

Linker 64 — Schluß mit dem Nachladen

Wenn Sie öfters mehrere Maschinenprogramme gleichzeitig im Speicher haben wollen und es leid sind, jedes Programm einzeln nachzuladen, dann ist der Linker das ideale Arbeitswerkzeug für Sie.

Seine Anwendungen sind vielfältig: vom einfachen Basic-Start-Generator für Maschinenprogramme bis zum Verketter (= Linker) von mehrteiligen Spielen. Das Endprodukt, also das generierte Programm, kann dann ganz einfach auch von der Datasette geladen werden, was bisher an den Diskettenladerroutinen der Programme scheiterte. Diese Laderoutinen müssen natürlich vorher entfernt werden.

Die Bedienung ist denkbar einfach: Diskette mit den einzelnen Programmen nehmen, Linker 64 (siehe Listing) starten und bedienen. Es wird eine lauffähige Version generiert, die man mit LOAD und RUN starten kann.

Die einzige Denkarbeit besteht in der Überlegung, in welcher Reihenfolge die Programme wieder an ihre ursprünglichen Adressen verschoben werden sollen. Das ist wichtig, da sonst während des Verschiebevorgangs bereits verschobene Programmteile, die noch nicht verschoben überschreiben könnten. Im allgemeinen wird man zuerst die Maschinenroutinen verschieben, die an die höchste Adresse geschrieben werden sollen.

Die Arbeitsweise des Linkers

Der Linker generiert drei Programmteile, die am Ende verkettet werden: Teil »S« besteht aus der Basic-Zeilenummer, dem SYS-Befehl und dem anschließenden Text, gleich einer REM-Zeile. Teil »V« besteht aus dem generierten Verschiebeprogramm. Teil »P« besteht aus dem generierten Verschiebeprogramm nach oben die betriebssysteminterne Blockverschieberoutine (\$a3bf) angesprungen wird. Bei Transfer nach unten kommt eine »handgestrickte« Routine zur Anwendung. Teil »P« enthält schließlich die aneinandergehängten Maschinenprogramme.

Der Linker kennt zwei Betriebsarten: entweder Verschiebeteil vor oder nach den gelinkten Programmen. Im Modus »0« ist die Reihenfolge der drei Teile S-V-P, im Modus »1« ist sie S-P-V. Der Modus »0« (LoMem-Modus) wird fast immer benutzt, da nur sehr selten Programmteile im Basic-Startbereich (2049 = \$0801) laufen. Sollte dies aber doch der Fall sein, oder sollen Basic-Programme mit Maschinenprogrammen zusammengeknüpft werden, so muß Modus »1« (HiMem-Modus) verwendet werden. Die Verschieberoutine würde sich sonst selbst überschreiben.

Anwendung des Linkers

Beispiel: Wir wollen aus dem SMON \$c000, einer RENEW-Routine mit der Startadresse 36 000 und einem Assembler ASS \$9000 ein einziges Programm machen. Dieses sollte

nach dem Laden die einzelnen Teile gleich an die richtigen Adressen im Speicher versetzen und mit »RUN« zu starten sein. Weiterhin soll danach gleich der SMON starten:

(Alle drei Programme müssen sich auf der gleichen Diskette befinden.) Zuerst lädt man den Linker und startet ihn. Dann gibt man die Zeilennummer, in der später der SYS-Befehl stehen soll, einen erklärenden REM-Text und den Namen ein, den das fertige Programm erhalten soll. Als Betriebsart wählen wir »0«, da der Basic-Startbereich von unseren Programmen nicht berührt wird. Die Frage nach der Anzahl der Programme beantworten wir mit »3«. Nun werden die Namen der einzelnen Programme eingegeben, woraufhin der Linker deren Startadressen berechnet und ausgibt. Diese könnten jetzt noch geändert werden, was aber nicht sehr ratsam ist. Maschinenprogramme haben im allgemeinen nämlich die Eigenschaft, nur in dem Speicherbereich zu laufen, für den sie geschrieben wurden. In unserem Fall übernehmen wir also die Startadressen mit der RETURN-Taste.

Jetzt fragt der Linker nach der Reihenfolge, in der später die Programme verschoben werden sollen (ist in unserem Fall egal, da sich unsere Beispielpprogramme nicht im geringsten gegenseitig stören). Der Linker meldet sich mit:

```
1 = SMON $c000
2 = ASS $9000
3 = RENEW
4 = ENDE
```

Wir tippen zum Beispiel »2« für die Verschiebung des Assemblers. Nun wird für ihn die Einsprungsart verlangt: EINSPRUNG: 0=KEINER, 1=JSR, 2=JMP

Liste der wichtigsten Variablen:

U\$ = Überschrift
T\$ = REM-Text, der hinter dem SYS-Befehl im generierten Programm steht.
NA(U) = Neue Anfangsadresse (Adresse, an der das Einzelprogramm nach dem Verschieben steht)
NE(U) = Neue Endadresse
S(U) = Alte Anfangsadresse (dort steht der Programmteil im fertigen Programm)
E(U) = Alte Endadresse
K = Anzahl der zu linkenden Programme (maximal acht)
BS\$ = Floppy-Befehlsstring
LH = LoMem- oder HiMem-Modus
NA\$ = Name des fertigen Programms

Die Variablenliste des Linkers

Eingangsvariable	Ausgangsvariable	Beschreibung	Einsprung
A=2 byte-Zahl	AL% (LO-Anteil) AH% (HI-Anteil)		940
X\$=filename	C1 0=vorhanden, 62=nicht vorhanden	Testet auf Vorhandensein von Programmen	950
		Fehlerbehandlung: Return nur bei ok oder File not found, sonst Abbruch.	960
F\$=filename	SA=Startadresse, EA=Endadresse V=Verbrauch	Berechnet SA, EA,V und schreibt aktuellen Track und Sektor	1 120

Tabelle. Beschreibung der wichtigsten Unterprogramme

Da nach der Verschiebung des Assemblers dieser nicht gleich gestartet werden soll, geben wir eine »0« ein. Es erscheint am Bildschirm:

```
1 = SMON $c000
2 =
3 = RENEW
4 = ENDE
```

Sie sehen, der Assembler ist »verschunden«. Der Linker wartet auf das nächste Kommando. Wir geben »3« für die RENEW-Routine ein. Da diese ebenfalls nicht gleich gestartet werden soll, beantworten wir die nachfolgende Frage mit »0«. Dann verschieben wir mit »1« den SMON. Wir haben vorher gesagt, daß das fertige Programm diesen gleich starten sollte. Also Einsprungsart »2«. Die Einsprungsadresse, die der Linker daraufhin verlangt, ist 49152 (SMON).

Es wird noch einmal nach der Einsprungsart gefragt, da bei der Eingabe von »1« (JSR) noch weitere Einsprünge erfolgen

könnten. Wir quittieren das Ganze mit »0« und das Abenteuer ist für uns erledigt.

Jetzt wirft der Linker die DOS-interne Copy-Routine an, die die vorher generierten Teile »V« und »S« auf der Diskette direkt aneinanderhängt. Nach einiger Zeit meldet sich der Computer mit READY, und der Link-Vorgang ist beendet. Auf der Diskette befindet sich jetzt das fertige Programm, das mit LOAD"na-me",8 geladen und mit »RUN« gestartet werden kann. Es werden zuerst die gelinkten Programme in der festgelegten Reihenfolge an ihren Platz im Speicher verschoben und die durch JMP oder JSR definierten Adressen angesprungen. Achtung: Vor einem erneuten Start durch »RUN« sollte das Programm erst noch einmal geladen werden, ebenso sollte es nach dem Verschieben nicht mehr gespeichert werden.

Die wichtigsten Unterprogramme des Linkers sind in der Tabelle beschrieben. Die Routine »Test auf Vorhandensein von Programmen auf Disk« eignet sich gut zum Einbau in eigene Programme. (Andreas Knipp/tr)

10 REM *****	<025>	510 PRINT#3,CHR\$(AH)CHR\$(133)CHR\$(96);	<215>
20 REM * LINKER 64 *	<204>	520 A=E(U)+1:GOSUB 940	<109>
30 REM * (C) ANDREAS KNIPP *	<188>	530 PRINT#3,CHR\$(169)CHR\$(AL)CHR\$(133)CHR\$(90)CHR\$(169);	<004>
40 REM *****	<055>	540 PRINT#3,CHR\$(AH)CHR\$(133)CHR\$(91);	<207>
50 POKE 53280,4:POKE 53281,5	<206>	550 A=NE(U)+1:GOSUB 940	<059>
60 U\$=" {CLR,ORANGE,RVSON,40SPACE,RVOFF}" + CHR\$(13)	<104>	560 PRINT#3,CHR\$(169)CHR\$(AL)CHR\$(133)CHR\$(88)CHR\$(169);	<049>
70 U\$=U\$+" {WHITE,11SPACE}L I N K E R {3SPACE}6 4"+CHR\$(13)	<103>	570 PRINT#3,CHR\$(AH)CHR\$(133)CHR\$(89);	<041>
80 U\$=U\$+" {6SPACE,DOWN}PROGRAMMED BY ANDREAS KNIPP"+CHR\$(13)	<060>	580 PRINT#3,CHR\$(32)CHR\$(191)CHR\$(163);:Z=Z+27:GOTO 740	<164>
90 PRINT U\$	<172>	590 A=S(U):GOSUB 940:A1=Z+4	<204>
100 OPEN 15,8,15,"I":GOSUB 960	<084>	600 PRINT#3,CHR\$(160)CHR\$(0)CHR\$(185)CHR\$(AL)CHR\$(AH)CHR\$(153);	<095>
110 Z=0:A\$="":LL\$=" {37SPACE}" + CHR\$(13) + " {U P}"	<177>	610 Z=Z+6:A=NA(U):GOSUB 940:A2=Z+1	<247>
120 INPUT " {HOME,10DOWN}ZEILENNUMMER ";ZN	<253>	620 PRINT#3,CHR\$(AL)CHR\$(AH)CHR\$(200)+CHR\$(192);:A=NE(U)+1-AL	<042>
130 INPUT " {DOWN}TEXT {9SPACE}";T\$	<041>	630 GOSUB 940:Z=Z+4	<046>
140 PRINT LL\$ZN;"SY\$RRRRR "T\$	<126>	640 A3=AH	<135>
150 INPUT " {DOWN}NAME DES GENERIERTEN PRG";NA\$:X\$=NA\$:GOSUB 950	<043>	650 PRINT#3,CHR\$(AL)CHR\$(240)CHR\$(12)CHR\$(152)CHR\$(208);	<214>
160 IF C1=0 THEN PRINT " {UP}"NA\$ " {2SPACE}BE REITS VORHANDEN {8SPACE}":GOTO 120	<086>	660 PRINT#3,CHR\$(242)CHR\$(238);	<057>
170 INPUT " {DOWN}WIEVIELE PRG WERDEN VERKET TET ";K	<215>	670 Z=Z+7:A=A1:GOSUB 940	<164>
180 PRINT " {DOWN}VERSCHIEBEPRG VOR (=0)"	<138>	680 PRINT#3,CHR\$(AL)CHR\$(AH)CHR\$(238);	<101>
190 INPUT " {DOWN}NACH (=1) PROGRAMMBLOCK";LH	<025>	690 Z=Z+3:A=A2:GOSUB 940	<191>
200 INPUT " {DOWN}ALLES RICHTIG J/N";AR\$	<004>	700 PRINT#3,CHR\$(AL)CHR\$(AH)CHR\$(24)CHR\$(144)CHR\$(233);	<253>
210 IF AR\$<>"J" THEN 120	<087>	710 PRINT#3,CHR\$(173)CHR\$(AL)CHR\$(AH);	<072>
220 DIM NA(20),NE(20),S(20),E(20),N\$(20),N1\$(20)	<122>	720 Z=Z+8	<218>
230 Z=2064+LEN(T\$)	<142>	730 PRINT#3,CHR\$(201)CHR\$(A3)CHR\$(144)CHR\$(237);:Z=Z+4	<066>
240 IF LH=0 THEN SE=Z:Z=Z+K*40:PA=Z	<112>	740 INPUT "EINSPRUNG: 0=KEINER,1=JSR,2=JMP";E:IF E<0 OR E>2 THEN 740	<166>
250 IF LH THEN Z=Z+2	<104>	750 IF E=1 OR E=2 THEN INPUT "EINSPRUNG";A:GOSUB 940:IF INT(AH/256)GOTO 750	<075>
260 PRINT U\$	<088>	760 IF E=2 THEN PRINT#3,CHR\$(76)CHR\$(AL)CHR\$(AH);:Z=Z+3	<118>
270 FOR U=1 TO K	<223>	770 IF E=1 THEN PRINT#3,CHR\$(32)CHR\$(AL)CHR\$(AH);:Z=Z+3	<162>
280 PRINT "NAME DES"U" {LEFT}.PRG":INPUT N\$(U):N1\$(U)=N\$(U)	<112>	780 IF E<>0 THEN 740	<025>
290 X\$=N\$(U):GOSUB 950	<247>	790 GOTO 420	<012>
300 IF C1 THEN PRINT "DISES PRG GIBT ES NICHT":GOTO 280	<017>	800 IF LH=0 THEN FOR U=(Z+1)TO(PA-2):PRINT#3,CHR\$(RND(U)*255);:NEXT	<052>
310 NEXT U	<236>	810 CLOSE 3	<075>
320 FOR U=1 TO K	<017>	820 SA\$=MID\$(STR\$(SE)+" {5SPACE}",2,5)	<117>
330 F\$=N\$(U):GOSUB 1120	<111>	830 OPEN 2,8,2,"@:+S,P,W":GOSUB 960	<168>
340 PRINT " {2DOWN}STARTADRESSE VON {2SPACE}"N\$(U):PRINT " {RIGHT}"SA:INPUT " {UP}";SA	<246>	840 A=ZN:GOSUB 940	<011>
350 EA=SA+V	<015>	850 PRINT#2,CHR\$(1)CHR\$(8)CHR\$(13+LEN(T\$))CHR\$(8);	<094>
360 NA(U)=SA:NE(U)=EA:S(U)=Z	<136>	860 PRINT#2,CHR\$(AL);	<067>
370 E(U)=S(U)+V:Z=E(U)+2	<034>	870 PRINT#2,CHR\$(AH);	<073>
380 NEXT U	<050>	880 PRINT#2,CHR\$(158);	<231>
390 IF LH=0 THEN Z=SE	<007>	890 PRINT#2,SA\$+" "+T\$;	<099>
400 IF LH=1 THEN Z=Z-2:SE=Z	<230>	900 FOR I=1 TO 4	<110>
410 OPEN 3,8,3,"@:+V,P,W"	<115>	910 PRINT#2,CHR\$(0);:NEXT:CLOSE 2	<065>
420 PRINT U\$ " {5DOWN}"	<144>	920 IF LH THEN PRINT#15,"R:@=+S":GOSUB 960:GOTO 1000	<104>
430 FOR I=1 TO K:PRINT I,N\$(I):NEXT:PRINT K+1,"ENDE":N\$(K+1)="1"	<024>	930 PRINT#15,"C:@=+S,+V":GOSUB 960:GOTO 1000	<143>
440 INPUT "NAECHSTE PRG-NUMMER";U:IF U>K+1 OR N\$(U)="" THEN 440	<248>	940 AH=INT(A/256):AL=A-AH*256:RETURN	<120>
450 IF U=K+1 THEN 800	<202>		
460 N\$(U)=""	<093>		
470 IF E(U)=NE(U)GOTO 740	<251>		
480 IF E(U)>NE(U)GOTO 590	<134>		
490 A=S(U):GOSUB 940	<134>		
500 PRINT#3,CHR\$(169)CHR\$(AL)CHR\$(133)CHR\$(95)CHR\$(169);	<238>		

Listing »Linker 64«. Beachten Sie bitte die Eingabehinweise auf Seite 6

```

950 OPEN 6,8,6,X$:CLOSE 6          <231>
960 INPUT#15,C1,C$,C2,C3:IF C1=0 OR C1=62
    THEN RETURN                    <000>
970 PRINT C1,C$,C2,C3              <063>
980 PRINT"PROGRAMMABBRUCH!!!!"     <050>
990 CLOSE 15:SYS 65511:OPEN 1,8,15,"S:+":?":
    CLOSE 1:END                    <035>
1000 FOR I=1 TO K:N$(I)=N1$(I)     <133>
1010 N1$(I)="+"+RIGHT$(STR$(I),1):PRINT#15
    ,"R: "+N1$(I)+"="+N$(I):NEXT  <093>
1020 BS$="C:K=0,"                  <215>
1030 FOR I=1 TO K                  <125>
1040 BS$=BS$+N1$(I)                <244>
1050 IF I=K THEN 1070              <013>
1060 BS$=BS$+","                  <036>
1070 NEXT                          <064>
1080 IF LH THEN BS$=BS$+" "+V"     <065>
1090 PRINT#15,BS$:GOSUB 960:PRINT#15,"S:0" <125>
1100 FOR I=1 TO K:PRINT#15,"R: "+N$(I)+"="+
    N1$(I):NEXT                    <164>
1110 PRINT#15,"R: "+NA$+"="+K":GOTO 990 <041>
1120 OPEN 2,8,2,"#"               <113>
1130 T=18:S=1                      <189>
1140 PRINT#15,"U1";2;0;T;S        <245>
1150 PRINT#15,"B-P";2;0           <182>
1160 GOSUB 1440:T=A                <255>
1170 GOSUB 1440:S=A               <008>
1180 FOR X=0 TO 7                  <097>
1190 PRINT#15,"B-P";2;X*32+3      <024>
1200 GOSUB 1440:T1=A              <243>
1210 GOSUB 1440:S1=A              <252>
1220 FF$=""                        <072>

1230 FOR Y=1 TO 16                 <238>
1240 GOSUB 1440                    <116>
1250 IF FF$=F$ THEN 1310           <234>
1260 FF$=FF$+A$                    <042>
1270 IF A$<>MID$(F$,Y,1) THEN Y=16 <109>
1280 NEXT Y                         <222>
1290 NEXT X                         <224>
1300 GOTO 1140                     <114>
1310 PRINT U$"{3DOWN}IN VERARBEITUNG:{RVSO
    N}"+F$+"{RVOFF}"              <138>
1320 T=T1:S=S1                     <214>
1330 PRINT#15,"U1";2;0;T;S:PRINT LL$;T,S:P
    RINT" {UP}";                  <179>
1340 PRINT#15,"B-P";2;0           <118>
1350 GOSUB 1440:T=A                <191>
1360 GOSUB 1440:S=A               <200>
1370 IF Q=0 THEN Q=1:GOSUB 1440:SL=A:GOSUB
    1440:SH=A:SA=SL+SH*256        <183>
1380 IF T=0 THEN EA=SA+PZ*254+S-3:V=EA-SA:
    GOTO 1400                     <149>
1390 PZ=PZ+1:GOTO 1330            <164>
1400 PRINT "{DOWN}STARTADRESSE: "SA <131>
1410 PRINT"ENDADRESSE {2SPACE}:"EA <171>
1420 PRINT"VERBRAUCH {3SPACE}:"V   <144>
1430 CLOSE 2:Q=0:PZ=0:RETURN       <009>
1440 GET#2,A$:IF A$="" THEN A$=CHR$(0) <247>
1450 A$=ASC(A$):RETURN             <248>

```

64'er

Listing »Linker 64«. (Schluß)

Variablen-Dump

Mit diesem kleinen Programm und einem SYS-Befehl haben Sie alle Ihre Basic-Variablen voll im Griff. Eine, nicht nur zum Austesten von Programmen, nützliche Erweiterung.

»Dump« bedeutet übersetzt Speicherauszug oder Speicher Ausdruck eines Computers. Ein Variablen-Dump ist eine Übersicht aller Variablen, die in einem (Basic-)Programm verwendet werden. Es gibt verschiedene Möglichkeiten, die Erweiterung »Dump« zu benutzen:

- SYS 49152,A: Listet alle im Programm verwendeten indizierten Variablen mit dem Namen A auf; also A(1), A(2) und so weiter.
- SYS 49152 ohne Zusatz: Listet alle nichtindizierten Variablen.
- SYS 49152,\$: Listet nur alle nichtindizierten Stringvariablen.
- SYS 49152,%: Listet nur alle nichtindizierten numerischen Variablen (das Gegenstück zu SYS 49152,\$).

(Herbert Kunz/tr)

```

programm : dump c000      c000 c2e2

c000 : a0 00 b1 7a f0 26 20 fd 19
c008 : ae 85 f7 20 13 b1 90 03 82
c010 : 4c 54 c1 20 73 00 a5 f7 b8
c018 : c9 24 d0 06 a9 01 85 02 a5
c020 : d0 0e c9 25 d0 06 a9 02 f6
c028 : 85 02 d0 04 a9 00 85 02 18
c030 : a5 2d a4 2e 85 14 84 15 90
c038 : c4 30 d0 02 c5 2f 90 01 a3
c040 : 60 69 02 90 01 c8 85 22 98
c048 : 84 23 a9 0d 20 d2 ff 20 43
c050 : 94 c0 20 b3 c0 20 ec c0 05
c058 : a5 d3 c9 03 f0 08 a9 20 f0
c060 : 20 d2 ff 4c 58 c0 a9 3d 1f
c068 : 20 d2 ff a5 09 c9 01 d0 2a
c070 : 06 20 30 c1 4c 84 c0 c9 4a
c078 : 02 d0 06 20 17 c1 4c 84 22
c080 : c0 20 27 c1 a5 14 a4 15 0a
c088 : 18 69 07 90 a7 c8 4c 34 83
c090 : c0 4c 08 af 20 87 ea a5 a4
c098 : cb c9 39 f0 04 20 2c a8 f8
c0a0 : 60 20 87 ea a5 cb c9 40 b0
c0a8 : f0 f7 c9 39 f0 f3 a9 00 83
c0b0 : 85 c6 60 a0 00 84 09 b1 70
c0b8 : 14 10 0a a9 02 85 09 c5 88
c0c0 : 02 f0 1e d0 12 c8 b1 14 32
c0c8 : 10 0d a9 01 85 09 c5 02 a5
c0d0 : f0 0f a5 02 d0 0c 60 a5 2c
c0d8 : 02 c9 00 f0 04 c9 02 d0 15
c0e0 : 01 60 a9 91 20 d2 ff 68 17

c0e8 : 68 4c 84 c0 a0 00 b1 14 a8
c0f0 : 29 7f 20 d2 ff c8 b1 14 70
c0f8 : f0 05 29 7f 20 d2 ff a5 89
c100 : 09 c9 01 d0 08 a9 24 20 e7
c108 : d2 ff 4c 16 c1 c9 02 d0 c4
c110 : 05 a9 25 20 d2 ff 60 a0 27
c118 : 00 b1 22 aa c8 b1 22 a8 c3
c120 : 8a 20 95 b3 4c 2a c1 20 f3
c128 : a6 bb 20 dd bd 4c 1e ab 7e
c130 : 20 4f c1 a0 02 b1 22 85 bd
c138 : 25 88 b1 22 85 24 88 b1 51
c140 : 22 85 26 f0 0a b1 24 20 cb
c148 : d2 ff c8 c4 26 d0 f6 a9 fd
c150 : 22 4c d2 ff a9 00 85 f7 ed
c158 : 85 f8 20 79 00 85 f7 20 dd
c160 : 73 00 f0 0a 90 08 20 79 0d
c168 : 00 20 13 b1 90 09 85 f8 cd
c170 : f0 36 20 73 00 f0 31 85 49
c178 : f9 20 73 00 a5 f9 c9 24 f8
c180 : d0 0d a9 01 85 09 a9 80 aa
c188 : 05 f8 85 f8 4c ac c1 c9 4e
c190 : 25 f0 03 4c 08 af a9 02 20
c198 : 85 09 a9 80 05 f7 85 f7 32
c1a0 : a9 80 05 f8 85 f8 d0 04 55
c1a8 : a9 85 05 a9 a5 2f a4 30 9b
c1b0 : 85 14 84 15 c4 32 d0 07 32
c1b8 : a5 14 c5 31 d0 01 60 a0 d7
c1c0 : 00 98 aa b1 14 c5 f7 d0 de
c1c8 : 07 c8 b1 14 c5 f8 f0 28 5a
c1d0 : a0 02 b1 14 18 65 14 85 68
c1d8 : f9 90 01 e8 c8 b1 14 18 11
c1e0 : 65 15 85 15 8a f0 02 e6 da

c1e8 : 15 a5 f9 85 14 c5 31 d0 d5
c1f0 : ce a5 15 c5 32 d0 c8 60 1c
c1f8 : a0 04 b1 14 85 f9 aa c8 ee
c200 : b1 14 99 37 03 a9 00 99 b9
c208 : 7f 03 c8 b1 14 99 37 03 62
c210 : a9 00 99 7f 03 ca d0 e7 a9
c218 : c8 98 18 65 14 85 fb 90 5e
c220 : 01 e8 a5 15 85 fc 8a f0 ee
c228 : 02 e6 fc a9 8d 20 d2 ff 37
c230 : 20 94 c0 20 ec c0 a9 28 9a
c238 : 20 d2 ff a5 f9 85 fa a5 79
c240 : fa 38 e9 01 0a a8 b9 85 c9
c248 : 03 aa b9 84 03 20 cd bd 83
c250 : c6 fa a5 fa f0 08 a9 2c aa
c258 : 20 d2 ff 4c 3f c2 a9 29 6e
c260 : 20 d2 ff a9 3d 20 d2 ff 3e
c268 : a5 fb 85 22 a5 fc 85 23 4f
c270 : a5 09 c9 01 d0 07 20 30 52
c278 : c1 a9 03 d0 10 c9 02 d0 e2
c280 : 07 20 17 c1 a9 02 d0 05 8d
c288 : 20 27 c1 a9 05 18 65 fb 80
c290 : 85 fb 90 02 e6 fc 85 f9 5c
c298 : 8a 0a 38 e9 02 a8 8c ff 0a
c2a0 : 03 b9 85 03 18 69 01 99 46
c2a8 : 85 03 b9 85 03 d0 09 b9 1c
c2b0 : 84 03 18 69 01 99 84 03 de
c2b8 : b9 84 03 d9 3c 03 d0 08 df
c2c0 : b9 85 03 d9 3d 03 f0 03 ee
c2c8 : 4c 2b c2 ca f0 12 a9 00 fa
c2d0 : 99 84 03 99 85 03 88 88 43
c2d8 : a9 8d 20 d2 ff 4c a1 c2 19
c2e0 : 60 00 00 00 00 00 00 41

```

Listing »Variablen-Dump«. Bitte beachten Sie die Eingabehinweise auf Seite 8.