

Computer Unser

Computer unser, der Du bist in der Zentrale
Geheiligt sei dein Bildschirm
Deine Eingabe komme

Dein Wille geschehe wie im Speicher
So auch auf dem Drucker
Unsere täglichen Listen gib uns heute

Und vergib uns unsere Fehler
Obwohl wir nicht vergeben denen
die falsch programmiert haben

Lass uns nicht zu lange warten
Und erlöse uns von teuren Ausgaben
Denn Dein ist die Firma

Die Macht und das Personal
In Ewigkeit

Enter

CLUB INFO

33. AUSGABE

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

Inhaltsverzeichnis

Clubinternes

Nordlichtertreffen 90

Termine

Vorstellung

Autor & Seite

Kurt Müller 1
Redaktion 2
3 - 4
Uwe Schoberth, Gert Halgasch

Software

Expertensystem in Basic ???!

Masken generieren mit TRS 80 MI und GENIE I

Egbert Schröer 5 - 24
25 - 27
Hardcopyroutine für SEKOSHA GP 550A 29
Hardcopyroutine für BROTHER M 1009 30
Schnelle Kreisfunktion unter BASIC80 31
Grafikdemo für uLabs Grafikkarte (M IV/IVp/III) 32

Frank Schoof
Schönere Programme mit Turbo-Pascal 33 - 37
Kleiner Terminkalender 38 - 40

Alexander Schmid
WIPE 41 - 44

Uwe Schoberth
Unter falschem Namen ALIASes mit ZCPR 45 - 46
DU3 für CP/M Plus 47 - 48

Alexander Schmid
Adreßautomat, die Zweite 49 - 53

Arnulf Sopp
Modell IV/IVp-INFO's 55 - 56

Jörg Brans
LeSript Druckersteuerzeichen 57 - 58

Frank Schoof

Börse

Wer hat was -- wer will was

Sonstiges

TRX-280 The Next Generation of TRS-80

Club-Bücherei

Buchvorstellungen

Die letzten Seiten

Impressum

Schluß

Mitgliederadressenliste

M IV/IVp Sonderinfo

Autor & Seite

59 - 60

61 - 64

65 - 66

Kurt Müller

67

68

Redaktion

am INFO-Ende

Beilage

Nordlichtertreffen 90

Auch dieses Jahr wieder eintreffen im Norden. Allerdings hatte ich mit dem Treffen so meine liebe Not. Bis zum Anmeldungsschluß lagen sage und schreibe drei Anmeldung vorliegen. Aufgrunddessen mußte ich das Treffen zu mir in den Keller verlegen, da ich nicht mit so wenig Leuten das halbe Lokal belegen konnte.

Am alles entscheidenden Tag ergab die Zählung der Häupter dann sechs Personen, mich eingerechnet:

Jörg Brans, Gerd Rinio, Gerald Schröder, Hans-Dieter Hartmann, Andreas Magnus und meine Wenigkeit

Andreas hatte sich kurzfristig per Telefon angekündigt und Gerd noch am Samstag eingefunden.

An digitalem Handwerkszeug waren vertreten zwei Model 4P (Jörg & Hans-Dieter) sowie ein Model II im Towergehäuse (Andreas). Meine Bitschleuder mußte mangels Monitor im Stall bleiben.

Tagesordnung gab es keine, es wurde nachdem alles "hergerichtet" (oder auch "hingerichtet"?) war, munter an den mitgebrachten Arbeitspaketen gewerkelt. Gerald versuchte aus der Hüfte geschossen eine Software zu erstellen, welche in der Lage war, die Filegröße und einige weitere interessante Dinge die CP/M nicht so ohne weiteres herausrückt, zu lesen. Jörg zeigte mir die neueste Version der Textverarbeitung LeScript. Leider hatte er mit seiner eingedeutschten Version bei den Umlauten formatierprobleme, die, wie er sagte, dem Hersteller Anitek wohl noch einige Schweißperlen abringen werden. Andreas kämpfte mit Refresh-Problemen. Seine Z280 CPU verstand sich nicht so ganz mit dem Rest der Welt (=Hardware). Leider gelang es auch mit vereinten Kräften nicht, die Ursache einzukreisen.

Gegen Sonntag Mittag, nach dem Essen wurde dann das große Einpacken und Heimreisen gestartet.

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

Jahreshauptversammlung	Weilburg	09.05.- 12.05.91
Clubtreffen Nord-91

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

Mai
Juli
Oktober
Dezember

IAA	Frankfurt	12.09.- 22.09.91
Büro-data	Berlin	16.10.- 19.10.91
SYSTEMS	München	21.10.- 25.10.91

Rest

Liebe Club 80 - Mitglieder, lieber Jens,

dem Trend des allgemeinen Du folgend hoffe ich auf Einverständnis mit dieser Anrede. Ich bin nun seit Herbst 90 Mitglied und möchte mich kurz vorstellen, nachdem Fritz Chwolka uns Neulinge dazu ja ausdrücklich aufgefordert hat. (Die Sitten eines Clubs muß man ja erst lernen).

Ich bin 45 Jahre alt, insofern also auch ein "Fossil", zwar nicht aus der Steinzeit, wie Fritz meinen Computer mal bezeichnete, aber doch nur eben noch ein Nachkriegsjahrgang und somit nicht mehr ganz jung. Bin verheiratet, habe zwei Töchter im juristischen Erwachsenenalter (persönlich wird man's ja erst etwas später), wohne im eigenen Haus (teure Sache) mit großem Garten, der viel Arbeit mit sich bringt und bin ganz froh, daß unsere kleine Ortsgemeinde mit nur ein paar Hundert Einwohnern gerade noch ganz abgelegen liegt.

Beruflich bin ich Ingenieur bei der Deutschen Bundespost Telecom (Neudeutsch!) und bekam hier diesen "meinen" Computer, einen Triumph Adler AlphaTronic P3. Betriebssystem CP/M 2.2. Dies führte mich damals über Uwe Herczeg zu Fritz Chwolka und dem Club 80. Meinen ersten Eindruck davon bekam ich beim Treffen in Mannheim, wo ich mich dann abends beim Grillessen heimisch zu fühlen begann. Dank sei hier nochmals vor allem Fritz Chwolka gesagt, der sich auch im Nachhinein voll für mich engagierte.

Und damit bin ich beim Eigentlichen. Mein Rechner hat eine 8085-CPU. Die ist leider nur 8080-kompatibel und viele der PD-Programme haben Z80-Befehle. Also, wer interessante 8080-Programme hat, wird bei mir immer auf Interesse stoßen. Und noch eine Bitte habe ich. Fritz Chwolka hat's schon angedeutet. Wegen eines fehlenden MOVECPM-Programmes für meinen Rechner können wir die Installation von ZCPR nicht (vorerst ?) durchführen. Wer kennt jemanden, der dies Programm für Triumph Adler AlphaTronic P3 hat ?

Ansonsten bemühe ich mich einerseits, Assemblersprache zu verstehen und andererseits die Steuerung der Diskettenlaufwerke zu begreifen. Ohne eine BIOS-Beschreibung ist das natürlich schwer.

Mit freundlichem Gruß

Uwe Schoberth
Petrus-Waldus-Str. 14
Schönenberg
W-7136 Ötisheim

Vorstellungen neuer
Clubmitglieder

Lieber Jens,

da ich mich nun doch entschlossen habe Mitglied im Club zu werden, möchte ich mich kurz vorstellen.

Auch wenn ich mich schon mehr als sieben Jahre mit Mikroelektronik befasse, bin ich doch mit meinen fast vierzig Jahren auf der Computerstrecke ein blutiger Anfänger. Ein Computer war für viele Interessierte in der Ex-DDR ein schwer erfüllbarer Wunsch. Für meinen Ersten, einen Philips VG 8010, mußte ich mir das Rauchen abgewöhnen. (Man braucht ja für die Frau Argumente !). Und dann habe ich mir den Kasten für unsere "stabile Ostmark" für den Preis gekauft, für den ich heute einen IBM-Compatible mit wenigstens 40-er Festplatte haben könnte. Im vorigem Jahr habe ich mir dann aus der Konkursmasse von Robotron den Alba-PC gekauft. Das ist eine modifizierte Variante des Bildungscomputers, kurz BIC genannt. Dieses Gerät sollte in allen Schulen und Bildungseinrichtungen der DDR eingesetzt werden. Es arbeitet mit der CPU U880 (Z80), ist mit 3,75 MHz getaktet und hat folgende Speicher: 40 kbyte ROM (Treiber, BASIC-Interpreter)

64 kbyte RAM

64 k x 16 bit VRAM

Grafik: 320 x 200 (max 6 Seiten)

640 x 200 (max 3 Seiten)

Diskeinheit : 2 x 5,25 Zoll, 780 kbyte

Durch Zufall fand ich in einer Zeitschrift die Adresse von Claus (Vorstellung im 32. Clubinfo). Er half mir über die ersten Hürden und interessierte mich für den Club.

Meine nächsten Ziele: System ausbauen (ich arbeite immer noch mit einem stinknormalen Kofferrfernseher), RAM-Floppy, Maus..., und mehr in die Programmierung (TPascal) schnüffeln. Vorausgesetzt mir bleibt die nötige Zeit.

Ich bin Ingenieur für Kraftwerksanlagen und arbeite als technischer Leiter eines Fernwärmeversoorgungsbetriebes. Privat interessiert mich Alles, was mit dem Thema Heizen. Heizanlagen. Eneriesparen usw. zu tun hat und natürlich auch alle Software zu diesem Thema.

Das sollte für eine kurze Vorstellung genügen.

Ich freue mich über jeden Kontakt.

Gert Halgasch
Großschönauer Str.26
O-8805 Jonsdorf

PS: Wer jetzt Jonsdorf sucht- es liegt im Dreiländereck
Polen, Tschechoslowakei, Deutschland- in der Nähe
der Stadt Zittau.

Expertensystem in Basic ??!!

KI Programmierung mit TRS 80/GENIE Computern

1. Einführung

Durch Zufall fiel mir vor ca. einem Jahr ein Buch mit folgendem Titel auf: KI Expertensysteme programmieren

Einführung in die Funktionsweise mit Beispielen in BASIC

Das Thema begann mich zu faszinieren und es reifte der Gedanke die im Buch vorgestellten Routinen in BASIC auf unserere Small System Computer zu übertragen.

Schwierigkeiten gab es genügend, aber mittlerweile liegt eine lauffähige Version vor, die ich hier vorstellen möchte. Das Programmsystem setzt sich, um Speicherplatz zu sparen, aus mehreren Einzelprogrammen zusammen und gibt auch für andere Problemlösungen in BASIC gute Programmierbeispiele.

Es stellte sich beim Programmieren und Durcharbeiten des Buches heraus, daß Expertensysteme nichts mystisches und keine KI im eigentlichen Sinn, sondern nur eine besondere Programmier-technik sind.

Aus diesem Grund habe ich mich trotz der Länge der Sources entschlossen diese abzdrukken. Falls jemand Lust bekommt den ganzen Krempel bis ins Detail nachzuvollziehen: die Zeilennummerierung entspricht den Programmierbeispielen im Buch
Was leistet nun der ganze Krempel ?

Als Testkomponente enthält die Disk eine Wissensbasis um ein Radio lediglich zu reparieren. Vorstellbar wäre das sich einige Hardware "Experten" eine Wissensbasis z.B. zur Reparatur des Trash 80 erarbeiten.

Allerdings: ein fertiges Expertensystem (entgegen manchen Versprechungen bei "großen Systemen") eignet sich immer nur für wenige gleichartige Probleme.

Das vorgestellte Programmsystem liefert aber ein funktionsfähiges Skelett, das sich zu einem voll einsatzfähigen Expertensystem ausbauen läßt.

Das Beispiel zur Arbeitsweise ist dem Eingangs vorgestellten Buch entnommen.

1.1 Arbeitsweise

Ein Expertensystem ahmt die Arbeitsweise einer Person nach, die aufgrund ihres Fachwissens Probleme in einem begrenzten Fachgebiet lösen kann.

Beispiel:

Mein Transistorradio gibt seinen Geist auf. Ich habe einen Bekannten mit dem üblichen Beruf des Radiotechnikers. Folgender Dialog könnte sich entwickeln:

Ich : Mein Radio ist defekt
Experte : Spielt das Radio, kann man Sprache oder Musik hören ?
Ich : Nein.
Experte : Hört man bei voller Lautstärke mindestens ein leises Rauschen ?
Ich : Ja.
Experte : Endstufe oder Lautsprecher sind wahrscheinlich in Ordnung. Schlägt die Empfangsanzeige aus ?
Ich : Ja, etwas.
Experte : Empfangsstufe ist auch oK. Der Fehler liegt zwischen

Eingangsstufe und Endstufe, wahrscheinlich ist der NF-Vorverstärker defekt.

Ein Experte stellt also Fragen um Fakten zu sammeln. Anschließend erfolgt eine Auswertung/Bewertung der gesammelten Fakten und eine Begründung.

Das Expertensystem soll diese Vorgehensweise nachahmen. Deshalb besteht es aus einer sogenannten Wissensbasis und einem Diagnosesystem.

2. Die Wissensbasis

Die Wissensbasis enthält das "Wissen" des Experten. Mit diesem Wissen arbeitet das Diagnosesystem (Inferenzkomponente), welches Eingabedaten vom Benutzer erwartet in der Form von, um bei dem Beispiel des Radios zu bleiben:

- nicht - EMPFANGSANZEIGE

Trifft diese Aussage zu Ja/Nein ?

Die Inferenzkomponente arbeitet also mit der Wissensbasis, die sinnvoll strukturiert sein muß. Z.B. kann man auf das Wissen eines Lexikons durch die alphabetische Sortierung nach Schlagworten leicht zugreifen.

Das Wissen kann in zwei Untereinheiten untergliedert werden, in Faktenwissen (ein Computer hat einen Prozessor) und Regelwissen (wenn der Computer einen Prozessor hat und keinen Strom, dann läuft er nicht).

Fakten sind im einfachsten Fall wahr oder falsch:

Aussage	Darstellung	Zustand
Der Computer hat einen bernsteinfarbenen Monitor	BERNSTEIN-MONITOR	wahr
Der Computer hat eine Floppy Disk	TAPE-RECORDER	falsch

Das Regelwissen hat eine größere Bedeutung für unser Expertensystem, da Regeln zu weiterem Wissen führen, wenn bestimmte Fakten vorhanden sind.

Beispiel:

Aus DIE-SONNE-SCHEINT ist wahr folgt ES-IST-TAG ist wahr.
Genug der grauen Theorie, es folgt die Praxis.

2.1 Aufbau der Wissensbasis

In der Wissensbasis sind FRAGEN mit MF% (max. Fragen) Fragen enthalten und werden wie folgt in Felder unterteilt:

DIM FTEXTS(FRAGEN%), (FAS(2, FRAGEN%), FSTATUS(FRAGEN%),
OBJEKT%(FRAGEN%))

FTEXTS(N%)	Text der N-ten Frage
FAS(1, N%)	Antwort, wenn der Zustand des zur Frage gehörigen Objektes wahr sein soll
FAS(2, N%)	s.o., aber Zustand falsch
FSTATUS(N%)	Zustand der Frage (0=unbestimmt, 1=erste Antwort wurde ausgewählt, -1=zweite Antwort wurde ausgewählt)
OBJEKTs(N%)	Name des Objektes, auf das sich die Frage bezieht

Für die Regeln wird nach folgendem Schema verfahren:

Die Regel

Wenn gilt	1.Prämisse-1	und
	2.Prämisse-2	und
	usw.	
dann	AKTION	

Beispiele:
Fragennummer : 1

Fragetext : Klingt Musik dumpf ?
Antworttext 1 : Ja
Antworttext 2 : Nein
Bezugsobjekt-Name: KLANG-DUMPF

Regelnummer : 1
Regel/Diagnose : Diagnose
Prämisse 1 : EMPFANGSTEIL-IN-ORDNUNG
Prämisse 2 : ENDSTUFE-KEIN-EINGANGSSIGNAL
Prämisse 3 : -
Prämisse 4 : -
Prämisse 5 : -
Aktion/Diagnose : NF-VORVERSTARKER-DEFEKT

Objektnummer : 1
Objekttyp : Regel
Objektname : KLANG-DUMPF
Für weiter Details verweise ich auf den Literaturhinweis

2.2 Der Editor

Das Programm WBEDIT ist der Editor um Fragen, Regeln, Objekte, Diagnosen und Fakten (Constraints) in die Wissensbasis einzutragen und zu speichern.

```
1 'Wissenserwerbskomponente
2 'Dr. H.-J. Soll 1987
3 'modifiziert für Tandy TRS80 Mod.1
4 'von Egbert Schröder 1990
5 CLEAR 7000
10 MO%=200
20 MF%=50
30 MR%=100
40 MC%=50
50 MQ%=20
60 DIM OAs(MO%),OSTATUS%(MO%),ODIAG%(MO%)
70 DIM FTEXTs(MF%),FAs(2,MF%),FSTATUS%(MF%),FObjEKts(MF%)
80 DIM RSTATUS%(MR%),WENNs(5,MR%),WF%(5,MR%),DANNs(MR%),TYP%(MR%)
90 DIM CNs(2,MC%),CS%(MC%)
100 DIM QUs(MQ%)
110 DIM Vs(5)
120 CLS
130 PRINT $5,"Was möchten Sie ....."
140 PRINT $133,"1. Bestehende Wissensbasis bearbeiten"
150 PRINT $197,"2. Neue Wissensbasis erzeugen"
160 PRINT $329,"Bitte wählen Sie : ";
170 Is=INKEY$:IF Is="" GOTO 170
180 PRINT Is:I%=VAL(Is)
190 PRINT $581,"Name der Wissensbasis : ",:LINEINPUT WNs
200 ON I% GOTO 230,510
220 PRINT $348," ":PRINT $606," ":GOTO 160
230 OPEN "I",1,WNs
240 INPUT #1,OBJEKTE%:INPUT #1,FRAGEN%:INPUT #1,REGELEN%
250 INPUT #1,CO%:INPUT #1,NQ%
260 IF OBJEKTE%=0 GOTO 300
270 FOR N%=1 TO OBJEKTE%
280 INPUT #1,OAs(N%):INPUT #1,OSTATUS%(N%):INPUT #1,ODIAG%(N%)
290 NEXT N%
300 IF FRAGEN%=0 GOTO 340
310 FOR N%=1 TO FRAGEN%
320 INPUT #1,FTEXTs(N%):INPUT #1,FAs(1,N%):
INPUT #1,FAs(2,N%):INPUT #1,FSTATUS%(N%):
INPUT #1,FObjEKts(N%)
```

```
330 NEXT N%
340 IF REGELEN%=0 GOTO 420
350 FOR N%=1 TO REGELEN%
360 INPUT #1,RSTATUS%(N%)
370 FOR M%=1 TO 5
380 INPUT #1,WENNs(M%,N%):INPUT #1,WF%(M%,N%)
390 NEXT M%
400 INPUT #1,DANNs(N%):INPUT #1,TYP%(N%)
410 NEXT N%
420 IF CO%=0 GOTO 460
430 FOR N%=1 TO CO%
440 INPUT #1,CNs(1,N%):INPUT #1,CNs(2,N%):INPUT #1,CS%(N%)
450 NEXT N%
460 IF NQ%=0 GOTO 480
470 FOR N%=1 TO NQ%:INPUT #1,QUs(N%):NEXT N%
480 CLOSE #1
490 GOTO 1010
500 'Neue Wissensbasis erzeugen
510 OBJEKTE%=0:FRAGEN%=0:REGELEN%=0:CO%=0:QU%=0
520 GOTO 1010
1000 'Wissensbasis bearbeiten
1010 CLS
1020 PRINT $5,"Was möchten Sie ....."
1030 PRINT $133,"1. Fragen bearbeiten"
1040 PRINT $197,"2. (Diagnose-)Regeln bearbeiten"
1050 PRINT $261,"3. Objekte bearbeiten"
1060 PRINT $325,"4. Sichern und Beenden"
1070 PRINT $584,"Bitte wählen Sie : ";
1080 Is=INKEY$:IF Is="" GOTO 1080
1090 PRINT Is:I%=VAL(Is)
1100 ON I% GOTO 2010,5010,8010,10010
1140 PRINT $603," ":GOTO 1070
2000 'Fragen bearbeiten
2010 CLS
2020 PRINT $5,"Was möchten Sie ....."
2030 PRINT $133,"1. Frage hinzufügen"
2040 PRINT $197,"2. Fragen auflisten"
2050 PRINT $261,"3. Fragen löschen"
2060 PRINT $325,"4. Eingangsfragen festlegen"
2070 PRINT $584,"Bitte wählen Sie : ";
2080 Is=INKEY$:IF Is="" GOTO 2080
2090 PRINT Is:I%=VAL(Is)
2100 ON I% GOTO 2210,2610,2810,12010
2140 PRINT $603," ":GOTO 2070
2200 'Frage hinzufügen
2210 GOSUB 4010
2220 IF FRAGEN%=MFRAGEN% GOTO 11010
2230 FRAGEN%=FRAGEN%+1
2240 PRINT $83,"":PRINT CHR$(15);:PRINT FRAGEN%;
2250 ZEILE%=2:K%=20
2255 ZEILE%=ZEILE%+1:IF ZEILE%=9 THEN ZEILE%=10:K%=22
2257 IF ZEILE%>10 THEN 2290
2260 PRINT $ZEILE%*64+K%,"":PRINT CHR$(14);
2265 As=INKEY$:IF As="" THEN 2265
2270 IF As=CHR$(13) THEN PRINT CHR$(15);:GOTO 2255
2275 IF POS(0)=61 THEN PRINT As;:PRINT CHR$(15);:GOTO 2255:ELSE
PRINT As;:GOTO 2265
2290 PRINT $910,"Eintrag in Ordnung ? (j/n)";
2300 Is=INKEY$:IF Is="" GOTO 2300
2310 IF (Is<>"j" AND Is<>"n") THEN FRAGEN%=FRAGEN%-1:GOTO 2210
2320 GOSUB 4110:GOTO 1010
2600 'Fragen auflisten
```



```

2610 CLS
2620 IF FRAGEN% <> 0 GOTO 2670
2630 PRINT S584, "xxx keine Fragen in der Wissensbasis xxx"
2640 PRINT S652, "xxx Bitte eine Taste drücken xxx"
2650 IS=INKEYS: IF IS="" GOTO 2650
2660 GOTO 1010
2670 PRINT S83, "": GOSUB 4010
2680 FOR N%=1 TO FRAGEN%
2690 PRINT S83, N%;
2700 PRINT S212, LEFT$(FTEXT$(N%), 42);:
PRINT S212+64, RIGHT$(FTEXT$(N%), 42);
2710 PRINT S340, LEFT$(FAS(1, N%), 42);:
PRINT S340+64, RIGHT$(FAS(1, N%), 42);
2720 PRINT S468, LEFT$(FAS(2, N%), 42);:
PRINT S468+64, RIGHT$(FAS(2, N%), 42);
2730 PRINT S662, FOBJEKTS(N%);: PRINT CHR$(15);
2740 PRINT S913, "Zum Abbruch <E> drücken"
2750 PRINT S968, "Zur Fortsetzung eine andere Taste drücken";
2760 IS=INKEYS: IF IS="" GOTO 2760
2770 IF (IS="E" OR IS="e") GOTO 1010
2780 NEXT N%
2790 GOTO 1010
2800 "Frage löschen
2810 CLS
2820 PRINT S261, "Geben Sie bitte die Nr. der zu"
2830 PRINT S325, "löschenden Frage ein : "; INPUT N%
2840 IF N% <= FRAGEN% GOTO 2890
2850 PRINT S584, "xxxxx Fragennummer zu groß xxxxx"
2860 PRINT S651, "Bitte eine Taste drücken"
2870 IS=INKEYS: IF IS="" GOTO 2870
2880 GOTO 1010
2890 IF NQ%=0 GOTO 2930
2900 FOR M%=1 TO NQ%
2910 IF FOBJEKTS(M%)=QUS(M%) THEN CLS:
PRINT S133, "xxx Frage ist Eingangsfrage xxx": GOTO 2860
2920 NEXT M%
2930 PRINT S83, "": GOSUB 4010
2940 PRINT S83, N%;
2950 PRINT S212, LEFT$(FTEXT$(N%), 42);:
PRINT S212+64, RIGHT$(FTEXT$(N%), 42);
2960 PRINT S340, LEFT$(FAS(1, N%), 42);:
PRINT S340+64, RIGHT$(FAS(1, N%), 42);
2970 PRINT S468, LEFT$(FAS(2, N%), 42);:
PRINT S468+64, RIGHT$(FAS(2, N%), 42);
2980 PRINT S662, FOBJEKTS(N%);: PRINT CHR$(15);
2990 PRINT S910, "Soll diese Frage gelöscht werden (j/n) ";
3000 IS=INKEYS: IF IS="" GOTO 3000
3010 IF (IS<"j" AND IS<"J") THEN FRAGEN%=FRAGEN%-1: GOTO 1010
3020 IF N%=FRAGEN% GOTO 3090
3030 FOR M%=N%+1 TO FRAGEN%
3040 FTEXT$(M%-1)=FTEXT$(M%)
3050 FAS(1, M%-1)=FAS(1, M%)
3060 FAS(2, M%-1)=FAS(2, M%)
3070 FOBJEKTS(M%-1)=FOBJEKTS(M%)
3080 NEXT M%
3090 FRAGEN%=FRAGEN%-1
3100 GOTO 1010
4000 "Fragenmaske
4010 CMD"load maskel/dum"
4070 RETURN
4100 "Einlesen aus Fragenmaske
4110 PRINT CHR$(15);: I2%=1: I3%=2

```

```

4120 FOR I1%=0 TO 2
4130 FOR I2%=I2% TO I3%: FOR K%=0 TO 41
4140 AS(I1%)=AS(I1%)+CHR$(PEEK((2+I1%)*64+15380+K%)):
POKE(2+I1%)*64+15380+K%, 46: NEXT K%, I2%=I2%+2: I3%=I3%+2:
NEXT I1%
4145 FOR K%=0 TO 39:
AS(3)=AS(3)+CHR$(PEEK(10*64+15382+K%)):
POKE(10*64)+15382+K%, 46: NEXT K%
4150 FTEXT$(FRAGEN%)=AS(0)
4160 FAS(1, FRAGEN%)=AS(1)
4170 FAS(2, FRAGEN%)=AS(2)
4180 FOBJEKTS(FRAGEN%)=AS(3)
4185 FOR I1%=0 TO 3: AS(I1%)="": NEXT I1%
4190 RETURN
5000 "Regeln bearbeiten
5010 CLS
5020 PRINT S5, "Was möchten Sie ....."
5030 PRINT S133, "1. (Diagnose-)Regeln hinzufügen"
5040 PRINT S197, "2. (Diagnose-)Regeln auflisten"
5050 PRINT S261, "3. (Diagnose-)Regeln löschen"
5060 PRINT S584, "Bitte wählen Sie : ";
5070 IS=INKEYS: IF IS="" GOTO 5070
5080 PRINT IS: I%=VAL(I%)
5090 ON I% GOTO 5210, 5810, 6110
5120 PRINT S603, " ": GOTO 5060
5200 "Regeln hinzufügen
5210 GOSUB 7010
5220 IF REGELN%=MR% GOTO 11010
5230 REGELN%=REGELN%+1
5240 PRINT S18, REGELN%;
5250 PRINT S147, "": INPUT DIAGNOSES
5260 DIAGNOSE%=0:
IF (LEFT$(DIAGNOSES, 1)="d" OR LEFT$(DIAGNOSES, 1)="D")
THEN DIAGNOSE%=1
5270 TYP%(REGELN%)=DIAGNOSE%
5280 PRINT S275, "": INPUT WS(1): WS=WS(1): GOSUB 5690
5290 PRINT S339, "": INPUT WS(2)
5300 PRINT S403, "": INPUT WS(3)
5310 PRINT S467, "": INPUT WS(4)
5320 PRINT S531, "": INPUT WS(5)
5330 PRINT S659, "": INPUT DS
5340 PRINT S910, "Eintrag in Ordnung ? (j/n)";
5350 IS=INKEYS: IF IS="" GOTO 5350
5360 IF (IS<"j" AND IS<"J") THEN REGELN%=REGELN%-1: GOTO 5210
5370 FOR N%=1 TO 5
5380 OBJEKTS=VS(N%)
5390 IF VS(N%)<>"- oder -" GOSUB 5610: OTYP%=0: GOSUB 5510
5400 WENN$(N%, REGELN%)=OBJEKTS: VF%(N%, REGELN%)=VO%
5410 NEXT N%
5420 OBJEKTS=DS
5430 OTYP%=DIAGNOSE%: GOSUB 5510
5440 DANN$(REGELN%)=OBJEKTS
5450 GOTO 1010
5500 "Objekte für Regeln erzeugen
5510 IF OBJEKTS="" THEN RETURN
5520 IF OBJEKTE%=0 GOTO 5570
5530 FOR NUMMER%=1 TO OBJEKTE%
5540 IF OBJEKTS=OAS(NUMMER%) THEN RETURN
5550 NEXT NUMMER%
5560 IF OBJEKTE%=MOBJEKTE% GOTO 11010
5570 OBJEKTE%=OBJEKTE%+1
5580 OAS(OBJEKTE%)=OBJEKTS: ODIAG$(OBJEKTE%)=OTYP%

```



```
5590 RETURN
5600 "Nicht" bei Regel abtrennen
5610 VO%=1
5620 IF LEN(OBJEKTS)<7 THEN RETURN
5630 US=LEFTS(OBJEKTS,6):GOSUB 20020:IF US<>"NICHT" THEN RETURN
5640 OBJEKTS=RIGHTS(OBJEKTS,LEN(OBJEKTS)-6)
5650 IF LEFTS(OBJEKTS,1)=" " THEN
  OBJEKTS=RIGHTS(OBJEKTS,LEN(OBJEKTS)-1):GOTO 5650
5660 VO%=-1
5670 RETURN
5680 "oder-Regel
5690 IF (LEFTS(WS,1)=" " OR LEFTS(WS,1)="-")
  THEN WS=RIGHTS(WS,LEN(WS)-1):GOTO 5690
5700 IF (RIGHTS(WS,1)=" " OR RIGHTS(WS,1)="-")
  THEN WS=LEFTS(WS,LEN(WS)-1):GOTO 5700
5710 US=WS:GOSUB 20020:IF US<>"ODER" THEN RETURN
5720 WS(1)="- oder -"
5730 RETURN
5800 "Regeln auflisten
5810 CLS
5820 IF REGELN%>0 GOTO 5870
5830 PRINT S584,"*** keine Regeln in der Wissensbasis ***"
5840 PRINT S657,"Bitte eine Taste drücken"
5850 IS=INKEYS:IF IS="" GOTO 5850
5860 GOTO 1010
5870 FOR M%=1 TO REGELN%
5880 GOSUB 7010
5890 PRINT S18,M%;
5900 DIAGNOSES="Regel          ":
  IF TYP%(M%)=1 THEN DIAGNOSES="Diagnose"
5910 PRINT S147,DIAGNOSES;
5920 FOR M%=1 TO 5
5930 WS=WENN$(M%,M%)
5940 IF WF%(M%,M%)=(-1) THEN WS="nicht "+WS
5950 IF WS="" THEN WS="-"
5960 PRINT S64*M%+211,WS;
5970 NEXT M%
5980 PRINT S659,DANN$(M%);
5990 PRINT S848,"Zum Abbruch <E> drücken"
6000 PRINT S904,"Zur Fortsetzung eine andere Taste drücken"
6010 IS=INKEYS:IF IS="" GOTO 6010
6020 IF (IS="E" OR IS="e") GOTO 1010
6030 NEXT M%
6040 GOTO 1010
6100 "Regeln löschen
6110 CLS
6120 PRINT S197,"Geben Sie bitte die Nr. der zu"
6130 PRINT S261,"löschenden Regel ein : ";:INPUT N%
6140 IF N%<=REGELN% GOTO 6190
6150 PRINT S584,"***** Regelnummer zu groß *****"
6160 PRINT S651,"Bitte eine Taste drücken"
6170 IS=INKEYS:IF IS="" THEN 6170
6180 GOTO 6110
6190 GOSUB 7010
6200 PRINT S18,N%;
6210 DIAGNOSES="Regel          ":
  IF TYP%(N%)=1 THEN DIAGNOSES="Diagnose"
6220 PRINT S147,DIAGNOSES;
6230 FOR M%=1 TO 5
6240 WS=WENN$(M%,M%)
6250 IF WF%(M%,M%)=(-1) THEN WS="nicht "+WS
6260 IF WS="" THEN WS="-"
```

```
6270 PRINT S64*M%+211,WS;
6280 NEXT M%
6290 PRINT S659,DANN$(M%);
6300 PRINT S904,"Soll diese (Diagnose-)Regel gelöscht werden ?
      (j/n)";
6310 IS=INKEYS:IF IS="" GOTO 6310
6320 IF (IS<>"j" AND IS<>"J") GOTO 1010
6330 PRINT S787,"--- Bitte warten ---"
6340 FOR K%=1 TO 5
6350 OBJEKTS=WENN$(K%,M%):GOSUB 7110
6360 NEXT K%
6370 OBJEKTS=DANN$(M%):GOSUB 7110
6380 IF M%=REGELN% GOTO 6460
6390 FOR M%=M%+1 TO REGELN%
6400 FOR K%=1 TO 5
6410 WENN$(K%,M%-1)=WENN$(K%,M%)
6420 WF%(K%,M%-1)=WF%(K%,M%)
6430 NEXT K%
6440 DANN$(M%-1)=DANN$(M%):TYP%(M%-1)=TYP%(M%)
6450 NEXT M%
6460 REGELN%=REGELN%-1
6470 PRINT S777,"          "
6480 GOTO 1010
7000 "Regelmaske
7010 CMD"load maske2/dum"
7100 RETURN
7110 "ggf. Objekt löschen
7120 ANZAHL%=0
7130 FOR M%=1 TO OBJEKTE%
7140 IF OBJEKTS=WENN$(1,M%) THEN ANZAHL%=ANZAHL%+1
7150 IF OBJEKTS=WENN$(2,M%) THEN ANZAHL%=ANZAHL%+1
7160 IF OBJEKTS=WENN$(3,M%) THEN ANZAHL%=ANZAHL%+1
7170 IF OBJEKTS=WENN$(4,M%) THEN ANZAHL%=ANZAHL%+1
7180 IF OBJEKTS=WENN$(5,M%) THEN ANZAHL%=ANZAHL%+1
7190 IF OBJEKTS=DANN$(M%) THEN ANZAHL%=ANZAHL%+1
7200 NEXT M%
7210 IF ANZAHL%>1 THEN RETURN
7220 FOR M%=1 TO OBJEKTE%
7230 IF OAS(M%)=OBJEKTS GOTO 7260
7240 NEXT M%
7250 RETURN
7260 IF M%=OBJEKTE% GOTO 7320
7270 FOR L%=M%+1 TO OBJEKTE%
7280 OAS(L%-1)=OAS(L%)
7290 OSTATUS%(L%-1)=OSTATUS%(L%)
7300 ODIAG%(L%-1)=ODIAG%(L%)
7310 NEXT L%
7320 OBJEKTE%=OBJEKTE%-1
7330 RETURN
8000 "Objekte anzeigen
8010 CLS
8020 PRINT S5,"Was möchten Sie ....."
8030 PRINT S133,"1. Objekte auflisten"
8040 PRINT S197,"2. Constraints auflisten"
8050 PRINT S261,"3. Constraints eintragen"
8060 PRINT S325,"4. Constraints löschen"
8070 PRINT S584,"Bitte wählen Sie : ";
8080 IS=INKEYS:IF IS="" GOTO 8080
8090 PRINT IS:I%=VAL(IS)
8100 ON I% GOTO 8210,8610,9010,9410
8140 PRINT S603," ":GOTO 8070
8200 "Objekte auflisten
```



```

8210 CLS
8220 IF OBJEKTE%(>0) GOTO 8270
8230 PRINT S584,"*** keine Objekte in der Wissensbasis ***"
8240 PRINT S657,"Bitte eine Taste drücken"
8250 IS=INKEYS:IF IS="" GOTO 8250
8260 GOTO 1010
8270 ZEILE%=2
8280 FOR NZ=1 TO OBJEKTE%
8290 IF ZEILE%=2 THEN PRINT S0,"Objektnr. Typ Objektname"
8300 PRINT S64*ZEILE%+4,NZ
8310 IS="P":IF ODIAG%(NZ)=1 THEN IS="D"
8320 PRINT S64*ZEILE%+12,IS
8330 PRINT S64*ZEILE%+15,OAS(NZ)
8340 ZEILE%=ZEILE%+2
8350 IF ZEILE%<14 GOTO 8400
8360 PRINT S904,"Bitte eine Taste drücken <E>=Ende"
8370 IS=INKEYS:IF IS="" GOTO 8370
8380 IF (IS="E" OR IS="e") GOTO 1010
8390 CLS:ZEILE%=2
8400 NEXT NZ
8410 PRINT S840,"***** Ende der Liste *****"
8420 PRINT S908,"Bitte eine Taste drücken"
8430 IS=INKEYS:IF IS="" GOTO 8430
8440 GOTO 1010
8600 Constraints auflisten
8610 CLS
8620 IF CO%(>0) GOTO 8670
8630 PRINT S584,"*** keine Constraints in der Wissensbasis ***"
8640 PRINT S657,"Bitte eine Taste drücken"
8650 IS=INKEYS:IF IS="" GOTO 8650
8660 GOTO 1010
8670 ZEILE%=2
8680 FOR NZ=1 TO CO%
8690 IF ZEILE%=2 THEN
PRINT S0,"Nr. Faktum 1 bedingt Faktum 2"
8700 PRINT S64*ZEILE%,NZ
8710 IS=CNS(1,NZ)+" ==> "
8720 IF CS%(NZ)=(-1) THEN IS=IS+"nicht "
8730 IS=IS+CNS(2,NZ)
8740 PRINT S64*ZEILE%+6,IS
8750 ZEILE%=ZEILE%+2
8760 IF ZEILE%<14 GOTO 8810
8770 PRINT S904,"Bitte eine Taste drücken <E>=Ende"
8780 IS=INKEYS:IF IS="" GOTO 8780
8790 IF (IS="E" OR IS="e") GOTO 1010
8800 CLS:ZEILE%=2
8810 NEXT NZ
8820 PRINT S840,"***** Ende der Liste *****"
8830 PRINT S908,"Bitte eine Taste drücken"
8840 IS=INKEYS:IF IS="" GOTO 8840
8850 GOTO 1010
9000 Constraints eintragen
9010 CLS
9020 PRINT S133,"Faktum 1 : "
9030 PRINT S261,"bewirkt ==>"
9040 PRINT S389,"Faktum 2 : "
9050 CO%=CO%+1:IF CO%>MC% GOTO 11000
9060 PRINT S133+12,"";:INPUT CNS(1,CO%)
9070 PRINT S389+12,"";:INPUT IS
9080 CS%(CO%)=1
9090 IF LEN(IS)<7 GOTO 9140
9100 US=LEFT$(IS,6):GOSUB 20020:IF US<>"NICHT " GOTO 9140

```

```

9110 IS=RIGHT$(IS,LEN(IS)-6)
9120 IF LEFT$(IS,1)="" THEN IS=RIGHT$(IS,LEN(IS)-1):GOTO 9120
9130 CS%(CO%)=-1
9140 CNS(2,CO%)=IS
9150 PRINT S648,"Eintrag in Ordnung ? (j/n)";
9160 IS=INKEYS:IF IS="" GOTO 9160
9170 IF (IS<>"j" AND IS<>"J") THEN CO%=CO%-1:GOTO 9010
9180 GOTO 1010
9400 Constraints löschen
9410 CLS
9420 PRINT S261,"";:
INPUT"Nr. des zu löschenden Constraints : ";NZ
9430 IF NZ<=MC% GOTO 9480
9440 PRINT S584,"***** Constraint-Nr. zu hoch *****"
9450 PRINT S643,"Bitte eine Taste drücken"
9460 IS=INKEYS:IF IS="" GOTO 9460
9470 GOTO 9410
9480 IS=CNS(1,NZ)+" ==> "
9490 IF CS%(NZ)=(-1) THEN IS=IS+"nicht "
9500 IS=IS+CNS(2,NZ)
9510 PRINT S325,IS
9520 PRINT S584,"Soll dies gelöscht werden ? (j/n)"
9530 IS=INKEYS:IF IS="" GOTO 9530
9540 PRINT S654,"--- Bitte warten ---"
9550 IF (IS<>"j" AND IS<>"J") GOTO 1010
9560 FOR NZ=NZ+1 TO CO%
9570 CNS(1,NZ-1)=CNS(1,NZ)
9580 CNS(1,NZ-1)=CNS(2,NZ)
9590 CS%(NZ-1)=CS%(NZ)
9600 NEXT NZ
9610 CO%=CO%-1
9620 PRINT S654," "
9630 GOTO 1010
10000 Sichern/Beenden
10010 OPEN "O",1,VNS
10020 PRINT #1,OBJEKTE%:PRINT #1,FRAGEN%:PRINT #1,REGELN%
10030 PRINT #1,CO%:PRINT #1,NQ%
10040 IF OBJEKTE%=0 GOTO 10080
10050 FOR NZ=1 TO OBJEKTE%
10060 PRINT #1,OAS(NZ):PRINT #1,OSTATUS%(NZ):PRINT #1,ODIAG%(NZ)
10070 NEXT NZ
10080 IF FRAGEN%=0 GOTO 10120
10090 FOR NZ=1 TO FRAGEN%
10100 PRINT #1,FTEXTS(NZ):PRINT #1,FAS(1,NZ):
PRINT #1,FAS(2,NZ):PRINT #1,FSTATUS%(NZ):
PRINT #1,FOBJEKTS(NZ)
10110 NEXT NZ
10120 IF REGELN%=0 GOTO 10200
10130 FOR NZ=1 TO REGELN%
10140 PRINT #1,RSTATUS%(NZ)
10150 FOR MZ=1 TO 5
10160 PRINT #1,WENNS(MZ,NZ):PRINT #1,WF%(MZ,NZ)
10170 NEXT MZ
10180 PRINT #1,DANNS(NZ):PRINT #1,TYP%(NZ)
10190 NEXT NZ
10200 IF CO%=0 GOTO 10260
10210 FOR NZ=1 TO CO%
10220 PRINT #1,CNS(1,NZ)
10230 PRINT #1,CNS(2,NZ)
10240 PRINT #1,CS%(NZ)
10250 NEXT NZ
10260 IF NQ%=0 GOTO 10280

```



```

10270 FOR N%=1 TO NQ%:PRINT #1,QUS(N%):NEXT N%
10280 CLOSE #1
10290 CLS:END
11000 'Wissensbasis zu klein
11010 CLS
11020 PRINT S10,"***** Wissensbasis zu klein *****"
11030 PRINT S140,"P r o g r a m m a b r u c h !":PRINT S584,"";
11040 STOP
12000 'Eingangsfragen
12010 CLS
12020 PRINT S5,"Was möchten Sie ....."
12030 PRINT S133,"1. Eingangsfragen hinzufügen"
12040 PRINT S197,"2. Eingangsfragen auflisten"
12050 PRINT S261,"3. Eingangsfragen löschen"
12060 PRINT S584,"Bitte wählen Sie : ";
12070 IS=INKEYS:IF IS="" GOTO 12070
12080 PRINT IS:I%=VAL(IS)
12090 ON I% GOTO 12210,12610,13010
12120 PRINT S603," ":GOTO 12070
12200 'Eingangsfrage hinzufügen
12210 CLS
12220 NQ%=NQ%+1:IF NQ%>MQ% GOTO 11010
12230 PRINT S133,"Fragennummer oder Bezugsobjekt-Name : "
12240 PRINT S261,"";:INPUT" ==> ";IS
12250 MX=VAL(IS)
12260 IF MX>FRAGEN% THEN
  PRINT S325,"*** Fragennummer zu groß ***":GOTO 12320
12270 IF MX<>0 THEN QUS(NQ%)=FOBJEKTS(MX):GOTO 12350
12280 FOR N%=1 TO FRAGEN%
12290 IF IS=FOBJEKTS(MX) THEN QUS(NQ%)=IS:GOTO 12350
12300 NEXT N%
12310 PRINT S453,"*** ";IS;" nicht vorhanden ***"
12320 PRINT S653,"Bitte eine Taste drücken"
12330 IS=INKEYS:IF IS="" GOTO 12330
12340 NQ%=NQ%-1:GOTO 12210
12350 PRINT S261,FTEXTS(MX)
12360 PRINT S325,FOBJEKTS(MX)
12370 PRINT S584,"Eintrag in Ordnung ? (j/n)";
12380 IS=INKEYS:IF IS="" GOTO 12380
12390 IF (IS<>"j" AND IS<>"J") THEN NQ%=NQ%-1:GOTO 12210
12400 GOTO 1010
12600 'Eingangsfragen auflisten
12610 CLS
12620 IF NQ%<>0 GOTO 12670
12630 PRINT S584,"*** keine Eingangsfragen vorhanden ***"
12640 PRINT S655,"Bitte eine Taste drücken"
12650 IS=INKEYS:IF IS="" GOTO 12650
12660 GOTO 1010
12670 ZEILE%=2
12680 FOR N%=1 TO NQ%
12690 IF ZEILE%=2 THEN PRINT S0,"Eingangsfragen : "
12700 PRINT S64*ZEILE%,N%
12710 FOR N%=1 TO FRAGEN%
12720 IF QUS(N%)=FOBJEKTS(MX) THEN IS=FTEXTS(MX):GOTO 12750
12730 NEXT N%
12740 IS="*** Frage nach "+QUS(N%)+ " nicht vorhanden ***"
12750 PRINT S64*ZEILE%+2,IS
12760 ZEILE%=ZEILE%+2
12770 IF ZEILE%<14 GOTO 12820
12780 PRINT S908,"Bitte eine Taste drücken      <E>=Ende"
12790 IS=INKEYS:IF IS="" GOTO 12790
12800 IF (IS="E" OR IS="e") GOTO 1010

```

```

12810 CLS:ZEILE%=2
12820 NEXT N%
12830 PRINT S840,"***** Ende der Liste *****"
12840 PRINT S908,"Bitte eine Taste drücken"
12850 IS=INKEYS:IF IS="" GOTO 12850
12860 GOTO 1010
13000 'Eingangsfragen löschen
13010 CLS
13020 PRINT S261,"Geben Sie bitte die Nr. der zu"
13030 PRINT S325,"";:INPUT"löschenden Eingangsfrage ein : ";N%
13040 IF N%<=NQ% GOTO 13090
13050 PRINT S584,"***** Fragennummer zu groß *****"
13060 PRINT S652,"Bitte eine Taste drücken"
13070 IS=INKEYS:IF IS="" GOTO 13070
13080 GOTO 13010
13090 CLS
13100 FOR N%=1 TO FRAGEN%
13110 IF QUS(N%)=FOBJEKTS(MX) GOTO 13170
13120 NEXT N%
13130 PRINT S261,"*** ";QUESTIONS(N%); " nicht vorhanden ***"
13140 PRINT S652,"Bitte eine Taste drücken"
13150 IS=INKEYS:IF IS="" GOTO 13150
13160 GOTO 1010
13170 PRINT S133,"Frage : ";FTEXTS(MX)
13180 PRINT S261,"Objekt : ";FOBJEKTS(MX)
13190 PRINT S584,"Soll diese Frage gelöscht werden ? (j/n)"
13200 IS=INKEYS:IF IS="" GOTO 13200
13210 IF (IS<>"j" AND IS<>"J") THEN NQ%=NQ%-1:GOTO 13010
13220 N%=NQ%
13230 IF N%=NQ% GOTO 13270
13240 FOR N%=N%+1 TO NQ%
13250 QUS(N%-1)=QUS(N%)
13260 NEXT N%
13270 NQ%=NQ%-1
13280 GOTO 1010
20000 'Upper $
20010 ^-> US ->
20020 UUS=""
20030 IF US="" THEN RETURN
20040 FOR U%=1 TO LEN(US)
20050 UTS=MIDS(US,U%,1)
20060 IF (ASC(UTS)>96 AND ASC(UTS)<123)
  THEN UTS=CHR$(ASC(UTS)-32)
20070 UUS=UUS+UTS
20080 NEXT U%
20090 US=UUS
20100 RETURN

```

Auf eines ist beim Aufbau einer Wissensbasis zu besonders zu achten:
Damit beim späteren Ablauf des Expertensystems eine kausale Kette entstehen kann, muß auch in Regeln die sich aufeinander beziehen derselbe Objektname angegeben werden. Dabei ist besonders auf Schreibfehler zu achten !!

3. Die Inferenz(=Diagnose/Begründungs)komponente
Die Inferenzkomponente hat die Aufgabe, aus den eingegebenen Daten für eine Wissensbasis zu Ergebnissen zu kommen. Dies geschieht durch eine Rückwärtsverkettung, bei dem versucht wird von einem Zielzustand den Anfangszustand zu erreichen. Alle Diagnosen sind Ziele, die durch Inferenz bestätigt oder widerlegt werden. Das System bewegt sich rückwärts zu den Prämissen der (Diagnose-)Regeln

und versucht festzustellen, ob diese wahr sind. Sind in der Wissensbasis keine Fakten vorhanden, die eine Entscheidung über den Zustand (wahr oder falsch) zulassen, so werden weitere Regeln herangezogen, die die Gültigkeit dieser Prämissen bestätigen können.

Beispiel:

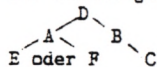
Fakten : A, B, C, D, E, F (D=Diagnose)

Regeln : 1) wenn A und B dann D

2) wenn C dann B

3) wenn E und F dann A

Zur Bestätigung der Diagnose ergibt sich folgende Verkettung:



Um zu einem Ergebnis=Diagnose zu kommen sind dann die Fragen E, F und C zu stellen.

3.1 rekursive Suche/Programmierung

Im Programm wird durch die rückwärtsverkettete Suche für eine Regel, bei der eine Prämisse unbestimmt ist, eine weitere Regel gesucht um den Wahrheitsgehalt der ersten Prämisse zu ermitteln. Vom Prinzip ist dies ein rekursiver Prozeß. Die Methode, mit der dies im Programm realisiert wird, ist die Verwaltung eines STACK, auf dem alle möglichen Zwischenergebnisse (Diagnosen) abgelegt werden, wenn die Prämisse unbestimmt ist. Wenn ein AKTIONsteil einer Regel schließlich ausgeführt werden kann, wird die oberste Regel wieder vom STACK geholt und die vorher unbekannte Prämisse kann weiter bearbeitet werden.

Beispiel:

Regel : wenn R1 dann D1 (D1=Diagnose)

wenn F1 und R3 dann R1

wenn F2 und F3 dann R3

Fragen: F1, F2, F3

```

Schritt 1: Stack: - zu untersuchende Diagnose D1
                  - unbekannte Prämissen R1
2: - D1 auf Stack ablegen
3: D1 zu untersuchendes Faktum R1
   - unbekannte Prämissen F1, R3
4: D1 R1 auf Stack ablegen
5: D1, Frage F1 stellen
   R1 unbekannte Prämisse R3
6: D1, zu untersuchendes Faktum R3
   R1 unbekannte Prämissen F2, F3
7: D1, Frage F2 stellen
   R1 unbekannte Prämissen F3
8: D1, Frage F3 stellen
   R1 unbekannte Prämissen -
9: D1, R3 bestimmen (F2,F3) sind bekannt.
   R1
10: D1, R1 vom Stack holen
    R1
11: D1 R1 bestimmen (F1,R1) sind bekannt
12: - D1 vom Stack holen
13: - D1 bestimmen (R1) ist bekannt
14: - D1 ausgeben und Ende

```

Auf in die Praxis !

3.2 Das Programm

Die Definition des Stack erfolgt durch SRNR%(STACK%) und PZAEHLER%(STACK%)=Prämissenzähler, sowie dem Wert MSTACK% (max. Stackgröße) und der Variablen STACK%, in der die aktuelle

Stacktiefe als Stackpointer enthalten ist. Um das Stellen von überflüssigen Fragen zu vermeiden, werden die Constraints=CO% eingesetzt, die in die Wissensbasis aufgenommen wurden.

Beispiel:

RADIO-SCHWEIGT --> nicht MW-EMPfang-GUT

MW-EMPfang-GUT --> nicht RADIO-SCHWEIGT

Mit dem Verfahren, daß Objekte einen Status haben, lassen sich die Auswirkungen der Constraints nur schlecht nachvollziehen. Deshalb wird zusätzlich noch mit einem Traceprotokoll, eine Art Buchführung über die durchgeführten AKTIONEN, gearbeitet. Auf den Aufbau will ich nicht näher eingehen, denn jeder kann sich bei einem Probelauf die Wirkung mit Ausgabe der Diagnosebegründung ansehen.

```

1 Expertensystem / Inferenzkomponente
2 Dr. H.-J. Soll 1987
3 modifiziert für Tandy TRS80 Mod.1
4 von Egbert Schröder 1990
10 CLS
15 CLEAR 9000
20 MSTACK%=100
25 MTRACER%=200
30 DIM SRNR%(MSTACK%),PZAEHLER%(MSTACK%)
40 PRINT $581,"Name der Wissensbasis : ";:LINEINPUT WNS
50 OPEN "I",1,WNS
60 INPUT #1,OBJEKTE%:INPUT #1,FRAGEN%:INPUT #1,REGELN%
70 INPUT #1,CO%:INPUT #1,ZZ%
80 DIM OAs(OBJEKTE%),OSTATUS%(OBJEKTE%),ODIAG%(OBJEKTE%)
90 DIM FTEXTs(FRAGEN%),FAs(2,FRAGEN%),FSTATUS%(FRAGEN%),
   FOBJEKTS(FRAGEN%)
100 DIM RSTATUS%(REGELN%),WENNs(5,REGELN%),WF%(5,REGELN%),
   DANNs(REGELN%),TYP%(REGELN%)
105 DIM Cns(2,CO%),CSZ(CO%)
107 DIM TRACERs(MTRACER%)
110 IF OBJEKTE%=0 GOTO 150
120 FOR N%=1 TO OBJEKTE%
130 INPUT#1,OAs(N%):INPUT#1,OSTATUS%(N%):INPUT#1,ODIAG%(N%)
140 NEXT N%
150 IF FRAGEN%=0 GOTO 190
160 FOR N%=1 TO FRAGEN%
170 INPUT #1,FTEXTs(N%):INPUT #1,FAs(1,N%):
   INPUT #1,FAs(2,N%):INPUT #1,FSTATUS%(N%):
   INPUT #1,FOBJEKTS(N%)
180 NEXT N%
190 IF REGELN%=0 GOTO 270
200 FOR N%=1 TO REGELN%
210 INPUT #1,RSTATUS%(N%)
220 FOR M%=1 TO 5
230 INPUT #1,WENNs(M%,N%):INPUT #1,WF%(M%,N%)
240 NEXT M%
250 INPUT #1,DANNs(N%):INPUT #1,TYP%(N%)
260 NEXT N%
270 IF CO%=0 GOTO 279
271 FOR N%=1 TO CO%
272 INPUT#1,Cns(1,N%)
273 INPUT#1,Cns(2,N%)
274 INPUT#1,CSZ(N%)
275 NEXT N%
279 CLOSE#1
280 N%=0
290 FOR N%=1 TO REGELN%
300 IF TYP%(N%)=1 THEN N%=N%+1

```



```
310 NEXT M%
320 CLS
330 PRINT $133,FRAGEN%;" Fragen"
340 PRINT $197,REGELN%;" Regeln"
350 PRINT $261,N%;" Diagnosen"
355 PRINT $325,CO%;" Abhängigkeiten"
360 PRINT $657,"Bitte drücken Sie eine Taste"
370 IS=INKEY$:IF IS="" GOTO 370
380 PRINT $657,"Bitte etwas Geduld"
390 PRINT $5,"Die Wissensbasis besteht aus : "
1000 'Status zurücksetzen
1010 FOR N%=1 TO OBJEKTE%
1020 OSTATUS%(N%)=0
1030 NEXT N%
1040 FOR N%=1 TO FRAGEN%
1050 FSTATUS%(N%)=0
1060 NEXT N%
1070 FOR N%=1 TO REGELN%
1080 RSTATUS%(N%)=0
1090 NEXT N%
1092 FOR N%=1 TO MTRACER%
1093 TRACERS(N%)=""
1094 NEXT N%
1095 TRACER%=1
1100 'alle möglichen Diagnosen auf Stack ablegen
1110 STACK%=0
1120 FOR N%=1 TO REGELN%
1130 IF TYP%(N%)=1 GOSUB 3220:TRACERS="1="+DANNS(N%):
    GOSUB 2810:"Stack ablegen"
1140 NEXT N%
2000 'eigentliches Infernzprogramm
2010 IF STACK%=0 GOTO 4010:"Diagnosen ausgeben"
2020 IF RSTATUS%(SRNR%(STACK%))=0 GOTO 2505
2030 STACK%=STACK%-1
2040 GOTO 2010
2500 'Regeln bearbeiten
2505 PRINT $517,"Regeln werden bearbeitet !"
2510 IF (PZAEHLER%(STACK%)>5 AND VENNS(1,SRNR%(STACK%))<>
    "- oder -") THEN RSTATUS%(SRNR%(STACK%))=1:
    NMS=DANNS(SRNR%(STACK%)):TRACERS="2="+NMS+"1="+
    STR$(SRNR%(STACK%)):GOSUB 2810:GOSUB 3430:
    OSTATUS%(N%)=1:GOSUB 5010:GOTO 2030
2520 IF (PZAEHLER%(STACK%)>5 AND VENNS(1,SRNR%(STACK%))=
    "- oder -") THEN RSTATUS%(SRNR%(STACK%))=-1:
    NMS=DANNS(SRNR%(STACK%)):TRACERS="5="+NMS+"-1="+
    STR$(SRNR%(STACK%)):GOSUB 2810:GOSUB 3430:
    OSTATUS%(N%)=-1:GOTO 2030
2530 NMS=VENNS(PZAEHLER%(STACK%),SRNR%(STACK%))
2540 IF NMS="" THEN PZAEHLER%(STACK%)=6:GOTO 2510
2550 IF NMS="- oder -" THEN PZAEHLER%(STACK%)=PZAEHLER%(STACK%)+1
    :GOTO 2510
2560 GOSUB 3430
2570 IF OSTATUS%(N%)=0 THEN 3020
2580 IF VENNS(1,SRNR%(STACK%))="- oder -" GOTO 2645
2590 'und-Regel
2600 WF%=WF%(PZAEHLER%(STACK%),SRNR%(STACK%))
2605 IF OSTATUS%(N%)*WF%<>1 THEN RSTATUS%(SRNR%(STACK%))=(-1):
    NMS=DANNS(SRNR%(STACK%)):TRACERS="2="+NMS+"-1="+
    VENNS(PZAEHLER%(STACK%),SRNR%(STACK%))+STR$(WF%):
    GOSUB 2810:GOSUB 3430:OSTATUS%(N%)=(-1):GOTO 2030
2610 PZAEHLER%(STACK%)=PZAEHLER%(STACK%)+1
2620 GOTO 2510
```

```
2630 'oder-Regel
2640 WF%=WF%(PZAEHLER%(STACK%),SRNR%(STACK%))
2645 IF OSTATUS%(N%)*WF%=1 THEN RSTATUS%(SRNR%(STACK%))=1:
    NMS=DANNS(SRNR%(STACK%)):TRACERS="5="+NMS+"1="+
    VENNS(PZAEHLER%(STACK%),SRNR%(STACK%))+STR$(WF%):
    GOSUB 2810:GOSUB 3430:OSTATUS%(N%)=1:GOSUB 5010:GOTO 2030
2650 PZAEHLER%(STACK%)=PZAEHLER%(STACK%)+1
2660 GOTO 2510
2800 'Tracer eintragen
2810 IF TRACER%=MTRACER% THEN TRACERS="9**** Überlauf ****"
2820 IF TRACER%<=MTRACER% THEN TRACERS(TRACER%)=TRACERS
2830 TRACER%=TRACER%+1
2840 RETURN
3000 'Objektstatus unbekannt
3010 '-> n% : Nr des unbestimmten Objektes
3020 FOR K%=1 TO REGELN%
3030 IF DANNS(K%)=OAS(N%) THEN M%=N%:N%=K%:GOSUB 3220:N%=M%:
    GOTO 2020
3040 NEXT K%
3050 GOSUB 4410:"Fragen stellen"
3060 OSTATUS%(N%)=1%:IF OSTATUS%(N%)=1 THEN GOSUB 5010:"Constraints
testen"
3070 GOTO 2580
3200 'Regeln auf stack legen
3210 '-> n% : Nummer der (Diagnose-)Regel
3220 IF STACK%=MSTACK% THEN CLS:
    PRINT"**** Stack-Überlauf ****":END
3230 STACK%=STACK%+1
3240 SRNR%(STACK%)=N%:PZAEHLER%(STACK%)=1
3250 RETURN
3400 'suchen der Objektnummer
3410 '->NAMES$ : Name des Objektes
3420 '<-N% : Nummer des Objektes
3430 FOR N%=1 TO OBJEKTE%
3440 IF OAS(N%)=NMS THEN RETURN
3450 NEXT N%
3460 CLS
3470 PRINT"**** Objekt ";NMS;" nicht vorhanden ****"
3480 END
4000 'Diagnosen ausgeben
4010 ANZAHL%=0
4020 FOR N%=1 TO REGELN%
4030 IF (TYP%(N%)=1 AND RSTATUS%(N%)=1) THEN ANZAHL%=ANZAHL%+1:
    DIAG%=N%
4040 NEXT N%
4050 CLS
4060 PRINT $5,"D I A G N O S E N"
4070 IF ANZAHL%=0 THEN PRINT $256,
    "**** Es konnte keine Diagnose bestätigt werden ****":
    GOTO 4240
4080 IF ANZAHL%=1 THEN PRINT $256,
    "Bestätigte Diagnose : ";DANNS(DIAG%): GOTO 4240
4090 PRINT $128,"Bestätigte Diagnosen : "
4100 ZEILE%=10
4110 FOR N%=1 TO REGELN%
4120 IF (TYP%(N%)<>1 OR RSTATUS%(N%)<>1) GOTO 4210
4130 IF ZEILE%<18 THEN PRINT SZEILE%*64+5,DANNS(N%):
    ZEILE%=ZEILE%+2:GOTO 4210
4140 PRINT $968,"Zur Fortsetzung eine Taste drücken";
4150 IS=INKEY$:IF IS="" THEN 4150
4160 CLS
4170 PRINT $5,"D I A G N O S E N"
```



```

4180 PRINT $128,"Fortsetzung bestätigte Diagnosen : "
4190 Z$ILEX=10
4200 GOTO 4130
4210 NEXT N%
4220 GOTO 4240
4230 `Programmende
4240 PRINT $968,"Bitte eine Taste drücken";
4250 $s=INKEY$:IF $s="" THEN 4250
4260 CLS
4270 PRINT $5,"Was möchten Sie ....."
4280 PRINT $133,"1. Programm beenden"
4290 PRINT $197,"2. Begründung ausgeben"
4300 PRINT $261,"3. Neuer Durchlauf"
4310 PRINT $584,"Bitte wählen Sie : ";
4320 $s=INKEY$:IF $s="" THEN 4320
4330 PRINT $s:IF $s=VAL($s)
4340 ON I% GOTO 4350,6010,280
4350 CLS:PRINT $5,"Programm beendet":PRINT:END
4370 GOTO 4270
4380 `Dialog-Komponente: Frage stellen
4390 `-> n% : Nr des unbestimmten Objekts
4400 `<- i% : Antwortcode
4410 FOR L%=1 TO FRAGEN%
4420 IF OAS(N%)=FOBJEKTS(L%) THEN 4520
4430 NEXT L%
4440 `keine Frage vorhanden
4450 CLS
4460 PRINT $128,"*** Es fehlt die Frage für ";OAS(N%);" ***"
4465 PRINT $128+64,"Bitte Antwort eingeben"
4470 PRINT $256+64,"Trifft diese Tatsache zu ?"
4480 PRINT $320+64,"Antwort (Ja,Nein) : ";:INPUT $s
4490 IF (LEFT$( $s,1)="J" OR LEFT$( $s,1)="j") THEN I%=1:
  TRACERS="4="+OAS(N%)+"-1=*** Frage fehlt ***= Trifft
  nicht zu !":GOSUB 2810:CLS:RETURN
4500 I%=(1):TRACERS="4="+OAS(N%)+"-1=*** Frage fehlt ***=
  Trifft nicht zu !":GOSUB 2810:CLS:RETURN
4510 `gefundene Frage nun stellen
4520 CLS
4530 PRINT $128,FTEXTS(L%)
4540 PRINT $256,"1 - ";FAS(1,L%)
4550 PRINT $384,"2 - ";FAS(2,L%)
4560 PRINT $968,"Welche Antwort ist korrekt ? (1,2) ";
4570 $s=INKEY$:IF $s="" THEN 4570
4580 PRINT $s
4590 IF $s="1" THEN I%=1:FSTATUS%(L%)=I%:TRACERS="4="+
  FOBJEKTS(L%)+"-1="+FTEXTS(L%)+"- "+FAS(1,L%):GOSUB 2810:CLS:
  RETURN
4600 IF $s="2" THEN I%=-1:FSTATUS%(L%)=I%:TRACERS="4="+
  FOBJEKTS(L%)+"-1="+FTEXTS(L%)+"- "+FAS(2,L%):
  GOSUB 2810:CLS:RETURN
4620 FOR I%=1 TO 5:CLS:PRINT $968,"*** Antwort ist falsch ***";:
  NEXT
4630 GOTO 4520
5000 ` Constraints auswerten
5010 IF OSTATUS%(N%)<>1 THEN RETURN: ` Trifft normalerweise
  nicht zu
5020 IF CO%=0 THEN RETURN
5030 FOR Z%=1 TO CO%
5040 IF C$S(1,Z%)<>OAS(N%) THEN 5110
5050 FOR ZZ%=1 TO OBJEKTE%
5060 IF C$S(2,ZZ%)<>OAS(ZZ%) THEN 5100
5070 IF OSTATUS%(ZZ%)<>0 THEN 5110

```

```

5080 OSTATUS%(ZZ%)=CS%(Z%)
5090 TRACERS="3="+C$S(2,Z%)+"- "+STR$(OSTATUS%(ZZ%))+
  "- "+C$S(1,Z%):GOSUB 2810
5100 NEXT ZZ%
5110 NEXT Z%
5120 RETURN
6000 `Begründungskomponente
6010 CLS
6020 PRINT $5,"B E G R U N D U N G E N"
6030 PRINT $133,"Folgende Ausgaben sind möglich:"
6040 PRINT $197+64,"1. Auflistung aller möglichen Fakten"
6050 PRINT $261+64,"2. Auflistung bestimmter Fakten"
6060 PRINT $325+64,"3. Anzeige einer bestimmten Regel"
6070 PRINT $389+64,"4. Ausdruck aller bestimmter Regeln"
6080 PRINT $453+64,"5. Beenden der Begründungskomponente"
6085 IF TRACER%=NTRACER% THEN PRINT $517+64,"***** Tracer
  Überlauf *****"
6090 PRINT $848,"Bitte wählen Sie : ";
6100 $s=INKEY$:IF $s="" THEN 6100
6110 PRINT $s:IF $s=VAL($s)
6120 ON I% GOTO 6210,6310,7010,10010,4260
6170 GOTO 6090
6200 `Auflistung aller möglichen Fakten
6210 WG%=0: `alle Fakten
6220 GOTO 6340: `Auflistung von Fakten
6300 `Auflistung der bestimmten Fakten
6310 WG%=1: `nur bestimmte Fakten
6320 GOTO 6340: `Auflistung von Fakten
6330 `Auflistung von Fakten
6340 CLS
6350 IF WG%=0 THEN PRINT "Auflistung aller möglichen Fakten"
6360 IF WG%=1 THEN PRINT "Auflistung der bestimmten Fakten"
6370 Z$ILEX=2
6380 FOR M%=1 TO OBJEKTE%
6381 FOR N%=1 TO TRACER%-1
6382 GOSUB 6730:IF (TYP%<>1 AND OAS(M%)=NMS) THEN 6400
6384 NEXT N%
6385 $s="?":GOTO 6420
6400 IF SUZ=1 THEN $s="+"
6410 IF SUZ=-1 THEN $s="-"
6420 IF WG%=0 THEN PRINT $ZEILEX*64,M%,".":PRINT $ZEILEX*64+6,$s;
  " ";OAS(M%):ZEILEX=ZEILEX+2
6430 IF WG%=1 AND $s<>"?" THEN PRINT $ZEILEX*64,M%,".":
  PRINT $ZEILEX*64+6,$s; " ";OAS(M%):ZEILEX=ZEILEX+2
6440 IF Z$ILEX<15 THEN 6510
6450 PRINT $968,"Bitte eine Taste drücken ! ( E = Abbruch)";
6460 $s=INKEY$:IF $s="" GOTO 6460
6465 IF $s="e" OR $s="E" THEN GOTO 6000
6470 CLS
6480 IF WG%=0 THEN PRINT "Fortsetzung: Alle möglichen Fakten"
6490 IF WG%=1 THEN PRINT "Fortsetzung: Nur bestimmte Fakten"
6500 Z$ILEX=2
6510 NEXT M%
6520 PRINT "-----"
6530 PRINT $968,"Ende - bitte eine Taste drücken !";
6540 $s=INKEY$:IF $s="" THEN 6540
6550 GOTO 6010
6700 `Tracer aufrüßeln
6710 `-> N%
6720 `<- TYP$,NMS,SUZ,NZ$
6730 TRACERS=TRACERS(N%)
6740 TYP%=VAL(LEFT$(TRACERS,1))

```



```

6750 TRACERS=RIGHTS(TRACE$S,LEN(TRACE$S)-2):IF TRACE$S=""
    THEN RETURN
6760 FOR K%=1 TO LEN(TRACE$S)
6770 IF MID$(TRACE$S,K%,1)="" GOTO 6800
6780 NEXT K%
6790 NMS=TRACE$S:RETURN
6800 NMS=LEFT$(TRACE$S,K%-1)
6810 TRACERS=RIGHTS(TRACE$S,LEN(TRACE$S)-K%)
6820 IF LEFT$(TRACE$S,1)="" THEN
    SUZ=1:TRACE$S=RIGHTS(TRACE$S,LEN(TRACE$S)-1):GOTO 6850
6830 IF LEFT$(TRACE$S,2)="" THEN SUZ=-1:TRACE$S=RIGHTS(TRACE$S,
    LEN(TRACE$S)-2):GOTO 6850
6840 PRINT 968,"**** Fehler in Tracer ****":STOP
6850 IF LEN(TRACE$S)=0 THEN RETURN
6860 TRACERS=RIGHTS(TRACE$S,LEN(TRACE$S)-1)
6880 FOR K%=1 TO LEN(TRACE$S)
6890 IF MID$(TRACE$S,K%,1)="" GOTO 6920
6900 NEXT K%
6910 N2$=TRACE$S:RETURN
6920 N2$=LEFT$(TRACE$S,K%-1)
6930 N3$=RIGHTS(TRACE$S,LEN(TRACE$S)-K%)
6950 RETURN
7000 "Anzeige einer bestimmten Regel
7010 CLS
7020 PRINT 95,"Anzeige einer bestimmten Regel:"
7030 PRINT 9645,"Bitte Nr. oder Namen eingeben : ";:LINEINPUT BN$
7050 IF LEN(BN$)=0 GOTO 7030
7060 FOR N%=1 TO LEN(BN$)
7070 IS=MID$(BN$,N%,1)
7080 IF (ASC(IS)<48 OR ASC(IS)>57) AND IS<>" " GOTO 7130:"Name !
7090 NEXT N%
7100 N%=VAL(BN$)
7110 BN$=OAS(N%)
7120 GOTO 7190
7130 FOR N%=1 TO OBJEKTEZ
7140 IF OAS(N%)=BN$ GOTO 7190
7150 NEXT N%
7160 PRINT 3197+64,"xxx Objekt ";BN$;" nicht vorhanden ***"
7180 GOTO 7030
7190 FOR N%=1 TO TRACERZ-1
7200 GOSUB 6730
7210 IF (NMS=BN$ AND TYP%<>1) GOTO 7275
7220 NEXT N%
7230 CLS:PRINT 3197+64,"Oben ";BN$;"liegen keine Informationen
    vor."
7240 PRINT 968,"Bitte eine Taste drücken"
7250 IS=INKEY$:IF INKEY$="" GOTO 7250
7260 GOTO 6010
7270 "Begründung anzeigen
7275 ON TYP% GOTO 7281,7520,8310,7310,7810
7281 PRINT 968,"**** Fehler in Trace ****":STOP
7300 "Frage anzeigen
7310 CLS
7320 PRINT 95,BN$;" trifft ";
7330 IF SUZ=(-1) PRINT "nicht ";
7340 PRINT "zu"
7350 PRINT 3133,"Dies wurde durch die Frage ermittelt : "
7360 PRINT 3197+64,N2$
7370 PRINT 3325+64,"Die Antwort war : "
7380 PRINT 3389+64,N3$
7390 PRINT 968,"Bitte eine Taste drücken"
7400 IS=INKEY$:IF IS="" GOTO 7400

```

```

7410 GOTO 6010
7500 "und-Regel anzeigen
7520 IF SUZ=-1 GOTO 7550
7530 IF SUZ=1 GOTO 7640
7540 "Regel trifft nicht zu
7550 PRINT 3133,NMS;" ist nicht erfüllt, weil"
7570 IS=N2$
7580 IF VAL(N3$)=(-1) THEN IS="nicht "+IS
7590 PRINT 3197+64,IS;" <=== nicht zutrifft"
7600 PRINT 968,"Bitte eine Taste drücken"
7610 IS=INKEY$:IF IS="" GOTO 7610
7620 GOTO 6010
7630 "Regel erfüllt
7640 PRINT 3133,NMS;" ist erfüllt weil"
7650 PRINT 3197,"alle Prämissen erfüllt sind : "
7660 N%=VAL(N2$)
7670 FOR N%=1 TO 5
7680 IS=VENNS(N%,N%)
7690 IF IS="" GOTO 7720
7700 IF VFM(N%,N%)=(-1) THEN IS="nicht "+IS
7710 PRINT 3261+64*N%,IS
7720 NEXT N%
7730 PRINT 968,"Bitte eine Taste drücken"
7740 IS=INKEY$:IF IS="" GOTO 7740
7750 GOTO 6010
7800 "oder-Regel anzeigen
7810 CLS
7820 IF SUZ=1 GOTO 7880
7830 IF SUZ=-1 GOTO 8110
7870 "Regel trifft zu
7880 PRINT 3133,NMS;" ist erfüllt, weil"
7890 PRINT 3197,"mindestens eine Prämisse der"
7900 PRINT 3261+7,"oder - Regel"
7910 PRINT 3325,"erfüllt ist"
7920 IS=N2$
7930 IF VAL(N3$)=(-1) THEN IS="nicht "+IS
7940 PRINT 3453,IS;" <=== trifft zu"
7950 PRINT 968,"Bitte eine Taste drücken"
7960 IS=INKEY$:IF IS="" GOTO 7960
7970 GOTO 6010
7980 "Regel unbestimmt
7990 PRINT 968,"Bitte eine Taste drücken"
8000 IS=INKEY$:IF IS="" GOTO 8000
8010 GOTO 6010
8020 RETURN
8030 FOR N%=1 TO 5
8040 IF VENNS(N%,N%)<>" " THEN PRINT 3197+N%*64,IS
8050 NEXT N%
8060 PRINT 968,"Bitte eine Taste drücken"
8070 IS=INKEY$:IF IS="" GOTO 8070
8080 RETURN
8090 "Regel nicht erfüllt
8110 PRINT 3133,NMS;" ist nicht erfüllte, weil"
8120 PRINT 3197,"keine Prämisse der oder-Regel"
8130 PRINT 3261+20,"-----"
8140 PRINT 3325,"erfüllt ist"
8150 N%=VAL(N2$)
8160 FOR N%=2 TO 5
8170 IS=VENNS(N%,N%)
8180 IF VFM(N%,N%)=(-1) THEN IS="nicht "+IS
8190 PRINT 3261+N%*64,IS;" <=== trifft nicht zu"
8200 NEXT N%

```

```

8210 PRINT 968,"Bitte eine Taste drücken"
8220 IS=INKEY$:IF IS="" GOTO 8220
8230 GOTO 6010
8300 "Constraint
8310 CLS
8320 PRINT 3133,BN$;"trifft ";
8330 IF SUZ=(-1) THEN PRINT "nicht ";
8340 PRINT "zu"
8350 PRINT 3197,"Dies gilt aufgrund des Constraint"
8360 PRINT 3261,N2$;" ==> ";
8370 IF SUZ=(-1) THEN PRINT "nicht ";
8380 PRINT NMS
8390 PRINT 968,"Bitte eine Taste drücken"
8400 IS=INKEY$:IF IS="" GOTO 8400
8410 GOTO 6010
10000 "Begründung ausdrucken
10010 PRINT 968,"**** Druckroutine nicht installiert ****"
10020 GOTO 6010

```

3.3 Ausblick

Die vorgestellte Inferenzkomponente kann noch verbessert werden, wenn, dies ist in dem Editor schon berücksichtigt, durch festgelegte Eingangsfragen Hypothesen erzeugt werden. Die Hypothesen sind eine Auswahl möglicher Diagnosen, die am Anfang der Auswertung durch festgelegte Eingangsfragen erzeugt werden. Das würde einer Vorwärtsverkettung entsprechen. Die weitere Diagnose erfolgt dann wieder durch Rückwärtsverkettung. Die Routine, um Diagnosen etc. auszudrucken, ist noch nicht installiert.

Ich hoffe, daß Programm ist Anregung genug für BASIC-Programmierer

März '91

Egbert

Literaturhinweis:

1. Soill, KI Expertensysteme programmieren, Franzis Verlag München

Masken generieren mit TRS 80 MOD 1 und GENIE I Computern

Ergänzung zum Artikel von Oliver Volz im Info 32

Im Info 32 stellte uns Oliver die Möglichkeit vor, in Basic bzw. Fortran Masken zu erzeugen. Ich möchte diese Möglichkeiten ergänzen. Unter NewDos/Gdos kann man über den Befehl DUMP,Filename/...:lw,Startadresse,Endadresse,Programmstart nicht nur den Inhalt eines Programmspeicherinhaltes, sondern auch den Bildschirminhalt auf Diskette speichern. Hierauf baut das vorgestellte System auf.

Wenn man sich eine Maske erzeugt hat, sei es mit einem Editor (s. Info 32) oder über Basic Programmzeilen, so speichert man nach Ausgabe auf den Monitor sofort mit dem Befehl DUMP,MASKE/...:3C00H,3FFFH,3C00H den Bildschirminhalt ab. Im eigentlichen (Basic) Programm kann dann über LOAD"MASKE/..." die Maske wieder eingeladen werden und zu Ein/Ausgabezwecken genutzt werden.

Bei Ausgaben mit der erzeugten Maske läßt sich durch ein und ausschalten des Cursors die Maske relativ einfach erhalten. Will man eine Maske aber als Eingabe- Maske nutzen, ist das aber Schwieriger, vor allen Dingen dann, wenn eine Eingabe über mehrere Zeilen erfolgen soll. Hierzu eine Routine die über PEEK/POKE zunächst Zeichenweise aus einer Maske Eingaben in einen String (A\$) übernimmt und die eingelesenen Zeichen durch einen Punkt (.) ersetzt. der Cursor springt dabei immer an die Eingabestellen der Maske und ist dabei abgeschaltet. Dadurch wird die Maske nicht zerstört.

Zunächst wird die für das eigentliche Programm benötigte Maske mit ein paar Basic-Zeilen generiert (oder nach der Methode von Oliver). Die letzte Zeile enthält dann den DUMP Befehl

```
4000 'Fragenmaske
4010 CLS
4020 PRINT $66,"Fragennummer      : ..."
4030 PRINT $194,"Fragetext       : ..."
4032 PRINT $212,"....."
4035 PRINT $276,"....."
4040 PRINT $322,"Antworttext 1    : ..."
4042 PRINT $340,"....."
4045 PRINT $404,"....."
4050 PRINT $450,"Antworttext 2    : ..."
4052 PRINT $468,"....."
4055 PRINT $532,"....."
4060 PRINT $642,"Bezugsobjekt-Name : ....."
4070 CMD"DUMP MASKE1/DUM:1,3C00H,3FFFH,3C00H"
```

Club 80
INFO 33
März 91

Seite
25

Die gespeicherte Maske läßt sich nun aus dem DOS mit LOAD MASKE1/DUM am Bildschirm darstellen.
In einem Programm kann man die Maske durch CMD"LOAD MASKE1/DUM:1" aufrufen und seine Eingaben machen, vorausgesetzt man achtet auf die Cursorsteuerung, da sonst die Maske zerstört wird.

Ein Programm kann dann wie folgt aussehen:

```
1000 'Hauptprogramm
1010 'Abfragen z.B.:
1020 'Möchten Sie Fragen hinzufügen, auflisten, ausdrucken
1020 'oder wie auch immer die "Fragenmaske/Eingabemaske
1030 'auch aussieht
```

```
2100 'Unterroutine "Fragen hinzufügen"
2200 'Frage hinzufügen
2210 GOSUB 4010           Unterprogramm Maske laden
2230 FRAGEN%=FRAGEN%+1
2240 PRINT $83,"";:      Cursor positionieren
      PRINT CHR$(15);:    Cursor ausschalten
      PRINT FRAGEN%;      Nummer der Frage
2250 ZEILE%=2:K%=20
2255 ZEILE%=ZEILE%+1:IF ZEILE%=9 THEN ZEILE%=10:K%=22
2257 IF ZEILE%>10 THEN 2290   Ende Eingabemaske
                                erreicht
2260 PRINT $ZEILE%*64+K%,"";:
      PRINT CHR$(14);:      Cursor einschalten und
2265 A$=INKEY$:IF A$="" THEN 2265
2270 IF A$=CHR$(13) THEN PRINT CHR$(15);: alles in A$ bis <ENTER>
                                gedrückt
      GOTO 2255
2275 IF POS(0)=61          Cursor-Position prüfen;
      THEN PRINT A$;:      wenn Cursor über
                                Zeilenende, dann Cursor
                                aus
                                und nächste Zeile
      PRINT CHR$(15);:GOTO 2255:
      ELSE PRINT A$;:GOTO 2265
2290 PRINT $910,"Eintrag in Ordnung ? (j/n)";
```

```
2300 I$=INKEY$:IF I$="" GOTO 2300
2310 IF (I$<>"j" AND I$<>"J")
      THEN FRAGEN%=FRAGEN%-1:
      GOTO 2210
2320 GOSUB 4110:          Alles klar, dann Einlesen
      GOTO 1010
```

```
2600 'Fragen auflisten
2610 CLS
2620 IF FRAGEN%<>0 GOTO 2670
2630 PRINT $584,"***      keine Fragen vorhanden      ***"
2640 PRINT $652,"*** Bitte eine Taste drücken ***"
2650 I$=INKEY$:IF I$="" GOTO 2650
2660 GOTO 1010
2670 PRINT $83,"":GOSUB 4010   Ausgabemaske auf Monitor
                                Cursor steht nun auf
2680 FOR N%=1 TO FRAGEN%      Position $83
                                vorher über Maske
                                eingegebenen Fragetext
                                mit den entsprechenden
                                Begrenzungen und
                                Positionen ausgeben
```

```
2700 PRINT $212,LEFT$(FTEXT$(N%),42);:
      PRINT $212+64,RIGHT$(FTEXT$(N%),42);
2710 PRINT $340,LEFT$(FAS$(1,N%),42);:
      PRINT $340+64,RIGHT$(FAS$(1,N%),42);
2720 PRINT $468,LEFT$(FAS$(2,N%),42);:
      PRINT $468+64,RIGHT$(FAS$(2,N%),42);
2730 PRINT $662,FOBJEKT$(N%);:
```

Club 80
INFO 33
März 91

Seite
26

PRINT CHR\$(15);

nach Ausgabe Cursor
auszuschalten, damit
Ausgabemaske nicht
zerstört wird.

Seite
27

```
2740 PRINT $913,"Zum Abbruch <E> drücken"
2750 PRINT $968,"Zur Fortsetzung eine andere Taste drücken";
2760 IS=INKEY$:IF IS="" GOTO 2760
2770 IF (IS="E" OR IS="a") GOTO 1010
2780 NEXT I%
2790 GOTO 1010
```

```
4000 'Fragenmaske
4010 CMD"load maske1/dum"
4070 RETURN
4100 'Einlesen aus Fragemaske
4110 PRINT CHR$(15);;
```

Cursor ausschalten:
Positionen der Zeilen und
Spalten auf Monitor
festlegen und dann
zunächst alles in AS
packen

```
I2%=1:
I3%=2
4120 FOR I1%=0 TO 2
4130 FOR I2%=I2% TO I3%:
FOR K%=0 TO 41
4140 AS(I1%)=AS(I1%)+CHR$(PEEK((2+I1%)*64+15380+K%)):
POKE(2+I1%)*64+15380+K%,46:
NEXT K%,I2%:
I2%=I2%+2:
I3%=I3%+2:
NEXT I1%
4145 FOR K%=0 TO 39:
AS(3)=AS(3)+CHR$(PEEK(10*64+15382+K%)):
POKE(10*64)+15382+K%,46:
NEXT K%
```

Jetzt kommt alles in die eigentlichen Variablen. Bleiben wir beim
Beispiel Fragen, dann sei FTEXT\$ der Fragentext in der Maske und
FAS die Antworttexte. Alle drei sind zu speichern.

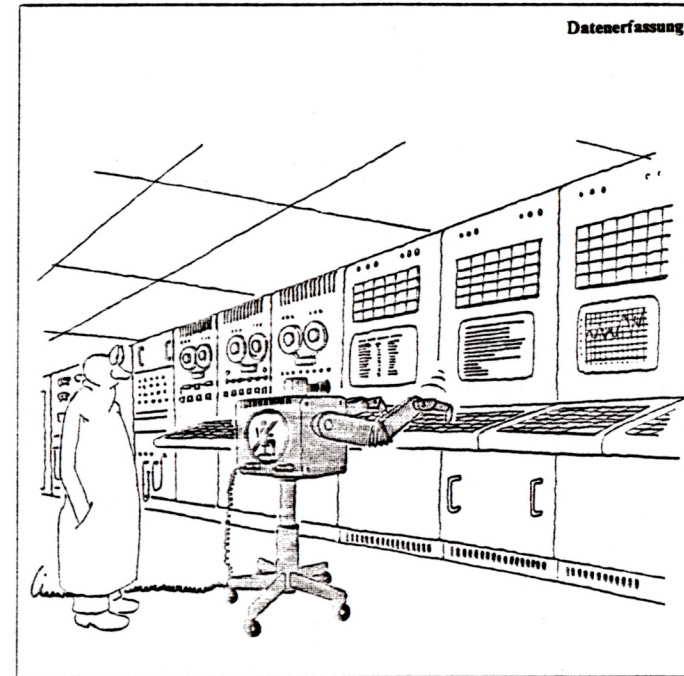
```
4150 FTEXT$(FRAGEN%)=AS(0)
4160 FAS(1,FRAGEN%)=AS(1)
4170 FAS(2,FRAGEN%)=AS(2)
4180 FAS(3,FRAGEN%)=AS(3)
```

Damit alles seine Ordnung hat, wird AS nun gelöscht.

```
4185 FOR I1%=0 TO 3:
AS(I1%)="":
NEXT I1%
4190 RETURN
```

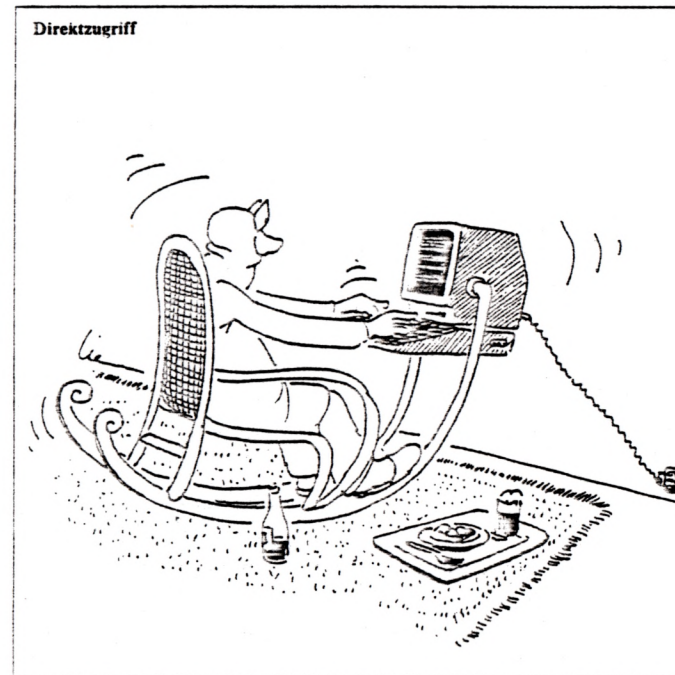
Viel Spaß beim Masken generieren
Egbert Schröder

Datenerfassung



Seite
28

Direktzugriff



Hardcopyroutine für SEIKOSHA GP 550A

Die meisten Hardcopyroutinen sind für die Drucker geschrieben, die EPSON kompatibel sind. Doch gibt es auch Drucker, die andere Steuerzeichen benötigen um Grafik ausdrucken zu können.

Hierzu gehört auch der SEIKOSHA GP 550 A. Dieser Drucker hat ein gutes Schriftbild und ist auch Grafikfähig.

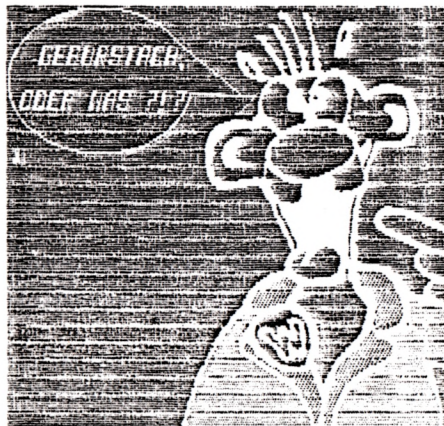
Das folgende Programm läuft auf einem GENIE I mit HRG 1B und der Software HRGPACK zusammen mit einem SEIKOSHA GP 550A.

Viel Spaß

```

10 DEFINT A-Z
20 LPRINT CHR$(27); "T16";
30 FOR Y=0 TO 191 STEP 8
40   LPRINT CHR$(27); "G383";
50   FOR X=0 TO 383
60     FOR O=0 TO 7
70       QDOT(X,Y+O,F)
80       IF F=-1 THEN DE=DE+240
90   PRINT"*";
100   NEXT
110   OUT 253,DE
120   DE=0
130 NEXT
140 LPRINT
150 NEXT

```



Das nachfolgende Programm stellt eine Hardcopyroutine für den BROTHER M 1009 und den TRS-80 Model IV/p in Verbindung mit der u-Lab HRG Karte da.

Unter der Treibersoftware steht zwar ein Hardcopy-befehl zur Verfügung, jedoch arbeitet dieser nicht zufriedenstellend mit dem M1009 zusammen. Deshalb dieses Programm.

Viel Spaß

```

1 DEFINT A-Z:CLS
50091 SOUND 1,.5
60000 LPRINT CHR$(27); "A";CHR$(8);CHR$(27); "2";
60010 FOR Y=20 TO 239 STEP 8
60011   LPRINT CHR$(27); "L";CHR$(128);CHR$(2);
60020   FOR X=0 TO 639 STEP 1
60030     FOR O=7 TO 0 STEP -1
60040       FARBE=@POINT(X,Y+O)
60050       IF FARBE=1 THEN DE=DE+2^(7-O)
60060     NEXT
60070 LPRINT CHR$(DE);:ZAHL=ZAHL+1
60071 DE=0
60080 NEXT
60090 LPRINT CHR$(10);
60091 SOUND 1,.5
60100 NEXT

```

HARDCOPY FUER TRS IV/P UND BROTHER M/1009

(C) 1990/91 FRANK SCHOOFF

Schnelle Kreisfunktion unter BASIC80

In den Zeitschriften wie c-t etc. werden bzw. wurden Programme veröffentlicht, in denen es immer um das besonders schnelle Zeichnen von Kreisen geht. Jedoch sind diese Programme alle nicht die wahren, wenn man die Geschwindigkeit liebt.

Das nachfolgende Programm soll einmal zeigen, dass es noch schneller geht als alles davor da gewesene. Es ist sogar schneller als der @CIRCLE-Befehl des GBASIC.

Der Trick: Es wird eine SIN/COS-Tabelle erstellt und diese dann beim Zeichnen ausgelesen.

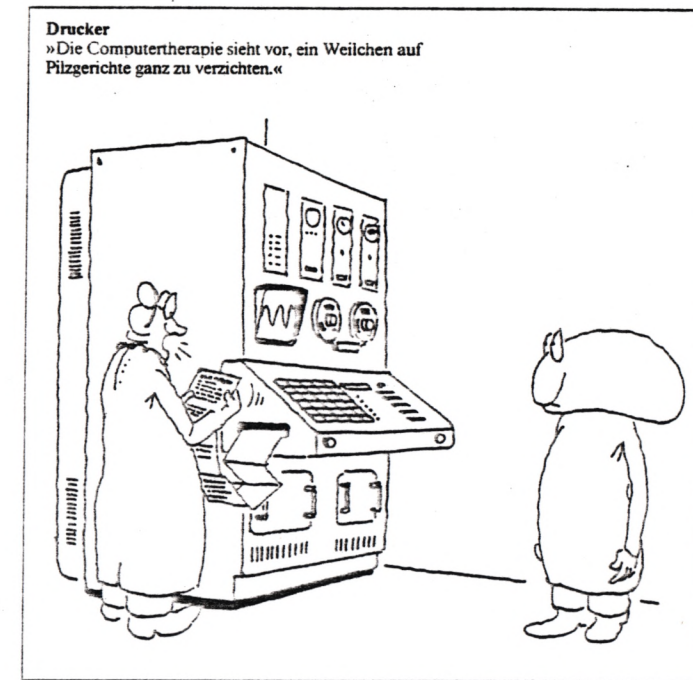
Das nachfolgende Programm zeigt das Zeichnen eines Kreises mit den Radien 200 und 80. Dazu wird eine Zeit von nur einer(1) Sekunde benoetigt!!!

Wird die Schrittweite in Zeile 30 von 10 erhöht, so lassen sich auch Vielecke zeichnen.

```

2 REM
3 REM (c) 1991 by Frank Schoof
4 REM
10 DIM XX(100),YY(100)
20 PI=4*ATN(1)
30 FOR W=0 TO 360*PI/180 STEP 10*PI/180
40   X=COS(W)
50   Y=SIN(W)
60   XX(Z)=X
70   YY(Z)=Y
80   Z=Z+1
90 NEXT
100 Z1=Z
110 @ON
120 @CLS
130 R=200:R1=80
140 CLS
150 X1=XX(0)*R+320:Y1=YY(0)*R1+120
160 PRINT TIME$
170 FOR I=0 TO Z-1
180   X=XX(I)*R+320
190   Y=YY(I)*R1+120
200   @LINE(X,Y,X1,Y1,1)
210   X1=X
220   Y1=Y
230 NEXT
240 PRINT TIME$

```



Grafikdemo fuer uLabs Grafikkarte (Model IV/IVp/III)

Ein kleines Programm zur Auflockerung der alltaeglichen Computertipperei.

```

10 DEFINT A-Z
20 Y=30
30 @CLS
40 CLS
50 @ON
60 FOR X=0 TO 239 STEP RND(20)+3
70 @LINE(Y,X,Y1-Y,120-(X/2),2)
80 Y=Y+10
90 NEXT
100 Y1=RND(600)
110 @CLS
120 Y=RND(40):GOTO 60

```

Frank Schoof

Schönere Programme mit Turbo-Pascal

Über den Sinn oder Unsinn von einheitlichen Schreibweisen läßt sich sicher schön streiten, aber ich finde z.B. Pascal-Programme, die durchgehend klein geschrieben sind, furchtbar unleserlich. Modula bietet hier, neben anderen Verbesserungen, den 'Vorteil', daß der Compiler meckert, wenn man sich nicht haargenau an die Vorschriften hält. Pascal ist in dieser Beziehung zwar viel liberaler, aber wenn man mal ein Programm von Pascal nach Modula umsetzen will, kann man das große Fluchen bekommen. Man kann es dann nämlich trotz der großen Ähnlichkeit der Sprachen fast neu abschreiben, weil der Modula-Compiler sich alle Nas' lang beschwert.

Aus einem Programm, das in einem Listing zur Verbesserung der Lesbarkeit die Schlüsselwörter hervorheben sollte, habe ich mir daher nun eines gebaut, das genau die gewünschte Umsetzung der reservierten Wörter vornimmt. Bei der Schreibweise habe ich mich dabei an die von Modula gehalten, damit, wie schon gesagt, eine Umsetzung möglichst einfach ist und ich mir keine zwei Versionen merken muß. Wenn im Hauptprogramm Include-Files benutzt werden, werden diese automatisch auch mitbearbeitet, Wörter in Kommentaren und Strings werden übergangen.

Wer das Listing nicht abtippen will, kann es von mir gerne gegen Rückporto auf Floppy haben.

PROGRAM Markres;

(* Schreibt alle reservierten Woerter in einer einheitlichen *)
(* Schreibweise *)

(*SV-*) (** Kein Range-Checking bei String-Uebergabe **)

TYPE Line = STRING [128];
Linelength = 0..128;
Bez = STRING [20];

CONST Standardbezeichner : ARRAY [1..102] OF Bez =
('Abs', 'Addr', 'Append', 'Arc', 'ArcTan', 'Assign',
'Bdos', 'BdosHL', 'Bios', 'BiosHL', 'BlockRead', 'BlockWrite', 'BOOLEAN', 'BYTE',
'Chain', 'CHAR', 'Chr', 'Close', 'ClrEol', 'ClrScr', 'Con', 'Concat',
'Copy', 'Cos', 'CrtExit', 'CrtInit',
'Delete', 'Delay', 'DelLine', 'Dispose', 'EOF', 'EOLN', 'Erase', 'Execute',
'Exp', 'FALSE', 'FilePos', 'FileSize', 'FillChar', 'Frac', 'FreeMem',
'GetMem', 'GotoXY', 'HALT', 'HeapPtr', 'Hi', 'HighVideo', 'Insert',
'InsLine', 'Int', 'INTEGER', 'IOResult', 'Kbd',
'KeyPressed', 'Length', 'Ln', 'Lo', 'LowVideo', 'Lst', 'Mark', 'MaxAvail', 'MaxInt',
'Mem', 'MemAvail', 'Move', 'New', 'NormVideo', 'Odd', 'Ord', 'ParamCount', 'ParamStr',
'Pi', 'Port', 'Pos', 'Pred', 'Ptr', 'Random', 'Randomize', 'Read', 'ReadLn', 'REAL',
'Release', 'Rename', 'Reset', 'Rewrite', 'Round',
'Seek', 'Sin', 'SizeOf', 'Sqr', 'Sqrt', 'Str', 'Succ', 'Swap', 'Tm', 'TRUE', 'Trunc',
'UpCase', 'Usr', 'Val', 'Write', 'WriteLn');

CONST ResWords : ARRAY [1..46] OF STRING [10] =
('ABSOLUTE', 'AND', 'ARRAY', 'BEGIN', 'CASE', 'CONST',
'DIV', 'DO', 'DOWNT', 'ELSE', 'END', 'END.', 'EXTERNAL',
'FILE', 'FOR', 'FORWARD', 'FUNCTION', 'GOTO', 'IF', 'IN',
'INLINE', 'LABEL', 'MOD', 'NIL', 'NOT', 'OF', 'OR', 'OVERLAY',
'PACKED', 'PROCEDURE', 'PROGRAM', 'RECORD', 'REPEAT', 'SET',
'SHL', 'SHR', 'STRING', 'TEXT', 'THEN', 'TO', 'TYPE', 'UNTIL',
'VAR', 'WHILE', 'WITH', 'XOR');

VAR InlineFlag,
KommentarFlag,
StringFlag : BOOLEAN;
EinFn, AusFn,
IniFn, IniFn2 : Bez;
EinFp, AusFp,
IniFp, IniFp2 : TEXT;
Progzeile : Line;
Check : INTEGER;

FUNCTION Gross (X : Line) : Line;
VAR I : Linelength;
BEGIN
FOR I:=1 TO Length (X) DO
X[I]:=UpCase(X[I]);
Gross:=X;
END;

FUNCTION Standard(VAR N : Bez) : BOOLEAN;
VAR B : Bez;
I, J, M : INTEGER;
flag : BOOLEAN;
BEGIN
B:= Gross(N); I:=1; J:=102;
IF NOT(KommentarFlag) AND NOT(StringFlag) THEN
REPEAT
M:=(I+J) DIV 2;
IF B<=Gross(Standardbezeichner[M]) THEN J:=M ELSE I:=M+1
UNTIL I=J;
flag:=(Gross(Standardbezeichner[I])=B);
IF flag THEN N:=Standardbezeichner[I];
Standard := flag;
END;

FUNCTION Reserviert (VAR X : Line) : BOOLEAN;
VAR I, J, M : 1..50;
H : BOOLEAN;
W : Line;
BEGIN
H:=FALSE : I:=1; J:=46; W:=Gross(X);
IF NOT(KommentarFlag) AND NOT(StringFlag) THEN
BEGIN
REPEAT
M:=(I+J) DIV 2;

```

      IF W<=Gross(ResWords[M]) THEN J:=M ELSE I:=M+1
    UNTIL I=J;
    H:=(ResWords[I]=W)
  END;
  IF H THEN X:=ResWords[I];
  Reserviert:=-H;
END;

PROCEDURE Markiere ( VAR X : Line; VAR Pointer:TEXT );
BEGIN
  Write(Pointer,X);
  LowVideo; Write(X); NormVideo;
END;

PROCEDURE Drucke ( VAR X : Line; VAR Pointer:TEXT );
VAR I : Linelength;
    A : Line;
BEGIN
  A:=''; I:=1;
  WHILE I<Length(X)+1 DO
  BEGIN
    IF X[I] IN ['^','A'..'Z','a'..'z'] THEN
    BEGIN
      A:=X[I]; I:=Succ(I);
      WHILE (X[I] IN ['.','^','0'..'9','A'..'Z','a'..'z','_']) AND (I<=Length(X)) DO
      BEGIN
        A:=A+X[I]; I:=Succ(I)
      END
    END
    ELSE
    BEGIN
      IF A<>' ' THEN
      IF Reserviert(A) OR Standard(A) THEN
        Markiere(A,Pointer)
      ELSE
      BEGIN
        Write(Pointer,A); Write(A);
      END;
      A:='';
      Write(Pointer,X[I]); Write(X[I]);
      IF NOT(StringFlag) AND ((X[I]='(') OR ((X[I]='(') AND (X[I+1]='*')) THEN
        KommentarFlag:=TRUE;

      IF NOT(StringFlag) AND ((X[I]=')') OR ((X[I]='*') AND (X[I+1]=')')) THEN
        KommentarFlag:=FALSE;

      IF NOT (KommentarFlag) AND (X[I]=Chr(39)) THEN
        StringFlag:=NOT(StringFlag);

      IF KommentarFlag AND (X[I]='$') AND ((X[I+1]='I') OR (X[I+1]='i')) AND
        (X[I+2]>'+' ) AND (X[I+2]>'-' ) THEN
        InlineFlag:=TRUE;

      I:=Succ(I)
    END
  END

```

```

  END;
END;
IF A<>' ' THEN
  IF Reserviert(A) OR Standard(A) THEN
    Markiere(A,Pointer)
  ELSE
  BEGIN
    Write(Pointer,A); Write(A);
    Write(Pointer); Write:
    StringFlag:=FALSE
  END;
  WriteLn(Pointer); WriteLn;
END;

END;

BEGIN
  IF ParamCount=0 THEN
  BEGIN
    ClrScr;
    WriteLn('Umwandlung aller reservierten Woerter in Grossbuchstaben');
    WriteLn;
    Write('Filename: '); ReadLn(EinFn);
  END
  ELSE
    EinFn:=ParamStr(1);
    IF Pos('.',EinFn)=0 THEN
      EinFn:=Concat(EinFn,'.PAS');
    Assign(EinFp,EinFn);
    Reset(EinFp);
    AusFn:='TEMP$$$TMP';
    Assign(AusFp,AusFn);
    Rewrite(AusFp);
    WHILE NOT EOF(EinFp) DO
    BEGIN
      ReadLn(EinFp,Progzeile);
      Drucke(Progzeile,AusFp);

      IF InlineFlag THEN
      BEGIN
        InlineFlag:=FALSE;
        Check:=Pos('$',Progzeile)+2;
        InlFn:=Copy(Progzeile,Check,Length(Progzeile)-Check);
        writeLn(InlFn);
        Assign(InlFp,InlFn); Reset(InlFp);
        Assign(InlFp2,'TEMP2$$$TMP'); Rewrite(InlFp2);
        WriteLn;WriteLn; WriteLn;
        WHILE NOT EOF(InlFp) DO
        BEGIN
          ReadLn(InlFp,Progzeile);
          Drucke(Progzeile,InlFp2);
          IF InlineFlag THEN
          BEGIN
            WriteLn('Verschachtelte INCLUDEs gehen noch nicht');
            HALT
          END
        END
      END
    END
  END

```



```

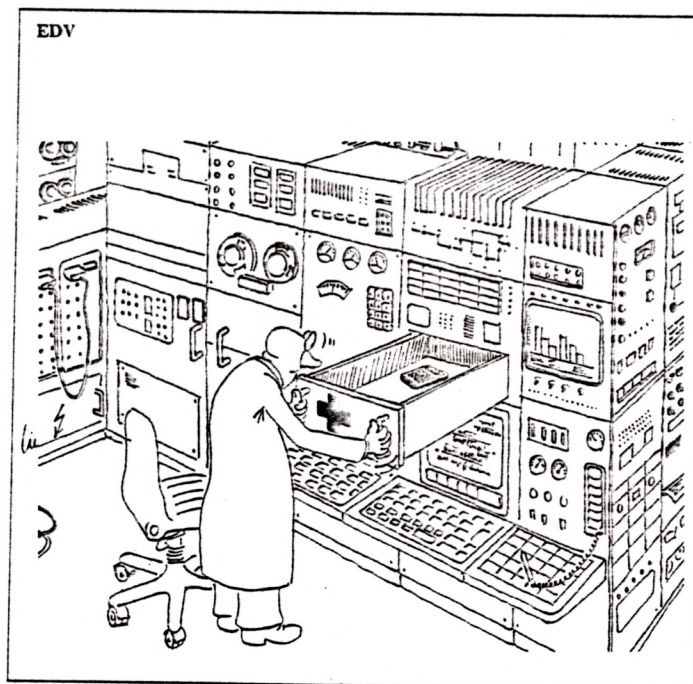
END;
Close(InlFp); Close(InlFp2);
Erase(InlFp); Rename(InlFp2,InlFn);
END;

END;
Close(EinFp);
Close(AusFp);
Erase(EinFp);
Rename(AusFp,EinFn);
END.

```

Das Programm erhebt keinerlei Anspruch auf besondere Originalität oder guten Programmierstil, aber es erfüllt seinen Zweck. Ich hatte einfach keine Lust, das Originalprogramm vollständig zu überarbeiten.

Alexander Schmid



Kleiner Terminkalender

Ist es Euch auch schon passiert, daß Ihr das Wochenende durchgehakt und dabei irgendeinen wichtigen Termin oder Ereignis verpaßt habt? Aus genau diesem Grund habe ich mir ein kleines Programmchen gebastelt, das mich jedesmal beim Booten daran erinnert, was in nächster Zeit alles zu tun ist. Das Eingabefile MEMO.DAT ist sehr einfach folgendermaßen aufgebaut: Datum des Termins, Leerzeichen, beliebiger Text, z.B.

```

19.05.91 Jahrestreffen des Club-80
22.06.91 Geburtstag von Hugo
24.12.91 Weihnachten

```

Das Programm gibt immer die Termine der nächsten 8 Wochen aus, die (hoffentlich nicht verpaßten) der letzten Woche werden als letzte Mahnung hervorgehoben dargestellt. Natürlich muß das DOS dafür eine Funktion haben, über die man die aktuelle Zeit erfahren kann, ansonsten bliebe nur die Möglichkeit, das Datum per Hand einzugeben. In meinem Startup-File wird übrigens die Existenz einiger Programme auf der Ramdisk abgefragt, sodaß der Kalender nur beim Kaltstart und nicht bei jedem Reset aufgerufen wird, aber das bleibt jedem selbst überlassen.

PROGRAM Memo;

```

CONST Tage:ARRAY[0..6] OF STRING[10] = ('Samstag ', 'Sonntag ', 'Montag ',
                                           'Dienstag ', 'Mittwoch ', 'Donnerstag ',
                                           'Freitag ');
CONST days:ARRAY[1..12] OF INTEGER = (31,28,31,30,31,30,31,31,30,31,30,31);

```

```

VAR Datum      :INTEGER;
    Tag,Monat,Jahr:INTEGER;
    Filename    :TEXT;
    Zeile       :String[80];
    Ad1,Ad2,Ad3 :String[2];
    Aj1,Aj2,Aj3 :INTEGER;
    AktJul      :INTEGER;
    Dummy       :INTEGER;

```

```

PROCEDURE GetDate(VAR datum:INTEGER);
VAR dat :ARRAY[0..4] OF BYTE;
    s   :INTEGER;
BEGIN
    s:=BDOS(105,ADDR(dat[0]));
    datum:=dat[0]+256*dat[1];
END;

```

```

FUNCTION Leap(jahr:INTEGER):BOOLEAN;
BEGIN
  Leap:=((jahr DIV 4)*4 = jahr);
END;

FUNCTION daysperyear(jahr:INTEGER):INTEGER;
BEGIN
  IF Leap(jahr) THEN
    daysperyear:=366
  ELSE
    daysperyear:=365;
  END;
END;

FUNCTION dayspermonth(month,jahr:INTEGER):INTEGER;
BEGIN
  IF Leap(jahr) AND (month=2) THEN
    dayspermonth:=29
  ELSE
    dayspermonth:=days[month];
  END;
END;

PROCEDURE BinToJul(julian:INTEGER; VAR tag,monat,jahr:INTEGER);
BEGIN
  tag:=1; monat:=1; jahr:=78;
  WHILE julian>daysperyear(jahr) DO
    BEGIN
      julian:=julian-daysperyear(jahr);
      jahr:=jahr+1
    END;
  WHILE julian>dayspermonth(monat,jahr) DO
    BEGIN
      julian:=julian-dayspermonth(monat,jahr);
      monat:=monat+1;
    END;
  tag:=julian;
END;

PROCEDURE JulToBin(tag,monat,jahr:INTEGER; VAR datum:INTEGER);
VAR year:INTEGER;
BEGIN
  datum:=0;
  year:=jahr;
  WHILE jahr>78 DO
    BEGIN
      datum:=datum+daysperyear(jahr-1);
      jahr:=jahr-1;
    END;
  WHILE monat>1 DO
    BEGIN
      datum:=datum+dayspermonth(monat-1,year);
      monat:=monat-1;
    END;
  datum:=datum+tag;
END;

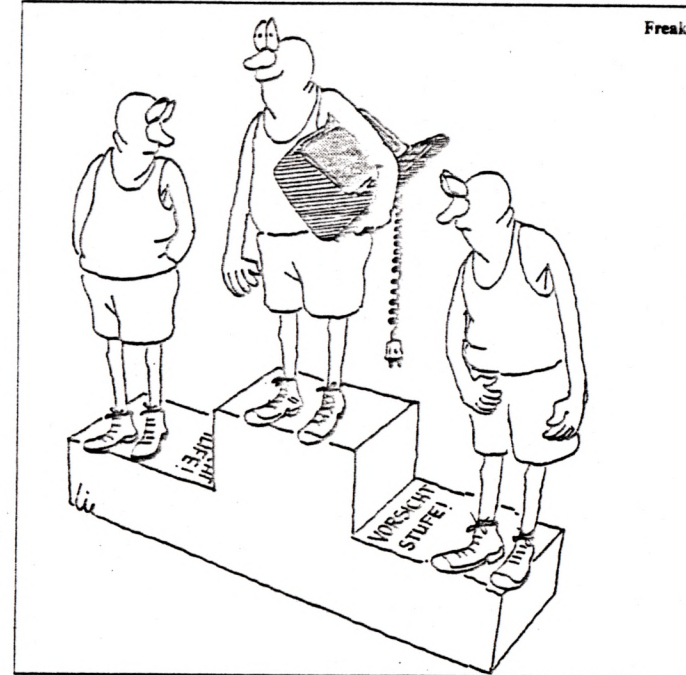
```

```

BEGIN
  GetDate(Datum);
  Assign(Filename,'E:MEMO.DAT');
  Reset(Filename);
  WHILE NOT EOF(Filename) DO
    BEGIN
      ReadLn(Filename.Zeile);
      Ad1:=Copy(Zeile,1,2); Ad2:=Copy(Zeile,4,2); Ad3:=Copy(Zeile,7,2);
      Val(Ad1,Aj1,Dummy); Val(Ad2,Aj2,Dummy); Val(Ad3,Aj3,Dummy);
      JulToBin(Aj1,Aj2,Aj3,AktJul);
      IF (AktJul<Datum) AND (Datum-AktJul<8) THEN
        BEGIN
          LowVideo; Write(TageAAktJul MOD 70,' ',Zeile); NormVideo; WriteLn;
        END;
      IF AktJul=Datum THEN
        BEGIN
          WriteLn;
          LowVideo; Write(TageAAktJul MOD 70,' ',Zeile); NormVideo; WriteLn;
          WriteLn;
        END;
      IF (AktJul>Datum) AND (AktJul-Datum<61) THEN
        WriteLn(TageAAktJul MOD 70,' ',Zeile);
      END;
      Close(Filename);
    END.

```

Alexander Schmid



Liebe Freunde im Club 80 !

Nachdem mich Fritz Chwolka nun ermutigt hat, es gäbe sicher noch einige "Nichtexperten" im Club, die sich genauso wie ich mit dem eigenen Rechner herumplagen, möchte ich hier schildern, wie ich in letzter Zeit Interesse an Assembler und ein wenig Kenntnis davon gewann.

Es fing damit an, daß ich von mehreren Freunden CP/M-PD-Software bekam, jedoch oft nur die COM-Datei. Vieles hat mich dabei gefreut, nur einige Programme nicht, denen man am Programmnamen nicht ablesen konnte, was sie wohl tun würden. Beim Durchprobieren der Programme ist mir dann immer wieder manch Malheur passiert, wenn sich so ein Programm verselbständigte und nicht mal mehr mit ^C abzubrechen war.

Mein Ärger auf die oft unbekannten Programmierer war groß, zumal sich die Verfasser auch nicht im Programmcode verewigt hatten. Es folgte mein Entschluß : Die Programme wollte ich so ändern, daß ich bei einem zukünftigen Aufruf vor unkontrolliertem Ablauf geschützt war.

Ja, wie mach' ich das nun? Disassembler hatte ich einige, der DASMB080 gefiel mir am Besten. Hier muß ich wohl erwähnen, daß mein Rechner noch eine 8085-CPU hat. Beschreibung zum DASM war zwar nicht zur Hand, aber die Befehle sind so gut wie identisch zum RESOURCE-Disassembler des Ward Christensen (diese DOC-File hatte ich).

Weiterhin brauchte ich einen Texteditor, einen Assembler und einen Linker. Ich habe da Wordstar, M80 und L80 (Macroassembler und Linker) benutzt. Es wird sich nachher zeigen, daß wegen der Besonderheiten des M80 einige Details ins Programm einzufügen sind, die vielleicht bei anderen Assemblern nicht notwendig sind. Ich habe da noch keine Ahnung.

Als Beispiel möchte ich hier das Programm WIPE anführen. Nach dem Aufruf löscht es auf der aktuellen Diskette sämtliche Dateien mit der Erweiterung .PRN, .BAK, .SYM, .REL und .HEX . Dies wird zwar mit einer Textausgabe am Bildschirm bekanntgegeben, doch dann ist es schon zu spät und der Löschvorgang läuft.

Wie ging ich nun vor ? Mit DASM WIPE versucht Dasm verschiedene noch nicht vorhandene WIPE-Files zu lesen und fragt schließlich, ob es WIPE.COM lesen solle. Hier natürlich Y für yes. Nun sollte man schon ein wenig Kenntnis vom Disassembler haben. Lest doch mal das DOC-File. Es folgt das Kommando A100,200. Die Endadresse 200 gab mir Dasm bekannt, der Start bei Adr. 100h ist bei meinem CP/M üblich. Mit diesem A-Kommando werden "mögliche" DB's gekennzeichnet (Speicherplätze mit 8 oder mehr aufeinander folgenden ASCII-Werten). Nun folgt B100,200. Es werden Symbols (Labels) gekennzeichnet. Mit L100 z. B. kann man nun die Assemblerbefehle ansehen und bei verständlichen Stellen einen Kommentar einfügen. Bei meinem Beispiel sind das die Kommentare in Großbuchstaben. Die BDOS-Aufrufe CALL 5 waren unübersehbar.

Hier lachte schon fast mein Herz, denn mir schien fast alles klar. Doch nur sachte. Meine ganze üble Erfahrung will ich hier nicht schildern, nur die Konsequenz daraus. Jetzt rasch die Assemblerfile sichern. SWIPENEU.ALL speichert alle notwendigen WIPE-Dateien und fragt, ob eine .ASM-Datei gewünscht wird. Ja

natürlich in meinem Fall. In die offene Datei wird zuerst mit P100,200 der sogenannte Prolog, der Assemblervorspann geschrieben, danach mit L100,200 die Mnemo-Befehle und dann die Datei mit dem Z-Befehl geschlossen, welcher ein 'END' an den Schluß anfügt. Das erste Drittel war getan.

Probeassemblieren, und schon zeigte der Assembler einen Fehler, das heißt, daß dieser so superschlaue Disassembler Murks gemacht hat (warum ? wer's weiß soll doch mal schreiben). Doch schwierig wars hier nicht. Es fehlte nur das Label L016F vor der Zeichenkette, die ausgegeben werden sollte (WIPE Version ...). Nun tritt der Texteditor in Aktion, mit dem das Verbessern kein Problem ist. Gleichzeitig habe ich dann die anderen Kommentare eingefügt, jetzt alle in Kleinbuchstaben (damit man's besser unterscheiden kann).

Das eingefügte Programmstück will ich hier nicht näher erläutern, es ist deutlich gekennzeichnet. Es fehlen nur noch die nachzutragenden Definitionen für bdos und wboot sowie der neu auszugebende Text (EXIT mit X, ...) und das ende:-Label. Für den M80-Assembler war dann noch die Voranstellung von 'aseg' nötig und eine Titelangabe für den Printerausdruck.

Halt, da war doch der Ärger mit dem unbekannten Programmierer ! Also schnell zu Anfang noch etwas Text in DB's eingefügt. Und wo startet nun das Programm? Mit Text bei Adresse 100h? Nee, so gehts nicht. Also 'ORG 103h', am Ende 'END start', start:-Label beim ersten Befehl und die Sache läuft, denn nun beginnt der Text bei 103, und der Linker hat von 100 bis 102 Platz, den Sprung auf die Startadresse einzutragen. Ob alle Linker das können weiß ich jedoch nicht.

Wieder assemblieren, linken und starten. Da zeigt sich dann, ob alles richtig gemacht wurde. Ich habe recht lange gebraucht, um diese ganze komplexe Angelegenheit soweit verstehen zu können. Bin halt ein Laie in Programmieretechniken. Ich hoffe jedoch, daß sich mehrere Clubfreunde finden, die manches besser wissen und dies hier dann schreiben, so daß ich irgendwann mal kein Anfänger mehr sein werde.

Herzliche Grüße

Uwe Schoberth
P.-Waldus-Str. 14
W-7136 Ötisheim

```
TITLE 'WIPE (VERSION 1.1) , März 1991 ' ; für M80
aseg ; für M80
ORG 0103H ; von 100h auf 103h (wegen Sprung auf start:)
db 'WIPE (Version 1.0), gefunden in der PD-Software'
db 'Änderung Uwe Schoberth, W-7136 dtisheim.'
db 'Neue Versions-Nr. 1.1, März 91'
```

```
; Das Programm wurde mit DASM rückassembliert
; und eine Abfrage eingefügt, um einen ungewollten
; Programmablauf (löschen von Dateien) zu verhindern.
```

```
L0000 EQU 00H
L0005 EQU 05H
bdos equ 10005 ; neu
wboot equ bdos ; neu
```

```
; BEZUGSLAUFWERK ERMITTELN
```

```
start: MVI C,19H ; 'start:' ist neu
```

```
CALL L0005
ADI 'A'
STA L01D0
```

```
; ZEICHENKETTE AUSGEBEN
```

```
MVI C,9
```

```
; " WIPE VERSION ...
```

```
LXI D,L016F
CALL L0005
```

```
; hier ist die richtige Stelle für eine Abfrage vor
; Programmablauf
; BDOS-Funktion Zeichenkette ausgeben
; die Zeichenkette steht beim Label 'text:'
```

```
mvi c,9h
lxi d,text
call bdos
mvi c,1h
call bdos
cpi 'X'
jz ende
cpi 'x'
jz ende
```

```
; BDOS-Funktion Konsoleneingabe erwarten
```

```
; ist es ein 'X' ?
; ja, Schluß jetzt
; oder ein 'x' ?
; dann auch Schluß
; Ende der Einfügung
```

```
LXI H,L013B
```

```
L0115:
```

```
MOV A,M
INX H
STA L0157
CPI '$'
JZ L0000
MOV A,M
INX H
STA L0158
MOV A,M
INX H
STA L0159
```

```
; DATEI LÖSCHEN
L0129:
```

```
MVI C,13H
LXI D,L014E
PUSH H
CALL L0005
POP H
CPI 0FFH
JZ L0115
JMP L0129
```

```
L013B:
```

```
DB 'PRN'
DB 'BAK'
DB 'SYM'
DB 'REL'
DB 'HEX'
DB 'SYMS'
```

```
L014E:
```

```
NOP
DB '????????'
```

```
L0157:
```

```
DB ' '
```

```
L0158:
```

```
DB ' '
```

```
L0159:
```

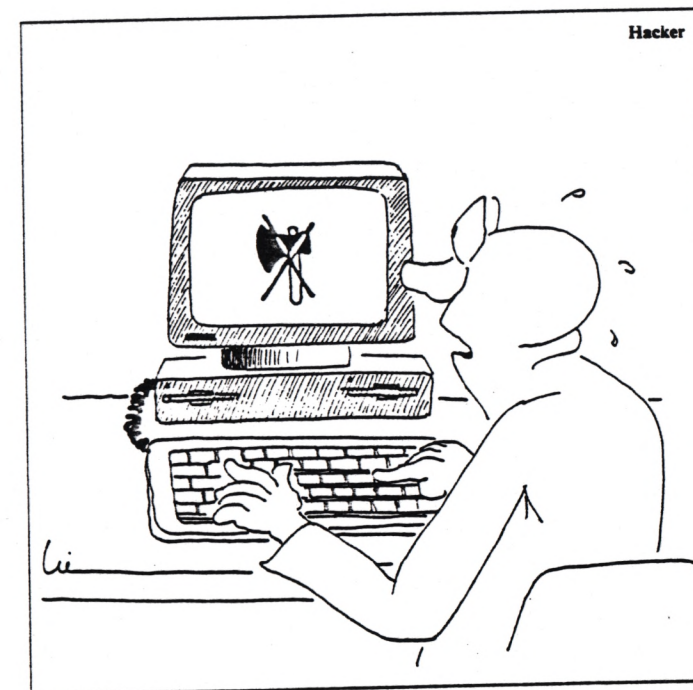
```
DB ' '
DS 0017H
```

```
L016F:
```

```
DB 'Wipe Version 1.1',0DH,0AH ; neue Versionsnummer
DB '-----',0DH,0AH
DB 0AH,'Erasing *.PRN, *.HEX, '
DB '*.SYM, *.REL and *.BAK from '
DB 'Drive '
```

```
L01D0:
```

```
DB ' ' ; Speicher für akt. Laufwerk
DB ':',0AH,0DH,'$'
text: db 0ah,0dh,'EXIT mit X , weiter mit <RET> . $' ; neu
ende: jmp wboot ; neu
END start ; neu
```



Unter falschem Namen ALIASES mit ZCPR

Bei Betriebssystemen, die hauptsächlich mit Floppies arbeiten, ist man ja immer bestrebt, nach Möglichkeit Platz zu sparen. Unter CP/M bieten sich Libraries dazu förmlich an, vor allem, da man so auch das Problem mit der begrenzten Anzahl an Files elegant umgehen kann. Ich habe z.B. ein File namens CHAIN.LBR, in dem Turbo-Pascal Chain-Files, und eine zweite Library namens COMMAND.LBR, in der sonstige COM-Files sind. Zwar gibt es Programme, die Files direkt aus solchen Libraries holen und ausführen können, aber die Tipperei ist dabei direkt proportional zum Platzgewinn. Viel Platz -> viel Tipperei, wenig Tipperei -> wenig Platzgewinn. Der Aufruf müßte dann z.B. folgendermaßen aussehen: "-R DDTZ". Sieht ja gar nicht so schlecht aus, aber das geht leider nur, wenn die Library auf dem aktuellen Laufwerk ist. Ansonsten heißt die Kommandozeile z.B. schon "-R -E:COMMAND.LBR DDTZ" und da hört der Spaß langsam auf.

Außer man schiebt sich ein ALIAS, dann kann man nämlich trotzdem einfach nur DDTZ eintippen und der Rest geht von ganz alleine. Zauberei? Keineswegs! Da der Command-Processor in diesem Fall natürlich kein DDTZ.COM findet, aktiviert er den ECP, den sog. Enhanced Command Processor und dieser sieht dann nach, ob er ein (Text-)File namens ALIAS.CMD findet. In diesem steht nun hoffentlich, was der Befehl DDTZ in Wirklichkeit bewirken soll und zwar ohne Netz und doppelten Boden so:

```
DDTZ      -R -E:COMMAND.LBR DDTZ $*      (mit Laufwerksangabe)
DDTZ      -R -UTIL:COMMAND.LBR DDTZ $*    (mit Directorynamen)
```

Statt DDTZ wird der gesamte Rattenschwanz rechts davon ausgeführt, sodaß DDTZ.COM aus der Library ausgepackt und mit sämtlichen Parametern aus der ursprünglichen Kommandozeile (das ist das \$*) gestartet wird. Entsprechendes geht mit den Chain-Files:

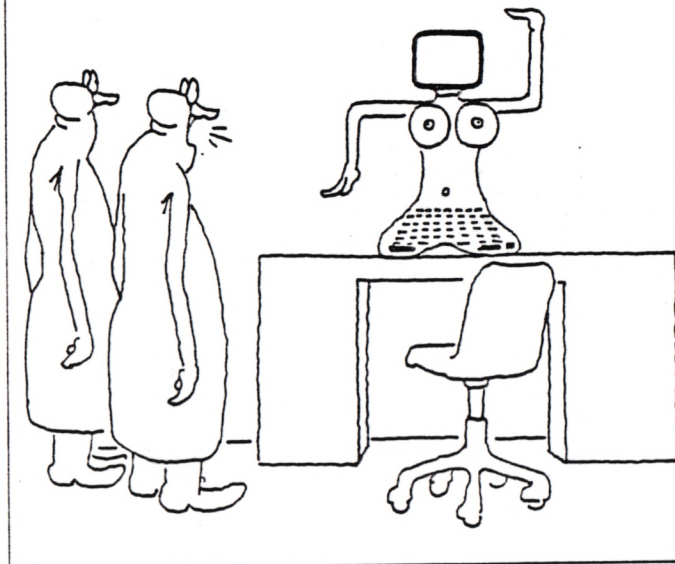
```
DPAR      -C -E:CHAIN.LBR DPAR $*
```

Es ist also völlig egal, ob die Files als 'echte' COM-File vorliegen, oder in irgendeiner Library, womöglich auf einem ganz anderen Laufwerk (Stichwort Harddisk), eingepackt sind. Den ganzen Verwaltungsaufwand nimmt uns ab jetzt das DOS ab. Wenn man will, kann man das ganze natürlich bis zum Abwinken treiben und noch ein paar 'Schmankerl' einbauen.

```
LOOK      IF EX $1;TYPE $1;ELSE;ECHO Schlafmütze;FI
```

Wenn das File vorhanden ist (EXist), wird es aufgelistet, ansonsten bekommt man einen netten Hinweis. Im Grunde ist nur die Phantasie die Grenze. Man könnte z.B. noch prüfen, ob es ein lesbares TXT-File ist und bei COM- und anderen binären Files die Floppy formatieren...

Hardware
»Wir versprechen uns von dieser Hardware eine
ungeheuer Marktbelegung.«



Mit folgendem Feature kann man z.B. DOSlern das Leben leichter machen:

```
ERA,SE=DEL,ETE      ERA $*
```

Durch diesen Trick bewirken ERA, ERASE, DEL und DELETE alle dasselbe, nämlich daß der interne ERA-Befehl ausgerufen wird. Die Verwendung ist also keineswegs auf den Start von Programmen beschränkt.

Noch ein Tip am Rande: Wenn man sicher weiß, daß der nächste Befehl für den ECP, d.h. als Alias gedacht ist, kann man mit einem Schrägstrich oder einem Blank vor dem eigentlichen Befehl diesen sofort an den ECP weiterreichen. So spart man die Sucherei nach dem COM-File und es geht schneller.

Übrigens, hatte da mal jemand gesagt, CP/M sei tot? Ich würde so was gerne mal unter DOMESDOS sehen (aber bitte nicht mit hunderten von Batch-Files, die für 20 Bytes Text 1K auf der Harddisk belegen). Die ALIASES stehen natürlich alle zusammen im ALIAS.CMD, fein säuberlich untereinander:

```
DDTZ      -R -E:COMMAND.LBR DDTZ $*
DPAR      -C -E:CHAIN.LBR DPAR $*
LOOK      IF EX $1;TYPE $1;ELSE;ECHO Schlafmütze;FI
ERA,SE=DEL,ETE      ERA $*
```


DU3 für CP/M Plus

Für ZCPR33 (CP/M2.2) gibt es ein tolles Disk-Utility mit dem Namen DU3. Damit kann man auf der Floppy alles machen, was man im täglichen Leben so braucht, aber ich will hier gar keine Werbung dafür machen. Gorm Helt-Hansen hat sich nun die Arbeit gemacht, das Programm an Z3PLUS anzupassen, aber die Benutzer von CP/M Plus (ohne Z3) schauen weiterhin in die Röhre, weil das Programm sich seine Steuerzeichen für den Bildschirm aus dem sog. TCAP des Z3PLUS holt, wo diese praktischerweise stehen. Da im normalen CP/M Plus die Terminalfunktionen nirgendwo allgemein zugänglich sind, funktioniert fast nichts mehr und das Ding ist absolut wertlos. Abhilfe: man spiele Z3PLUS und baue sich sein TCAP selber. Die richtigen Steuerzeichen müßte man theoretisch auch per Debugger in das COM-File eintragen können, aber die Initialisierungsroutinen stammen aus einer Library, deren Source-Code ich nicht habe und daher nicht weiß, wo was stehen muß. Aber warum kompliziert, wenn es auch viel einfacher geht.

Der Anfang des ersten Sektors von DU3 sieht so aus:

```
0100 C3 9A 01 5A 33 45 4E 56 01 00 00 49 4E 49 54 49 C..Z3ENV..>INITI
0110 41 4C 20 4D 41 43 52 4F 53 3A 47 30 2C 44 0D 2D AL MACROS:GO.D.-
0120 44 0D 2B 44 0D 2D 38 44 0D 2B 38 44 0D 54 30 2C D.+D.-8D.+8D.T0,
```

An der Stelle 109/10A trägt der Lader im Z3PLUS die Adresse des sog. Environments ein, womit die Steuerzeichen für die Initialisierungsroutinen zugänglich sind und genau hier greifen wir ein. An dieser Stelle kann nämlich eine beliebige Adresse stehen, sie wird vom Lader später einfach sang- und klanglos überschrieben, aber eben NUR unter Z3PLUS. Das Programm geht im Urzustand bis 3E7F, sodaß wir unser Pseudo-TCAP einfach bei 3E80 dahinter hängen und diese Adresse natürlich bei 109/10A eintragen. Das alte CP/M Plus verändert diese Adresse nun eben NICHT (wieso auch) und bei der Initialisierung wird unser Kuckucksei verwendet. Der Sektor sieht dann so aus:

```
0100 C3 9A 01 5A 33 45 4E 56 01 80 3E 49 4E 49 54 49 C..Z3ENV..>INITI
0110 41 4C 20 4D 41 43 52 4F 53 3A 47 30 2C 44 0D 2D AL MACROS:GO.D.-
0120 44 0D 2B 44 0D 2D 38 44 0D 2B 38 44 0D 54 30 2C D.+D.-8D.+8D.T0,
```

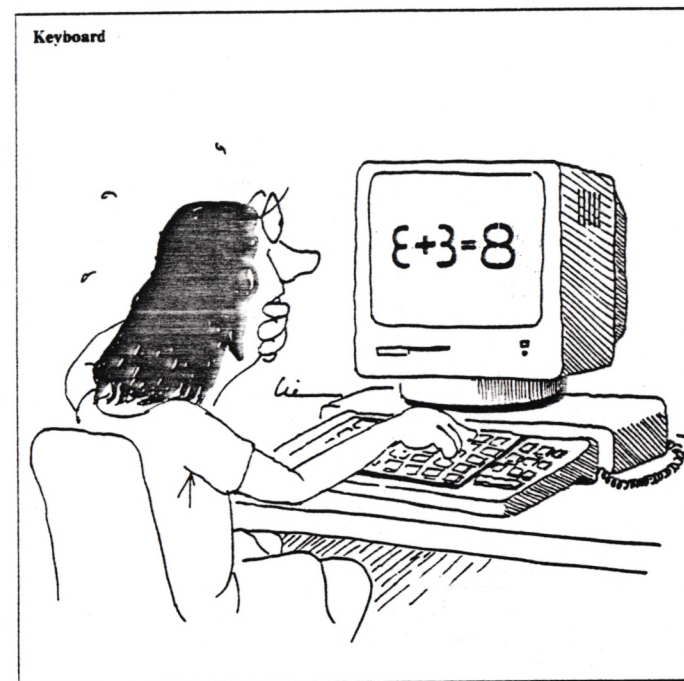
Das benötigte TCAP steht nun genau 128 Bytes hinter dem Anfang des Environments, sodaß wir bei 3E80 etwas Platz lassen müssen. Der Speicher hinter DU3 muß also folgendermaßen gefüllt werden:

```
3E80 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 .....
3E90 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 .....
3EA0 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 .....
3EB0 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 .....
3EC0 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 .....
3ED0 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 .....
3EE0 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 .....
3EF0 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 .....
```

```
3F00 43 50 55 32 38 30 2C 56 54 35 32 20 20 20 20 01 CPU280,VT52 .
3F10 00 00 00 00 00 07 00 1B 45 1B 48 00 1B 59 25 2B .....E.H..YZ+
3F20 20 25 2B 20 00 1B 4B 00 1B 70 00 1B 71 00 00 00 %+ ..K..p..q...
3F30 1B 4D 00 1B 4C 00 1B 4A 00 00 1B 46 00 1B 47 00 .M..L..J...F..G.
3F40 1B 78 35 00 1B 79 35 00 66 00 63 00 65 00 64 00 .x5..y5.f.c.e.d.
3F50 61 00 60 00 1B 70 20 1B 71 00 69 00 73 00 75 00 a.'..p..q.i.s.u.
3F60 62 00 74 00 76 00 00 00 00 00 00 00 00 00 00 b.t.v.....
3F70 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 .....
```

Die Steuerzeichen hier gelten für ein VT52, das TCAP für z.B. ADM3A kann Euch aber jeder Z3PLUS-Besitzer geben. Die Initialisierungsroutine denkt nun, daß das TCAP hier steht und holt sich die Steuerzeichen wie gewünscht ab. Damit läuft das Programm natürlich (unter CP/M Plus !!!) nur mit diesem einen Terminal. Unter Z3PLUS paßt es sich aber weiterhin automatisch an.

Alexander Schmid



Adreßautomat, die Zweite

Vor Urzeiten stellte ich hier einmal einen Patch in TSCRIPS vor, mit dem häufig gebrauchte Texte, in der Hauptsache natürlich Adressen, rationell in andere Texte, also z. B. Briefköpfe, eingebaut werden können. Das war im Clubinfo Nr. 19. Hier soll nur noch kurz angeschnitten werden, was man dazu wissen und machen muß:

Auf der Arbeitsdiskette befindet sich ein File namens ADRESSEN/LIB. Es enthält eine Bibliothek der wiederkehrenden Texte. Jeder ist genau einen Sektor lang. Seine beiden ersten Zeichen sind das Suchkriterium, etwa die Anfangsbuchstaben des Namens.

Im Haupttext, also z. B. einem Briefformular, ist an der gewünschten Stelle ein Block von genau 254 Bytes Länge mit dem Blocknamen A markiert. In die ersten beiden Stellen des Blocks werden die beiden Suchzeichen eingetippt. Drückt man nun die Tastenkombination @-K, dann sucht TSCRIPS selbsttätig in ADRESSEN/LIB nach der Adresse und überträgt sie in den Block. Nebenbei wird damit auch das aktuelle Datum in einen evtl. vorhanden Block namens D eingetragen. Wer seinerzeit den notwendigen Patch in TSCRIPS nicht vornahm, braucht es allerdings gar nicht erst zu versuchen.

Diese Erleichterung ist besonders dann willkommen, wenn eine öfter gebrauchte Adresse mit kompliziert einzugebenden SteuerCodes gespickt ist. Hier ein besonders fieses Beispiel:

Marilyn Monroe
c/o C P G I
Station du Bénin
B. P. 08-0948
Cotonou
R. du Bénin

In der TSCRIPS-Codierung sieht das folgendermaßen aus (Druckercodes des NEC P6, auf IBM-Zeichensatz eingestellt - weitgehend Epson-"Norm"):

SW&01&Marilyn Monroe
&1B5300&c&1B54&/&1B5301&o&1B54& &1C4502&C P G I
&1C4501&Station du B&82&nin
B. P. 08-0948&1C4502&
Cotonou&1C4501&
R. du B&82&nin&1C4500&

So weit der Schnee von gestern. Die Adressenautomatik versagte leider beim Beschriften von Adreßaufklebern. TSCRIPS hat nämlich den Trieb, nach jedem Text ans Ende der Seite zu gehen. Dabei überspringt der Drucker dann die nächste Million kleiner Aufkleber, bevor der folgende zum Bedrucken an der Walze bereitsteht. Dem ist nun abgeholfen:

Ein kleines Anwenderprogramm macht im Prinzip dasselbe wie der Adressenautomat von TSCRIPS, verkneift sich aber dessen Unarten. Zusätzlich wird die Absenderadresse mit ausgedruckt, so daß man sich auch diese auf dem Umschlag schenken kann. Das kann dann z. B. so aussehen:

Arnulf Sopp, Wakenitzstr. 8, D-2400 Lübeck 1

Herrn
H. Bernhardt
Hafenstr. 7

2305 Heikendorf

Club 80
INFO 33
März 91

Seite
49

Das Programm berücksichtigt auch die TSCRIPS-Sonderzeichen \$ und &, allerdings keine weiteren. Mehr davon einzubauen, erschien mir überflüssig, weil mehr in einer Adresse nicht zu erwarten ist. Gegf. kann man sich anders behelfen: Alles, was TSCRIPS mit Hilfe bestimmter reservierter Codes kann, kann der Drucker schon lange. So läßt sich z. B. eine Unterstreichung ohne weitere TSCRIPS-Unterstützung einfügen mit \$-&01& (Epson

usw.).

Ein anderes TSCRIPS-Feature wurde jedoch berücksichtigt, das durchaus vorkommen kann: Wenn nämlich der User z. B. zusätzlich die Telefonnummer, den Geburtstag oder etwas anderes über den Adressaten in einer Kommentarzeile (eingeleitet mit ">") abgelegt hat. Auch andere Kommandos, etwa Randeinstellungen, (eingeleitet mit ">") im Adreßblock wären denkbar. Sie werden nicht mit ausgedruckt.

Ein einzelner Aufkleber ist fix mit der Hand beschriftet; man wird dafür kaum das Schreibpapier aus dem Drucker nehmen, die Endlos-Aufkleber einführen und das Programm starten. Es geht in der Praxis also wohl immer um mehrere Adressen. Anschriften von Gruppen (z. B. Mitgliederliste des Clubs) stehen sinnvollerweise hintereinander in ADRESSEN/LIB. Die Suche nach einem Namenskürzel braucht also nicht jedesmal von vorne in der Datei zu beginnen. Dies berücksichtigt das Programm, indem es nach Erreichen des Dateiendes noch einmal von vorne sucht, dann aber endgültig den Suchfehler anzeigt.

Im Hinblick darauf, daß in der Regel zusammengehörige Gruppen, also auf dem File wohl auch benachbarte Adressen ausgedruckt werden sollen, gibt es auch ein "von-bis"-Feature. Man kann mit der Eingabesyntax AA-ZZ den Ausdruck mehrerer Adressen veranlassen. Diese Bequemlichkeit machte ein zusätzliches Feature erforderlich:

Stellen wir uns vor, die erste eingegebene Adresse gibt es wohl, die zweite aber nicht (Tippfehler). Das Programm findet die erste. Es sucht überhaupt nur nach ihr, um anschließend alle folgenden zu drucken, bis ein Vergleich mit der zweiten Eingabe zeigt, daß schließlich auch sie gefunden wurde. Wurde sie natürlich nicht, weil es sie in unserem Beispiel nicht gibt. Also weiter drucken, bis ADRESSEN/LIB vollständig ausgelutscht ist!

Dem ist mit einem gemeinen Trick abgeholfen: Ein Probelauf ohne Druckerbeteiligung wird durchgeführt, wobei das zweite Kürzel das Suchkriterium ist. Sollte er einen Fehler zur Folge haben, der auch gemeldet wird, wird das Unternehmen abgebrochen und nach einem oder zwei neuen Kürzeln gefragt. Bei erfolgreicher Suche wird der FCB wieder auf den Fileanfang gestellt, denn das erste Kürzel meint natürlich eine davor liegende Adresse. Ein solcher Probelauf mit dem ersten Kürzel erübrigt sich, weil nach ihm tatsächlich gesucht wird.

Das Programm ist ganz und gar auf die Aufkleber abgestimmt, die ich z. Zt. benutze: 88.9 X 35.7 mm. Nötigenfalls braucht für andere Abmessungen nur die Länge des Zeilenvorschubs pro Aufkleber geändert zu werden.

Es versteht sich, daß alle im Programm vorkommenden Druckercodes gegf. dem eigenen Gerät angepaßt werden müssen. Die Kommentare erklären, was da gerade gesteuert wird. Und dann wäre da noch das heikle Problem mit den DOS-Aufrufen. Mein DOS hat mit dem Original nur noch eine entfernte Ähnlichkeit. Sollte etwas nicht klappen, muß der geneigte Leser das dort aufgerufene Feature (s. Kommentare) eben selber programmieren oder gegf. die bei ihm zutreffende Adresse im Original-DOS verwenden.

Arnulf

00001 :	Utility zum Beschriften von Adreßaufklebern
00002 :	die Adreßbibliothek ADRESSEN/LIB muß vorhanden sein
00003	
5200 00004	ORG 5200h
00005	
00006 :	Anschriftenfile ADRESSEN/LIB eröffnen
5200 00007 start	LD HL,filenam ;Dateiname
5203 118453	LD DE,fcbl ;FCB für ADRESSEN/LIB
5206 CD1C44	00009 CALL 441ch ;Namen in den FCB übertragen
5209 210054	LD HL,secbuff ;Sektorbuffer
520C 45	00011 LD B,L ;logische Recordlänge = 256 (L = 00h)
520D CD2444	00012 CALL 4424h ;Datei eröffnen
5210 206D	00013 JR NZ,disperr ;Ende, falls ADRESSEN/LIB nicht existiert
00014	
00015 :	Drucker auf Bereitschaft testen, Font des Druckers einstellen

Club 80
INFO 33
März 91

Seite
50


```

5212 3EFE 00016 LD A,Ofeh ;Requestcode für SYS28/SYS
5214 0E05 00017 LD C,05h ;Zeiger auf Druckertest-Routine in SYS28
5216 C00244 00018 CALL 4402h ;dort RST 28h: SYS28/SYS aufrufen
5219 211E53 00019 LD HL,prtinit ;Initialisierungsdaten des Druckers
521C C06A44 00020 CALL 446ah ;ausgeben
00021
00022 ;eine Adresse oder "von-bis" mit Kürzel/n erfragen
521F 210153 00023 newname LD HL,getaddr ;Frage nach dem/den Adresskürzel/n
5222 C06744 00024 CALL 4467h ;anzeigen
5225 217E53 00025 LD HL,inbuff ;Eingabepuffer
5228 0605 00026 LD B,05h ;max. 5 Zeichen für das/die Namenskürzel
522A C0D905 00027 CALL 05d9h ;Zeichen einlesen
522D DA2D40 00028 JP C,402dh ;ENDE, falls mit BREAK abgebrochen
5230 3A8053 00029 LD A,(inbuff+2) ;dort CR oder "bis"-Zeichen
5233 FE2D 00030 CP ' ' ;für "bis"? (zwei Kürzel "von-bis")
5235 2A7E53 00031 LD HL,(inbuff) ;1. Namenskürzel
5238 2022 00032 JR NZ,oneaddr ;falls nur ein Kürzel
00033
00034 ;es wurden zwei Kürzel für "von-bis" eingegeben
523A E5 00035 PUSH HL ;1. Kürzel retten
523B 2A8153 00036 LD HL,(inbuff+3) ;2. Kürzel
523E 227E53 00037 LD (inbuff),HL ;als 1. patchen
5241 3EC9 00038 LD A,0c9h ;Opcode RET
5243 329752 00039 LD (found2),A ;Druckroutine unwirksam machen
5246 3E01 00040 LD A,01h ;Flag: nur 1 Durchlauf der Suche, weil am
5248 08 00041 EX AF,AF' ;Fileanfang begonnen
5249 C06752 00042 CALL readsec ;Programm aufrufen (Probelauf)
524C E1 00043 POP HL ;1. Kürzel
524D 227E53 00044 LD (inbuff),HL ;restaurieren
5250 3E21 00045 LD A,21h ;Opcode LD HL,nn
5252 329752 00046 LD (found2),A ;Druckroutine restaurieren
5255 20C8 00047 JR NZ,newname ;falls Adresse nicht vorhanden
5257 C03F44 00048 CALL 443fh ;FCB auf Fileanfang zurückstellen
525A 1803 00049 JR alladdr ;dort weiter
00050
00051 ;eine einzelne Adresse suchen und ausdrucken
525C 228153 00052 oneaddr LD (inbuff+3),HL ;1. Kürzel als 2. patchen (wg. Vergleich)
00053
00054 ;gemeinsam weiter bei 1 und 2 Kürzeln
525F C06452 00055 alladdr CALL prtaddr ;Such- und Druckprogramm aufrufen
5262 188B 00056 JR newname ;von vorne weiter
00057
00058 ;eine Adresse gem. dem Kürzel finden und ausdrucken
5264 3E02 00059 prtaddr LD A,02h ;Flag: 1. Durchgang bei der Suche
5266 08 00060 EX AF,AF' ;retten
5267 C0D652 00061 readsec CALL loadsec ;Sektor laden
526A 200D 00062 JR NZ,notfnd ;falls Ende der Datei oder anderer Fehler
00063
00064 ;im Adressenfile nach demselben Kürzel suchen
526C 2A0054 00065 LD HL,(secbuff) ;Namenskürzel in ADRESSEN/LIB
526F ED487E53 00066 LD BC,(inbuff) ;1. eingegebenes Namenskürzel
5273 ED42 00067 SBC HL,BC ;bei Gleichheit ist das Z-Flag gesetzt
5275 2820 00068 JR Z,found2 ;falls die richtige Anschrift gefunden
5277 18EE 00069 JR readsec ;bis zum 1. gefunden Kürzel
00070
00071 ;ADRESSEN/LIB ist überschritten, diese Anschrift gibt es evtl. nicht
5279 FE1C 00072 notfnd CP 1ch ;Fehler "Ende der Datei angetroffen"?
527B 2805 00073 JR Z,seekerr ;falls ja
527D FE1D 00074 CP 1dh ;Fehler "Hinter Ende der Datei"?
527F C2D944 00075 disperr JP NZ,4409h ;ENDE, falls anderer Fehler
5282 08 00076 seekerr EX AF,AF' ;Durchgangszähler
5283 3D 00077 DEC A ;schon zweiter Durchgang?
5284 2806 00078 JR Z,adrfail ;falls ja: Adresse nicht vorhanden
5286 08 00079 EX AF,AF' ;falls nein: Zähler wieder retten

```

```

5287 C03F44 00080 CALL 443fh ;FCB auf Anfang des Files stellen
528A 180B 00081 JR readsec ;ab Fileanfang weiter versuchen
00082
00083 ;Fehlerroutine: Adresse nicht vorhanden
528C 215F53 00084 adrfail LD HL,errmsg ;Fehlermeldung
528F C06744 00085 CALL 4467h ;anzeigen
5292 B7 00086 OR A ;NZ-Flag setzen (Fehlerbedingung)
5293 C9 00087 RET
00088
00089 ;für "von-bis"-Ausdruck fortlaufende Sektoren laden
5294 C0D652 00090 found1 CALL loadsec ;nächsten Sektor laden
00091
00092 ;1. Anschrift gefunden: ausdrucken
5297 212453 00093 found2 LD HL,sender ;Absenderadresse
529A C06A44 00094 CALL 446ah ;ausdrucken
529D 210254 00095 LD HL,secbuff+2 ;Beginn der eigentlichen Empfängeradresse
52A0 7E 00096 ortloop LD A,(HL) ;Zeichen aus dem String
52A1 23 00097 INC HL ;auf nächstes Zeichen
52A2 FE24 00098 CP ' ' ;ESC in TSCRIPS-Codierung?
52A4 2002 00099 JR NZ,noesc ;falls nein
52A6 3E1B 00100 LD A,ESC ;sonst ASCII-ESC daraus machen
52A8 FE26 00101 noesc CP ' ' ;TSCRIPS-Flag für Hex-String?
52AA 2833 00102 JR Z,hexloop ;falls ja
00103
00104 ;normale Zeichen ausdrucken
52AC FE8D 00105 CP CR+80h ;Zeilenende? (TSCRIPS setzt Bit 7 bei CR)
52AE 200A 00106 JR NZ,nocr ;falls nein
52B0 7E 00107 LD A,(HL) ;Zeichen an der nächsten Stelle laden
52B1 FE3E 00108 CP ' ' ;Beginn einer Befehlszeile?
52B3 3E8D 00109 LD A,CR+80h ;Zeichen restaurieren
52B5 2003 00110 JR NZ,nocr ;falls nein
52B7 44 00111 LD B,H ;Länge der Suchstrecke groß genug
52B8 EDB1 00112 CPIR ;Zeiger hinter abschließenden CR stellen
52BA E67F 00113 nocr AND 7fh ;Bit 7 = 0 wg. TSCRIPS-Eigentümlichkeiten
52BC CDB405 00114 CALL 05b4h ;Zeichen ausdrucken
52BF 7C 00115 LD A,H ;MSB der Pufferadresse
52C0 FE54 00116 CP secbuff/100h ;Sektorende überschritten?
52C2 28DC 00117 JR Z,prtloop ;falls noch nicht
00118
00119 ;Adresse ist gedruckt; Walze auf nächsten Aufkleber stellen
52C4 215B53 00120 LD HL,nextlbl ;Zeilenschaltungen zum nächsten Aufkleber
52C7 C06A44 00121 CALL 446ah ;ausdrucken
52CA 2A0054 00122 LD HL,(secbuff) ;Namenskürzel dieses Sektors
52CD ED488153 00123 LD BC,(inbuff+3) ;2. Namenskürzel
52D1 ED42 00124 SBC HL,BC ;gleich? (2. Kürzel erreicht, Adr. gedr.)
52D3 C8 00125 RET Z ;fertig, falls ja
52D4 18BE 00126 JR found1 ;sonst weiter drucken
00127
00128 ;einen Sektor in den Puffer laden
52D6 210054 00129 loadsec LD HL,secbuff ;Sektorpuffer
52D9 118453 00130 LD DE,fcB ;FCB-Adresse
52DC C33644 00131 JP 4436h ;Sektor laden und zurück
00132
00133 ;Bearbeitung von Hex-Codes zwischen &-Zeichen
52DF CDF652 00134 hexloop CALL gethex ;1. Hex-Ziffer einlesen
52E2 07 00135 RLCA ;unteres ins obere Nibble rotieren
52E3 07 00136 RLCA
52E4 07 00137 RLCA
52E5 07 00138 RLCA
52E6 4F 00139 LD C,A ;puffern
52E7 CDF652 00140 CALL gethex ;2. Hex-Ziffer einlesen
52EA B1 00141 OR C ;jetzt eine Hex-Zahl komplett im Akku
52EB CDB405 00142 CALL 05b4h ;Zeichen ausdrucken
52EE 7E 00143 LD A,(HL) ;nächstes Zeichen

```


Copyright

- a) Hinweis für Raubkopierer („Kopier – aber richtig“).
b) Das Urheberrecht verhindert durch das Copyright, daß sich feige Programmierer in der Anonymität verstecken können.

CPU

Gerichtsstand des Rechners, in dem sämtliche Entscheidungen (meist zu Ungunsten des Anwenders) gefällt werden. Der Chip sieht aus wie ein Käfer mit 32 Beinen. Bewegt er sich, dann ist es sogar einer.

D

Datei

- a) Jede halbwegs locker zusammengehörende Anzahl von Bytes auf

einem Speichermedium, gleichgültig, ob sie überhaupt einen Zweck erfüllen, und welcher dies nun gerade sein könnte.

- b) Datenbankinhalt, jene auf immer und ewig verschwundenen, vor langer Zeit eingetippten Angaben über irgend etwas, was man nicht vergessen wollte.

Datenbus

Vielpolige, gelegentlich auch ganzvielpolige Leitung in der CPU zum Datentransport. Immer viel zu langsam und immer mit zu geringer Kapazität. Deshalb in Anlehnung an öffentliche Verkehrssysteme „Bus“ genannt.

Datenerfassung

Neue Volkssucht, die, ausgehend von öffentlichen Institutionen, auf alle Computerbesitzer übergriff. Datenerfasser, die mit Lust bei der Sache sind, haben in ihrer Datenbank nicht nur die Maße und Haarfarben aller Bunnies (Retusche-Häschen) des Jahrgangs 1967–1989, sondern können diese auch nach der metrisch spezifizierten Größe ordnen und ausdrucken (siehe auch Volkszählung).

Datenreise

Nächtliche DFÜ-Verbindung mit einer Datenbank. Für Hacker ist dies mit Rauschgiftkonsum gleichzusetzen. Der übermächtige, bleiche User sitzt mit erweiterten Pupillen und eingefrorenem Grinsen vor seinem Bildschirm. Gelegentlich wird die Datenreise zum Horrortrip, etwa weil das Passwort unüberwindlich ist. Um die Kosten ihrer Sucht, die Telefonrechnung, zu bezahlen, schrecken Hacker auch vor kriminallistischen Handlungen nicht zurück. Sie verfassen Handbücher oder Programme.

Defender

- a) Computerspiel, bei dem es darum geht, die Erde gegen angreifende Weltraummonster und glitschige Aliens zu verteidigen.
b) User, dessen Lebensinhalt darin besteht, den Computer gegen eifersüchtige Ehefrauen und neidische Kinder zu verteidigen.

Directory

Verzeichnis auf einer Diskette oder Festplatte. Dabei verhalten sich die Dateien oft wie Socken oder Unterhosen in einer Kommodenschublade: sie verschwinden auf Nimmerwiedersehen.

Direktzugriff

- a) Unangenehme, tätliche „Anmaché“ eines Vorstadt-Casanovas. Führt gelegentlich zur Ohrfeige.
b) Direkter Zugriff eines Programms auf Daten der Festplatte. Führt gelegentlich zum Headcrash (siehe dort).

Club 80
INFO 33
März 91

Seite
54

```
52EF FE26 00144 CP 'a' :abschließendes a-Zeichen?
52F1 2DEC 00145 JR NZ,hexloop :falls es in Hex weitergeht
52F3 23 00146 INC HL :a-Zeichen Überspringen
52F4 18AA 00147 JR ortloop :weiter drucken
00148
00149 :eine Hexziffer einlesen
52F6 7E 00150 gethex LD A,(HL) :Hex-Zeichen laden
52F7 23 00151 INC HL :auf nächstes Zeichen
52F8 FE41 00152 CP 'A' :höher als 0-9?
52FA 3602 00153 JR C,ciphok :falls nein
52FC D607 00154 SUB 07h :Hex an Dez angleichen
52FE E60F 00155 ciphok AND 0fh :ASCII -> binär
5300 C9 00156 RET
00157
00158 :verschiedene Codes, Texte und Puffer
0000 00159 NUL EQU 00h :null
0003 00160 ETX EQU 03h :end of text
0007 00161 BEL EQU 07h :beep
0000 00162 CR EQU 0dh :carriage return
001E 00163 ESC EQU 1bh :escape
5301 48 00164 getaddr DB 'Kürzel für die Adresse(n)? '.BEL.CR :Aufforderung
531E 18 00165 ortinit DB ESC,'S',ESC,'b',01h,ETX :Druckerinitialisierungen:
00166 :Nanezu-Reset, Proportionalischn.
5324 18 00167 sender DB ESC,'-',01h,ESC,'S',01h :Unterstreichungs, Subscript
532A 41 00168 DM 'Arnulf Sobb. Wakenitzstr. 3, D-2400 Lübeck ' :Absender
5356 18 00169 DB ESC,'T',ESC,'-',NUL :zurück auf Normalschrift
535B 18 00170 nexttbl DB ESC,'J',0fh,CR :15/180 Zoll Vorschub, CR
535F 41 00171 errmssg DM 'Adresse fehlt in ' :Fehlertext
5370 41 00172 filename DM 'ADRESSEN/LIB',BEL.CR :Filename
0006 00173 inbuff DS 06h :Puffer für Tastatureingaben
0020 00174 fcb DS 20h :Puffer für den FCB
5400 00175 secbuff EQU $+100h&0ff00h :Sektorpuffer an gerader Adresse
00176
5200 00177 END start
```

00000 Fehler

```
Club 80
INFO 33
März 91

BEL 0007 CR 0000 ESC 001B ETX 0003 NUL 0000 adrfail 528C
alladdr 525F ciphok 52FE disoerr 527F errmssg 535F fcb 5384 filename 5370
found1 5294 found2 5297 getaddr 5301 gethex 52F6 hexloop 52DF inbuff 537E
loadsec 5206 newname 521F nexttbl 535B noer 52BA noesc 52A8 notind 5279
oneaddr 525C ortaddr 5264 ortinit 531E ortloop 52A0 readsec 5267 secbuff 5400
seekerr 5282 sender 5324 start 5200
```


Liebe Computer-Freunde,

mit diesem kleinen Artikel habe ich endlich meine guten Vorschläge etwas zur Clubzeitschrift beizutragen in die Tat umgesetzt. Ich werde auch zukünftig (hoffentlich regelmäßig) für die Clubfreunde die ein Modell 4 oder 4P besitzen über PD-Software, "Neuigkeiten" usw. berichten. Da es für mich nur ein gutes Betriebssystem gibt (LSDOS 6.3) werden sich meine Tips und Patches in der Regel nur auf dieses Betriebssystem beschränken. Ein kleiner Hinweis: Alle Tips und Patches sind nicht auf meinem Mist gewachsen, sondern stammen in der Regel aus der Zeitschrift Computer News 80 die ich aus Amerika beziehe.

Selbstverständlich habe ich, soweit das möglich ist, alle Patches vorher getestet. Es wird sich aber nie ausschließen lassen, daß es zu Problemen kommen kann. Also Vorsicht bei allen Änderungen des Betriebssystems.

So nun soll es genug sein mit der langen Vorrede, jetzt endlich zu den versprochenen Informationen.

Im letzten Jahr habt ihr zusätzlich zum Clubinfo eine Aufstellung über PD-Software (Mod. 4) erhalten. Ein Teil dieser Programme ist jetzt vom Club gekauft worden, und steht den Mitgliedern auf Wunsch zur Verfügung. Diese Software wird von mir verwaltet und evtl. bei Zustimmung durch den Vorstand um weitere interessante Programme erweitert. Zur Zeit sind folgende Disketten im Bestand:

MOD4UTL Nr.: 3 - 5, 9 - 12, 14, 16 - 18, 20 - 31, 33 - 36,
38 - 40, 43 - 48, 53 - 54, 60

MOD4COM Nr.: 26

Eine kurze Beschreibung der Programme ist aus dem Verzeichnis ersichtlich. Ich habe auch vor einige Programme in der Clubzeitschrift vorzustellen, möchte aber jetzt bereits an jeden die Bitte richten selbst nach Erhalt der Software einen kleinen Erfahrungsbericht bei Jens abzuliefern.

Da die Programme vom Club gekauft wurden entstehen also nur Kosten Diskette/Postgebühr. Wenn mir Leerdisketten zugesandt werden dann bitte ordentlich verpackt.

Bei dieser Gelegenheit etwas zum aktuellen Stand der Technik beim Modell 4 hinsichtlich Speichererweiterungen. Die Firma Anitek (USA) bietet ihre MegaMem-Boards mit bis zu 8 MByte an. Es gibt verschiedene Boards von 1 bis 8 MByte (130 bis 200 US \$/ohne Speicherb.) die sich mit SIP Bausteinen bestücken lassen. Der Einbau soll angeblich ohne löten möglich sein. Das Board hat nach Angaben von Anitek eine Größe von

3.5 x 5.4 inch. Ein z.B. 6 MByte Board kann auch mit lediglich 1, 2 usw. MByte bestückt werden, so daß einem späteren "Vollausbau" nichts im Wege steht. Nachdem man das Board mit Speicherbausteinen bestückt hat muß lediglich die CPU entfernt werden, das Board in den Sockel der CPU und die CPU in den Sockel auf dem Board gesteckt werden. Hört sich doch wirklich gut an.

Da es bisher lediglich ein Programm gab (LeScript v. Anitek) welches Speichererweiterungen über 128 K nutzt, konnte man im Normalfall die Speichererweiterung lediglich als schnelle Ramdisk nutzen. Jetzt gibt es aber das Programm HYPERSVC, ebenfalls von Anitek, welches völlig neue Aspekte eröffnet. Dieses Programm patcht das Betriebssystem (LS/TRS-DOS 6.xx) beim Start in der Art und Weise, das auch andere Programme wie z.B. PRO-WAM, Double-Douty (neue Vers.), Little Brother Data Base, Busy Calc und Busy Calc Extra, Fast Term (Comm. Progr.), M-Ras Assembler, M-C Compiler und PRO-ENH Basic Compiler die Fähigkeit verleiht die Speichererweiterung über 128K zu nutzen. Das ganze soll rd. 20 US \$ kosten. Ob dieses Programm auch mit dem an die deutsche Tastatur angepassten Betriebssystem funktioniert, werde ich noch klären.

Anitek bezeichnet das ganze als "intelligent patch". Unabhängig vom benutzten Betriebssystem sucht sich das Programm den Bereich der geändert werden soll selbst.

Zum Schluß ein kleiner Tip für alle die in Basic programmieren. Statt des Befehles SYSTEM reicht es völlig aus das Ausrufungszeichen ! einzugeben. Hierzu ein Beispiel:

! "DIR 1" entspricht SYSTEM "DIR 1"

Da mir nicht genau bekannt ist wer alles ein Mod. 4/4P besitzt, bin ich auf eure Reaktionen sehr gespannt. Wenn ich etwas Resonanz finde werde ich euch in Kürze z.B. Informationen über mehrere wirklich gute "Patches" für LSDOS 6.3 und andere Programme liefern.

Jörg Brans

Hallo LeScript User,

bestimmt fahren viele von euch das Textverarbeitungsprogramm LeScript und haben auch einen Drucker angeschlossen (Wer schreibt schon fuer die Diskette?). Jedoch gibt es bei den verschiedenen Drucker Probleme beim Ausdruck. So habe ich einen kleinen (jedoch sehr guten) Drucker, den BROTHER M1009 an meinem IVp. Er druckte zwar alles, jedoch keine Steuerzeichen wie Breit, Tief, etc.!

Diesem kann durch Steuerzeichen im Text abgeholfen werden. Das einzigste Problem: er kann noch keine Umlaute (aber das bekomme ich auch noch hin).

Die nachfolgende Liste zeigt, welche Tasten fuer welche Funktion stehen.

Steht in der Tabelle CLEAR-SHIFT-1 so ist die CLEAR-Taste zusammen mit der SHIFT-Taste und der Taste 1 zu druecken. Es sollte dann ein halbes Dreieck erscheinen, was den CHR\$(27) darstellt. Nun koennen weitere Zeichen folgen.

Soll also unterstrichen gedruckt werden, so werden folgende Tasten gedrueckt.

CLEAR-SHIFT-1 dann
- (Minus) und dann noch
CLEAR-SHIFT-A

Alles klar?! So lassen sich alle Steuerzeichen auf dem BROTHER 1009 drucken. Auch Exoten wie den SEIKOSHA GP 550A habe ich auch das richtige Drucker beigebracht.

Viel Spass und wenig Papierverbrauch!

Frank

Steuerzeichen Brother M1009

Unterstreichen EIN CLEAR-SHIFT-1

-

CLEAR-SHIFT-A

Unterstreichen AUS CLEAR-SHIFT-1

-

CLEAR-SHIFT-@

Club 80
INFO 33
März 91

Breit EIN CLEAR-SHIFT-N
Breit AUS CLEAR-SHIFT-T

Seite
57

Schmal EIN CLEAR-SHIFT-O
Schmal AUS CLEAR-SHIFT-R

Hochgestellt

Tiefgestellt

Normal

Vergr. und Breit

Normal

Druckersteuerzeichen für den SEIKOSHA GP550A

Unter der Textverarbeitung LeScript kann der Drucker SEISKOSHA GP 550A mit den nachfolgenden Steuerzeichen angesteuert werden.

Möchte man z. B. die LQ ELITE einschalten, so drückt man die Tasten SHIFT CLEAR und die Taste 1 zusammen. Es erscheint dann ein Dreieck auf dem Schirm. Nun noch das Steuerzeichen Q eingeben und der Drucker druckt alles in LQ ELITE. Die Steuerzeichen können auch gemischt im Text verwendet werden.

Alle Zeichen beziehen sich auf die eingedeutschte Version.

Frank Schoof

Breitschrift EIN: SHIFT CLEAR 1 SHIFT CLEAR N
Breitschrift AUS: SHIFT CLEAR 1 SHIFT CLEAR Q

Standard PICA : SHIFT CLEAR 1 N
Standard ELITE : SHIFT CLEAR 1 E

Schmalschrift : SHIFT CLEAR 1 C

LQ PICA : SHIFT CLEAR 1 H
LQ ELITE : SHIFT CLEAR 1 Q

Italic kursiv : SHIFT CLEAR 1 B

Proportional : SHIFT CLEAR 1 P

CLEAR-SHIFT-1

S

CLEAR-SHIFT-@

CLEAR-SHIFT-1

S

CLEAR-SHIFT-A

CLEAR-SHIFT-1

T

Superscript : SHIFT CLEAR 1 U
Subscript : SHIFT CLEAR 1 O

Unterstreichen : SHIFT CLEAR 1 X
Ausschalten : SHIFT CLEAR 1 Y

CLEAR-SHIFT-1

W

CLEAR-SHIFT-A

CLEAR-SHIFT-1

W

CLEAR-SHIFT-@

Club 80
INFO 33
März 91

Seite
58

IC IC IC IC IC IC IC IC IC IC IC IC IC IC IC
IC IC IC IC IC IC IC IC IC IC IC IC IC IC IC
IC IC IC IC IC IC IC IC IC IC IC IC IC IC IC
Wer benötigt für seine Rechner GENIE I,
GENIE II, III, TRS-80 III, etc. noch
IC's ???

Habe noch über 40 IC's wie z. B. 4116, 4164,
280, EPROMS, ROMS, etc. sowie einen Floppy-
controller (GENIE I), 1 Floppy 40 DD/DOS
1 Centronics Interface GENIE I, 1 Minitoool,
1 HRG IB unbestückt sowie 2 Platinen zum
ausschlachten !!!

Anfragen bitte schriftlich an:
Frank Schoof
Elpkje 5
4890 Bielefeld 1

Suche Bilder
IBM od
Format dürfte bis hi
Hauptsächlich sch

HRG-fähiger, sonst aber
eher einfacher Matrix-
drucker mit Epson-kom-
patiblem Befehlssatz ge-
sucht. Angebote bitte an

Arnulf Sopp
Wakenitzstr. 8
2400 Lübeck 1
T. 0451-793054

Suche Bilder/Grafiken auf
IBM oder ATARI
Format dürfte bis hin zu VGA fast egal sein.
Hauptsächlich schwärme ich für farbiges.
Wer tauschen möchte:

Jens Neueder
0791/42877

TRX-280
The Next Generation of TRS-80
by Peter Ray (Anitek)

How would you like it if your TRS-80 had the expandability of an IBM XT, yet had more built-in features, ran faster, had more on-board memory, and all without having to make a single modification to your TRS-80 mother board? Sound fantastic? It is actually possible, and may soon even become a reality, with the proposed "Next Generation" replacement mother board, called the "TRX-280". This marvel of modern technology is already in the earlier stages of development as this article is being written.

Anitek Software Products has already begun developing a totally new replacement mother board for the TRS-80 that would empower your Model 4 and 4P to new levels of performance that were only before dreamed about by TRS-80 enthusiasts, like yourself. Our design criteria for the TRX-280 were simple. First, design a mother board the same size and dimensions of the TRS-80 mother board that would already have built into it most, if not all, the features that could, up until now, only be added to a TRS-80 by purchasing and installing a full compliment of the currently available modification kits. Second, design into the TRX-280 mother board many features that are not currently available to TRS-80 users even as modification kits - features that the original TRS-80 designers should have designed into it in the first place, but didn't - features like on-screen underlining, bold and italic which is a most needed feature in both the word processing and data processing environments. And third, even with all this new power, the TRX-280 had to be designed to continue to be totally TRS-80 compatible so that any programs that now run on a stock TRS-80 Model 4 or 3 will still run on the new TRX-280 mother board (except programs that use the cassette port).

The following is a list of the proposed features that Anitek believes can be designed into the TRX-280 replacement mother board, the size of the TRS-80 mother board, using the state-of-the-art technology that is available today.

1. A Z-280 microprocessor running at 10 MHz (which is equivalent to the speed of a Z-80 running at about 15 MHz).
2. On-board RAM expandable to 4 megabytes (possibly even 16 Meg).
3. Built-in bit-mapped graphics.
4. Built-in clock/calender.
5. Built-in hard drive host adapter.
6. On-screen text underlining.
7. On-screen text bolding.
8. On-screen italics.
9. On-screen expanded text.
10. Ability to display IBM characters, TRS-80 characters, block graphics, bit-mapped graphics, inverse video, bold text, underlined text, italics, expanded characters and normal size characters all on the screen at the same time.
11. A floppy disk controller that is capable of handling 180K, 360K, 720K, 1.2 Meg, and 1.4 Meg floppy drives.
12. Ability to boot directly from a hard drive without the need to first insert a floppy.
13. Ability to boot Model 3 DOSs directly on a 4P without MODELA/III.
14. One, or possibly even two, expansion slots, compatible with the IBM PC 8-bit expansion card architecture.

The on-board expansion slots would make it possible for the user to further expand his system with inexpensive half-size expansion cards currently available for the IBM PC. Some examples might include: hand scanners, modem cards, color monitor cards, mouse cards, and hard drive interface cards, just to name a few.

Sound too good to be true? Admittedly, it sounds fantastic, but it is absolutely possible! Much of the conceptual design work has already been done. In fact, about one year from today it could even be possible for you to purchase one of these new TRX-280 mother boards and have all of these features in your TRS-80.

Parenthetically let me state that even though we are very sure that we could design and build a multi-featured mother board the size and dimensions of the TRS-80 mother board, and even though we are very sure we could design a mother board with all the above features, it is still a little early yet to know for sure whether or not we can do both - i.e. make the mother board the size of TRS-80 mother board and at the

same time give it every one of the new features mentioned above. The physical size of the mother board may impose a limit on the number of new features that we could design into it. However the TRX-280, if produced, will be the size of the TRS-80 mother board (that is a must) and it will have most, if not all, the above-mentioned features - all of the most important features for sure.

Yet, before Anitek Software Products can move from the safe and cozy arena of conceptual design on the TRX-280 project into the financially risky arena of actually developing and manufacturing such an awesome product, it will be necessary to first do a some careful market analysis to determine just how successful a product like the TRX-280 mother board will be in the TRS-80 marketplace. We already know that a mother board could be designed that could have most, if not all, of the features mentioned above, plus possibly even some additional features that were not mentioned. What we don't know is, are there enough people like yourselves who own TRS-80s that would be interested in buying the new TRX-280 mother board to justify to us the huge investment of time and money that it would take to develop and produce it. This is where you can help us out.

We would like to see what kind of interest a proposed product of this type will generate in the TRS-80 marketplace. If the TRX-280 is to become a reality, we will need to have this information early on, before we begin full-scale development. If the TRX-280 mother board were available today, would you buy it for your computer? Why or why not? If it were available a year from now (which is about the earliest it could be ready) would you buy it? Do the features sound interesting to you? Do you feel that some of the proposed features are of more value to you than other features? If so, which features do you rate the highest and which the lowest? Does the fact that you could have virtually every modification known for the TRS-80 built right into the mother board as standard equipment interest you? Would having a TRS-80 that you could plug off-the-self IBM compatible expansion card into interest you? Does a TRS-80 that could out perform an IBM XT interest you?

What other features have you thought would be nice to have built into a TRS-80 that are not mentioned in the above list?

We want your responses. If you have questions or comments regarding the TRX-280 mother board project, we want to hear from you. But we need to have you write your letter to CN 80. The editors have very graciously and enthusiastically agreed let us use this publication as an open forum for sharing ideas and questions regarding the TRX-280. By you sharing your thoughts with the magazine, you will also be giving the other readers the opportunity to see what you are thinking about the "Next Generation", TRX-280 concept. And at the same time, you will have the advantage of seeing what others are saying about this fantastic idea, as well. If any of the letters submitted to CN 80 need our attention for commenting on, the editors will make sure Anitek sees a copy of the letter ahead of time so that our comments can be published along with your letter.

If you are feeling like you can't live another day without one of these TRX-280 mother boards for your computer, or if you just think the concept sounds interesting, please let me strongly encourage you to, in the next day or two, write it down on a piece of paper and send it in to CN 80. And when you read the comments from other readers in next month's issue, if you have anything other ideas you would like to voice, write those down, too, and send them in. Your positive interest and enthusiasm shared in this publication is what it may take to generate in the other readers a strong interest as well. Excitement breeds excitement. Enthusiasm breeds enthusiasm. This way we can get the fires of interest in this "Next Generation" of TRS-80s burning faster and hotter. No comment or question is too small. If you have something you wish to voice about the TRX-280 before we begin development, please send it CN 80. It's going to take all of us to turn the TRX-280 from merely a great possibility into a great reality.

In anticipation of some of the questions that you might already have, I have answered what I feel may be the more obvious questions. Keep in mind though that no one can predict the future with absolute certainty. The answers given below are

based on possibilities and expectations of future events.

Q. How much will the new TRX-280 mother board cost?

A. Probably somewhere between \$700 to \$900 depending on the number of boards that we feel will be purchased. We will have a better idea after we have had the chance to measure the market interest and to tally the advanced orders before we go into the final design stages of production. The price could even be considerably less if a good number of people commit ahead of time to the purchase of the TRX-280 board by placing advanced orders. We are also looking into the possibility of offering an extra discount as an incentive to those who are able to place advanced orders, probably in the form of granting a discount equal to 1/2 the purchaser's advanced down payment. This could bring the price down to as little as \$500 - \$600 for those who are able to put that much down on an advanced order. Under this proposed plan a customer making an advanced payment of 2/3 of the cost of a TRX-280 mother board would get the last 1/3 free. This would be a fantastic way to substantially lower the cost of owning a TRX-280 mother board. But even at the expected normal price of \$700 to \$900, this is really still quite a bargain when you consider that it would cost over twice this amount to buy even half these features as individual modification kits, not to mention the days or weeks it would take you to install each one. By the way, if advanced orders are received with advanced payment but the number of which is not large even to justify the final development and manufacturing of the TRX-280, then the advanced payments would be returned.

Q. Will my other TRS-80 programs still work on the new TRX-280 mother board?

A. Yes. This is our number one most important design objective. We will do everything in our power to ensure that all TRS-80 Model 3/4 software (except programs that access the cassette port) will work as well or better on the new replacement mother board as they do on a stock TRS-80. It is a little early yet to know whether or not any DOS device drivers might require patches to work at the faster CPU speed. If there is any clever

way to design around this in hardware, we will.

Q. Will the bit-mapped graphic circuitry of the TRX-280 be compatible with the Grafyx Solution board.

A. Yes. It is our number two most important design objective to make sure that all your Grafyx Solution programs work as well on the TRX-280 as they do on the TRS-80 mother board.

Q. How soon will the new TRX-280 mother board be ready to buy?

A. Once we begin full-scale development we expect it will only take about 12 months. The sooner we know that the TRX-280 is a product that that TRS-80 marketplace will buy, the sooner we will be able to start the full-scale development phase, and the sooner it will be available for you to purchase.

Q. Will I have to put a heavier power supply in my computer?

A. At this point we feel that it won't be necessary to change the power supply as long as it is in good condition and is not one of the earlier supplies that were small and under powered. Our objective is to design the board in such a way that it will work with the same power supply that you now have.

Q. Can I trade in my old mother board and get a credit toward the purchase of a TRX-280 mother board?

A. We may offer some sort of credit if you wish to trade in your old mother board, but it may only be for about \$25 to \$50. The fact is that the technology will be so much more advanced on the new mother board that there will be only a few components on the old TRS-80 mother board that will be usable on the new TRX-280 boards.

Q. Can I put one of the new TRX-280 mother boards in my Model 3.

A. Yes. Since the Model 3 has basically the same case and mother board housing as the non-gate-array Model 4, as far as we can tell you would be able to put a TRX-280 mother board in a Model 3.

Q. Will technical information be published to explain how to take advantage of all

the new features?

A. Yes. We want you to get the most out of the product as possible, so we will publish what all the port addresses are and what they do, and the memory maps and all hardware technical info we can think of to benefit those who want that type of information.

Q. Will the TRX-280 mother board be available for the Model 4 first, or for the 4P?

A. That all depends on which one of the two versions we get the most advanced orders for before we begin final development.

Q. Will the new mother board be able to run MS-DOS programs?

A. No. We don't feel that there is any need for us (or anyone else) to come up with another IBM clone. If we thought you wanted a computer that could run MS-DOS programs we would have thought that you would have bought an MS-DOS computer. The TRX-280 mother board is for people who love their TRS-80s and don't care to switch over to the MS-DOS world. Yet, at the same time, they see what the MS-DOS machines can do and they long for a way that they could have the same power and performance on their TRS-80.

If you have any other questions, and I am sure you do, please send them into CN 80 as soon as possible so they can get into the next issue.

In closing, let me stress that the outcome of the TRX-280 replacement mother board project is now in the hands of you, the readers. I know you have a great attachment to your TRS-80, or you wouldn't be reading this magazine. We at Anitek also share in this love for what was and is still one of the greatest little computers ever made. But our strong appreciation for the TRS-80 only makes us more keenly aware of its several designed-in limitations and makes us long for the TRS-80 that could be, the TRS-80 that is now possible with the advent of the TRX-280 replacement mother board. But the bottom line is this, we know that we can design and build the TRX-280, but whether this product ever sees the light of day is something that you now have the

opportunity to decide. Are you ready to usher-in the "Next Generation" of power, speed, features, capabilities, and expandability - or are you satisfied with the current generation of TRS-80 just the way it is? Cast your vote of support for the TRX-280 by writing a letter to this publication. And if you are already sold on the idea and would like to place an advanced order to lock-in a lower price for yourself, let us know that too, so we can send you the details on how to place an advanced order. But again, please do so through CN 80 so that all readers can see the degree of interest this new product is generating. I am eager to hear your votes, answer your questions, respond to your comments, and discuss more of the aspects of the "Next Generation" with you again next month.

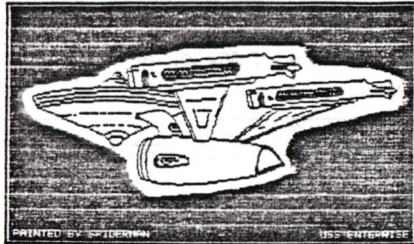
-Peter Ray (Anitek)

MOUSE ON YOUR MODEL 4 (Without an Interface) Part II by David Goben

I've received numerous complaints that I did not follow my first mouse article with the follow-up in the very next issue, or the one after that. Actually, I had only said that I would cover some things in my next article on the mouse, and did not promise to cover it immediately. Finally, I apologize for the technical depth of this article, but due to space constraints, I cannot expect to be able to use the entire issue of CN80 to explain the intricate working of a mouse in a more comprehensible format.

The idea of communicating with a serial mouse may have seemed complicated, because no one seemed to think it could be done. It was probably just one of those things where one simply assumes that such a task to be incredibly complex. Fortunately, Scott McBurney did not think so, and blazed the trail toward using a mouse on a TRS-80, thus sparking my own underlying interest. My contribution to this effort was to improve on the basic driver design, simplify the interfacing process, give it more sophisticated commands, and allow it to work on both a 2-button and a 3-button

Hallo Clubfreunde,



heute gehe ich auf Entdeckungsreise in der Club-Bibliothek. Jeder kennt sicherlich noch den Spruch: "Unendliche Weiten (und weit und breit kein Spock zu sehen) ...". In Anlehnung hieran gilt wohl ähnliches für unsere emsigen Club-Bücher-Leser. Unendliche Mengen an Büchern, nur keiner will sie lesen. Bei mir stehen sie - wie wohl auch zuvor bei den anderen Bibliothekaren - zwar warm und trocken, trotzdem wäre es wesentlich angebrachter ihnen die Weiten Deutschlands und die Wohnorte der Club 80er zu zeigen. Scheinbar wähnt jeder sein eigenes Regal viel interessanter als das was hier so steht. Weit gefehlt liebe Leute! Hier könnt ihr noch stöbern und dabei so manchen lukrativen "Hecht" an Land ziehen. Bereits beim ersten Griff ins Gedränge werde ich fündig:

Etliche Bände zum TRSDOS, TRS-80 und Innenleben ...

1. **BASIC FASTER AND BETTER & OTHER MYSTERIES**
A guided tour of BASIC programming tricks and techniques.
2. **TRS-80 DISK & andere Geheimnisse**
Ratgeber zur Datenwiederfindung
3. **TRSDOS 2.3 DECODED & OTHER MYSTERIES**
Commented Source and guide to TRSDOS 2.3 for the Model I
4. **BASIC DISK I/O FASTER AND BETTER & OTHER MYSTERIES**

TRS-80 Random and Sequential File Programming: Beginner, Intermediate, Advanced

5. **THE CUSTOM TRS-80 & OTHER MYSTERIES**
The Complete guide to Customizing TRS-80 Software and Hardware

... eine gebündelte Ladung an echt guten Dingen. Nun mal nicht gleich abgewunken !!! Das etwas gealterte TRSDOS ist zwar inzwischen der letzte Schrei, aber wir sind doch wohl immer noch ein Club der Z80-Freunde, ODER?! Na also.

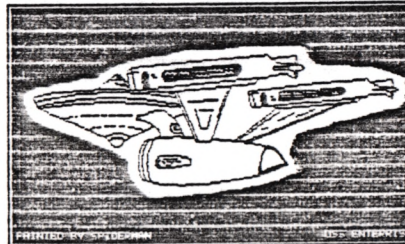
Gut! Kommen wir zum Inhalt. Am interessantesten - und wer hat schließlich schon problemlos Zugriff auf das Listing eines Betriebssystems - scheint mir der Band von Punkt 3 zu sein. Hier kann man nach aller Herzenslust in den SYS-Modulen des TRSDOS blättern und sich über die Funktionsweise unterrichten lassen oder auch nur nach Tips, Tricks und guten Ideen ausschau halten.

Wem das nicht so zusagt, kann sich auch den Punkt 1 näher ansehen und sich dort einige Inspirationen zum Thema BASIC abholen. Der Band enthält eine ganze Reihe von interessanten Utilities zur BASIC-Erweiterung oder um sich das debuggen von Basic-Programmen zu erleichtern. Lohnenswert allemal ...

Ein Problem will ich jedoch nicht verschweigen, etwas Englisch sollte man beherrschen. Oder man gibt sich mit den Listings zufrieden, denn was dort steht ist ohne Untertreibung auch schon "sein Geld wert".

-----<#>-----

Hallo Clubfreunde,



es ist ein Neuzugang in unsere Bibliothek zu verzeichnen. Oliver Volz hat es mir geschickt. Wie ich feststellen konnte, handelt es sich um die englischsprachige Ausgabe der bereits in der Clubbibliothek vorhandenen deutschen Version. Beim Vergleich beider Bücher fiel mir auf, daß in der englischen Version das Inhaltsverzeichnis deutlich detaillierter und besser gegliedert ist (!). Vom Inhalt her scheint aber in beiden das Gleiche zu stehen.

Beim zwanglosen Durchblättern fand ich in der Rubrik "4. Betriebssysteme" eine Kommentierung der Generationsfolge der TRS-80 Systemsoftware. Angefangen mit TRSDOS 2.1 ... 2.2. NEWDOS 2.2 ... 2.2. VTOS etc. Sogar ein Ausblick auf zukünftiges in dieser Richtung ist zu finden (NEWDOS 80 wird als heiser Tip gehandelt und seine tollen neuen Fähigkeiten angerissen!). Leute!, die Geschichte gibt mehr her, als man der Zukunft nachsagt. Auf jeden Fall ist klar zu übersehen, was man verpaßt hat.

So weit, so gut. Gehen wir wieder einmal auf Entdeckungsfahrt. Beim Stöbern bin ich auf zwei Bücher gestoßen, deren Inhalt sich recht interessant liest. Die Buchtitel lauten:

1. **Listing-Verwaltung in MBASIC**
aus dem W.-D. Luther-Verlag
2. **Mikrorechner in der Ansteuertechnik**
aus dem Militärverlag der DDR (VEB)

Buch 1:
Es wird in allen Einzelheiten ein Verwaltungsprogramm zum Erfassen von Buch-, Zeitschriftentiteln und einzelnen Artikeln entwickelt. Erfassbare Daten sind:

Zeitungstitel, Titel, Ausgabe, Seite, Seitenumfang, Serien, Stichwörter, Rechnerotypen, Programmiersprachen, Textsprachen und Programmarten.

Das ganze ist in MBASIC für einen CP/M-Rechner geschrieben. Leichte Anpassungen in Bezug auf das Zusammensetzen der File- und Laufwerksbezeichnungen zum vollen Dateinamen sind entsprechend dem eingesetzten Betriebssystem notwendig. Das Programm scheint so gut parametrisiert zu sein, daß Änderungen nur an wenigen Stellen zu tätigen sind um zur gewünschten Anpassung zu kommen. Zentrale Drehscheibe des Gesamtprogramms ist ein Menüprogramm namens "START". Von hier aus werden die einzelnen Funktionen per RUN gestartet. Es könne also u.U. Teile einfach ausgelassen werden, bzw. die Eingabe des Programmsystems kann übersichtlich in handlichen Schritten erfolgen. Wer also eine umfangreiche Loseblattsammlung an Artikeln aller Art sein Eigen nennt, hat hier die Möglichkeit, ein für allemal Ordnung zu schaffen. Das Buch hat einen Umfang von ca. 110 Seiten und ist zu gut 80% Listing. Es sind 8 Module mit durchschnittlich 500...600 Zeilen Umfang abzutippen. Etwas Geduld sollte also vorhanden sein. Aber es lohnt sich (wer wagt es ???).

Buch 2:
Wer sich für Hardware in Verbindung mit Rechneransteuerung interessiert, findet hier eine Menge Anregungen zum Experimentieren. Vom Autor werden rechneransteuerbare Sinusgeneratoren, Netzgeräte D/A und A/D-Wandler, Schrittmotoransteuerungen, Frequenzzähler etc. besprochen. Die zur Ansteuerung erforderlichen Routinen sind überwiegend in Z80-Assembler verfaßt. Es handelt sich aber nicht um vollständige Programme, sondern nur um die Treiber (!). Die Applikation muß der Anwender selbst schreiben. Etwas gewöhnungsbedürftig sind die Schaltbilder. Hier wird man mit dem DDR-System konfrontiert. Es ist aber entzifferbar. Mit den IC-Bezeichnungen ist es auch so ein Ding. Die Bezeichnungen sind nicht zu gebrauchen. Hier muß aus der Funktion und dem Text auf das entsprechende TTL-IC geschlossen werden! Da die Schaltungen aber meist von geringem Umfang sind (4...8 TTL-IC's bzw. eine Handvoll analoger Bauteile wie OP's und Transistoren), ist es kein Problem. Am Ende des Buches wird auch noch einmal auf unterschiedliche Verfahren zur Meßwertglättung eingegangen (stärken & schwächen) und eine Assembleroutine dazu vorgestellt. Wer häufiger Meßprobleme hat und eine "schnelle Lösung" sucht, sollte hier einen Blick hineinwerfen.

-----<#>-----

Impressum

<u>1. Vorsitzender</u>	Fritz Chwolka Saarstraße 4 5173 Aldenhoven Tel.: 0 24 64/89 20
<u>2. Vorsitzender</u>	Gerald Schröder Am Schützenplatz 14 2105 Seevetal 1 Tel.: 0 41 05/26 02
<u>Hardwarekoordinator</u>	Andreas Magnus Pommernstraße 4 4650 Gelsenkirchen Tel.: 02 09/87 02 30
<u>Newdos-Diskothekar</u>	Oliver Volz Waldburgstraße 73 7000 Stuttgart 80 Tel.: 07 11/7 35 38 17
<u>CP/M-Diskothekar</u>	Rüdiger Sörensen Thomas Mann Straße 3a 6500 Mainz 1 Tel.: 0 61 31/3 28 60
<u>Clubbücherei</u>	Kurt Müller Sophie-Scholl-Ring 38 2054 Geesthacht Tel.: 0 41 52/7 06 43
<u>Redaktion</u>	Jens Neuder Rudolf-Then Straße 32 7178 Gschlachtenbretzingen Tel.: 07 91/4 28 77
<u>Autoren</u>	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. 8870 Günzburg
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,

wie schon kurz berichtet, haben wir uns nun mit der INFO-Produktion selbstständig gemacht. Layout und Vervielfältigung der INFO's erfolgt nun aus einer Hand. Der große Vorteil des clubeigenen Kopierers ist der Zeitgewinn für die Fertigstellung des INFO's. So habe ich jetzt die Möglichkeit auch nebenher die eine oder andere Seite vorzubereiten. Auch die Ergebnisse, die wir mit unserem clubeigenen Gerät erzielen, sind doch recht annehmbar.

Leider hatte ich nicht genügend Material, um in allen Sparten unseres Club-INFO's etwas zu bringen. Ich hoffe aber, das dies bis zum nächsten INFO der Fall sein wird. Zumindest werde ich mich, für meinen Teil, zukünftig bemühen, das jeweils fällige INFO ca 3/4 Wochen nach dem Redaktionsschluß fertig zu haben. Der nächste Redaktions-Schluß ist, nochmals zur Erinnerung, Ende Mai.

Als Beilage gibt es diesmal ein Sonder-Heft für die Modell-4-User:

Portbelegung und BOOT-ROM-Dokumentation des M IV

Abschließend hoffe ich, daß Euch unsere neueste Club-80-Paperware zusagt und Ihr viel Spaß daran - und Anregungen darin findet.

Übrigens:

Vergesst nicht, Euch zum Clubtreffen, vom 09.05.91 bis 12.05.91 in Weilburg, anzumelden!

Das in der Anmeldung "**Ich reise am April gegen Uhr an.**" war natürlich ein kleiner Scherz zum 1. dieses Monates. Es heißt natürlich "**Mai**"! Sicher haben dies alle bemerkt. Eine Anmeldung kann aber selbstverständlich trotzdem mit diesem Blatt erfolgen.