

# Das erste

# »Strubs« Listing

Ich war so begeistert, daß ich mich gleich hingesezt und ein Programm geschrieben hatte, um die Fähigkeiten von Strubs zu testen, und ich muß sagen, Strubs ist ein Programm, das hält, was es verspricht.

Das von mir mit Strubs entwickelte Programm »Menü« liegt als Quell- und Objektprogramm vor. Der Benutzer kann damit seine Programme nur mit Hilfe der Funktionstasten und der Returnntaste auswählen und starten, so daß die umständliche Ladeprozedur per Hand entfällt.

Will man mit »Menü« arbeiten, trägt man zunächst in Data-Zeilen ab Zeilennummer 10 den Text, der auf dem Bildschirm erscheinen soll (maximal 34 Zeichen ein und dann in der gleichen Zeile — durch Komma getrennt — den genauen Namen des Programms, das geladen werden soll. Die Verwendung des Jokers (\*) zur Abkürzung ist dabei auch möglich. Anschließend speichert man Menü ab und kann es starten.

»Menü« faßt immer 10 Programme auf einer Bildschirmseite zusammen und blättert — je nachdem man mit dem Pfeil, der auf die Programme zeigt, nach unten oder nach oben aus dem Bildschirm heraus will — weiter oder zurück.

Bei Programmen, die größer sind als »Menü« ist darauf zu achten, daß in der ersten bearbeiteten Programmzeile eine Routine steht, die die Pointer auf das Basic-Ende ausrichtet (siehe dazu Zeile 1 von »Menü«).

Ersetzt man in den Programmen, die durch »Menü« geladen werden sollen noch die END-Anweisung — oder das Programmende, falls keine END vorhanden — durch LOAD"MENU",8, so wird nach Beendigung des gewählten Programms automatisch wieder »Menü« geladen.

Vorsicht ist geboten bei Programmen, die den BASIC-Anfang oder das RAM-Ende verschieben, da diese Programme zum Systemabsturz führen, wenn man die entsprechenden Vektoren nicht zurücksetzt.

»Menü« müßte auch auf dem C 64 laufen, ich konnte es aber leider nicht testen.

(Gerd Sombetzki)

Unser Kurs über den Precompiler »Strubs« ist erst in dieser Ausgabe beendet worden. Und doch hat uns schon jetzt das erste mit »Strubs« erstellte Listing erreicht.

Folgende Änderungen nahm ich für meinen VC 20 mit 40/80 Zeichenkarte, 32 K-RAM, Floppy 1541 und VC 1526 vor. Zunächst verlegte ich die Maschinenroutine in den Kassettenspeicher. Zeile: 45610 POKE 828 + i, w: ...usw.

Der Sprung zur Interpreterroutine \$A7E7 = #167 liegt beim VC 20 bei \$C7E7 = #199. Außerdem fehlte das im Assemblerlisting vorhandene PLP = \$28 = #40.

D.h.: Zeile 45620 DATA32, 115,0,8,201,33,240,4,40,76,231, 199,40

45630 DATA169,18,133,44,169,138,76,231,199,999

Verschieben der Initialisierungsroutine:

45650 FORI = 0 TO 10: READ W: POKE 850 + I, W

45670 SYS 850

45680 DATA 169, 60, 141, 8, 3, 169, 3, 141, 9, 3, 96

Die Änderung des Wertes 8 in 18 in der Zeile 45630 ist notwendig, da der Basic-Anfang beim VC-20 mit Erweiterung) = 8K bei \$1200 beginnt, also Highbyte \$12 = #18.

```

0 '==== DIE ZEILE 1 RICHTET DIE POINTER AUF DAS BASIC-ENDE AUS. =====
1 POKE 45,PEEK(174) : POKE 46, PEEK(175) : CLR : GOTO EHPTRG
2 '..... AB ZEILE 10 STEHEN DIE MENUEDATEN .....
5 REM"DATAZEILEN_ENTHALTEN_<PROGRAMMNAME>,<LADENAME>
98 DATA "ENDE....."
100 DATA @,@
200 '
1000 £MENUE
1030 :
1040 :
1050 : !IF AN/10-INT(AN/10) = 0 THEN
1060 : M'ASKEN'D'URCHLAEUFE' = AN/10-1
1070 : !ELSE
1080 : MD = INT(AN/10)
1090 : !FI
1100 : .....HAUPTSCHLEIFE.....
1110 : M'ASKENZAehler' = 0
1120 : !LOOP
1130 : GOSUB £MASKE
1140 : T'ASTENZAehler' = 1
1150 : !LOOP
1160 : GOSUB £PFEIL : GOSUB £TASTE
1170 : !CASEOF TA'STE' = 0 THEN
1180 : !EXIT
1190 : !OF TA'STE' = -1 THEN
1200 : T = T-1 : IF T < 1 THEN !EXIT
1210 : !OF TA'STE' = 1 THEN
1220 : T = T+1 : IF T > 10 OR T+M*10 > AN THEN !EXIT
1230 : !ECASE
1240 : !LOOP
1250 : !CASEOF TA'STE' = 0 THEN
1260 : !EXIT
1270 : !OF TA'STE' = -1 THEN
1280 : M = M-1 : IF M < 0 THEN M = 0
1290 : !OF TA'STE' = 1 THEN
1300 : M = M+1 : IF M > MD THEN M = MD
1310 : !ECASE
1320 : !LOOP
1325 : .....HAUPTSCHLEIFE ENDE ..
1330 GOTO £LADEN
1340 '
1500 £MASKE
1510 :
1520 : GOSUB £ZEILE1
1530 : I = M*10+1 ' M IST MASKENSEITENZAehler
1540 : !WHILE I <= M*10+10 AND I <= AN'ZAHL' !DO
1550 : PRINT : PRINT TAB(5) P'RG'N'AME'$(I)
1560 : I = I+1
1570 : !EWHILE
1580 GOSUB £HELP
1590 '
1700 £ZEILE1
1710 : PRINT CHR$(147)TAB(12)"PROGRAMMAUSWAHL"
1720 : PRINT TAB(12)"=====
1730 RETURN
1740 '
1800 £HELP
1810 :
1820 : Z = 24 : S = 2 : GOSUB £CRSRPO.
1823 : PRINT RO$"F1 => OBEN"RF$ " ;
1826 : PRINT RO$"F3 => UNTEN"RF$ " ;
1830 RETURN
1840 '
1900 £CRSRPO.
1910 :
1920 RETURN
1930 '
2000 £PFEIL
2005 :
2010 : Q IST DER VEKTOR ZUM LOESCHEN ODER SETZEN DES PFEILS
2020 :
2030 : Z = T*2+2 : S = 2 : GOSUB £CRSRPO.
2040 : IF Q = 1 THEN £LBL1
2050 : PRINT "=>" : GOTO £LBL2
2060 : £LBL1
2070 : PRINT " " : Q = 0
2072 '
2200 £TASTE
2210 :
2220 : !LOOP
2230 : GET TA'STE'$ : IF TA'STE'$ = "" THEN £THIS
2240 : !CASEOF TA'STE'$ = CHR$(133) THEN 'F1-TASTE GEDR.
2250 : Q = 1 : GOSUB £PFEIL
2260 : TA'STE' = -1 : !EXIT
2270 : !OF TA'STE'$ = CHR$(13) THEN 'RETURN GEDR.
TA'STE' = 0 : !EXIT

```

Strubs-Listing »Menü«  
(Anfang)

Strubs-Listing »Menü«  
(Schluß)

```

2280 :
2290 :
2300 :
2310 :
2320 :
2330 RETURN
2332 :
2400 £LADEN
2410 :
2420 :
2430 :
2440 :
2450 :
2460 :
2470 :
2490 'ENDE
2495 :
2700 £SCHLUSS
2710 :
2720 :
2730 :
2740 :
2750 :
2760 :
2770 :
2780 :
2790 :
2800 :
2810 :
2820 :
2830 :
2840 :
2850 :
2860 :
2870 :
2880 :
2890 :
2900 :
2910 :
2920 :
2930 :
2940 :
10000 £HPTPRG
10010 :
10012 REM" ***** HAUPTPROGRAMM *****
10014 REM" _____MENUE_VERSION_VOM_01.05.84_____
10016 REM" _____GERD_____SOMBETZKI_____
10018 REM" _____FUHRMANNSTR.____47_____
10020 REM" _____4600_DORTMUND_13_____
10022 : _____TEL.:_0231/213656_____
10030 :
10040 :
10050 :
10060 :
10070 :
10100 £INIT
10130 :
10140 :
10150 :
10160 :
10170 :
10180 :
10190 :
10200 :
10210 :
10220 :
10230 :
10235 :
10240 :
10250 :
10260 :
10290 RETURN
10295 :
10300 £ANZAHL
10310 :
10320 :
10330 :
10340 :
10350 :
10360 RETURN
10370 :
READY.

!OF TA'STE' = CHR$(134) THEN 'F3-TASTE GEDR.
Q = 1 : GOSUB £PFEIL
TA'STE' = 1 : !EXIT
!ELOOP

NUMMER' = M'ASKENZAEBLER' * 10 + T'ASTENZAEBLER
Z'EILE' = T*2+2 : S'PALTE' = 6 : GOSUB £CRSRPO.
PRINT RO$ P'RG'N'AME'$(NU) RF$
IF LEFT$(P'RG'N'AME'$(NU),4) = "ENDE" THEN £SCHLUSS
(NOCH ZU IMPLEMENTIERENDE FUNKTION: LADEN VON MPGS)
LOAD L'ADE'N'AME'$(NU),8

PRINT CHR$(147);
PRINT LEFT$(CD$,10) TAB(5);
PRINT RO$ " F5 => MENUE "
PRINT TAB(5) RO$ " F7 => ANDERE DISK "
PRINT TAB(5) RO$ "RETURN => ENDE "
!LOOP
GET TA'STE' : IF TA'STE' = "" THEN £THIS
!CASEOF TA'STE' = CHR$(135) THEN 'F5-TASTE GEDR.
RUN10000
TA'STE' = CHR$(136) THEN 'F7-TASTE GEDR.
PRINT : PRINT TAB(5) "BITTE ANDERE DISK EINLEGEN,"
GET TA'STE' : IF TA'STE' = "" THEN £THIS
OPEN15,8,15,"I" : CLOSE15
LOAD "MENUE",8
!OF TA'STE' = CHR$(13) THEN 'RETURN GEDR.
PRINT CHR$(147)
PRINT "BITTE DISKETTE DEM LAUFWERK ENTNEHMEN,"
PRINT "DANN LAUFWERK UND RECHNER ABSCHALTEN."
END
!ELOOP

*****
GOSUB £INIT
GOTO £MENUE

*****
GOSUB £ANZAHL : IF AN'ZAHL' = 0 THEN £SCHLUSS
DIM P'RG'N'AME'$(AN'ZAHL'), L'ADE'N'AME'$(AN'ZAHL')
RESTORE
FOR I = 1 TO AN'ZAHL
READ P'RG'N'AME'$(I), L'ADE'N'AME'$(I)
NEXT

-----GLOBALVARIABLEDEFINITION-----
CD$ = " " 'CURSOR DOWN' : RO$ = " " 'RVS ON
CR$ = " " 'CURSOR RIGHT' : RF$ = " " 'RVS OFF
FOR I = 1 TO 7
CD$ = CD$+CD$ : CR$ = CR$+CR$
NEXT

RESTORE : AN'ZAHL' = -1
!REPEAT
READ P'RG'N'AME'$, L'ADE'N'AME'$
AN'ZAHL' = AN'ZAHL' + 1
!UNTIL P'RG'N'AME'$ = "e"

```

```

1 POKE45,PEEK(174):POKE46,PEEK(175):CLR:GOTO10000
5 REM"DATAZEILEN_ENTHALTEN_<PROGRAMMNAME>,<LADENAME>
98 DATA"ENDE.....","....."
100 DATA,e
1000 :
1030 IFNOT(AN/10-INT(AN/10)=0)THEN1051
1040 MD=AN/10-1
1050 GOTO1070
1051 :
1060 MD=INT(AN/10)
1070 :
1090 M=0
1100 :
1110 GOSUB1500
1120 T=1
1130 :
1140 GOSUB2000:GOSUB2200
1150 IFNOT(TA=0)THEN1171
1160 GOTO1241
1170 GOTO1230
1171 IFNOT(TA=-1)THEN1191
1180 T=T-1:IFT(1)THEN1241
1190 GOTO1230
1191 IFNOT(TA=1)THEN1230
1200 T=T+1:IFT(100RT+M*10)ANTHEN1241
1230 :
1240 GOTO1130
1241 :
1250 IFNOT(TA=0)THEN1271
1260 GOTO1321
1270 GOTO1310
1271 IFNOT(TA=-1)THEN1291
1280 M=M-1:IFM<0THENM=0
1290 GOTO1310
1291 IFNOT(TA=1)THEN1310
1300 M=M+1:IFM>MDTHENM=MD
1310 :
1320 GOTO1100
1321 :
1330 GOTO2400
1500 :
1510 GOSUB1700
1520 I=M*10+1
1530 IFNOT(I<M*10+10ANDI(=AN)THEN1561
1540 PRINT:PRINTTAB(5)PN$(I)
1550 I=I+1
1560 GOTO1530
1561 :
1570 GOSUB1800
1580 RETURN
1700 :
1710 PRINTCHR$(147)TAB(12)"PROGRAMMAUSWAHL"
1720 PRINTTAB(12)"=====
1730 RETURN
1800 :
1810 Z=24:S=2:GOSUB1900
1820 PRINTRO$"F1 => OBEN"RF$ " ;
1823 PRINTRO$"F3 => UNTEN"RF$ " ;
1826 PRINTRO$"RETURN => LOAD"RF$;
1830 RETURN
1900 :
1910 PRINTCHR$(19)LEFT$(CD$,Z-1)LEFT$(CR$,S-1);
1920 RETURN
2000 :
2020 Z=T*2+2:S=2:GOSUB1900
2030 IFQ=1THEN2050
2040 PRINT"==" :GOTO2060
2050 PRINT" " :Q=0
2060 :
2070 RETURN
2200 :
2210 :
2220 GETTA$:IFTA$=""THEN2220
2230 IFNOT(TA$=CHR$(133))THEN2261
2240 Q=1:GOSUB2000

```

```

2250 TA=-1:GOTO2321
2260 GOTO2310
2261 IFNOT(TA$=CHR$(13))THEN2281
2270 TA=0:GOTO2321
2280 GOTO2310
2281 IFNOT(TA$=CHR$(134))THEN2310
2290 Q=1:GOSUB2000
2300 TA=1:GOTO2321
2310 :
2320 GOTO2210
2321 :
2330 RETURN
2400 :
2410 NU=M*10+T
2420 Z=T*2+2:S=6:GOSUB1900
2430 PRINTRO$PN$(NU)RF$
2440 ILEFT$(PN$(NU),4)="ENDE"THEN2700
2470 LOADLN$(NU),8
2700 :
2710 PRINTCHR$(147);
2720 PRINTLEFT$(CD$,10)TAB(5);
2730 PRINTRO$" F5 => MENUE "
2740 PRINTTAB(5)RO$" F7 => ANDERE DISK "
2750 PRINTTAB(5)RO$"RETURN => ENDE "
2760 :
2770 GETTA$:IFTA$=""THEN2770
2780 IFNOT(TA$=CHR$(135))THEN2801
2790 RUN10000
2800 GOTO2920
2801 IFNOT(TA$=CHR$(136))THEN2861
2810 PRINT:PRINTTAB(5)"BITTE ANDERE DISK EINLEGEN,
2820 PRINTTAB(5)"DANN BEL. TASTE DRUECKEN!"
2830 GETTA$:IFTA$=""THEN2830
2840 OPEN15,8,15,"I":CLOSE15
2850 LOAD"MENUE",8
2860 GOTO2920
2861 IFNOT(TA$=CHR$(13))THEN2920
2870 PRINTCHR$(147)
2880 PRINT"BITTE DISKETTE DEM LAUFWERK ENTNEHMEN,"
2890 PRINT"DANN LAUFWERK UND RECHNER ABSCHALTEN."
2900 PRINT:PRINT"BY, BY!"
2910 END
2920 :
2930 GOTO2760
2931 :
10000 :
10012 REM"_____MENUE__VERSION__VOM__01.05.84__
10014 REM"_____GERD_____SOMBETZKI_____
10016 REM"_____FUHRMANNSTR.____47_____
10018 REM"_____4600_DORTMUND__13_____
10020 REM"_____TEL.:_0231/213656_____
10030 GOSUB10100
10040 GOTO1000
10100 :
10130 GOSUB10300:IFAN=0THEN2700
10140 DIMP$(AN),LN$(AN)
10150 RESTORE
10160 FORI=1TOAN
10170 READPN$(I),LN$(I)
10180 NEXT
10220 CD$="":RO$=""
10230 CR$="":RF$=""
10240 FORI=1TO7
10250 CD$=CD$+CD$:CR$=CR$+CR$
10260 NEXT
10290 RETURN
10300 :
10310 RESTORE:AN=-1
10320 :
10330 READPN$,LN$
10340 AN=AN+1
10350 IFNOT(PN$="0")THEN10320
10360 RETURN

```

READY.

So würde das Programm ohne den  
Precompiler aussehen.