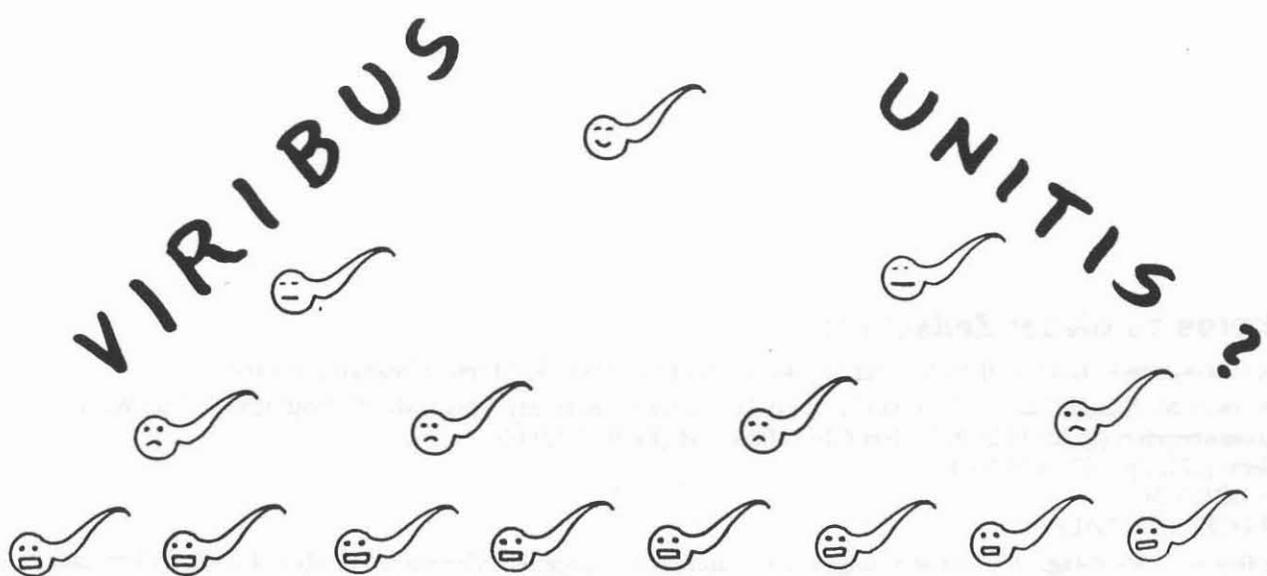


PC-NEWS

Das offizielle Mitteilungsblatt
des
PCC-TGM

(Personal Computer Club - Technologisches Gewerbe-
Museum)



Inhalt

Beitragsteil	1
<u>Software-Entwicklungen</u>	1
TURBO-Basic mehr als ein Compiler	1
Telefonverzeichnis	7
Word-Druckertreiber für den STAR LC-10Color	7
<u>Software-Beschreibungen</u>	10
Intelligence-Compiler - Expert System Shell	10
Ein kurzer Erfahrungsbericht über den Umgang mit "TeX"	10
Kurz-Anleitung zum Arbeiten mit "muMATH-83" + Übungs- und Demo-Texte	15
Programmieren mit muSIMP	16
<u>Tips und Tricks</u>	23
Tips und Tricks zur Herculeskarte	22
Resetfeste RAM-DISK	24
Fehler im Phoenix ROM BIOS Ver. 2.27 ?	24
FRAMEWORK II mit QUADRAM 386 XT	24
Listschutz ade!	25
Computerviredn	25
<u>Clubteil</u> : die Clubmitteilungen sind wegen des geringen Umfangs diesmal Bestandteil des Beitragsteils !	
Wichtig ! Clubabende für April und Juni ! Wichtig !	28
Fragen mit und ohne Antworten	28
TGM-Opus (2:310/1.0)	28

Näheres zu dieser Zeitschrift:

Bezugsbedingungen: Einzelheft öS 50,-, für Mitglieder des PCC-TGM im Mitgliedsbeitrag enthalten.

Impressum: Medieninhaber: PCC-TGM (Personal-Computer-Club-TGM), Wexstraße 21, Postfach 59, 1202 Wien.

Anrufbeantworter: (0222)/35 23 983 (Herr GINTHÖR): Mo,Fr: 9.00-12.00h

Mailbox: (0222)/604 92 48 (8-N-1)

BTX: 912213028

TELEBOX: RA2 FIALA.

Grundlegende Richtung: Auf Anwendungen im Unterricht bezogene Informationen über Personal-Computer-Systeme. Berichte über Veranstaltungen des Vereins. Beratung der Vereinsmitglieder gemäß den Statuten des PCC-TGM.

Layout und Satz: Walter Riemer, Rosengasse 9, 2102 Bisamberg

Druck: CA (Creditanstalt Bankverein), Wallensteinplatz, 1200 Wien

Erscheinungsort: Wien

Redaktion und für den Inhalt verantwortlich: Franz Fiala, Siccardsburggasse 4/1/22, 1100 Wien.

TURBO BASIC mehr als ein Compiler

TGM_96: ZGEN.TXT, TGM_97: ZGEN.ARC
 Rober Smola, #486

Wenn man TB (Turbo Basic) nur zum Compilieren von GW-Basic Programmen benützt, nutzt man nur einen geringen Teil der Möglichkeiten, die dieses Programm bietet. Deshalb möchte ich Ihnen TB etwas näher bringen.

Als erstes möchte ich den hervorragenden Editor nennen, der schon fast eine Textverarbeitung darstellt, wenn man einmal davon absieht, daß keine Drucker-Steuerung und Textvormatierung möglich ist. Der Compiler wurde in Assembler geschrieben und ist dementsprechend schnell bei der Übersetzung. So schnell, daß er ein Arbeiten fast wie bei einem Interpreter ermöglicht. Das heißt, in der Testphase kann man sein Programm jederzeit ändern und neu compilieren, ohne umständliches laden und längeren Wartezeiten. Fehler werden nicht nur bei der Übersetzung angezeigt, auch während des Programmablaufes auftretende Fehler werden automatisch im Quelltext angezeigt. Für die Fehlerkorrektur muß nicht einmal in den Editor gesprungen werden, denn man befindet sich automatisch in diesem. Das erzeugte Programm ist gegenüber GW-Basic rund um den Faktor fünf schneller. Unter gewissen Voraussetzungen sind Geschwindigkeitssteigerungen bis zum Zehnfachen möglich. Aber Geschwindigkeit und Komfort ist nicht alles, was TB zu bieten hat.

TB ist auch eine eigene Programmiersprache, die gegenüber GW-Basic einige Vorteile bietet.

Es kann in TB zur Gänze auf Zeilennummern verzichtet werden, Sprungziele und Unterprogramme werden mit Namen (Label) aufgerufen. Unterprogramme können, unabhängig vom übrigen Programm, mit lokalen Variablen geschrieben werden. Diese werden dann mit CALL und Übergabe von Parametern aufgerufen. Überhaupt lehnt sich TB bei Prozeduren, Schleifen und Unterprogrammen stark an Pascal an, aber ohne der üblichen Nachteile.

Selbstverständlich ist man mit TB an keine 64K Grenze gebunden. Es stehen für Variablen und Strings je 64K Speicher zu Verfügung, wobei einzelne Strings bis 32k lang sein können. Datenfelder (Arrays) können je 64k belegen und der Rest des Speichers steht dem Programm zu Verfügung.

Dies sind aber noch nicht alle Vorteile, die TB bietet, deshalb möchte ich Ihnen ein kleines Programm vorstellen, damit sie sich selbst von den Möglichkeiten überzeugen können.

ZGEN ist ein Zeichengenerator für die Erstellung beliebiger Zeichen für den Drucker. Erstellt wurde das Programm für den Seikosha SL-80AI, es sollte aber auch für andere 24-Nadeldrucker geeignet sein. Für andere Drucker müßte das Programm erst umgeschrieben werden, wobei ich Ihnen aber gerne behilflich bin.

Am Ende gebe ich Ihnen noch eine kleine Funktionsbeschreibung des Programmes, das dem Club auch auf Diskette und als EXE File zu Verfügung steht.

Jeder gesetzte Punkt am Bildschirm entspricht einem Nadelanschlag im 24-Nadel-180 TPI Grafikmodus.

```

*****
**** ZEICHEN GENERATOR FÜR 24 NADELDRUCKER (SEIKOSHA SL-80 AI 180 TPI) ****
**** V 1.1 vom 19.10.88 von Robert Smola in Turbo Basic ****
*****
COLOR 7,0
KEY OFF
CLS
PRINT "
PRINT " ZEICHEN - GENERATOR FÜR 24 NADELDRUCKER (SEIKOSHA SL-80 AI 180 TPI)
PRINT : PRINT
PRINT " Dieser Zeichen-Generator hilft bei der Erstellung eigener Zeichen am
PRINT " Drucker, die beim IBM Zeichensatz nicht enthalten oder nicht dar-
PRINT " stellbar sind.
PRINT " Das Zeichen wird in Großformat am Bildschirm erstellt und kann an-
PRINT " schließend mit den dazugehörenden Datenbytes ausgedruckt werden.
PRINT " Diese Datenbytes können dann in Programmen eingebunden werden, oder
PRINT " auch in Texte, wenn Ihr Textverarbeitungsprogramm direkte Drucker-
PRINT " steuerung zuläßt.
PRINT " Der Drucker gibt Ihr Zeichen mit folgender Befehlssequenz aus:
PRINT " 27;42;39;18;0;... Hier müssen nun die 54 Bytes Ihres Zeichens folgen.
PRINT " Sie können aber auch die Bytes auf Disk abspeichern und später in
PRINT " Ihr Programm laden oder direkt mit dem 'type' Befehl ausdrucken.
PRINT " Z.b. TYPE Name.ZGN>PRN (Anschließend Zeilenvorschub senden )
PRINT " Der Steuerkode für den Drucker wird dabei automatisch abgespeichert.
PRINT " Das Zeichen als solches kann aber auch auf Disk gespeichert und zur
PRINT " späteren Bearbeitung wieder geladen werden.
PRINT : PRINT
COLOR 31,0
PRINT "
CALL WARTEN
'Bildschirm löschen
NEU:
COLOR 7,0
CLS
'Integer Zuweisung aller nicht deklarierten Variablen
DEFINT A-Z

```

```
'Raster erzeugen
  FOR I = 1 TO 24
    LOCATE I,1
    PRINT "HHHHHHHHHHHHHHHHHHHHHH";I;
  NEXT I
  LOCATE 25,1
  PRINT "1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 ";

'Text ausgeben
  LOCATE 1,45
  PRINT "Auto Curser "; : COLOR 0,7 : PRINT "EIN";
  COLOR 7,0 : PRINT " oder AUS mit .... C";

  LOCATE 3,45
  PRINT "Punkt setzen oder löschen mit ENTER";
  LOCATE 5,45
  PRINT "Bewegung mit ..... CURSER Tasten";
  LOCATE 7,45
  PRINT "Zeichen invertieren mit ..... I";
  LOCATE 9,45
  PRINT "Zeichen ausdrucken mit ..... Z";
  LOCATE 11,45
  PRINT "Bytes ausdrucken mit ..... B";
  LOCATE 13,45
  PRINT "Bytes auf Disk speichern mit .... S";
  LOCATE 15,45
  PRINT "Zeichen abspeichern mit ..... A";
  LOCATE 17,45
  PRINT "Zeichen laden mit ..... L";
  LOCATE 19,45
  PRINT "Programm beenden mit ..... END";
  LOCATE 21,45
  PRINT "Alles löschen mit ..... HOME";

'Variablen Zuweisung
  ZEILE = 4 : SPALTE = 5
  ALT$ = "H"
  ALTZEILE = ZEILE : ALTSPALTE = SPALTE
  RICHTUNG = 2
  DIM WERTE (54)
  AUTO = 1
  SPEICHER = 1
```

```
'Cursor setzen
SET:
  LOCATE ALTZEILE,ALTSPALTE,0
  COLOR 7,0
  PRINT ALT$;

SETX:
  IF SCREEN (ZEILE,SPALTE) = 195 AND SCREEN (ZEILE,SPALTE,1) > 1 THEN
    ALT$ = "H"
    LOCATE ZEILE,SPALTE,0
    COLOR 7,0
    PRINT "H";
  ELSE
    ALT$ = "■"
    LOCATE ZEILE,SPALTE,0
    COLOR 0,7
    PRINT "H";
  END IF
  ALTZEILE = ZEILE : ALTSPALTE = SPALTE

'Tastaturabfrage
IN:
  IN$ = INKEY$ : IF IN$ = "" THEN IN
    IF IN$ = CHR$(13) THEN SETZEN
    IF IN$ = CHR$(67) OR IN$ = CHR$(99) THEN AUTO
    IF IN$ = CHR$(83) OR IN$ = CHR$(115) THEN SPEICHERN
    IF IN$ = CHR$(66) OR IN$ = CHR$(98) THEN BYTES
    IF IN$ = CHR$(90) OR IN$ = CHR$(122) THEN ZEICHEN
    IF IN$ = CHR$(73) OR IN$ = CHR$(105) THEN INVERT
    IF IN$ = CHR$(65) OR IN$ = CHR$(97) THEN ABSPEICHERN
    IF IN$ = CHR$(76) OR IN$ = CHR$(108) THEN LADEN
    IF LEFT$(IN$,1) <> CHR$(0) THEN BEEP : GOTO IN
  IN$ = RIGHT$(IN$,1)
  IF IN$ = CHR$(75) THEN SPALTEDECR
  IF IN$ = CHR$(77) THEN SPALTEINCR
  IF IN$ = CHR$(72) THEN ZEILEDECR
  IF IN$ = CHR$(80) THEN ZEILEINCR
  IF IN$ = CHR$(79) THEN COLOR 7,0 : CLS : END
  IF IN$ = CHR$(71) THEN NEU
  BEEP
GOTO IN
```

```
'Spalte links
SPALTEDECR:
    IF SPALTE > 1 THEN SPALTE = SPALTE - 2 ELSE BEEP
    RICHTUNG = 1
    GOTO SET

'Spalte rechts
SPALTEINCR:
    IF SPALTE < 35 THEN SPALTE = SPALTE + 2 ELSE BEEP
    RICHTUNG = 2
    GOTO SET

'Zeile hinauf
ZEILEDECR:
    IF ZEILE > 1 THEN DECR ZEILE ELSE BEEP
    RICHTUNG = 3
    GOTO SET

'Zeile hinunter
ZEILEINCR:
    IF ZEILE < 24 THEN INCR ZEILE ELSE BEEP
    RICHTUNG = 4
    GOTO SET

'Punkt setzen
SETZEN:
    SPEICHER = 1
    IF SCREEN (ZEILE,SPALTE) = 204 THEN ALTS$ = "■" ELSE ALTS$ = "H"
    IF AUTO = 0 THEN SET
    ON RICHTUNG GOTO SPALTEDECR, SPALTEINCR, ZEILEDECR, ZEILEINCR

'Zeichen invertieren
INVERT:
    SPEICHER = 1
    LOCATE ALTZEILE,ALTSPALTE
    COLOR 7,0
    PRINT ALTS$;
    IF ALTS$ = "H" THEN ALTS$ = "■" ELSE ALTS$ = "H"
    FOR I = 1 TO 24
        FOR J = 1 TO 35 STEP 2
            LOCATE I,J
            IF SCREEN (I,J) = 195 THEN PRINT "■"; ELSE PRINT "H";
        NEXT J
    NEXT I
    GOTO SET
```

```
'Automatischer Curser an/aus
AUTO:
    IF AUTO = 0 THEN
        AUTO = 1 : LOCATE 1,57 : COLOR 0,7 : PRINT "EIN";
        COLOR 7,0 : PRINT " oder AUS";
    ELSE
        AUTO = 0 : LOCATE 1,57 : COLOR 7,0 : PRINT "EIN oder ";
        COLOR 0,7 : PRINT "AUS";
    END IF
    GOTO IN

'Zeichen drucken
ZEICHEN:
    ON ERROR GOTO DRUCKERFEHLER
    CALL ZEICHEN
    LPRINT CHR$(27);CHR$(42);CHR$(39);CHR$(18);CHR$(0);
    FOR I = 0 TO 53
        LPRINT CHR$(WERTE(I));
    NEXT I
    LPRINT : LPRINT : LPRINT : LPRINT
    ON ERROR GOTO
    GOTO SET

'Bytes drucken
BYTES:
    ON ERROR GOTO DRUCKERFEHLER
    CALL ZEICHEN
    WIDTH "LPT1:", 72
    FOR I = 0 TO 53
        IF WERTE(I) < 10 THEN LPRINT "00"; FNBYTES$;
        IF WERTE(I) >9 AND WERTE(I) < 100 THEN LPRINT "0"; FNBYTES$;
        IF WERTE(I) >99 THEN LPRINT FNBYTES$;
    NEXT I
    LPRINT : LPRINT : LPRINT : LPRINT
    ON ERROR GOTO
    GOTO SET
```

```
'Abspeichern auf Disk
SPEICHERN:
  CALL ZEICHEN
  LOCATE 23,43
  COLOR 0,7
  PRINT "Bitte Filenamen eingeben";
  COLOR 7,0
  PRINT "      .ZGN";
  LOCATE 23,68,1,11,12
  CALL IN
  IF FILE$ = "" THEN PRINT FNEDE$ : GOTO SET
  FILE$ = FILE$ + ".ZGN"
  LOCATE 23,43
  PRINT " ";
  COLOR 0,7
  PRINT "An bestehendes File anhängen ? J/N";
  COLOR 7,0
  PRINT " ";
  ON ERROR GOTO DISKFEHLER

INST:
  WHILE NOT INSTAT : WEND
  A$ = INKEY$
  IF A$ = CHR$(74) OR A$ = CHR$(106) THEN _
  OPEN FILE$ FOR APPEND AS #1 : GOTO AUS
  IF A$ = CHR$(78) OR A$ = CHR$(110) THEN _
  OPEN FILE$ FOR OUTPUT AS #1 : GOTO AUS
  BEEP : GOTO INST

AUS:
  PRINT#1, CHR$(27);CHR$(42);CHR$(39);CHR$(18);CHR$(0);
  FOR I = 0 TO 53
  PRINT#1, CHR$(WERTE(I));
  NEXT I
  CLOSE#1
  ON ERROR GOTO
  PRINT FNEDE$
  GOTO SET
```

```
'Zeichen abspeichern
ABSPEICHERN:
```

```
  LOCATE ALTZEILE,ALTSPALTE
  COLOR 7,0
  PRINT ALT$;
  LOCATE 23,43
  COLOR 0,7
  PRINT "Bitte Filenamen eingeben";
  COLOR 7,0
  PRINT "      .ZEI";
  LOCATE 23,68,1,11,12
  CALL IN
  IF FILE$ = "" THEN PRINT FNEDE$ : GOTO SET
  FILE$ = FILE$ + ".ZEI"
  ON ERROR GOTO DISKFEHLER
  OPEN FILE$ FOR OUTPUT AS #1
  FOR I = 1 TO 24
    FOR J = 1 TO 36
      A$ = CHR$(SCREEN(I,J))
      WRITE#1, A$
    NEXT J
  NEXT I
  CLOSE#1
  ON ERROR GOTO
  PRINT FNEDE$
  GOTO SET
```

```
'Zeichen laden
```

```
LADEN:
  SPEICHER = 1
  LOCATE 23,43
  COLOR 0,7
  PRINT "Bitte Filenamen eingeben";
  COLOR 7,0
  PRINT "      .ZEI";
  LOCATE 23,68,1,11,12
  CALL IN
  IF FILE$ = "" THEN PRINT FNEDE$ : GOTO SET
```

```

FILES$ = FILES$ + ".ZEI"
ON ERROR GOTO DISKFEHLER
OPEN FILES$ FOR INPUT AS #1
  FOR I = 1 TO 24
  FOR J = 1 TO 36
  LOCATE I,J
  INPUT#1, AS$
  PRINT AS$;
  NEXT J
NEXT I
CLOSE#1
ON ERROR GOTO
PRINT FNEDE$
GOTO SETX
'Drucker Fehler abfangen
DRUCKERFEHLER:
  BEEP
  LOCATE 23,43
  COLOR 31,0
  PRINT " Drucker  FEHLER      Taste drücken";
  CALL WARTEN
  LOCATE 23,43
  PRINT FNEDE$
  RESUME SET
'Disk Fehler abfangen
DISKFEHLER:
  BEEP
  LOCATE 23,43
  COLOR 31,0
  PRINT " Disk      FEHLER      Taste drücken";
  CALL WARTEN
  LOCATE 23,43
  PRINT FNEDE$
  RESUME SET

```

```

'Prozedur zur Zeichen Berechnung
SUB ZEICHEN
  LOCAL Z, I, K, WERT
  SHARED WERTE (), ALTZEILE, ALTSPALTE, ALT$, SPEICHER
  IF SPEICHER = 0 THEN EXIT SUB
  SPEICHER = 0
  LOCATE ALTZEILE,ALTSPALTE
  COLOR 7,0
  PRINT ALT$;
  Z = 0
  FOR I = 1 TO 35 STEP 2
  WERT = 0
    FOR K = 1 TO 8
    IF SCREEN (K,I) = 219 THEN GOSUB WERT
    NEXT K
  WERTE (Z) = WERT : INCR Z
  WERT = 0
    FOR K = 9 TO 16
    IF SCREEN (K,I) = 219 THEN GOSUB WERT
    NEXT K
  WERTE (Z) = WERT : INCR Z
  WERT = 0
    FOR K = 17 TO 24
    IF SCREEN (K,I) = 219 THEN GOSUB WERT
    NEXT K
  WERTE (Z) = WERT : INCR Z
  NEXT I
  EXIT SUB
'Unterprogramm zur Ermittlung eines Bytes
WERT:
IF K = 1 OR K = 9 OR K = 17 THEN WERT = WERT + 128
IF K = 2 OR K = 10 OR K = 18 THEN WERT = WERT + 64
IF K = 3 OR K = 11 OR K = 19 THEN WERT = WERT + 32
IF K = 4 OR K = 12 OR K = 20 THEN WERT = WERT + 16
IF K = 5 OR K = 13 OR K = 21 THEN WERT = WERT + 8
IF K = 6 OR K = 14 OR K = 22 THEN WERT = WERT + 4
IF K = 7 OR K = 15 OR K = 23 THEN WERT = WERT + 2
IF K = 8 OR K = 16 OR K = 24 THEN WERT = WERT + 1
RETURN
END SUB

```

```
'Prozedur Warten auf Taste
SUB WARTEN
  LOCAL A$
  WHILE NOT INSTAT : WEND : A$ = INKEY$
END SUB
'Funktion Bytes drucken
DEF FNBYTES$
  SHARED WERETE()
  FNBYTES$ = (RIGHT$(STR$(WERTE(I)),LEN (STR$(WERTE(I)))-1) + " ")
END DEF
'Prozedur Filename schreiben
SUB IN
  LOCAL A$, Z
  SHARED FILE$
INS:
  A$ = INKEY$ : IF A$ = "" THEN INS
  IF A$ = CHR$(8) THEN DEL
  IF A$ = CHR$(13) THEN SCHREIBE
  IF A$ = CHR$(45) THEN SETS
  IF A$ > CHR$(47) AND A$ < CHR$(58) THEN SETS
  IF A$ > CHR$(64) AND A$ < CHR$(91) THEN SETS
  IF A$ = CHR$(95) THEN SETS
  IF A$ > CHR$(96) AND A$ < CHR$(123) THEN SETS
  IF A$ = CHR$(129) OR A$ = CHR$(154) THEN SETS
  IF A$ = CHR$(132) OR A$ = CHR$(142) THEN SETS
  IF A$ = CHR$(148) OR A$ = CHR$(153) THEN SETS
  IF A$ = CHR$(225) THEN SETS
  BEEP : GOTO INS
SETS:
  IF POS(0) = 76 THEN BEEP : GOTO INS
  PRINT A$;
  GOTO INS
DEL:
  IF POS(0) = 68 THEN BEEP : GOTO INS
  LOCATE 23,POS(0)-1
  PRINT " ";
  LOCATE 23,POS(0)-1,1,11,12
  GOTO INS
```

```
SCHREIBE:
  FILE$ = ""
  IF POS(0) = 68 THEN EXIT SUB
  Z = 68
  DO UNTIL SCREEN(23,Z) = 32 OR SCREEN(24,Z) = 46
  FILE$ = FILE$ + CHR$(SCREEN(23,Z))
  INCR Z
  LOOP
  END SUB
'Funktion löschen
DEF FNENDE$
  LOCATE 23,43
  PRINT " ";
  END DEF
```

```
'*** ENDE ***
```

```
*****
```

C...Auto Cursor ein oder aus	: Automatisches Weiterrücken des Cursors ein,- oder ausschalten.
I...Zeichen invertieren	: Alle nicht gesetzten Punkte werden gesetzt, alle gesetzten gelöscht.
Z...Zeichen ausdrucken	: Das Zeichen wird am Drucker ausgegeben und anschließend werden vier Zeilenvorschübe gesendet.
B...Bytes ausdrucken	: Die Datenbytes des Zeichens werden ausgegeben und anschließend werden vier Zeilenvorschübe gesendet.
S...Bytes auf Disk speichern	: Die Steuerkodens für den Drucker und die Datenbytes des Zeichens werden auf das aktuelle Laufwerk gespeichert. Dabei besteht die Möglichkeit, an bestehende Files anzuhängen und somit ganze Zeichenketten zu erstellen.
A...Zeichen abspeichern	: Der Bildschirmteil, der das Zeichen darstellt, wird auf das aktuelle Laufwerk gespeichert.
L...Zeichen laden	: Ladet das Zeichen, das mit A gespeichert wurde.
END...Programm beenden	: Das Programm wird beendet.
HOME...Alles löschen	: Das Programm wird neu gestartet.

Untenstehend gebe ich Ihnen noch eine kleine Funktionsbeschreibung des Programms, das dem Club auch auf Diskette und als EXE-File zur Verfügung steht.
 Jeder gesetzte Punkt am Bildschirm entspricht einem Nadelanschlag im 24-Nadel-180 TPI Grafikmodus.

Telefonverzeichnis

TGM_96: TELEFON.TXT, TGM_97: TELEFON.ARC
Dipl.-Ing. Günther NEUBAUER

Anm.d.Red.: Dieses Programm ist, wie viele andere in den PC-NEWS, nicht eigens für die Veröffentlichung geschrieben worden. Das Programm wurde vom Autor zur Verwaltung seiner Telefonnummern geschrieben. Es dient uns hier als ein weiteres Beispiel für die Anwendung von DBASE. Die Redaktion wurde gebeten, bei der Weitergabe an Dritte das Einverständnis des Autors einzuholen. Wenn Sie das Programm für kommerzielle Nutzung einsetzen wollen, wenden Sie sich bitte an den Autor unter der Rufnummer 515 22/492. Es ist gerne bereit, Änderungen gegen einen Unkostenbeitrag durchzuführen. Damit wir das Programm dennoch weitergeben können, finden Sie es in kompilierter Form auf der Diskette.

Erklärungen zum Programm

Beim Autor sind die Programmteile auf der Diskette im Laufwerk A und das Programm dbase auf der Harddisk C. Es erscheint am Schirm

C>

Danach schreiben Sie bitte "cd dbase"

Es erscheint C\dbase

Schreiben Sie "dbase"

Es erscheint ein '.' am Bildschirm

Geben Sie spätestens jetzt die Diskette mit Ihren Daten ins Laufwerk A (oberes LW)

Schreiben Sie "set default to a:"

Es erscheint wieder ein '.' am Bildschirm.

Schreiben Sie "do menu"

Es erscheint folgendes Bild am Schirm:

Telefonverzeichnis

- (0) Verlassen des Telefonverzeichnisses
- (1) Hinzufügen neuer Namen
- (2) Drucken oder Auflisten des Telefonverzeichnisses
- (3) Suchen eines Namens
- (4) Ausbessern des Telefonverzeichnisses
- (5) Löschen eines Namens (Feld)

Geben Sie Ihre Wahl ein (0 - 5)

Nun können Sie mit der Eingabe der Ziffer 0 - 5 Ihren gewünschten Menüpunkt auswählen. Wollen Sie die Arbeit beenden, drücken Sie bitte "0" und anschließend "quit". Es wäre günstig, vor Verlassen des Arbeitsplatzes eine Sicherungskopie anzulegen. Drücken Sie "a:"

Es erscheint A > am Schirm

Legen Sie die Sicherungsdiskette in LW b:, verriegeln es, und geben nun ein "copy tel-bbd b:" ein.

Somit werden die Daten auf die Sicherungsdisk kopiert.

Bei eventuell auftretenden Problemen, Fragen oder Verbesserungen wenden Sie sich bitte an die BBD-Wien, Abt. 406, Hr. Neubauer unter der Rufnummer 515 22 Klappe 492.

Gutes Gelingen !

Word-Druckertreiber für den STAR LC-10colour

TGM_96: LC10.TXT, TGM_97: LC10.ARC

Harald Ludwig, N85b, #146

Untenstehend finden Sie den ganzen Beitrag als Original-Ausdruck mit dem beschriebenen Treiber:

Steuerzeichen

Damit eines der folgenden Steuerzeichen zum Drucker gesendet wird, muß man nur den entsprechenden Buchstaben in den Text einfügen und diesen Buchstaben mit der Schriftart "Steuerzeichen" formatieren. Der so formatierte Buchstabe wird dann nicht am Drucker ausgegeben, sondern es wird der entsprechende Steuercode zum Drucker gesendet.

Buchstabe	Steuercode	Wirkung
< @	^H (Text)	Backspace (letzte(s) Zeichen können noch einmal überdrückt werden) Vermerk
A und a	ESC ^L	Reverse Paper Feed (geht an den Beginn der Seite zurück)

D	ESC w1	Doppelt hohe Zeichen EIN
d	ESC w0	Doppelt hohe Zeichen AUS
E	ESC E	Emphasized (Schattendruck) EIN
e	ESC F	Emphasized (Schattendruck) AUS
K	ESC G	Double Strike (Fettdruck) EIN
k	ESC H	Double Strike (Fettdruck) AUS
L und l	ESC ^J	Reverse Line Feed (macht einen Papierrücktransport um eine Zeile)
P	ESC 9	Paper out detector (Papierendesensor) EIN
p	ESC 8	Paper out detector (Papierendesensor) AUS
Q	ESC h ^B	Quad (vierfach) hohe Zeichen EIN
q	ESC h ^@	Quad (vierfach) hohe Zeichen AUS
W und w	ESC ^G	Glocke (Druckersummer) kurz einschalten
Z	ESC ~1	Slash Zero (durchgestrichene Null) EIN
z	ESC ~0	normal Zero (normale Null) EIN
_	ASCII 32	Blank (für geschützte Wortzwischenräume)
b	ESC r ^B	Farbumschaltung blue (blau)
B	ESC r ^G	Farbumschaltung green (grün)
c	ESC r ^@	Farbumschaltung noire (schwarz)
C	ESC r ^O	Farbumschaltung orange (orange)
o	ESC r ^C	Farbumschaltung red (rot)
O	ESC r ^D	Farbumschaltung violet (violett)
v	ESC r ^C	Farbumschaltung violet (violett)
V	ESC r ^D	Farbumschaltung yellow (gelb)

Neben diesen Steuerzeichen können auch noch die Klammerbefehle verwendet werden. Diese Klammerbefehle sollten auch in der Schriftart "Steuerzeichen" formatiert sein, da sonst der Blocksatz nicht korrekt durchgeführt wird.

Klammerbefehl

Wirkung

((O))	Automatische Papierzufuhr AUS
((4))	Automatische Papierzufuhr EIN
((B))n	Schattendruck
((C))n	Ausdruckfarbe
((F))n	Schriftart
((I))	Kursivdruck
((R))	Automatischer Blattauswurf
((S))n	Zeichngröße
((T))n	Neuen Blattanfang setzen bei automatischem Einzelblatteinzug

Für alle anderen, nicht oben angeführten Zeichen wird der Code ASCII 0 gesendet.
Die Breite aller Zeichen im Steuerzeichensatz ist Null (wichtig für den korrekten Blocksatz), bis auf die Zeichen "BLANK", "@" und "_".

WICHTIGE Änderung: Das Zeichenattribut 'Durchgestrichen' wird beim Ausdrucken in 'Überstreichen' umgewandelt.

Schriftart

Beispiel

Draft_____Pica	ABCabc123
Courier_____Pica	ABCabc123
Sanserif_____Pica	ABCabc123
OratorCAP_Pica	ABCABC123
OratorLOW_Pica	ABCabc123

Draft Elite
Courier Elite
Sanserif Elite
OratorCAP Elite
OratorLOW Elite

ABCabc123
ABCabc123
ABCabc123
ABCABC123
ABCabc123

PS_Draft Pica
PS_Courier Pica
PS_Sanserif Pica
PS_OratorCAP Pica
PS_OratorLOW Pica

ABCabc123
ABCabc123
ABCabc123
ABCABC123
ABCabc123

PS_Draft Elite
PS_Courier Elite
PS_Sanserif Elite
PS_OratorCAP Elite
PS_OratorLOW Elite

ABCabc123
ABCabc123
ABCabc123
ABCabc123
ABCabc123

Double_Draft Pica
Double_Courier Pica
Double_Sanserif Pica
Double_OratorCAP Pica
Double_OratorLOW Pica

ABCabc123
ABCabc123 ABCabc123
ABCabc123 ABCABC123
ABCabc123

Double_Draft Elite
Double_Courier Elite
Double_Sanserif Elite
Double_OratorCAP Elite
Double_OratorLOW Elite

ABCabc123
ABCabc123 ABCabc123
ABCabc123 ABCABC123
ABCabc123

Double_PS_Draft Pica
Double_PS_Courier Pica
Double_PS_Sanserif Pica
Double_PS_OratorCAP Pica
Double_PS_OratorLOW Pica

ABCabc123
ABCabc123 ABCabc123
ABCabc123 ABCABC123
ABCabc123

Double_PS_Draft Elite
Double_PS_Courier Elite
Double_PS_Sanserif Elite
Double_PS_OratorCAP Elite
Double_PS_OratorLOW Elite

ABCabc123
ABCabc123 ABCabc123
ABCabc123 ABCABC123
ABCabc123

Quad_Draft

ABab12

Quad_Courier

ABab12

Quad_Sanserif

ABab12

Quad_OratorCAP

ABAB12

Quad_OratorLOW

ABab12

Quad_PS_Draft

ABab12

Quad_PS_Courier

ABab12

Quad_PS_Sanserif

ABab12

Quad_PS_OratorCAP

ABAB12

Quad_PS_OratorLOW

ABab12

Intelligence Compiler - Expert System Shell

TGM_96: IC.TXT
D.I. Peter KRIEG

Hersteller: IntelligenceWare Inc. 1986

Version: 1.0b

Verteilung: 5 Stück 360kB Disketten (nicht kopiergeschützt) (System 1, System 2, Editor, Tutorial, Examples

Installation auf einer Harddisk:

Create eines Subdirectories "c:\ic" auf der Harddisk und Überspielen des auf "System 1" befindlichen HINSTALL.BAT Batch-Files. Anschliessend dieses Batch-File aufrufen und die Disketten der geforderten Reihe nach einlegen. Die "Examples" werden nicht überspielt. Inklusive Tutorial (aber ohne Examples) benötigt "Intelligence Compiler" auf der Harddisk mindestens 950kB Platz. Der minimale Hauptspeicherbedarf wurde nicht ermittelt.

Charakteristika

"Intelligence Compiler" ist eine Expert System Shell, das heisst es stellt eine Entwicklungsumgebung zum Erstellen von Expertensystemen zur Verfügung. Dazu zählen:

- Menügeführter Editor, der sprachsensitiv die für "Intelligence Compiler" notwendigen Daten- und Sprachstrukturen aufbauen hilft (Wissenserwerbskomponente).
- Inference Engine (Schlussfolgerungsmaschine) ermöglicht Forward Chaining (datengetriebenes Schlussfolgern) sowie Backward Chaining (zielgetriebenes Schlussfolgern) gemischt.
- Erklärungskomponente, die während der Konsultation einer Expertensystemanwendung über das WARUM und nach Abschluss der Konsultation über das WIE Auskunft gibt.
 - WARUM ... warum eine bestimmte Frage gestellt wurde
 - WIE ... wie ein bestimmtes Ergebnis zustandekam (Backtracing)
- Dialogkomponente, die, sofern benötigt, den Benutzer automatisch nach Daten (Fakten) fragt.

Das Wissen wird in Form von IF-THEN Regeln in einer Wissensbasis abgespeichert. Auf der IF-Seite der Regeln werden jene Bedingungen formuliert, die erfüllt sein müssen, um die THEN-Seite der Regel auszuführen. Fakten (Daten) werden in Form von Frames abgespeichert. Als Besonderheit muss die Möglichkeit des "inexact reasoning" hervorgehoben werden (sowohl Regeln wie auch Fakten können Wahrscheinlichkeiten mitgegeben werden, mit deren Hilfe ein sogenanntes inexaktes Schlussfolgern möglich wird).

Das Expertensystem wird "kompiliert", das heisst in einen interpretierbaren Zwischencode umgewandelt, und arbeitet somit schneller als Systeme die auf LISP oder PROLOG aufbauen. Wieviele Regeln maximal in eine Expertensystemanwendung eingebunden werden können und ob eine Aufteilung auf mehrere Regelgruppen möglich ist, konnte nicht festgestellt werden. Ebenso unbekannt sind die Hardwareerfordernisse.

Zum Abschluss sei noch erwähnt, dass ein Tutorial eine kurze Einführung in die Welt der Expertensysteme wie auch in die Besonderheiten und Prinzipien von INTELLIGENCE gibt. Eine Reihe von Beispielen demonstriert die

Wirkungsweise von INTELLIGENCE bei unterschiedlichen Anwendungen. Ein Handbuch wird dadurch aber

Ein kurzer Erfahrungsbericht über den Umgang mit "TeX"

TGM_96: TEX.TXT
Fridbert Widder, #326

Dr. Fridbert Widder, Inst.für Theoret.Physik, Universitätsplatz 5, A-8010 Graz

Im Mathematik- und Physik-Unterricht und auch beim Veröffentlichenden von Forschungsergebnissen braucht man immer wieder Text-Systeme, die etwas mehr können, als nur ASCII-Zeichen aufs Papier zu bringen. Ich hatte deshalb mit "Scientex" und mit "ChiWriter" gearbeitet und war mit beiden eigentlich recht zufrieden gewesen: man sieht jedenfalls sofort beim Editieren am Bildschirm, wie der Text, und vor allem, wie die Formeln aussehen. Zwar hatte ich auch immer wieder von einem "sagenhaft komplizierten" System namens "TeX" - sprich: "tech" - gehört, war aber durch dessen Ruf, daß es unheimlich schwierig zu erlernen wäre derart abgeschreckt, daß ich es nicht einmal anschauen wollte.

Dann wurde mir jedoch heuer im Frühjahr die Aufgabe übertragen, für den Springer-Verlag einen Konferenzbericht herauszugeben - und dabei sollten die Manuskripte gleich in "TeX"-Dateien geschrieben und als solche dem Verlag geschickt werden. Also schaute ich mir einmal das "TeX-book" von Donald E. Knuth an, gab das Lesen darin aber bald auf: es ist zwar recht lustig geschrieben, aber unheimlich langatmig! Wenn man so beginnt, verliert man leicht die Freude (und den Überblick). Glücklicherweise gibt es von unserem Rechenzentrum eine 32-seitige Kurzanleitung: wenn man diese gelesen hat, kann man tatsächlich schon recht erfolgreich "ans Werk" gehen - und in der Praxis lernt man den Umgang mit "TeX" erfreulich einfach und rasch! - Eine wertvolle Hilfe, vor allem wenn man weitergehende "Feinheiten" des Systems ausnutzen will, stellt das (deutsch geschriebene) Buch aus dem Addison-Wesley Verlag dar: Norbert Schwarz "Einführung in TeX"; nach 170 einführenden Text-Seiten (mit zahlreichen praktischen Beispielen) findet man im Anhang - auf fast 100 Seiten - eine sehr übersichtliche und enorm hilfreiche Liste (und Kurzbeschreibung) der "TeX"-Befehle. - Damit ist man höchstwahrscheinlich jedem Problem gewachsen.

Was ist "TeX"?

Dieses von D.E.Knuth an der Stanford University entwickelte Programm wird am zutreffendsten als Textsatzsystem bezeichnet; es eignet sich zum Gestalten beliebiger Dokumente - seine besonderen Vorteile bringt es jedoch bei mathematisch-physikalischen Formeln zur Geltung. Die derzeit letzte Version 2.0 vom April 1986 gilt als weitgehend fehlerfrei - die von Knuth ausgesetzte Belohnung für den ersten, der darin einen Fehler findet, hat noch niemand in Anspruch nehmen können. Die "TeX-Source", das eigentliche Programm-Paket, ist "Freeware" und frei kopierbar - lediglich die Anpassungen an diverse Computersysteme, die von verschiedenen Firmen vorgenommen wurden, sind nicht kostenlos erhältlich.

Mir steht beispielsweise die "ST-TeX" Version 2.1 (von der Firma Tools) für den Atari ST zur Verfügung, sowie das von Artaker vertriebene PC-TeX (Version 1.50f). Die Unterschiede zwischen den verschiedenen Implementationen

sind geringfügig - sie liegen hauptsächlich im "Komfort" und bei den Ausgabe-Treiberprogrammen. - Auch das ein gewaltiges Plus für "TeX": Eine auf irgendeiner Maschine erstellte TeX-Datei ist mit jeder anderen zu verarbeiten!

Was ist "TeX" nicht?

Es ist (leider?) kein komfortables Textverarbeitungssystem nach dem WYSIWYG-Prinzip, es hat weder einen integrierten Editor noch eine direkte Drucker-Schnittstelle... Daher wohl ist es als so kompliziert in Verruf gekommen; zu Unrecht, denn die schon erwähnten Implementationen für PC's machen diese kleinen Nachteile ohnehin (nahezu) wett.

Wie erstellt man ein Dokument?

Der Weg bis zum fertigen Ausdruck besteht im wesentlichen aus drei Schritten:

1. Mit einem beliebigen Text-Editor, der eine gewöhnliche ASCII-Datei zu erzeugen gestattet, schreibt man den Text. Dabei sind nur ein paar Regeln zu beachten, etwa, daß das Ende eines Absatzes durch eine Leerzeile (oder durch den Befehl "\par" - die meisten TeX-Befehle erkennt man an dem vorangestellten "backslash" -) definiert wird. Griechische Buchstaben nennt man einfach beim Namen: \Delta liefert das große, \delta das kleine Delta - das war jetzt nicht ganz richtig: um diese Sonderzeichen und mathematische Symbole wie Integral, Summenzeichen, ... usw. zu bekommen, muß man vom Text- in den Formelzeichen-Modus umschalten; im fortlaufenden Text setzt man die Sonderzeichen zwischen einfache Dollarzeichen - freistehende, vom Text abgesetzte Formeln setzt man zwischen je zwei Dollarzeichen. - Klingt komplizierter als es ist - Beispiele sollen das zeigen (- siehe weiter unten)!
2. Die erstellte ASCII-Datei wird nun mit Hilfe des TeX-Programmes übersetzt: Wenn das System dabei Fehler vorfindet, meldet es das - eine Liste aller Fehler wird in den "LOG"-File geschrieben, die übersetzte Datei wird mit der Endung "DVI" (= DeVice Independent!) abgespeichert. Wenn (zu viele, "fatale") Fehler aufgetreten sind, muß man zunächst die ASCII-Datei korrigieren und noch einmal übersetzen. Sind keine oder nur unwesentliche Fehler vorgekommen (- das System versucht, nach "bestem Wissen" Fehler selbst auszubessern, aber das klappt nicht immer zufriedenstellend! -), dann kann man sich im dritten Schritt das fertige Produkt ansehen oder gleich ausdrucken lassen:
- 3.a. "Preview" auf dem Bildschirm - dazu (und zum Drucken) benötigt man die in den kommerziellen Implementationen mitgelieferten Treiberprogramme.
- 3.b. Ausdruck. Es gibt meist Druckertreiber für 9- und 24-Nadel-Drucker, für Laserdrucker und sogar für Lichtsatzmaschinen. - Schriftcharakter, Größe der Zeichen, das ganze Dokument- und Seiten-Layout, all das ist völlig druckerunabhängig! Nur werden die Zeichen von hochauflösenden Druckern eben feiner und präziser aufs Papier gebracht.

Wie das hier beschrieben ist: so einfach ist es auch in Wirklichkeit! Seinen ersten "TeX"-Text hat man, sofern die Implementation schon fertig installiert ist, in wenigen Minuten auf dem Bildschirm bzw. auf dem Papier. Doch dann wird man "neugierig", möchte wissen, was "TeX" sonst noch alles kann: welche Schriftarten, Sonderzeichen, Layout-Befehle, ... Dann muß man eine (der erwähnten) Beschreibung(en) zurate ziehen ... und einfach viel probieren!

Von den vorgegebenen Seitenformaten, Schriftenfamilien usw. abweichende Wünsche schreibt man vor den Beginn des ASCII-Textes. So lieferte uns beispielsweise der Springer-Verlag alle seine Format-Wünsche in Form einer ASCII-Datei, die sämtliche Formatierungs- und Layoutbefehle beinhaltet; sie heißt SPRINGER.CMM. Und wir brauchen nur noch vor den jeweiligen Text den Befehl "\input springer.cmm" zu schreiben. Seiten- und Fußnoten-Numerierung, Zeilen- und Seiten-Umbruch, Einfügen von Leerraum für Abbildungen an möglichst geeigneter Stelle... alles das besorgt "TeX" (aufgrund dieser Befehls-Liste am Beginn des Dokumentes) nach den Erfordernissen und Wünschen des Verlages.

Bevor ich nun am Beispiel einer beliebig herausgegriffenen Seite aus einem Dokument zeige, wie die ASCII-Datei und wie das (auf einem NEC P5) ausgedruckte Ergebnis aussehen, gebe ich noch eine Liste der Vorteile von "TeX" gegenüber "richtigen" Textverarbeitungssystemen an.

- 1.a. Der Formelmodus gestattet es, auch komplizierte mathematische Ausdrücke einfach und "sauber" zu setzen: wie gedruckt!
- 1.b. Akzente, Tilden, Querstriche etc. sind über (oder auch in) jedes Zeichen (hinein) zu setzen: das dürfte für manche fremdsprachlichen Texte von Bedeutung sein.
2. Sehr vielfältige und komfortable Möglichkeiten, Tabellen zu setzen.
3. Wahlweise englische oder deutsche Trennhilfe - natürlich auch abschaltbar.
4. Inhaltsverzeichnis und Stichwortlisten können automatisch erzeugt werden.
5. Fußnoten-Verwaltung.
6. Leicht selbst zu programmierende automatische Kapitelnumerierung, ...
7. ... und das scheint mir das Wesentlichste: die "Portierbarkeit", d.h. Unabhängigkeit vom Computertyp, und die Fehlerfreiheit!

Alle meine Erfahrungen und Aussagen beziehen sich übrigens auf die "Urfassung", auf das sogenannte "plain-TeX". Mittlerweile gibt es schon viele Erweiterungen wie z.B. "LaTeX": hier wird noch mehr Komfort geboten - einige der Dinge, die man mit plain-TeX noch programmieren muß, sind da schon vorgefertigt. Man verliert aber dadurch auch den großen Vorteil der universellen Austauschbarkeit. Und ein bißchen "Programmieren" macht ja manchmal auch Spaß: So habe ich mir, einem Beispiel von N.Schwarz folgend, die Befehle für ein zweiseitiges Seitenlayout zusammengestellt: funktioniert hervorragend - ganz unabhängig von Art oder Größe der verwendeten Schriften! Die "normale" Schrift ist eine 10-Punkt "roman" (Proportional-schrift). Die Größe ist von 5-Punkt bis 17-Punkt (und in manchen Implementationen noch größer) wählbar. Diese Schrift kann weiters geneigt ("slanted"), "italic" (= kursiv und etwas abgerundet) und fett "angefordert" werden. Dann gibt es noch sehr schöne "sans serif"-Schriftenfamilien, die überreichen Sonderzeichen-Fonts und, und, und ...! Man kommt kaum aus dem Schwärmen heraus, wenn man einmal mit "TeX" angefangen hat.

Und es ist in der Tat so einfach zu handhaben, daß ich bereits ganz gewöhnliche Briefe "in TeX" zu schreiben beginne; dazu ist das ST-TeX besonders geeignet, da dieses auch eine deutsche "plain-G" Version bietet, in der die Umlaute und das "ß" nicht als "Sonderzeichen" einzugeben sind! (Im reinen plain-TeX ist das ä beispielsweise doch

"umständlich" als \a zu schreiben; man kann das zwar im Editor durch einen globalen Ersetz-Befehl ausführen, aber es ist halt doch ein bisserl umständlich.)

Anschließend folgt der Teil einer ASCII-Datei aus einem astrophysikalischen Text, der einer Seite (mit vielen

Formeln) entspricht. Den entsprechenden (24-Nadel-Drucker-) Ausdruck des übersetzten ("DVI"-) Files auf Seite 13 sowie einen mit Laserdrucker hergestellte Ausdruck auf Seite 14 können Sie vergleichen:

```
Hence
$$T={G\over \sqrt{2e}}\{1\over (2\pi)^{9/2}\}k^2 \ ; \ \sqrt{\text{residue}}\ ; \ \bar{u}_{\nu}(p_2)\text{sla}\{e\} (1-\gamma_5) v_{\nu}(p_1).$$
In the longitudinal case we obtain from this
$$\text{align}\{w^{\text{rm}}\} \ \&={G^2\over 2e^2}\{1\over (2\pi)^2\}\{k^2\over \partial \epsilon^{\text{rm}}\} \ \cdot \ \text{cr} \ \&\cdot \ \int \{d^3p_1\over 2p_1^0\} \ \{d^3p_2\over 2p_2^0\} \ \delta(k-p_1-p_2) \ \underbrace{\text{Tr}\{\text{sla}\{e\}(1-\gamma_5) \ \text{sla}\{p\}_1 \ \text{sla}\{e\}-(1-\gamma_5) \ \text{sla}\{p\}_2\}}_{\displaystyle 8[2(e\cdot p_1)(e\cdot p_2)+(p_1\cdot p_2)]} \ \ . \ \text{cr}\}$$
Using
$$\int \{d^3p_1\over 2p_1^0\} \ \{d^3p_2\over 2p_2^0\} \ \delta(k-p_1-p_2) \ \text{p}_I^{\text{rm}} \ \{\mu\} \ \text{p}^{\text{nu}}_{-2}=\{\text{p}_I \ \over 24\} \ [2k^{\text{rm}} \ \{\mu\} \ k^{\text{nu}}+g^{\text{rm}} \ \{\mu\} \ \{\nu\} \ k^2]$$
we finally get
$$w^{\text{rm}}\{l\}=\{2\over 3\} \ ; \ \{G^2\over 4\pi e^2\} \ ; \ \{(k^2)^2\over \partial \epsilon^{\text{rm}}\} \ \cdot \ \text{eqno}\{\text{rm D.}26\}$$
Similarly one finds
$$w^{\text{rm}}\{t\}=\{2G^2\over 3(4\pi e^2)\} \ \text{eck}\{ \ \omega \ \text{rund}\{2\epsilon^{\text{rm}}\{t\}+ \ \omega \ \partial \epsilon^{\text{rm}}\{t\}\over \partial \omega}\} \ \cdot \ \text{eqno}\{\text{rm D.}27\}$$
For the total decay rate into all three neutrino flavors we conclude from (III.14) that the expressions (D.26) and (D.27) have to be multiplied by
$$C^2_{\text{rm V}}+2(C_{\text{rm V}}-1)^2 \ \text{simeq} \ 0.9. $$

\titleb {4. Calculation of the Dielectric Tensor}
For an approximate calculation of $\epsilon^{\text{rm}}\{t\}$ and $\epsilon^{\text{rm}}\{l\}$ we start from Eqs. (D.14) and (D.15) which we repeat here
$$J_{\text{rm}}\{\mu\}^{\text{rm}}\{\text{Pol}\}=\text{ave}\{j_{\text{rm}}\{\mu\}\} \ \text{eqno}\{\text{rm D.}14\}$$
$$J_{\text{rm}}\{\mu\}^{\text{rm}}\{\text{Pol}\}(k)=\text{Pi}_{\text{rm}}\{\mu\} \ \{\nu\}(k) \ \text{ave}\{A^{\text{rm}}\{\nu\}(k)\}. \ \text{eqno}\{\text{rm D.}15\}$$

Both $\text{ave}\{j_{\text{rm}}\{\mu\}\}$ and $\text{ave}\{A^{\text{rm}}\{\nu\}\}$ are only needed in a linear response approximation of the external field $\mathcal{A}^{\text{rm}}\{\nu\}(x)$ (corresponding to the external current $J_{\text{rm}}\{\mu\}(x)$).

Now we suppose we can express $J_{\text{rm}}\{\mu\}^{\text{rm}}\{\text{Pol}\}$ (in linear response approximation) in the following form
$$J_{\text{rm}}\{\mu\}^{\text{rm}}\{\text{Pol}\}(x)=\int K_{\text{rm}}\{\mu\} \ \{\nu\}(x-x') \ \mathcal{A}^{\text{rm}}\{\nu\}(x') \ d^4x'. \ \text{eqno}\{\text{rm D.}28\}$$
What is the relation between $\text{Pi}_{\text{rm}}\{\mu\} \ \{\nu\}$ and $K_{\text{rm}}\{\mu\} \ \{\nu\}$?

\noindent From Maxwell's equations we obtain
$$\text{A}_{\text{rm}}\{\mu\}(x)=\text{A}^{\text{rm}}\{\text{in}\}(x)+\int \text{nullueber}\{D\}_{\text{rm}}\{\mu\} \ \{\nu\}^{\text{rm}}\{\text{rm}}\{R\}(\bar{x}-x') \ [J^{\text{rm}}\{\nu\}(x')+j^{\text{rm}}\{\nu\}(x')] \ d^4x',$$
```

Hence

$$T = \frac{G}{\sqrt{2}e} \frac{1}{(2\pi)^{9/2}} k^2 \sqrt{\text{residue}} \bar{u}_\nu(p_2) \not{\epsilon}(1 - \gamma^5) v_\nu(p_1).$$

In the longitudinal case we obtain from this

$$w^1 = \frac{G^2}{2e^2} \frac{1}{(2\pi)^2} \frac{k^2}{\partial \epsilon^1 / \partial \omega} \int \frac{d^3 p_1}{2p_1^0} \frac{d^3 p_2}{2p_2^0} \delta(k - p_1 - p_2) \frac{\text{Tr}[\not{\epsilon}(1 - \gamma^5) \not{p}_1 \not{\epsilon}(1 - \gamma^5) \not{p}_2]}{8[2(e \cdot p_1)(e \cdot p_2) + (p_1 \cdot p_2)]}.$$

Using

$$\int \frac{d^3 p_1}{2p_1^0} \frac{d^3 p_2}{2p_2^0} \delta(k - p_1 - p_2) p_1^\mu p_2^\nu = \frac{\pi}{24} [2k^\mu k^\nu + g^{\mu\nu} k^2]$$

we finally get

$$w^1 = \frac{2}{3} \frac{G^2}{4\pi e^2} \frac{(k^2)^2}{\partial \epsilon^1 / \partial \omega}. \quad (\text{D.26})$$

Similarly one finds

$$w^t = \frac{2G^2}{3(4\pi e^2)} \left[\omega \left(2\epsilon^t + \omega \frac{\partial \epsilon^t}{\partial \omega} \right) \right]^{-1} k^2 [\omega^2 (\epsilon^t - 1)]^2. \quad (\text{D.27})$$

For the total decay rate into all three neutrino flavors we conclude from (III.14) that the expressions (D.26) and (D.27) have to be multiplied by

$$C_V^2 + 2(C_V - 1)^2 \simeq 0.9.$$

4. Calculation of the Dielectric Tensor

For an approximate calculation of ϵ^t and ϵ^l we start from Eqs. (D.14) and (D.15) which we repeat here

$$J_\mu^{\text{Pol}} = \langle j_\mu \rangle \quad (\text{D.14})$$

$$J_\mu^{\text{Pol}}(k) = \Pi_{\mu\nu}(k) \langle A^\nu(k) \rangle. \quad (\text{D.15})$$

Both $\langle j_\mu \rangle$ and $\langle A^\nu \rangle$ are only needed in a linear response approximation of the external field $A^\nu(x)$ (corresponding to the external current $J_\mu(x)$).

Now we suppose we can express J_μ^{Pol} (in linear response approximation) in the following form

$$J_\mu^{\text{Pol}}(x) = \int K_{\mu\nu}(x - x') A^\nu(x') d^4 x'. \quad (\text{D.28})$$

What is the relation between $\Pi_{\mu\nu}$ and $K_{\mu\nu}$?

From Maxwell's equations we obtain

$$A_\mu(x) = A_\mu^{\text{in}}(x) + \int \overset{\circ}{D}_{\mu\nu}^{\text{R}}(x - x') [J^\nu(x') + j^\nu(x')] d^4 x',$$

Hence

$$T = \frac{G}{\sqrt{2e}} \frac{1}{(2\pi)^{9/2}} k^2 \sqrt{\text{residue}} \bar{u}_\nu(p_2) \not{\epsilon}(1 - \gamma^5) v_\nu(p_1).$$

In the longitudinal case we obtain from this

$$w^1 = \frac{G^2}{2e^2} \frac{1}{(2\pi)^2} \frac{k^2}{\partial \epsilon^1 / \partial \omega} \cdot \int \frac{d^3 p_1}{2p_1^0} \frac{d^3 p_2}{2p_2^0} \delta(k - p_1 - p_2) \frac{\text{Tr}[\not{\epsilon}(1 - \gamma_5) \not{p}_1 \not{\epsilon}(1 - \gamma_5) \not{p}_2]}{8[2(e \cdot p_1)(e \cdot p_2) + (p_1 \cdot p_2)]}.$$

Using

$$\int \frac{d^3 p_1}{2p_1^0} \frac{d^3 p_2}{2p_2^0} \delta(k - p_1 - p_2) p_1^\mu p_2^\nu = \frac{\pi}{24} [2k^\mu k^\nu + g^{\mu\nu} k^2]$$

we finally get

$$w^1 = \frac{2}{3} \frac{G^2}{4\pi e^2} \frac{(k^2)^2}{\partial \epsilon^1 / \partial \omega}. \quad (\text{D.26})$$

Similarly one finds

$$w^t = \frac{2G^2}{3(4\pi e^2)} \left[\omega \left(2\epsilon^t + \omega \frac{\partial \epsilon^t}{\partial \omega} \right) \right]^{-1} k^2 [\omega^2 (\epsilon^t - 1)]^2. \quad (\text{D.27})$$

For the total decay rate into all three neutrino flavors we conclude from (III.14) that the expressions (D.26) and (D.27) have to be multiplied by

$$C_V^2 + 2(C_V - 1)^2 \simeq 0.9.$$

4. Calculation of the Dielectric Tensor

For an approximate calculation of ϵ^t and ϵ^l we start from Eqs. (D.14) and (D.15) which we repeat here

$$J_\mu^{\text{Pol}} = \langle j_\mu \rangle \quad (\text{D.14})$$

$$J_\mu^{\text{Pol}}(k) = \Pi_{\mu\nu}(k) \langle A^\nu(k) \rangle. \quad (\text{D.15})$$

Both $\langle j_\mu \rangle$ and $\langle A^\nu \rangle$ are only needed in a linear response approximation of the external field $\mathcal{A}^\nu(x)$ (corresponding to the external current $J_\mu(x)$).

Now we suppose we can express J_μ^{Pol} (in linear response approximation) in the following form

$$J_\mu^{\text{Pol}}(x) = \int K_{\mu\nu}(x - x') \mathcal{A}^\nu(x') d^4 x'. \quad (\text{D.28})$$

What is the relation between $\Pi_{\mu\nu}$ and $K_{\mu\nu}$?

From Maxwell's equations we obtain

$$A_\mu(x) = A_\mu^{\text{in}}(x) + \int \overset{\circ}{D}_{\mu\nu}^{\text{R}}(x - x') [J^\nu(x') + j^\nu(x')] d^4 x',$$

Kurz-Anleitung zum Arbeiten mit "muMATH-83" + Übungs- und Demo-Texte

TGM_96: MUMATH.TXT, TGM_97: MUMATH.ARC
Dr. Fridbert Widder, #326

Auf der PCC-TGM Diskette #1187 befinden sich

- das Programm MUSIMP.COM
- und die vier "memory image files"
ALGEBRA.SYS
CALCULUS.SYS
MATSOL.SYS
PDS.SYS

Um algebraische Umformungen zu machen bzw., um die vier Rechentechnik-Übungen CLES1.ALG bis CLES4.ALG zu bearbeiten, startet man am besten mit (der DOS-Eingabe):

MUSIMP ALGEBRA

Nach kurzer Lade-Zeit kommt die System-Meldung und das "?", das eine muSIMP-Eingabe erwartet.

Man probiert zuerst gleich einmal den in Abschnitt 6-2 beschriebenen "interaction cycle" aus: Abschließen der muSIMP-Eingaben mit dem ";" nicht vergessen - sonst geschieht nichts!

(Verlassen des Programms mit: SYSTEM();)

Oder/und man geht der Reihe nach die "Calculator mode LESsons" durch - mit dem "Read Select"-Kommando:

RDS(CLES1, ALG, A);

gelangt man sofort "in" die erste Übung (, wenn sich CLES1.ALG auf der Diskette im angegebenen Laufwerk A befindet).

Diese "CLESSons" erklären "sich" und den Umgang mit muMATH (und muSIMP) (fast) "von selbst" - trotzdem, oder weil's mir so gut gefallen hat, reproduziere ich in der Folge die (mir wichtig erscheinenden) Text-Teile dieser Übungen (und danach die Demonstrations-Beispiels-Beschreibungen, die man am Ende der diversen "muMATH source files" findet).

(Vorher noch ein Hinweis zu den weiteren bereitgestellten "memory image files": Der Zweck der zwei Dateien CALCULUS und MATSOL.SYS ist offensichtlich - PDS.SYS hilft beim Studieren der Programmier-Lektionen PLES1 bis PLES5 und PLES7 und PLES8.PDS. "SYS"-Dateien werden wesentlich schneller geladen, als "source files" eingelesen werden könnten!)

Bevor nun die (ein wenig gekürzten) Texte zu den Rechen-Modus-Übungen und anschließend die Texte zu den "muMATH"-Demonstrationsbeispielen folgen, füge ich*) noch ein "Inhalts-Verzeichnis" dieser Zusammenfassung ein -- die Seiten-Numerierung wurde so gewählt, daß man diese Seiten als Ergänzung zu den Seiten 8-1 und 8-2 des Handbuches ansehen kann.

INHALT der Zusammenfassung der Übungs- und Demonstrations-Beispiele "muMATH- 83"

Seite Inhalt:

- 8-3 CLES1: Übungsablauf, Eingeben und Auswerten einfacher algebraischer Formeln, Zuweisungen (von Werten an Variable)
- 8-5 CLES2: Kontrollvariable RDS, ECHO, BELL, POINT, PBRCH, Funktion RADIX, Faktorielle, #PI, #I, #E
- 8-7 CLES3: Variable (mit bzw. ohne Wertzuweisung - der 'Operator), Funktionen EXPAND, EXPD, FCTR, DIVOUT, PQUOTE, PREM, PGCD, PARFRAC, NUM, DEN, COEFF, CODIV, BASE, EXPON, CONJ, RATIONALIZE, Speicherplatz-Abfrage RECLAIM();
- 8-9 CLES4: Aktuelle Werte von Variablen - EVAL, ESUB Kontrollvariable (Übersicht - FLAGS()); - NUMNUM, ... DENDEN
- 8-12 Ergänzungen, NEWLINE,
- 8-13 ARITH.MUS: Grundlagen der Arithmetik, Funktionen ABS, MIN, GCD, LCM
- 8-14 ALGEBRA.ARI: Funktionen EXPAND etc..., NUM, DEN
- 8-15 EQN.ALG: Der Gleichungs-Operator = , Umformen von Gleichungen, SOLVE.EQN: Lösen mit SOLVE(Glg, Var) - vgl. S. 8-21: LINEQN.MAT
- 8-16 ODE.SOL: "Ordinary Differential Equations"
- 8-17 ODENTH.ODE: " --- " - of N-TH order"
- 8-18 ODEMOR.E.ODE: " --- " - , MORE about"
- 8-19 ARRAY.ARI: (Spalten- und Zeilen-)Vektoren, Matrizen
- 8-20 MATRIX.ARR: Matrix-Operationen
- 8-21 LINEQN.MAT: Lineare Gleichungssysteme - Matrix-Inversion ABSVAL.ALG: ABS als Präfix-Operator
- 8-22 LOG.ALG: Logarithmus-Funktionen
- 8-23 TRG.ALG: Trigonometrische Funktionen - Umkehrfktn.: ATRG.TRG
- 8-25 ATRG.TRG und die hyperbolischen Funktionen in der Datei HYPER.ALG
- 8-26 DIF.ALG: Differentiation, TAYLOR-Entwicklung
- 8-27 INT.DIF: bestimmte und unbestimmte Integrale
- 8-28 INTMORE.INT: Mehr darüber - " -LIM.DIF: Grenzwertbestimmung
- 8-29 SIGMA.DIF: Summationen
- 8-30 VEC.ARR: Vektor-"In"- und "Ex"-Produkt
- 8-31 VEC.DIF.VEC: Vektor-Differential- und Integral-Operationen, (DIV, CURL, GRAD, ...)

Anm.d.Red.: Den Abdruck der hier nur zusammenfassend erwähnten Übungs- und Demonstrationsbeispiele finden Sie in der Datei MUMATH.DOC.

PROGRAMMIEREN mit musIMP

TGM_96: MUSIMP.TXT, TGM_97: MUSIMP.ARC
 Dr. Fridbert Widder, #326

musIMP (und ähnliche Programm-Sprachen wie z.B. ST-MATH für den ATARI-ST) sind gewissermaßen LISP-

"Dialekte". Wesentlich dabei ist, daß Daten und Operationen mit "Zeigern" versehen sind, welche die Verknüpfungen und die Lage im Speicher bestimmen. Das gestattet sehr platzsparende und schnelle Daten-Transfers und Listen-Operationen.

1.1 STRUKTUR der DATEN

Es gibt drei elementare Grundstrukturen:

NAMEn
 INTEGERS = Zahlen
 NODEs = Knoten

Namen und Integers werden gemeinsam als "ATOM"e bezeichnet - sie haben selbst folgende Unterstruktur:

NAME: Ein Name besteht aus drei Zeigern - die beiden ersten Zeiger erhält man durch Anwendung der Funktionen FIRST bzw. REST:

1. Zeiger (FIRST-->) weist auf den Wert des Namens; bei der ersten Verwendung eines Namens weist der 1. Zeiger auf den Namen selbst.
2. Zeiger (REST -->) weist auf die "property"-Liste; bei der ersten Verwendung wird dem Namen die "leere Eigenschaft" FALSE beigegeben.
3. Zeiger weist auf Funktions-Definition, falls der Name eine Funktion definiert - sonst auf FALSE.

Hinter diesen drei Zeigern steht noch die ASCII-Zeichenkette des "Namens" (= des "Bezeichners" -) des Namens; der Bezeichner, unter dem wir uns diese Variable merken, darf bis zu 254 Zeichen enthalten. Gewöhnlich beginnen Namen mit Buchstaben - setzt man die Zeichen zwischen "-Zeichen, dann sind beispielsweise auch "137" und "" vom Typ NAME (" " ist der leere Name).

INTEGER: ist aus verarbeitungs-technischen Gründen analog strukturiert:

1. Zeiger --> Wert der Zahl
2. Zeiger --> Vorzeichen (als "property") der Zahl: + = FALSE
 - = TRUE
 Daher also Vorzeichentest durch REST(zahl)& - Eingabe!
3. Zeiger weist auf den Speicherbereich, in dem die Zahl binär gespeichert ist. (Kein Zugriff durch den Benutzer)

Mit den "recognicer"-Funktionen NAME(argum) und INTEGER(argum) kann man feststellen, von welchem Typ das ATOM "argum" ist.

NODE: Ein Knoten besteht aus ZWEI ZEIGERN, die auf die Adressen beliebiger Daten-Strukturen zeigen - sowohl auf Atome, als auch wieder auf Knoten, so daß beliebig komplexe Netzwerke entstehen können.

Eine einfache graphische Darstellung: Knoten-Nummer => Zahl.
 1. Zeiger (FIRST) => |
 2. Zeiger (REST) => ---

Einfache Beispiele:

```
1.---31
 |
FRITZ
```

```
1.---3.---D
 |   |
2.-B C
 |
A
```

```
1.---2.---3.---4.---FALSE
 |   |   |   |
A   B   C   D
```

Die dritte Ketten-Struktur hat einen eigenen Namen - es ist eine LISTE. Listen spielen in muLISP eine zentrale Rolle - daher gibt es neben dem ; Zeichen zum Abschluß eines Rechen-Ausdruckes auch den Listen-Ausgabe & Befehl. - Am besten dazu ein Beispiel probieren:

Falls Sie nicht ohnehin muSIMP geladen haben, dazu also MUSIMP PDS eingeben (- lädt den Programm-Demo System-file -) und beispielsweise den Unterschied zwischen den Ausgabe-Befehlen ; und & ausprobieren:

```
? a: -5$
? a;
? a&
```

Viele weitere Übungen zu den Grund-Strukturen sind im muMATH-Paket PLES1.PDS enthalten - einzulesen mit RDS(PLES1,PDS<,Laufw.>);

In der Folge beziehe ich mich auf diese erste "Programming LESSon 1".

Dort findet man eine spezielle Art der EINGABE von NODE-Strukturen genau erklärt - es handelt sich um die "dotted pair"-Darstellung, die wie folgt eingegeben wird:

Beginn mit dem (= Präfix-Operator) ' ein Knoten, dessen 1. bzw. 2. Zeiger auf A bzw. B weist, wird dargestellt durch:

```
(A . B)
```

Abschluß (am besten immer) mit dem Listen-Abschluß-Symbol &

Probieren Sie, einfache Knoten- und Listen-Gebilde einzugeben - und zu analysieren! - Dazu dienen die SELECTOR-Funktionen FIRST, REST etc...

Diese können beliebig geschachtelt werden - z.B. FIRST(REST(FIRST(xy)))&

Die wichtigsten Kombinationen liegen aber schon als fertige Funktionen vor:
 SECOND() = FIRST(REST()) (auch als FREST zu bezeichnen)
 THIRD() = FIRST(REST(REST())) (analog: FRREST) -
 RREST, FFIRST, RFIRST, FFREST, RFREST, RRREST, FFFIRST, FRFIRST, RFFIRST, RRFIRST.

Man kann sich (eingelese) existierende muMATH-Funktionen mit dem DISPLAY('funktionsname)\$ - Befehl ansehen .

Man kann aber auch eigene Funktionen erstellen. Dazu (und auch sonst zum muSIMP-Programmieren) eignet sich gut der

1.2 muSIMP "PDS"-EDITOR

(Wie der Name sagt, im PDS.SYS-file integriert - beschrieben in PLES1.PDS) Er bietet die wesentliche Erleichterung, mit den "Pfeiltasten" auch die Zeilen wechseln zu können, der Insert/Delete-Umschalter wirkt u.s.w.

Um eine Funktion, beispielsweise eine "recognizer"-Funktion namens NODE zu erzeugen, gibt man (bei geladenem PDS.SYS) ein:

```
EDIT ('NODE) &
```

- der Schirminhalt verschwindet zugunsten des Editor-Displays: in der ersten Zeile steht bereits FUNCTION (), und in der 2. das abschließende ENDFUN\$

- dazwischen schreibt man seine Funktion hinein, im Beispiel etwa:

```
FUNCTION NODE (X),
  NOT ATOM (X),
  ENDFUN$
```

Verlassen des PDS-Editors mit [Ctrl-K] !

Darauf wird einem angeboten, mit "redefine" die editierte Funktion zu übernehmen - oder auszusteigen.

Um die Funktion NODE als "Quell-Datei" zu sichern: FLAGSAVE('NODE) \$

Um sie auf Diskette (Laufw. A) zu schreiben: WRITE('NODE, 'MUS, A) &

Bevor ich mit dem Programmieren und den weiteren muMATH - "PLES"sons weitermache, möchte ich einen Übersicht-Abschnitt "GRUNDFUNKTIONEN" einfügen.

2. GRUNDFUNKTIONEN

Bereits bekannt sind u.a. die Zuweisungsfunktion :

und die Terminatoren, die nach der Auswertung des Ausdrucks folgende Ausgabe bewirken:

```
$ keine Ausgabe am Bildschirm
; Ausgabe - womöglich in mathematischer Notation
& Listen-Ausgabe
```

Um die Unterschiede an Beispielen zu demonstrieren, seien folgende Namen, Zahlen und Listen definiert:

```
NAM1: MAX
NAM2: MORITZ
```

ZAHL1: 100
 ZAHL2: -23
 ZAHL3: 0

L1: LIST(1,2,3,4,5,6)
 L2: LIST(KARIN,ELKE,THEA,HEIDI)
 L3: LIST(3,ANDREA,4,5,BEATE)
 L4: LIST(NAM1,ZAHL1,ZAHL2,ZAHL3)
 L5: LIST(10,11,LIST(12,13),LIST(14),15)

Dabei wurde die LISTen-Erzeugungsfunktion LIST(ob1,ob2,...,obN) benutzt, die im Abschnitt 2.2 noch einmal beschrieben wird.

2.1 Die Terminatoren ; und &

bewirken bei Zahlen und Namen genau die gleiche Ausgabe - Unterschiede gibt es bei Listen:

L1& --> (1 2 3 4 5 6)
 L1; --> 1 (2, 3, 4, 5, 6)

; unterteilt eine Liste in FIRST- und (REST)-Teil, dabei werden Elemente durch Beistriche getrennt. Ein weiterer Unterschied bei geschachtelten Listen:

L5& --> (10 11 (12 13) (14) 15)
 L5; --> 10 (11, 12(13), 14(), 15)

& zeigt Anfang und Ende jeder Ebene durch (und) an.
 ; zeigt die REST-Listen in () an -- einzelne Atome, die als Liste eingegeben wurden, werden mit der leeren Liste () angezeigt.

In jedem Falle werden Listen vor der Ausgabe ausgewertet, also:

L4; --> MAX (100, -23, 0)

2.2 Selektions- und Konstruktions-Funktionen

wurden z.T. auch schon im 1.Abschnitt behandelt. So die SELEKTIONS-Funktionen: FIRST, REST, SECOND, ... RRREST

Mit diesen kann man beispielsweise beliebige Elemente aus Listen heraus schälen:

RRREST(L2)& --> (HEIDI)

Von den KONSTRUKTIONSFunktionen kennen wir LIST(ob1, ... obN) durch den Gebrauch in der Praxis.

Um einen Knoten (NODE) zu konstruieren, gibt es auch die Funktion

ADJOIN(ob1,ob2) -- sie liefert den Knoten in der "dotted pair" Darstellung

ADJOIN(a,b)& --> (a . b)
 ADJOIN(1,FALSE)& --> (1)
 ADJOIN(NAM1,L1)& --> (MAX 1 2 3 4 5 6)

Während bei Listen die REST-Zelle immer auf einen Knoten oder auf FALSE zeigt, können bei dieser Konstruktion auch Paare von Atomen erzeugt werden.

Man beachte den Unterschied zwischen der Konstruktion eines Knotens mit ADJOIN bzw. durch die direkte Konstruktion mittel "evaluator" ':

'(a . b) &
 Vergleiche dazu die "Programming LESSon"2 -- s.S. PLES2-5 I

REVERSE(liste) dreht die Reihenfolge der Listen-Elemente um:

REVERSE(L5)& --> (15 (14) (12 13) 11 10)

REVERSE(liste1, liste2) dreht liste1 um und vereinigt sie mit liste2:

REVERSE(L1,L2)& --> (6 5 4 3 2 1 KARIN ELKE THEA HEIDI)

Wird aber statt listel nur ein Atom eingegeben, dann wird nur liste2 ausgegeben:

```
REVERSE(7,L1)& --> (1 2 3 4 5 6)
```

aber bei REVERSE(L1,7)& --> (6 5 4 3 2 1 . 7)

OBLIST()& konstruiert eine Liste, in der ALLE NAMEN aufscheinen, die zur Zeit im Speicher sind ! Schon beim Start sind das eine ganze Menge - später selbst eingeführte scheinen am Schluß der Liste auf.

2.3 ARITHMETISCHE FUNKTIONEN

Neben den Operatoren +, -, * gibt es auch Funktionen PLUS(a,b), DIFFERENCE(a,b), TIMES(a,b). Und neben / gibt es die (drei) Divisions-Funktionen:

```
QUOTIENT(5,2)& --> 2
```

```
QUOTIENT(-5,2)& --> -3
```

```
MOD(5,2)& --> 1
```

```
MOD(-5,2)& --> 1
```

und als Kombination der beiden: DIVIDE(var1,var2) --> (quot . mod)

```
DIVIDE(5,2)& --> (2 . 1)
```

2.4 LOGISCHE FUNKTIONEN

NOT(obj) gibt dann und nur dann TRUE, wenn obj auf FALSE zeigt.

Bekannt sind die Wirkungen der "Infix"-Operatoren AND und OR.

2.5 MODIFIZIERUNGS-FUNKTIONEN

Sie greifen direkt auf Knoten zu und können diese (manchmal gefährlich!) verändern.

REPLACEF(o1, o2) ersetzt die FIRST-Zelle von "o1" durch einen Zeiger auf "o2". Wenn "o1" ein Atom ist, wird die FIRST-Zelle des Atoms ersetzt: gefährlich wäre z.B.: REPLACEF(1,2) !
"o1" = Liste: erstes Element der Liste durch "o2" ersetzt
"o1" = Knoten: das linke Element wird durch "o2" ersetzt.

REPLACER(o1, o2) ersetzt die REST-Zelle von "o1" durch Zeiger auf "o2".

CONCATEN(liste, objekt) verbindet eine Liste mit beliebigem Datenobjekt.

2.6 PROPERTY-FUNKTIONEN

Aufbau und Verwendung von "Property"-Listen.

Eine Propertyliste wird immer unter einem NAMEN gespeichert - die REST-Zelle des Namens enthält den Zeiger auf den Anfang der P-Liste. In der FIRST-Zelle eines Knotens ist der Indikator, in der REST-Zelle der zum Indikator passende Eigenschaftswert. Den Aufbau besorgt die Funktion

```
PUT(name, indikator, eigenschaft)
```

Beispiel: Erstellen einer Telefon-Liste "telefon":

```
PUT(telefon,HANS,12345)&
```

```
PUT(telefon,MAX,23456)&
```

```
PUT(telefon,BOB,34567)&
```

Zum Ansehen von Propertylisten dient die Funktion REST(name).

Die Antwort im Beispiel:

```
REST(telefon) ---> ((BOB . 34567) (MAX . 23456) (HANS . 12345))
```

Die Speicherbelegung sieht kompliziert aus:

NODE-Nummer	Inhalt der FIRST-Zelle	Inhalt der REST-Zelle
1	Zeiger auf node2	Zeiger auf node3
2	Zeiger auf BOB	Zeiger auf 34567
3	Zeiger auf node4	Zeiger auf node5
4	Zeiger auf MAX	Zeiger auf 23456
5	Zeiger auf node6	Zeiger auf node7
6	Zeiger auf HANS	Zeiger auf 12345

Als Eigenschaften können selbstverständlich auch Namen und sogar Listen eingegeben werden.

Änderungen in schon aufgebauten P-Listen besorgt ebenfalls die Funktion PUT(name, indikator, property). Sollte also "BOB" eine neue Telefon-Nummer bekommen haben:

PUT(telefon, BOB, MNN)

Arbeiten mit Propertylisten.

ASSOC(NAM, PLIST) prüft, ob der Indikator "NAM" in der P-Liste "PLIST" enthalten ist; dabei muß aber die P-Liste selbst (und nicht bloß ihr Name) als zweiter Parameter übergeben werden. Im Beispiel:

ASSOC(HANS, REST(telefon)) ----> (HANS . 12345)

ASSOC(HANS, telefon) ----> telefon

ASSOC(MITZI, REST(telefon)) --> FALSE

GET(name, indikator) liest den Eigenschaftswert aus der P-Liste, die unter "name" eingerichtet wurde, der dem eingegebenen Indikator zugeordnet ist; wenn keiner vorhanden: FALSE

GET(telefon, HANS) ----> 12345

GET(telefon, Maxi) ----> FALSE

2.7 "SUBATOMARE" FUNKTIONEN

Wie der Name andeutet, können diese (drei) Funktionen ATOME "spalten".

LENGTH(objekt) - wenn objekt ein NAME ist, ----> Anzahl der Zeichen
 - wenn object = INTEGER ----> Anzahl der Ziffern (bezogen auf die geltende Basis)
 - wenn objekt = LISTE ----> Anzahl der Listenelemente.

EXPLODE(atom) - NAME / INTEGER ----> in einzelne Zeichen / Ziffern aufgespalten (Listenform der Ausgabe)

Umgekehrt kann man die einzelnen Bestandteile wieder zum ATOM verschmelzen mit COMPRESS(liste).

2.8 TEST-Funktionen

Dienen zur Abfrage von (Eigenschaften von) Datenobjekten - als Ergebnis erhält man die Antwort TRUE bzw. FALSE.

NAME(obj) prüft, ob es sich bei "obj" um einen NAMEN handelt -
 NAME(LISTE1) ----> FALSE
 NAME('LISTE1) ----> TRUE (zu "" vgl. auch 2.10!)

INTEGER(obj) INTEGER-Prüfung

ATOM(obj) ATOM-Prüfung

EMPTY(obj) dient dazu, leere Listen zu erkennen. Nur wenn "obj" gleich FALSE ist, wird als Ergebnis TRUE geliefert. EMPTY() ist der Funktion NOT() identisch.

POSITIVE(obj) ----> TRUE, wenn "obj" eine positive Zahl ist; analog

NEGATIVE() für negative Zahlen-Prüfung und

ZERO(o) ----> TRUE, wenn "o" Null ist.

EVEN(obj) Abfrage auf gerade Zahlen.

2.9 VERGLEICHS-Funktionen

geben (wie Testfunktionen) TRUE oder FALSE als Antwort, wenn der Vergleich stimmt oder nicht.

GREATER(zahl1,zahl2)
LESSER(zahl1,zahl2)

Das Ergebnis ist sicher FALSE, wenn eines der Vergleichsobjekte keine Zahl ist. Dieselben Abfragen realisiert man i.a. einfacher durch die Infix-Operatoren ">" und "<", z.B.: zahl1 > zahl2&

objekt1 EQ objekt2 -- Der Infix-Operator "EQ" prüft auf Gleichheit (von Namen, Zahlen, Listen). Beispiele:

NAM1 EQ MAX --> TRUE
100 EQ ZAHL1 -> TRUE

LIST(1,2,3) EQ LIST(1,2,3) ----> FALSE, weil bei der Auswertung dieser Funktion zwei an verschiedenen Speicherplätzen stehende - also verschiedene! - Listen erzeugt werden. Listen werden nur durch folgende Zuweisung identisch gemacht:

LISTE6:LISTE1\$ - danach gibt
LISTE1 EQ LISTE6 tatsächlich TRUE.

Der Infix-Operator "=" wirkt etwas "schwächer" als "EQ"; die Vergleichs-Objekte müssen nicht mehr identisch sein - es genügt, wenn sie gleich sind, um als Antwort TRUE zu bekommen. Das hat nur bei Listen eine Bedeutung: LIST1 = LIST2 ----> TRUE, wenn LIST1 strukturell gleich LIST2 ist!

MEMBER(objekt,liste) prüft, ob "objekt" in "liste" enthalten ist.

ORDERP(name1,name2) vergleicht, ob "name1" in der OBLIST() vor "name2" steht. (Jeder neu eingeführte Name wird in die OBLIST() aufgenommen und zwar in der Reihenfolge der Einführung: früher eingeführte stehen weiter rechts in der OBLIST().)

2.10 ZUWEISUNGS-Funktionen

Der Infix-Operator ":" ist schon bekannt. Links muß immer ein NAME stehen - rechts kann ein beliebiges Daten-Objekt stehen; bei der Zuweisung mit ":" wird dann die FIRST-Zelle des NAMEs entsprechend modifiziert. Z.B.:

? NAME2 : SUSI &
@: SUSI

Nach & und "return"-Eingabe geschieht folgendes: Wenn "NAME2" noch nicht in der OBLIST() enthalten ist, wird er initialisiert, indem seine FIRST-Zelle auf sich selber zeigt. Das gleiche geschieht mit "SUSI". Dann werden "NAME2" und "SUSI" als Parameter des ":"-Operators interpretiert --> die FIRST-Zelle von "NAME2" wird mit der Adresse von "SUSI" belegt.

Der Prefix-Operator "" bewirkt, daß NICHT der WERT einer Variable, sondern ihr Name, ihre Bezeichnung übernommen wird:

NAME2 ----> SUSI
'NAME2 ----> NAME2
'ZAHL1 ----> ZAHL1

Der QUOTE-Operator ' verhindert also die Auswertung des Daten-Objektes, vor dem er steht.

Etwas "trickreicher" wirkt die ASSIGN(name,objekt) Funktion; ein Beispiel:

? NAME2:SUSI&
@: SUSI

? SUSI:BLOND\$

? SUSI&
@: BLOND

und jetzt:

```
? ASSIGN(NAME2, '?)&
@: ?
```

```
? NAME2&
@: SUSI
```

```
? SUSI&
@: ?
```

Die ASSIGN-Funktion belegt nicht wie ":" die FIRST-Zelle von "name", sondern verändert die FIRST-Zelle der FIRST-Zelle von "name"!

PUSH(objekt,liste) legt das "objekt" auf die "liste", die gewissermaßen als ein Stapel dient:

```
? STAPEL&
@: STAPEL
```

```
? PUSH(KARIN,STAPEL)$
```

```
.
```

```
? PUSH(HEIDI,STAPEL)$
```

```
? STAPEL&
```

```
@: (HEIDI, . . KARIN . STAPEL)
```

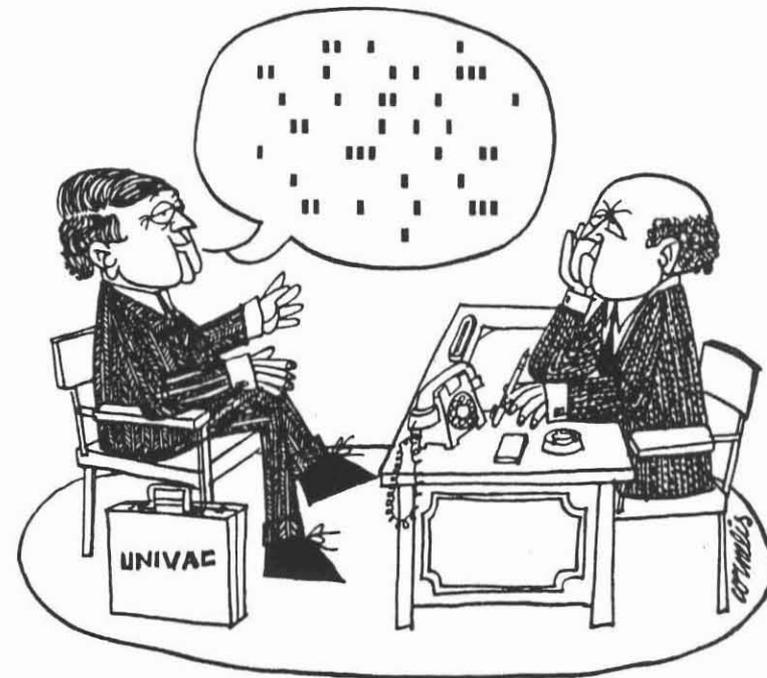
Die "Umkehr"-Funktion POP(liste) nimmt das oberste Element von der "liste":

```
? POP(STAPEL)&
@: HEIDI
```

```
? STAPEL&
```

```
@: ( . . KARIN . STAPEL)
```

Eine "echte" Liste wird mit PUSH(... ,stap) erzeugt, wenn der Wert von "stap" am Anfang auf FALSE gesetzt wird.



Fachausdrücke wie record, file, message, real time, disc, interrupt usw. hinterlassen beim EDV-Neuling immer einen positiven Eindruck.

Sie erreichen damit, daß Ihr Gesprächspartner mit Sicherheit annimmt, Sie seien Experte.

Tips und Tricks zur Herculeskarte

TGM_96: HERCULES.TXT, TGM_97: HGC.ARC
 Helmut Hörner, #513

Besitzer einer Herculeskarte haben nicht immer nur Grund zur Freude. Der Grund: Viele Programme verweigern sich einfach, sobald im Rechner eine Herculeskarte steckt; sie ziehen sich ins Schmallwinkerl zurück und geben - wenn man Glück hat - vielleicht t noch Töne, aber überhaupt kein Bild mehr von sich.

Eine mögliche Ursache kann nun darin liegen, daß das Programm für eine andere Grafikkarte geschrieben wurde. Das Programm EGA-Paint zB. ist - unschwer zu erraten - für eine EGA-Karte geschrieben, und nur auf einer solchen lauffähig.

Viele professionelle Softwarepakete (ACAD, Framework, RS/1 usw.) kann man aber bei der Installation oder mit Hilfe eines Setup-Programms auf eine bestimmte Grafikkarte einstellen. Ist dies geschehen, gibt es zwei Möglichkeiten: Das Programm läuft - oder es läuft noch immer nicht. STATGRAF und RS/1 gehören zB. zu dieser hinterlistigen zweiten Gruppe.

Hier beginnt der Punkt, an dem viele aufgeben. War halt nichts. Fragt man den Händler, bekommt man meist zu hören: "Die geraubte Software ist schuld..." (Das ist übrigens eine sehr häufige Ausrede, auch bei Hardwarefehlern!). Aber da wir ohnedies alle Originaldisketten verwenden und unseren Lizenzvertrag haben, wissen wir: Das ist nicht war. Auch die Originale laufen nicht.

Wahr ist vielmehr, daß die Hardwarehändler in gröblicher Vernachlässigung ihrer Pflicht zumeist vergessen, zusammen mit der Herculeskarte zwei sehr wichtige Programme mitzuliefern: HGC und INT10.

Wer folgende Sequenz in sein AUTOEXEC.BAT einbaut, müßte alle Programme, die von sich aus versprechen, auf Hercules zu laufen, auch tatsächlich zum Funktionieren bringen:

HGC FULL
 INT10

Was tun nun diese Programme? Das Programm HGC initialisiert die Register der Herculeskarte, und das Programm INT10 ersetzt die Video-IO-Routine des BIOS durch eine "herculesfähige".

Das Programm HGC habe ich dem Club zusammen mit diesem Artikel übermittelt und das Programm INT10 findet sich auf der Diskette GRAPH X, zu der es auch eine Anleitung gibt.

Außerdem auf dieser Diskette ist das Programm HARD-COPY, welches ein Grafik- Hardcopy auch von der Herculeskarte ermöglicht. Es ist übrigens mit Hilfe eines "printer definition files" PRINTER.DEF auf jeden Drucker anpaßbar.

Das war aber noch nicht alles: Auch jene Programme, die nur mit der CGA- Karte funktionieren (zB. Printmaster und die meisten Spiele), können zumeist auf der Herculeskarte zum laufen gebracht werden.

Zu diesem Zweck gibt es Emulationsprogramme, die eine CGA-Karte zu emulieren imstande sind. Ich selber habe zwei davon getestet: Multigraph II (kurz: MGII) und Magic Key (kurz: MK).

Vergleicht man die beiden Programme, stellt man fest: Beide Programme bieten verschiedene Emulationsmodi,

sowohl was die Emulationstiefe betrifft (also wie "gründlich" die CGA-Karte emuliert wird), als auch was die Art der Darstellung betrifft.

Standard ist bei beiden Programmen ein sehr schneller Modus, bei dem allerdings das Bild in der Höhe, wegen der unterschiedlichen Auflösung, ziemlich gestaucht erscheint. Beide Programme bieten aber auch einen Modus an, in dem das Bild in voller Höhe erscheint, aber dazwischen leere Zeilen bleiben (wer waagrechte Streifen mag, sollte diesen Modus wählen!).

Außerdem versprechen beide Programme einen Modus, in dem diese Leerzeilen interruptmäßig ca. 7 mal in der Sekunde aufgefüllt werden. Vorteil: Schönes Bild in richtiger Höhe. Nachteil: Verlust an Geschwindigkeit, bei Actionspielen häßliche Nachzieheffekte.

Übrigens: MGII verspricht diesen Modus als zusätzliche Neuheit in einem README-File, aktivieren konnte ich ihn aber nicht. Bei MK funktionierte er aber klaglos. Und damit wären wir auch schon bei den Unterschieden: MGII bietet zusätzlich einen Modus, in dem jedes Halbbild etwas versetzt dargestellt wird. Das ergibt ein Bild in richtiger Höhe ohne großen Geschwindigkeitsverlust - aber mit starkem Flimmereffekt. Diese Betriebsart kann nur mit sehr stark nachleuchtenden Monitoren verwendet werden.

Versuchsweise habe ich das Programm Printmaster mit beiden Programmen zu starten versucht. Funktioniert hat es bei beiden: Aber: Bei MGII waren in den Menüs die normalerweise hell unterlegten Menüpunkte auf einmal nicht von den anderen zu unterscheiden. MK funktionierte hier besser!

Aber MK hat nicht nur Vorteile: Während man MGII problemlos aus einem Batch- File aufrufen und dabei zwischen verschiedenen Modi umschalten kann (auch zurück zur reinen Herculesgrafik), muß man bei MK einen sehr langweiligen Titelaufbau abwarten und kann das Programm nur mehr mit obskuren Tastenkombinationen (zB.: ALT-CTRL-F1) umschalten. Abschalten kann man es nicht - nur mit einem Reset.

Vereint werden die Programme aber wieder durch die Fähigkeit, auch bootfähige Spiele laufen lassen zu können. Dazu wird zuerst die entsprechende Boot-Version von MGII oder MK aufgerufen und anschließend das Spiel ins Laufwerk A: gelegt. Weiter: Enter drücken. Und schon erscheint PAC MAN am Monitor!

H. Hörner N88A, #513

PS.: Für Fragen stehe ich gerne zur Verfügung. Meine Telefonnummer ist 95 29 535. Gegen ein kleines Entgelt (ich bin jetzt ein armer Student) bin ich aber auch gerne bereit, persönlich allfällige Probleme bei der Installation von Software (nicht nur im Zusammenhang mit der Herculeskarte) zu lösen.

Resetfeste RAM-DISK

TGM_96: RAMDISK.TXT
Johann WALZER, #328

Die Freude des Benutzers über den schnellen Zugriff zu einer virtuellen Diskettenstation wird getrübt durch das Risiko eines Datenverlustes beim 'Absturz' des Rechners. Diesen Zustand kann im allgemeinen nur ein Kalt- oder Warmstart beenden.

1. Hardware-Reset bzw. Kaltstart

1.1 Reset nach dem Einschalten (Power-Up)

In diesem Fall ist der RAM-Inhalt nicht definiert und ein Restaurieren von Daten daher nicht mehr möglich.

1.2 Reset durch Resetaste

Bei vorhandenem Reset-Taster wird nicht nur der Prozessor rüchgesetzt, sondern auch die Peripheriebausteine, wie Timer und DMA-Controller. Die dynamischen RAMs benoetigen aber alle 4 ms einen Auffrischungszyklus, der vom Timer 8253 ausgelöst und vom DMA-Controller 8273 durchgeführt wird. Das heißt, für dynamische RAMs fallen die Auffrischungszyklen aus, und sie verlieren damit ihren Inhalt. eine Abhilfe wäre durch die Verwendung eines Monoflops möglich, um die Resetzeit zu verkürzen, aber auch diese Methode führt nicht zum Ziel (siehe Pkt.2, Test der Speichergröße).

2. Software-Reset bzw. Warmstart

Wird durch die Tastenkombination CTRL-ALT-DEL ein Software-Reset eingeleitet, so merkt sich der Prozessor diesen Umstand durch ein Flag (RESET_FLAG auf Adresse 0000:0472). Auf diese Adresse wird 1234H geschrieben, ein Inhalt, der beim normalen Einschalten des PC's sicher nicht vorkommt. Dann verzweigt das Programm auf die gleiche Einsprungstelle (Label: START_1), wie beim Hardware-Reset. Die Peripherie-Bausteine werden nicht zurückgesetzt, d.h. das Auffrischen der RAMS läuft normal weiter.

Der Prozessor führt anschließend einen Selbst-Test durch und initialisiert die Peripherie-Bausteine neu (kurze Unterbrechung des Auffrisch-Zyklus). Ist das RESET_FLAG gesetzt (d.h. Inhalt gleich 1234H), so wird zwar kein Speichertest durchgeführt, aber er danach wird die Speichergröße getestet:

Diese Routine prüft in 64k Schritten, ob ein Speicher vorhanden ist (von 64k bis 640k auf dem Motherboard und bei ATs zusätzlich von 1M bis 16M-128k extended memory). Und in dieser Routine passiert!!! Es wird nach jedem Test der komplette 64k Block mit Nullen gefüllt und damit der vorige Inhalt zerstört.

3. Welche Möglichkeiten gibt es nun, das Zerstören des Speicher-Inhalts zu vermeiden?

3.1 Man ändert die entsprechenden Stellen im BIOS-EPROM. Vermutlich handelt man sich dabei wieder andere Schwierigkeiten ein, und besonders elegant ist es auch nicht.

3.2 Man verwendet ein residentes Programm /1/, das sich in die Tastatur-Routine einhängt und auf eine neue, halbsprecherische Tasten-Kombination wartet. Wird diese Kombination erkannt, so wird das momentan laufende Programm abgebrochen und in das Betriebssystem zurückgekehrt. Es gibt aber Einschränkungen,

unter denen dieses Programm nicht funktioniert (z.B.: Interrupts gesperrt, Vektoren verbogen...). Es sollte allerdings meistens funktionieren.

3.3 Gegen gesperrte Interrupts gibt's noch die Möglichkeit, den NMI-Eingang (Non-maskable-Interrupt) zu verwenden; das erfordert aber einen zusätzlichen kleinen Hardware-Eingriff.

Kann das laufende, aber abgestürzte Programm mit der oben beschriebenen Methode abgebrochen werden, so ist auch der Inhalt der RAM-Disk noch intakt! Hat man gerade in Turbo-Pascal gearbeitet, so kann man mit dem Debugger noch den vorhandenen Source-Code retten (ein bißchen mühsam zwar, aber es ist zumindest möglich).

/1/ Ralf Preller, Notausstieg, c't, 88/1, S 120..122.

Fehler im Phoenix ROM BIOS Ver.

2.27 ?

TGM_96: PHOENIX.TXT
Richard Skriwanek, N84c, #173

Es trat häufig folgender Fehler auf:

FATAL: Internal Stack Failure, System Halted

Diese Fehlermeldung zeigte sich besonders oft bei manchen Turbopascal-Programmen (Turbo 4.0 -- es kommt wahrscheinlich auf bestimmte Programmstrukturen an).

Nach mehreren Programmänderungen trat dieses Problem bei jedem Programmstart auf (einmal früher - einmal später -- früher, wenn das Programm mit Hilfe einer Autoexec-Datei gestartet wird).

Da es nur auf 2 Computern (Future Technologie - Phoenix Bios) zu Schwierigkeiten kam, lag es nahe, diesen Geräten die Schuld zu geben.

Tatsächlich konnte das Problem durch Tauschen des Bios-Roms beseitigt werden.

FRAMEWORK II MIT QUADRAM

386XT

TGM_96: WRUNTST.TXT
W.Nitsche/TGM/#27

FRAMEWORK-II-Anwender, die einen umfangreichen Frame bearbeiten wollen, kommen mit dem 640kB großen Arbeitsspeicher ihres XT meist nicht aus, weil das Programm und der Bibliothek-Frame mehr als die Hälfte des RAM verbrauchen. Bei Anwendung des Makros {Alt-F5} zeigt Ihnen FRAMEWORK, daß vielleicht nur mehr 236.608 Bytes vom Arbeitsspeicher frei sind. Festplattenbesitzer können in der Installation einen Teil ihrer Festplatte als Speichererweiterung konfigurieren, sodaß größere Frames bearbeitet werden können. Die Freude über diese Möglichkeit wird allerdings durch den erheblichen Zeitaufwand für das Ein- und Auslagern getrübt (Variante 1). Schneller geht es nur mit einer Speichererweiterung in Form eines RAM-Boards. Wenn der XT mit einem QUADRAM 386XT Board nachgerüstet wird, verkürzt sich der Ladevorgang eines großen Frames schon allein

deshalb, weil zusätzlich etwa 880 kB als Speichererweiterung zur Verfügung stehen. Diese können als Extended Memory (Variante 2) oder als Expanded Memory (Variante 3) konfiguriert werden.

Im folgenden ist eine Gegenüberstellung der Ladezeiten und des noch verfügbaren Arbeitsspeichers für 4 verschiedene Varianten angegeben. Zum Vergleich sind in der Va-

riante AT die Ergebnisse eines PHILIPS P 3202 (80286 Prozessor und Coprozessor) mit aufgerüstetem RAM wiedergegeben.

Hardware-Konfiguration:

XT kompatibler PC, nachgerüstet mit QUAD386XT
2 Floppylaufwerke, 1 20MB-Festplatte (NEC)

Variante 1	Variante 2	Variante 3	Variante AT
Konfiguration des QUAD386XT-RAM als EMS Emulation in der CONFIG.SYS (/a=xxx):			
beliebig	0 kB	880 kB	----
installierter EXTENDED-MEMORY Treiber im FW2-Setup: "Festpl./RAM-Disk" "RAM oberh. 640K" "Quadram Liberty"			
installierte Größe des EXTENDED-MEMORY: einheitlich 880K			
starten von FRAMEWORK II:			
1004.736 frei	1004.608 frei	955.312 frei	875.088 frei
aufrufen des Laufwerkverzeichnis C:			
999.696 frei	999.568 frei	950.272 frei	874.896 frei
aufrufen eines Unterverzeichnisses:			
993.696 frei	993.760 frei	944.464 frei	nicht vorhanden
einen 295.296 Bytes großen Frame laden:			
2 min 38 sek	26,0 sek	7,5 sek	26,1 sek
660.208 frei	660.240 frei	615.008 frei	539.056 frei
noch einmal diesen Frame laden:			
2 min 59 sek	55,8 sek	42,2 sek	1 min 27 sek
293.920 frei	294.016 frei	244.000 frei	172.016 frei

Listenschutz ade!

TGM_96: BASLIST.TXT

Andreas Leisser, #422, Hanfthal 160, 2136 Laa

Es herrschte bisweilen die Fehlmeinung, daß der Listenschutz in BASIC mit Hilfe von BLOAD und BSAVE entfernt werden könnte. Da nach dem Laden eines geschützten Programmes jedoch auch diese Befehle gesperrt werden, muß man auf Umwegen dem sonst unbezwingbaren Listenschutz zu Leibe rücken.

Folgendes Kurzprogramm ist mit DEBUG einzugeben:

```
e 100 ff 1a
n frei.bas
r cx
2
```

Mit w <ENTER> schreiben und mit q <ENTER> DEBUG verlassen.

Nach dem Laden des geschützten Programmes kann nun ohne Probleme der Listenschutz mit LOAD "FREI.BAS" entfernt werden.

FREI.BAS setzt an die für den Schutz zuständige Speicherstelle 1124 den Zustand wie vor dem Laden des geschützten Programmes.

COMPUTERVIREN

TGM_96: VIREN.TXT

Helmut Schluderbacher/TU/#528

Ein kurzes Vorwort: DER Virus oder DAS Virus? Diese Frage beschäftigte mich noch bevor ich diesen Artikel zu schreiben begann. Im DUDEN finde ich "DAS Virus" ist richtig, aber "DER Virus" ist zulässig. Da ich selbst DER Virus sage (weil ER ja was anstellt), habe ich mich entschlossen, obwohl es DAS Programm heißt, ES zu einem ER zu machen. Ich hoffe, mir ist niemand gram deswegen.

Viren, jeder, der mit Computern arbeitet, kennt sie und viele haben schon einmal mit ihnen Bekanntschaft gemacht. Manchmal gutmütig ("Type: Happy Christmas"), manchmal bößartig (FAT wird überschrieben, Directory wird durcheinander gebracht, Files verändert oder gelöscht), manchmal politisch ("FRIEDE ALLEN ANWENDERN" erscheint auf dem Schirm) beeinflussen sie uns und im schlimmsten Fall auch unsere Daten. Der elektronische Virus hängt vereinfacht dargestellt an einem Programm und kopiert, nach einem Aufruf des Programms, ein Duplikat von sich an ein oder mehrere Unverseuchte(s).

Die Auswirkungen sind in JEDEM Fall negativ. Sei es, weil man sich in seiner Arbeit (sprich Konzentration) gestört fühlt, weil Daten, WICHTIGE Daten, zerstört werden oder aber, weil Anwenderprogramme verschwinden oder abstürzen.

WIE KOMMT MAN ZU VIREN?

Jemand, der wie ich an der TU-Wien studiert, befindet sich damit sicher an einem der Hauptumschlagplätze von Computerseuchen. Viele Rechner, alle untereinander kompatibel, so etwas fördert natürlich sehr die Verbreitung. Wenn man nun auf einer dieser Maschinen arbeiten muß, so infiziert man sich mit großer Wahrscheinlichkeit.

Ich selbst habe mir auf diese Art zwei Viren "geholt" und nach Erkennen liebevoll kultiviert. So lustig das jetzt klingen mag, so wenig lustig war es ALLE Programme (*.COM und *.EXE) von der Festplatte zu löschen, nur um ein wenig damit herumzuspielen. Und hier kommen wir eigentlich zu einem der Kernpunkte:

WAS MACHEN DIESE PROGRAMME EIGENTLICH?

Diese Frage läßt sich so einfach nicht beantworten. Denn einerseits wird zu den Programmen kein Benutzerhandbuch mitgeliefert und die Viren selbst geben erst sehr viel später Auskunft, andererseits ist auch der Produzent (ich sage immer Schwachkopf) leider nicht bekannt. Um jedoch Ihnen als Clubmitglied die Möglichkeit zu geben, zu erkennen ob Ihr Rechner verseucht (versaut) ist, liste ich Ihnen die mir in den Auswirkungen bekannten Viren auf. Mir sind leider viele Namen nicht geläufig, unter denen diese Seuchen firmieren, und so sind viele Bezeichnungen eher praktischer Natur. Aber da niemand ein Copyright auf die Programme und Namen hat, ist es ohnehin gleich, wie man sie bezeichnet.

BAD-SEKTOR-VIRUS

Nach einem DIR Befehl hat die Diskette nachher 1024 Bytes in "Bad Sectors". Auch ist dann der Speicher um 2k veringert.

COMMAND.COM-VIRUS

Wenn mit einem verseuchten COMMAND.COM gestartet wird, genügt ein einfaches Zugreifen mit DIR, TYPE, usw. auf eine andere Diskette (Festplatte), sodaß auch dort der COMMAND.COM, wenn vorhanden, infiziert wird, wobei nur das Datum nicht jedoch die Länge des Eintrags geändert wird. Nach vier Infektionen löscht sich die Diskette (Festplatte) so gründlich, das heißt Bootsektor und FAT werden auf Null gesetzt, daß eine Rekonstruktion praktisch unmöglich ist.

Maßnahmen zur Erkennung: Booten Sie mit einer virusverdächtigen Disk und versuchen Sie dann auf eine schreibgeschützte Diskette, die einen COMMAND.COM hat mit DIR zuzugreifen.

DISK-VIRUS (BRAIN-VIRUS)

Wenn von einer verseuchten Diskette gebootet wird, installiert sich der Virus im Speicher und infiziert nach einer gewissen Zeit jede Diskette (nur DSDD), auf die zugegriffen wird. Dies geschieht durch Ersetzen des Bootsektors und das Setzen von drei Sektoren auf "Bad". Schließlich wird noch "(c) Brain" als neuer Volume Label Eintrag geschrieben. Der Eintrag scheint jedoch erst auf, wenn zwei oder mehr Files auf der Diskette abgespeichert sind. Weitere Auswirkungen sind derzeit nicht bekannt.

Maßnahmen zur Erkennung: Formatieren Sie eine Diskette, kopieren Sie einige Files darauf und sehen sich dann den Volume-Label-Eintrag an.

HERBST-VIRUS

Nach dem Aufruf eines infizierten Programmes bleibt der Virus im Speicher. Er hängt sich beim Aufruf an *.COM Files an und vergrößert sie um 1701 Bytes. Datum und Uhrzeit des Eintrages bleiben gleich. Wenn nun das Datum auf Oktober 1988 gesetzt wird, fallen alle Buchstaben am Bildschirm in die letzte Zeile. Bis jetzt konnten sonst keine anderen Auswirkungen festgestellt werden. Das Überschreiben des verseuchten Programmes mit dem gleichen unverseuchten ALLEINE genügt nicht (so wurde mir gesagt), da das Virusprogramm am Ende des Files steht.

Von diesem Virus gibt es allerdings noch eine sehr interessante Nebenlinie. Und zwar mit den gleichen Auswirkungen, allerdings nur, wenn die Maschine nicht von IBM!! ist. Das heißt IBM-Rechner sind von vorneherein immun gegen Aktionen. Die Verbreitung (Infektion) ist jedoch nicht eingeschränkt.

Maßnahmen zur Erkennung: Kopieren Sie ein Programm, rufen Sie es auf und vergleichen sie anschließend die beiden Files.

JERUSALEM-VIRUS (HEBREW UNIVERSITY VIRUS)

Es werden *.COM und *.EXE Files beim Aufruf infiziert. Die *.EXE Files werden ständig größer, die *.COM Files werden nur einmal vergrößert. Nach einiger Zeit aktiviert der Virus einen "aktiven Seitenvorschub" von der 5ten Zeile (Spalte 5) bis zur 16ten Zeile (Spalte 16), was Ihnen den Bildschirmaufbau zerstört. Zudem verlangsamt sich die Arbeitszeit. An jedem Freitag, den 13. wird jeder Befehl mit "Falscher Befehl oder Dateiname" beantwortet. Der COMMAND.COM wird aber nicht infiziert. Wenn jedoch das Datum auf 1987 gesetzt ist, passiert nichts (Don't ask me why).

Maßnahmen zur Erkennung: Kopieren Sie ein Programm, rufen Sie es auf und vergleichen sie anschließend die beiden Files.

VANDERMEERSCHENSER-VIRUS

Es verseucht *.COM Programme und produziert ab der fünften Generation nach einiger Zeit die Meldung: "Program sick error: call doctor or buy Pixel for cure description" und einen Systemabsturz. Benannt ist er nach seinem Entdecker (und nicht Erfinder). Nachzulesen in der Juniausgabe des "Ordinateur Individuel".

VIRUS1

Trieb im Wintersemester 87/88 an der TU-WIEN sein Unwesen. Sein Kennzeichen: Vergrößerung der Files um ungefähr 650 Byte. Er durchsucht das Directory nach *.COM Files solange, bis ein unverseuchtes Programm aufgefunden wird. Jedes achte Mal wird das Programm mit einem Code überschrieben, der einen Reset auslöst, ansonsten wird ein unverseuchtes Programm verseucht.

Maßnahmen zur Erkennung: Kopieren Sie ein Programm, rufen Sie es auf und vergleichen sie anschließend die beiden Files.

VIRUS2

Trieb im Sommersemester 88 an der TU-WIEN sein Unwesen. Vermutlich ident mit dem JERUSALEM-VIRUS.

VIRUS3

Trieb im Wintersemester 88/89 an der TU-WIEN sein Unwesen. Ident mit dem HERBST-VIRUS.

Womit bewiesen wäre: Jedes Semester hat seinen Virus.

Es gibt natürlich noch eine Menge Viren, die durch die Gegend seuchen. Zum Beispiel ein spezieller AT Virus der sich in den AT Uhrpufferspeicher schreibt und nach Rebooten alle erreichbaren Medien (1,2MB + 360kB + Festplatte + usw.) formatiert. Einzige Möglichkeit = Gerät ca. 24 Stunden abstecken und darauf warten, daß der Akku sich geleert hat. Alle Computerseuchen aufzulisten, ist mir leider nicht möglich, denn natürlich kenne ich nur einen Bruchteil, und andererseits höre ich von vielen Seiten Schauermärchen, die sich bei genauerer Analyse als einfache Bedienermängel herausstellen.

Wie kommen wir nun eigentlich dazu, uns mit so schwachsinnigen PROGRAMMEN herumzuärgern. Hier gibt es, wie es scheint, ganz interessante Details. Ich habe vor kurzem erst gelesen, daß ein Virus sich mit ganz bestimmten Applikationen beschäftigt und man vermutet, daß diese Seuche ein ausgekommener KOPIERSCHUTZ von einem bestimmten Programm ist. Hier stellt sich für mich die Frage:

"JA SIND DENN DIE NOCH ZU RETTEN?"

Scheinbar haben einige Softwarehersteller Seuchen produziert, um das Kopieren ihrer Software zu erschweren (oder ziehen es zumindest in Erwägung). Wie dumm so eine Art der Sicherung ist, liegt für einen, nur halbwegs INTELLIGENTEN, Menschen auf der Hand. Welche Firma die auch nur EINEN PC-kompatiblen Rechner hat, kann mit Sicherheit ausschließen, daß nicht einer ihrer Mitarbeiter das eine oder andere privat organisierte Programm auf die Festplatte spielt. Die Lächerlichkeit eines solchen "Kopierschutzes" wird noch deutlicher, wenn man bedenkt, daß dieser Schutz nur die Kleinen trifft und sicherlich nicht die professionellen Raubkopierhändler mit ihren großen Kopiergeräten. Leider gibt es noch immer Leute, die der Meinung sind, daß man auch für Programme, die man kommerziell nutzt, nichts bezahlen muß. Wovon der Softwarehersteller (sprich Programmierer) leben soll, fragt ja niemand. Diesen Leuten sei gesagt: "Auch kleine Programme haben ihren Autor und fallen nicht vom Himmel". Wenn mehr gekauft und weniger geklaut würde, könnten die Programme billiger werden. Die Frage ist leider nur: **WERDEN SIE AUCH BILLIGER?**

Ich möchte hier aber auch ganz ausdrücklich jene Gruppen ausnehmen, die, wie ich glaube, von den Produzenten schwer vernachlässigt werden. Ich meine damit Lehrer, Schüler und Studenten (Reihenfolge beliebig veränderbar). Wie soll bitte eine nur halbwegs gute Ausbildung funktionieren, wenn alles einigermäßen Aktuelle für diese Gruppen (da ja kein Gewinn damit erwirtschaftet wird) praktisch unerschwinglich ist. Und so sind wir in einer Situation, die eigentlich alle nicht wollen (Ich kann hier natürlich nur für die Studenten sprechen). Wenn es die PUBLIC DOMAIN nicht gäbe, säßen wir alle auf dem Trockenen, was vielleicht ganz lustig klingt aber nicht ist. Jemand aus diesen Kreisen hat mein vollstes Verständnis, wenn er mit "Kopien" arbeitet.

Doch zurück zu den Seuchenproduzenten. Ich kann nur hoffen, daß solche EINFÄLLE Einzelfälle bleiben und kein Mensch wegen so einer Dummheit Daten verliert. Diese Biertischidee von Kopierschutz muß aufhören, und die Verantwortlichen gehören meines Erachtens schonungslos gefeuert.

Doch es muß natürlich noch andere geben, denn viele Computerviren sind natürlich nur zum reinen Selbstzweck erstellt.

DIE ANDEREN

Da gibt es also Leute die nichts wichtigeres zu tun haben, als einen neuen Virus zu entwickeln und sich (scheinbar) köstlich amüsieren, wenn andere mit den Viren ihren Ärger haben. Abgesehen davon, daß diese DUMMHEIT nicht mehr zu überbieten ist, ist sie auch noch ausgesprochen KURZSICHTIG. Wer kann heute mit ruhigem Gewissen schon sagen, daß seine Programme keinen Fehler (Nebeneffekt) aufweisen? Wie kann heute ein Viruscode- Erzeuger sicher sein, daß sein Produkt nicht in der nächsten MS-DOS (PC-DOS) Version furchtbares Unheil anrichtet. Außerdem kann der "Spaß" ja nur zur SELBSTBEFRIEDIGUNG dienen, denn den Erfolg, also wenn jemand eine Arbeit von sagen-wir-einem- Jahr verliert, bekommt der Erzeuger der Seuche ja gar nicht mit!!

Nicht viel besser sind Leute, die Seuchen leichtfertig bei Freunden, Bekannten oder anderen Personen einschleusen. Zum einen verstoßen sie gegen das Vertrauen, das ihnen entgegen gebracht wird, zum anderen kann auch ihnen eine Arbeit, in die sie viel Zeit und Mühe investiert haben in Bruchteilen von Sekunden unwiederbringlich zerstört werden, weil sich jemand "einen Spaß machen wollte". Bedenken Sie daher, daß Sie monatelange oder gar jahrelange Arbeiten durch einen "Spaß" gefährden können und helfen sie mit, die Umwelt unserer Computer wieder sauber zu bekommen.

EINIGE RATSCHLÄGE

Um rasch festzustellen, daß sich ein Virus auf ihrer Festplatte befindet (Anwender ohne Festplatte sind, was die Verbreitung betrifft, etwas im Vorteil), sollten sie sie nach Häufigkeit der Datenveränderung schlichten. Dies ist glaube ich, eine der einfachsten Arten, Viren festzustellen. Zum Beispiel:

- 1) Alle Daten in Subdirectories speichern. Ausnahmen: Config.sys, Autoexec.bat und der Einfachheit halber Command.com. Die versteckten Files müssen klarerweise im Rootdirectory bleiben.
- 2) Die Directories müssen nach Häufigkeit geschachtelt werden. Das heißt nach den versteckten Files das System-directory, anschließend das Tool- directory, daran anschließend die Sprachen usw., damit sich ungefähr diese Struktur ergibt:
 SYSTEM <DIR >
 TOOLS <DIR >
 EDITOREN <DIR >
 BATCH <DIR >
 SPRACHEN <DIR >
 WORKING <DIR >
 wobei nur in den Sprachen- und im Working-directory Datenfiles (Textfiles) angelegt werden. Dadurch wird erreicht, daß die Systemfiles ganz nach außen auf die Festplatte geschrieben werden und ihre Position praktisch nie ändern.
- 3) TÄGLICH "Aufräum"-Programme (z.B. SD.EXE von Norton) nach der Arbeit starten und bei der Garbage-collection ZUSEHEN. Sinnvoll natürlich nur bei Programmen, die einem zeigen WO etwas getan wird.

ERGEBNIS: Nach einem arbeitsreichem Tag ohne Kopieren von Programmfiles sollten sich nur im Sprachen- oder/und im Working-dir Veränderungen ergeben. Außerdem verkürzt sich das Aufräumen der Platte enorm. Rote Lampen sollten leuchten wenn im System-dir Umschichtungen stattfinden, außer sie haben selbst dort Files gelöscht oder hinzugefügt.

HINWEIS: SD.EXE von Norton hat die Angewohnheit, nach einer Veränderung im Root-dir die Directories alphabetisch zu ordnen. Entweder man benennt die Directories vorher kurz um oder man überarbeitet mit NU.EXE die Einträge nochmals (nicht für Anfänger). Jedes Schlichtungsprogramm, das gewisse Mindestvoraussetzungen erfüllt (kein ständiges Schlichten der Dir's nach dem Alphabet, Graphische Darstellung der Arbeit), ist für diese Aufgabe geeignet. Das ständige Schlichten hat natürlich noch einen Vorteil. Sollten wirklich einmal ihre FAT's zerstört werden, ist eine Rekonstruktion vielleicht mühevoll aber zu mindest nicht unmöglich.

Aber auch jede andere Methode ist natürlich besser als gar nichts zu tun.

VERSEUCHT

Wenn man einmal einen Virus auf der Festplatte hat: "Keep cool, take it easy and no panic".

Es ist heute schwierig zu sagen, welche Viren von welchen Antivirus- Programmen erkannt werden, da die Programme selbst keine Auskunft darüber geben. Um so mehr ist es wichtig, viele Erkennungsprogramme in Reserve zu haben, wenn auch einige vielleicht schon veraltet sind und einige der verseuchten Programme nicht mehr erkennen. Um jetzt zu wissen, ob ein Programm von einer Seuche befallen ist, ist es gut, möglichst alle Antiseuchen-Programme in den Test mit einzubeziehen. Um auch EXE-Files mehreren Tests zu unterziehen, genügt es, sie einfach in COM-Files umzubenennen. Vergessen sie nur nicht hinterher sie wieder rückzubenennen.

Ein Hinweis: Auch wenn sie ihre Endungen auf CMD, EXC, etc. geändert haben können ihre Programme infiziert sein !!!

Es werden jetzt Antivirusprogramme produziert, die Seuchen nach einem gewissen Schema zu erkennen versuchen. Wann sie allerdings in den Handel kommen, weiß ich leider nicht (vielleicht gibt es sie schon).

Sollten sie vermuten, daß sie Viren in ihren Programmen haben, gilt folgender Rat.

1. Rebooten mit einer geschützten Diskette die sicher virusfrei ist.
2. Kein Programm starten, vom dem sie nicht sicher sein können, daß es seuchenfrei ist. Meist sind, wenn Viren auftreten, die Arbeits-Disketten völlig verseucht.
3. Die Antivirus-Programme von einer geschützten Diskette aus starten. Es sind zwar einige gegen bestimmte Viren immun, d.h. sie können nicht infiziert werden, doch eine generelle Aussage kann natürlich nicht gemacht werden.
4. Keinesfalls sollten als verseucht erkannte Programme gestartet werden.
5. Sollte nun ein Programm als verseucht erkannt werden, mit dem sie das System hochgefahren haben, kein weiteres Programm starten, sondern sofort mit Punkt eins wieder beginnen.
6. Verseuchte Programme entweder sofort löschen oder entseuchen lassen, wobei es vielleicht doch sicherer ist sie völlig zu löschen.

Sollten Sie Probleme mit Seuchen haben, rufen sie den TGM-CLUB oder ihren Betreuer an. Wir helfen ihnen gerne weiter. Wir freuen uns auch, wenn sie uns über ihre eigenen Erfahrungen mit Viren eine Information (vielleicht auch die Tatwaffe) zukommen lassen. Sie könnten damit einem Clubkollegen helfen.

Wichtig ! Clubabende für April und Juni ! Wichtig !

Der nächste Clubabend findet am Mittwoch, den 19. April um 18.00 Uhr im 14. Stock des TGM statt.

Voraussichtliches Thema :

"Die Mailbox".

Herr Illsinger wird einführende Erläuterungen, sowie interessante Neuerungen besprechen.

Geänderte Telefonnummer der Mailbox: 602 21 36

Der darauffolgende Clubabend wird, Änderung vorbehalten, am Donnerstag, 22. Juni mit dem Thema: Netzwerke, ebenfalls um 18.00 Uhr abgehalten werden.

Wir bitten um zahlreiches Erscheinen!

Fragen mit und ohne Antworten

Wer hat Erfahrungen mit dem Umbau eines XT in einen AT ?

Ich bitte um Erfahrungsberichte, Literaturhinweise oder Hinweise und Tips jeder Art.

Oder wer möchte selbst seinen XT in einen AT umbauen? Der kann sich zum Erfahrungsaustausch oder gemeinschaftlichen Einkauf bei mir melden:

Meine Adresse: Robert Smola, Gerschlagergasse 10, 2324 Rannersdorf.

TGM-Opus (2:310/1.0)

TGM_96: SYSINFO.TXT

Operator Werner Illsinger, Sonnwendgasse 38/12, A-1100 Wien

Telefon 43 222 35 23 981 1200/ 2400 Baud
43 222 604 92 48 VOICE

I) Allgemeines

Mein Name ist Werner Illsinger. Ich bin am 18.05.1968 geboren und bin der Systembetreuer dieser Mailbox. Ich werde für diese Arbeit nicht bezahlt. Die Zeit, die ich vor dieser Mailbox verbringe, entspricht etwa meiner Arbeitszeit in der Firma (ca. 20 Stunden). Nebenbei studiere ich noch Informatik an der TU-Wien. Meine weiteren Hobbys sind Briefmarkensammeln und Modellbau, für die mir allerdings keine Zeit mehr bleibt, seitdem ich diese Box betreibe.

Dieses System soll primär zum Austausch von Mitteilungen und Files (Dateien) dienen. Das System ist 24 Stunden am Tag on-Line. Es kann jedoch vorkommen, daß durch irgendeinen Fehler das System abstürzt. Da ich untertags arbeiten gehe und nebenbei studiere, kann es mitunter recht lange dauern, bis ich bemerke, daß die Box nicht abhebt. Ich bitte daher um Verständnis, daß es zu Systemausfällen kommen kann, die einen Tag dauern. Seien Sie nicht ungehalten, wenn das System nicht abhebt. Es kann auch sein, daß gerade Wartungsarbeiten durchgeführt werden. Versuchen Sie dann einige Minuten (10 bis 20 Minuten) später nochmals anzurufen.

II) Aufbau des Systems

Das System ist in 2 Bereiche unterteilt. Der eine Bereich heißt Message Section und dient dazu, Mitteilungen zu lesen und zu schreiben. Der TGM Opus ist mit dem internationalen FidoNetzwerk verbunden. Dieses Netzwerk umfaßt derzeit ca. 2500 Rechner weltweit. In Österreich gibt es derzeit ca. 10 Nodes (Netzknoten) in Wien und Graz. Dieses Netzwerk erlaubt es dem Benutzer dieses BBS, Nachrichten in die Ganze Welt zu verschicken. Zu diesem Zweck gibt es verschiedene Areas in der Message Section.

1) MESSAGE SECTION

Dieser Teil der Mailbox dient vorwiegend dazu, Mitteilungen mit anderen Benutzern von FidoNet und diesem BBS auszutauschen.

Es gibt folgende Arten von Areas:

LOCAL AREA: Ein lokales Area ist ein Area, in dem es nur Mitteilungen gibt, die in diesem BBS geschrieben worden sind. Areas von diesem Typ sind die Areas 1 und 2.

FIDONET NETMAIL AREA: Von diesem Typ gibt es nur ein Area in der Box. Dieses Area dient dazu, gezielt eine Message von dieser Box an eine andere zu senden. Das Versenden von privaten Mitteilungen kostet Geld. Falls Sie gerne Mitteilungen in diesem Area versenden würden, schicken Sie eine Mitteilung an den SYSOP.

NATIONALE KONFERENZEN: Mitteilungen, die in diesen Areas eingegeben werden, werden automatisch an alle anderen angeschlossenen Boxen weitergeschickt. Die Sprache in diesen Areas ist Deutsch, da nur österreichische Boxen an diesen Konferenzen teilnehmen. Die Teilnahme an diesen Konferenzen ist derzeit kostenlos, da der Import und Export der Mitteilungen jedoch über Telefonleitungen vorgenommen wird, kostet der Betrieb dieser Konferenzen Geld. Wenn jemand zum Betrieb dieser Konferenzen beitragen will, bitte ich Sie einen Betrag Ihres Ermessens an: Werner Illsinger, PSK Kto. Nr. 7.918.896, mit Zahlungszweck SPENDE ECHOMAIL

zu überweisen.

Derzeit angebotene nationale Konferenzen sind:

Area Name	Beschreibung
9 [BAZAR]	Tausche Verkäufe (Gratisservice)
10 [VIE_CHAT]	Wiener Tratsch (gemütliche Plaudereien)
12 [MSDOS]	Diskussion über IBM PSs / PS/2
13 [TECHNICS]	Diskussion jeglicher technischer Probleme
17 [GAMES]	Lösungen zu allseits beliebten Spielen
21 [NOFUN]	Witze, damit die anderen auch was zu lachen haben.

In diesen Areas sind keine privaten Mitteilungen erlaubt. Verwenden Sie für private Mitteilungen bitte das Netmail Area.

INTERNATIONALE KONFERENZEN: Mitteilungen, die in diesen Areas eingegeben werden, werden automatisch an alle anderen Boxen weitergeschickt. Die Sprache in diesen Areas ist Englisch. Die Teilnahme an diesen Konferenzen ist kostenlos. Ich bitte Sie jedoch ebenfalls zum

Erhalt dieser Konferenz beizutragen, indem Sie einen kleinen Betrag auf das obengenannte Konto überweisen.

Derzeit angebotene internationale Konferenzen sind:

Area Name	Beschreibung
11 [CLONE]	IBM-CLONES (siehe auch MSDOS Konferenz)
18 [COMMS]	Communication-Problems
19 [DBASE]	DBASE-Users
20 [PGMRS]	Programmers

In diesen Konferenzen sind private Mitteilungen ebenfalls nicht erlaubt. Bitte verwenden Sie das Netmail Area.

2) FILE SECTION

Der zweite Teil der Mailbox dient dazu, Dateien auszutauschen. Dateien sind Programme, Textfiles usw. Um Platz und Übertragungszeit zu sparen, werden Dateien mit einem speziellen Utility komprimiert. Dieser Vorgang wird ARcEn genannt. Dateien, die mit diesem Utility gearced wurden, erkennt man an der Erweiterung '.ARC'. Wenn Sie so eine Datei downloaden wollen, müssen Sie zuerst das Entkomprimier-Utility downloaden. Dieses Programm finden Sie in der IBM-PC-Section unter dem Namen PK361.EXE. Falls Sie für andere Benutzer Programme in der Mailbox anbieten wollen, ARcEn Sie die Programme bitte vor dem Upload. Es erspart Ihnen Zeit und mir Platz auf der Festplatte.

Es gibt ein Gesetz im FidoNet: FidoNet oder eine seiner Boxen darf nicht dazu benützt werden, geklaute, raubkopierte Software zu vertreiben. Bitte schicken Sie also keine Programme an die Mailbox, von denen Sie nicht genau wissen, ob sie Public Domain sind, oder nicht. Wenn Sie dennoch ein raubkopiertes Programm an die Mailbox senden, sehe ich mich gezwungen, Sie als User zu löschen. Und Sie werden in keiner anderen FidoNet-Box als User willkommen sein.

III) IDEOLOGIE HINTER FIDONET

FidoNet lebt ausschließlich von seinen Usern. Die gesamte Software, die für die Aufrechterhaltung dieses Netzes dient, ist Shareware und von Idealisten in ihrer Freizeit geschrieben worden. Die meisten Teile der Software kosten keinen Schilling. Die Autoren bitten nur darum, daß man, falls einem die Software gefällt einen geringen Betrag (ca. 10 Dollar) an sie überweist. Alle Sysops im FidoNet sind unbezahlte Amateure, die sich die Nächte damit um die Ohren schlagen, warum denn der Batch-Job schon wieder an der falschen Stelle das Falsche tut. Die Mailbox besteht nicht aus einem, sondern aus sehr vielen einzelnen, inner-einergreifenden Programmen.

Ich glaube daher, daß es als Benutzer nicht richtig ist, an die Box Anforderungen zu stellen. Jedes nicht kommerzielle BBS ist das, was die User daraus machen. Man sollte daher nicht sagen: 'Woanders gibt es was Besseres' oder 'Die Radio Austria' hat aber noch einen TELEX Dienst. Sondern 'Ich schreib' hier was rein, was die anderen auch interessiert' oder 'Ich red' mal mit dem Sysop, vielleicht könnte man da auch noch was verbessern'. Nach einem schönen Wiener Sprichwort heißt es: 'Von nix kummt nix', und so sollte man auch FidoNet sehen. Ich bin gerne jederzeit bereit, jemandem weiterzuhelfen oder irgend ein Stück Software hier einzubauen, damit die Box noch besser wird. Aber die Inhalte können nur von den Benutzern gestaltet werden.

Wien, den 4. November 1988

Werner Illsinger, Sysop of 310/1

CA, die Bank zum Erfolg.



Demmer & Merlicok

Der Kredit zum Erfolg. Am 19. Geburtstag wird vieles anders. Die CA-Card wird zur Scheckkarte mit Bankomat-Möglichkeit.

Und bei uns sind Sie wahlberechtigt: es gibt günstigen Kredit – wofür, das entscheiden Sie ganz allein.



CREDITANSTALT