange Wartezeit beim Beantworten der Briefe, aber dies ließ sich nicht immer vermeiden.
Zwischendurch habe ich den Herrn Thomas Holte ausfindig gemacht und angebaggert. Herr Holte hat für den Genie IIIs eine gute Addaption für CP/M+ geschrieben und mir (inclusive Quellcode) zugesandt. Gleichfalls wurde mir erlaubt den Quellcode des Herrn Holte zum privaten Gebrauch weiterzugeben. Wer also das CP/M+ BIOS für GENIE IIIs im Quellcode haben möchte schicke mir bitte drei rückadressierte Disketten zu.
Abschließend wünsche ich Euch erhohlsame und freudige Feiertage und einen schönen und ruhigen Jahreswechsel.

Mit vielen Grüßen

Rein informativ möchte ich Euch noch die neuen Mitglieder kurz vorstellen.

Neue Mitglieder seit meiner Amtsübername:

Achim Benner,Vorm Mühlberg 1, 5190 Kreuztal 3
Rechner CPC,Genie IIIs. Achim hat einiges an PD von mir erhalten,

Willi Johnen, Hansemannstr.1,5160 Düren
Rechner Genie IIIs mit 10MB HD, Willi sitzt auf der GDOS Schiene und ließ sich noch nicht zum CP/M überreden. Er hat viel mit dem GBASIC gemacht und läßt Uns hoffentlich was zukommen.

Franz Mössel,Schaferstr.12,I-39012 Meran
Rechner CPC. Franz ist auch im SCUG Mitglied und inzwischen Z3PLUS Anwender.

Jürgen Peters, Heukoppel 14, 2000 Hamburg 71
Rechner SHARP MZ800. Jürgen macht Amateurfunk (YAPP) und hat für das Z-MODEM Programm aus der PD eine Anpassung für den Sharp geschrieben. Nebenbei ist Jürgen noch im SCUG-München.

Werner Schilling, Ehndorfer Str.340, 2350 Neumünster
Rechner CPC 6128, Applle kompat., Werner ist Packet-Radio User mit Rufzeichen DB20U.

Rainer Schmitz, Dornierstraße 17, 7320 Göppingen
Rechner Joyce + Vortex 5.25". Rainer ist Modelleisenbahn-Fan und macht da auch einiges mit dem Joyce. Er ist ebenfalls Z3PLUS Anwender.

Uwe Schobert, Petrus-Waldus-Str. 14, 7136 Oetisheim
Rechner Triumpf-Adler P3 (8085). Uwe sucht ständig nach Programmen für den 8085 Prozessor und arbeitet (noch) mit MICRO SHELL. Momentan versuchen wir ZCPR für 8085 zu installieren.

Frank Schoof, Elpke 5, 4800 Bielefeld 1
Rechner Genie I (64KB). Frank arbeitet mit GDOS/NEWDOS und hat eine grafische Benutzeroberfläche entwickelt, welche er uns demnächst wohl vorstellen wird.

Svend A. Sørensen, Bogholder Allee 76A, DK-2720 Vanløse
Svend hat einen Eigenbaurechner und mir auf mein letztes Schreiben noch nicht geantwortet. Vielleicht hat jemand Interesse an Dänemark und schreibt Ihm ? Ebenso kann Svend einiges an PD gebrauchen.

Stefan Stumpferl, Hasenbergl-Straße 57, 8000 München 45
Rechner: CPC6128, Genie IIIs, PROF 180x
Stefan arbeitet an Betriebssystemerweiterungen und hätte gerne Informationen zu Erweiterungen für den Genie und PD-Software.

Wolfgang Schwarz, Schwedenring 6, 8850 Donauwörth
Rechner C128(D). Wolfgang hat noch nicht viel an PD für seinen C128, kann aber Kaypro-Format zwecks Konvertierung lesen. Ich hoffe daß wir da etwas helfen können.

Dies als Kurzinformation über die neuen Mitglieder, welche sich aber hoffentlich 'persöhnlich' vorstellen werden.

Lieber Jens,

da ich nun seit kurzem auch diesem Club angehöre, bin ich mal so frech und fange gleich mit "Du" an. Ich hoffe, niemand nimmt mir das übel. Wie ich dem Probeheft des Club-Infos entnommen habe, ist es üblich sich erst mal vorzustellen. Diesen Brauch finde ich gut. Ich will gleich damit anfangen.

Ich bin schon ein ziemlicher Oldie. Seit kurzem bin ich stolze 36 Jahre. Von daher fällt mir manches nicht ganz so leicht, wie den übrigen Vollprofis in diesem Club. Vor allem auch, weil ich ganz und gar kein Profi bin. Ich habe zwar meinen Computer selbst (nach)gebaut. Aber ich glaube, das kann im Prinzip jeder, der es will. Ich habe bei mir den HDR-Klein-Computer stehen. Er läuft mit einer stinknormalen Z80 mit gut 6 MHz. Die Grafikkarte kann 256 x 512 Punkte auf 4 Seiten darstellen und hat eine eigene CPU (EF 9366) und natürlich eigenen Speicher.

Ansonsten habe ich nichts Weltbewegendes zu bieten. Ich kann mit einem Spezial-Bios verschiedene Formate bearbeiten (leider nicht alle). Den Einstieg in die DFÜ habe ich noch nicht gefunden. Daran wird aber gerade gearbeitet.

Ich arbeite meisten mit MZ-COM (phantastisch!!). Projekte, die ich anfange, brauchen meist ziemlich viel Zeit. Derzeit stehen an: DFÜ, Installation von CP/M+, ZCPR usw.

Für Hilfe bin ich immer dankbar. Natürlich versuche ich auch gern zu helfen.

Nebenbei bin ich bei der hiesigen Sparkasse beschäftigt. Ich habe die unangenehme Aufgabe ausstehende Beträge "einzutreiben". Ich bin verheiratet und Vater einer Tochter.

Soviel zu mir selbst.

Claus Littmann
Plockhorst, Zum Spring 15, 3155 Edemissen

Termine... Termine... Termine... Termine... Termine

Clubtreffen Süd	Röfingen	25.01.- 27.01.91
Jahreshauptversammlung	.....	.....- 91
Clubtreffen Nord	.....	.....- 91

Redaktionsschluß für die Clubinfo's 1991 ist jeweils der letzte Tag der folgenden Monate:

März
Mai
Juli
Oktober
Dezember

Internationale Musikmesse	Frankfurt	März 91
CEBIT	Hannover	13.03.- 20.03.91
IAA	Frankfurt	12.09.- 22.09.91
Büro-data	Berlin	16.10.- 19.10.91
SYSTEMS	München	21.10.- 25.10.91

Traurige Nachricht!
Mit großem Bedauern mußten wir erfahren, daß unser langjähriges Clubmitglied und guter persönlicher Freund Walter Piller am Montag den 07. Januar plötzlich verstorben ist.

Walter, der in der Nähe von Zürich zuhause war, hatte trotz seiner siebzig Lenze noch viele Pläne. Noch am Wochenende vor seinem Tod hat er uns angerufen und angekündigt, daß er bei gutem Wetter zum Südländertreffen kommen möchte. Um so unerwarteter erreichte uns die Nachricht von seinem plötzlichen Ableben durch Herzversagen.

Meine Frau und ich haben, auch im Namen des CLUB 80. Walters Frau, Helene Piller, herzliches Beileid ausgesprochen.

Jutta und Hartmut Obermann

Einige Gedanken zum Fortbestand des Club 80

P'anta rh'ei, alles fließt, wußte schon der griechische Philosoph Heraklit vor über 2000 Jahren zu sagen. Das ist heute aktueller denn je! Wenn ich mir betrachte, welche Fortschritte die Computertechnik allein in den letzten 10 Jahren gemacht hat, kann ich nur staunen. Leider hat unser Club 80 sich diesem Tempo nicht anpassen können.

Trotz eindringlicher und warnender Appelle ist nun passiert, was der Jens vorausgesagt hat: Mangels Masse war er nicht in der Lage, mehr als zwei INFOS in diesem Jahr herauszubringen. Nun ist auch der beste Redakteur nicht in der Lage, aus null Beiträgen ein INFO zu machen. Aber warum ist die früher so reichlich sprudelnde Quelle an Beiträgen versiegt? Liegt das vielleicht auch daran, daß sich viele Mitglieder heimlich oder unheimlich mit dem ach so geschmähten MS-DOS befassen und bis zur Stunde keine Basis in unserem Club 80 gefunden haben?

Warum nur ist dieses MS-DOS in unseren Reihen so verpönt? Kann man den nicht auch unter diesem Betriebssystem in Assembler oder den verschiedensten Hochsprachen programmieren? Ist das das DOS so ausgereift, daß es daran nicht zu verbessern gibt? Und kann man nicht auch an diesen Rechnern schrauben und löten?

Meiner Meinung nach befinden wir uns an einem Scheideweg. Machen wir weiter so wie bisher, wird der Club 80 bald sanft entschlafen sein. Es möge daher jedes Mitglied eindringlich mit sich zu Rate gehen, ob wir uns nicht den Realitäten beugen und MS-DOS in unseren Reihen zulassen sollen.

Ich würde eine Öffnung sehr begrüßen. Es wäre schade, wenn eine Einrichtung wie der Club 80 so sang- und klanglos untergehen würde. Aber ich bin zuversichtlich, daß wir auch diese Kurve kriegen und den Club 80 zu seiner alten Vitalität zurückführen können.

"In den Sternen steht's...

...nicht geschrieben", was aus dem CLUB 80 wird!
Oder haben wir auch einen Astrologen unter uns? Der verzeihe mir meine "Ungläubigkeit" und erstelle uns auch sogleich ein Horoskop, bitte! Ist es positiv, so wollen wir daran glauben. Andernfalls nicht.
Auch ohne der Astrologie zu frönen, hatte ich im INFO 28 ein "Horoskop" veröffentlicht. Ohne "Tierkreiszeichen"und ähnliche Mätzchen - Verzeihung: Mystik, sondern mit jener Mathematik, die man Extrapolation nennt. Extrapolationen sind - sofern überhaupt zulässig - ihrem Wesen nach stets ungenau. So sind es bis heute (wir schreiben im Moment den 6.11.1990) zwar nicht, wie von mir extrapoliert, nur zwei, aber doch bloß ganze drei INFOs in 1990, also genau so wenig wie in 1989!
Gewiß kommt es nicht auf die Anzahl, also die Quantität, sondern auf die Qualität an; dennoch:
Erinnert sei an den eindrucksvollen Bericht des holländischen TRS-80-Users *Joop Groenendijk*, Sekretär der "TRS-80 Gebruikers Vereniging", im INFO 21 (bereits September 1987!) Bei dem klangvollen "Grünen Deich" sonnten sich 2000 (zweitausend!) Z80-Anhänger! Und früher waren es sogar noch mehr! Dagegen sind wir ein klägliches Häuflein von z.Z. 63. Und über noch mehr km² verstreut als die Holländer!
Was können wir von *Joop* lernen?
Sinngemäßes Zitat: "Diese Zahl (2000) steigt z.Z. wieder, seit wir den Entschluß gefaßt haben, daß wir uns auch auf den Gebrauch des PC einrichten!" (Hört hört!)
In ähnlichem Sinn äußerte sich *Paul-Jürgen Schmitz* auf der nächsten Seite jenes INFOs.

Nachdem ich nochmal alle INFOs auf die Früchte von (Hard- und Software-)Bastlerfleiß abgeklopft und diese Früchte (=Arbeitsergebnisse, besser: Hobby-Ergebnisse) mit der Leistungsfähigkeit meines neuen, erweiterten XT's von EPSON (mit Festplatte und - pfui! - MS-DOS) verglichen hatte, ging mir ein Licht auf, worauf die Treue zum TRS-80 bzw. GENIE beruht: Es ist das Vergnügen, mit eigenen (!) Einfällen und Anstrengungen in diese Geräte das hineinzupacken und aus ihnen das herauszuholen, was die "Kompatiblen" (die "IBM-Diener"), die nun unwiderruflich en vogue sind, mit Leichtig- und Selbstverständlichkeit an Leistung und Komfort von Haus aus mitbringen!!

Dieses Vergnügen kann in Frust enden - oder gelingen; je nach Geschick und Geschicklichkeit. Maßstab für den Erfolg dieses Vergnügens bzw. Bemühens, wie gut das Ziel erreicht wurde, wird stets und in Zukunft immer stärker der "Kompatible" sein. An dem müssen wir unsere Künste und deren Früchte messen lassen.

die „Trem-Automatik"!

"Messen" aber heißt "Vergleichen"!
(Hier spricht ausnahmsweise mal der Physiker.)

Sonst schweben wir im luftleeren Raum - und somit "entschwebt" uns einer nach dem anderen... und früher oder später sind es nur noch 6 statt 60. (Gewiß, auch das ist noch eine hübsche Zahl. Aber diese Sechs hocken ohnehin seit eh und je eng zusammen.)

Kurzum:
Ihr "Halbaussteiger" (das ist wohl die Hälfte aller), schreibt über eure Erfahrungen mit MS-DOS und Zubehör, vergleicht (z.B. Programme, die ihr für beide Systeme geschrieben habt; Utilities beider oder auch mehrerer Systeme - etwa nebst CP/M - u.a.) und zieht Schlüsse daraus, die für uns alle von Nutzen sind!

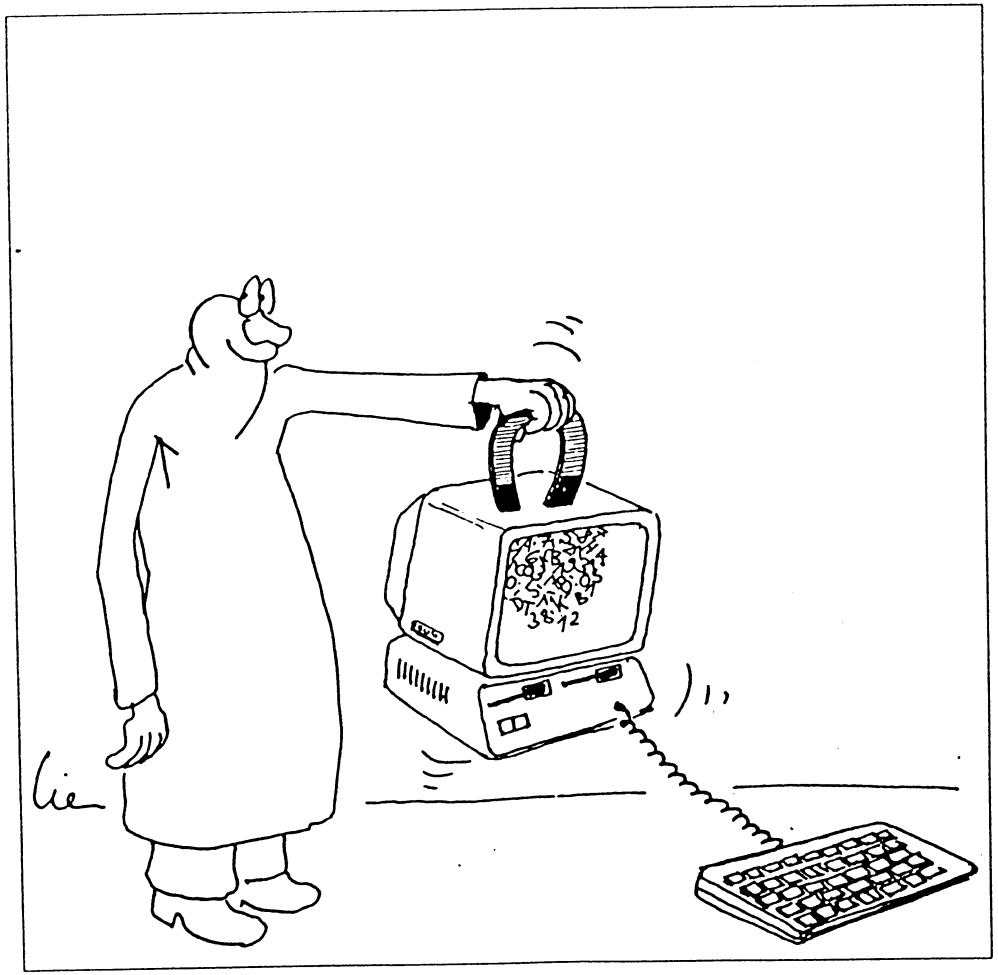
Übrigens: INFO Nr.31, Seite 6 ("Nachbemerkung" von Arnulf) läßt hoffen! Denn:

EX ORIENTE LUX!

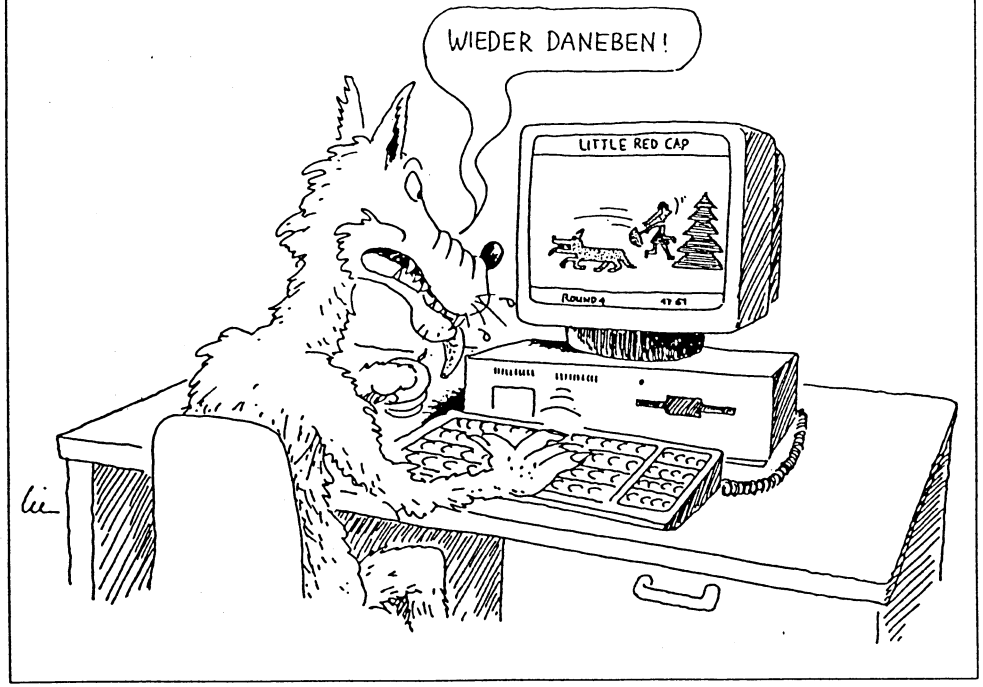
Das Tolle dabei:
"Oriente" ist diesmal nicht der Orient oder Nah- oder Fernost, sondern sehr nahes neues altes Deutschland!
Deren Freaks werden sich wohl bald auch die neuen Kisten mit MS-DOS anschaffen - es sei denn, ihr schenkt ihnen eure alten (und macht dann selbst mit den "Kompatiblen" weiter!)
Also heißt es dann sowieso:

Macht)
Schnell)
–)
Das Tor) ⟹ z.B.: MS-DOS!
Offen für andere)
Systeme!)

In diesem Sinn: Euer Kay 7. XI. 90



Adventure



Schneller Booten

Wenn man das System, einfach so wie es ist, nur bis zum DOS-Prompt hochfährt, läßt sich das kaum beschleunigen, es dauert halt seine Zeit, bis die Systemspuren gelesen sind. Wenn aber anschließend per Autostart noch einige Programme für diverse Einstellungen aufgerufen werden, läßt sich einiges tun. Wenn der Rechner beim Booten ewig auf der Floppy rumsägt und von Anschlag zu Anschlag hetzt, ist es dafür allerhöchste Zeit. Beim NEWDOS/GDOS kann man leider nicht viel an der Dateiorganisation drehen, aber beim CP/M ist es so, daß Files immer schön aufsteigend in die nächsten freien Sektoren hinter dem Directory kopiert werden. Der Trick ist nun, daß man die beim Booten benötigten Programme als erste auf eine leere Floppy kopiert. Dann hat man die Bootspur(en) ganz außen, anschließend kommt das Directory und gleich dahinter die Startup-Files. So muß der Kopf nur über höchstens 4 bis 5 Spuren bewegt werden und es geht auf einmal viel schneller und leiser.

Alexander Schmid

Fehler in EASE.COM

EASE ist zur Zeit mein liebster Error-Handler, weil er zwei Features miteinander verbindet. Einmal kann man eine fehlerhafte Befehlszeile editieren (wie sich das für einen Error-Handler nunmal gehört) und zweitens kann man vergangene Eingaben zurückholen, er ist also Error-Handler und History-Shell in einem. Wenn man will, zeigt er einem mit einem Patch auch zusätzlich noch die aktuelle Uhrzeit an.
Nur einen kleinen Fehler hat das Ding. Wenn man einen Befehl aus dem RCP mit einem Parameter (z.B. P 200) aufruft wird jedesmal das Byte an der Adresse 100 gelöscht. Das ist sehr ärgerlich, weil dort ausgerechnet ein Sprung steht (stehen muß) und so der Restart-Trick mit der leeren Datei bzw. das GO nicht mehr funktionieren kann. Auch nach SAVE wird man sich wundern, daß das gerettete Programm nicht mehr läuft.
Statt nun dort jedesmal das 'C3' von Hand wieder einzufügen sollte man lieber gleich EASE patchen, sodaß das ein für allemal aufhört. Die gesuchte Stelle ist bei 0D4DH (Version 2.0), wo ein LD BC,0080 steht, womit der Buffer von 80H bis FFH gelöscht werden soll. Für diese Aktion reicht aber ein LD BC,007F und das Byte bei 100 bleibt anschließend verschont.

Alexander Schmid

Die RS232 des Model 4p

Bei der Anpassung von Terminalprogrammen an den eigenen Rechner stellen sich immer wieder dieselben Fragen, nämlich auf welche Ports man wie zugreifen muß und welche Bits man braucht. Am besten macht man sich einen kleinen Zettel und legt oder klebt ihn ins Handbuch, weil man das sonst mit Sicherheit bis zum nächsten mal wieder vergessen hat.

Für das Model 4p sieht das folgendermaßen aus:

Stellt fest, ob ein Byte zum Abholen bereit steht:

```
INPSTAT   IN   A,(0EAh)
          AND  80h
          RET
```

Liest ein Byte von der seriellen Schnittstelle ein:

```
READ      IN   A,(0EBh)
          RET
```

Stellt fest, ob ein Byte gesendet werden kann:

```
OUTPSTAT  IN   A,(0EAh)
          AND  40h
          RET
```

Gibt ein Byte an die serielle Schnittstelle aus:

```
WRITE     OUT  (0EBh),A
          RET
```

Einige Programme erwarten auch, daß die Leseroutine erst dann RETurned, wenn ein Byte eingelesen wurde. Das geht dann so:

```
READ      IN   A,(0EAh)
          AND  80h
          JR   Z,READ
          IN   A,(0EBh)
          RET
```

Die Baudrate kann man folgerndermaßen einstellen:

```
BAUD      LD   A,xx
          OUT  (0E9h),A
          RET
```

Die Initialisierung (Anzahl der Datenbits, Parity usw.) erfolgt zumindest beim CP/M beim Booten, sodaß diese kurze Routine, wenigstens dann, als ich es auf Hartmuts Rechner ausprobiert hatte, ausreichend ist. Der Wert von xx steht im Handbuch und ist je nach gewünschter Baudrate einzusetzen. Einige Werte sind FFh für 19200 Baud, EEh für 9600, CCh für 4800, AAh für 2400, 77h für 1200, 66h für 600 und 55h für 300 Baud.

Alexander Schmid

Hallo Clubkameraden, 28.10.90

Euer Diskothekar hat seit unserem letzten Treffen von Eurer Seite nichts mehr zu tun bekommen. Liegt es daran, daß ich keine Programme vorgestellt habe, oder unsere Zeitschrift nur noch sehr selten erscheint, oder haben alle ihren Z80 Rechner verkauft und machen jetzt in MS-DOS oder TOS? Zu meiner Entschuldigung möchte ich vorbringen, daß ich seit Juli stolzer Vater bin und vor kurzem (vielleicht auch schon länger??) habe ich mir, aus geschäftlichen Gründen auch so eine Maschine zugelegt. Das heißt aber nicht, daß ich nichts mehr mit NEW_DOS zu tun haben will. Schon von meinem Brötchengeber her muß ich, was Betriebssysteme angeht, sehr flexibel sein. Da ich MS-DOS, VMS (auf unserem Großrechner), RT11 und RSX (auf unseren Prozessrechnern) fahren darf.

Ich will Euch heute zeigen wie man einen sauberen Bildschirmaufbau (ansprechende Masken) ohne sehr viel Aufwand in verschiedenen Programmiersprachen realisieren kann. Als erstes das Prinzip. Wie Ihr sicher alle wißt läßt sich der Bildschirm in Zeilen und Spalten unterteilen. Ihr könnt sogar jede einzelne Position lesen und schreiben, was bei meinem Programm aber nicht nötig ist. Die Sache läßt sich in zwei Abschnitte einteilen.
Man nehme irgend einen Editor und gestalte sich sein Menü. Einzige Anforderung, Ihr müsst für jede Zeile ein Zeilenendezeichen (CR) stehen haben. Wie Ihr seht alles ganz einfach. Es sind sogar Sonderzeichen erlaubt die man sonst nur über CHR$() auf den Monitor bekommt.
Jetzt kommt der Trick, wie vorher schon angedeutet, geht einfach davon aus der Monitor sehe aus wie ein Vektor (Array,Feld) oder wie man es sonst noch nennen will. Jede Zeile ist ein Element dieses Vektors und die Dimension hängt nur von der Zeilenzahl ab die Ihr verwendet. Auf jeden Fall dimensioniert man ein Feld DIM MASKE$(16) für 16 Zeilen. Dieses Feld ist bei mir z.B. maximal 16 * 64 = 924 Byte groß. 16 Zeilen mit maximal 64 Zeichen pro Zeile und jedes Zeichen braucht 1 Byte Platz. Wenn man jetzt viele Masken in einem Programm hat kann es schon zu Speicherknappheit kommen, wesshalb es zwei Wege gibt. Entweder schnelles Maskenwechseln und viel Speicherplatzverbrauch oder wenig Speicherplatzverbrauch und lange Wartezeiten bis von der Diskette die Maske nachgeladen ist.

Die Beschreibung für eine Maske im schnellen Verfahren.

Sprung ins Unterprogramm fürs Maskenladen. Hier öffnet man den Maskenfile als sequentielle Datei. Nun folgt eine FOR NEXT - Schleife die den File Zeile für Zeile in unseren Vektor MASKE$(I) ladet, I ist der Schleifenzähler. Wie oft die Schleife durchlaufen wird hängt von der Anzahl der Zeilen ab. So jetzt sind die Daten (die Maske) im Speicher und wenn Ihr sie braucht gebt sie einfach mit PRINT und einer FOR NEXT - Schleife aus, aber vergesst nicht beim letzten PRINT den Zeilenvorschub zu unterbinden sonst ist die Maske oben mit der ersten Zeile schon wieder rausgerutscht. Solltet Ihr mehere Masken brauchen, dann müsst Ihr das obige für jede Maske einmal durchführen.
Ist der Speicher knapp so kann man das Unterprogramm immer dann aufrufen, wenn die Maske gebraucht wird. Wenn viel Platz vorhanden ist ladet man die Maske(n) am Anfang in den Speicher und hat sie anschließend alle jederzeit zur Verfügung. Man muß sie dann nur noch ausgeben.

Im Folgenden zeige ich Euch wie so etwas in BASIC und FORTRAN aussieht. Beim BASIC verwende ich die Befehle von GENIE-BASIC beim FORTRAN die von MS-FORTRAN unter MS-DOS. Übrigens vielleicht hat einer Unserer Assemblerspezialisten Lust die Ausgabe auf den Bildschirm in Assembler zu schreiben, dann läuft alles noch schneller.

Datei einlesen:

```
10 DIM MASKE$(16)
20 OPEN"I",1,"MASKE1.DAT:1"
30 FOR I=1 TO 16
40    LINEINPUT#1,MASKE$(I)
50 NEXT I
60 CLOSE 1
```

Maske auf den Bildschirm ausgeben:

```
1000 FOR I = 1 TO 15
1010    PRINT MASKE$(I)
1020 NEXT I
1030 PRINT MASKE(16);          vergesst das Semikolon nicht!!!!!
```

Die obigen Routinen könnt Ihr zum Beispiel im Editorprogramm EDITOR1/BAS auf Diskette 2 verwenden das von mir kommentiert wurde und jetzt EDITOR/BAS heißt.

Nun das ganze in MS-FORTRAN

Als erstes die notwendige Definitionen

```
      CHARACTER MASKE(24)*80     das Feld MASKE mit 24 Elementen a 80 Zeichen
                                 definieren
```

Den File einlesen

```
      OPEN(10,FILE='MASKE.001') Fiele öffnen
      REWIND 10                 Zurückspulen, den Zeilenzähler natürlich
      READ(10,'(A)')MASKE       den ganzen File in den Vektor
      CLOSE(10)                 den File schließen nicht vergessen
```

Maske auf den Bildschirm

```
      WRITE(*,'(24A\)')MASKE    den Vektor auf den Bildschirm ausgeben
```

Cursor positionieren und numerische Variable I einlesen

```
      CALL CURPOS (1,2)         Cursor positionieren auf Zeile 1, Spalte 2
      READ(*,'(I1)')I           I einlesen
```

Das Ganze funktioniert mit dem Treiber Namens ANSI.SYS dem eine entsprechende Steuersequenz geschickt wird.

```
      SUBROUTINE CURPOS (Z,SP)
******* AUFRUF : CALL POS (1,10)
*******     POSITIONIEREN AUF DEM BILDSCHIRM Z= ZEILE SP = SPALTE **********
*******     ERSTES ZEICHEN IM HAUPTPROGRAMM IST KEIN STEUERZEICHEN MEHR *****
*******     Z = 1-25              SP = 1-80
      INTEGER Z,SP
      WRITE (*,'(1X,A,I2.2,A,I2.2,A)') ' [',Z,';',SP,'H'
      END
```

Wie Ihr seht ist es ganz einfach einen ansprechenden Bildschirmaufbau zu machen ohne, daß man sich mit PRINT und einer Menge Blanks herumschlägt.

WINDOW.NDR
eine Umsetzung von WINDOW.BIB
aus Chip Spezial Turbo-Pascal Nr. 3
an die speziellen Moeglichkeiten von FlomonCC

Mitte 1989 habe ich die allgemeine Version von WINDOW.BIB an den NKC angepasst. Kurz darauf lief dann auch FlomonCC auf meinem Rechner. Es erwachte fast sofort der Wunsch, die Fenstertechnik auf Monitorebene zu nutzen. Da in der Reihe Turbo-Spezial von Chip viele gute Programme die in Heft 3 veroeffentlichte Bibliothek WINDOW.BIB verwenden, sollte die neue Bibliothek natuerlich kompatibel dazu sein. Lediglich die kostbare TPA muss entlastet werden. (WINDOW.BIB speichert alle Ausgaben in ein Fenster in der TPA. Der Speicher wird da recht schnell knapp.) Das Ergebnis ist ein kurzes Programm in Turbo-Pascal. Dieses Programm nutzt nicht alle Moeglichkeiten von FlomonCC. Hier ist (in Grenzen) noch Raum fuer eigene Kreativitaet. Die vorhandenen Prozeduren sollten jedoch nicht in ihrer Funktion veraendert werden, um die Kompatibilitaet zu erhalten.

Allgemeine Hinweise

WINDOW.BIB in der allgemeinen Version erwartet verschieden Parameter (z. B. Zeichen fuer Rand, Anzahl Zeilen etc). Diese Parameter waren in der Datei WINDOW.PAR gespeichert. Diese wird, wie auch WINDOW.BIB, vom Quellprogramm mit der Include-Anweisung (z. B. {$I WINDOW.BIB}) geladen. Die von Chip veroeffentlichten Programme erwarten eine Datei mit diesem Namen. Die Parameter sind fuer WINDOW.NDR ohne Bedeutung. Die Datei kann also leer sein. Natuerlich kann auch die entsprechen Include-Anweisung im Quelltext geloescht werden. Da dies meist vergessen wird, halte ich eine leere Datei mit dem Namen WINDOW.PAR auf meiner Bibliotheks-Diskette bereit.

WINDOW.NDR arbeitet nach meinen bisherigen Beobachtungen mit allen bisher von Chip veroeffentlichten Programmen. Auch die Bibliotheken SELECT.BIB, CHOOSE.BIB etc. koennen benutzt werden. Inwieweit die Maus funktioniert, konnte ich mangels Maus bisher nicht testen.

Die einzelnen Prozeduren

InitWindows
Parameter: keine
Aufruf nicht erforderlich. Prozedur wurde nur der kompatibilitaet halber eingefuegt. Prozedur wird beim ersten Aufruf von OpenWindow mit aufgerufen, wenn 'Initialisiert' nicht TRUE ist.

ExitWindows
Parameter: keine
Alle Fenster werden geschlossen. Initialisiert wird auf FALSE gesetzt.

OpenWindow
Parameter: x1,y1,x2,y2
Oeffnet ein Fenster mit den Koordinaten x1,y1 = Ecke links oben und x2,y2 = Ecke rechts unten.

CloseWindow
Parameter: keine
Schliesst das letzte Fenster.

SelWindow
Parameter: Nr
Das Fenster mit der ID 'Nr' wird zum aktuellen Fenster. Alle Ausgaben erfolgen jetzt in diesem Fenster. Das Fenster wird, wenn erforderlich, in den Vordergrund geholt. In der allgemeinen Bibliothek bezieht sich diese Prozedur nur auf sich ueberlappende Fenster. (Wer kompatible Programme schreiben will, sollte das beachten!)

ChangeWindow
Parameter: Nr
Funktion wie SelWindow. In der allgemeinen Version bezog sich diese Prozedur nur auf nicht ueberlappende Fenster. Wer kompatible Programme schreiben will, die dann auch problemlos auf anderen Rechnern (nicht NDR) laufen, sollte diesen Unterschied streng beachten. Auf dem NKC funktionieren beide Prozeduren gleich. ChangeWindow ruft hier lediglich SelWindow auf.

Funktionen

WhereX
Parameter: keine
Rueckgabe: Integer
Ermittelt die Spalte, in der der Curser steht.

WhereY
Parameter: keine
Rueckgabe: Integer
Ermittelt die Zeile, in der der Curser steht.

Globale Variablen

Initialisiert
Typ: Boolean
Gibt an, ob Fenstertechnik initialisiert wurde. Fuer FlomonCC ist das nicht wichtig, aber in der allgemeinen Version geht ohne Initialisierung gar nichts.

Inverse
Typ: Boolean
Wenn Inverse = True, dann erfolgt die Ausgabe in inverser Darstellung.

ScreenPtr
Typ: Byte
Gibt die ID des gerade aktiven Fensters an.

MaxScreen
Typ: Byte
Enthaelt die Anzahl der geoeffneten Fenster.

Zum Schluss noch einige Bemerkungen

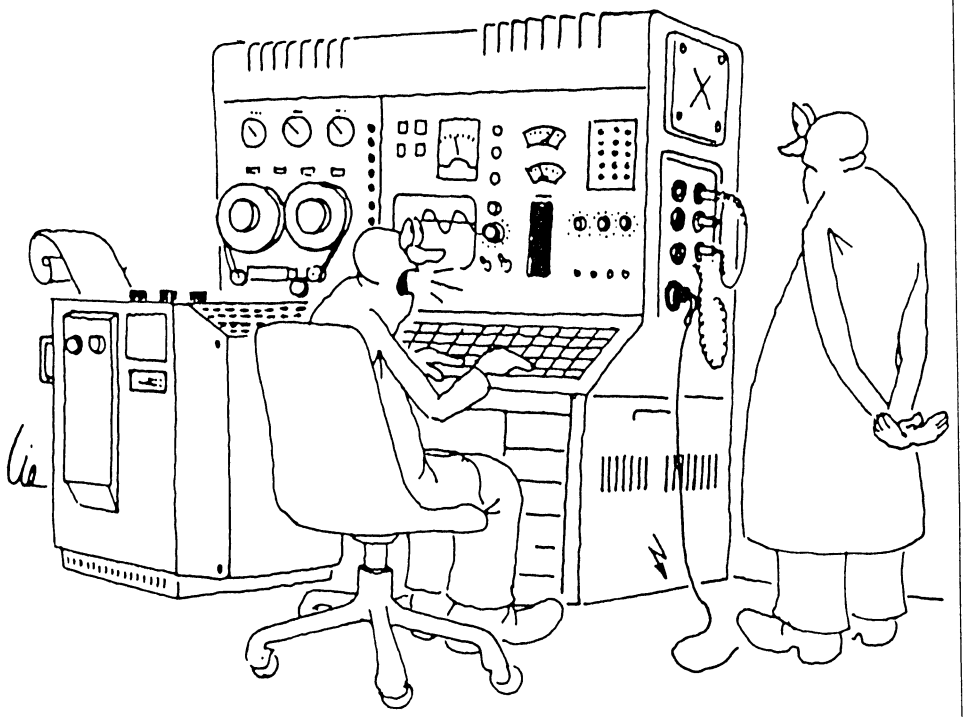
Ich moechte nochmals davon Abraten, die Bibliothek voll an die vorhanden Moeglichkeiten des FlomonCC anzupassen. Mit Sicherheit kann das Ganze einfacher und eleganter geloest werden. Die Verwendung der vorliegenden Bibliothek hat aber den grossen Vorteil, dass es eine auf jedem Rechner arbeitende Version gibt. Programme, die diese Bibliothek verwenden, laufen also auf jedem

Rechner. Sie muessen lediglich unter der Verwendung von WINDOW.BIB und WINDOW.PAR neu kompiliert werden. Es ist lediglich darauf zu achten, dass in der TPA ausreichend Platz zum speichern der Fenster ist. Da in der Regel aber nur sehr wenige Fenster gleichzeitig geoeffent sind,ist das meistens kein Problem.

Wer sich das Eintippen sparen moechte, kann mir auch eine formatierte Diskette (5 1/4 oder 3 1/2) schicken. Aber bitte ausreichend Rueckporto beifuegen. Ueber Anregungen und Beispielprogramme wuerde ich mich natuerlich auch freuen.

Ich habe in diesem Artikel uebrigen bewusst auf Umlaute etc. verzichtet, da mir fuer BRADFORD noch keine Deutschen Zeichensaetze zur Verfuegung stehen. Da ich aber die ausserordentlichen Faehigkeiten dieses Druckprogramms nutzen wollte, um das ganze lesbarer zu gestalten, musste der normale ASCII-Zeichensatz genuegen. Naeheres zu diesem (und weiteren interessanten) Programmen soll in weiteren Artikeln folgen.

Ausgabe
»Die Computer-Diagnose ergab, daß wir uns gar keinen Computer mehr leisten können.«




```
*************************************************************
* Bibliotheks-Modul: WINDOW.NDR*
**
* Spezielle Anpassung von WINDOW.BIB aus Chip Spezial Turbo-Pascal
* an den NDR-Klein-Computer mit FlomonCG und GDP64HS. Ob das ganze
* auch mit der alten GDP64 funktioniert, wurde bisher nicht getest
* Die Urversions von Chip benoetigt noch die Include-Datei WINDOW.
* Diese Datei kann hier leer sein oder fehlen. Um Aenderungen an*
* bestehenden Programmen so gering wie moeglich zu halten, wird *
* empfohlen, diese Datei zu erstellen.*
**
* Das Modul liefert die folgenden Bezeichner und Routinen:*
**
* p InitWindowsursprueglich zur Initialisierung gedacht*
*bei FlomonCG leere Procedure*
* p ExitWindowwie InitWindows*
* p OpenWindow (x1,y1,x2,y2)*
* oeffnet ein Fenster von (x1,y1) = LinksOben*
*bis (x2,y2) = RechtsUnten*
* p CloseWindowschliesst das letzte Fenster*
* p SelWindow (NR)Holt das Fenster (Nr) nach von und macht es*
*aktiv (in der Urversion fuer ueberlappende *
*Fenster gedacht)*
* p ChangeWindow (Nr) macht Fenster (Nr) aktiv; Aktion wird von*
*SelWindow erledigt. Ist erforderlich, um alte*
*Programme nicht aendern zu muessen. In der*
*Urversion war die Procedure fuer nicht*
*ueberlappende Fenster zustaendig. Fuer *
*FlomonCG gibt es da keinen Unterschied.*
* v inverse : boolean gibt an, ob gerade LowVideo gilt.*
* v ScreenPtr : bytegibt die Nr. des gerade aktiven Fensters *
* v initialisiert : boolean*
*gibt an, ob InitWindows aufgerufen wurde. Hat*
*eigentlich keine Bedeutung mehr. Da aber alte*
*Programme dies evtl. abfragen, wird das Flag*
*entsprechend gesetzt.*
* v MaxScreen : byteist die Anzahl der geoeffneten Fenster*
* f WhereX : integerCurser-Spalte*
* f WhereY : integerCurser-Zeile*
* *
* Neu definiert wurden noch die Proceduren NormVideo und HighVideo
* Bei korrekter Installation von Turbo Pascal waere auch dies nich
* erforderlicht. Aber wer hat schon immer alles korrekt installier
**
* Das urspruengliche Modul geht auf eine Veroeffentlichung von *
* Chip (Turbo-Pascal Spezial Nr. 3) zurueck. Es wurden jedoch *
* lediglich die Bezeichner der Proceduren und Variablen uebernomme
* um mit diesem Modul kompatibel zu bereits geschriebener Software
* zu bleiben. Das Copyright dieses Moduls liegt bei mir. Die *
* Verwendung und Verbreitung wird zu den ueblichen Public-Domain-*
* Bedingungen gestattet. Die gewerbliche Nutzung (was immer das au
* sein mag) ist nur mit meiner ausdruecklichen Genehmigung gestatt
* Auch eine Veroeffentlichung ist vorher mit mir abzustimmen (ich
* erwarte in der Regel die Zusendung eines Belegexemplares).*
**
* Dies ist die Version 1.0 von 11/89*
**
* Fehlerhinweise, Verbesserungen und Fragen bitte an:*
**
```

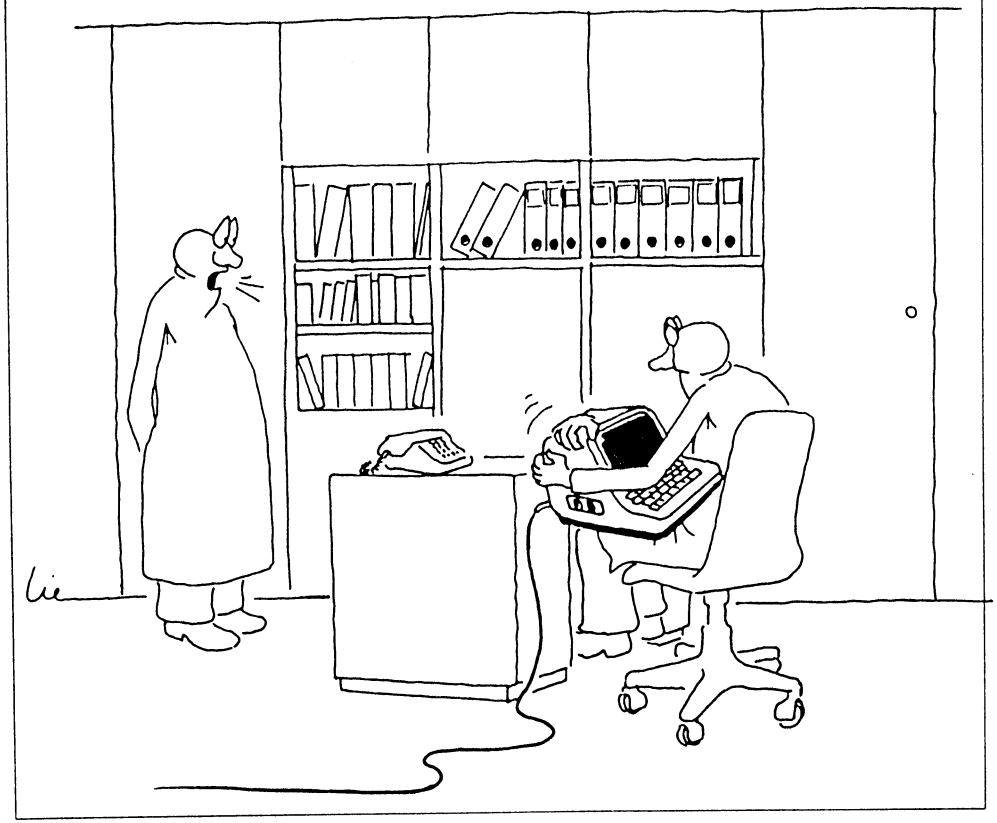
```
  * Claus Littmann, Plockhorst, Zum Spring 15, 3155 Edemissen*
  * Tel. 05372-7796 (bitte nicht nach 21.00!!)*
  ***************************************************************


Const   ScreenPtr     : Byte = 0;   { das momentan aktive Fenster }
        MaxScreen     : Byte = 0;   { Anzahl der geoeffneten Fenster
        Initialisiert : Boolean = False; { wg. Kompatibilitaet }
        Inverse       : Boolean = False; { Flag wg. Inverser Darstel


Procedure InitWindows;        { nur wg. kompatibilitaet noetig }

Begin   Initialisiert := True;  End;

Procedure ExitWindows;  {alle Fenster schliessen, Fenster 0 dann akt

Begin
   WriteIn (#27,'$@');{ Fenster normieren = alle Fenster schliessen,
  Fenster 0 mit Standardgroesse oeffnen }
   ScreenPtr := 0; MaxScreen := 0; { alle Pointer auf 0 setzen)
   Initialisiert := False;   { der Komptibilitaet wegen }
End;

Function WhereX : Integer; { Wo steht der Curser }

Var St1  : String[2];

Begin
   Write (#27,'?'); Read (Kbd,St1); { Curser mit Flomon abfragen }
   WhereX := Ord(Copy(St1,2,1))-32; { Wert fuer x-Koordinate normier
End;

Function WhereY : Integer;      { Wie WhereX, nur fuer Y-Koordinate }

Var St1  : String[2];

Begin
   Write (#27,'?'); Read (Kbd,St1);
   WhereY := Ord(Copy(St1,1,1))-32;
End;

Procedure NormVideo;     { Darstellung mit normalen Zeichen }

Begin
   Write (#27,'('); { Flomon-Befehl Invers aus }
   Inverse := False; { Flag setzen, damit man weiss, was los ist }
End;

Procedure LowVideo;{ LowVideo hier als Invers interpretiert }

Begin
   Write (#27,')');{ Flomon-Befehl Invers an }
   Inverse := True;{ Und wieder das Flag setzen }
End;

Procedure HighVideo;{ = NormVideo, da wir sowas nicht haben }

Begin NormVideo End;
```



```
Procedure SelWindow (x : Byte);{ Ein Fenster anwaehlen }

{  Das Fenster x wird in den Vordergrund gebracht; alle  A
erfolgen  jetzt in diesem Fenster. Fenster muss  definiert
Ausserdem muss das Flag 'initialisiert' = TRUE sein. Dies
FlomonCG  zwar  nicht  erforderlicht,  aber  in  der  allg
Version     dieser     Bibliothek     sehr     wichtig.     da     in
Initialisierungsprocedure  die Ausgaberoutiene  von  Turbo
auf  eine eigene Routiene umgebogen wird. Um alle Programm
diese Bibliothek verwenden, ohne Aenderungen weiter verwen
koennen,  wurde  dieses  hier  eigentlich  nicht  erforderlich
gesetzt. }

Begin
   If Not initialisiert Then Halt;
   If X <= MaxScreen Then
      Begin
         Write (#27,'$S',x);{ Fenster in den Vordergrund }
         WriteIn;
         GotoXY(0,0);{ Curser setzen }
         ScreenPtr := x;{ Fenster ID aktualisieren }
      End;
End;

Procedure ChangeWindow (x : Byte);   { wg. kompatibilitaet }

{   Diese  Procedure  war  urspruenglich  erforderlich,  d
allgemeine Version einen Unterschied zwischen hintereinand
nebeneinander  liegenden  Fenstern  machte.  Um  Aenderung
bestehenden  Programmen unoetig zu machen, wurde diese  Pr
erhalten. Die Arbeit macht aber SelWindow. }

Begin
   SelWindow(x);
End;

Procedure OpenWindow (a1,b1,a2,b2 : Byte);

{ Hier wird ein Fenster geoeffnet. Die Koordinaten sind :
  a1,b1 = links oben bis a2,b2 = rechts unten. Diese Angab
  werden auf das Flomon-Format umgerechnet. }

Var  Zeilen,Spalten,
     Offx,Offy       : Char;
     ID              : Integer;

Begin
   If Not initialisiert Then InitWindows;
   Spalten := Chr(32+(a2-a1)); Zeilen := Chr((b2-b1)+32);
   Offx   := Chr(a1+32);      Offy    := Chr(b1+32);
   ScreenPtr := Succ(ScreenPtr);  { enthaelt die ID des Fenstern }
   ID := ScreenPtr;
   Write (#27,'$O',ID,Zeilen,Spalten,Offy,Offx); { Fenster oeffnen }
   WriteIn (#27,'$E');      { und sichtbar machen }
   MaxScreen := ScreenPtr  { Anzahl der geoeffneten Fenster }
End;

Procedure CloseWindow;

{ Mit dieser Procedure wird das letzte geoeffnete Fenster
  geschlossen. }
```

```
Begin
     If (ScreenPtr=0) Or Not initialisiert Then Halt;
                                        { kein Fenster offen }
     Writeln (#27,'$C');  { letztes Fenster schliessen }
     ScreenPtr := Pred(ScreenPtr); { Zeiger berichtigen }
     MaxScreen := ScreenPtr;  { Anzahl der Fenster berichtigen }
End;
```

Büro
»Sie müssen einfach vergessen, Meier, daß Sie vor der Rationalisierung ein herzliches Verhältnis zu einer Sekretärin hatten.«



*** Umlaute in *LeScript* 1.65 ***

Wer von Euch mit *LeScript* (Vers. 1.65) seine Texte schreibt, wird das Problem sicher kennen: Das kleine "ö" (ö!) läßt sich auf dem Bildschirm nicht darstellen, da es als Trennungszeichen zwischen den einzelnen Buchstaben im Breitschriftmodus (CLEAR-X) verwendet wird. Überall dort, wo *LeScript* im Text ein "ö" vorfindet, schaltet der Druckertreiber auf gedehnte Schrift. Dies hat auch den Nachteil, daß der Text, der in Breitschrift geschrieben wurde, fast nicht mehr zu erkennen ist:

öBöröeöiötösöcöhöröiöföt ö(öCöTöRöLö-öXö)

Durch die unten aufgeführten Änderungen wird die Tastatur ein wenig anders belegt, so, daß nun auch Umlaute erreichbar sind. Außerdem wird das "ö" im CLEAR-X-Modus durch ein Grafikzeichen ersetzt. Hierfür habe ich den Grafikblock B0h (176d) gewählt, der - meiner Meinung nach - den Text am Besten in Breitschrift simuliert. Die Einstellung läßt sich natürlich beliebig ändern, bis auf einige Ausnahmen:

- 7Fh (127d) = "hartes Leerzeichen"
- 83h (131d) = Blockmarkierung Anfang/Ende
- 88h (136d) = Markierung Absatzende
- 8Ah (138d) = Einleitung eines Druckersteuerbefehls
- 8Ch (140d) = Satzendzeichen

Die oben genannten Codes also vermeiden, damit sie von *LeScript* nicht in die entsprechenden Kommandos umgesetzt werden.

Die Änderungen sind, z.B. mit SUPERZAP o. ä. , im Hauptmodul von *LeScript* vorzunehmen (LESCRIPT/CIM bzw. LESCRIPT/CMD).

1. Patch - Umbelegung der Tastatur für die Umlaute:

Sektor:	Byte:	Alt:	Neu:	Zeichen:
15	C2	1B	5B	"Ä"
15	C4	1C	5C	"ö"
15	C6	1D	5D	"Ü"
15	C8	1E	5E	"^"
15	CA	1F	5F	"_"
15	CC	5E	7B	"ä"
15	CE	7E	7C	"ö"
15	D0	5B	7D	"ü"
15	D2	5D	7E	"ß"
15	D4	00	1B	ESCAPE
15	D8	7B	00	(Frei)
15	DA	5F	00	(Frei)
15	DC	7D	00	(Frei)
15	DE	5C	00	(Frei)

2. Patch - Grafikblock statt "ö" bei CLEAR-X:

Sektor:	Byte:	Alt:	Neu:	Zeichen:
32	F7	7C	B0	Grafik
69	FA	7C	B0	Grafik
70	06	7C	B0	Grafik

Die Tastatur ist nun folgendermaßen belegt:

```
<SHIFT>-<CLEAR> "1"  =  Ä
<SHIFT>-<CLEAR> "2"  =  ö
<SHIFT>-<CLEAR> "3"  =  ü
<SHIFT>-<CLEAR> "4"  =  ^
<SHIFT>-<CLEAR> "5"  =  _
<SHIFT>-<CLEAR> "6"  =  ä
<SHIFT>-<CLEAR> "7"  =  ö
<SHIFT>-<CLEAR> "8"  =  ü
<SHIFT>-<CLEAR> "9"  =  ß
<SHIFT>-<CLEAR> "*"  =  ESCAPE-Code (erscheint als "A")
```

Die Tasten <CLEAR>-<SHIFT>-",", <CLEAR>-<SHIFT>-"-", <CLEAR>-<SHIFT>-"." und <CLEAR>-<SHIFT>-"/" lassen sich frei belegen. (Sektor 15, Bytes D8, DA, DC und DE)

In den Sektoren 14 und 15 läßt sich mit einem Diskettenеditor die gesamte Tastaturbelegung von *LeScript* ändern.
Einer "DIN"-ähnlichen Belegung steht also nichts im Wege ...

Deutsche Version von *LeScript* (mit DIN-Belegung) kann gegen eine formatierte Diskette und Rückporto bei mir angefordert werden.

Jens Günther

(PS: Suche außerdem neuere Version von *LeScript* (1.7 ? oder so)

Download noch einfacher

Wenn man seinem Drucker, sofern er softwaremäßig Zeichensätze laden kann, neue Zeichen beibringen will, kann einen das manchmal den letzten Nerv kosten. Bei 9-Nadel-Druckern geht es ja noch, aber bei 24ern wird's lästig. Eine Matrix mit 24 Zeilen und 32 Spalten auszuzählen macht nicht sehr viel Spaß und ist noch dazu einigermaßen fehlerträchtig. In einem CHIP-Special war zwar mal ein bildschirmorientierter Editor, aber der arbeitet stur mit 8x11 Punkten, die eben nur für einen 9-Nadler reichen. Die Lösung: in der c't 12/90 ist ein Programm, das mit beliebigen Formaten und also auch mit beliebigen Druckern arbeiten kann. Das Geheimnis liegt darin, daß man die Matrix und die notwendigen Steuerzeichen in ein Textfile schreibt, das dann vom Programm interpretiert wird. Ein Beispiel für so ein File könnte folgendermaßen aussehen:

```
File für DOWNLOAD.PAS aus der c't 12/90
Download-Zeichen für HP DeskJet mit FX-80-Emulation

#27#58#00#00#00                 ROM ins RAM kopieren

#27#38#00#35#35#139             umdefiniertes Zeichen: #
.M 8 11 X                       Punkt-Befehl: 8 Zeilen, 11 Spalten
XXXXXXXXXXX
XXX..X..XXX
.....X.....
.XXXXXXXXX.
X.........X
X.........X
XXXXXXXXXXX
...........

#27#38#00#36#36#139             umdefiniertes Zeichen: $
.M 8 11 X
.....X.....
.....X.....
.XXXXXXXXX.
.....X.....
.....X.....
...........
.XXXXXXXXX.
...........

#27#37#01#00                    Zeichensatz im RAM aktivieren
```

Die Datei kann wirklich genauso aussehen, wie sie hier abgedruckt ist. Mit dem "Punktbefehl" M 8 11 X sagt man dem Programm, daß die Zeichenmatrix, die in der nächsten Zeile beginnt, 8 Zeilen und 11 Spalten hat und daß die gesetzten Punkte durch ein X repräsentiert werden. Alle anderen Zeichen werden ignoriert, daher kann man mit den Kommentaren sehr großzügig sein ! Die Zahlen hinter einem # werden als ASCII-Werte betrachtet und als ein Zeichen an den Drukker geschickt. So könnte man auch, ohne ein neues Zeichen zu defi-

nieren, einfach nur ein paar Steuerzeichen (Initialisierung, Fett, Doppeldruck...) an den Drucker schicken.
Zu beachten wäre noch, daß der "echte" FX-80 (und auch andere Nadeldrucker) aufgrund der langsamen Mechanik keine zwei direkt benachbarten Punkte drucken können. Die Matrix müßte dann so aussehen (bitte ausprobieren):

```
X.X.X.X.X.X
X.X..X..X.X
.....X.....
.X.X.X.X.X.
X.........X
X.........X
X.X.X.X.X.X
...........
```

Welche Steuerzeichen Euer Drucker im einzelnen braucht, müßt Ihr leider im Handbuch nachsehen. Auch in der c't ist das ganze noch etwas ausführlicher beschrieben. Wer nicht gerne tippt, kann das Programm gegen Rückporto von mir auf Floppy haben.

Noch eine Idee am Schluß. Thema: Artikelflut für's Info. Wenn die meisten sich nicht in der Lage sehen, großartige eigene Artikel über den ultimaten Patch im DOS oder sonstwas zu schreiben, wie wäre es denn mit Artikeln wie diesem ? Jeder wird doch wohl irgendeine Zeitung (vielleicht nicht gerade die BILD) lesen und den ein oder anderen Artikel finden, der von allgemeinem Nährwert ist. Pascal- und BASIC-Programme laufen, wenn sie nicht zu tief ins System einsteigen, auf fast jedem Rechner oder lassen sich zumindest relativ einfach anpassen.

```
(*****************************************************************)
(*                                                               *)
(* Dieses Programm wertet eine als Parameter übergebene Datei    *)
(* aus und schickt die darin enthaltenen Zeichen zum Drucker,    *)
(* um einen Sonderzeichensatz zu installieren.                   *)
(*                                                               *)
(* File          : DOWNLOAD.PAS                                  *)
(* Sprache       : TURBO PASCAL 3.0                              *)
(* Autor         : Heinz Hagemeyer                               *)
(*                                                               *)
(* (c) September 1990 Verlag Heinz Heise  Redaktion c't          *)
(*                                                               *)
(* c't 12/90, Seite 304                                          *)
(*                                                               *)
(*****************************************************************)

PROGRAM Download;

CONST Trennzeichen = '#';
      Blank        = ' ';
```



```
TYPE  Zeile         = STRING[255];
      Strg80        = STRING[80];

VAR   DateiName     : Strg80;
      Datei         : TEXT;
      Zeilennr      : INTEGER;
      Druckerfehler: BYTE;
      Code          : INTEGER;

PROCEDURE WriteXY(x,y:BYTE; was:Zeile);
BEGIN
  GotoXY(x,y);
  Write(was);
END;

PROCEDURE Error(t:Strg80; VAR x:TEXT; ZeilenNr:INTEGER);
VAR d:CHAR;
BEGIN
  WriteXY(36,2,'Fehler: '+#7);
  WriteXY((80-Length(t)) DIV 2,3,t);
  IF ZeilenNr>0 THEN
  BEGIN
    WriteXY(35,4,'Zeile: ');
    Write(ZeilenNr);
  END;
  (*$I-*)
  Close(x);
  (*$I+*)
  HALT;
END;

FUNCTION Umwandlung(VAR Datei:TEXT; DateiName:Strg80; x:Zeile;
                    ZeilenAnzahl:INTEGER):Zeile;
VAR zk:Zeile;
    p :INTEGER;

  FUNCTION ASCII(zk:Zeile):CHAR;
  VAR Fehler,Zahl:INTEGER;
  BEGIN
    Val(zk,Zahl,Fehler);
    IF (Fehler=0) AND (Zahl>=0) AND (Zahl<=255) THEN
      ASCII:=Chr(Zahl)
    ELSE
      Error('Zahl falsch in Datei '+DateiName,Datei,ZeilenAnzahl);
  END;

BEGIN
  p:=Pos(Blank,x);
  IF p>0 THEN
    x:=Copy(x,1,p-1);
  zk:='';
  REPEAT
    p:=Pos(Trennzeichen,x);
    IF p>1 THEN
```

```
      zk:=zk+ASCII(Copy(x,1,p-1))
    ELSE
      IF p=0 THEN
        zk:=zk+ASCII(x);
    x:=Copy(x,p+1,255);
  UNTIL (p=0) OR (x='');
  Umwandlung:=zk;
END;

PROCEDURE SendeZeichen(VAR Datei:TEXT; DateiName:Zeile;
                       ZeilenAnzahl:INTEGER);
CONST Ausdehnung=128;
      Punkt     ='.M';
VAR   i,Zeilen,Spalten:INTEGER;
      h,x             :Zeile;
      Eins            :CHAR;

  PROCEDURE Dimension(h:Zeile; VAR z,s:INTEGER; VAR Eins:CHAR);
  VAR i:BYTE;

    PROCEDURE GibZahl(VAR z:INTEGER);
    VAR Fehler:INTEGER;
        st    :STRING[3];
    BEGIN
      st:='';
      WHILE (h[i]<'0') OR (h[i]>'9') DO
        i:=i+1;
      WHILE (h[i]<='9') AND (h[i]>='0') DO
      BEGIN
        st:=st+h[i];
        i:=i+1;
      END;
      Val(st,z,Fehler);
      IF (z<1) OR (z>Ausdehnung) THEN
        Error('Dimension der Matrix falsch in '
              +DateiName,Datei,ZeilenAnzahl);
    END;

  BEGIN (* Dimension *)
    i:=1;
    h:=h+' ';
    GibZahl(z);
    GibZahl(s);
    WHILE h[i]=' ' DO
      i:=i+1;
    Eins:=h[i];
  END;

  PROCEDURE SendeMatrix(Nadeln,Spalten:BYTE; Eins:CHAR;
                        DateiName:Zeile);
  VAR i,ByteAnzahl,
      Anzahl        :BYTE;
      Zahl,Spalte   :BYTE;
      Matrix        :ARRAY [1..Ausdehnung] OF STRING[Ausdehnung];
  BEGIN
    ByteAnzahl:=Nadeln DIV 8;
    FOR i:=1 TO Nadeln DO
    BEGIN
      (*$I-*)
      ZeilenAnzahl:=ZeilenAnzahl+1;
      ReadLn(Datei,Matrix[i]);
      (*$I+*)
      IF IOResult<>0 THEN
        Error('Matrix in Datei '+DateiName+' fehlerhaft',
              Datei,ZeilenAnzahl);
    END;
    FOR Spalte:=1 TO Spalten DO
    BEGIN
      FOR Anzahl:=0 TO ByteAnzahl-1 DO
      BEGIN
        Zahl:=0;
        FOR i:=1 TO 8 DO
          IF Matrix[i+Anzahl*8][Spalte]=Eins THEN
            Zahl:=2*Zahl+1
          ELSE
            Zahl:=2*Zahl;
        (*$I-*)
        Write(Lst,Chr(Zahl));
        (*$I+*)
        IF IOResult<>0 THEN
          Error('Drucker nicht bereit',Datei,0);
      END;
    END;
  END;

  BEGIN (* SendeZeichen *)
    WHILE NOT EOF(Datei) DO
    BEGIN
      REPEAT
        ZeilenAnzahl:=ZeilenAnzahl+1;
        ReadLn(Datei,h)
      UNTIL (Pos(TrennZeichen,h)>0) OR EOF(Datei) OR
            (Pos(Punkt,h)=1);
      IF Pos(TrennZeichen,h)>0 THEN
        Write(Lst,Umwandlung(Datei,DateiName,h,ZeilenAnzahl))
      ELSE
        IF Pos(Punkt,h)=1 THEN
        BEGIN
          Dimension(h,Zeilen,Spalten,Eins);
          SendeMatrix(Zeilen,Spalten,Eins,DateiName);
        END;
    END;
  END;

BEGIN (* Main *)
  ClrScr;
  IF ParamCount=0 THEN
    Error('Falscher Aufruf, Dateiname fehlt',Datei,0)
```

```
    ELSE
    BEGIN
      DateiName:=ParamStr(1);
      Assign(Datei,DateiName);
      (*$I-*)
      Reset(Datei);
      (*$I+*)
      IF IOResult<>0 THEN
        Error(DateiName+' nicht gefunden',Datei,0);
      SendeZeichen(Datei,DateiName,ZeilenNr);
      Close(Datei);
    END;
END.
```

Alexander Schmid

Brainstorming

After a long, long Time ...

Von CP/M nach MSX

Vor ein paar Wochen unterrichtete mich mein lieber Freund Kajot Mühlenbein von einem Unglück, an das zu denken jedem wirklichen Z8Ø-Freak den Angstschweiß auf die Stirne treibt. Nachdem sein, von Helmut Bernhard mühsam bis auf CP/M-Fähigkeit aufgebohrtes, Model 1 seinen Geist aufgegeben hat (das kann ja schon mal passieren und ist halb so wild) kaufte er sich, noch dazu zu einem unverschämt günstigen Preis, einen MSDOS-kompatiblen PC (was die eigentliche Katastrophe darstellt)!

Spätestens jetzt, vielleicht aber schon nach dem Entziffern des letzten Wortes der Überschrift, haben die oben angesprochenen orthodoxen Z8Ø-Freaks diese Seite überblättert und sind wahrscheinlich gerade dabei, Kajot's und meine Adresse aus ihren Adressbüchern, Computern und Gehirnen zu tilgen! All diejenigen, die sich bis hierher durchgerungen haben und noch dazu ein wenig an dBase interessiert sind, kann ich beruhigen: das folgende kann man durchaus auch dann sinnvoll verwenden, wenn man nicht zu den abtrünnigen Überläufern ins MSDOS-Lager gehört!

Nachdem die oben schon beschriebene Katastrophe ihren irreversiblen Lauf genommen hatte, bat mich Kajot, wohl wissend, daß auch ich seit geraumer Zeit nicht mehr mit ungetrübtem Gewissen den Z8Ø-Gurus unter die Augen treten darf, um einen Gefallen. Er hatte sich unter CP/M, dBase II benutzend, eine mehrere hundert Records (=Datensätze) große Datenbank angelegt. Da er sich die Arbeit ersparen wollte, alle Datensätze nochmals unter MSDOS einzutippen (die Z8Ø'er hätten es ihm wohl gegönnt), richtete er die Anfrage an mich, ob man die Daten wohl zwischen den Systemen transferieren könne. Die Antwort auf diese Frage erinnert ein wenig an Radio Eriwan: "Im Prinzip ja, aber ...".

Bei der Übertragung von Daten zwischen CP/M- und MSDOS-Systemen treten zunächst einmal zwei grundsätzliche Probleme auf:

1. MSDOS-Systeme können mit CP/M-Disketten (ohne spezielle Programme) nichts anfangen (umgekehrt natürlich auch nicht!)

2. die Zeichensätze unter CP/M und MSDOS unterscheiden sich bei den Umlauten, beim 'ß' sowie bei manchen Sonderzeichen (z.B. §).

Zusätzlich besteht bei der Übertragung von dBase-Dateien ein Unterschied zwischen den von dBase II und dBase III angelegten Datendateien.

Das Problem der unterschiedlichen Zeichensätze läßt sich, sowohl unter CP/M als auch unter MSDOS, durch recht einfache Programme in den verschiedensten Sprachen lösen. Wer nicht selbst zum Interpreter oder Compiler greifen will, kann von mir einen entsprechenden Turbo PASCAL-Quelltext bekommen.

Das Problem der physikalischen Übertragung der Daten ist da schon schwieriger zu knacken. Prinzipiell gibt es vier Lösungsansätze:
1. ein Programm, mit dem der CP/M-Rechner MSDOS-Disketten bearbeiten kann;
2. ein Programm, mit dem der MSDOS-Rechner CP/M-Disketten bearbeiten kann;
3. die serielle Übertragung der Daten zwischen CP/M- und MSDOS-Rechner;
4. die parallele Übertragung der Daten zwischen CP/M- und MSDOS-Rechner.

Wer glücklicher Besitzer eines TRS 8Ø Model 4/4p ist, hat es gut. Für diesen Rechner gibt es von Montezuma Micro/JOB ein Programm zum Bearbeiten von MSDOS-Disketten (DBLCROSS). Ganz nebenbei ermöglicht DBLCROSS auch noch den Datentransfer mit dem Model III-TRSDOS 1.3 und dem Model 4-TRSDOS 6.x. Weiterhin bekommt man für fast alle New-/TRSDOS-Versionen das Programm HyperCross, welches sowohl den Umgang mit CP/M- als auch mit MSDOS-Disketten beherrscht. Wer keinen Tandy-Rechner sein Eigen nennt, dem kann eventuell mit den Programmen TRANSFER oder PC-READ geholfen werden. Beide Programme liegen im Turbo PASCAL-Quellcode vor und sind Public Domain.

Unter MSDOS gibt es eine recht große Anzahl von Programmen, mit denen man CP/M-Disketten lesen kann. Leider findet man nur selten eines, welches gerade das eigene, meist recht exotische CP/M-Format beherrscht. Noch seltener sind solche Programme, bei denen man die Liste der lesbaren CP/M-Formate ändern/erweitern kann. Zu diesem Thema will ich mich aber an dieser Stelle nicht weiter auslassen, eventuell gibt es dazu demnächst mal einen gesonderten Artikel.

Auch die serielle Übertragung von Daten ist eigentlich relativ einfach zu bewerkstelligen. Außer einer seriellen Schnittstelle, über welche aber die meisten CP/M-Rechner verfügen, braucht man dazu nur zwei Terminalprogramme (jeweils eines unter CP/M und eines unter MSDOS), welche die gleiche Sprache sprechen (soll heißen, welche das gleiche Übertragungsformat benutzen). Hier bieten sich KERMIT und MODEM7 an, beides Programme, die sowohl unter CP/M als auch unter MSDOS zur Verfügung stehen. Zusätzlich benötigt man noch ein Übertragungskabel, welches aber, zumindest bei nicht allzu hoch geschraubten Geschwindigkeitsansprüchen, mit drei Adern auskommt (TxD, RxT und Masse).

Wer weder eine serielle Schnittstelle hat, noch eines der oben genannten Programme zum Laufen bringt, der kann eventuell auf die zuletzt genannte Möglichkeit, die parallele Übertragung, zurückgreifen. Hierbei werden die Daten auf dem CP/M-Rechner ausgedruckt. Statt dem Drucker hängt aber ein PC mit entsprechendem Programm an der Schnittstelle. Informationen zu dieser, zugegeben etwas exotischen Art der Datenübertragung, findet man in der Zeitschrift c't 6/88 Seite 166 ff (PC Bausteine: Details über die parallele Schnittstelle).

Nun, nach mehr als einer Seite, sind wir endlich beim Kernthema dieses Artikels (Arnulf und Kajot haben bestimmt schon mit Rotstift die Randbemerkung "Thema verfehlt!" angebracht), der Inkompatibilität zwischen den Daten-(dbf-)dateien von dBase II und dBase III. Wie wir gleich sehen werden, läßt sich dieses Problem aber, im Verhältnis zu den oben genannten, relativ einfach lösen!

Zunächst könnte man, welch schrecklicher Gedanke, auch unter MSDOS mit dBase II arbeiten. Tatsächlich gibt es eine Version II des wohl bekanntesten Datenbankprogramms auch für MSDOS. Wer aber einmal, als eh nur sporadischer und damit ständig handbuchblätternder dBase II-Nutzer, dBase III gestartet und mit ASSIST (einem Menüsystem für Leute, die nicht gerne Handbücher wälzen) gearbeitet hat, der wird dieses Ansinnen als unannehmbar ablehnen!

Nach längerer Suche im Handbuch findet sich dann aber doch eine Möglichkeit, sowohl die Fingerkuppen als auch die Tastatur des MSDOS-Rechners zu schonen und die "alten" Daten zu übernehmen. Dazu geht man wie folgt vor:

1. unter dBase II die gewünschten Daten exportieren (Funktion COPY)
2. die exportierten Daten auf MSDOS übertragen (siehe oben)
3. unter dBase III die Daten wieder importieren (Funktion APPEND).

Dabei ist zu beachten, daß die Struktur der neu angelegten dBase III-Datenbank der exportierten dBase II-Daten entsprechen muß. Um die Vorgehensweise deutlicher zu machen, folgt hier als Beispiel die Übertragung bestimmter Felder aller Datensätze der Mitgliederverwaltung des CLUB 80.

Als Quelle wird die Datei PDATEN.DBF benutzt. Wie schon erwähnt, gehört sie zur Mitgliederverwaltung des CLUB 80, geschrieben von unserem ehemaligen Mitglied Harald Mand in dBase II auf einem Model 4p unter CP/M 2.2. Nachdem dBase gestartet und die Datumsabfrage beantwortet wurde, erhält man den obligatorischen Punkt als Prompt (auch Eingabeaufforderung genannt). Nun muß zunächst die Datenbankdatei geöffnet werden. Dies geschieht mit dem Befehl:

```
. USE PDATEN.DBF
```


Danach muß man sich unbedingt über die Struktur der Datenbank im klaren werden. Dazu kann man die Befehle DISPLAY STRUCTURE oder LIST STRUCTURE benutzen. Die Antwort von dBase II sieht wie folgt aus:

```
STRUCTURE FOR FILE:  A:PDATEN  .DBF
NUMBER OF RECORDS:   00074
DATE OF LAST UPDATE: 29/04/90
PRIMARY USE DATABASE
FLD       NAME       TYPE WIDTH   DEC
001       NAME          C    020
002       VORNAME       C    020
003       STRASSE       C    025
004       LAND          C    003
005       PLZ           C    005
006       WOHNORT       C    025
007       TELP          C    020
008       TELG          C    020
009       GEBOREN       C    010
010       EINTRITT      C    010
011       BEITRAG       L    001
012       NUMMER        N    004
** TOTAL **              00164
```

Nun kann man sich überlegen, welche Daten man übernehmen will. Ich habe für dieses Beispiel die Felder NAME, VORNAME, STRASSE, PLZ, WOHNORT und TELP ausgewählt. Diese Felder können nun mit dem Befehl COPY in eine Datei kopiert werden. Dieser Befehl hat so viele mögliche Parameter, daß vor dem Beispiel erst einmal ein paar Erläuterungen angebracht sind. Der Befehlssyntax lautet:

```
COPY TO <Dateiname> [<Geltungsbereich>] [FIELD <Felderfolge>]
     [FOR <Ausdruck>] [WHILE <Ausdruck>]
     [SDF] [DELIMITED [WITH <Begrenzung>]
```

Hierbei bedeuten:

Dateiname	: Name der Datei, in die die Datensätze geschrieben werden sollen
Geltungsbereich	: ALL = alle Datensätze (Voreinstellung!) NEXT (n) = vom aktuellen Datensatz ab die nächsten n Sätze RECORD (n) = nur der Datensatz mit der Nummer n
FIELD <Felderfolge>	: die Felder, die übernommen werden sollen (wenn keine Felder angegeben werden, werden alle Felder übernommen)
FOR <Ausdruck>	: alle Sätze, die die logische Bedingung des <Ausdrucks> erfüllen
WHILE <Ausdruck>	: alle Sätze, die auf den aktuellen Datensatz folgen, bis die im <Ausdruck> angegebene logische Bedingung nicht mehr erfüllt ist.
SDF	: Anlegen einer Datei im Standard-ASCII-Format (als Dateityp wird, wenn nicht anders angegeben, .TXT eingesetzt!)
DELIMITED	: in der Zieldatei werden alle Felder durch Komma getrennt und die Felder vom Typ "Zeichenkette" in Hochkomma (') eingeschlossen.
WITH <Begrenzung>	: sollen die Zeichenreihen durch ein anderes Sonderzeichen eingeschlossen werden, so kann das gewünschte Zeichen hier angegeben werden. Handelt es sich dabei um ein Komma, werden nicht belegte Stellen am Ende von alphanumerischen Feldern bzw. am Anfang von numerischen Feldern entfernt. Außerdem wird der Wert alphanumerischer Felder nicht in Sonderzeichen gesetzt.

Das war jetzt eine ganze Menge recht trockene Information, hier nun zwei Beispiele. Mit dem Befehl

. COPY TO PDATEN FIELD NAME, VORNAME, STRASSE, PLZ, WOHNORT, TELP SDF

wird eine Standard-ASCII-Datei mit Namen PDATEN.TXT erzeugt. Die Datei enthält die Felder NAME, VORNAME, STRASSE, PLZ, WOHNORT, TELP aller in der aktuellen .DBF-Datei enthaltenen Datensätze. Die einzelnen Datensätze sehen wie folgt aus:

```
0        10        20        30        40        50        60        70        80        90        100       110
.........:.........:.........:.........:.........:.........:.........:.........:.........:.........:.........:.....
Ahlers          Armin           Rotdornstr. 13          2116 Hanstedt/Nindorf       04184   /7667
Bernhardt       Helmut          Hafenstr. 7             2305 Heikendorf             0431    /241987
```

Wie man sieht, wird jedes Feld linksbündig geschrieben und bis auf seine maximale Länge mit Leerstellen aufgefüllt. In diesem Beispiel sind keine numerischen Felder enthalten, dort wäre es aber genau anders herum!

Der Befehl

. COPY TO PDATEN.DEL FIELD NAME, VORNAME, STRASSE, PLZ, WOHNORT, TELP FOR PLZ > '5999' DELIMITED

erzeugt eine Standard-ASCII-Datei mit Namen PDATEN.DEL. Die Datei enthält die Felder NAME, VORNAME, STRASSE, PLZ, WOHNORT, TELP aller Clubmitglieder, deren Postleitzahl größer als 5999 ist. Die einzelnen Sätze sehen wie folgt aus:

```
0         10        20        30        40        50        60        70        80
.........:.........:.........:.........:.........:.........:.........:.........:..
'Betz','Heinrich','St.Wolfgangstr. 13','8551','Hausen','09191    /31698'
'Brandl','Hermann','Ciciliastr. 30','8439','Postbauer-Heng','09188    /493'
```

Wie nach der Lektüre der Beschreibung des COPY-Befehl nicht anders erwartet, sind die einzelnen Felder durch Komma getrennt und die Zeichenketten in Hochkomma eingeschlossen. Dem aufmerksamen Beobachter wird weiterhin auffallen, daß die Zeichensatzkonvertierung hier noch nicht durchgeführt ist.

Welchen Vorteil hat nun das DELIMITED- gegenüber dem SDF-Format? Zunächst einmal fällt auf, daß die DEL-Datei um einiges kleiner ist, als die TXT-Datei. Die Platzersparnis hängt allerdings vom Füllungsgrad der einzelnen Felder ab. Läßt man alle Daten der Mitgliederdatei PDATEN.DBF in eine SDF-Datei kopieren, hat diese einen Umfang von 10 Kilobyte. Eine DEL-Datei mit gleichem Inhalt belegt nur 6 kByte. Dieser Größenunterschied könnte (bei größeren Datenbanken) beim Übertragen über eine serielle Schnittstelle ein Grund sein, sich für das DELIMITED-Format zu entscheiden.

Als zweites Argument könnte dienen, daß es sowohl unter CP/M als auch unter MSDOS eine stattliche Anzahl von Programmen gibt, die zwar das DEL-, nicht aber das SDF-Format importieren können. Insbesondere die Klassiker WordStar und Mailmerge sowie einige Tabellenkalkulationsprogramme bieten diese Möglichkeit. Man sollte bei der Wahl eines mit WITH gewählten Trennzeichens übrigens beachten, daß dBase selbst nur DEL-Dateien wieder einwandfrei importieren kann, bei denen die Felder in Hochkomma oder Anführungsstrichen eingeschlossen sind.

Und gerade das wollen wir ja! Die exportierte SDF- oder DEL-Datei wird nun auf den MSDOS-Rechner übertragen und der Zeichensatz angepaßt. Fertig? Dann kann es jetzt mit dBase III weitergehen!

Nach dem Start von dBase III und dem Erscheinen des obligatorischen '.' als Prompt, fühlt man sich fast wieder auf einen CP/M-Rechner versetzt (ich weiß, das ist bei weitem nicht die einzige Gemeinsamkeit zwischen CP/M- und MSDOS-Rechner!) und kann daher ungehemmt draufloshacken.

Zunächst muß man mit dem Befehl

. CREATE PDNEU.DBF

eine neue Datenbank eröffnen und deren Struktur definiert werden. Dabei muß, sonst klappt der spätere Import der alten Daten nicht, die Struktur der unter dBase II exportierten Daten streng eingehalten werden.

	Feldname	Typ	Länge	Dez
1	NAME	Zeichen	20	
2	VORNAME	Zeichen	20	
3	STRASSE	Zeichen	25	
4	PLZ	Zeichen	5	
5	WOHNORT	Zeichen	25	
6	TELEFON	Zeichen	20	

Da die Eingabe der Feldnamen, -typen und -längen (sowie, falls erforderlich der Anzahl der Dezimalstellen) ebenfalls nur im Detail von der Vorgehensweise unter CP/M abweicht, dürfte auch dieser Schritt keine Probleme mit sich bringen. Aber selbst ohne dBase-Erfahrung ist diese Hürde recht leicht zu nehmen. Wenn alle erforderlichen Felder definiert sind, kann man die obligatorische Frage, ob gleich Daten eingegeben werden sollen, verneinen. Schließlich wollen wir ja Daten importieren!

Dazu brauchen wir den Befehl APPEND. Dieser hat erheblich weniger mögliche Parameter als COPY. Neben der Importdatei und deren Format (SDF oder DELIMITED WITH) kann man noch die Parameter FOR <Ausdruck> und WHILE <Ausdruck> angeben. Für unseren Fall reicht folgender Befehl:

. APPEND FROM PDATEN.TXT SDF

oder

. APPEND FROM PDATEN.DEL DELIMITED WITH '

falls man das platzsparendere DEL-Format gewählt hat.

Was, ihr glaubt nicht, daß das alles war? Doch, doch! Keine Hexerei, kein Woodou, keine Anwendung von Zaubersalz oder sonstigen Hilfsmitteln! Ehrlich!!!

Nachworte

I. für die notorischen Nörgler und Besserwisser:

1. Ich weiß, daß die Befehle COPY und APPEND noch weitere mögliche Parameter und Einsatzmöglichkeiten haben. Der Grund für die Unterschlagung dieser Möglichkeiten ist der, daß sie für den beschriebenen Zweck nicht von Interesse sind.

2. Schon vor Jahren hat mir Gerald Dreyer erzählt, daß es ein Programm mit Namen DCONVERT geben soll, welches nicht nur die Anpassung von dBase II-Daten-, sondern auch Programmdateien erlaubt. Dies wurde mir vor kurzem von Fritz Chwolka bestätigt. Da ich dieses Programm aber nicht hatte (was nicht ist, kann ja noch werden), ...

II. für die zufriedenen CP/M-User (ein paar soll es ja tatsächlich geben)

Ich wünsche euch, daß euer treuer CP/M-Rechnknecht mindestens so lange funktioniert wie mein Model 4p, welches auf meinem Schreibtisch, trotz MSDOSe, immer noch einen Stammplatz hat!

I'm back now!

H.O.

Die Interrupts im Griff

Insbesondere bei der Bastelei am Betriebssystem kommt es häufig vor, daß die Interrupts mit DI vereitelt werden müssen, weil sonst eine Katastrophe passieren würde. Es ist absolut kein Problem, sie anschließend wieder mit EI zuzulassen. Was aber, wenn der CALLer bereits mit disableten INTs auf dieses Programm zugriff? Er wird seine ehrenwerten Gründe dafür gehabt haben. Also den Befehl EI weglassen? Wenn aber nun der CALLer auf INTs angewiesen ist, weil z. B. die Software-Uhr weiterlaufen oder das Tastatur-Repeat aktiv bleiben soll?

Es gibt eine Möglichkeit, den Interrupt-Status schon beim Einsprung festzustellen, um ihn nach der Bearbeitung wieder zu restaurieren. Fragt mich nicht, wieso, ich weiß es auch nicht: Beim Auslesen des R- und des I-Registers mit LD A,R oder LD A,I wird der gegenwärtige Zustand des Interrupt-Flipflops 0 (IFF0) in das P/V-Flag geladen. Ist er 1, dann sind Interrupts zugelassen, ist er 0, dann nicht. Man kann also nach dem Lesen von R oder I mit einem bedingten Sprung (JP PO,nn, JP PE,nn, CALL PO,nn, CALL PE,nn) an eine Stelle verzweigen, wo EI oder DI neu befohlen wird, je nach altem INT-Status.

Weil es mir nun mal so in den Kram paßte, geht das hier vorgestellte Programm (Listing 1) einen anderen Weg: Um für möglichst viele Zwecke dienlich zu sein, lädt es je nach INT-Status lediglich den Akku mit dem Opcode für EI (FBh) oder DI (F8h) und kehrt zurück. Das ist schön kurz, verändert aber den Akku-Inhalt und alle Flags außer dem Carry-Bit. Man kann dem mit PUSH AF begegnen, wenn es erforderlich ist (anschließend POP AF nicht vergessen!).

Die Ladeadresse der Routine ist vollkommen egal. Bei mir lautet sie 3BB4h, weil ich da gerade noch Platz hatte. Dort ist sie auch nicht im Bereich des User-RAMs ab 5200h (unter G-DOS), so daß ORG-Konflikte nicht zu erwarten sind.

Listing 2 zeigt ein Programm, das sich dieser Routine bedient. Es blendet das HRG-RAM in den Speicher ein. Das ist bei meinem Genie IIIs ein hochexplosiver Betriebszustand; jeder Interrupt hätte jetzt einen bühnenreifen Absturz zur Folge. Zugriffe auf die Graphik kommen sowohl mit enableten als auch mit disableten INTs vor. Während des Zugriffs müssen sie auf jeden Fall disablet sein. Mit Hilfe der neuen Routine kann nun der alte Zustand, egal, wie er war, wiederhergestellt werden.

So weit die guten Nachrichten. Die schlechte: Es gibt auch ein Interrupt-Flipflop 1 (Rodnay Zaks numeriert die beiden mit 1 und 2; ich erlaube mir, nach altem Hacker-Brauch von Nr. 0 und 1 zu sprechen. Sorry, Rodnay-Honey!). Tritt ein Interrupt auf, dann wird der gegenwärtige INT-Status aus IFF0 nach IFF1 kopiert. Der Sinn ist, daß dann mit den Befehlen RETI und RETN der alte Zustand restauriert werden kann: IFF0 holt sich sein früheres Bit aus IFF1 zurück. Das hat Folgen:

Wenn während einer Interrupt-Bearbeitung die Routine *chckint* geCALLt wird, kann nur DI im Akku stehen, weil zumindest nicht maskierbare Interrupts (NMI) sich als erstes die Konkurrenz von weiteren INTs vom Halse halten. Das machen sie, indem sie IFF0, also das für die CPU gegenwärtig gültige INT-Enable-Flag, auf 0 setzen. IFF1 bleibt auf dem alten Wert. Wird nun mit der neuen Routine DI befohlen (EI kann hier nicht auftreten, s. o.), dann setzt dieser Befehl IFF0 *und* IFF1 auf 0. So kann während eines NMI-Service die Suppe versalzen werden, weil auf jeden Fall DI dabei herauskommt.

Aber Kopf hoch: Der geneigte User soll eben aufpassen, daß seine NMI-Routine oder ein Glied in der INT-Kette von *chckint* die Flossen läßt. Das ist überhaupt kein Problem. G-DOS ist bezüglich der Interrupt-Bearbeitung wasserdicht (auch CP/M, wie ich hoffe). Programme, die in diese Falle stolpern könnten, sind also immer auf dem Mist des geneigten Lesers gewachsen. Demnach darf diese Routine dort nicht aufgerufen werden, wo ein INT bedient wird. Das trifft z. B. auf die Erkennung der 3-Tasten-Befehle des G-DOS zu (JKL, 123, DFG) und auf Programme, die über einen CALL 4410h in die INT-Kette eingefügt werden sollen. Aber das macht eh' keiner.

Arnulf

```
                00001 ; Listing 1:    Den Interrupt-Status gemäß IFF0 feststellen
                00002 ;               und entsprechend EI oder DI in den Akku laden
                00003
3BB4            00004         ORG     3bb4h           ;oder wo auch immer
3BB4  ED5F      00005 chckint LD      A,R             ;IFF0 ins P/V-Flag kopieren
3BB6  3EF3      00006         LD      A,0f3h          ;Opcode DI
3BB8  E0        00007         RET     PO              ;falls IFF0 = 0
3BB9  3EFB      00008         LD      A,0fbh          ;sonst Opcode EI,
3BBB  C9        00009         RET                     ;falls IFF0 = 1
3BB4            00010         END     chckint

00000 Fehler

chckint 3BB4
```

```
                00001 ; Listing 2:    beliebiges Programm, das die Interrupts disablen muß
                00002 ;               und hinterher den alten INT-Status wieder restauriert
                00003
5200            00004         ORG     5200h
5200  CDB43B    00005 start   CALL    chckint         ;je nach INT-Status EI oder DI -) Akku
5203  321452    00006         LD      (eidi),A        ;Befehl dort patchen
5206  DBFA      00007         IN      A,(0fah)        ;Systemport, höchst sensibel!
5208  F5        00008         PUSH    AF              ;Zustand retten
5209  F640      00009         OR      40h             ;Graphik ab 8000h eingeblendet
520B  F3        00010         DI                      ;um Himmels Willen keine Interrupts!
520C  D3FA      00011         OUT     (0fah),A        ;Systemport neu schreiben
520E  2AFC80    00012         LD      HL,(80fch)      ;zwei Graphikbytes abholen
5211  F1        00013         POP     AF              ;alten Systemport
5212  D3FA      00014         OUT     (0fah),A        ;restaurieren
5214  00        00015 eidi    NOP                     ;hier landet EI oder DI
5215  C9        00016         RET
3BB4            00017 chckint EQU     3bb4h           ;Adresse des IFF0-Detektorprogramms
5200            00018         END     start

00000 Fehler

chckint 3BB4    eidi    5214    start   5200
```

ZCPR - nicht nur eine Erweiterung zum CP/M sondern ....

(Allgemeine Information und Beschreibung der TCAP)

Inzwischen wurde schon einiges in den letzten Clubinfos über ZCPR geschrieben, wobei ich unter anderem an die Texte von Rüdiger und Alexander denke.
Hier möchte ich mal allgemein erläutern, was dieses ominöse ZCPR eigendlich ist, und einige Informationen zur Terminal-installierung geben.

Geschichtliches :

Wann und wie CP/M entstand, kann man überall nachlesen und soll hier draußen bleiben. Das Schöne am CP/M ist die Struktur des Betriebssystemes, welche eine Anpassung an fast alle Z80 (8080,8085) Systeme ermöglicht.
Wie bekannt enthält das Betriebssystem einen hardwareunab-hängigen Teil , welcher im eigentlichen Sinne das CP/M darstellt, und einen Teil zur Anpassung an die entsprechende Hardware.

Der von DRI erhältliche Programmteil zum CP/M ist in 8080 geschrieben und beinhaltet keine Hardwareanpassung. Die Hardwareanpassung wird vom Distributor des Rechners erstellt oder selbst geschrieben (XBIOS,CBIOS oder ähnliches).
Leider ist das CP/M sehr sparsam mit Befehlen ausgestattet worden, und CP/M 2.x hat nur 5 interne Befehle zur Bearbeitung. Diese sind DIR, TYPE, ERASE, SAVE und RENAME.

Die Ausführung der Befehle ist Sache des Command Controll Prozessors, kurz CCP genannt. Er übernimmt die Tasteneingaben und führt die o.a. Befehle mit Hilfe des restlichen Betriebssystemes aus. Bei CP/M+ existiert der CCP als eigene Datei (CCP.COM), unter CP/M 2.x befindet sich der CCP im Speicher unterhalb dem BDOS (Basic Disk Operating System) und kann zwecks Änderung exakt lokalisiert werden.

Wie kann man aber den CCP ändern ?

Mit Einführung des Z80 bekam man eine Möglichkeit. Durch den umfangreicheren Befehlssatz des Z80 konnte man den CCP im Z80 Code schreiben und hatte noch Platz für Erweiterungen, da das Programm durch die Möglichkeiten des Z80 kürzer wurde. So wurde aus dem CCP der ZCPR (Z80 ControlPRozessor). Die vielen Zwischenstationen und andere gleichartige Programme (XCCP,CCP105..usw) werden kaum noch genutzt und nur der Vollständigkeit halber erwähnt.
Die heutige Version des ZCPR erweitert nicht nur die Befehle des CCP, sondern bildet eine neue Commandoumgebung und verbindet hierfür geschriebene Programme untereinander. Viele Funktionen wurden -ähnlich UNIX- eingebunden, wie z.B. die Beschreibung des Terminals (TCAP).

Nun aber mal ans Eingemachte. Im INFO wurde bis jetzt nur die Public Domain Version des ZCPR behandelt, die Version 33. Deren Installierung setzt aber doch genaue Kenntnisse des Systems voraus und es sollte das Programm MOVCPM vorhanden seien. Oft ist aber nur SYSCOPY, SYSGEN o.a. vorhanden, mit denen man nicht ein neues CP/M für andere Speichergröße erzeugen kann, sondern nur die Systemspuren kopiert. (Anm. Mit MOVCPM kann das CP/M entsprechend der Speichergröße generiert werden. MOVCPM beinhaltet ein Abbild des CP/M's).
Auch ist man bei einer 'Handinstallierung' auf eine Speichergröße festgelegt. Dies muß aber nicht sein, denn inzwischen bekommt man für CP/M2.2 und CP/M+ ein selbstinstallierendes ZCPR, welches allerdings COPRIGHTED und folglich nicht ganz um sonst ist.

Die ZCPR-Version für CP/M2.2 heißt 'NZCOM' , die für CP/M+ 'Z3PLUS'. Im weiteren Text schreibe ich aber für alle Versionen ZCPR , da ich mich auf alle beziehe und bei Unterschieden darauf hinweise.
Beide copyrighted ZCPR - Umgebungen sind in ca.5 Minuten installiert und bieten eine größere Flexibilität als die Pd-Version.

T E R M I N A L - B E S C H R E I B U N G

Eine Grundfunktion innerhalb des ZCPR ist der Installierung der entsprechenden Terminalbefehle zuzuordnen. Diese Installierung kann per Programm oder zu Fuß erfolgen.

Folgende Programme kann man zur Erstellung der Terminalbeschreibung, im weiteren Text TCAP (TerminalCAPability) genannt, nutzen:

TCMAKE.COMP	Erstellt interaktiv eine TCAP - Datei
TCSELECT.COM	Wählt aus einer Datendatei ein Terminal aus und erstellt eine TCAP - Datei.
TCVIEW.COM	Zeigt die Parameter des eingestellten Terminals auf und hilft so bei der Kontrolle eines 'zu Fuß' installierten Terminals.
TCCHECK.COM	Testet eine TCAP-Datei auf korrekten Aufbau und gibt Fehler an.
TCSRC.COMP	Erstellt einen kommentierten SOURCE-CODE (MYTERM.Z80) aus einem TCAP-File und dient zur korrekten Installierung eines TCAP für ZCPR33/34 da hier zum Original von Richard Conn einige Verbesserungen bzw. Erweiterungen innerhalb der TCAP gemacht wurden.
TCAP.COM	Beispielprogramm zur Nutzung des TCAP von Basic aus .

WAS soll diese Terminalbeschreibung ?

Programme, welche für ZCPR geschrieben werden, kennen die Existenz der TCAP und können so ohne gesonderte Installierung ablaufen. Hiermit können Programme direkte Cursorsteuerung nutzen und laufen auf jedem Rechner mit ZCPR. Neue Funktionen wie PULL-DOWN Menues oder Buisinessgraphic werden hiermit auf einfacher Weise möglich und dies ohne eine jeweilige Neuinstallierung.

Die Terminalbeschreibung ab März 1989:

Ab ZCPR-Version 3.3 wurde die Terminalbeschreibung um einige Funktionen erweitert. Vor allem ist nun die Möglichkeit der Grafikeinbindung gegeben und einige Beispielprogramme sind in der 'SAYGE-PD' hierfür enthalten

TCAP Definition

Die TCAP ist ein 128 Byte langes Feld und befindet sich direkt hinter der Umgebungbeschreibung (ENV). Hier möchte ich den Aufbau und die Struktur der TCAP anhand eines Beispieles aufzeigen.

Aufbau und Strucktur der Terminalbeschreibung (TCAP):

BYTE	Bedeutung
0	Terminalname , bestehend aus 15 alphanumerische Zeichen. das erste Zeichen darf kein 'SPACE' sein, da so ein uninstalliertes Terminal definiert wird.
15	Grundlegende Eigenschaft des Terminals.

Jedes der 8 Bit von BYTE 15 hat eine festgelegte Bedeutung, welche nachfolgend aufgezeigt werden.

BIT	Wert	Definition
0	- 1	= Revers oder helle Darstellung
0	- 0	= Dunkle oder in der Helligkeit reduzierte Darstellung
1	- 1	= AutoLineFeed beim Schreiben in der letzten Spalte (weiterschreiben in neuer Zeile)
1	- 0	= Kein Auto-LF
2	- 1	= Scrollen beim Schreiben in letzter Spalte und/oder Zeile.
2	- 0	= Kein Scrollen beim Schreiben in letzter Spalte und/oder Zeile.
3	- 1	= 10 Sekunden Delay beim Einschalten
3	- 0	= Kein Delay

Die folgenden vier Bytes definieren die CURSOR-Tasten des Terminals. Falls Euer Terminal keine CURSOR-Tasten oder anderweitig definierte Cursorsteuerung hat, kann man jeden Controll- oder ESC-Code mit 1 byte Länge wählen. Hier sind auf jeden Fall die Cursor-Codes von WORDSTAR vorzuziehen und werden allgemein benutzt.

Wordstar- 'DIAMANT'

Control-E (Hoch)

Control-S (Links) (Rechts) Control-D

Control-X (Runter)

BYTE	Bedeutung	
16	Cursor Up	Cursor UP Befehl des Terminals (^E)
17	Cursor Down	Cursor DOWN Befehl des Terminals (^X)
18	Cursor Right	Cursor RIGHT befehl des Terminals (^D)
19	Cursor Left	Cursor LEFT Befehl des Terminals (^S)

Einige Terminals brauchen DELAYS zur Durchführung spezieller Funktionen. Hier werden die Wartezeiten (Delays) im MILLI-sekunden angegeben. Zum Erreichen der maximalen Ausgabe-geschwindigkeit des Terminals sollten die Wartezeiten aus den Serviceunterlagen übernommen werden.

Wartezeiten:

BYTE	Bedeutung	
20	Clear Screen	Zeit in Millisekunden (0 to 255)
21	Move Cursor	Zeit in Millisekunden (0 to 255)
22	Clear to EOL	Zeit in Millisekunden (0 to 255)

Ab Byte 23 vom Beginn der TCAP werden die Befehlzeichenketten mit einem NULL-Byte abeschlossen. Hierdurch ist die Befehlslänge nicht auf nur ein oder zwei Bytes begrenzt.
Falls einige der vorgesehenen Möglichkeiten im eigenen Terminal nicht vorhanden ist, wird dies durch eine einzelne Null '0' angegeben (binäre 0).

Befehl	Bedeutung
Clear Screen	(CL) - Löscht Bildschirm; Cursor linke obere Ecke
Cursor Movement	(CM) - Gesamte Zeichenfolge zur Cursorpositionierung.Die Einzelheiten sind am Ende der Auflistung angegeben oder in 'THE MANUAL' von Richard Conn nachzuschlagen. Hier wird am Ende der Sequenz ein '%N' angehängt um eine binäre 0 ans Terminal zu senden.
Clear to End-of-Line	(CE) - Lösche von Cursor bis Zeilenende.
Standout Begin	(SO) - Normaler Darstellungsmodus, im allg. halbe Helligkeit und nicht Revers.
Standout End	(SE) - Ende Standout Modus , ab hier REVERS oder/und doppelte Helligkeit.
Initialize Terminal	(TI) - Initialisiert das Terminal
De-initialize Term	(TE) - Setzt die Initialisierung zurück
Delete Line	(LD) - Löscht die Cursorzeile und scrollt die unteren Zeilen um 1 höher.
Insert Line	(LI) - Fügt vor der Cursorzeile eine leere Zeile ein.
Clear to EOS	(CD) - Löscht Bildschirm ab Cursor bis Schirmende.

Die nun folgenden Definitionen implementieren Grafik-möglichkeiten.

Graphics Delay - Delay for Graphics On/Off (0-255). Sollte größer als 2 sein (Angabe in Millisekunden).

Graphics Mode On	(GO)	- Grafik an
End Graphics Mode	(GE)	- Grafik aus, Textwidergabe an
Cursor Off	(CDO)	- Cursor aus, wird für Pull-Down Menues benötigt.
Cursor On	(CDE)	- Cursor an , s.o.

Die Grafikzeichen sind bei jedem Terminal anders, und es könnten sogar normale Textzeichen verwendet werden, zum Beispiel das Asterix '*' als Ecke, '!' als vertikale Linie und '-' als horizontale Linie. Hiermit ist auch auf einem nicht-grafikfähigem Terminal Buisiness-Grafik möglich.

```
Befehl                    Bedeutung
-------------------------------------------------------------------
Upper-Left Corner      (GULC) -   **STRING***
Upper-Right Corner     (GURC) -     "     "    "
Lower-Left Corner      (GLLC) -     "     "    "
Lower-Right Corner     (GLRC) -     "     "    "
Horizontal Line        (GHL)  -     "     "    "
Vertical Line          (GVL)  -     "     "    "
Solid (Full) Block     (GFB)  -     "     "    "
Hashed Block           (GHB)  -     "     "    "
Upper Intersect        (GUI)  -     "     "    "
Lower Intersect        (GLI)  -     "     "    "
Intersection           (GIS)  -     "     "    "
Right Intersect        (GRTI) -     "     "    "
Left Intersect         (GLTI) -     "     "    "
```

Der Rest der TCAP sollte mit binären Nullen aufgefüllt werden, um zukünftige Ergänzungen einfach zu handhaben.
Das folgende Beispiel könnte man auch als Assembler-File für das eigene Terminal ändern, assemblieren und als TCAP übernehmen.

Beispiel:

```
; Z3TCAP:  HEATH19.Z80
; Author:  Harold F. Bower

; Z3 Termcap implementing the Heath/Zenith-19 command set

ESC     EQU     27              ; Escape character

; New Terminal Capabilities Data

Z3TCAP: DEFB    'Heath/Zenith-19' ; Name of terminal (15 chars)

TRMMOD: DEFB    00000111B       ; B3 = 0 - Term Ready quickly
                                ; B2 = 1 - Reverse Video,
                                ; B1 = 1 - Wrap $ EOL,
                                ; B0 = 1 - Scroll $ EOP

        DEFB    'E'-'$'         ; Cursor up  (WS Diamond)
        DEFB    'X'-'$'         ; Cursor down
        DEFB    'D'-'$'         ; Cursor right
        DEFB    'S'-'$'         ; Cursor left

; Heath-19 doesn't need delays

        DEFB    00              ; Cl delay
        DEFB    00              ; Cm delay
        DEFB    00              ; Ce delay

; Strings start here.

        DEFB    ESC,'E',0       ; CL str (Clear, Home Cursor)
        DEFB    ESC,'Y%+ %+ ',0 ; CM str (Cursor positioning)
        DEFB    ESC,'K',0       ; CE str (Clear to End-of-Line)
        DEFB    ESC,'p',0       ; SO str (Go to Revers Video)
        DEFB    ESC,'q',0       ; SE str (Return Normal Video)
        DEFB    0               ; TI str (Initialize Terminal)
        DEFB    0               ; TE str (De-initialize Term)


; Extensions to Standard TCAP
```



```
; Extensions to Standard TCAP
        DEFB    ESC,'M',0       ; LD str (Delete Line)
        DEFB    ESC,'L',0       ; LI str (Insert Line)
        DEFB    ESC,'J',0       ; CD - Clear to EOS String
        DEFB    0               ; GO/GE - Graphics On/Off Delay
        DEFB    ESC,'F',0       ; GO - Graphics Mode On
        DEFB    ESC,'G',0       ; GE - Graphics Mode End
        DEFB    ESC,'x5',0      ; CDO - Cursor Off string
        DEFB    ESC,'y5',0      ; CDE - Cursor Enable string
        DEFB    'f',0           ; GULC - Upper Left Corner [*]
        DEFB    'c',0           ; GURC - Upper Right Corner [*]
        DEFB    'e',0           ; GLLC - Lower Left Corner [*]
        DEFB    'd',0           ; GLRC - Lower Right Corner [*]
        DEFB    'a',0           ; GHL - Horizontal Line [-]
        DEFB    '',0            ; GVL - Vertical Line [¦]
        DEFB    'i',0           ; GFB - Full Block String [*]
        DEFB    'w',0           ; GHB - Hashed Block String [▒]
        DEFB    'u',0           ; GUI - Upper Intersection [+]
        DEFB    's',0           ; GLI - Lower Intersection [+]
        DEFB    'b',0           ; GIS - Intersection [+]
        DEFB    'v',0           ; GRTI - Right Intersection [+]
        DEFB    't',0           ; GLTI - Left Intersection [+]
;  File unused space with Nulls

        REPT  128-[$-Z3TCAP]
        DEFB  0
        ENDM

      END
;------------- End of Sample TermCap -------------
```

Soweit mit dem Beispiel. Wie man sieht, geht Grafik auch ohne grafikfähigem Terminal und diese 'grafische Darstellung' war für Buisinessgrafik durchaus üblich.
Da die Hilfsprogramme TCSELECT und TCMAKE nur die Grundfunktionen des Terminals installieren, muß man zur Komplettinstallation das Programm TCSRC nehmen und das erstellte *.Z80 File auf das gewünschte Terminal ändern. Nach assemblieren vom MYTERM.Z80 und umbenennen auf den gewünschten Namen kann diese TCAP ins ZCPR eingebunden werden. Hierbei ist natürlich TCVIEW zur Kontrolle eine große Hilfe.

Noch eine Anmerkung zu den ZCPR-Versionen. Ich gebe persöhnlich den selbstinstallierenden ZCPR-Versionen NZCOM und Z3PLUS den Vorzug, habe aber auch auf meinem MORROW-MD3 ZCPR33 per 'Hand' installiert. Diejehnigen unter Euch, welche auch eine 'Handinstallierung' machen wollen sollten in der PD-Software nach einer eventuellen Installierung für Ihren Rechner nachschlagen. In der SIG ist z.B. für Kaypro, Applle (?) und Morrow ein installiertes ZCPR vorhanden. Ebenfalls ist für 8080/85 eine Version zum ZCPR2 vorhanden, allerdings laufen die neueren Utillities nicht unter 8080/85.
Eine große Hilfe ist das von Helmut Jungkunz zusammengestellte 'ZCPR - Installation-Kit'. Es beinhaltet aus der PD die nötigen Dateien und Informationen zur 'Handinstallierung'. Diese Disketten sind demnächst auch bei Rüdiger Sörensen erhältlich . Ebenfalls sollte jeder Interessent an PD oder ZCPR die SAGE-PD beim Rüdiger anfordern.
    Diese (25 * 780KB) Disketten wurden durch die Initiative von Helmut Jungkunz durch G.Sage, einem der Programmierer des ZCPR's ,nach GERMANY gesandt. Hier findet man viele neue Programme und unter anderem auch eine Menge zum ZCPR.

Literaturhinweise:

The Z-System User's Guide(Bruce Morgen.Richard Jakobson). Eine Beschreibung des Z-Systemes für weniger versierte Anwender.

The ZCPR3.3 User's Guide(G.Sage). Dies ist eine Beschreibung des ZCPR Version 3.3 Commandoprozessor. Es beinhaltet viele Beispiele für die vorteilhafte Nutzung des ZCPR und gilt meist auch für ZCPR Version 3.4 (ZCPR 3.4 ist copyrighted).

ZCPR3: The Manual(Richard Conn). Dies war die 'Bibel' für ZCPR3, aber vieles darin ist heute nicht mehr aktueller Stand. Die Beschreibungen darin zum Z-Help , den Menü Shells und der TCAP (einschließlich TCSELECT,TCMAKE,TCCHECK) sind sehr hilfreich.

Für diejehnigen unter euch, welche das ZCPR selber installieren wollen ist folgendes zu empfehlen:

ZCPR3:The Libraries(Richard Conn). Dieses Buch beschreibt komplett einige hundert Routinen, welche das Schreiben vom Z-System Programmen so einfach macht wie 'Programming in high level language', obwohl man in assembler programmiert. Die letzte Version der Routinen (V4LIB) ist auf SAGE.PD 25.

ZCPR3 Shells(David McCord).Eine Beschreibung des SHELL Systemes unter ZCPR, welche zur effizienten Nutzung und Programmierung beiträgt.

Hier noch einige Literaturangaben zum CP/M:

Betriebssystem CP/M(Jürgen Plate). Dieses Buch wurde ergänzend zum NDR oder KLEIN - Rechner geschrieben und erläutert den Aufbau und die Programmstrucktur von CP/M2.x. Es ist nicht unbedingt als Bedienungsanleitung zu nutzen.

CP/M Handbuch mit MP/M(Rodnay Zaks). Der Autor dürfte vielen für seinen lehrreichen Schreibstiel bekannt sein. Diese Buch ist eine ausführliche Bedienungsanleitung für CP/M, geht aber nicht näher auf Systemprogrammierung ein.

CP/M Anwenderhandbuch(Thom Hogan). Ein Handbuch zum CP/M (Version bis 2.x) und CP/M86, welches keinem fehlen sollte. Thom Hogan hat auch das Handbuch zum OSBORNE 1 geschrieben.

****

Die oben vorgestellten Programm(packet)e NZCOM und Z3PLUS sind copyrighted und dürfen nicht an Dritte weitergegebn werden. Ebenso ist ZCPR3.4 copyrighted.
ZCPR3.3 ist Public Domain Software und zum *privaten Nutzen* freigegeben.

Da Herr Helmut Jungkunz in Deutschland alleiniger Distributor der Fa. Alpha Systems,711 Chatsworth Place, San Jose, CA 95128 ist, ist er für weitere Informationen zu NZCOM und Z3PlUS , ebenso wie für Copyrighted Software wie Turbo Pascal, Wordstar 4 etc.. der Ansprechpartner.

Mein Dank auch G.Sage, Bridger Mitchel, Helmut und allen anderen, welche sich für den Z80 (und kompatible) Prozessor verdient machen.

F. Churchen.

A

AAA

Die drei großen A's des Computerfreaks bedeuten Auspacken, Anschließen und Abhauen. Wer nicht abhaut, muß eben anfangen und wird Byte-süchtig.

Absturz

a) Verlust aller Sinne bei leichtsinnigem Umgang mit Alkohol.
b) Verlust aller Programmdaten bei leichtsinnigem Umgang mit dem Computer, meist nach vorangegangenem leichtsinnigen Umgang mit Alkohol.

Action

a) Computerspielgrenze, bei der es kracht und zischt, daß es eine Freude ischt.

b) Mahnungen diverser Inkassobüros an Anwender, die mit ihren Raten für den Computer im Rückstand sind.

Download vom Brett "MS_DOS-Anwendungen" der BRAINSTORM
Datum: 18-Nov-90 15:29
Von : Michael Mueller
An : All

Betreff : Korrekte Syntaxbeschreibung von Befehlszeilen !

Korrekte Syntaxbeschreibungen in der Backens-Naur-Form (BNF)

Immer wieder taucht nach dem Herunterladen von Programmen, vor allem von Hilfsprogrammen, wie Entpackern, das Problem auf, wie man Befehle und Parameter korrekt eingibt. Dazu ist es wichtig, die mitgelieferten Syntaxbeschreibungen richtig zu verstehen oder selbst eine korrekte Beschreibung zu erstellen.

Fuer diesen Zweck gibt es die sog. "Backens-Naur-Form" (BNF), eine "Metasprache", d.h. eine Sprache, die eine andere Sprache beschreibt. Mit der BNF lassen sich auch ganze Programmiersprachen spezifizieren. Besonders geeignet fuer Mailboxen ist sie, weil sie ohne graphische Hilfsmittel auskommt und daher als einfache ASCII-Datei uebertragen werden kann. Viele Dokumentationen verwenden daher dieses Format, wenn auch nicht immer ganz richtig.

Wie ist die BNF nun aufgebaut?
Im Grunde laeuft es so ab, dass alle Elemente der Programmiersprache oder der Eingabezeile nacheinender definiert werden. Das sieht so aus:

```
SYMBOL::=LISTE
```

Die das SYMBOL ist die Bezeichnung, fuer das, was definiert werden soll, z.B. eine Befehlszeile. Die LISTE muss alle(!) Moeglichkeiten enthalten, die fuer das Symbol erlaubt sind. Die LISTE kann weitere SYMBOLE enthalten, die an anderer Stelle definiert werden. Solche SYMBOLE heissen auch "Nichtterminale Symbole".

Hierzu gibt es einige Zeichen oder Zeichenfolgen, die Teile der Metasprache BNF sind, und nicht zur beschriebenen Sprache gehoeren muessen. Solche Zeichen heissen auch "Metazeichen".

Folgende Metazeichen gibt es:

- `::=` Das haben wir schon kennengelernt. Es kommt in jeder Definition vor und steht immer nach dem zu definierenden Symbol.
- `|` Der senkrechte Strich steht zwischen Symbolen, zwischen denen eines (!) ausgewaehlt werden muss.
- `"` Anfuehrungszeichen schliessen Symbole ein, die nicht mehr anderswo naeher definiert werden, sondern Teil der zu beschreibenden Sprache sind. Das heisst, dass das, was in den Anfuehrungszeichen steht, woertlich so geschrieben werden muss. Solche Symbole heissen auch "Terminale Symbole".
- `{}` Geschweifte Klammern schliessen Elemente ein, die "freiwillig" (alternativ) verwendet und beliebig oft wiederholt werden koennen.
- `[]` Eckige Klammern sind eigentlich kein Teil der BNF (man kommt auch ohne sie aus), aber man stoesst trotzdem haeufig auf sie, da man mit ihnen manches kuerzer ausdruecken kann. Sie schliessen dann Elemente ein, die alternativ verwendet koennen, aber NICHT wiederholt werden duerfen.

WICHTIG: Jedes verwendete, nichtterminale Symbol MUSS anderswo definiert werden. Wenn in dessen Definition wieder nichtterminale Sybole verwendet werden dann muss auch dieses definiertwerden und so weiter. Man darf die so entstehende Kette erst dann abbrechen, wenn sich jedes nichtterminale Symbol ausschliesslich auf terminale Symbole zurueckfuehren laesst.

Rekursive Definitionen sind erlaubt, d.h. in der LISTE darf das zu definierende Symbol erneut auftauchen.

ACHTUNG: BNF beschreibt nur die reine Syntax, d.h. welche Eingaben erlaubt sind. Die Bedeutung der Befehle muss extra beschrieben werden.

Beispiele aus der Programmiersprache BASIC:

```
EINGABEZEILE::=ZEILENNUMMER BEFEHLSFOLGE|BEFEHLSFOLGE
```

Das Gleiche mit eckigen Klammern:

```
EINGABEZEILE::=[ZEILENNUMMER]BEFEHLSFOLGE
```

ZEILENNUMMER und BEFEHLSFOLGE sind nichtterminale Symbole.
Sie muessen naeher definiert werden:

```
ZEILENNUMMER::=ZIFFER{ZIFFER}
```

Das bedeutet, dass man mindestens einmal ZIFFER schreiben muss. Die maximale Anzahl der Ziffern ist noch nicht bestimmt, und muss in einer Erlaeuterung angegeben werden. Erlaeuterungen erfolgen in der Ragel erst nach allen Definitionen z.B. in Form einer Fussnote. Erlaeuterungen gehoeren NICHT zur BNF.

Weiter in der Definition: Auch ZAHL ist nichtterminal:

```
ZIFFER::="0"|"1"|"2"|"3"|"4"|"5"|"6"|"7"|"8"|"9"
```

Nun ist man endlich bei terminalen Symbolen, naemlich den Ziffern angekommen und hat damit ZEILENNUMMER vollstaendig definiert.

Nun zur Definition von BEFEHLSFOLGE:

```
BEFEHLSFOGE::=BEFEHL{":"BEFEHL}
```

Man koennte anstelle der geschweiften Klammern auch mit einer rekursiven Definition arbeiten:

```
BEFEHLSFOLGE::=BEFEHL|BEFEHL":"BEFEHLSFOLGE
```

Die Rekursion bricht ab, sobald man die erste Alternative waehlt.
Jetzt muss BEFEHL definiert werden.

```
BEFEHL::=PRINTBEFEHL|INPUTBEFEHL|ZUWEISUNG  usw.
```

Jetzt muessen saemtliche Befehle definiert werden. Das wuerde hier zu weit fuehren. Eine komplette Syntaxdefinition von Standard-BASIC wuerde etwa 5 engbedruckte Seiten fuellen, aber auch saemtliche Zweifen an der richtigen Schreibweise eines Befehls ausraeumen.

Ich hoffe, das genuegt, um BNF hinreichend und verstaendlich zu beschreiben. Wer ein Programm verbreitet, das irgendwelche Texteingaben verlangt, sollte in der Dokumentation dazu, die Eingaben auf diese Art definieren. Das mag zwar manchmal etwas umstaendlich erscheinen, erhoeht jedoch die Qualitaet der Dokumentaion enorm. Merke: Bei professionellen Programmen mit guter Dokumentation, wird mit der Erstellung derselben ein aehnlicher Aufwand getrieben wie mit der Erstellung des Programmes selbst! Das wuerde niemand machen, wenn es nicht wirklich sinnvoll waere!!

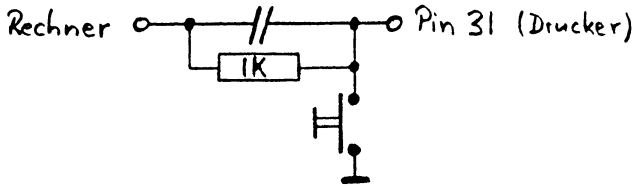
Mit diesem Hinweis: Ciao

Michael Mueller

..Hardware ..Hardware ..Hardware

Drucker-Reset

Neuere Drucker haben in der Regel einen eingebauten Buffer mit 16 bis 64KB Kapazität und mehr. Normalerweise ist das ganz angenehm, wenn der Rechner fast sofort nach dem Start des Ausdrucks schon wieder zur Verfügung steht, aber manchmal kann das auch zu Problemen führen. Dann nämlich, wenn man im Ausdruck einen Fehler bemerkt und den Drucker gerne anhalten würde, bevor er noch den Rest seines Buffers völlig sinnlos auf's Papier kleckert. Einige Drukker, wie z.B. der HP Deskjet, haben dafür einen Reset-Knopf, die meisten anderen lassen sich aber nur materialzermürbend mit dem Holzhammer (Netzschalter) bremsen. Die Centronics-Schnittstelle bietet nun aber die relativ unbekannte Möglichkeit, dasselbe viel sanfter zu erledigen. Wie Ihr in der folgenden Schaltung, die aus der Elektor stammt, sehen könnt, ist der Aufwand denkbar gering. Hoffentlich haben sich alle Hersteller an die Norm gehalten und nicht gemeinerweise dort den Selbstzerstörungseingang hingelegt. Garantien irgendwelcher Art möchte ich lieber nicht übernehmen.



Alexander Schmid

Megabit-RAMs einfach erfrischt

Helmut Bernhardt

Im letzten Info hat Alexander eine aufwendige Erweiterung der Erweiterung der 7Bit-Refreshadresse des Z80 auf 8 Bit beim 256k-Banker und nun 9 Bit für Megabit-RAMs vorgestellt. Einerseits wäre ich ziemlich stolz gewesen, sowas selbst ersonnen zu haben, andererseits bin ich aber viel zu faul, die bisherige aus der c't übernommene Schaltung zu durchblicken und dann noch auf 9 Bit zu erweitern - es geht nämlich auch ohne.

Wie aus Datenblättern und den beiden üppigen Knowledge-Anhäufungen zu DRAMs in den Heften 10 und 11 aus c't 1990 von Johannes Assenbaum zu entnehmen ist, gibt es für alle 1M-Bit-DRAMs die Möglichkeit, das Refreshing durch den /CAS-before-/RAS-Rfresh zu erledigen. Bei 256K-Bit--DRAMs wird das nur bei einigen Herstellern unterstützt - hier sollte man in entsprechenden Schaltungen diese Technik nicht anwenden, weil nicht gesichert ist, ob die darin eingesetzten Chips den Sport mitmachen.

Beim /CAS-before-/RAS-Refresh muß nur ein low aktives /CAS an die RAMs gelegt werden, bevor ein /RAS kommt. Dann wird im RAM mit einem internen Zähler eine Columnadresse erzeugt und die damit adressierten Speicherzellen werden refreshed. Ein externer Memorycontroller (bei uns z.B. der Z80) braucht sich dann nicht mehr um eine von außen zugeführte Refreshadresse zu kümmern. Damit ist es auch nicht mehr nötig, daß in der Breite der von den RAMs verwendeten Refreshadresse die niedrigen Adressen der CPU zusammen beim /RAS durch die Adreßmultiplexer durchgeschaltet sein müssen. Die CPU-Adressen können in wilder Folge an die A- und B-Eingänge der Multiplexer gelegt werden - wie es für das Layout gerade am besten kommt.

Und wie erzeuge ich einen /CAS-before-/RAS-Refresh? Ich leite wie immer ein aus Adreßdecodierung für die RAM-Bank und Timing (relativ zu /RAS und MUX) zusammengesetztes /CAS her und lege es über ein AND-Gater mit /RFSH bzw. /RFS beim HD64180 zusammen und verabreiche den RAMs das Produkt dieser geringen Mühe an den /CAS-Eingängen.

Beim Refresh der CPU erscheint das /RFSH-Signal sehr viel früher als das als /RAS verwendete /MERQ. Damit ist die Bedingung der RAMs für den /CAS-before-/RAS-Refresh erfüllt. Bei normalen RAM-Zugriffen wird das herkömmliche /CAS, das zeitlich versetzt nach /RAS und MUX aktiv wird, zu den RAMs durchgeschaltet.

Wer mehr dazu wissen möchte, kann an anderer Stelle in diesem Info eine Anwendung in Form einer 2MB-RAM-Floppy finden, wo diese Refreshart den Schaltungsaufwand wesentlich reduziert hat.

2MB-RAM-Erweiterung für Prof 180x

Helmut Bernhardt

Zur Erweiterung des Speichers über die 512K auf dem CPU-Board des Prof 180x hinaus hat Conitec die Turbo-RAM vorgesehen. Das 'Turbo' in dieser Bezeichnung ist aber mit Vorsicht zu genießen. Da diese Speichererweiterung auf dem ECB-Bus steckt, liegen zwischen CPU und diesem Speicher 2 Datentreiber. Bei 9,2 MHz ist die Turbo-RAM nur mit einem Wait zu betreiben, was dann auch beim Zugriff auf die ursprünglichen 512K enorm bremst. Hier sei eine Alternative vorgeführt, die ohne Waits eine Erweiterung um 2MB in Form eines Huckepack-Boards auf dem CPU-Board bringt.

Um zu erklären, wie das machbar ist, soll zunächst auf die Verhältnisse eingegangen werden, wie im Prof der Zugriff auf verschiedene Speicher geregelt ist. da dies alles hauptsächlich im PAL 14L8 gesteuert wird, sei das disassemblierte PAL-Listing hier wiedergegeben:

```
PAL14L8
ausgelesen mit c't PAL-Brenner, 1988 Be
Grundversion nach Conitec

me ioe rd lir busak a18 e mm0  mm1 adr a5 gnd
a2 a7 a19 rom cas0 cas1 fdc gate iosel dir a19 vcc

/dir    = /ioe * /rd * /adr * a7 * /a5          ;Richtungssteuerung
        + /me * /rd * /a19                      ; ECB-Bus-Daten-
        + rd * lir * busak                      ; -treiber
/iosel  = /ioe * lir * /a18 * /adr * a7 * /a5 * /a2
/fdc    = /ioe * lir * /a18 * /adr * a7 * /a5 * a2
/gate   = mm0 * /mm1                            ;Freigabe ECB-Bus-
        + mm1                                   ; -treiber
/a19    = /mm1                                  ;Freigabesignal für
        + a18 * /mm0                            ; externe Speicher
        + /a18 * mm0
/rom    = /me * /a18 * /rd * /mm0 * /mm1
/cas0   = /me * /a18 * e * rd * /mm0 * /mm1
        + /me * /a18 * e * mm0
/cas1   = /me *  a18 * e * /mm0
        + /me *  a18 * e *  mm0 * /mm1
```

'adr' ist ein Eingangssignal für das PAL, in dem A3, A4 und A6 entsprechend den zu erzeugenden I/O-Freigabesignalen im Bereich D8H-DFH decodiert sind. 'E' ist ein spezielles Taktsignal des HD64180, das hier zum Timing der /CAS-Signale verwendet wird. MM0 und MM1 sind Ausgänge eines 74LS259-Latches mit einzeln zu setzenden Bits. Die übrigen Eingangssignale des PALs sind die normalen Signale der CPU.

Die I/O-mapped Freigabesignale /IOSEL und /FDC sind hier uninteressant. Für die Steuerung des ECB-Businterface und der memory-mapped Resourcen werden im PAL die Signale A19, /GATE, /ROM, /CAS0 und /CAS1 erzeugt. /ROM, /CAS0 und /CAS1 sind low aktive Freigabesignale für das ROM und für je einen 256K-Block RAM auf dem CPU-Board. /GATE, das an die Enable-Pins 19 der 74ALS245-Bustreiber gelegt ist, steuert die grundsätzliche Möglichkeit der CPU, auf den ECB-Bus zugreifen zu können. Dieses Signal muß high sein, wenn die Prof 180x-Karte im Slave-Modus für sich allein werkeln soll und eine andere CPU-Karte den Bus kontrollieren soll.

/DIR steuert die Richtung, in der der 74ALS245-Datentreiber die Daten vom/zum ECB-Bus treibt; /DIR = 0 bedeutet, daß die Daten vom Prof 180 zum ECB-Bus getrieben werden. In die Richtungssteuerung geht nicht nur ein, ob gelesen oder geschrieben wird (/RD) sondern auch, ob die adressierte Baugruppe auf dem CPU-Board oder 'draußen' auf dem Bus liegt. Wenn eine Baugruppe auf dem CPU-Board gelesen wird, muß der Bustreiber zum Bus hin treiben, um auf dem internen Datenbus keinen Datenkonflikt zu erzeugen. Außerdem wird auch das Signal /BUSAK ausgewertet, das bei Zugriff eines externen DMA-Controllers auf Baugruppen des CPU-Boards die Treiberrichtung in Bezug auf /RD umkehrt.

A19 ist ein high aktives Freigabesignal, das auf den ECB-Bus gelegt ist und bei Conitec u.a. für die Steuerung der Turbo-RAM benutzt wird.

Die Analyse des disassemblierten PAL-Listings ergibt für die Steuerung der Freigaben von ROM und internem und externem RAM die Beziehungen:

Steuersignale MM1	MM0	A18	freigegebene Baugruppe	Pegel v. /GATE	Pegel v. A19
0	0	0	ROM bei /RD; RAM bei /WR	1	0
0	0	1	RAM Bank 1	1	0
0	1	0	RAM Bank 0	0	0
0	1	1	RAM Bank 1	0	0
1	0	0	externes RAM	0	1
1	0	1	RAM Bank 1	0	0
1	1	0	RAM Bank 0	0	0
1	1	1	externes RAM	0	1

Wenn MM0 und MM1 beide low sind, ist /GATE high und es kann nicht auf Speicher auf dem ECB-Bus zugegriffen werden. Durch Umschalten auf MM0=1 wird das beim Lesen in die unteren 256K eingeblendete ROM abgeschaltet. Mit MM1=1 wird der grundsätzliche Zugriff auf externe Speicher ermöglicht. Dann wird mit MM0 gesteuert, ob der externe Speicher in die unteren oder die oberen 256K des HD64180-Adreßraums eingeblendet wird.

Für das Bereitstellen des bei Conitec 'externen' Erweiterungsspeichers nun innerhalb des CPU-Boards muß die Bussteuerung nun etwas anders gestaltet werden. Außerdem müssen Freigabesignale für zwei weitere RAM-Banks mit je 1 MB geschaffen werden. Dafür muß das PAL zunächst entrümpelt werden. Auf das Signal /GATE kann man verzichten, wenn man die CPU-Karte nur als Master fahren will. Die Pins 19 der 74ALS245-Bustreiber werden einfach an Masse gelegt.

Da kein externer Speicher zu steuern ist, wird auch das Signal A19 nicht mehr benötigt. Es sind dann zwei Ausgänge und ein Eingang am PAL freigeworden. Die beiden Ausgänge sollen die Freigabesignale für die beiden zusätzlichen 1MB-RAM-Blocks ergeben und der Eingang wird für die Selektion eines dieser beiden Blocks mit einem Signal Q2 beaufschlagt. Dann bekäme ein neues PAL den Inhalt:

```
PAL20L8
08.08.90                        H. Bernhardt
Erweiterung des Prof 180x auf 2,5 MB RAM

me ioe rd lir busak a18 e mm0  mm1 adr a5 gnd
a2 a7 cas2 rom cas0 cas1 fdc cas3 iosel dir q2 vcc
```

```
/dir    = /ioe * /rd * /adr * a7 * /a5
        +/me * /rd
        + rd * lir * busak
/iosel  = /ioe * lir * /a18 * /adr * a7 * /a5 * /a2
/fdc    = /ioe * lir * /a18 * /adr * a7 * /a5 *  a2
/rom    = /me * /a18 *     /mm0 * /mm1 * /rd
/cas0   = /me * /a18 * e * /mm0 * /mm1 *  rd
        + /me * /a18 * e *  mm0
/cas1   = /me *  a18 * e * /mm0
        + /me *  a18 * e *  mm0 * /mm1
/cas2   = /me * /a18 * e * /mm0 *  mm1 * /q2
        + /me *  a18 * e *  mm0 *  mm1 * /q2
/cas3   = /me * /a18 * e * /mm0 *  mm1 *  q2
        + /me *  a18 * e *  mm0 *  mm1 *  q2
```

Das Steuersignal Q2 und auch noch zwei Pseudoadressen zu Selektion eines von 4 Blocks mit 256K innerhalb eines 1MB-Blocks im Erweiterungsspeicher werden mit einem zusätzlich bereitgestellten Latch 74LS174 erzeugt. Dieses Latch wird durch einen OUT-Befehl an Port D7H beschrieben. D0 und D1 geben darin die Pegel der Pseudoadressen A18 und A19 und D2 den Pegel von Q2 vor.

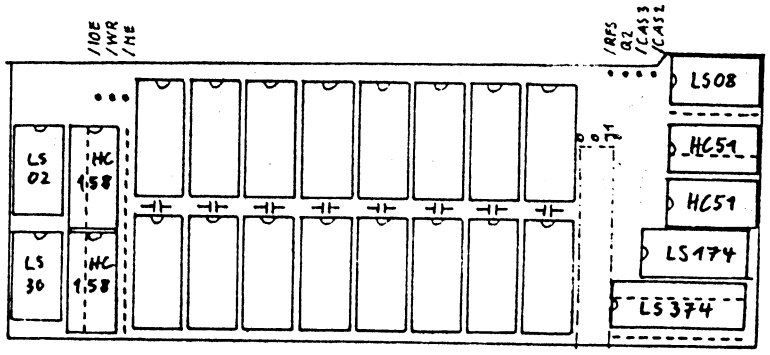
Die gesamte Erweiterungsschaltung incl. RAMs kommt auf einem Huckepackboard unter, das in die Sockel der Adreßmultiplexer Z17 und Z18 (74HC158) sowie Z11 (74HC51) und des Datenlatch Z15 (74LS374) der Druckerschnittstelle gesteckt wird und diese ICs mit aufnimmt. Damit stehen alle gemultiplexten Adressen, die Daten, die Steuersignale /RAS, MUX und /MUX auf dem Huckepackboard zur Verfügung. Die Pseudoadressen A18' und A19' werden mit einem zusätzlichen 74HC51-Gatter gemultiplext.

Es müssen dann nur noch folgende Signale durch freie Verdrahtung zwischen Huckepackboard und CPU-Board verbunden werden:

/IOE (74LS02, Pin6)	von Pin5 des Z7, 74ALS245
/WR (RAMs, Pins2)	von Pin6 des Z7, 74ALS245
/ME (RAMs, Pins3)	von Pin2 des Z7, 74ALS245
/RFS (74LS08, Pins1,13)	von HD64180, Pin57
Q2 (74LS174, Pin5)	an Pin23 des PAL
/CAS3 (74LS08, Pin12)	von Pin18 des PAL
/CAS2 (74LS08, Pin1)	von Pin13 des PAL

Und wofür ist so viel Speicher gut? Natürlich wird der erstmal als RAM-Floppy genutzt. Anstatt diesen Speicher zusammen mit den restlichen 320K, die im Grundspeicher schon als RAM-Floppy genutzt werden, als eine RAM-Floppy zu betreiben, soll hier eine zweite RAM-Floppy von 2MB Kapazität eingerichtet werden. Das vereinfacht die Software etwas, wenn nicht ständig geprüft werden muß, ob ein zu übertragender Sektor nun auf der Erweiterung oder im Grundspeicher zu holen ist. Der benötigte Treiber läßt sich aus dem für die 'alte' RAM-Floppy bestehenden Treiber ohne große Änderungen umformen. Es kommt beim Login nur hinzu, daß MM1 auf high gesetzt werden muß.

Bei den Lese- und Schreib-Routinen ist noch nachzurüsten, daß der in (@trk) übergebene Wert aufzuteilen ist in D0 und D1, die zusammen mit einem gesetzten D2 das Adreßzusatzbyte des DMAC ergeben, und daß D2 -D4 nach zweimaligem Rechtsshiftenals D0-D2 an den Port D7H ausgeben werden. In der Login-Routine der 'alten' RAM-Floppy sollte vorsichtshalber grundsätzlich MM1 low gesetzt werden.



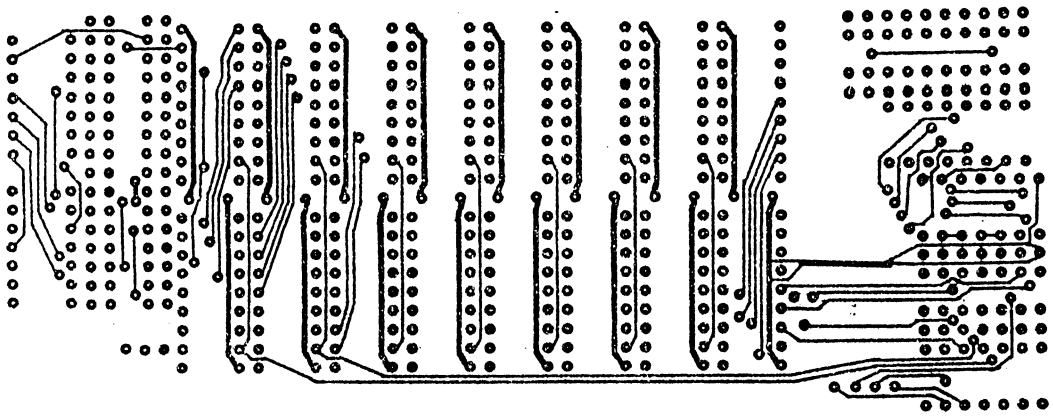
Bestückungs- und Anschlußplan; J1 bleibt bei 1MB-Chips offen

Folgende Datenstrukturen sind für das BDOS vorzugeben:

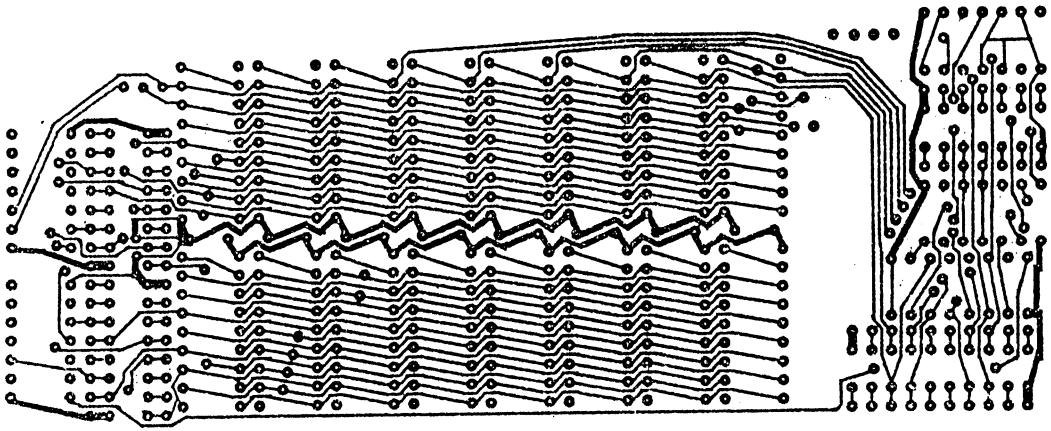
```
        cseg            ;Drive Table
@dtbl   dw      fdsd0,fdsd1,fdsd2,fdsd3 ; 4 Floppy Drives
        dw      fdram                   ; 'alte' RAM-Floppy
        dw      f2dram                  ; 2MB-RAM-Floppy
        dw      hdsk0,hdsk1;hdsk2,hdsk3 ; 4 Harddisk-Partitionen
        dw      0,0,0,0,0,0

        dseg            ;dph's koennen in Bank 0 liegen
;       extended disk parameter header 2MB-RAM-Floppy
        dw      f2write         ;Routinen, um Sektoren
        dw      f2read          ; zu schreiben und zu lesen
        dw      f2login         ;2MB-RAM-Floppy einloggen
        dw      f2init          ;Initialisierung
        db      0,0             ;kein Sektor-Skew
f2dram  dph     0,dpbr2         ;Zeiger auf DPB

        cseg            ;disk parameter block fuer 2MB-RAM-Floppy
dpbr2   dpb     1024,64,2048,4096,512,0,8000h
                ;1024 Bytes/Sektor,  64 Sektoren/Track,
                ;2048 Tracks, Blockgröße 4K, 512 DIR-Entries,
                ;keine Systemspuren, unremovable media
```

Da diese 2MB (wie das sonst meistens bei RAM-Floppies für 8Bit-Computer aussieht) nicht I/O-mapped sind, kann man darin auch Programme laufen lassen. Es liegt dann allerdings am Können des Programmierers, mit dem BDOS ein Abkommen zu treffen, daß dieses nach dem Abarbeiten einer BDOS-Funktion wieder ordnungsgemäß die Bank und auch noch die entsprechende 256K-Portion einschaltet, in der das Programm den BDOS-Call abgesetzt hat.

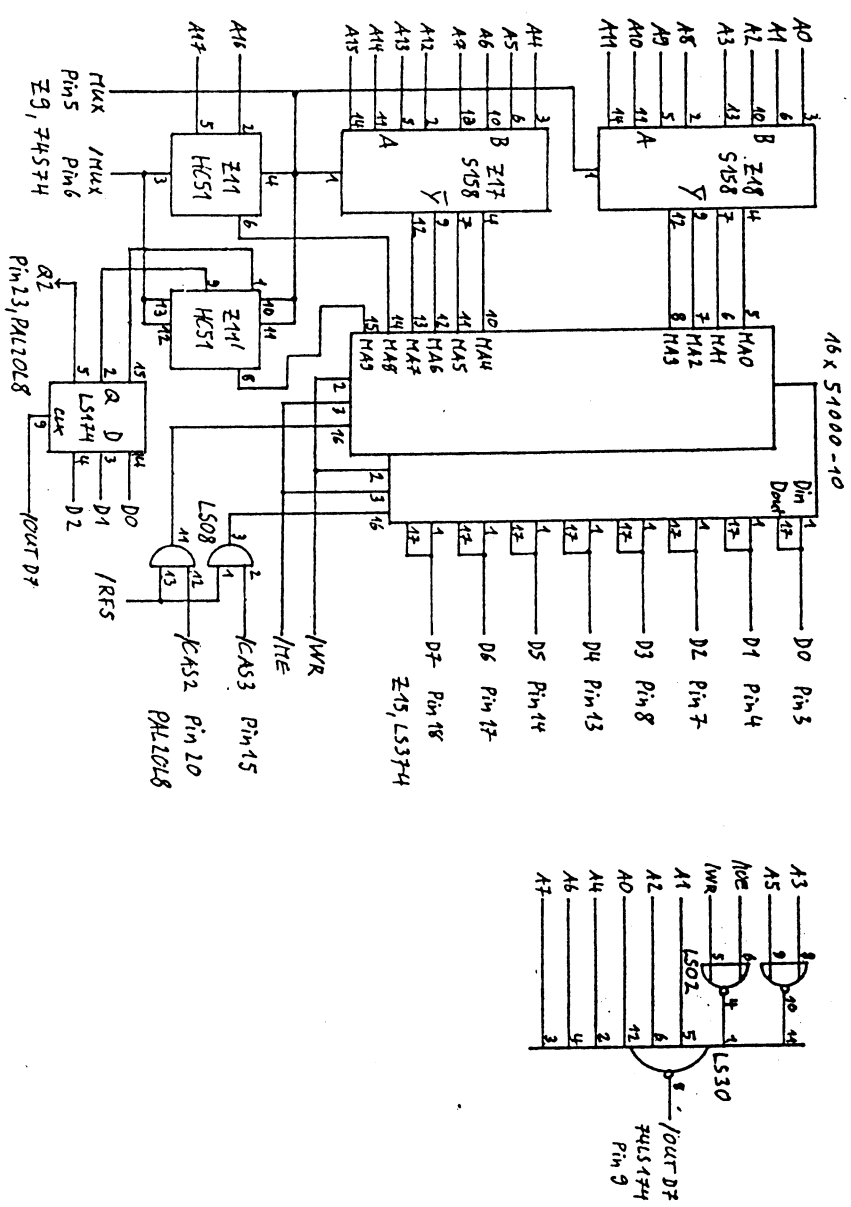
Dieses Problem sei an die Soft-Cracks weitergegeben.



Layout der Bestückungsseite der 2MB-Erweiterung



Layout der Lötseite der 2MB-Erweiterung



Club 80 Börse --- Club 80 Börse

Suche folgende Programme für Genie II (**40 Spuren DD**)
(gegen Tausch oder Bezahlung):

"**Warrior of RAS Trilogy**" von MED Systems:
- Volume I DUNZHIN
- Volume II KAIV
- Volume III THE WYLDE

"**DunjonQuest**"-Serie von Automated Simulations (EPYX):
- Hellfire Warrior
- Keys Of Acheron
- Sorcerer Of Siva
- Temple Of Apshai
- Upper Reaches Of Apshai

"**MACES & MAGIC**"-Serie von Adventure International:
- Balrog
- Morton's Fork
- Reign Of The Red Dragon und Hidden Valley

Data-Writer von Software Options Inc.

Copy Art II von Simutek

WordStar (Vers. 3.00) von Micropro unter NewDOS/80

sowie **L-DOS 5.1.1** und **MultiDOS** unter Double Density ...

Bitte melden bei Jens Günther (Tel.: 02681 / 1553)

Und für die Börse habe ich auch etwas. Da mir mein TRS-80 Mod 4P genügt, möchte ich mich von meinen Speedmaster (80 Zeichen-, Ton-, Uhren- und RS 232-Karte) trennen. Mitgeliefert werden: Monitor, Laufwerkgehäuse mit 1x 40 Trk SS - 1x 40 Trk DS - 1x 80 Trk 5.25 Lw, 1x 8" SS Lw und 8" DS Lw, eine Typenradschreibmaschine (Juki 2200 mit Parallel-Interface), Software satt (8" und 5.25 Disk) und Literatur. Für 500.- DM geht alles an einen Selbstabholer weg.

Wilhelm Tornow
Görlitzer Str. 16

Ich habe einen CP/M-Rechner abzugeben:

Morrow MD10 (Micro Decisions) mit 2 Laufwerken 5.25" und bei Bedarf etwas Software (Turbo Pascal o.ä.) Preis: VHB 150,- DM

Mit freundlichem Gruß,

Sven Uhlin Freiburger Str. 5 D-7500 Karlsruhe 41

Sven Uhlin

K O N D I G U N G

Hiermit kündige ich meine Mitgliedschaft im CLUB 80 zum Ende dieses Jahres.
Grund: Ich gebe auf.

Gleichzeitig biete ich meinen MZ 3541 zum Kauf an.

Und hier einige technische Einzelheiten zu diesem Gerät:
Es handelt sich um einen Personal-Computer mit Z80-CPU und einem Arbeitsspeicher mit einer Größe von 256 KB, von denen bis zu 176 KB als schnelle RAM-DISK verwendet werden können.
Zusätzlich verfügt das Gerät über einen Gafikspeicher von 96 KB. Dadurch ist es möglich, Grafik mit einer Auflösung von 640 mal 400 Punkten und 8 Farben auf dem Bildschirm darzustellen. Es können 4 Bildschirmseiten (3mal Grafik und 1mal Text) gleichzeitig oder einzeln oder auch gemischt auf dem Bildschirm ausgegeben werden. Die Grafikbildschirmseiten können nicht gescrollt werden.
Es sind 2 Stück Doppelseitige 5,25"-Laufwerke mit je 385 KB eingebaut.
Ferner sind vorhanden: eine parallele Druckerschnittstelle nach Centronics-Norm, eine serielle Schnittstelle sowie zwei Bildschirmausgänge, die über getrennte Kanäle bedient werden können. Außerdem befindet sich auf der Rückseite noch ein Anschluß für zwei externe Minifloppy-Laufwerke.
Zum Lieferumfang gehört ein SW-Monitor (grün).
Zwei Betriebssysteme gehören auch dazu:
"FDOS", ein reines Basic-System.
Und "EOS-V3", ein schnelles CPM-kompatibles Betriebssystem.
Von beiden Betriebssystemen werden die grafischen Fähigkeiten des Rechners mit einem komfortablen Befehlssatz voll unterstützt. Die mitgelieferten Handbücher sind als Lehrbücher für Anfänger bestens geeignet.

Wer Interesse daran hat, sollte sich das Gerät vorher bei mir anschauen. (Nach tel. Vereinbarung: 04193/6125)

Ich denke, DM 1000.-- ist ein angemessener Preis.

Henstedt-Ulzburg, den 26.10.1990
Mit den besten Grüßen
Harald Hirsekorn

Harald Hirsekorn
Beckersbergweg 131
2359 Henstedt-Ulzburg 1
Telefon (0 41 93) 61 25

Sehr geehrter Herr Chwolka,

da ich mich beruflich mit MS/DOS und UNIX beschäftigen muß, habe ich keine Zeit mehr für meinen CP/M-Rechner.

Ich verkaufe deshalb:

Hardware:		
1 Z80 Rechner	alphaTronic PC mit eingebauter Bicom-Graphic	
1 Diskettenlaufwerk	Original TA F1(TEAC) mit Controller für 2 Laufwerke	
1 Laufwerkgehäuse	Original TA für F2	
1 Farbmonitor	PRISM QL14	
Software:		
1 Betriebssystem	TA-PC Disc-Basic V. 5.26B ▮	
1 Betriebssystem	CP/M 2.2/3.1 + CP/M 2.2 mit Super BIOS	
1 Programmier-Sprache	Microsoft Basic ▮	
1 Programmier-Sprache	Turbo Pascal 3.0 + TURBO TUTOR 1.0 ▮	
1 Textprogramm	Wordstar 3.0 ▮	
1 Datenbank	dbase II ▮	
	(▮ mit Handbuch)	
Gesamt-Preis:	DM 300,-	

Kurt Klucina
Flurstraße 4

8891 Adelzhausen

Tel.: 08258-348

Weiterhin verkaufe ich verschiedene Bücher:

Das alphaTronic PC-Buch (Basic-Lehrgang)	Erhard Unger	Heim	DM 15,-
BASIC mit dem AT-PC Band1	S. Lumma	Heim	DM 18,-
BASIC mit dem AT-PC Band2	S. Lumma	Heim	DM 18,-
alphtronic PC BASIC Handbuch	Karl-Heinz Hauer	SYBEX	DM 10,-
Assembler-Tricks am AT-PC	Patrick V.Thomas	Heim	DM 15,-
TAalpaTronicPC Almanach 85/86 (Band 1 und 2)	H.Stöber	Stöber	DM 18,-
Vom Umgang mit CP/M	Bernd Pol	IWT	DM 20,-
Betriebssystem CP/M	Plate	Franzis	DM 25,-
Programmieren mit CP/M	Alan R.Miller	SYBEX	DM 20,-
computer compact CP/M	Joseph Reymann	Goldmann	DM 4,-
CP/M kompakt	Plate	Franzis	DM 4,-
dbase II Band1 Einführung	W.Eggerichs	Hüthing	DM 15,-
dbase II Band3 Aufbau und Nutzung v. Datenbanken	W.Eggerichs	Hüthing	DM 15,-
Mikrocomputer Grundkurs Turbo Pascal	Kaier/Rudolfs	Vieweg	DM 12,-
Mikrocomputer Grundkurs Turbo Pascal Übungen	Kaier/Rudolfs	Vieweg	DM 10,-
Mikrocomputer Aufbaukurs Turbo Pascal	Kaier/Rudolfs	Vieweg	DM 20,-
Turbo Pascal 3.0 QUICKMANUAL	L.Lessner	MERKUR	DM 12,-
Wordstar Tuning	Werner Bonsbach	HEISE	DM 10,-
Hardware-Erweiterungen für Z80-Rechner	Martin Aschoff	Vogel	DM 12,-
Microsoft-Basic: Konzepte,Algorithmen, Datenstrukturen	Röckrath	Franzis	DM 12,-

Gesamt-Preis

DM 250,-

F&S VERTRIEB
computer division

Ihr Partner für
Hard- und Software

Telefon 05042 1253
05042 4215

An
Club 80
Herrn Schmid
St. Cajetan Str. 38
W-8000 Muenchen 80

20. Oktober 1990

SONDERRABATT FÜR COMPUTERCLUB-MITGLIEDER

Sehr geehrte Clubmitglieder,

Aus aktuellem Anlaß machen wir Ihnen als Computerclub ein Angebot:

Wir wollen Ihr Clublieferant werden !

Wir haben Ihnen eine Auswahl aus unserer derzeit gültigen Preisliste für Hard- und Software zusammengestellt, die wir aufgrund der hohen Qualität und des guten Preis-Leistungsverhältnisses empfehlen möchten.
Für Sammelbestellungen aus dieser Preisliste gewähren wir Ihnen

5% SONDERRABATT !

Sammelbestellungen sind schriftliche Bestellungen von drei Verpackungseinheiten oder mehr. Dieses Angebot ist bis auf Widerruf gültig.

Auf gute Zusammenarbeit

Fiedler & Schuster
-computer division-

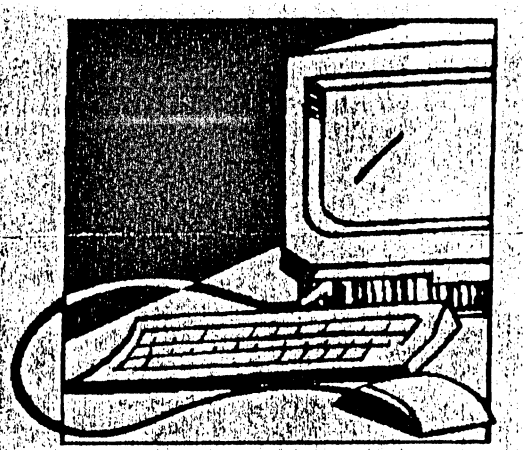
P.S. Sollten Sie einen Artikel auf unserer Preisliste vermissen, schreiben Sie uns. Wir besorgen Ihnen jeden Artikel zu dem günstigen Preis-Leistungsverhältnis, für das wir bekannt sind.

100 DISKETTEN

	NONAME	BASF EXTRA
5.25" 2 D 360 KB	DM 69,-	159,-
5.25" HD 1.2 MB	DM 139,-	289,-
3.5" 2 D 720 KB	DM 139,-	289,-
3.5" HD 1.44 MB	DM 299,-	599,-

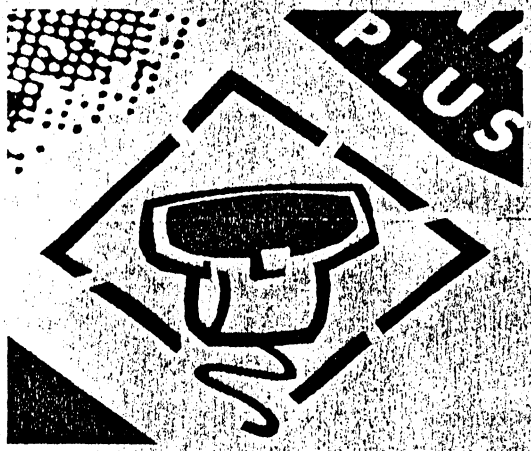
10 BASF CARTRIDGES

DC 300 L	DM 439,-
DC 450 L	DM 475,-
DC 450 LS	DM 489,-
DC 600 H	DM 539,-
DC 600 HS	DM 629,-
DC 2000 L	DM 489,-



5 DRUCKER FARBBÄNDER

Epson FX80 (G635) / STAR LC 10	DM 49,-
Epson LQ400 /LQ500/LQ550 (G633)	DM 69,-
NEC P6+/P7+	DM 79,-
NEC P2 / P6 (G666)	DM 79,-



MÄUSE + SCANNER

SCANMAN PLUS DM 419,-
mit PAINTSHOW PLUS! Deutsche Originalversion!

LOGIMOUSE S9 DM 199,-
Hochauflösende Profi - Maus mit Software POP-UP DOS und MOUSEWARE . Als BUS, SERIELL oder PS/2 Version . Auflösung 320 dpi!

LOGIMOUSE PILOT DM 119,-
Profi - Maus mit Software PAINTSHOW PLUS und MOUSEWARE in deutsch ! Serielle Version für den normalen und einfachen Anschluß an Ihren Rechner.Auflösung 200 dpi!

LOGITECH TRACKMAN DM 299,-
Hochauflösende stationäre Profi - Maus mit Hochleistungssoftware . Auflösung 50 bis 15 000 dpi!

LOGI FINESSE VERSION 3.1 DM 499,-
Das Desktop Publishing Programm, mit dem dieses Blatt erstellt wurde!

Sonderpreisliste gültig ab 01. Oktober 1990
Kurzfristige Preisänderungen müssen vorbehalten bleiben. Lieferung an Einzelpersonen nur gegen Vorkasse oder Nachnahme. Versandkosten werden extra berechnet. Bitte bestellen Sie nur schriftlich, Postkarte oder kurze Mitteilung genügt.

Fiedler & Schuster GbR
HARD u. SOFTWARE Vertrieb
POSTFACH 1116
3252 BAD MÜNDER 1

COMPUTER

PT-AT TOP

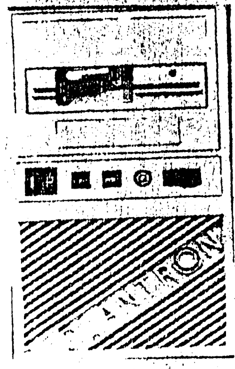
- 80286 CPU
- 8/12 MHz (0 Wait State)
- 1 MB RAM
- VGA Grafikkarte 16 Bit, 800 x 600 Punkte
- 1 parallele 2 serielle Schnittstellen, Game Port
- Floppy-/Harddiskcontroller 1:1
- Diskettenlaufwerk 1.2 MB, 5.25"
- Festplattenlaufwerk 42 MB 28 ms
- Große deutsche Tastatur

NUR DM 2899,-

PT-386 SX/20-1 Tower

- 80386/SX CPU
- 10/20 MHz (0 Wait State)
- 1 MB RAM Neat
- VGA Grafikkarte 16 Bit, 800 x 600 Punkte
- 1 parallele 2 serielle Schnittstellen, Game Port
- Floppy-/Harddiskcontroller 1:1
- Diskettenlaufwerk 1.2 MB, 5.25"
- Festplattenlaufwerk 42 MB 28 ms
- Große deutsche Tastatur

NUR DM 3898,-



PT386 25 CACHE Tower

- 80386-25 CPU
- 25 MHz
- 2 MB RAM
- VGA Grafikkarte 16 Bit, 800 x 600 Punkte
- 1 parallele 2 serielle Schnittstellen, Game Port
- Floppy-/Harddiskcontroller 1:1
- Diskettenlaufwerk 1.2 MB, 5.25"
- Diskettenlaufwerk 1.44 MB, 3.5"
- Festplattenlaufwerk 65 MB 28 ms
- Große deutsche Tastatur

NUR DM 6898,-

PT386 33 CACHE Tower

- 80386-33 CPU
- 33 MHz
- 2 MB RAM
- VGA Grafikkarte 16 Bit, 800 x 600 Punkte
- 1 parallele 2 serielle Schnittstellen, Game Port
- Floppy-/Harddiskcontroller 1:1
- Diskettenlaufwerk 1.2 MB, 5.25"
- Diskettenlaufwerk 1.44 MB, 3.5"
- Festplattenlaufwerk 65 MB 28 ms
- Große deutsche Tastatur

NUR DM 8498,-

386-16 SX Tower AT

- 80386-16 SX CPU
- 12/16 MHz
- 1 MB RAM (0 Wait State)
- 4x16 Bit und 1x8 Bit Erweiterungsslots
- Monochrom-Grafikkarte, Herkules-kompatibel
- 2 parallele Schnittstelle, 2 serielle Schnittstellen
- Marken Diskettenlaufwerk 1.2 MB, 5.25"
- Marken Diskettenlaufwerk 1.44 MB, 3.5"
- FDD/HDD Controller
- MF-II kompatible Tastatur mit 102 Tasten

NUR DM 2799,-

DISKETTENLAUFWERKE

5.25"	360KB	179,-
5.25" HD	1.2MB	199,-
3.5"	720KB	189,-
3.5" HD	1.4MB, mit Rahmen	199,-

FESTPLATTEN

VOM MARKTFÜHRER SEAGATE

ST 225	21.4 MB		409,-
ST 238R	32.7 MB	70 ms	499,-
ST 251-1	42.8 MB	28 ms	699,-
ST 296N	85 MB	28 ms	999,-
ST 124	21.4 MB	40 ms	649,-
ST 125-1	21.4 MB	28 ms	669,-
ST 157N	48.8 MB	28 ms	799,-
ST 151	43.5 MB	28 ms	799,-

DRUCKER

NEC PINWRITER P6 PLUS	1399,-
NEC PINWRITER P7 PLUS	1599,-
CITIZEN SWIFT 24	899,-
HP LASERJET II P	2999,-
HP LASERJET III	4899,-

MONITORE

14" TTL FLAT MONITOR AMBER	299,-
14" TTL FLAT MONITOR PAPERWHITE	299,-
14" VGA MONO MONITOR	399,-
14" VGA COLOR MONITOR	799,-
14" NN MULTISCAN MONITOR	999,-
14" NEC MULTISYNC 3D	1799,-
14" NEC MULTISYNC 2A	1299,-

GRAFIKKARTEN

KOMPATIBEL ZU IBM'S CGA,MGA,EGA,VGA

VGA KARTE 256KB		199,-
JUKO VGA PREMIUM	16 BIT 256KB	299,-
JUKO VGA PREMIUM II	16 BIT 512KB	399,-
TVGA VGA TRIDENT	16BIT 512KB	299,-

Alle Preise sind in DM und enthalten die gesetzl. Mehrwertsteuer. Versandkosten werden extra in Rechnung gestellt.

Kurzfristige Preisänderungen müßen leider vorbehalten bleiben. Lieferung solange Vorrat reicht.

EMERSON ACCUCARD

Notstromversorgung für jeden XT/AT
286-386-486 oder kompatiblen.

2/3-lange 8-Bit Steckkarte, die bei einem Stromausfall automatisch ein Backup fährt und alle Daten, die im Augenblick genutzt werden, auf die Festplatte sichert, und dann das System abschaltet. Kommt der Strom wieder, kommen auch Ihre Daten wieder. Nach der normalen Startprozedur des Rechners wird automatisch ein Restore gefahren und das System nimmt seinen Dienst an exakt der Stelle wieder auf, wo es zuvor durch den Stromausfall unterbrochen wurde!

Stück DM 999,-

Der Laserdrucker zum Preis eines Matrixdruckers!

Qume CrystalPrint WP Plus

das preiswerte Einstiegsmodell - entwickelt für die Anwender, die einen kleinen und leisen Arbeitsplatz-Drucker benötigen - zu einem Preis, der trotz perfekter Schriftqualität und Grafikfähigkeit in einer vernünftigen Relation zum Rechner steht. Durch Flüssigkristall-(LCS) Technologie beträgt die Stellfläche der Drucker nur noch ca 60 % derer von normalen Laserdruckern. Auch die Geräuschentwicklung liegt niedriger.

Crystal-Print WP Plus

1999,- Nur DM

286-12 Desktop

Intel 80286-12 mit 6 und 12 MHz (Landmark 16), 1 MB RAM bestückt, onboard erweiterbar bis 4 MB RAM EMS-kompatibel, Suntac Chip-Set, extended Bios mit Set-Up, 1 Floppydisklaufwerk 1.2 MB (Markenfabrikat), 1 Floppydisklaufwerk 1.44 MB (Markenfabrikat), FDD/HDD Kombi-Controller 1:1 mit Kabelsatz, Echtzeituhr, Schnittstellen: 1 x parallel, 1 x seriell und Gameport, Monochrom-Grafikkarte mit Printerport, 6 x 16 Bit und 2 x 8 Bit Erweiterungsslots, 200 Watt Schaltnetzteil, Qualitätstastatur KT 2000 mit 102 Tasten, verwindungssteifes Desktopgehäuse, Reset- und Turboschalter, Speed-Display, Ringbuchschuber, 80seitiges deutsches Handbuch, ZZ-Zulassung, CETERA DOS 3.41, lizenzierte Version des aktuellen DR DOS 3.41, komplett mit deutschen Handbüchern, wahlweise auch auf 3.5" FD, inklusive Norton Commander 2.01.

PREIS NUR 1599,-

Neue Drucker braucht das Land

Nachdem mir mein RX-80 seit 6 Jahren treue Dienste geleistet hat und noch immer keine Neigung zeigt, kaputt zu gehen, habe ich ihn nun zwangsbeurlaubt und mir einen HP Deskjet 500 gekauft. Warum nun ausgerechnet den ?
Mein Forderungskatalog sah folgendermaßen aus:

- 24-Nadleln (wenn schon, denn schon)
- automatischer Einzelblatteinzug (muß einfach sein, s.u.)
- leise (Eltern schlafen nebenan)
- gute Schriftqualität (fast selbstverständlich)
- gute Verarbeitung (auch klar)
- nach Möglichkeit Epson-kompatibel (Tscrips ade)

Also bin ich von Geschäft zu Geschäft gepilgert und habe mich umgehört. 24-Nadler gibt es zwar schon ab rund 500 Mark, aber deren Verarbeitung und auch die Einzelblatteinzüge sahen nicht besonders vertrauenerwechend aus und wenn man dauernd einen Papierstau hat, kann man gleich beim Handbetrieb bleiben. Warum überhaupt einen automatischen Einzug ? Ganz einfach, weil mir die Einspannerei per Hand im Laufe der Zeit gewaltig auf die Nerven gegangen ist. Wenn man ein File mit 50 Seiten ausdruckt und nicht weggehen kann, weil man jede Minute eine neue Seite nachlegen muß ist das einfach lästig. Der Einzelblatteinzug ist beim HP übrigens schon fest eingebaut, sodaß die Konstruktion solide und sicher ist. Verarbeitet werden zwar nur A4-Blätter (die per Befehl auch quer bedruckt werden können !), aber ich weiß schon gar nicht mehr, wann ich das letzte mal auf Endlospapier gedruckt habe. Briefumschläge schluckt er, nebenbei bemerkt, auch.
Die nächste Frage war, ob Nadeln oder Tintenspritzer. Von der Qualität her geben die sich nicht viel, obwohl beim HP auch im Draft-Modus keine Sägezähne, wie bei allen Nadeldruckern, die ich kenne, im Schriftbild zu erkennen sind. Der Ausdruck ist lediglich etwas heller und auf Recyclingpapier sogar besser zu lesen, weil die Schrift im Schönschriftmodus dort wegen der größeren Tintenmenge und der rauheren Oberfläche fast verläuft. Auch braucht man längst kein spezielles Tintendruckpapier mehr, normales Kopierpapier tut es genauso. Ein großer Vorteil ist aber einmal die Geräuscharmut und zweitens, daß einem keine Nadeln abbrechen oder sich verbiegen können. Wenn die Tintenpartone leer ist wird sie mitsamt dem Druckkopf wegeworfen und man hat wieder einen völlig Neuen. Im Gegensatz zu anderen Druckern, z.B. von Epson, wird nichts nachgefüllt, sodaß auch keine Schläuche verstopfen können. Bei den billigeren Nadeldruckern kostet ein neuer Kopf mindestens die Hälfte des Neupreises und ob sich das dann bezahlt macht, ist eine andere Frage. Da es kein Farbband gibt, daß immer mehr ausgelutscht wird, bleibt die Druckqualität natürlich bis zum Ende immer gleichmäßig und gut.
Ein besonderes Bonbon, womit schließlich auch der letzte Punkt erfüllt wäre, ist die Epson-Emulations-Kassette, die einfach in einen der beiden Erweiterungsschächte gesteckt wird. Mit dieser Kassette emuliert der HP einen FX-80 und zwar so gut, daß man keine weißen Streifen im Grafik-Ausdruck hat ! Ich habe es selber ausprobiert, es funktioniert. So kann man alle alten Programme ohne irgendwelche Softwarekrücken (Arnulf, nimm's bitte nicht persönlich) weiterverwenden, die sich nicht oder nur schwer an andere Drucker anpassen lassen (TSCRIPS usw.). Der HP selber ist zum LaserJet kompatibel, für den sich in neueren Programmen mit Sicherheit ein Treiber findet. Bis auf die Tatsache, daß er wegen des fehlenden RAMs keine ganzen Grafikseiten im Speicher aufbauen kann, kann er den Laserdrucker vollständig ersetzen, sogar die Auflösung mit 300 dpi ist identisch.
Im Prinzip könnte man sich dann ja gleich einen Laser kaufen, die billigsten gibt es schon für knapp 2000 Mark, aber bei denen sind die Verbrauchskosten nochmal doppelt so hoch, wie beim Tintenspritzer und der kostet schon rund das doppelte eines Nadeldruckers.
Zusätzlich zu den eingebauten drei Schriftarten gibt es natürlich noch Schriftkassetten, man kann aber auch Fonts downloaden. Features wie Proportionalschrift usw. sind ja sowieso selbstverständlich.
Leider hat auch der beste Drucker seine Schattenseiten, das 'Handbuch' ist nämlich nur eine bessere Referenzkarte. Scheinbar geht HP davon aus, daß der moderne DOS-Geschädigte den Drucker sowieso nur aus den diversen Softwarepaketen heraus per Menü anspricht und nichts selber machen will oder kann. Beispiele sucht man weitgehend vergebens und gerade die Grafik wird sehr stiefmütterlich behandelt.
Wenn Ihr Euch den HP in einem Geschäft mal anseht, paßt auf, daß man Euch den richtigen zeigt. Letztes Jahr im Oktober war erst ein Test des HP DeskJet in der c't, etwas später kam der DeskJet Plus und seit kurzem gibt es schließlich den DeskJet 500. Ich glaube zwar nicht, daß es bis zum Erscheinen des Infos schon wieder einen neuen gibt, aber möglich ist alles. Darum Vorsicht, mir hat man auch versucht, noch den Plus anzudrehen, obwohl der nur eine eingebaute Schrift hat und sogar teurer ist, als der 500. Ich habe für den Drucker 1450.- gezahlt und die Epson-Emulation kommt auf 200.- Mark.

Alexander Schmid

DFÜ — was bringts?

Vor einigen Monaten bin ich, warum eigentlich weiß ich heute selbst nicht mehr zu sagen, in die Übertragung von Daten über Telefon eingestiegen. Am Anfang war DFÜ für mich Selbstzweck, sprich ich betrieb die Datenübertragung der Datenübertragung wegen. Mittlerweile hat sich das aber erheblich geändert. Inzwischen ist dieser Nebenzweig der Computerei aus meinem Tagesablauf gar nicht mehr wegzudenken. Heute leere ich mein elektronisches "Postfach" in der Mailbox genauso selbstverständlich, wie meinen Briefkasten und mein Postfach im Postamt und nutze die vielfältigen Möglichkeiten intensiv.

Dabei beschränken sich meine Aktivitäten (zur Zeit noch) nur auf den "Besuch" von Mailboxen und den Gastzugang zu BTX. Die Möglichkeiten von DATEX-P sind mir zur Zeit noch aus finanziellen Gründen verschlossen. Demnächst werde ich hoffentlich auch vollwertiger FIDO*-Teilnehmer sein, zur Zeit funktioniert aber mein Mailerprogramm noch nicht richtig!

Mailboxen bieten aber heutzutage schon so viel, daß man durchaus auch auf BTX (geht eh nur mit C64, ATARI, PC oder speziellen BTX-Geräten), DATEX-P und FIDO (bzw. andere Netze) verzichten kann. Neben den öffentlichen Brettern (PIN's genannt) und dem privaten Postfach gibt es vor allem Downloadareas für viele verschiedene Rechner.

Um euch einmal eine Übersicht über die Vielfalt der öffentlichen PIN's zu geben, hier eine Liste aus der BRAINSTORM, meiner "Heimatbox":

BUECHER.PIN	BUECHER	POP.PIN	POP-MUSIC
ANDY_AMI.PIN	SW-TIPS AMIGA	VIREN.PIN	VIREN
HACKER.PIN	NICE HACKING	UMWELT.PIN	UMWELT
KONTAKTE.PIN	KONTAKTE	ESOTERIK.PIN	ESOTERIK
DFUE.PIN	DFUE/FIDO	ANDY_AND.PIN	SW-TIPS DIVERSE
ANDY_MS.PIN	SW-TIPS MS-DOS	DATEX_P.PIN	DATEX-P
MS_DOS.PIN	MS-DOS ECKE	NEWSYSOP.PIN	NEUES VOM SYSOP
POLITINL.PIN	POLITIK	NUTS.PIN	- NUTS -
SOFTWARE.PIN	SOFTWARE-TIPS	KASSE.PIN	KASSE BRAINSTORM
ALLGMAIL.PIN	ALLGEMEINE MAIL	SUCHE.PIN	SUCHE/TAUSCHE
ATARI.PIN	RUND UM ATARI	TIERVERS.PIN	TIERVERSUCHE
TIPSBAU.PIN	HARDWARE TIPS/BAUANLEITUNGEN	WITZE.PIN	LACH-HAFT
DX.PIN	DX-PIN	MECKER.PIN	MECKERECKE
ADALLG.PIN	ALLERLEI/ADVENTURES	GREENPEA.PIN	GREENPEACE
AMIGA.PIN	AMIGA ALLGEMEIN	HEAVY.PIN	HEAVY NEWS/VORST.
CD-EXTRA.PIN	CD CHARTS TOP 15	PDIVERS.PIN	ALLERLEI (P)
LABER.PIN	- RABARBERSUELZ -	GERUECHT.PIN	GERUECHTE-KUECHE
PSTORY.PIN	GESCHICHTEN FÜR ERWACHSENE	VERKAUF.PIN	VERK./VERSCH.
MVA.PIN	MUELLVERBRENNUNGSANLAGEN		

Wie jeder echte Z80-Freak sofort feststellen wird, gibt es tatsächlich eine Menge interessanter Sachen, aber nichts für sein 8-Bit-Herz. Wie mir aber erst gestern Gerd Neebe am Telefon sagte, gibt es in vielen Mailboxen noch CP/M-Bretter. Meine Wahl fiel nur deshalb auf die BRAINSTORM, da sie in meinem Ortsbereich liegt und damit im 8- bzw. 12-Minutentakt zu erreichen ist. Mit den Downloadbereichen, aus denen man oft sehr gute Public Domain Programme herauskopieren kann, sieht es ähnlich aus. Fast in allen Boxen findet man Programme für MSDOS, ATARI und C64, aber auch CP/M-Programme gibt es noch häufiger, als man sich das vorstellt!

Daß durchaus auch nicht computerspezifische Informationen über Mailboxen laufen, kann man an der oben abgedruckten PIN-Liste schon sehen. Daß es aber auch Bereiche gibt, die man nicht (oder doch zumindest ich nicht) in Mailboxen vermuten würde, zeigt die Box "Lebensinterface" in Aalen. Dort gibt es unter anderem eine Bibeldatenbank "Neues Testament" mit folgenden Unterpunkten:

*FIDONet ist ein privates, internationales Rechnernetzwerk, das derzeit (März '90) aus etwas mehr als 5000 Rechnern ("Nodes") besteht.

‹1› Matthäus - Evangelium	‹2› Markus - Evangelium	‹3› Lukas - Evangelium
‹4› Johannes - Evangelium	‹5› Apostelgeschichte	‹6› Römer
‹7› 1. Korinther	‹8› 2. Korinther	‹9› Galater
‹A› Epheser	‹B› Philipper	‹C› Kolosser
‹D› 1. Thessalonicher	‹E› 2. Thessalonicher	‹F› 1. Timotheus
‹G› 2. Timotheus	‹H› Titus	‹I› Philemon
‹K› 1. Petrus	‹L› 2. Petrus	‹M› 1. Johannes
‹N› 2. Johannes	‹O› 3. Johannes	‹P› Hebräer
‹Q› Jakobus	‹R› Judas	‹S› Offenbarung

Wie ihr seht, ist das Angebot vielfältig, teilweise sogar unüberschaubar, auf jeden Fall aber sehr interessant!

Für mich ist, ich habe es oben schon erwähnt, der Bereich der Übertragung "privater" Post von großer Wichtigkeit. Als kleines Beispiel hier eine Nachricht, die mir Hans Günther Hartmann kürzlich zukommen lies:

MsgNr: 1
*MB*1002/HANS-G HARTMANN am 23.11.90 um 09:13:50*

Betreff: zcpr3 config fuer 512k RAM

Moin Hartmut.
Ich habe dir in die priv softbiblio eine zip-Datei gelegt.
Hoffentlich kannst Du sie entzippen!
Wenn nicht, kurze msg. Falls Du den Fehler in den dateien findest,
die das Booten auf dem 4p verhindern------> auch kurze msg.
Gruss
Hans-G, UID 1002

Wie man der Nachricht entnehmen kann, ist es auch möglich, eigene Programme oder Daten ohne Zugang für andere Mailboxnutzer zu übertragen. Solche Nachrichten, offline geschrieben (d.h. mit einem Texteditor erstellt, bevor man die Mailbox anruft) und dann übertragen, sind oft billiger und auf jeden Fall schneller als Briefpost.

Für den Zugang zu einer Mailbox braucht man, neben dem Computer, ein Modem (direkt ans Telefonnetz anschließbar) oder einen Akustikkoppler ("Anschluß" über den Telefonhörer) und ein Datenübertragungsprogramm. Dabei würde ich, wegen der besseren Übertragungssicherheit, ein Modem jederzeit einem Akustikkoppler vorziehen. Leider gibt es zur Zeit prinzipiell nur zwei Alternativen beim Modem-Kauf. Entweder illegal (weil ohne Postzulassung) und billig (um ca. 250-300 DM) oder legal und doppelt bis dreifach so teuer. Zum Glück gibt es unter CP/M wenigstens genug Übertragungsprogramme zum Nulltarif (z.B. KERMIT und MODEM7). Bei der Wahl des Terminalprogramms sollte man aber darauf achten, daß das Programm wenigstens das XMODEM-Protokoll (Übertragungsprotokoll) beherrscht. Dies ist bei einigen älteren Programmen unter NewDOS nicht der Fall!

Für alle die, die schon DFÜ betreiben und all jene, denen ich den Mund wässrig gemacht habe und die demnächst in die Datenfernübertragung einsteigen, hier die Nummer und die Parameter der Mailbox, unter der ich zu erreichen bin: BRAINSTORM 07309/6753, 8N1, 300/1200/2400 User-ID: 1061.

Ich würde mich sehr freuen, wenn ich in meinem Postfach mal eine Message von euch finden würde.

In diesem Sinne allzeit Good Hacking,

Hartmut Obermann

PS: Übrigens gibt es in unseren Reihen sogar einen Mailbox-Betreiber. Klaus Hermann ist SYSOP der DISCOVERY, die unter der Nummer 07127/70107 mit den Parametern 300/1200/2400 N81 zu erreichen ist.

Befehlssatz Hayes-kompatibler Modeme

Der Begriff 'Hayes' ist schnell erklärt. Es handelt sich dabei um einen amerikanischen Modemhersteller, dessen 'Smartmodem' mittlerweile als das Referenzgerät für den Hayes-Befehlssatz gilt. Es ist heute nahezu undenkbar, daß auf dem internationalen Markt ein Modem hergestellt wird, welches nicht diesen De-facto-Standard erfüllt.

Daß der Hayes-Befehlssatz ein Industriestandard wurde, hat seine Ursache in dem einfachen Konzept, das hinter der Modemansteuerung steht. Völlig unabhängig reagiert das Modem auf einfache ASCII-Zeichen, die über die serielle Schnittstelle an das Modem gelangen. Das Modem verfügt über einen Prozessor, der diese Zeichen in Befehle umwandelt. So lassen sich auf einfachste Art Kommunikationsprogramme schreiben. Ein weiterer Grund für die Etablierung des Standards ist die Fähigkeit des Modems, sich selbständig auf die Parameter eines Anrufers einstellen zu können.

Hier nun kurz die Beschreibung der wichtigsten Befehle:

'D' - Dial
Das Modem wählt eine Nummer. Dabei gibt es zwei Spielarten:

- 'DP' Pulse-Dialing (in Europa üblich)

- 'DT' Touch-Tone-Dialing (in den USA üblich)

Beide Wählarten können auch gemischt werden. Der Befehl 'ATDT0,P01188' würde beispielsweise mit Touch-Tone eine Null wählen (Amtsleitung anfordern), kurz warten und danach mit Pulse-Dialing die Fernsprechauskunft anrufen.

'A/' - Wiederholung
Das Modem wiederholt die letzte Funktion. Häufigste Anwendung ist, die letzte Nummer noch einmal zu wählen.

'H0' - Auflegen
Die Verbindung wird abgebrochen, der 'Hörer aufgelegt'.

'H1'- Abheben
Der 'Hörer' wird 'abgehoben', die Leitung erscheint nach außen hin besetzt.

'A' - Antworten
Das Modem sendet einen 'Antwortton' aus. Es versucht eine Verbindung aufzubauen, ohne vorher zu wählen. Nach diesem Befehl befindet man sich nicht mehr im Befehlsmodus, den erreicht man erst wieder über die 'Escape'-Sequenz.

'S0=<n>' - Automatisch antworten
Dieser Befehl schaltet den Automatisch-antworten-Modus ein und aus. Bei n=0 antwortet das Modem nicht auf ankommende Anrufe, jede andere Zahl zwischen 1 und 255 legt fest, beim wievielten Klingeln das Modem 'abhebt' und antwortet.

'+++' - Escape
Nachdem man eine Verbindung aufgebaut hat, befindet sich das Modem nicht mehr im Befehlsmodus. Alle Zeichen, die man nun an das Modem schickt, leitet dieses ins Telefonnetz weiter. Empfängt es jedoch nach einer Sekunde Pause die drei Pluszeichen und danach wiederum für eine Sekunde kein Zeichen, schaltet es sofort in den Befehlsmodus zurück. Nun kann man wieder mit 'AT' Parameter ändern oder einfach den 'Hörer auflegen'.

Die restlichen Befehle und Register-Funktionen sind in der folgenden Tabelle zusammengefasst:

Hayes-Modem-Befehle:

AT - Beginn eines Hayes-Befehls
+++ - Zurück in den Kommandomodus (vor und nach dem Befehl muß eine Sekunde gewartet werden)
A/ - Automatische Wiederholung des letzten Befehls
D - Wählen › Ziffern: 0 - 9
› Symbole: # und * (zur übersichtlichen Gestaltung der Nummer)
› Befehle: P Pulse-Dialing (Europa)
T Touch-Tone-Dialing (USA)
, 2 Sekunden Wählpause
/ 125 ms Wählpause
; nach dem Wählen in den Befehlsmodus
R ein Originate-Modem anrufen
A - Beantworten eines Anrufs ohne zu wählen
B - B0 = CCITT V.21, V.22 (Europa)
B1 = Bell 103/212A (USA)
C - C0 = Carrier-Signal aus
C1 = Carrier-Signal an
E - E0 = Echo aus
E1 = Echo an
F - F0 = Halb-Duplex
F1 = Voll-Duplex
H - H0 = Modem legt auf
H1 = Modem hebt ab
I - I0 = Ausgabe des Produktcodes
I1 = Ausgabe der Firmware-Version
I2 = Selbsttestfunktion
M - M0 = Lautsprecher aus
M1 = Lautsprecher an, bis Carrier detect
M2 = Lautsprecher immer an
O - zurück in Online-Modus
Q - Q0 = Sende Modem-Antwort an Rechner
Q1 = keine Antwort Senden
Sr? - Zeige Wert von Register r
Sr=n - Setze Register r auf Wert n
V - V0 = Numerische Antworten
V1 = Ausgeschriebene Antworten
X - X0 = Grundantworten (OK und CONNECT)
X1 = Antworten (CONNECT <Baudrate>)
X2 = Erkenne "NO DIALTONE"
X3 = Erkenne "BUSY"
X4 = Erkenne "NO DIALTONE" und "BUSY"
Z - Software-Reset

Hayes-Register-Funktionen:

S0 - Anzahl der Klingelzeichen vor dem Abheben (Parameter: 0-255)
S1 - Zähler für die Klingelzeichen (Parameter: 0-255)
S2 - ESCAPE-Zeichen (Parameter: 0-127 ASCII)
S3 - RETURN-Zeichen (Parameter: 0-127 ASCII)
S4 - LINE-FEED-Zeichen (Parameter: 0-127 ASCII)
S5 - BACKSPACE-Zeichen (Parameter: 0-127 ASCII)
S6 - Warte auf Freizeichen (Parameter: 2-255 Sekunden)
S7 - Wartezeit auf einen Carrier nach Wahl oder Antwort
S8 - Pausenzeit für ein Komma (Parameter: 0-255 Sekunden)
S9 - Carrier-Erkennungszeit (Parameter: 1-255 in 1/10 Sekunden)
S10 - Wartezeit zwischen Carrier-Ausfall und dem Auflegen (Parameter: 1-255 in 1/10 Sekunden)
S12 - Wartezeit auf ein ESCAPE-Zeichen (Parameter: 0-255 in 1/10 Sekunden
S16 - Selbsttest des Modems (Parameter: 0-2)

Wer sich etwas mit hayeskompatiblen Modemen auskennt, dem wird auffallen, daß die obrige Liste nicht ganz komplett ist. Das hat seinen Grund darin, daß es inzwischen viele Modeme gibt, die gegenüber dem ursprünglichen 'Smartmodem' von Hayes viele erweiterte Funktionen haben. Vor allem das Kommando 'B' ist bei vielen neueren Modemen erweitert, da sie zusätzliche Normen erfüllen (z.B. CCITT V.23 75/1200 Baud für BTX). Da diese erweiterten Kommandos jedoch nicht genormt sind, wurden nur die Funktionen aufgeführt, die in allen hayeskompatiblen Modemen vorhanden sind.

Der Text und die Liste basieren auf einem Artikel aus der c't 2/88 und erheben nicht den Anspruch der Vollständigkeit! Der Text wurde mehrfach korrigiert und vor allem auf seine sachliche Richtigkeit überprüft. Sollte sich trotzdem ein Tipppfeelrteuffel eingeschlichen haben, bitte ich um Entschuldigung.

Hartmut Obermann

PS: Diesen Text habe ich als Antwort auf eine Anfrage in der BRAINSTORM geschrieben und dort als Textfile in der Downloadarea DFUE abgelegt. Inzwischen habe ich den Text schon in mehreren anderen Boxen im süddeutschen Raum wiedergefunden, und die Verbreitung wird wohl noch ein bißchen weitergehen, bevor das Thema uninteressant geworden ist (was natürlich in meinem Sinne ist).

Die beiden Terminalprogramme MODEM7 und KERMIT, die es unter CP/M als PD-Software gibt, können beide mit hayeskompatiblen Modemen zusammenarbeiten. Die Geräte gibt es, allerdings ohne FTZ-Prüfnummer und damit zum Betrieb am Netz des Gilb (TELECOM) nicht zugelassen (Anschluß strafbar!), schon ab ca. 250,-- DM.

Adventure

Computerspiel mit vorweihnachtlicher Spielidee. Meist geht es bei dem besinnlichen Game um die aufregende Suche nach einem Schatz und die brutale Vernichtung von möglichst vielen Gegnern – ganz wie bei dem letzten verkaufsoffenen Samstag vor Heiligabend.

AT

Ein durch vielerlei kostspieligen Schnickschnack Ausgebauter Taschenrechner.

Newdos, CP/M, MS-DOS, UNIX

von Gerald Schröder

Nun habe ich Euch lange genug mit (All-)Gemeinheiten über Betriebssysteme gequält. Also jetzt mal was Konkretes. Heute werde ich versuchen, die vier oben genannten Betriebssysteme vorzustellen. Für die meisten von Euch sind das natürlich alte Hüte, aber vielleicht kennt ja nicht jeder alle Systeme. Außerdem werde ich sie aus meiner Sicht vergleichen. "Meine Sicht" bedeutet: Nicht als Anwender, sondern als Programmierer, und besonders als notorischer Nörgler.

[Wer im folgenden TOS (Atari ST) vermißt, sollte einfach "MS-DOS" durch "TOS" ersetzen (und "IBM" durch "Atari"), schon stimmt es (fast) wieder. Wer AmigaDOS (Commodore) oder ähnlich Exoten vermißt, möge Nachsicht mit mir haben, denn alles kenne ich nun auch nicht.]

Fangen wir mit unseren altbekannten BSen an: NEWDOS und CP/M. Beide wurden für Computer mit Z80- (bzw. 8080-) Prozessoren entworfen, womit ein Haupt-Parameter festgelegt war: 8-Bit-Prozessor mit 16-Bit-Adressen = 64 Kb Speicher (RAM und/oder ROM). Damit hören aber schon die Gemeinsamkeiten auf.

NEWDOS

Beim Entwurf von NEWDOS (bzw. dessen Vorläufer TRSDOS) gab es feste Vorgaben: Das BS sollte nur auf einem bestimmten Computer laufen, dem TRS-80. Dieser ist mit einem 12 Kb großen ROM(-BASIC) ausgestattet, das bei 0000h anfängt. Daran kann das BS schon mal nichts ändern. Um den Platz aber nicht ganz zu verschwenden, sollten natürlich möglichst viele ROM-Routinen benutzt werden. Dann liegen irgendwo mitten im Adreßraum (bei 3xxxh) alle Verbindungen zu den Peripherie-Geräten ("Memory Map"), als da wären: Tastatur, Bildschirm, Disk(-Controller) und Drucker(-Interface). Da kann das BS auch nichts dran ändern, aber es darf sich darauf verlassen, daß die Sachen bei jedem TRS-80 dort an der selben Stelle stehen und immer gleich behandelt werden.

Darüber ist dann endlich RAM angesagt, ab 4000h aufwärts, also 48 Kb. Hier muß sich das BS einnisten, aber nicht nur das BS, sondern auch die Anwender-Programme sollen noch reinpassen! Für die Anwender-Programme wurde der Platz ab 5200h (bis OFFFFh) vorgesehen, also muß unser NEWDOS mit genau 4,5 Kb auskommen, kein Stück mehr!

Eins war damals (scheinbar) klar: Der Computer würde sich nie mehr ändern. Also bliebe das BS immer unverändert und die Entwickler brauchten sich um "Änderungsfreundlichkeit" keine Gedanken zu machen. Aber sie hatten ein viel größeres Problem: In die 4,5 Kb läßt sich kein ordentliches BS pressen, also griffen sie in die Trickkiste und holten "Overlays" heraus. Das bedeutet schlicht: Ein (kleiner) Teil des BSs bleibt immer im Speicher (hier: SYS0/SYS), alle anderen Teile werden je nach Bedarf (von der Diskette) nachgeladen, und zwar immer an den gleichen Platz (übereinander). Das bedeutet auch, daß zwei Overlays nie gleichzeitig im Speicher stehen können. Es geht eine Menge Zeit (für das Suchen und Laden der Overlays) und eine Menge Platz (auf der Diskette) verloren.

Um von einem Overlay aus ein anderes (bzw. eine Routine daraus) aufrufen zu können, kommt natürlich ein einfacher Unterprogrammaufruf (CALL) nicht in Frage, sondern es ist komplizierter: Das rufende Overlay gibt dem residenten Teil (SYS0) bekannt, welches Overlay es mit welchem "Befehl" (z.B. "formatiere Diskette") aufrufen möchte, SYS0 lädt dann das entsprechende Overlay und gibt die Meldung weiter.

Moderne Betriebssysteme nennen so etwas gerne "virtuelle Speicherverwaltung", d.h. virtuell gibt es ein sehr großes BS, von dem aber immer nur ein kleiner Teil real im Speicher vorhanden ist. Allerdings reicht bei modernen Prozessoren doch wieder ein "CALL" aus, ohne das Overlay-Gedöns. Damit kann dann das BS (oder ein sehr großes Programm) doch ganz im Speicher gehalten werden, wenn der groß genug ist. Ansonsten wird doch wieder von Festplatte (wer arbeitet denn noch mit Disketten?) nachgeladen.

Um Mißverständnissen vorzubeugen: Mit "modern" ist sicherlich nicht der IBM PC (samt Kompatiblen) gemeint (siehe dazu auch weiter unten). Der kennt neben den Overlays noch ganz andere Krücken (Stichworte: "Expanded" und "Extended" Memory), um mit seinen Konstruktionsfehlern leben zu können.

Wie Ihr seht, ist NEWDOS also ein hochmodernes BS, das nur einen kleinen Fehler hat: Es ist total auf den TRS-80 zugeschnitten. In jedem Overlay wird garantiert irgendwas benutzt, was nur auf dem TRS-80 so geht. Das BS ist zwar schön modularisiert (durch die Overlays, die jedes für sich eine andere Aufgabe übernehmen), so daß Änderungen eigentlich leicht möglich sein sollten, aber leider gibt es dazu keine Dokumentation (mal abgesehen von Grossers "DOS-Buch"). Vor allem werden weder der Aufbau der Overlays bekanntgegeben noch die gegenseitigen Aufrufe erklärt. Sprich: Es fehlt ein "Systems Guide", wie es zu jeder CP/M-Version existiert, wo ziemlich genau die Anpassung des BS (also CP/M) an beliebige andere Z80-Rechner beschrieben und außerdem festgelegt wird, was auf jedem Rechner gleich sein muß (Einsprungadressen usw.).

Von der Seite der Anwenderprogramme aus, sieht die Sache so aus: Es gibt keine einheitliche Schnittstelle zwischen Anwenderprogramm und Betriebssystem (sprich: einen bestimmten Punkt, an dem das BS per "CALL" betreten werden kann), sondern in der Anleitung stehen gleich mehrere! Das hat zwei große Nachteile für den, der NEWDOS ändern möchte: a) Diese Adressen sind fest, hier darf also nie was anderes als vorgesehen passieren. b) Wo sollen neue (eigene) Funktionen eingefügt werden?

Natürlich sind auch alle Anwender-Programme so geschrieben: Sie sollten nur auf dem TRS-80 und nur mit diesem BS laufen. Wenn irgendein Bit nicht dem im Original-TRS-80/-NEWDOS entspricht, kann das fatale Folgen haben, denn es werden fröhlich alle (auch die undokumentierten) Eigenschaften des TRS-80 und des NEWDOS ausgenutzt.

Eine Kleinigkeit sollte ich noch erwähnen, auf die mich Hartmut hingewiesen hat: Im Newdos gibt es keine Trennung zwischen BS und Kommando-Interpreter! In allen folgenden Betriebssystemen werden alle Benutzer-Eingaben wie COPY, DIR usw. von einem Programm (genannt: CCP, Kommando-Interpreter, COMMAND.COM, Shell, o.ä.) entgegengenommen und dann in BS-Aufrufe umgewandelt. So kann die Oberfläche (der Kommando-Interpreter) leicht gegen eine andere (ein anderes Programm) ausgetauscht werden, z.B. gegen eine grafische Oberfläche wie GEM oder WINDOWS. Das geht bei Newdos nicht, weil dort eine enge Verzahnung von BS und Kommando-Interpreter besteht und der

Kommando-Interpreter kein eigenständiges Programm ist. Was übrigens das Newdos größer macht, als es eigentlich sein müßte. Dafür kennt der integrierte Kommando-Interpreter aber viel mächtigere Befehle als alle anderen mir bekannten (man siehe nur den COPY-Befehl mit den tausend Parametern!). Was aber dem Programmierer nicht viel bringt, wenn er ein Programm schreibt.

Um es noch einmal zusammenzufassen: Der TRS-80 ist eine vollkommen festgelegte Maschine und das NEWDOS ein genau darauf zugeschnittenes BS. Beide werden vom Anwenderprogramm als unveränderlich angenommen.

CP/M

Ganz anders die Ausgangssituation bei CP/M: Hier gibt es nicht einen Computer, auf dem das BS laufen soll. Nein, jeder Computer mit Z80-/8080-CPU sollte CP/M "fahren" können. Allerdings werden auch bei CP/M einige Vorgaben gemacht, die aber nicht von einem existierenden Computer vorgegeben wurden, sondern allein aus den Erfordernissen des BSs entspringen. Das zu lösende Problem hieß schlicht: Wie schaffen wir es, (Anwender-)Programme auf jedem Computer laufen zu lassen, ohne daß die Konfiguration des Rechners bekannt ist? "Konfiguration" soll hier heißen: Ausstattung des Rechners mit Speicher und Peripherie-Geräten wie Floppy-Controllern, Bildschirmen, Tastaturen usw.

Wie wir schon mal gehört haben, soll das BS die Verwaltung und das Ansprechen der Peripherie-Geräte für die Anwenderprogramme übernehmen, und zwar vollkommen. Beim TRS-80/NEWDOS kann das leicht umgangen werden, denn jeder weiß ja, wie die Maschine aussieht und kann selber daran rum"PEEKen" und -"POKEn" (was auch jeder tut). Beim CP/M sollte aber gerade das vermieden werden, denn jedes Programm soll ja auf jeder Maschine laufen. Also wurde genau festgelegt, was auf jeder Maschine gleich aussieht. Diese Dinge darf der Anwendungsprogrammierer benutzen, alles andere ist für ihn tabu! Moderner ausgedrückt: Das BS stellt dem Programm eine "virtuelle Maschine" (also nicht eine real existierende) zur Verfügung, deren Eigenschaften auf jedem Rechner die gleichen sind.

Unsere virtuelle Maschine sieht so aus: Das RAM muß bei 0000h beginnen. Anwenderprogramme werden ab 0100h geladen. Das CP/M steht irgendwo am oberen Ende des Speichers. Es ist nicht festgelegt, wieviel RAM zur Verfügung steht. Das Programm kann über einen Wert erfahren, wieviel RAM es benutzen darf (darüber gehört alles dem BS). Wenn das Programm irgendwas vom Betriebssystem will, muß es einige Register laden und dann "CALL 0005h" ausführen. Damit wird dann irgendwas vom BS erledigt, und zwar auf jedem Rechner.

Natürlich gehören noch einige Sachen mehr zu den Festlegungen, z.B. welche Register-Inhalte beim "CALL 0005h" denn nun was bedeuten und was man zurück bekommt. Das würde jetzt aber zu weit führen. Wichtig ist nur: Es gibt eine genau dokumentierte Schnittstelle zwischen Anwenderprogramm und BS. Das Programm darf sich nur auf die festgelegten Sachen verlassen und auf kein Stück mehr. Wie das BS die von ihm geforderten Tätigkeiten erledigt, ist allein seine Sache, und kann sich von Zeit zu Zeit bzw. von Rechner zu Rechner auch mal ändern. Dem Anwenderprogramm muß es vollkommen egal sein, auf welchem Rechner es läuft (d.h. es darf nichts voraussetzen, z.B. daß ein "PEEK (3880h)" die SHIFT-Taste abfragt).

Für das Anwenderprogramm ist damit die Sache klar. Aber was passiert, wenn wir einen vollkommen jungfräulichen Z80-Rechner gebaut oder gekauft haben und nun gerne unsere CP/M-Programme darauf laufen lassen möchten? Nun, ein CP/M muß her! Danach läuft alles.

Damit nicht für jeden Rechner ein vollkommen neues CP/M geschrieben werden muß, teilten die CP/M-Urväter das Ding in zwei Teile: Ein Teil, das BDOS, läuft auf jedem Rechner, ein anderer Teil, das BIOS, nur auf einem bestimmten. Nun muß für jeden neuen Rechner nur noch ein neues BIOS geschrieben werden. Und wenn sich an einem Rechner etwas ändert, z.B. eine Festplatte eingebaut wird, ändert man (wer immer das auch sein mag) nur noch einen Teil des BIOS. Das ganze Prinzip hatte ich ja schon mal erklärt.

Wichtig ist hier der Unterschied zum NEWDOS: Dort gibt es kein BIOS, das man mal kurz anpaßt, wenn eine Festplatte eingebaut wird. Entweder ändert man das ganze BS und läuft dabei Gefahr, daß einige Programme mit diesen Änderungen nicht zurechtkommen, oder läßt es bleiben. Im CP/M dagegen ist so etwas vollkommen unkritisch, denn es bekommt ja kein Programm mit, daß sich etwas geändert hat.

Aber natürlich hat auch ein CP/M seine Tücken: Beim Erstellen der Schnittstelle mußte ja schon bedacht werden, wie ein Rechner aussehen könnte bzw. was er alles kann. Also wurde die oben erwähnte "virtuelle Maschine" so entworfen, daß ihre Eigenschaften "irgendwie" auf die (damals) real existierende Hardware übersetzt werden konnten.

Diese virtuelle Maschine hatte einen 8080-Prozessor (deswegen werden alle schönen Eigenschaften des Z80 nicht benutzt), bis zu 16 Laufwerke, eine Konsole, einen Lochstreifen-Stanzer und -Leser usw. Wie gesagt: virtuell, also nicht vorhanden. In Wirklichkeit hat unser realer Rechner vielleicht einen HD64180 als Prozessor, nur 2 Laufwerke nebst Festplatte, ein Grafik-Terminal und eine serielle Schnittstelle. Das BS übernimmt es nun für das Anwenderprogramm, diese Umsetzung von der virtuellen auf die reale Maschine (möglichst sinnvoll) vorzunehmen.

Wenn es zu einem virtuellen Gerät keine Entsprechung auf der realen Maschine gibt, es uns das ziemlich egal (dann wird eben das 16. Laufwerk nicht benutzt), aber umgekehrt wird die Sache ärgerlich: Wir haben ein reales Gerät, das nirgends in unsere virtuelle Maschine paßt, z.B. eine Grafikkarte mit 256 Farben. Ist das nun eher ein Laufwerk oder ein Lochkarten-Stanzer? Weder noch, es paßt eben nicht. Das wurde damals nicht bedacht und damit aus.

Halt, so einfach geben wir ja nicht auf! Die virtuelle Maschine muß einfach nur (z.B. um ein virtuelles Gerät) erweitert werden und schon sind wir wieder im Geschäft. Das hat Digital Research mit den verschiedenen CP/M-Versionen auch gemacht. Trotzdem ist unsere Grafikkarte immer noch nicht drin, was nur beweist, wie kurzsichtig einige BS-Bauer (nicht mit Landwirten zu verwechseln!) sind. Aber auch mit einer Erweiterung um eine Grafik-Schnittstelle würde kein älteres Programm sie benutzen können (vorher gab es sie eben nicht) und neuere Programme würden auf den alten BS-Versionen nicht mehr laufen. Das Problem kennen nicht nur CP/Mler, sondern auch die MS-DOSler, die inzwischen bei Version 5 anlangen, während mit OS/2 schon der Nachfolger auf der Matte steht, genauso wie damals MS-DOS in der Blütezeit des CP/M.

Dieses Prinzip der virtuellen Maschine (wie in CP/M) statt der realen Maschine (wie in NEWDOS) findet sich in den meisten modernen Betriebssystemen. Ansonsten dürfte ein Rechner und sein darauf zugeschnittenes BS nie geändert werden und nach dem Veralten des Systems wären alle darauf geschriebenen Programme Müll. Wie wir gesehen haben, können sie ansonsten auch Müll werden, aber vielleicht nicht ganz so schnell.

MS-DOS

(Alle MS-DOS-Freaks überspringen bitte dieses Kapitel. Ich bitte darum, von Briefbomben abzusehen, sondern lieber auf das nächste Clubtreffen zu warten, wo Ihr dann persönlich handgreiflich werden könnt.)

Verlassen wir den Z80-Bereich und stoßen wir weiter vor ins Land der begrenzten Möglichkeiten, "IBM-Country". IBM entwickelte mal in grauer Vorzeit (also vor ca. 10 Jahren) einen Rechner, der ungefähr einem um den Faktor 10 vergrößerten TRS-80 entspricht: Prozessor 8088/8086 (leicht aufgemotzter Nachfolger des 8080), 20-Bit-Adressen, also 1 MB Adreßraum, aber höchstens 640 KB Hauptspeicher (RAM), ROM(-BASIC) eingebaut, alles drin und drumherum genau festgelegt (Memory-Map, Bildschirmspeicher usw.). Allerdings wurde das Ding manchmal schon mit einer Grafikkarte ausgeliefert!

Dafür brauchte IBM natürlich ein Betriebssystem (auch wenn die Kassetten-Schnittstelle wie beim TRS-80 nicht fehlen durfte), wobei seltsamerweise NEWDOS nicht in die engere Wahl kam. Aber CP/M war dabei, nur hatten wir ja keinen 8080/Z80 mehr da drin, also hätte es (leicht) umgeschrieben werden müssen. Irgendwie vergrätzte Digital Research aber IBM und somit bekam Microsoft die Aufgabe, ein BS für den IBM PC zusammenzuschustern. Was sie dann auch genauso taten (eben "zusammenschustern").

Eigentlich handelte es sich dabei dann schließlich nur um ein aufgemotztes CP/M, damit die Anwender-Programmierer aus dem CP/M-Bereich nicht umlernen mußten und die CP/M-Programme schnell angepaßt werden konnten. Nur wurde statt des "CALL 0005h" ein "INT 21h" genommen, was das gleiche ist, nur etwas kürzer. (Beim Z80 gab's das auch schon, hieß nur "RST xxh" und wurde unter CP/M nicht benutzt, obwohl es der 8080 auch konnte (glaube ich). Aber NEWDOS hat es benutzt!)

Ansonsten wurden wieder zwei Teile genommen, BDOS und BIOS, die aber diesmal getrennt auf der Diskette stehen. Teile des BIOS sind auch im ROM (bzw. manchmal auch mehreren ROMs) vorhanden. Außerdem mußten natürlich einige Erweiterungen an der "virtuellen Maschine" vorgenommen werden, damit alle vorhandenen und geplanten Eigenschaften des IBM PC eine Entsprechung hatten.

Bis hierher sieht die Sache noch rund aus. Aber nun kam ein (meiner Meinung) fataler Fehlschritt: Statt das MS-DOS als Grundlage (=virtuelle Maschine) für alle Anwender-Programme zu nehmen, wurde von IBM die Hardware genau fest- und offengelegt, wie beim TRS-80! (Während beim MS-DOS einiges im Dunkeln blieb, wie beim NEWDOS.) Außerdem erwies sich das MS-DOS als nicht besonders schnell, war unflexibel und fehlerhaft. Also dachten sich unsere Anwender-Programmierer: "Egal, ich kenne ja die Hardware, greife ich doch direkt darauf zu!" Und schon war die Grafik schneller und farbiger o.ä.

Aber: Jede anders aufgebaute Hardware eines anderen Herstellers (sei es nun ein ganzer Rechner oder nur eine neue Grafik-Karte) läßt sich von solchen Programmen nicht benutzen, weil sie ja genau auf die IBM-Hardware zugeschnitten sind. Dementsprechend liegen jedem Programm auch tausend Treiber (meist zu Grafikkarten und Druckern) bei, weil das MS-DOS eben zu blöde ist. Und deshalb heißt es "IBM-kompatibel" statt "MS-DOS-kompatibel". Ein kleiner Vorteil: Jedes MS-DOS müßte auf jedem IBM-Kompatiblen sofort laufen, ohne Anpassung des BIOS.

Womit wir wieder bei den TRS-80/NEWDOS-Verhältnissen wären. Jeder Rechner, auf dem MS-DOS-Programme laufen sollen, muß genauso aussehen wie der Original-IBM PC. Rechner, die von dem (Hardware-)"Standard" auch nur ein Bit abweichen, bekommen böse Probleme (woher kennen wir diesen Satz?). Die "virtuelle Maschine" wurde also zugunsten der realen Maschine fallengelassen.

Bleiben wir aber beim MS-DOS: Im Gegensatz zum NEWDOS, und das ist immerhin ein Fortschritt, sind die genauen Adressen des MS-DOS (was steht wo?) nicht mehr festgelegt. Es gibt nur die von CP/M bekannte Schnittstelle per CALL/INT. Leider wurden, wie gesagt, alte CP/M-Hüte übernommen, so daß wirkliche Neuerungen (z.B. Unterverzeichnisse und die File-Handles statt der FCBs) erst beim XT (MS-DOS 2.0) eingeführt wurden (Info von Hartmut). Natürlich muß man kompatibel bleiben und so bleiben alte Sünden lange fortbestehen... Die "Neuerungen" stammen übrigens aus UNIX, womit wir beim nächsten Thema wären.

UNIX

Lassen wir die profane PC-Welt hinter uns und steigen ein ins Großrechner- und Hochschul-Geschäft. Als CP/M gerade anfing, die Mikrocomputer zu erobern, gab es ja auch noch große und sehr große Rechner. Einige davon stehen immer in irgendwelchen Entwicklungslabors und die Leute, die daran arbeiten, sind meistens mit den gelieferten BSen nicht zufrieden. So werden immer neue BSe entwickelt (genauso wie Programmiersprachen), die aber kaum mal die Labors verlassen.

Anders dagegen UNIX: Zuerst war es ein Public-Domain-BS, das besonders in Entwicklungs-Labors und Hochschulen populär wurde, weil es auch einfach und sehr gut dokumentiert war. Erst mit dem wachsenden wirtschaftlichen Erfolg wurde das "Public Domain" wieder zurückgezogen. Trotzdem ist UNIX samt Nachfolgern und Nachmachern (z.B. ULTRIX, XENIX eben alles mit "-IX" am Ende) vor allem noch im Hochschulbereich anzutreffen, denn es ist eine Art Lehrstück für "gute" (hardware-unabhängige, leicht übertragbare, erweiterungsfähige) BSe.

Damit hätte ich auch die Hauptsache schon gesagt: Dieses BS ist nun wirklich von keinerlei Hardware mehr abhängig; selbst der Prozessor ist nicht mehr wichtig! Wie das? Nun, UNIX und alle wichtigen Programme selbst sind nicht in Assembler geschrieben, sondern in C, einer "höheren" Programmiersprache. C wurde parallel zu und mit UNIX entwickelt, um das BS und die Utilities zu schreiben. (Ähnlich wurde CP/M in PL/M entwickelt, das aber nie bekannt wurde.)

Wenn nun ein neuer Rechner mit UNIX samt Programmen ausgestattet werden soll, müssen a) ein C-Compiler, der Code für diesen Prozessor erzeugt, und b) einige hardwarenahe Teile des BS neu geschrieben werden. Wenn dann das BS und seine Programme, die ja alle im C-Quellcode vorliegen sollten, neu compiliert sind, gehört der neue Rechner zur Familie der UNIX-Rechner.

Leider weiß ich auch nicht mehr darüber, denn ein "Systems Guide", wie es Digital Research zu jeder CP/M-Version liefert und in dem alle interessanten Informationen für die Installation auf einem neuen Rechner stehen, habe ich für UNIX noch nicht gesehen. Dem interessierten Leser empfehle ich Tanenbaum's MINIX, ein UNIX-Abkömmling, der wieder Public Domain ist.

Nun fragt sich natürlich: Was hebt UNIX über die Masse von NEWDOS, CP/M und MS-DOS noch hinaus? Nun, da UNIX aus dem Großrechner-Bereich kommt, wurde alles vorbereitet, um parallele Prozesse, viele Benutzer, viele Dateien und viele Peripherie-Geräte zu unterstützen. Die "viele" sind wörtlich zu nehmen - "fast unendlich viele" wäre vielleicht besser.

Haupt-Ansatzpunkt dabei ist das File-System. Während bisher immer noch eine Trennung zwischen Speicher- und Peripherie-Geräten stattfindet, wird hier alles zusammengeworfen und als Files behandelt. D.h., eine Diskette oder Festplatte ist nur ein File, entweder ein sehr großer oder unterteilt in mehrere kleine, wie wir es schon kennen. Aber auch ein Drucker ist nur ein File (auf den wir normalerweise nur schreiben), eine Tastatur ebenfalls (meistens nur gelesen) usw. Da die Anzahl der Files (fast) unbegrenzt groß ist, können auch beliebig viele Peripherie-Geräte angeschlossen werden - jedes ist ja nur ein neuer File. Sie bekommen dann so nette Namen wie "PRT" (Drucker), "FD40DSDD" (Floppy, 40 Track, doppelseitig, doppelte Schreibdichte), "TTY" (Bildschirm) usw. Kennen wir ja schon aus CP/M, nur war das Konzept da etwas anders.

Somit haben wir ein BS, das sich a) leicht auf (fast) jeden Rechner übertragen und b) (fast) beliebig erweitern läßt. Warum ist es dann trotzdem noch nicht so verbreitet? Dafür mag es mehrere Gründe geben, über die ich hier nur spekulieren kann. Wahrscheinlich reiten alle Hersteller von Computern lieber auf ihren eigenen Entwicklungen rum (besonders IBM). Außerdem ist das BS für (heutige) Mikro- und Personalcomputer zu groß (Tanenbaum spricht von einer Minimal-Größe von 128 KB). Parallele Prozesse und beliebige Erweiterbarkeit sind im Hobby- und Büro-Bereich noch nicht gefragt und wurden (von der Hardware-Seite) auch nicht unterstützt. Das ändert sich langsam. Und nicht zu vergessen: UNIX steht und fällt mit C, das zwar immer mehr Anhänger findet, aber auch viele Kritiker hat (ich zähle mich gern dazu).

Schluß

Jetzt habe ich mich einmal quer durch den Garten der Betriebssysteme genörgelt. Besonders IBM/MS-DOS und NEWDOS haben ihr Fett abbekommen, aber das liegt an meiner Sichtweise. Bei UNIX könnte ich auch sicherlich noch einige Nachteile finden, wenn ich nur so ein System näher kennen würde. Ich habe es hier nur der Vollständigkeit halber aufgenommen.

Natürlich dürft Ihr über alle Punkte anderer Meinung sein, insbesondere andere Aspekte in den Vordergrund stellen und somit zu einer anderen Wertung kommen. Aber dazu müßt Ihr schon selbst mal zur Feder greifen (bzw. in die Tasten hauen), um mir "Kontra" zu geben. Ich nutze hier jedenfalls schamlos Eure Schreibfaulheit aus, um meine Meinung als die (scheinbar) allgemeingültige kundzutun. Ätschibätsch!

Der Kapitän starrte verblüfft auf die linke Hand seines Copiloten, der den Anflug auf Frankfurt durchführte. Sie bediente die drei Schubhebel der DC 10. Es war böig, und wegen der starken Geschwindigkeitsschwankungen mußte bald Motorleistung nachgeschoben, bald stark zurückgenommen werden. Das automatische Schubhebelsystem (Automatic Throttle System), das normalerweise die Schubhebel reguliert, war 'übungshalber' abgeschaltet worden. Doch seltsam: Der Pilot bediente die Hebel genauso ruckartig und 'überreagiert', als sei die wenig sensible Automatik eingeschaltet. Tatsächlich bedient jeder Pilot die Schubhebel von Hand gefühlvoller: seine Erfahrung sagt ihm zum Beispiel, daß ein plötzlicher Böenschlag, der Geschwindigkeitszuwachs bringt, überhaupt nicht berücksichtigt zu werden braucht, weil im nächsten Augenblick die gegenläufige, ausgleichende Tendenz einsetzt. Eine Automatik hingegen, sei sie noch so feinfühlig, muß auf jeden Impuls von außen reagieren.

Hinterher stellte sich heraus, daß der Copilot, unter Streß, instinktiv die Bewegungen des Auto-Throttle-Systems an den Schubhebeln nachgeahmt hatte, obwohl er weitaus sensibler hätte fliegen können.

Dieses Beispiel aus dem Anfang der achtziger Jahre läßt Prokrustes wiederauferstehen: Wenn die Technik die Forderungen nicht erfüllen kann, muß der Mensch sich eben nach ihren Möglichkeiten einrichten. Bei der eben erwähnten 'Bettstreckung' erfolgte die Anpassung sogar, wenn auch unter Streß, freiwillig.

Fälle, bei denen die Technik die Forderungen ihrer Benutzer nicht erfüllen kann, gibt es mehr, als der mit den Erfolgsmeldungen aus dem Elektronikbereich überschüttete naive Laie glaubt. Einer von vielen - ebenfalls aus der Luftfahrt: Für die halbjährlichen Überprüfungen der Zivilpiloten hatten die Verkehrsfluggesellschaften ein schwieriges fliegerisches Manöver eingebaut, das als Canyon Approach bekannt ist: Ein Flugzeug macht einen Anflug auf einen Platz in einem Canyon, der am Ende 'verschlossen' ist. Der Anflug mißlingt, es muß durchgestartet werden. Gleichzeitig mit dem komplizierten Verfahren des Powergebens, Fahrwerk- und Klappeneinfahrens muß nun auch noch eine steile Kehrtwende um 180 Grad eingeleitet werden.

Diese Verfahren war äußerst kompliziert, es schied aber auch die 'Könner' von den 'Nichtkönnern', weil dabei ein äußerst schwieriges Zusammenspiel von Triebwerkbedienung, exakter Ruderbedienung und rascher Entscheidungsfähigkeit gefordert wurde. Kein Wunder, daß dieser Canyon Approach von den ersten Kolbenflugzeugen a la Superconstellation bis zu den Jets die Hohe Schule des Verkehrsfliegens war. Wer damit nicht klarkam, fiel durch den Check.

Doch dann wurden diese Überprüfungen nicht mehr im echten Flugzeug, sondern im Simulator praktiziert. Es stellte sich heraus, daß Simulatoren der damaligen Generation diese komplizierten Flugvorgänge nicht befriedigend simulieren konnten. Schickte man die unbrauchbaren Geräte an ihre Hersteller zurück? Natürlich nicht: Man strich schlicht diese Übung aus dem Überprüfungsprogramm. Plötzlich war nicht mehr wichtig, was über zwanzig Jahre lang so manchem Prüfling Durchfälle beschert hatte - in doppelter Bedeutung.

Wir stehen heute vor der Tatsache, daß unsere elektronischen 'Hilfen' nicht das leisten, was wir von ihnen fordern, sondern daß wir uns mit unseren Forderungen ihren Leistungen angepaßt haben. Ein primitives Beispiel aus dem Haushaltsalltag: Früher konnten wir unsere Bank- und Postgiroüberweisungen handschriftlich ausfüllen, wie uns die Klaue gewachsen war. Hauptsache: überhaupt lesbar. Weil der moderne Computer dazu absolut unfähig ist, haben wir nun, als Mehrarbeit, gefälligst sorgfältig in Großdruckbuchstaben zu schreiben, und, bitte, nur ein Buchstabe pro vorgedrucktes Kästchen. Sonst muß der Herr Lesecomputer den Dienst verweigern. Und niemand ist da, der diesen, als modernsten Fortschritt deklarierten Zivilisationsschrott den Banken und der Bundespost als Plunder an den Kopf wirft, mit der Auflage, endlich einen Computer zu liefern, der das kann, was jeder Aushilfslehrling kann: individuelle Handschrift lesen. (Daß es sehr wohl Computer gibt, die Handschriften lesen, identifizieren und vergleichen können, ist kein Gegenargument, solange sie so teuer und fehleranfällig sind, daß sie sich ein normaler Betrieb nicht leisten kann.)

An diesem allgemein verständlichen Beispiel zeigt sich ein Dilemma, das von Computergeneration zu Computergeneration akuter wird: Der Computer ist unfähig zu einer echten 'Companionship' mit seinem Partner, dem Menschen. Eine 'Zusammenarbeit' kann nur stattfinden, sofern der menschliche Teil des Teams sich den Eigenarten, Unarten und mangelnden Fähigkeiten der Elektronik anpaßt. Diese Art einer Partnerschaft wirft eigentlich alles über den Haufen, was zum Beispiel von den Fluggesellschaften seit mehr als einem Jahrzehnt als Heilige Kuh der Cockpitarbeit dogmatisiert worden ist: Teamwork. Ein Besatzungsmitglied - Captain eingeschlossen - das zu einer fruchtbaren Anpassung und einem Zusammenspiel mit den anderen nicht fähig ist, fällt beim nächsten Check durch. Bezeichnenderweise werden die halbjährlichen Überprüfungen schon lange nicht mehr für eine Einzelperson (Captain, Copilot, Flugingenieur) durchgeführt, sondern gleichzeitig für die gesamte Cockpitcrew. Als Besatzungsmitglied wäre der Computer wegen seiner

mangelnden Anpassungsfähigkeit und seiner Eigengesetzlichkeit eine Katastrophe. Und dabei soll er doch zumindest den Flugingenieur, zum Teil auch noch Copilot und Captain ersetzen! Er spielt nur mit, wenn man bereit ist, auf seine Art einzugehen, wie er 'bedient' werden möchte. (Das Wort ist verräterisch.)

Kein Wunder, daß bei der eben noch hochgepriesenen Bordcomputergeneration die Fehlerlimits bei automatischen Flugvorgängen höher lagen, als wenn der Pilot das gleiche Verfahren von Hand flog. Er mußte zum Beispiel von Hand exakter Geschwindigkeiten einhalten, als wenn er die Automatik dafür einschaltete. Mit ihr durfte schlampiger geflogen werden. Von einer gleichwertigen Partnerschaft kann also nicht die Rede sein. Hingegen ist der Computer dort überlegen, wo er blitzschnell Informationen liefern soll oder/und Abflugrouten, Luftstraßen und so weiter optisch darstellt. Die Dutzende von Kursen und Richtungsänderungen, die früher im Kopf der Piloten nur mühsam zu einem Bild wurden, werden jetzt in Sekundenschnelle auf den Schirm projiziert, mitsamt dem sich bewegenden Flugzeug. So läßt sich auch, ein Beispiel von Dutzenden, beim Abstieg aus der Reiseflughöhe erkennen, an welchem geografischen Punkt das Flugzeug bei einer bestimmten Sinkrate in einer bestimmten Höhe ankommen wird. Eine Information, die früher mühsame Kopfrechenvorgänge benötigte.

Fast könnte man sagen: Der Computer entwickelt so etwas wie ein Schamgefühl, weil er sich der menschlichen Art zu denken, zu erkennen, zu reagieren, nur mangelhaft anpassen kann. Er kompensiert diesen Mangel durch eine wahre Sturzflut von Informationsmöglichkeiten und Automatikfähigkeiten, die so exzessiv nun auch wieder nicht gebraucht werden. Ein Beispiel von vielen:

Schon auf der DC 10 gab es die Möglichkeit, Gewitterfronten automatisch zu umfliegen, indem man bestimmte Kurse in den Navigationscomputer eintippte, so daß nach dem Umfliegen die Maschine sofort wieder auf den alten Kurs zurück gebracht wurde. In der Praxis kenne ich niemanden, der das, außer bei einem Checkflug, jemals gemacht hätte: Das Vorprogrammieren war weitaus mühsamer, als wenn man in alter Weise schlicht am Kursknopf des Autopiloten drehte und nach Sicht und eigenem Gusto die Wolken umkurvte, unter Umständen sogar von Hand.

Nun ist es freilich nicht der Computer, der ein Schamgefühl entwickelt. Vielmehr wird er ja von einem Menschen konstruiert. Dieser Mensch freilich ist keinesfalls der, der ihn dann - bleiben wir im Luftfahrtbereich - benutzt, um mit ihm den Nordatlantik zu überqueren. Im Gegenteil, dieser Ingenieur hat von den Problemen eines gestreßten Flugkapitäns, der auf 30W in eine nicht vorausgesagte Gewitterfront gerät, nicht die geringste Ahnung. Er kann sie daher seinem Gerät auch nicht als 'einprogrammierte Gene' mitgeben. Nach wie vor kann der Computer nur dann vollwertige Arbeit leisten, wenn er vorschriftsmäßig fehlerfrei programmiert wird. Damit aber fällt er im Vergleich zu einem simplen Copiloten, der gerade hundert Stunden Flugerfahrung hat, auf den Status eines Vollidioten zurück. Denn ein solcher Copilot wird eine unter Streß gemachte falsche Anordnung eines anderen Crewmitglieds erkennen und sie nicht ausführen, sondern zumindest darüber diskutieren.

Das völlige Versagen des Computers beim Fehlverhalten seines Betreibers (von der 'Partnerschaft' sollte endgültig nicht mehr geredet werden) ist aber genau der Gefahrenpunkt, der von Fortschritt zu Fortschritt größer wird. Jeder Autofahrer, der in einem computerüberwachten Parkhaus parkt, kennt das Problem: Obwohl pausenlos Fahrer hinausfahren und bestätigen, es sei Platz genug, zeigt der Rechner besetzt an und verweigert die Einfahrt. Der Grund: Er zählt schlicht und einfach die Einfahrenden, ohne zu berücksichtigen, daß viele illegal ihren Wagen auf verbotenen Parkflächen abstellen (und somit echte Parkflächen freihalten). Jeder Hilfsarbeiter, der bis zwanzig zählen kann, wäre hier als Parkwächter dem kostspieligen Computer überlegen.

Und während alle Journalistenwelt, aus den Hochglanzwerbebroschüren zitierend, die vollautomatischen Fähigkeiten einer Elektronikgeneration lobt, die längst und besser nicht nur den Flugingenieur, sondern auch schon fast den Copiloten ersetzt hat, wird einfach unterschlagen, daß nun kein Flugingenieur mehr da ist, der über die Schulter der beiden Piloten mit überprüft, ob die Checklisten richtig gelesen werden und der hinausblickt, um beim Anflug auf die Kollisionsgefahr mit anderen Flugzeugen zu achten.

Die absolute Mißachtung menschlicher Fehlleistungen ist genau die Hauptfehlleistung von Computeranlagen. In Tschernobyl war die Automatik genauso wenig in der Lage, die falschen Absichten der Programmierer am Pult (Simulation bestimmter Notzustände) zu erkennen, wie die Computer im Airbus 320, als der französische Pilot seinen Airbus bei einer Vorführung in den Wald flog. Wollte der Pilot das wirklich? Keinesfalls. Er wurde nur durch seinen Ehrgeiz getrieben, ein wenig riskanter und tiefer als normal zu fliegen. Immerhin war er als Chefpilot ja ausgewählt worden, andere Piloten auf diesem neuartigen Flugzeug, 'das keine Pilotenfehler mehr zuläßt', auszubilden. Wer hatte da wirklich 'menschlich versagt'? Von den Airline-Managern, den Psychologen, den Vorstandsmitgliedern, die diesen Mann zum Chefpiloten gemacht hat-

ten, war nie die Rede. Und die neuartige Elektronik hatte nicht einmal ein bißchen menschliche Eitelkeit mit ihren Fehlleistungen erkennen können. Wir alle müssen wohl weitaus anspruchsvoller gegenüber dem technischen Fortschritt werden.

In Zukunft darf keine neue Computergeneration eine Chance haben, die menschliche Fehlleistungen, die nun mal 'normal' sind, nicht bewältigen kann. Sei es, daß ein Girokontenformularausfüller die Großbuchstaben zu undeutlich schreibt, Falschparker die elektronische Zählung verfälschen, ehrgeizige Piloten risikoreicher als einprogrammiert fliegen. Und wieso eigentlich verlangen wir von unserer Kaffeemaschine nicht, daß sie das im Mitternachtssuff einprogrammierte 15:08 Uhr rechtzeitig als vertippt und als 08:15 Uhr für die erste Tasse Kaffee erkennt? Unsere technischen Wunder der siebenten Computergeneration können es nicht. Wie gesagt, man sollte anspruchsvoller werden.

Die Frage, ob die immer wieder sensationell herausgestellten Leistungen moderner Computersysteme wirklich einen Fortschritt bedeuten, entscheidet sich an der Klippe der Umfelderweiterung. Soll die gegenwärtige Computerentwicklung zu einem echten Fortschritt werden, müßte sie zumindest die schlichten Fehlleistungen aus dem weiteren Umkreis erkennen und ausgleichen. Allerdings ist den fortschrittsgläubigen Ingenieuren noch völlig unbekannt, daß sich gerade durch ihre fortschrittlichste Technik neuartige juristische Probleme ergeben könnten:

Seit Mitte der sechziger Jahre wurden immer wieder Meldungen lanciert, nach denen nun die vollautomatische Landung bei Nullsicht gelungen sei. Keinem der aus den Hochglanzprospekten abschreibenden Reporter fiel jemals auf, daß diese Demonstrationslandung a) stets ohne Passagiere mit einer speziell ausgebildeten Testcrew oder b) bei Wetterlagen durchgeführt wurden, die weitaus besser als Nullsicht waren.

Jetzt, mit einer mehr als zwanzigjährigen Verspätung nach der ersten triumphalen Vollzugsmeldung, sind endlich diese Landungen in den Bereich des Machbaren (mit Passagieren) gerückt. (Freilich mit der Einschränkung, daß bestimmte Querwindstärken nicht überschritten werden dürfen.) Und schon treten völlig andersartige Probleme auf, mit denen die von der Wirklichkeit isolierten Ingenieure nicht gerechnet hatten: Die Piloten sehen sich außerstande, wegen der zu kurzen Reaktionszeiten für das manuelle Eingreifen noch irgendeine Verantwortung zu übernehmen. In Zukunft also werden 'technisch machbare' Projekte scheitern, weil das menschliche Umfeld stets völlig ausgeklammert wurde. So zeigt sich in den reduzierten Leistungsmöglichkeiten der Computer die Unfähigkeit ihrer Hersteller, größere Zusammenhänge im menschlichen Bereich herzustellen.

Versagen ist auch immer dort vorprogrammiert, wo es zum sogenannten Multiple Failure oder sogar Triple Failure kommt: Ein Triebwerk gerät in Brand, gleichzeitig fällt die Feuerwarnanlage aus, und auf einem anderen Triebwerk fällt der Öldruck auf Null. Jetzt geht es um Prioritäten: Welche Störung muß zuerst behoben werden? Diese Rangordnung kann nie durch den Computer bestimmt werden, sondern nur durch den Captain aufgrund der akuten Situation. Sie wird jedesmal anders ausfallen, je nachdem, ob er sich über der einsamen Sahara, über Mitteleuropa mit seinen Hunderten von Flughäfen befindet, ob er den Dienst frisch angetreten hat oder seine zwölfte Dienststunde absolviert, ob er in eine Schlechtwetterzone einfliegen muß oder einen Rückwind von 200 Knoten hat.

Wir alle sind betroffen. Fällt in einem Atomkraftwerk nicht nur ein Kühlsystem, sondern am Kontrollpult durch eine elektrische Störung zusätzlich das Warnsystem und die Darstellung des Störverlaufs aus, dürfte der wachhabende Ingenieur schon dadurch total überfordert sein, zumindest, wenn auch die Abschaltautomatik gestört wird. Er dürfte kaum halbjährlich auf Triple Failures trainiert werden, wie ein Airline-Captain. Gerade aber, wenn das gesamte Abschaltprogramm vollautomatisch ablaufen würde, ließe sich vorstellen, daß es für bestimmte Störfälle gar kein Automatikprogramm gibt, einfach, weil sich die Programmierer einen solchen Fall nicht vorstellen können (oder dürfen).

Ein simpler Fall aus der Luftfahrt von 1959, als Computer noch gar keine Rolle spielten, verdeutlicht das Problem, das heute verstärkt auftritt. Da die damalige viermotorige Superconstellation bei einem Vollaststart nur schwer vom Boden zu kriegen war, hatte sich ein Flugkapitän als Privatverfahren zur Auftriebserhöhung folgendes ausgedacht: Für den Start werden die Klappen auf 'Startstellung' gefahren. Doch bevor sie, bei zunehmender Geschwindigkeit, den Auftrieb vergrößern können, bieten sie beim Anrollen nur Widerstand und Beschleunigungsverzögerung. Also legte er vor dem Anrollen den Klappenhebel zwar auf 'Start', zog jedoch vorher die elektrische Sicherung, die das Ausfahren der Klappen bewirkt. Sie blieben also auf Null. Als dann der Moment der notwendigen Auftriebserhöhung da war, wurde die Sicherung eingedrückt, die Klappen fuhren auf 'Startstellung', die Maschine hüpfte in die Luft. Das Privatverfahren bewährte sich bis zu jenem Tag, als genau beim Eindrücken der Sicherung die Elektrik total versagte. Die Klappen fuhren nicht aus. Ergebnis: Aufschlagbrand in Paris Le Bourget, in dem

alle Insassen umkamen.

Es gehört zur Alltagsphilosophie moderner Elektronikprogramme, daß derartige menschliche Reaktionsweisen weder vorgesehen noch abgedeckt sind. Für die Programmierer sind sie schlechterdings 'unvorstellbar'.

Die Möglichkeit zu einer echten Partnerschaft zwischen Mensch und Computer zu gelangen, wird freilich nicht nur durch die Unfähigkeit der Softwaresysteme selber eingeschränkt, sondern mehr noch durch die überzogenen Anpreisungen, mit denen die zuständige Industrie ihre Produkte auf Ansprüche hochzupushen versucht, die diese nun ganz bestimmt nicht erfüllen können. Die unzähligen unerwarteten Störfälle in unseren Atomkraftwerken (die angeblich sichersten überhaupt, auch wenn seit Biblis, einem Musterfall für 'menschliches Versagen', etwas weniger laut das Lob der deutschen KKWs hinausposaunt wird), das Auftreten von 'Phantomerscheinungen', die sich die ohnehin recht mangelhafte Phantasie der im klinisch-sterilen Raum arbeitenden Ingenieure nie vorstellen konnten, die Materialermüdungen, die Haarrisse in den Kühlanlagen der Reaktoren, die elektronischen Störungen der Warnsysteme, die sich schon mit einem aufs Kontrollpult gelegten Schlüsselbund erzeugen ließen - all diese 'Abfallprodukte' einer maßlos optimistischen Aberdaslaßtsichdochallesmachen-Philosophie werden am besten illustriert durch ein typisches Beispiel aus dem Waffengeschäft.

Als die Pershing-II-Raketen an den Mann gebracht werden sollten (das gilt ähnlich für alle anderen Raketensysteme auch), fand vor den Militärkunden ein Demonstrationsschießen statt. Weil die Rakete nicht annähernd zielgenau war, verringerte man nicht nur die Reichweite des Demonstrationszieles drastisch, sondern installierte dort auch einen Sender, der die ziemlich ziellos herumirrende Rakete auf einer bestimmten Frequenz heranlocken mußte. Trotzdem lag die Trefferrate von 21 Raketen bei weniger als 20 Prozent. Nach dem gleichen Procedere werden wohl auch unsere Atomkraftwerke verkauft.

Allerdings hat die Atom- und Computerlobby stets einen Verantwortlichen als Sünder griffbereit: den, der am Pult des Atomkraftwerkes, im Airbuscockpit oder im Führerstand des ICE-Triebkopfes 'menschlich versagt' hat.

Der Schluß daraus wird nur ungern oder nie gezogen: Wenn Mensch und Computer zusammen eine Situation nicht meistern können, kann eigentlich der Mensch nicht schuld sein (es sei denn, er ist ein Selbstmörder). Seine Verhaltensschemata sind seit Jahrtausenden bekannt. Hingegen ändern sich die Möglichkeiten technischer Systeme von Jahr zu Jahr. Hat sich auch die neueste fortschrittliche Computergeneration wieder einmal nicht den Menschen anpassen können, sein Versagen nicht ausgleichen können? Wieso ist sie dann fortschrittlicher als die vorige?

Der Mensch, so der allgemeine Tenor der Fluggesellschaften, wird sich in absehbarer Zeit im Cockpit nicht ersetzen lassen. Er ist einzigartig dadurch, daß er seine Erfahrung, seinen Intellekt, Instinkt, seine Prioritätenentscheidung der akuten Situation anpassen kann. Der Mensch reagiert auf unvorhergesehene Störungen und Systemausfälle auf jeder Fall flexibler als ein Computer, dem die Reaktionen darauf gar nicht einprogrammiert wurden, weil sie im Bewußtsein des Programmierers nicht existent waren.

So könnten sich beide Systeme bestens ergänzen und zu einer echten Partnerschaft kommen: Der Mensch mit seiner Flexibilität und seinem Erfahrungsschatz, der Computer mit seinem Überangebot an Informationen und Rechenvorgängen.

Dieser Forderung steht zur Zeit noch der maßlose Anspruch der Technologen gegenüber, durch den Computer einen 'besseren Menschen' schaffen zu wollen, der an seiner Perfektion zur Zeit nur noch durch die Fehlleistungen des niederen Menschen gehindert werde.

- ### -

Rudolf Braunburg war 20 Jahre lang Flugkapitän und lebt heute als freier Schriftsteller in Walderöl. Er ist Autor mehrerer Reise- und Sachbücher und beschäftigt sich vor allem mit Fragen des Fliegens und der Flugsicherheit.

Railroad Tycoon

Eine komplexe Wirtschaftssimulation, rund um die Eisenbahn!

Als ich meinen Aufruf an den Jens abgeschickt hatte, kam mir die Idee, einfach mal mit einer MS-DOS Ecke anzufangen und einige meiner Lieblingsprogramme zu beschreiben. Vielleicht fühlt sich der eine oder andere 'Untergrundkämpfer' angeregt, auch über seine Erfahrungen mit Dingen rund um MS-DOS zu berichten.

In irgendeiner Computerzeitung wurde ein Spiel namens Railroad Tycoon begeistert beschrieben. Da ich auch ein Eisenbahnfan bin, habe ich es noch zur Stunde bestellt. Der erste Eindruck verlief allerdings recht enttäuschend! Einen solch ewig langen Vorspann habe ich sogar bei der Shareware selten angetroffen. Auch von der Grafik hatte ich mir mehr versprochen, als ich zu Beginn die Option VGA gewählt hatte. Aber die Handlung hat es in sich und verspricht interessante Beschäftigung über Tage und Wochen.

Grundlage des Spiel ist das Errichten und Betreiben von Eisenbahngesellschaften. Es stehen verschiedene Landschaften und Zeitpunkte für den Beginn zur Auswahl:

Ostküste der USA um 1830
Westküste der USA um 1866
England und Wales um 1828 und
Mitteleuropa um 1900

Das Spiel beginnt damit, daß eine Eisenbahngesellschaft gegründet wird. Es werden 100.000 Aktien im Wert von 5 $ (bzw. 5 £) ausgegeben, sowie ein Darlehen über 500.000 $ aufgenommen. Mit diesem Geld kann innerhalb der angezeigten Landschaft eine beliebige Eisenbahnlinie gebaut werden.

Die Kosten für eine Meile sind je nach Gelände unterschiedlich hoch. Die Bahnhöfe (Depot, Station, Terminal) kosten unterschiedlich viel Geld, haben dafür aber auch unterschiedlich große Einzugsbereiche. Je nach Zeitpunkt und Örtlichkeit stehen verschiedene Lokomotiven zur Verfügung. Um 1830 kleine, wenig leistungsfähige Maschinchen. Man muß bei Streckenlänge und Transportvolumen starke Einschränkungen machen. Mit fortschreitenden Spieldauer kommen größere und leistungsfähigere Lokomotiven auf den Markt. Dann kann man die Zugmaschinen austauschen und längere Züge mit größeren Umläufen zusammenstellen.

Dann muß mit dem Aufstellen eines Fahrplanes begonnen werden. Beim Errichten der Bahnstation wird angezeigt, mit wievielen Wagen an Passagieren, Post und Gütern im Jahr gerechnet werden kann.

Wie im richtigen Leben verläuft die wirtschaftliche Entwicklung in Wellen. Es gibt Zeiten, da boomt das Transportaufkomen. Dann ist es, gerade mit den kleinen Maschinen, fast unmöglich, alle gefülltem Waggons zu befördern. In Zeiten der Rezession stehen die Züge auf den Bahnhöfen herum und warten auf Transportaufgaben.

Es ist natürlich möglich, beliebig viele Züge einzusetzen, um den vielfältigen Transportaufgaben nachzukommen. Jede Lokomotive hat aber, je nach Alter und Auslastung, bestimmte Betriebskosten, die sich am Jahresende auf das Betriebsergebnis auswirken.

Wenn Güter nicht innerhalb eines bestimmten Zeitraumes von einem Bahnhof befördert werden können, gehen sie auf anderen Transportwege über. Es empfiehlt sich deshalb, die Bahnhöfe mit Güterschuppen, Postämtern, Restaurants und Hotels für die Reisenden auszustatten. An Bahnhöfen mit viel Verkehr dauert das Umsetzten der Züge besonders lange, kostbare Zeit geht verloren. Die Anlage eines Rangierbahnhofes mit Drehscheibe ist hier notwendig. Um die Betriebskosten der Lokomotiven niedrig zu halten, können an beliebigen Stationen Werkstätten eingerichtet werden.

All das kostet, besonders in der Startphase, eine Menge Geld. Es kann zu Beginn mit einer Minimalausstattung operiert werden und die Investitionen aus dem laufenden Gewinn erbracht werden. Leider arbeiten aber noch vier konkurrierende Unternehmen in dem gleichen Gebiet! Um eine gewisse Strecke zu belegen ist es vielleicht besser, weitere Darlehen aufzunehmen. Der Zinsaufwand richtet sich dabei nach der augenblicklichen Wirtschaftslage und schmälert natürlich das Betriebsergebnis.

Jeweils nach zwei Jahren wird Bilanz gezogen. Alle Erträge und Aufwendungen werden gegenüber gestellt und der Gewinn (positiv wie negativ) ermittelt. War der Gewinn positiv, sind die Aktionäre zufrieden. Ist er negativ freut es sie weniger. Erwirtschaftet der Manager über fünf Jahre hinaus Verluste, wird er gefeuert und das Spiel ist beendet!

Wenn die Eisenbahnlinie dann gut läuft und sich das Geld häuft, kann man zu allerhandlei Schandtaten aufbrechen. Ist eine konkurrierende Eisenbahnlinie in der Nähe, kann ich mich an einen ihrer Bahnhöfe anschließen. Wer in den nächsten zwei Bilanzperioden mehr Güter abfährt als die Konkurrenz, hat gewonnen und wir alleiniger Betreiber. Es ist aber auch möglich, eine Eisenbahngesellschaft über den Aktienmarkt zu übernehmen. Dazu muß man mehr als 50% der entsprechenden Aktien aufkaufen. Wird gekauft, steigen natürlich die Preise. Also sollte man in Zeiten der Rezession kaufen, wenn die Kurse fallen. Natürlich ist es auch möglich, die Aktien des eigenen Unternehmens zu kaufen. Sind mindesten 50% im Bestand, kann die Gesellschaft nicht übernommen werden, noch können die Aktionäre bei Verlusten den Manager feuern!

Neben der Planung und Anpassung des Fahrplanes und den Geldgeschäften kann sich, wer will, auch als Fahrdienstleiter beschäftigen. Die Strecken können mit Signalbrükken in Blocks zerlegt werden. Die "eingebauten" Fahrdienstleiter sehen alles ein wenig konservativ und öffnen das Blocksignal erst, wenn der Block frei ist. Hier kann man eingreifen, und die Signale früher öffnen, oder auch schließen um Züge zu stoppen.

Nachdem ich ganze Nächte damit verbracht habe, dutzende von Eisenbahngesellschaften zu errichten, ist mir das Spiel immer noch nicht langweilig geworden. Leider habe ich eine englische Version erwischt. Das Handbuch ist aber leicht und flüssig geschrieben und bereitet wenig Schwierigkeiten.

Allerdings gibt es auch einige Mängel. So gibt es spaßige Einlagen, wie den Brückenbau. Die ersten zehn Male ist das ja ganz lustig, dann fängt es aber an zu nerven. Auch ist die Auswahl der Lokomotiven nicht ganz ideal. Man sollte sich nicht daran stören, daß z.B. die rhätische Ge 6/6 (Meterspur!!!) oder englische Typen durch ganz Europa sausen. Auch ist die Verteilung der Ressourcen ab und zu ein wenig verwunderlich, oder hat man z.B. schon von Weinbergen bei Königsberg gehört?

Aber alles in allem: Railroad Tycoon ist ein spannendes Strategiespiel (es muß nicht immer Kriegsspiel sein) das langen Spielspaß verspricht.

Zilogs späte Ernte

Der amerikanische Halbleiterhersteller Zilog, Entwickler des in Rekord-Stückzahlen verkauften Mikroprozessors Z8Ø, hat sich wieder an sein "verlorenes Kind" erinnert. Während der Zeit, als Zilog zum Ölmulti Exxon gehörte, meinte man mit Neuentwicklungen das große Geschäft machen zu können. Doch diese Entwicklungen konnten in keinem Fall an den Erfolg des Z8Ø anknüpfen. Zum Teil waren sie sogar Flops am Markt. Als der erhoffte Umsatz ausblieb, wurde die Halbleiter-Tochter für Exxon zu kostspielig. Der Unternehmensteil wurde verkauft, zum großen Teil an die eigenen (Zilog-) Mitarbeiter.

Seitdem entfalten die Californier eine rege Aktivität in Entwicklung und Marketing. Der serielle Communications-Controller Z853Ø ist einer der weit verbreiteten Schaltkreise in modernen Personal Computern. Aber auch auf die Qualitäten und Markterfolge des einstigen Zugpferdes Z8Ø besann man sich. Inzwischen erkannte man bei Zilog, daß man mit hochintegrierten Z8Ø-Controllern gute Geschäfte machen kann, denn das Know-How und die Programmiererfahrung für diesen Mikroprozessor ist weit verbreitet. Lest dazu einen Auszug aus der Zeitschrift "Elektronik-Informationen" Heft 1Ø/9Ø.

Heinrich Betz

1989 war für Zilog aber ein weiteres Erfolgsjahr. Es wurden mehr neue Bausteine vorgestellt als in jedem anderen Jahr. Welche Schwerpunkte nun Zilog setzt und welche weiteren Bausteine zu erwarten sind, erläutert **Dipl.-Ing. Eckhard Stock,** Managing Director der deutschen Niederlassung.

ELEKTRONIK INFORMATIONEN:
Herr Stock, welche Auswirkungen hatte die Übernahme auf die deutsche Niederlassung, und haben alle Mitarbeiter Anteile übernommen?

E. Stock: Weltweit wurde ca. 500 Mitarbeitern angeboten, sich an der Übernahme von Zilog zu beteiligen. Dies gilt auch für alle Mitarbeiter der deutschen Niederlassung. Dieses Angebot wurde von allen angenommen, so dass es bis heute ca. 500 Zilog-Angestellte mit Anteilen an der Firma gibt. Sie können sich sicher vorstellen, dass die Motivation, für eine Firma zu arbeiten, von der einem ein Teil gehört, grösser ist, zumal man sich vom Gang an die Börse einen finanziellen Vorteil verspricht.

ELEKTRONIK INFORMATIONEN:
Spielt heute der Z80 für Zilog noch eine Rolle?

E. Stock: Der Z80 spielt für Zilog nach wie vor eine grosse Rolle. Wenn man sich heute den Markt für Mikroprozessoren ansieht, kann man feststellen, dass die 8-Bit-Prozessoren nach wie vor stückzahlenmässig eine dominierende Rolle spielen. Innerhalb der 8-Bit-Prozessoren basieren mehr als die Hälfte auf dem Z-80-Befehlssatz. Der Z-80-Befehlssatz, den man sicher als Industriestandard bezeichnen kann und für den eine grosse Menge an Software existiert, ist heute noch Basis für viele Neuentwicklungen unserer Kunden. Dem Wunsch der Entwickler nach ständiger Leistungssteigerung der Systeme und höherer Integrationsdichte hat Zilog schon seit einigen Jahren entsprochen, indem man die Z-80-Produkte in CMOS weiterentwickelte, die Taktraten kontinuierlich erhöhte (den Z80 gibt es heute als 20-MHz-Variante), Z-80-Bausteine zu höher integrierten Bausteinen entwickelte und verschiedene Gehäuseformen wie PLCC oder Quad-Flat-Pack für höher integrierte Bausteine herausbrachte.

Ein wenig Nachdenkliches, Wissenswertes und Bekanntes für CP/M-Freunde. Leider in Englisch, das Übersetzen müßt ihr selbst besorgen. Den Artikel bekam ich über INFOCPM§FINHUTC.BITNET, der "Weiterreicher" David Goodenough ist Autor vieler PD-Programme, darunter ZSM, UUCP und QTERM. Viel Spaß beim Entziffern, Rüdiger.

A04980 (401 lines) CPM.PuDoSe 11/04/90 0700.8 fwt Sun cpm
To: äforum >site>t>digests>cpmü

INFO-CPM Digest Thu, 1 Nov 90 Volume 90 : Issue 169

Today's Topics:
Zen and the art of CP/M

Date: 1 Nov 90 05:30:59 GMT
From:
usc!zaphod.mps.ohio-state.edu!think.com!mintaka!spdcc!mirror!pallio!dg
§ucsd.edu (David Goodenough)
Subject: Zen and the art of CP/M
Message-ID: <XX00011f10§pallio.UUCP>

The following article appeared in the PIPMAG online magazine on GEnie. I found it interesting because of the history it contains, but also because it explains why I spend my time beating my head against a 14" green phosphor tube, covered with the Z80 assembler neumonics that make up QTERM, when I could do it in C on the 386 UNIX crate at work, and get it written much quicker.

It is reproduced with permission of the author.
--
dg§pallio.UUCP - David Goodenough +---+
IHS ô +-+-+
..... !harvard!xait!pallio!dg +-+-+ ô
AKA: dg%pallio.uucp§xait.xerox.com +---+

Zen and the Art of CP/M

by

Robert Coleman (R.COLEMAN3)

It was late. Maybe midnight, and quiet in the house. Upstairs, my wife and son were probably asleep. I could hear the faint sound of the TV droning; that and the soft breeze outside my opened window. Whoosh! Drive A on my computer was busy at work. I was backing up some files and doing general housekeeping after a long session at the keyboard. Tic-tic, whoosh, tic-tic! Drive B kicked in. I watched. I had to smile. It was noisier than drive A -- always was (I could close my eyes and tell which of my drives were running, just by the sounds they made). Drive B was perhaps the better of the two. It was newer. But they were both slow; just as slow and tired as I was now. I looked up at the window and felt the cool breeze on my skin. No, they weren't the fastest drives. Not like the 1.2 Meg on my 386 at work. That was much faster; faster by a country mile. And I pushed it too -- pushed the hell out of it. But that was work. Now, just being here late at night with this old CP/M computer, it was somehow different...

Zen and the Art of CP/M? (...like Motorcycle Maintenance? Ä10) Crazy? Hey, I'm not the first to get this feeling of religious fervor over a computer operating system. How about 'The SOUL of CP/M' and 'The CP/M BIBLE' Ä2,30. Mr. Waite and associates, they too had reverence for this IDEA. In spite of MS-DOS, countless thousands still cling to CP/M. Why? Good question. Perhaps it has more to do with the intangibles -- faith, acceptance, quiet devotion, and a brand of grass-roots, seat-of-the-pants wisdom. The CP/Mer assumes a tongue-in-cheek posture; all the while, watching the rest of the computer world spin its wheels in a deluge of glitter, gloss and Madison Avenue hype. It appears that regardless of the odds, in an era of disposable widgets and planned obsolescence, in the face of such staunch and resolute adversity, CP/M is here to stay. In the words of one PIPMAG author, Jim Taylor, CP/M is 'the peoples operating system'.

It is interesting that in spite of CP/M's continued popularity, the name 'CP/M' has been interpreted to mean several things: The CP/M Bible refers to it as 'control program for microprocessor'. In one Digital Research, Inc. CP/M Users Manual dated 1979, it is referred to as 'control program for microcomputer'. Still, I have heard it referred to as 'control program monitor'. These discrepancies of course only add to the mystique. I guess any CP/Mer worth his salt has read the basics, but just where DID CP/M come from? Where does it fit into the computer world as we enter the 1990s?

Perhaps the largest MAINFRAME computer, and the grand-daddy of them all, was the UNIVAC computer of 1950. Built entirely of vacuum tubes, it was so big it filled a large room. And the PDP-8 MINICOMPUTER, costing a mere $50,000, brought computing to corporate business in the mid-60s. But it was the advent of the microprocessor that made the MICROCOMPUTER a reality.

In 1969, engineers at Datapoint Corporation of San Antonio, Texas, designed a simple central processing unit and contracted both Intel and Texas Instruments to implement the design on a single piece of silicon. Intel succeeded, but the product ran ten times slower than had been agreed upon, so Datapoint backed out of the deal. Intel decided to go ahead and market the device and called it the 8008 microcomputer Ä40. Little did they know that this humble device signaled the beginning of the microcomputer revolution.

The 8008 was truly limited, and was used as a controller such as in robot stepper motors and push button TVs. The 8080, more-or-less its direct descendent, was the first microprocessor with an instruction set powerful enough to do real data processing. But these were still humble beginnings. In 1974, a microcomputer kit was sold to hobbyists called the Altair 8800. It featured switches to enter data, and LEDs as a monitor device. There was no keyboard, no CRT, and no floppy drives, which meant no stored programs. All programs had to be tediously entered by hand, byte by byte. This was indeed the stone age of microcomputing. They say that necessity is the mother of invention -- thus the birth of an operating system. CP/M was originally developed by Gary Kildall in 1973 for testing 8" Shugart disk drives. Later, in 1976, he founded Digital Research, Inc. and released CP/M, targeted at 8080 microcomputer hobbyists Ä50. But the 8080 was slow, and needed three different voltage levels and several support ICs to operate. It was soon replaced by the 8085 which required only +5 volts to run. Then, later came the Z80. It was not

only 8080/8085 software compatible, but included many additional, more powerful instructions. By 1980, with the Z80, CP/M had truly become the industry standard.

This is where Big Blue enters the picture. Microcomputers by this time were starting to find use in the business world. IBM realized the vast commercial potential and was gearing up for a major onslaught. They had been associated with quality mainframe computers, and most experts agreed that they would become the leader in the microcomputer field -- in essence, they would decide the future of microcomputing. IBM chose the 8088 microprocessor by Intel, and with Microsoft, they developed a new operating system called PC-DOS (essentially the same as MS-DOS -- Microsoft's stand alone version). Today, PC doesn't just mean personal computer -- it means IBM compatible.

For the uninitiated, let's take a look inside these machines and see what all the fuss is about. The 8088/8086 microprocessor, which is used in the PC-XT and its clones, is vaguely similar to our 8080 (the NEC V20 version can run 8080 instructions). It has an 8-bit data bus but 16-bit internal architecture and is considered a 16-bit microprocessor. The 8086 is an upgrade with a full 16-bit data bus. The major feature of the 8088/8086 is the ability to address up to 1M of RAM using 'segment address-extension' registers. This method is similar to CP/M 3.0's method of using multiple banks, only in the 8088/8086, this technique is used to the fullest. Much like the 8080, the 8086 can only use 64K at a time, but it has 10 banks of 64K each. This makes a total of 640K of RAM. By using other tricks it can access many megabytes of additional memory.

Another member of the 8086 family is the 80286. It provides 12 Mhz operation, a crude form of multi-tasking (running several SMALL programs at once) and is used in the PC-AT and its clones. The 80386 is another upgrade which provides up to 33 Mhz operation, a full 32-bit bus and addressing up to 4 Gigabytes of memory. And the 80486 is yet the latest upgrade, which provides further streamlining with a built-in cache memory controller and a coprocessor to speed up floating point calculations. These microprocessors are currently considered the 'leading edge technology' and are the darlings of the micro world Ä60. But even now, there is talk of MS-DOS being replaced in the 1990s with UNIX, a true multi-user/multi-tasking operating system.

There is another class of super microprocessors, such as the Motorola 88000. These new devices use RISC technology (RISC -- 'Reduced Instruction Set Computer'). They provide even faster execution by implementing a more efficient set of instructions and executing these within one machine cycle. Whew!

Okay, so what does this mean for CP/M? The microcomputer, which started out as an experiment and a boon for hobbyists has now become the workhorse of the corporate business establishment. The 80286-386-486 explosion is being fueled by big business, where money is no object and there is a constant demand for ever more speed and efficiency. Through CAD (Computer Aided Design), these powerful microcomputers are now being used by scientists and engineers. By connecting them together with LANs (Local Area Networks), some experts predict microcomputers will be direct competition for the minicomputer market which has already taken a beating in recent times. But while these advances are impressive, it leaves one piece of the puzzle missing. What about the consumer? It is true, most people can afford a simple XT. What does that mean really? Today, most GOOD software written for MS-DOS is expensive. And to enjoy the fruits of the leading edge, you have to constantly upgrade your equipment. The latest software offerings for PCs aren't designed to run on two floppies -- a hard drive is MANDATORY. A VGA monitor alone can cost more than an entire monochrome XT package. Getting caught up in this spiraling staircase of madness is frightening and can lead to gross over-spending and perhaps even what I like to call 'PC-phobia' -- the fear of owning an inferior computer. In reality, the PC revolution could be thought of as a plush banquet, where Big Business is picking up the tab, and the gamers are picking up the crumbs, hanging on to Sugar Daddy's coat tails.

The dream among countless, now defunct computer manufacturers in the early 80s of fully automated lifestyles with a computer in every home was just that, a dream, and one that I fear will not soon be realized. Let's face it, using a computer requires some effort, some learning on the part of the user. You can't just turn it on like a VCR or a Nintendo. Even the 'point and click' fad is too strenuous for most of the general public. I venture to say that there is NO well defined home computer market. All the non-IBM/clone platforms are beginning to fall away; ie: Commodore C-series, Atari, even some Apple II-series. The only true home computer market is really the COMPUTER HOBBYIST, and CP/M is still the ideal operating system for such a phenomenon.

It is conceivable that a CP/M supercomputer could be designed to have many of the bells and whistles of MS-DOS. The HD64180 is an integrated Z80 microcontroller with several built-in features: 10 Mhz operation, two DMA controllers, two serial ports, and a programmable timer. But the most notable feature is the ability to address up to 1 Meg of memory using a built-in MMU (Memory Management Unit -- similar to the 'segment address-extension registers' in the 8086 family, previously mentioned). A SCSI bus (Small Computer System Interface) could be used to interface up to eight devices; ie: hard drives, floppy drives, tape drives, printers, etc. This system could have high-resolution bit-mapped graphics, and perhaps even some form of multi-tasking (MP/M-II). The only problem is that CP/M as we know it has no provisions for handling graphics, let alone 1 Meg of memory! Alas, I have found myself fantasizing at times about such a machine. I have personally designed several 68000 microprocessor-driven systems, and it would not be exceptionally difficult to implement the 64180. Of course, outside of a few crazies like myself, such a machine would have no commercial value in light of the current technologies. It would merely be one CP/Mers dream of power and glory. (NOTE: anyone out there who has similar ideas of building a CP/M supercomputer or who knows of such a beast, please drop me a line via GEnie mail. ÄNote - I posted asking about that large multi-user CP/M system in the Netherlands a few days ago, I'd like to pass that information on to Rob Coleman -- dg0

Anyway, what makes CP/M so special? Well, to begin with, it was the first successful disk operating system. Living life as a microcomputerist before CP/M was akin to living without indoor plumbing. And to think that a boni fide computer system like the UNIVAC, that once filled an entire room, could now sit on a table top, I find astonishing. And further, the BDOS/BIOS concept is ingenious. The compatablitily of CP/M operated systems allows the user to share the experience with dissimilar hardware platforms -- thus a whole computing community is enriched. And even more, anyone who lives in the MS-DOS world can feel comfortable with CP/M. After all, MS-DOS

really grew out of CP/M; many of the commands and file handling schemes are similar. But more than anything else, maybe SIMPLICITY is what makes CP/M so appealing. As in Art, there is a beauty in simplicity and symmetry, combined with functionality. And CP/M is just that, a simple, straight-forward solution to operating a personal microcomputer system.

I guess personal computing is different for others. There's Dan-the-man, my snippidy next door neighbor, the one with the super AT with VGA. I have been to his house to marvel over his computer before, and it IS a remarkable machine. It would be ridiculous to ignore the fact that this is no doubt a far superior machine to anything in the CP/M world. The speed is overwhelming, the graphics dazzling. It is awe inspiring, the shear, unbridled power! (Excuse me, I get excited about computers -- any flavor.) Dan-the-man is indeed one happy camper, as well he should be. But I am baffled. What I find unsettling is the way he PERCEIVES personal computing.

Case in point: No sooner did he get this amazing machine, then he was planning to buy another, more powerful one. Why? I'm not sure. Is it the 'first on the block' syndrome, or is it that he really needs that kind of computing power? Dan-the-man plays many computer games and experiments with various software packages. Now six months later, he's ready for a new machine. He hasn't even learned how to use THIS one yet. Speaking of falling in love with one's computer; Dan-the-man not only has no loyalty, he is a technological flirt! In this relationship, there seems to be something missing. Hmmm...

Anyone who would care to notice, would be astounded at the longevity of the Commodore 64 computer. True, this little machine is edging its way to obsolescence, but what a journey it has seen! Nearly eight years after its debut, there is still a large group of devotees. There are, as a rough guesstimate, about seven or eight million of these machines around the world. Not all gamers either I remind you; there are multitudes of application programs that have been written for it. With its handful of ROM-based kernel calls and its graphics capabilities, the C64 is a veritable programmer/hacker's delight. How many imaginations have been stirred by these 8-bit wonders? How many computer software careers do you suppose have been launched by these little 'toy' computers? I have read articles in some trade magazines by professional programmers who STILL enjoy hacking on them. These people really love the little computer. In this relationship, there seems a great deal of dedication. Obviously, personal computing means different things to different people.

Meanwhile, back in my den, whoosh-tic! Both drives came to a halt. I read the directory of the backed-up floppy. Everything looked good. I wondered, should I check for bad sectors? But then again, the format command should have detected any of these. I yawned. Now I yearned for sleep. Inadvertently, I ran scan.com to look for a bad sector, only I entered the A drive by mistake. I cringed as I realized that my boot disk was now being scanned for bad sectors. This is NOT what I had intended. So much for working when I was tired. But behold! There was a bad sector detected on my boot floppy! I glared at the message on the monitor with disbelief. I thought to myself, maybe the A drive was overheating. That had to be it. I ran scan.com again with the boot disk in the B drive. Again -- a bad sector -- the same one. I decided to fix the problem NOW. No matter how tired I was, I couldn't go to sleep having my boot disk corrupted. No way. That would be sacrilege.

I pulled out my back-up boot disk (yes, I had a back-up, but that is a different story!). I pip'd the contents to my RAM drive and then reformatted the corrupted boot disk. Outside my window, I heard a bird call, close in, near the house. I looked briefly to the open window, then back to the task at hand. Once again, I ran scan.com. Good! No bad sectors. Patiently, I waited as the RAM disk contents were copied back onto the boot floppy.

'Booting' -- what an odd term. In the early 1970s, practically all memories were magnetic core types. These memories had tiny toroid-shaped magnets spaced between a lattice-work of wires. The direction of the current through one wire determined the logic value, and current through the other determined whether that value was read or written into the magnet-element. These devices were non-volatile, and held their contents even when the power was removed. But when this data was corrupted, a small program had to be entered to load the system code. This was a tedious process in those days -- done either with switches or paper tape input. The solution? A small loader program was entered manually which loaded another program which actually loaded the system code. This small loader program was called a 'bootstrap' loader. The system was loaded, or pulled up by its own bootstraps. Hence the term -- 'boot' A70.

Whoosh-tic! The A drive came to a stop. I then rebooted. It was done. All was fine in the world. Again, there was the bird outside my window. I got up and walked over to have a look. Again, I heard the cooing. It came from the bushes near the tree out front. It was really too dark to see. Up in the sky, I could see the crescent moon hanging lazily in the heavens, amidst a slight veil of wispy cloud. I listened and heard silence. In the dim light, I glanced back at my computer; my reconditioned boot disk. Yes... all WAS fine in the world. I was ready for sleep.

It has been said that perhaps the true delight is not in arriving at the destination, but in the journey itself; to occasionally stop and enjoy the view. If even just to simply smell the fragrance of flowers, or ponder over a private thought. I, myself, find the personal computing experience to be just that -- PERSONAL. It is a quiet time alone, away from the stress and strain of everyday life. And it is during these times that I learn about strength and weakness, success and failure; the value of work; and play for the sake of play; and ways of acceptance. Perhaps it is in this process of DOING that I find the real reward. Knowing the joy of getting a five page, hand typed program to finally run. Learning to accept that slow, noisy drive. Learning that there is always another corner to turn, another page to read. Learning that maybe there is wisdom in discovering even just one little, simple thing each day. If only for this, it has been worth the while. Oh yes, it could be model cars, or stamp collecting, or flower arranging. But it is when these things cease to be mere objects and become vehicles, taking us someplace else, to some secret, private place, they bring us closer to the inner self. By pouring over the tired guts of my old CP/M computer, cursing myself for having misplaced my screw driver AGAIN, perhaps I learn a little more about myself.

References / Odds and Ends

1. Zen and the Art of Motorcycle Maintenance, Robert M. Pirsig, Bantam. This book is not about fixing motorcycles, or even CP/M, but I strongly recommend it.

2. The Soul of CP/M, The Waite group, Howard Sams. Out of print, but a real gem for programmers. I have read excerpts from some old copies of Computers and Electronics magazine, but I don't own a copy myself. I'm still looking but to no avail. (If anyone knows of an available copy please let me know.)

3. The CP/M Bible, Waite and Angermeyer, Howard Sams. Out of print. This book is useful for someone running several versions of CP/M. It includes a handy cross reference section for 1.X, 2.X, and 3.X systems. I found a NEW hard-bound copy in a used book shop for $4. Note: Many CP/M computer books are now out of print but can be found at 'fire-sale' prices in used book stores. With little money, you can build an impressive CP/M library.

4. An Introduction to Microcomputers, Vol. 1 Basic Concepts, Adam Osborne, Osborne/Mcgraw Hill. A good book for newcomers or anyone who wants to know generally how a microcomputer works.

5. TC128 Compendium #1, Issue #2, CP/M Update, by Todd Madson. Any one who uses a C128 should subscribe to Twin Cities 128 magazine. It deals ONLY with the Commodore 128. But I must say, I wish there was more CP/M coverage.

6. EDN magazine, #24, Nov. 23, 1989. The Annual uP/uC directory. This magazine is written for professional electronic design engineers and is a good source of info on new technology.

7. CP/M Assembly Language Programming, Ken Barbier, Prentice Hall. As far as I know, this book is out of print. However, it is an excellent tutorial on CP/M system programming with the 8080 instruction set.

----- About the Author -----

Robert Coleman is an electronics engineer for Data General, Telecommunication Products Division. Besides designing microprocessor-based electronic circuits, and fiddling around with CP/M in his spare time, he enjoys reading, writing and playing bass in a part-time rock 'n' roll band.

Ausgabe

a) Kosten, die bei weitem einen Jahresverdienst überschreiten und die zur Anschaffung einer Computerausrüstung notwendig sind.
b) Die – mehr oder weniger – freiwillige Herausgabe von Daten auf einem Bildschirm oder Drucker, mit denen ein Computer vergeblich versucht, die unter a) gemachten Ausgaben halbwegs zu rechtfertigen.

Auflösung

a) Zustand, in dem sich ein User befindet, nachdem er festgestellt hat, daß es möglich ist, mit einem kleinen Befehl eine große Festplatte zu löschen. Der Auflösung des Users folgt – falls vorhanden – ein handfester Streit mit der Ehefrau oder – falls vorhanden – ein heiteres Haareausraufen.
b) Je größer die Anzahl und je kleiner die Punkte sind, aus denen eine Bildschirmgrafik zusammengesetzt ist, desto besser ist die Auflösung und damit der optische Eindruck. Wann werden dies die Hersteller begreifen?

Die Public Domain - Ecke

FATCAT

Die Leute mit Festplattenlaufwerken werden immer zahlreicher, und eine Platte mit 20 MByte ist heute nichts besonderes mehr, auch nicht unter CP/M. Hinzu kommt die Fülle von Software, die auf hunderten von Disketten ihr magnetisches Dasein fristet. Einen Überblick über diese Sammlungen zu bekommen und zu pflegen, ist die Aufgabe eines Disk-Katalog-Programmes. FATCAT ist so ein Programm, und es ist, wie immer in dieser Ecke, ein PD-Programm.
In der neuesten Version 2.4 ist FATCAT sehr schnell und intelligent, Nach dem Aufsetzen verschiedener Parameter muß man nur noch die Disketten einschieben, die in den Katalog aufgenommen werden sollen. Erst wenn alle Disketten gelesen sind, beginnt der zeitaufwendige Teil des Sortierens. Das ist sehr anwenderfreundlich. Andererseits verlangt FATCAT auf jeder Diskette eine Datei mit einem Namen, der einer speziellen Syntax genügen muß. Diese Dateien müssen vor oder während des Katalogisiervorganges angelegt werden. Das kann etwas bremsen.
FATCAT ist in der Lage, .LBR Dateien zu lesen und die einzelnen Mitglieder der Libraries in den Katalog aufzunehmen. Ein sehr hübsches Feature!

Die eigentliche Stärke von FATCAT liegt im Wiederfinden von Dateien. Wie war das doch gleich... ich hatte doch mal ein DBASE-Programm... aber wo?

FATCAT erlaubt es, Bereiche anzugeben, in denen gesucht werden soll, also z.B. die Disketten 100 bis 199. Und, FATCAT-Suchbegriffe sind rudimentäre reguläre Ausdrücke, es sind also Wildcards erlaubt. Die Suche nach der obigen Datei gestaltet sich also, was den Suchbegriff angeht, vielleicht so : "*.CMD".

FATCAT ist in Turbo-Pascal geschrieben und kommt ohne Quellen, muß für das System konfiguriert werden. Das ist erst nach Studium der DOCs möglich. Die Bedienung ist sehr einfach, wenn man nur die Grundfunktionen benötigt, für spezielle Sachen kann es etwas aufwendig werden. FATCAT Version 2.4 gibt es beim CP/M Bibliothekar, es benötigt CP/M ab Version 2.2.

BUSH

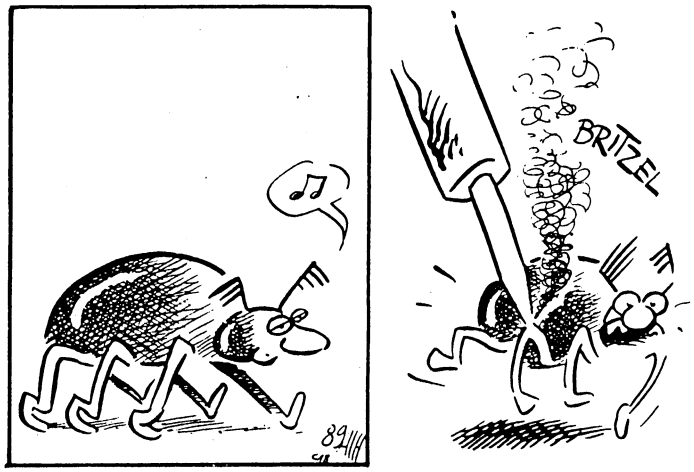
BUSH ist eine Z-Utility, benötigt also ZCPR3x. BUSH ist die Abkürzung für BackUpSHell, also eine Oberfläche, mit der Backups von Disketten oder Festplatten schnell und komfortabel gemacht werden können. Auch hier reduziert sich die Aufgabe des Bedieners weitgehend auf das Einlegen der Disketten.
BUSH muß konfiguriert werden, speziell die Druckersteuerung, denn BUSH kann ausgefeilte Reports erstellen, damit man nach dem Backup auch weiß, was wo steht.
Interessant ist folgendes: man kann BUSH mitteilen, das gewisse Filetypen, wie .BAK, .SYM usw. nicht berücksichtigt werden sollen, und man kann Files in ARChive zusammenfassen lassen.
BUSH arbeitet recht flott, wenn man bedenkt, das nur DOS-Calls verwendet werden, um ein Backup zu erstellen. Der Vorteil liegt darin in der leichten Zugreifbarkeit auf Backups, da sie als Files abgelegt werden, kann man sie ohne BUSH wieder retrieven. Das muß man allerdings auch manchmal. Mir ist es bisher nicht gelungen, ein komplettes Festplattenbackup einer Partition wieder von Disketten auf die Platte zu schaffen. Laut DOCs ist es möglich, alle USER abzuklappern, aber es funktioniert nicht. Bleibt nur die CP/M-Methode mit PIP oder sonstwas.
Aber ein Backup ist ja in erster Linie eine Sicherheitskopie, und die legt man sicher weitaus öfter an als man sie benötigt. Und dafür ist BUSH sehr gut zu gebrauchen.

BUSH ist in Z80-Assembler geschrieben (keine Quellen) und benötigt CP/M 2.2 oder höher und ZCPR3x.

PACK

Zum Schluß noch ein kleines Prögrämmchen, das wahre Wunderdinge leistet: PACK. Es handelt sich um einen Disk-Arranger, also ein Programm, das die Fragmentierung von Disks/Festplatten aufhebt oder zumindest verringert. Bei CP/M ist das sehr wichtig, eine Festplatte, die lange nicht mehr "aufgeräumt" wurde, ist wesentlich langsamer als eine frisch formatierte, auf der die Files noch alle schön in einer Reihe stehen. Darüber hinaus ist der Zugriff auf Dateien, die im Directory weit vorn stehen, schneller, denn CP/M 2.2 sucht linear. PACK erlaubt, eine Prioritätenliste anzugeben, mit der die Files arrangiert werden, so können z.B. oft benötigte Utilities an den Anfang der Platte geschoben werden. Ich benutze PACK regelmäßig, denn merke: Je öfter PACK gestartet wird, desto schneller ist es fertig!

PACK ist in Z80-Assembler geschrieben (keine Quellen) und benötigt CP/M 2.2 oder höher sowie ZCPR3x.



Kompression von Dateien unter CP/M.

Als CP/M Diskothekar bekomme ich immer wieder Anfragen, was diese seltsamen Files mit der Extension .LBR, .ARK oder .DZC wohl bedeuten mögen. Dabei handelt es sich um komprimierte Dateien. Man unterscheidet zwei prinzipielle Arten der Kompression:

1. Datei-Kompression
2. Daten-Kompression

zu 1).

Datei-Kompression bedeutet, daß mehrere Dateien in eine einzige zusammengefaßt werden, um Directory-Slots und Diskettenplatz zu sparen. Unter CP/M gibt es zwei bekannte Verfahren : Libraries und Archive. Zunächst zur Motivation.
Bekanntlich vergibt CP/M Diskettenplatz nicht in Sektoren oder gar byteweise, sondern in Blöcken (Groups). Ein Block kann unterschiedliche Größen haben, üblich sind Werte von 1 KByte bis 8 KByte. Im letzten Fall würde eine Datei, die nur aus einem einzigen Byte besteht, 8 KByte Diskettenplatz und einen Directoryeintrag verbraten. Das ist nicht sehr effektiv. Wenn nun mehrere Dateien, die überdies vielleicht auch noch logisch zusammenhängen, wie etwa ein Compiler nebst Overlays, Linker und Assembler, zusammen gespeichert werden sollen, organisiert man sie in einer Library bzw. einem Archiv. Man kann natürlich alle möglichen Dateien in einer LBR oder einem Archiv zusammenfassen, z.B. um ein Backup einer Diskette herzustellen (was allerdings auf Speicherprobleme stoßen dürfte).
LBRs und ARKs haben ihrerseits ein Inhaltsverzeichnis, das die Größe und den Offset einer Datei in dieser Library angibt. Der Vorteil liegt auf der Hand : Es wird nur noch ein Directoryeintrag verbraucht und der Diskettenplatz wird ökonomischer verwaltet, denn in der Library nimmt jede Datei nur den tatsächlich beanspruchten Platz ein. Dasselbe gilt für Archive. Libraries sind an der Extension .LBR zu erkennen, Archive heißen entweder .ARK oder .ARC. Erstere sind unter CP/M zusammengestellt, letztere unter MSDOS, aber beide können unter CP/M ausgepackt werden.
Sowohl Libraries als auch ARKs lassen u.U. das Einfügen von zusätzlichen Dateien zu.

Standard-Utility für LBRs ist NULU.COM, für ARKs UNARC.COM. Zum Herstellen von Archiven ist neuerdings ARK Version 1.1 erhältlich, ein CP/M-Hacker namens Tillmann Reh aus Siegen hat eine Version 2.0 erstellt, um die ich mich bemühe. Es hat lange (ca. 2 Jahre) gedauert, bis die erste ARK-version verfügbar war, damals noch Version 0.7, voller Bugs.

Es gibt zunehmend auch noch andere Archivierungsmöglichkeiten, die unter CP/M Verbreitung finden, so zum Beispiel ZIPs oder ZOOs. Im Moment ist ihr Aufkommen jedoch noch sehr gering.
Die Extension .LBR ist NICHT mit .LIB zu verwechseln! Eine LIB ist normalerweise keine Library, sondern z.B. ein INCLUDE-File für einen Assembler. Auch TURBO-Modula-2 benützt LIBs, dabei handelt es sich tatsächlich um Libraries, allerdings nicht im LBR - Format.

zu 2).

Daten-Kompression ist die Reduktion von Dateien, d.h. die Entfernung redundanter Daten. Solche komprimierten Dateien sind am mittleren Buchstaben ihrer Extension zu erkennen, das ist ein Q, Z oder Y.
Kanditaten für DatenKompression sind alle Dateien, speziell Texte und vor allem Grafik-Daten. Dabei werden nicht selten Reduktionen von 50 % erreicht.
Ich weiß nicht genau, welche Algorithmen bei diesen Kompressionsmethoden benutzt werden. Es gibt eine Reihe verschiedener Algorithmen : Run-Length-Encoding, Squeezing, Crunching (GEL und LZH), Packing, Squashing...
Unter CP/M sind drei verbreitet.

a) SQUEEZING. Solche Dateien tragen in der Extension den Buchstaben Q. Viele Utilities erlauben das Entsqueezen von Dateien: WASH, NULU u.a.

b) CRUNCHING (GEL). Eine etwas effektivere Methode. Zu erkennen am Z in der Extension. Utility : UNCR.

c) CRUNCHING (LZH). Das Neueste. Die letzten Uncruncher erkennen die Methode und behandeln die Datei entsprechend. Zu erkennen am Buchstaben Y in der Extension.

Die neueste Utility, UNCRUNCH.COM Version 1.1, behandelt GEL, LZH und gesqueezete Dateien gleichermaßen.
Archive als auch Libraries enthalten in der Regel komprimierte Dateien, die nach dem Auspacken zu expandieren sind. Neuere Programme, z.B. LBREXT17.COM, expandieren automatisch.

Unter ZCPR33 gibt es eben dieses LBREXT17, ferner LPUT, LGET und VLU. Damit ist das Arbeiten wesentlich angenehmer. VLU ist z.B. Screen-orientiert und erlaubt das selektive Binden von Dateien in eine Library, außerdem das Betrachten von Library-Members, auch, wenn diese gecrunched oder gesqueezed sind.

Rüdiger.

Der Editor.

Der Editor heißt PMATE. Es gibt ihn für CP/M, MSDOS und CP/M 86. Er kann alles.
PMate ist unter CP/M (und diese Version, genauer gesagt 3.02, beschreibe ich hier) ein ca. 20 KByte großer COM-File. Das ist recht viel, aber PMate kann Texte verarbeiten, die länger sind als der vorhandene Speicher.
Wenn man PMate startet, sieht man eine recht spartanische Infozeile, in der der Name der Arbeitsdatei und die Cursorposition steht, ab und zu tauchen auch noch andere Meldungen auf. Unter der Infozeile ist die Kommandozeile, denn PMate ist ein Kommando-orientierter Editor.
Ein Kommando besteht aus einer Zeichenfolge, die mit der Eingabe zweier ESC beendet wird. Das ESC wird als '$' dargestellt. Das Kommando :

```
10m$$
```

bewegt den Cursor 10 Zeichen nach rechts. Um den gleichen Betrag nach links verrücken läßt er sich mit

```
-10m$$
```

Ganz logisch. Eine Zeile nach unten : 1$$. eine Zeile nach oben : -1$$. Zehn Zeilen zurück : -10l$$. Ganz klar. Lösche das Zeichen under dem Cursor : d$$. Füge einen String an der Cursorposition ein : idas steht jetzt hier$$. Natürlich kann PMate auch suchen und ersetzen. smate$$ findet den String MATE, mate oder MaTe, cmate$matt$$ ersetzt jedes Vorkommen von MATE durch den String matt. (Das $ steht immer für ESC).
Im Gegensatz zu WordStar-like Editoren sind die meisten PMate-Kommandos nicht an eine Taste bzw. Tastenkombination gebunden, sondern sie müssen als Kommando eingegeben werden. Dazu gibt es neben Insert- und Overwrite-Modus noch den Kommandomodus. Hier eine kurze Aufstellung der an Tasten gebundenen Kommandos mit Erläuterungen (^A = CONTROL-A):

Cursortasten	bewegen den Cursor
^A	zum Anfang/Ende der Datei
^D	lösche Zeichen
BS	lösche Zeichen vor dem Cursor
^P	Wort rechts
^O	Wort links
^Q	lösche Wort links
^W	lösche Wort rechts
^U	Seite zurück
^B	Seite vor
^T	Setze Marke
^E	lösche von Marke bis Cursor
^R	hole vom Garbage Stack
^C	Breche Kommando ab
^X	Kommando Modus
^V	Overwrite Modus
^N	Insert Modus
^L	Neue Zeile
^Y	lösche Zeile

Hoffentlich habe ich nicht viel vergessen. Alle diese Tasten sind konfigurierbar, die Liste ist die Belegung meines PMate. Es kommen noch hinzu einige Tasten-Kommandos zur Text-Formatierung, aber PMate ist kein Textverarbeiter, sondern ein Editor, hier liegen seine starken Seiten.

PMate hat 11 Buffer, sie heißen T und 0 bis 9. Buffer T ist der Text-Buffer, mit ihm wird normalerweise gearbeitet. Aber man kann auch in die anderen Buffer Texte laden und sie modifizieren und speichern, auch zwischen den Buffern hin- und herschieben. Buffer 0 spielt eine besondere Rolle: hier wird abgelegt, was bei einem ^E gelöscht wurde. Deshalb kann man auch die WordStar Block-Operationen ^KB, ^KK, ^KV und ^KC nebst ^KY nachbilden. Wenn man will. Die Buffer 1 bis 9 dienen normalerweise als Hilsfbuffer für komplexe Kommandofolgen.

Die Kommandosprache von PMate ist unglaublich mächtig. Man kann damit regelrechte Programme schreiben, mit Arithmetik, Bedingungen, Wildcards, Schleifen, Breakpoints, Variablen, Ein/Ausgabe. Wenn ein solcher Kommandostring in einem der Buffer 0 - 9 steht, kann er als "Macro" aufgerufen werden, indem einfach ein ".", gefolgt vom Buffernamen, eingegeben wird. Ein Beispiel : Buffer 2 enthält den folgenden Text:

```
[M(@T=" )]
```

Das ist eine Kommandofolge. Sie könnte auch in der folgenden, kommentierten Form dort stehen:

```
[               ; Anfang Schleife
M               ; Cursor nach rechts
(               ; Bedingung
@T              ; Zeichen unter dem Cursor
=" )            ; gleich Leerzeichen/Ende Bedingung
]               ; Ende Schleife
```

Gehen wir zurück in den T-Buffer mit dem Kommando bte$$ (Buffer T Enter). Mit .2$$ wird nun das Programm ausgeführt. Solange das Zeichen unter dem Cursor kein Space ist, wird der Cursor nach rechts bewegt. Dann wird die Schleife und damit das Programm verlassen, der Cursor steht nun auf dem nächsten Leerzeichen hinter seiner Startposition.
Einfaches Drücken der ESC-Taste führt das Kommando nochmals aus. Sehen wir uns ein anderes, nützlicheres Macro an.

```
a               ; go to start of buffer
{               ; start loop ({ und [ ist beides m!glich)
(@t=";)                 ; if @t = ';'
[               ; then
k               ; kill rest of line
13i             ; insert chr(13)
1qr             ; update screen
]               ; end then
[               ; else
m               ; move 1 char
]               ; end else
}               ; end loop
```

Dieses Macro wird mit einem Fehler beendet, wenn das Ende der Datei erreicht wurde. Fehler können abgefangen werden. Man kann, neben der Schleifenkonstruktion, die if-then-else

Bedingung erkennen. Das Macro durchsucht die Datei nach dem Semikolon. Wenn es eines findet, wird ab dieser Position die Zeile gelöscht und ein <CR> wieder eingefügt. Sonst wird der Cursor auf das nächste Zeichen positioniert. Auf diese Weise kann ein Assemblerprogramm oder ein PMate-Macro von Kommentaren befreit werden. Ein '?' in einem Macro stellt einen Breakpoint dar, der Programmlauf stoppt und man kann im single-step durchlaufen.
Es gibt drei Schleifentypen : REPEAT_UNTIL ,WHILE_END und die Endlosschleife. Labels können definiert und, auch bedingt, angesprungen werden.

Im Zusammenhang mit Bedingungen ist es nützlich, wenn man PMates Variablen kennt. Hier sind einige:

```
'@0-@9      : Benutzervariablen, numerisch.
 @b         : der aktive Buffer
 @e         : Error-Flag
 @l         : Zeilennummer
 @x         : Spaltennummer
 @t         : das Zeichen unter dem Cursor
 @m         : der freie Speicher
```

Einige Operatoren :

```
 +          : Addition
 -          : Subtraktion
 *          : Multiplikation
 /          : Division (integer)
 !          : ODER
 &          : UND
 '          : NICHT
```

Die behandeleten Macros können natürlich als Datei gesichert und bei Bedarf geladen werden, man kann sie aber auch "permanent" machen. Dazu hat PMate eine seltsame Eigenschaft: er kann sich selbst klonen. hat man während einer PMate-Sitzung einige Installationen verändert, so sagt man einfach:

```
XDNEUPM.COM$$
```

und es gibt einen neuen PMate mit Namen NEUPM.COM auf der Disk. Das könnte z.B. ein PMate mit WordStar-Eigenschaften sein.

Die Kommandos, die PMate beherrscht, sind bei weitem zu zahlreich, als das sie hier aufgeführt werden könnten. Zusammen mit der mächtigen Macro-Sprache läßt sich PMate vermutlich dazu bringen, C-Quellcode in Assembler zu übersetzen. Ich habe ein Macro, das mit zwei Argumenten aufgerufen wird und den Textbuffer auf "matching parenthesis" untersucht, das können Klammerpaare sein, wie '[' und ']', aber auch Schlüsselworte wie 'begin' und 'end' oder '(*' und '*)'. Ein Pascal-Programmierer weiß sowas zu schätzen.

Habe ich etwa übertrieben?

Rüdiger

Impressum

1. Vorsitzender — Fritz Chwolka
Tel.: 0 24 64/89 20 — Saarstraße 4
5173 Aldenhoven

2. Vorsitzender — Gerald Schröder
Tel.: 0 41 05/26 02 — Am Schützenplatz 14
2105 Seevetal 1

Hardewarekoordinator — Andreas Magnus
Tel.: 02 09/87 02 30 — Pommernstraße 4
4650 Gelsenkirchen

Newdos-Diskothekar — Oliver Volz
Tel.: 07 11/7 35 38 17 — Waldburgstraße 73
7000 Stuttgart 80

CP/M-Diskothekar — Rüdiger Sörensen
Tel.: 0 61 31/3 28 60 — Thomas Mann Straße 3a
6500 Mainz 1

Clubbücherei — Kurt Müller
Tel.: 0 41 52/7 06 43 — Sophie-Scholl-Ring 38
2054 Geesthacht

Redaktion — Jens Neuder
Tel.: 07 91/4 28 77 — Rudolf-Then Straße 32
7178 Gschlachtenbretzingen

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

Bankverbindung des CLUB 80
Postgirokonto Sonderkonto CLUB 80
Obermann H. 6209 Heidenrod
Konto Nr. 496 071-605 Postgiroamt Frankfurt BLZ 500 100 60

Eine Zensur oder Kontrolle der INFO-Beiträge erfolgt nicht.
Die Redaktion.

Schluß

Hallo Club 80'er,

wieder einmal ist ein Jahreswechsel in greifbarer Nähe und somit auch die Zeit jedem das Beste für die Feiertage und einen erfolgreichen Sprung nach 91 zu wünschen. Gleichzeitig möchte ich mich bei allen Autoren und Mithelfenden für die geleistete Arbeit bedanken.
Ohne Euch wäre die TRS-80-Welt fast ausgestorben. Ich hoffe, daß wir auch im nächsten Jahr wieder gemeinsam einige INFO's und Sonderhefte zum Druck und natürlich an das Clubmitglied bringen werden.

Mit diesem Info werden Euch diesmal zwei Sonderhefte erreichen. Zum Einen über GEGRAS eine Benutzeroberfläche -welche die HRG auf dem TRS-80/Genie und HRGPACK benötigt (Softwarepaket GEGRAS ist in der Club-PD)- und Genie 3s Z180 -ein Hard- und Softwaretunup für den 3s-.
Wir hoffen, daß Ihr Gefallen an den Werken findet. Es steckt ja doch recht viel Arbeit dahinter, bis so etwas entsteht. Die Autoren werden sich sicher über ein kleines "Feedback" eurerseits freuen. Vielleicht ist es auch eine Animation für den einen oder anderen gleiches zu tun, und mir ein Sonderheft zur "Endfertigung" zuzusenden.

Als weiteres möchte ich, um die Qualität des INFO's zu steigern, wieder einmal einen Hinweis loswerden und ein Angebot machen. Zuerst zum Hinweis:
Vermeidet bitte die Verwendung von Umweltschutzpapier und älteren Farbbändern! Wie Ihr -teilweise in diesem INFO- seht, leidet der Kontrast und somit die INFO-Qualität darunter. Ich bin natürlich auch für Umweltschutz und optimaler Ausnutzung der Energie, aber für das INFO möchte ich doch um "ungeknicktes in schwarz auf weiß" bitten.
Nun zu meinem Angebot: Wie Ihr sicher am Druckbild dieses Textes erkennen könnt, habe ich die Möglichkeit meine schwarzen Buchstaben mit dem Laser zu brennen. Wenn Bedarf besteht kann ich ja für Euch Euren vorgefertigten Text ausdrucken. Dies erfordert aber, daß ich von Euch den Text auf IBM-kompatiblen Scheibchen (peinlicherweise doch wieder DoMesTos) zugeliefert bekomme. Größe und Art der Bespurung sind egal, sofern sie sich auf normale IBM-XT/AT-Formate beschränken. Der Text sollte am besten unter Word erstellt werden, da ich hiermit arbeite, es geht auch Text im ASCII-Format.
Sollten besondere Wünsche beim Layout berücksichtigt werden, ist ein Probeausdruck als Muster für mich sicher sehr hilfreich. Zumindest brauche ich dann eine genaue Anleitung. Bilder und Grafiken sind bitte extra mitzusenden, da Word nicht jedes Grafikformat im Text mit verarbeitet.

Zum Abschluß dieses INFO's möchte ich mich bei Euch bis zum nächsten INFO oder Treffen verabschieden. Viel Spaß mit dem neuen Lesestoff wünscht Euch

Jens Neuber

Club 80 Mitgliederadressenliste

Detlef Behrendt	Schlosserbreite 1a	D-	8018	Grafing
Achim Benner	Vorm Mühlberg 1	D-	5910	Kreuztal 3
Helmut Bernhardt	Hafenstraße 7	D-	2305	Heikendorf
Heinrich Betz	St. Wolfgangstraße 13	D-	8551	Hausen
Jörg Brans	Tieloh 55	D-	2000	Hamburg 60
Harald Braun	Postfach 8011	D-	2300	Kiel 17
Ulrich Böckling	Jochaczstraße 61	D-	5410	Hör-Grenzhausen
Fritz Chwolka	Saarstraße 34	D-	5173	Aldenhoven
Oskar Drechsler	Duckterather Busch 2	D-	5060	Bergisch Gladbach 2
Hans-Joachim Eilers	Theodor-Heuss-Straße 129	D-	2900	Oldenburg
Richard Frey	Wolfsgrube 45	D-	3422	Bad Lauterberg
Werner Förster	Christoph-Krebs-Straße 9	D-	8720	Schweinfurt
Jens Günther	Bannerscheid 7	D-	5231	Neitersen
Hans-Günther Hartmann	Möwenstraße 9	D-	2876	Berne 2
Manfred Held	Stirner Straße 22	D-	8835	Pleinfeld
Werner Hentz	Am Tränkgarten 20	D-	6457	Maintal 2
Klaus Hermann	Forchenstraße 8	D-	7401	Pliezhausen
Matthias Homann	Hermesweg 21	D-	2000	Hamburg 90
Willi Johnen	Hansemannstraße 1	D-	5160	Düren
Jürgen Kemmer	Dorfberg 7	D-	8701	Sulzdorf
Mary Jo Kostya	Balberstraße 68	CH-	8038	Zürich
Eckehard Kuhn	Im Dorf 14	D-	7443	Frickenhausen
Claus Littmann	Plockhorst, Zum Spring 15	D-	3155	Edemissen
Walter Lorenz	Mahräckerstraße 9	D-	6000	Frankfurt /Main 50
Andreas Magnus	Pommernstraße 18	D-	4650	Gelsenkirchen
Herbert Mahlert	Baumschulstraße 7	D-	4100	Duisburg
Harald Mand	Kl. Flintbeker Straße 7	D-	2302	Flintbek /Kiel
Holger May	Marienstraße 9	D-	5768	Sundern 2
Heinz-Dieter Meklenburg	Amrumer Weg 1	D-	2262	Leck/NF
Christian Menk	Ollsener Straße 52	D-	2116	Hanstedt
Franz Mössel	Schafferstraße 12	I-	39012	Meran
Klaus-Jürgen Mühlenbein	Am Mönchgarten 28	D-	6940	Weinheim-Lützelsachsen
Kurt Müller	Sophie-Scholl-Ring 3b	D-	2054	Geesthacht
Gerhard Neebe	Märkische Straße 186	D-	4600	Dortmund 1

Club 80 Mitgliederadressenliste

Jens Neueder	Rudolf-Then-Straße 32	D-	7178	Gschlachtenbretzingen
Christof Neumann	Theodor-Heuss-Straße 8	D-	7265	Oberhaugstett
Stefan Nitschke	Germanenstraße 5	D-	7519	Walzbachtal 1
Hartmut Obermann	Mozartring 23	D-	8870	Günzburg
Bernd Retzlaff	Kleiner Sand 98	D-	2082	Uetersen
Gerd Rinio	Rennbahnstraße 9	D-	2000	Hamburg 74
Claus Ruschinski	Pommernstraße 21	D-	4370	Marl
Werner Schilling	Ehndorfer Straße 340	D-	2350	Neumünster
Alexander Schmid	St. Cajetan Straße 38/VII	D-	8000	München 80
Paul-Jürgen Schmitz	Bremer Straße 9	D-	6236	Eschborn
Rainer Schmitz	Dornierstraße 17	D-	7320	Göppingen
Frank-Michael Schober	Weberweg 2	O-	7590	Spremberg
Uwe Schoberth	Petrus-Waldus-Straße 14	D-	7136	Oetisheim
Frank Schoof	Elpke 5	D-	4800	Bielefeld 1
Horst-Dieter Schroers	Breslauer Straße 9	D-	8016	Feldkirchen
Gerald Schröder	Am Schützenplatz 14	D-	2105	Seevetal 1
Peter Schröder	Theodor-Fahr-Straße 32	D-	2000	Hamburg 62
Egbert Schröer	Joachimstraße 18	D-	4270	Dorsten 1
Andre Schut	Sanderstraße 26	D-	1000	Berlin 44
Wolfgang Schwarz	Schwedenring 6	D-	8850	Donauwörth
Walter Schäfer	Rathausstraße 4	D-	8160	Miesbach
Jörg Seelmann-Eggebert	Henri-Spaak-Straße 96	D-	5305	Alfter 4
Svend A. Soerensen	Bogholder Allee 76A	DK-	2720	Vanloese
Arnulf Sopp	Wakenitzstraße 8	D-	2400	Lübeck
Hans-Martin Stephan	Am Glasesch 9a	D-	4506	Hagen a. TW
Stefan Stumpferl	Hasenberglstraße 57	D-	8000	München 45
Rüdiger Sörensen	Thomas-Mann-Straße 3a	D-	6500	Mainz 1
Klaus Thieleke	Ahrenshooper Straße 47	O-	1093	Berlin
Wilhelm Tornow	Görlitzer Straße 16	D-	2190	Cuxhaven 13
Richart Vollkmer	Am Spörkel 69	D-	6700	Dortmund 50
Oliver Volz	Waldenburgstraße 73	D-	7000	Stuttgart 80
Michel Waccus	Mühlhofweg 2a	CH-	8266	Steckborn
Heinz Wittkamp	Hindenburgstraße 37	D-	5630	Remscheid
Hans-Otto Wulf	Im Brahmkamp 38	D-	4250	Bottrop

04723/1355