

C64-Programme für den VC20

Mit diesem Programm ist es möglich, auch C64-Programme auf Ihrem VC20 laufen zu lassen.

Das Programm »Reloctable« konvertiert Maschinenspracheroutinen, die für den C64 geschrieben worden sind, spielend leicht in ablauffähige Programme auf dem VC20. Es werden pro Sekunde etwa 50 Byte übersetzt und in den Speicher des VC20 geschrieben.

Zuerst muß die zu übersetzende Maschinenroutine in den Speicher Ihres VC20 gebracht werden. Dabei darf nur das von Basic aus erreichbare RAM beschrieben werden. Es ist dabei völlig unerheblich, an welche Stelle das Programm geschrieben (gePOKEt) wird, denn meist sind die Adreßräume des C64 für den VC20 sowieso nicht erreichbar. Aber bedenken Sie, daß das Übersetzungsprogramm noch nachgeladen werden muß (am besten den RAM-Top heruntersetzen).

Jetzt wird das Übersetzungs-Programm nachgeladen und mit RUN gestartet. Kurze Zeit später wird nach der Länge des zu bearbeitenden Programmes gefragt. Nach Eingabe der Bytezahl wird die Adresse (dezimal) erfragt, ab der das Maschinenprogramm steht. Eine weitere Abfrage klärt, ob, und wenn ja, wohin das Programm ablauffähig verschoben werden soll. Gibt man an dieser Stelle einfach RETURN ein, wird auf den Übersetzungsmodus von C64 auf VC20 umgeschaltet. Schließlich muß noch die Anfangsadresse eingegeben werden, ab der das Programm eigentlich im C64-Speicher stehen sollte. Mit der Meldung »Übersetzungsversuch« beginnt die Routine zu arbeiten. Nach READY kann das Programm mit SYS Anfangsadresse gestartet werden (Einsprungpunkt beim C64 beachten!).

Die Übersetzung klappt in 90 Prozent aller Fälle. Bei den

restlichen 10 Prozent handelt es sich um Maschinenprogramme, die den SID oder die Grafik des C64 direkt ansprechen. Da vergleichbare Adressen beim VC20 nicht vorhanden sind, mußte an dieser Stelle ein Kompromiß eingegangen werden. Das Umsetzungsprogramm reagiert in diesem Fall mit einer Meldung, daß die Übersetzung kritisch wird. Trotzdem kann man die Übersetzung starten, ohne einen Computerabsturz nach dem Start des behandelten Programmes befürchten zu müssen. An den entsprechenden, nicht konvertierbaren Stellen werden Unterprogramme eingesetzt. Ein Versuch lohnt sich allerdings immer. So gelingt die Übersetzung von Basic-Erweiterungen fast ausnahmslos. (Der Autor übersetzte nahezu die gesamte Maschinenspracheroutine-sammlung aus »C64 Tips and Tricks«).

Eine gleichwohl interessante Möglichkeit des Reloctable-Programms besteht im ablauffähigen Verschieben von VC20-Maschinenroutinen. Mit diesem Programmteil wird es leicht möglich, Speicherbereiche inklusive Adreßumrechnung frei im RAM zu verschieben. Sicher eine willkommene Hilfe bei der Zusammenstellung von Maschinensprache-Programmpaketen ohne die sonst üblichen lästigen ungenutzten Bereiche. Die Verschiebung klappt in mindestens 95 Prozent aller Fälle. Lediglich Maschinenprogramme, innerhalb derer Listen oder Strings abgelegt sind, erfordern eine gewisse »Nachbehandlung« mit einem Disassembler und dem manuellen Setzen der relevanten Bytes. Das läßt sich aber mit etwas Übung ganz gut machen. Aber auch ohne irgendwelche Maschinenkenntnisse ist die Trefferwahrscheinlichkeit des Programms groß genug. Übrigens können auch die übersetzten C64-Programme im zweiten Schritt frei verschoben werden. Für eine sinnvolle Arbeit mit dem vorliegenden Programm ist mindestens eine 8 KByte-Erweiterung notwendig. (Manfred Zimmermann/dm)

100 REM REOCTABLE	<139>	410 FS=37888+4*(PEEK(36866) AND 128)	<208>
110 REM TRANSLATION	<253>	420 PRINT:PRINT CHR\$(18)"UEBERSETZUNGSVERS	
120 REM C. BY M.ZIMMERMANN,WESEL	<115>	UCH!!"CHR\$(146):GOTO 510	<200>
130 REM	<192>	430 IF ZI>827 AND ZI+AN<1021 THEN 480	<048>
140 REM	<202>	440 IF ZI<PEEK(49)+256*PEEK(50) OR ZI+AN+6>	
150 PRINT"(CLR,2DOWN)*****		49151 THEN PRINT"(DOWN)UNGUELTIQUES ZIE	
*{DOWN}	<246>	L(2SPACE,2DOWN)":GOTO 300	<194>
160 PRINT"(5SPACE)RELOCTABLE(2DOWN)"	<250>	450 IF SA<ZI THEN IF SA+AN+1>ZI THEN PRIN	
170 PRINT"*****"	<167>	T"UNGUELTIQUES ZIEL(2DOWN)":GOTO 300	<181>
180 DIM OP%(255)	<087>	460 IF ZI<=SA THEN IF ZI+AN+1>SA THEN PRIN	
190 FOR T=0 TO 255	<120>	T"UNGUELTIQUES ZIEL(2DOWN)":GOTO 300	<235>
200 READ A%	<066>	470 PRINT:PRINT CHR\$(18)"VERSCHIEBUNGSVERS	
210 OP%(T)=A%	<089>	UCH!!"CHR\$(146)	<083>
220 NEXT	<230>	480 FOR T=0 TO AN-1	<237>
230 REM	<036>	490 POKE ZI+T,PEEK(SA+T)	<023>
240 REM STANDORTE	<088>	500 NEXT	<000>
250 REM	<056>	510 REM	<062>
260 INPUT"(DOWN)ANZ. D. BYTES ";AN	<206>	520 REM ANALYSE	<252>
270 IF AN=0 THEN 260	<191>	530 REM	<084>
280 INPUT"(DOWN)STARTADRESSE ";SA	<255>	540 PRINT:PRINT"WARTEN AUF "CHR\$(18)"READY	
290 ZI=SA	<188>	"CHR\$(146)" !!"	<199>
300 INPUT"(DOWN)ZIELADRESSE ";ZI	<149>	550 FOR T=0 TO AN-1	<053>
310 IF ZI<>SA THEN 430	<083>	560 WE=PEEK(SA+T)	<033>
320 TR\$="J"	<140>	570 IF KK<<"J" THEN 640	<050>
330 INPUT"(DOWN)START C-64 ";TS	<027>	580 IF WE=201 OR WE=224 OR WE=192 THEN 600	<165>
340 DIM C%(211),V%(211)	<232>	590 GOTO 660	<146>
350 FOR T=0 TO 210 STEP 2	<067>	600 H=PEEK(ZI+T+1)	<148>
360 READ CH,CL,VH,VL	<233>	610 IF H>26 AND H<42 THEN H=22+H-40:POKE Z	
370 C%(T)=CH:C%(T+1)=CL	<026>	I+T+1,H:GOTO 660	<170>
380 V%(T)=VH:V%(T+1)=VL	<179>	620 IF H>18 AND H<27 THEN H=23+H-25:POKE Z	
390 NEXT	<146>	I+T+1,H	<221>
400 BS=4*(PEEK(36866) AND 128)+64*(PEEK(368		630 GOTO 660	<186>
69) AND 120)	<245>	640 IF WE=169 OR WE=162 OR WE=160 THEN GOS	

Listing. »Reloctable«

```

UB 970
650 IF OP%(WE)=2 THEN GOSUB 880
660 T=T+OP%(WE)
670 NEXT
680 IF K<>"J" THEN 0800
690 IF K$="J" THEN K$="":KK$="J"
700 PRINT "CLR,2DOWN)DAS PROGRAMM VER-(DOWN)
N)"
710 PRINT "WENDETE ADRESSEN DES (DOWN)"
720 PRINT "BILDSCHIRM- UND FARB-(DOWN)"
730 PRINT "BEREICHS.C64 UND VC20 (DOWN)"
740 PRINT "UNTERSCHIEDEN SICH DA (DOWN)"
750 PRINT "WESENTLICH.EIN UEBER-(DOWN)"
760 PRINT "SETZUNGSVERSUCH IST (DOWN)"
770 PRINT "KRITISCH ! SOLL ICH ES"
780 PRINT "VERSUCHEN?(2DOWN)":INPUT "(J/N) "
;A$
790 IF A$="J" THEN 550
800 DH=INT(ZI/256)
810 DL=ZI-256*DH
820 POKE ZI+AN+2,DL
830 POKE ZI+AN+3,DH
840 POKE ZI+AN+4,197
850 POKE ZI+AN+5,206
860 POKE ZI+AN+6,196
870 END
880 REM
890 REM 3 BYTE-BEFEHL
900 REM
910 H=PEEK(SA+T+1)+256*PEEK(SA+T+2)
920 G=T
930 IF TR$="J" THEN GOSUB 1240
940 IF H<SA OR H>SA+AN-1 THEN RETURN
950 GOSUB 1510
960 RETURN
970 REM
980 REM LDA /LDX/ LDY
990 REM
1000 IF T=G AND T<>0 THEN RETURN
1010 G=T:GE=WE:KK=0
1020 H1=PEEK(SA+T+1)
1030 G=G+OP%(GE)+1
1040 IF G>AN OR KK>1 THEN G=T:RETURN
1050 GE=PEEK(SA+G)
1060 IF GE=169 OR GE=162 OR GE=160 THEN 10
80
1070 KK=KK+1:GOTO 1030
1080 H2=PEEK(SA+G+1)
1090 H=H1+256*H2
1100 IF TR$="J" THEN GOSUB 1240
1110 IF H<SA OR H>SA+AN-1 THEN RETURN
1120 GOSUB 1510
1130 RETURN
1140 REM
1150 REM BETRIEB
1160 REM
1170 L=0:H1=0:H2=0:IF H=58592 OR H=58602 T
HEN 1610
1180 H2=256*CX(L)+CX(L+1)
1190 IF H2=H THEN H=256*VX(L)+VX(L+1):RETU
RN
1200 IF H2>H THEN H=256*VX(L-2)+VX(L-1)+H-
H1:RETURN
1210 H1=H2:L=L+2
1220 IF L<211 THEN 1180
1230 RETURN
1240 REM
1250 REM UEBERSETZUNG
1260 REM
1270 N$=""
1280 REM ZERO
1290 IF H=784 OR H=785 OR H=786 THEN H=H-7
84:GOTO 1510
1300 REM BILDSCHIRM
1310 IF H>1023 AND H<2024 THEN H=BS+H-1024
:K$="J":GOTO 1510
1320 REM SPRITES
1330 IF H>2039 AND H<2048 THEN N$="J"
1340 REM INTERPRET 1
1350 IF H>40959 AND H<49151 THEN H=H+8192:
GOSUB 1590:GOTO 1510
1360 REM MASCH. RAM
1370 IF H>TS-1 AND H<TS+AN THEN H=SA+H-TS:
GOTO 1510
<177>
<011>
<161>
<172>
<095>
<079>
<075>
<242>
<047>
<182>
<228>
<134>
<222>
<031>
<175>
<138>
<003>
<102>
<110>
<058>
<162>
<006>
<154>
<110>
<180>
<197>
<200>
<063>
<149>
<244>
<101>
<000>
<000>
<014>
<088>
<034>
<225>
<208>
<086>
<115>
<109>
<155>
<015>
<065>
<104>
<037>
<160>
<017>
<172>
<172>
<186>
<181>
<206>
<222>
<157>
<020>
<123>
<163>
<082>
<016>
<030>
<215>
<050>
<079>
<139>
<235>
<192>
<181>
<116>
<195>
<181>
<043>
1380 IF H>49151 AND H<53248 THEN N$="J"
<221>
1390 REM VIDEO/SOUND
<164>
1400 IF H>53247 AND H<55295 THEN N$="J"
<155>
1410 REM INTERP. 2
<112>
1420 IF H>55295 AND H<56296 THEN H=FS+H-55
296:K$="J":GOTO 1510
<144>
1430 REM BETR. UNKLAR
<087>
1440 IF H>56295 AND H<57344 THEN N$="J"
<126>
1450 REM INTERPR. 2
<097>
1460 IF H>57343 AND H<58463 THEN H=H-3:GOS
UB 1590:GOTO 1510
<184>
1470 IF H>58462 AND H<65418 THEN GOSUB 117
0:GOTO 1510
<191>
1480 REM NOTLOESUNG
<224>
1490 IF N$="J" THEN DL=234:DH=DL:POKE ZI+T,
DH:POKE ZI+G,DH:GOTO 1540
<065>
1500 RETURN
<032>
1510 DD=H+ZI-SA
<037>
1520 DH=INT(DD/256)
<242>
1530 DL=DD-256*DH
<054>
1540 IF G=T THEN G=T+1
<070>
1550 POKE ZI+T+1,DL
<153>
1560 POKE ZI+G+1,DH
<063>
1570 IF G=T+1 THEN G=0
<013>
1580 RETURN
<114>
1590 IF WE=141 OR WE=142 OR WE=140 THEN 16
60
<064>
1600 RETURN
<134>
1610 PRINT "CLR,2DOWN)UEBERSETZUNG NICHT(D
OWN)
<110>
1620 PRINT "MOEGlich, DA ENT-(DOWN)
<035>
1630 PRINT "SPRECHENDE ROUTINE (DOWN)
<225>
1640 PRINT "BEIM VC-20 NICHT VOR-(DOWN)
<096>
1650 PRINT "HANDEN IST ! (DOWN)":END
<239>
1660 PRINT "CLR,2DOWN)UEBERSETZUNG NICHT(D
OWN)
<160>
1670 PRINT "MOEGlich, DA INS RAM (DOWN)
<098>
1680 PRINT "GEPOKED WIRD (2DOWN)"
<086>
1690 PRINT "STELLE (DEZ): ";H
<170>
1700 PRINT "(DOWN)":END
<074>
1710 REM
<248>
1720 REM BEFEHLSLAENGE
<098>
1730 REM
<012>
1740 DATA 0,1,0,0,1,1,1,0,0,1,0,0,2,2,2,0
<015>
1750 DATA 1,1,1,0,1,1,1,0,0,2,0,0,2,2,2,0
<193>
1760 DATA 2,1,0,0,1,1,1,0,0,1,0,0,2,2,2,0
<051>
1770 DATA 1,1,1,0,1,1,1,0,0,2,0,0,2,2,2,0
<213>
1780 DATA 0,1,0,0,0,1,1,0,0,1,0,0,2,2,2,0
<047>
1790 DATA 1,1,1,0,0,1,1,0,0,2,0,0,0,2,2,0
<209>
1800 DATA 0,1,0,0,1,1,1,0,0,1,0,0,2,2,2,0
<077>
1810 DATA 1,1,1,0,1,1,1,0,0,2,0,0,2,2,2,0
<255>
1820 DATA 1,1,0,0,1,1,1,0,0,1,0,0,2,2,2,0
<105>
1830 DATA 1,1,1,0,1,1,1,0,0,2,0,0,2,2,2,0
<019>
1840 DATA 1,1,1,0,1,1,1,0,0,1,0,0,2,2,2,0
<253>
1850 DATA 1,1,1,0,1,1,1,0,0,2,0,0,2,2,2,0
<039>
1860 DATA 1,1,0,0,1,1,1,0,0,1,0,0,2,2,2,0
<145>
1870 DATA 1,1,1,0,0,1,1,0,0,2,0,0,0,2,2,0
<035>
1880 DATA 1,1,0,0,1,1,1,0,0,1,0,0,2,2,2,0
<165>
1890 DATA 1,1,1,0,0,1,1,0,0,2,0,0,0,2,2,0
<055>
1900 REM
<184>
1910 REM SYSTEMADR.
<102>
1920 REM
<204>
1930 DATA 228,95,228,41,229,0,229,0,229,5,
229,5,229,10,229,10
<146>
1940 DATA 229,24,229,24,229,68,229,95,229,
102,229,129,229,160,229,187
<156>
1950 DATA 229,180,229,207,229,202,229,229,
230,50,230,79,230,132,230,184
<067>
1960 DATA 230,182,230,234,232,218,233,33,2
32,234,233,117,233,200,234,86
<107>
1970 DATA 233,255,234,141,234,28,234,161,2
34,36,234,178,234,49,234,191
<139>
1980 DATA 234,135,235,30,235,72,235,220,23
5,121,236,70,235,177,236,94
<243>
1990 DATA 236,68,237,33,236,120,237,105,23
6,185,237,228,236,231,237,243
<186>
2000 DATA 236,240,237,254,237,9,238,20,237
,12,238,23,237,64,238,228
<181>
2010 DATA 237,185,238,192,237,199,238,206,
237,239,238,246,237,254,239,4
<192>
2020 DATA 238,19,239,25,238,179,239,150,23
8,187,239,163,239,74,240,39
<100>
2030 DATA 240,20,240,237,240,134,241,79,24
0,164,241,96,240,189,241,116
<164>

```

Listing. »Reloctable« (Fortsetzung)

```

2040 DATA 241,43,241,224,241,87,242,14,241
    ,202,242,122,242,14,242,199 <233>
2050 DATA 242,80,243,9,242,145,243,74,243,
    15,243,207,243,31,243,223 <083>
2060 DATA 243,47,243,239,243,74,244,10,244
    ,158,245,66,245,175,246,71 <246>
2070 DATA 245,210,246,106,245,221,246,117,
    246,143,247,40,246,155,247,52 <131>
2080 DATA 246,221,247,96,246,228,247,103,2
    46,237,247,112,246,251,247,126 <075>
2090 DATA 247,44,247,175,247,106,247,231,2
    47,208,248,77,247,215,248,84 <086>
2100 DATA 247,234,248,103,248,13,248,138,2
    48,23,248,148,248,46,248,171 <034>
2110 DATA 248,56,248,183,248,65,248,192,24
    8,74,248,201,248,100,248,234 <175>

2120 DATA 248,107,248,234,248,190,249,47,2
    48,225,249,75,249,44,249,142 <021>
2130 DATA 251,151,248,219,251,166,251,234,
    251,205,252,11,252,184,252,246 <254>
2140 DATA 252,202,253,11,252,209,253,17,25
    2,219,253,27,252,226,253,34 <000>
2150 DATA 253,2,253,63,253,16,253,77,253,2
    1,253,82,253,48,253,109 <094>
2160 DATA 253,80,253,141,253,155,253,109,2
    53,249,254,73,254,0,254,80 <090>
2170 DATA 254,7,254,87,254,24,254,102,254,
    28,254,106,254,33,254,246 <146>
2180 DATA 254,37,254,115,254,52,254,130,25
    4,67,254,169,254,194,255,92 <167>
2190 DATA 255,72,255,114,255,129,255,138
    0 64'er <196>

```

Listing. »Relactable« (Schluß)

Tips & Tricks zum C16

Auch zu Commodores Kleinsten haben unsere Leser ein paar nützliche Tricks entdeckt, die nicht im Handbuch stehen, den Umgang mit diesem Computer aber wesentlich erleichtern.

Wer vom VC 20 oder C 64 auf den C 16/116 umsteigt, wird im ersten Augenblick die bewährte »RESTORE«-Taste vermissen. Hatte sich der Computer einmal »aufgehängt« (zum Beispiel durch eine falsche WAIT-Anweisung oder einen falschen Sprung in ein Maschinenprogramm), so konnte man ihn fast immer mit RUN/STOP-RESTORE zurückholen. Was macht man nun beim C 16/116, wenn sich der Computer in eine Endlosschleife verabschiedet? Nicht gleich ausschalten! Hier gibt es einen kleinen Trick:

Zuerst den »RESET«-Taster drücken (aber noch nicht loslassen). Dann gleichzeitiges Drücken der »RUN/STOP«- und »Commodore«-Taste. Anschließend »RESET«-Taster loslassen und schon befindet man sich im Monitor-Programm, das mit »X« wieder verlassen werden kann. Der alte Speicherinhalt bleibt erhalten!

(C. Q. Spitzner/tr)

Diverses

Der Bildschirm des C 16 beginnt bei \$0C00 (= # 3072), der zugehörige Farb-RAM bei \$0800 (= # 2048) und die hochauflösende Grafik (HiRes) bei \$2000 (= # 8192). Es können also einfarbige Grafiken vom C 64 problemlos übernommen werden. Die Farbtintensität für HiRes ist angesiedelt bei \$1800(bis \$1FFF). Einen Basic-Warmstart erreicht man durch »SYS 32768« (\$8000), hier beginnt auch der ROM-Teil. »SYS 62114« (= \$F2A4) macht einen kompletten Reset. Wer von anderen Computern gewohnt ist, seinen Monitor durch einen »SYS«-Befehl zu starten, kann dies mit »SYS 62533« (= \$F445) tun.

Die ESC-Funktionen lassen sich auch von Maschinensprache aus nutzen (siehe Tabelle). Für das Definieren von Bildschirmfenstern ist es notwendig, die Eckpunkte anzugeben. Hierbei ist \$CD = Zeile und \$CA = Spalte, also zum Beispiel:

```

LDX # $02
STX $CA ; dritte Zeile und
STX $CD ; dritte Spalte
JSR $DE5E ; »T«-Befehl
LDX # $12
STX $CA ; 18. Zeile und
STX $CD ; 18. Spalte
JMP $DE60 ; »B«-Befehl

```

definiert ein Bildschirmfenster ab dritter Zeile und Spalte mit 15 Zeilen (18 - 3) Länge zu je 15 Zeichen.

Gerade die Möglichkeit, mit »Windows« zu arbeiten, hebt den C 16/116 weit über seine Konkurrenten hervor. Auch die anderen Escape-Funktionen bieten umfangreiche Möglichkeiten der Bildschirmgestaltung.

Übrigens, die »JMP«-Anweisung am Schluß des kleinen Beispielprogramms ist gleichbedeutend mit »JSR« und anschließendem »RTS«.

Tabelle. Die Escape-Funktionen und ihre Adressen:

ESC+	Adresse Dez/Hex	Bedeutung
A	57129/\$DF29	Automatisch Einfügen
B	56928/\$DE60	Rechte, untere Fensterecke fixieren
C	57126/\$DF26	Automatisch Einfügen aufheben
D	56992/\$DEA0	Zeile löschen
I	56971/\$DE8B	Zeile einfügen
J	57218/\$DF82	Cursor an Zeilenanfang setzen
K	57237/\$DF95	Cursor an Zeilenende setzen
L	57117/\$DF1D	Scrolling-Modus einschalten
M	57120/\$DF20	Scrolling-Modus ausschalten
N	55432/\$D888	Bildschirm löschen und auf normale Größe schalten
O	56475/\$DC9B	Einfüge-, Anführungszeichen-, Reverse- und Blinkmodus ausschalten
P	57057/\$DEE1	Zeile löschen
Q	57035/\$DECB	Zeile ab Cursor löschen
R	56904/\$DE48	Bildschirm löschen und verkleinern
T	56926/\$DE5E	Linke, obere Fensterecke fixieren
V	57078/\$DEF6	Bildschirm nach oben scrollen
W	57092/\$DF04	Bildschirm nach unten scrollen
X	-/-	Escape-Modus ausschalten

Weitere Tips

- »WAIT 1,192« wartet darauf, daß eine Taste am Recorder gedrückt wird.
- Der Speicher für die Funktionstasten beginnt bei \$0567 und endet bei \$05E6.
- Die Einsprungadresse für die USR-Funktion steht in #1281/1282 = \$0501/0502. (Frank Plachetta/tr)