Der Monitor zum Ascompiler

Dieser Maschinensprache-Monitor ist die ideale Ergänzung zum Ascompiler 64 aus der letzten Ausgabe: Speichern, Laden und Ändern von übersetzten Programmen wird damit zum Kinderspiel.

Dieser Monitor wurde speziell für den Ascompiler 64 (Listing des Monats in der Ausgabe 1/86) entwickelt. Der Ascompiler ist ein einfacher, aber extrem schneller Übersetzer für ein sogenanntes »Tiny Basic« (ein Integer-Basic mit eingeschränktem Befehlssatz).

Natürlich ist der hier vorgestellte Monitor vollständig im Ascompiler-Format geschrieben. Weil der Compiler nur einen sehr eingeschränkten Basic-Befehlssatz verarbeitet, wurde die Programmierung auch etwas umständlich. Als normales Basic-Programm läuft der Monitor nur bedingt, da einige Maschinenroutinen aufgerufen werden, die von Basic aus zu langsam bearbeitet werden. Es ist daher auf jeden Fall eine Übersetzung mit dem Ascompiler 64 notwendig. Ist dies geschehen, dann läuft das Programm einwandfrei und kann mit jedem in Assembler geschriebenen Maschinenmonitor mithalten, und das sowohl hinsichtlich der Geschwindigkeit als der Bedienungsfreundlichkeit. Wie bei jedem professionellen Monitor üblich, sind generell alle Adressenangaben hexadezimal. Der »SPITZMON 64« genannte Monitor hat folgenden Befehlssatz:

Memory Dump: »M XXXX YYYY«

Hexdump des Speicherbereichs von $XXXX bis $YYYY. Der ausgegebene Memory-Dump kann natürlich beliebig editiert werden. Dazu fährt man mit dem Cursor auf die zu ändernde Speicherstelle und überschreibt diese. Nach RETURN werden alle 8 Byte einer Zeile neu in den Speicher geschrieben.

Exit: »X«

Dieser Befehl beendet das Programm und führt zurück ins Basic.

Save: »S"Name" AA XXXX YYYY«

Dieser Befehl speichert den Bereich von $XXXX bis $YYYY als Maschinenprogramm »Name« auf Datasette oder Diskette. Die genaue Bedeutung der einzelnen Parameter:

$AA = Geräteadresse
(AA=01: Datasette, AA=08: Diskettenlaufwerk).
$XXXX = Programmanfang
$YYYY = Programmende plus eins

Beispiel: Wenn Sie das Monitor-Programm (Listing) genau abtippen und mit dem Ascompiler nach $4000 compilieren, so liegt das Maschinenprogramm zwischen $4000 und $5520. Sie können nun den Spitzmon von Basic aus mit »SYS 4*4096« (4*4096=$4000) starten und ihn als erstes sich selbst als Maschinenprogramm für zukünftige Verwendung speichern lassen. Dazu geben Sie ein:

»S"SPITZMON64.$4000" 08 4000 5521«. Falls Sie mit Datasette arbeiten, müssen Sie die »08« natürlich in »01« ändern.

Load: »L"Name" AA«

Dieser Befehl lädt ein Maschinenprogramm an seine Originaladresse.

$AA ist wieder die Geräteadresse (AA=01: Datasette, AA=08 Diskettenlaufwerk)

Go: »G XXXX«

Ein Maschinenprogramm mit der Startadresse $XXXX wird gestartet. Mit diesem Befehl muß sehr vorsichtig umgegangen werden, denn wenn an der angegebenen Adresse kein sinnvolles Maschinenprogramm steht, dann führt das in der Regel zum »Absturz« des Computers.

Fill: »F XXXX YYYY AA«

Dieser Befehl füllt den Speicherbereich von $XXXX bis $YYYY mit dem Byte $AA.

Transfer: »T XXXX YYYY ZZZZ«

Transfer verschiebt den Speicherbereich von $XXXX bis $YYYY nach $ZZZZ.

Compare: »C XXXX YYYY ZZZZ«

Dieser Befehl überprüft, ob sich ab $ZZZZ die gleichen Daten im Speicher befinden, wie zwischen $XXXX und $YYYY. Treten Unterschiede auf, so werden diese Adressen in hexadezimaler Schreibweise ausgegeben.

Hunt: »H XXXX YYYY AA BB CC«
oder: **»H XXXX YYYY AA BB«**
oder: **»H XXXX YYYY AA«**

Der Hunt-Befehl durchsucht den Speicherbereich von $XXXX bis $YYYY nach maximal 3 Bytes. Findet das Programm die gesuchten Bytes, so werden die Adressen des jeweils ersten Bytes (»AA«) in Hex-Schreibweise ausgegeben.

Beispiel: »H E000 FFFF 20 D2 00« durchsucht den Speicherbereich zwischen $E000 und $FFFF nach dem Auftreten der Bytefolge $20, $D2, $00.

Vom Umgang mit dem Spitzmon

Für den Maschinensprache-Unkundigen ist in erster Linie die Funktion »Save« interessant, denn dadurch wird es möglich, mit dem Ascompiler übersetzte Programme direkt als Maschinenprogramme zu speichern. Vorausgesetzt, Sie haben ein Programm für den Ascompiler 64 geschrieben und natürlich den Ascompiler selbst und eine bereits übersetzte Version des Spitzmon zur Hand, dann funktioniert das Arbeiten mit Ascompiler und Spitzmon folgendermaßen:

— Ascompiler 64 laden und starten
— Nach »Ready«-Meldung das eigene Tiny-Basic-Programm eintippen oder laden
— Mit »SYS 49152« den Ascompiler 64 aktivieren.
— Startadresse des zu erzeugenden Maschinenprogramms angeben — Der Ascompiler übersetzt nun das Tiny-Basic-Programm im Speicher und gibt anschließend die Anfangs- und Endadresse des erzeugten Maschinenprogramms an. Diese beiden Adressen aufschreiben
— Eine bereits übersetzte Version des Spitzmon mit »LOAD "Name",8,1« (»..,1,1« bei Datasette) laden und mit SYS starten, wobei hinter dem SYS-Befehl die dezimale Startadresse des Spitzmon angegeben werden muß (Beispiel: Der Spitzmon wurde ab Adresse $8000 compiliert, dann lautet der SYS-Befehl: »SYS 8*4096«)
— Sie sind jetzt im Spitzmon 64 und können das vorhin übersetzte Tiny-Basic-Programm mit »S"Name" 08 XXXX YYYY« auf Diskette speichern, wobei Sie für XXXX und YYYY die vorhin notierte Anfangs- und Endadresse des übersetzten Programms angeben. Bei der Arbeit mit dem Spitzmon ist darauf zu achten, daß alle Eingaben in hexadezimaler Schreibweise gemacht werden. Das »$«-Zeichen wird aber natürlich nicht eingegeben.

Der Spitzmon überprüft jede Eingabe des Benutzers. Mehrere Zahlenangaben müssen durch eine Leerstelle (Blank) voneinander getrennt werden. Natürlich sind für die Zahleneingabe nur die Ziffern 1 bis 9 und die Buchstaben A bis F erlaubt. Werden diese Bedingungen nicht erfüllt, so erscheint als Fehlermeldung ein Fragezeichen hinter der Fehlerstelle. Der Befehl wird dabei natürlich abgebrochen.

Hinweise zum Abtippen

Wenn Sie den Ascompiler 64 aus der Ausgabe 1/86 nicht besitzen, dann ist es sinnlos, sich die Mühe des Abtippens zu machen.

Bevor Sie darangehen, den Spitzmon 64 (Listing) abzutippen, sollten Sie als erstes den Ascompiler laden und starten, um den Spitzmon nach der Eingabe unverzüglich compilieren zu können. Beim abgedruckten Listing des Spitzmon wurde auf Prüfsummen verzichtet, da es beim Zusammenwirken von Checksummer und Ascompiler 64 möglicherweise zu Problemen kommen kann. Steuer- und Grafikzeichen wurden allerdings in der bewährten Klartext-Form ausgegeben.

Auf folgende Punkte sollten Sie besonders achten:

— Verändern oder vereinfachen Sie keine Zeilen oder Variablen.

— Mehrere Befehle in einer Zeile sind unzulässig.

— Lesen Sie nochmals genau die Anleitung für den Ascompiler 64 (Ausgabe 1/86) durch.

Der Spitzmon 64 kann natürlich noch beliebig erweitert werden (zum Beispiel Dissasembler etc.). Es ist aber darauf zu achten, daß nur die Variablennamen von A-Z zur Verfügung stehen. Um das Maschinenprogramm so kurz wie möglich zu halten, sollte man von Multiplikation und Divisionen wenig Gebrauch machen. Die Anweisung »LET K=J*2« benötigt in der Übersetzung mehr als 70 Byte (!). In diesem Fall könnte man das gleiche Ergebnis durch Addition erreichen (»LET K=J+J). Im Spitzmon 64 wurden für Multiplikation und Division spezielle Unterprogramme eingerichtet: (12310, 13290, 13310, 15020). Durch diese »Rationalisierung« konnte das gleiche Programm um mehr als 1 KByte verkürzt werden (!).

Erweiterungen sind möglich

Ein letztes Wort wäre noch zur Compilierung zu sagen: Sie können den Spitzmon beliebig irgendwo im Basic-Speicherbereich ansiedeln. Es ist empfehlenswert, sich verschiedene Versionen des Spitzmon 64 für unterschiedliche Adressenbereiche zu erzeugen, um dann jeweils die geeignete Version auswählen zu können. Haben Sie beispielsweise ein Tiny-Basic-Programm geschrieben, das mit dem Ascompiler 64 übersetzt im Adressenbereich ab $3000 liegt, dann können Sie, um dieses Programm zu speichern, natürlich nicht eine Spitzmon-Version verwenden, die im gleichen Adressenbereich liegt. Es ist also sinnvoll, sich verschiedene Versionen für unterschiedliche Adreßbereiche herzustellen und mit dem Ascompiler 64 auch gar kein Problem.

(Christian Quirin Spitzner/ev)

```
100 REM*********************************
110 REM*                               *
120 REM*   SPITZMON 64 FUER ASCOMP 64  *
130 REM*   CHRISTIAN Q. SPITZNER       *
140 REM*   GRUBERSTR. 53               *
150 REM*   8011 POING                  *
170 REM*                               *
180 REM*********************************
190 REM
1000 REM ***        HAUPTPROGRAMM      ***
1010 PRINT"(CLR,GRAPHIC)*****************
     **********************(5SPACE)F*****
     ******S"
1020 PRINT"(LEFT)**(5SPACE)_SPITZMON 64_(2
     SPACE)FUER(2SPACE)ASCOMP 64(3SPACE)**
     (5SPACE)F**********X"
1030 PRINT"(LEFT)**(DOWN,2LEFT)**(6SPACE)C
     OPYRIGHT (C) POING 1985(8SPACE)**(DOW
     N,2LEFT)*";
1040 PRINT"*(6SPACE)BY CHRISTIAN QUIRIN SP
     ITZNER(4SPACE)*";
1050 PRINT"*(DOWN,2LEFT)*****************
     **********************"
1060 POKE 204,0
1070 POKE 650,128
1080 PRINT">";
1090 LET F=0
1100 SYS 42336
1110 LET Z = 512
1120 LET V = PEEK(512)
1130 IF V=0 THEN 1080
1140 IF V=62 THEN 1160
1150 GOTO 1220
1160 LET X=Z+1
1170 LET C=PEEK(X)
1180 POKE Z,C
1190 IF C=0 THEN 1110
1200 LET Z=Z+1
1210 GOTO 1160
1220 IF V=77 THEN 3020
1230 IF V=71 THEN 6020
1240 IF V=88 THEN 4020
1250 IF V=76 THEN 5020
1260 IF V=83 THEN 5020
1270 IF V=70 THEN 7020
1280 IF V=84 THEN 8020
1290 IF V=72 THEN 9020
1300 IF V=67 THEN 8020
2000 REM
2010 REM ***        ZEILENEINGABE      ***
2020 LET J=0
2030 LET K=J+J
2040 LET K=K+J
2050 LET T=4+K
2060 GOSUB 10020
2070 IF F=1 THEN 1080
2080 LET J=J+1
2090 IF J<8 THEN 2030
2100 GOSUB 11020
2110 IF C=28 THEN 2130
2120 GOTO 14020
2130 LET S=1
2140 LET A=1
2150 LET G=0
2160 LET T=4
2170 GOSUB 12020
2180 IF F=1 THEN 1080
2190 LET D=G
2200 LET J=0
2210 LET K=J+J
2220 LET K=K+J
2230 LET A=6+K
2240 LET S=3
2250 LET G=0
2260 LET T=A+1
2270 GOSUB 12020
2280 IF F=1 THEN 1080
2290 LET E=D+J
2300 POKE E,G
2310 LET J=J+1
2320 IF J<8 THEN 2210
2330 GOTO 1080
3000 REM
3010 REM ***        MEMORY-DUMP        ***
3020 LET T=1
3030 GOSUB 10020
3040 LET T=6
3050 GOSUB 10020
3060 IF F=1 THEN 1080
3070 GOSUB 11020
3080 IF C=11 THEN 3100
3090 GOTO 14020
3100 LET S=1
3110 LET A=3
3120 LET G=0
3130 LET T=12
3140 GOSUB 12020
3150 IF F=1 THEN 1080
3160 LET H=G
3170 LET S=1
3180 LET A=8
3190 LET G=0
3200 GOSUB 12020
3210 IF F=1 THEN 1080
3220 LET Y=H
3230 PRINT">";
3240 GOSUB 13020
3250 PRINT" ";
3260 LET I=0
3270 LET D=I+H
3280 LET Y=PEEK(D)
3290 GOSUB 13120
3300 PRINT" ";
3310 LET I=I+1
3320 IF I<8 THEN 3270
3330 POKE 2,13
3340 SYS 65490
3350 LET T=PEEK(197)
3360 IF T=63 THEN 1080
3370 LET H=H+8
3380 IF H>G THEN 1080
3390 GOTO 3220
4000 REM
4010 REM ***        EXIT               ***
4020 POKE 650,0
4030 POKE 204,1
4040 END
5000 REM
5010 REM ***        LOAD/SAVE          ***
5020 LET T=1
5030 LET D=PEEK(513)
5040 IF D=34 THEN 5080
5050 POKE 211,T
5060 PRINT"(UP,RIGHT)?"
5070 GOTO 1080
5080 LET M=0
5090 LET N=515+M
5100 LET T=M+4
5110 IF M>15 THEN 5050
5120 LET D=PEEK(N)
5130 IF D=34 THEN 5160
5140 LET M=M+1
5150 GOTO 5090
5160 LET T=M+4
5170 GOSUB 10020
5180 LET W=17
5190 IF V=83 THEN 5220
5200 LET W=7
5210 IF V=76 THEN 5260
5220 LET T=T+3
5230 GOSUB 10020
5240 LET T=T+5
5250 GOSUB 10020
5260 IF F=1 THEN 1080
5270 GOSUB 11020
5280 LET N=M+W
5290 IF C=N THEN 5320
5300 LET T=N
5310 GOTO 5050
5320 LET S=3
5330 LET A=M+6
5340 LET G=0
5350 LET T=A+1
5360 GOSUB 12020
5370 IF F=1 THEN 1080
5380 LET O=G
5390 LET P=M+1
5400 IF V=76 THEN 5720
5410 LET S=1
5420 LET A=M+9
5430 LET G=0
5440 LET T=A+3
5450 GOSUB 12020
5460 IF F=1 THEN 1080
```

Listing »Spitzmon 64«. Dieses Programm muß mit dem Ascompiler 64 aus der Ausgabe 1/86 compiliert werden.

```
5470 GOSUB 15020
5480 LET R=H
5490 LET Q=L
5500 LET A=M+14
5510 LET G=0
5520 LET T=A+3
5530 LET S=1
5540 GOSUB 12020
5550 IF F=1 THEN 1080
5560 GOSUB 15020
5570 LET A=PEEK(45)
5580 LET B=PEEK(46)
5590 GOSUB 16020
5600 POKE 251,Q
5610 POKE 252,R
5620 POKE 780,251
5630 POKE 781,L
5640 POKE 782,H
5650 POKE 20,216
5660 POKE 21,255
5670 SYS 57648
5680 POKE 45,A
5690 POKE 46,B
5700 PRINT
5710 GOTO 1080
5720 PRINT"{2UP}"
5730 GOSUB 16020
5740 POKE 780,0
5750 POKE 20,213
5760 POKE 21,255
5770 SYS 57648
5780 PRINT
5790 GOTO 1080
6000 REM
6010 REM ***        GOTO                ***
6020 LET T=1
6030 GOSUB 10020
6040 IF F=1 THEN 1080
6050 GOSUB 11020
6060 IF C=6 THEN 6080
6070 GOTO 14020
6080 LET S=1
6090 LET A=3
6100 LET G=0
6110 LET T=12
6120 GOSUB 12020
6130 IF F=1 THEN 1080
6140 GOSUB 15020
6150 POKE 20,L
6160 POKE 21,H
6170 SYS 57648
6180 GOTO 1080
7000 REM
7010 REM ***        FILL                ***
7020 LET B=513
7030 LET D=PEEK(B)
7040 IF D=32 THEN 7060
7050 GOTO 2020
7060 LET T=6
7070 GOSUB 10020
7080 LET T=11
7090 GOSUB 10020
7100 IF F=1 THEN 1080
7110 GOSUB 11020
7120 IF C=14 THEN 7140
7130 GOTO 14020
7140 LET S=1
7150 LET A=3
7160 LET G=0
7170 LET T=15
7180 GOSUB 12020
7190 LET H=G
7200 LET S=1
7210 LET A=8
7220 LET G=0
7230 GOSUB 12020
7240 LET I=G
7250 LET S=3
7260 LET A=13
7270 LET G=0
7280 GOSUB 12020
7290 IF F=1 THEN 1080
7300 POKE H,G
7310 LET H=H+1
7320 IF H>I THEN 1080
7330 GOTO 7300
8000 REM
8010 REM ***    TRANSFER/COMPARE        ***
8020 LET T=1
8030 GOSUB 10020
8040 LET T=6
8050 GOSUB 10020
8060 LET T=11
8070 GOSUB 10020
8080 IF F=1 THEN 1080
8090 GOSUB 11020
8100 IF C=16 THEN 8120
8110 GOTO 14020
8120 LET S=1
8130 LET A=3
8140 LET G=0
8150 LET T=15
8160 GOSUB 12020
8170 LET H=G
8180 LET S=1
8190 LET A=8
8200 LET G=0
8210 GOSUB 12020
8220 LET I=G
8230 LET S=1
8240 LET A=13
8250 LET G=0
8260 GOSUB 12020
8270 IF F=1 THEN 1080
8280 IF V=67 THEN 8370
8290 LET J=PEEK(H)
8300 POKE G,J
8310 LET H=H+1
8320 LET G=G+1
8330 IF H>I THEN 1080
8340 GOTO 8290
8350 REM
8360 REM ***        COMPARE             ***
8370 LET J=PEEK(H)
8380 LET K=PEEK(G)
8390 IF K=J THEN 8410
8400 GOTO 8470
8410 LET H=H+1
8420 LET Q=PEEK(197)
8430 IF Q=63 THEN 9740
8440 LET G=G+1
8450 IF H>I THEN 9740
8460 GOTO 8370
8470 LET Y=H
8480 GOSUB 13020
8490 PRINT" ";
8500 GOTO 8410
9000 REM
9010 REM ***        HIND                ***
9020 LET T=1
9030 GOSUB 10020
9040 LET T=6
9050 GOSUB 10020
9060 LET T=11
9070 GOSUB 10020
9080 IF F=1 THEN 1080
9090 GOSUB 11020
9100 IF C=14 THEN 9220
9110 LET T=14
9120 GOSUB 10020
9130 IF F=1 THEN 1080
9140 GOSUB 11020
9150 IF C=17 THEN 9220
9160 LET T=17
9170 GOSUB 10020
9180 IF F=1 THEN 1080
9190 GOSUB 11020
9200 IF C=20 THEN 9220
9210 GOTO 14020
9220 LET S=1
9230 LET A=3
9240 LET G=0
9250 LET T=C-1
9260 GOSUB 12020
9270 LET H=G
9280 LET S=1
9290 LET A=8
9300 LET G=0
9310 GOSUB 12020
9320 LET I=G
9330 LET S=3
9340 LET A=13
9350 LET G=0
9360 GOSUB 12020
9370 LET J=G
9380 IF F=1 THEN 1080
9390 IF T=13 THEN 9520
9400 LET S=3
9410 LET A=16
9420 LET G=0
9430 GOSUB 12020
9440 LET K=G
9450 IF F=1 THEN 1080
9460 IF T=16 THEN 9520
9470 LET S=3
9480 LET A=19
9490 LET G=0
9500 GOSUB 12020
9510 IF F=1 THEN 1080
9520 LET L=PEEK(H)
9530 LET O=H+1
9540 LET P=H+2
9550 LET M=PEEK(O)
9560 LET N=PEEK(P)
9570 IF T=13 THEN 9630
9580 IF T=16 THEN 9610
9590 IF N=G THEN 9610
9600 GOTO 9640
9610 IF M=K THEN 9630
9620 GOTO 9640
9630 IF L=J THEN 9700
9640 LET H=H+1
9650 LET Q=PEEK(197)
9660 IF Q=63 THEN 9740
9670 IF H>I THEN 9740
9680 IF H=0 THEN 9740
9690 GOTO 9520
9700 LET Y=H
9710 GOSUB 13010
9720 PRINT" ";
9730 GOTO 9640
9740 PRINT""
9750 GOTO 1080
10000 REM
10010 REM ***       BLANK VORHANDEN ? ***
10020 LET B=T+512
10030 LET D=PEEK(B)
10040 IF D=32 THEN 10080
10050 POKE 211,T
10060 PRINT"{UP,RIGHT}?"
10070 LET F=1
10080 RETURN
11000 REM
11010 REM ***       LAENGE DER ZEILE    ***
11020 LET C=0
11030 LET B=C+512
11040 LET B=PEEK(B)
11050 IF B = 0 THEN 11080
11060 LET C=C+1
11070 GOTO 11030
11080 RETURN
12000 REM
12010 REM ***       HEX/DEZ             ***
12020 LET B=511+A
12030 LET C=PEEK(B)
12040 IF C<65 THEN 12080
12050 IF C>70 THEN 12080
12060 LET C=C-55
12070 GOTO 12110
12080 IF C<48 THEN 12290
12090 IF C>57 THEN 12290
12100 LET C=C-48
12110 IF S=1 THEN 12150
12120 IF S=2 THEN 12180
12130 IF S=3 THEN 12210
12140 IF S=4 THEN 12230
12150 LET U=4096
12160 GOSUB 12310
12170 GOTO 12230
12180 LET U=256
12190 GOSUB 12310
12200 GOTO 12230
12210 LET U=16
12220 GOSUB 12310
12230 LET G=G+C
12240 IF S=4 THEN 12280
12250 LET S=S+1
12260 LET A=A+1
12270 GOTO 12020
12280 RETURN
12290 GOSUB 10050
12300 RETURN
12310 LET C=C*U
12320 RETURN
13000 REM
13010 REM ***       DEZ/HEX             ***
13020 LET U=4096
13030 GOSUB 13290
13040 GOSUB 13220
13050 GOSUB 13310
13060 LET Y=Y-B
13070 LET U=256
13080 GOSUB 13290
13090 GOSUB 13220
13100 GOSUB 13310
13110 LET Y=Y-B
13120 LET U=16
13130 GOSUB 13290
13140 GOSUB 13220
13150 GOSUB 13310
13160 LET Y=Y-B
13170 LET A=Y
13180 GOSUB 13220
13190 LET B=A
13200 LET Y=Y-B
13210 RETURN
13220 IF A>9 THEN 13270
13230 LET C=A+48
13240 POKE 2,C
13250 SYS 65490
13260 RETURN
13270 LET C=A+55
13280 GOTO 13240
13290 LET A=Y/U
13300 RETURN
13310 LET B=A*U
13320 RETURN
14000 REM
14010 REM ***       FEHLER              ***
14020 LET T=C-1
14030 GOSUB 10050
14040 GOTO 1080
15000 REM
15010 REM ***       LOW-HIGH-BYTE       ***
15020 LET H=G/256
15030 LET I=H*256
15040 LET L=G-I
15050 RETURN
16000 REM
16010 REM ***   PARAMETEREINSTELLUNG ***
16020 POKE 186,0
16030 POKE 185,1
16040 POKE 780,P
16050 POKE 781,2
16060 POKE 782,2
16070 POKE 20,189
16080 POKE 21,255
16090 SYS 57648
16100 RETURN
```

© 64'er

Listing »Spitzmon 64« (Schluß)