```
3335 GOSUB 3420:GOTO 3300                          <225>
3337 IF LEFT$(ZE$,1)="L" THEN GOSUB 3420:G
     OTO 3300                                      <128>
3340 ZE=VAL(ZE$)                                   <004>
3350 ZE$=MID$(ZE$,3)                               <078>
3360 IF ZE>9 THEN ZE$=MID$(ZE$,2)                  <032>
3370 SC$(ZE)=ZE$                                   <141>
3380 GOSUB 2630:IF BE$="-->" THEN GOTO 322
     0                                             <037>
3390 GOTO 3310                                     <124>
3393 :                                             <067>
3395 REM ************************                  <227>
3400 REM *****  LIST SCREEN  *****                 <098>
3405 REM ************************                  <237>
3408 :                                             <082>
3410 GOSUB 3110                                    <128>
3420 PRINT"{CLR}";                                 <056>
3430 FOR ZE=0 TO 23                                <248>
3440 PRINT RIGHT$(STR$(ZE),2);":";                 <157>
3450 PRINT LEFT$(SC$(ZE),38)                       <018>
3460 NEXT ZE:RETURN                                <184>
3463 :                                             <137>
3465 REM ***********************                   <048>
3470 REM ** HOLE WERT VOM TOS **                   <047>
3475 REM ***********************                   <058>
3480 AD=52992+PEEK(2)                              <065>
3490 X=PEEK(AD-1)+256*PEEK(AD-2)                   <138>
3500 POKE 2,PEEK(2)-2:RETURN                       <243>
3503 :                                             <177>
3505 REM ***********************                   <088>
3510 REM ***   SAVE-SYSTEM   ***                    <056>
3515 REM ***********************                   <098>
3518 :                                             <192>
3520 GOSUB 2630                                    <118>
3530 OPEN 1,8,15,"S:"+BE$+".*":CLOSE 1             <206>
3540 OPEN 2,8,2,BE$+".VOC,P,W"                     <175>
3550 PRINT#2,AN:PRINT#2,VOC                        <063>
3560 FOR ZE=39 TO AN                               <075>
3570 PRINT#2,WO$(ZE)                               <202>
3580 PRINT#2,AD(ZE)                                <160>
3590 NEXT ZE                                       <082>
3600 CLOSE 2:BE$=BE$+".CODE"                       <016>
3610 POKE 187,FN L(720):POKE 188,FN H(720)         <013>
3620 FOR I=1 TO LEN(BE$)                           <095>
3630 POKE 719+I,ASC(MID$(BE$,I,1))                 <115>
3640 NEXT I:POKE 183,LEN(BE$)                      <233>
3650 POKE 186,8:POKE 185,1                         <199>
3660 POKE 251,FN L(BE):POKE 252,FN H(BE)           <173>
3670 POKE 780,251                                  <205>
3680 POKE 781,FN L(VOC)                            <225>
3690 POKE 782,FN H(VOC)                            <007>
3700 SYS 216+256*255                               <020>
3710 GOTO 700                                      <130>
3713 :                                             <133>
3715 REM ***************************               <012>
3720 REM ****    LOAD SYSTEM    ****                <056>
3725 REM ***************************               <022>
3728 :                                             <148>
3730 GOSUB 2630                                    <074>
3740 OPEN 2,8,2,BE$+".VOC,P,R"                     <217>
3750 INPUT#2,AN,VOC                                <249>
3760 FOR ZE=39 TO AN                               <021>
3770 INPUT#2,WO$(ZE)                               <068>
3780 INPUT#2,AD(ZE)                                <026>
3790 NEXT ZE:CLOSE 2                               <162>
3800 SYS 50139,BE$+".CODE",8                       <110>
3810 GOTO 700                                      <230>
3813 :                                             <233>
3815 REM ***************************               <112>
3820 REM ***        DATA        ***                <160>
3825 REM ***************************               <122>
3828 :                                             <248>
3830 DATA 166,251, 32,198,255,160,  0, 32,
     207,255,201, 13,240,  7,153,  0               <161>
3840 DATA  2,200, 76, 69,  3,169,  0,153,
      0,  2, 76,204,255                            <248>
3850 FOR I= 830 TO 858:READ A:POKE I,A:Z=Z
     +A:NEXT I                                     <157>
3860 IF Z<>3379 THEN PRINT"{RVSON}FEHLER I
     N DATA{RVOFF}":END                            <107>
3870 RETURN                                        <118>
```

© 64'er

Listing 1. Tiny-Forth-Compiler (Schluß)

Hypra-Save

Hypra-Save ist eine Ergänzung zu Hypra-Load. Es speichert Programme 3- bis 5mal schneller und kann mit Hypra-Load verwendet werden.

Ein großer Nachteil der Diskettenstation VC 1541 ist die durch den seriellen Bus und durch das DOS V2.6 bedingte geringe Geschwindigkeit. Inzwischen gibt es mehrere Programme, die das Laden von Diskette beschleunigen. Mit der hier vorgestellten Routine geht jetzt auch das Speichern von Programmen mit dem C 64 wesentlich rascher.

Hypra-Save ist 3- bis 5mal so schnell wie die Originalroutine. Es verträgt sich mit Hypra-Load und vielen anderen, auch professionellen, Programmen und Basic-Erweiterungen. Zur Bedienung von Hypra-Save sollten Sie folgendes beachten: Die Eingabe muß mit dem MSE erfolgen. Nach dem Laden startet man es wie gewohnt mit RUN. Danach sollte man NEW eingeben, wenn man ein eigenes Programm schreiben will.

Hypra-Save kann Files mit oder ohne Verify speichern.

Gibt man vor dem Filenamen als erstes Zeichen einen Stern ein, so wird nicht verifiziert. Feststellbar an bis zu 5mal schnelleren Speicherzeiten. Mit Verify ist Hypra-Save etwa 3mal schneller als die Original-SAVE-Routine. Wer einen »25, WRITE ERROR« bisher nur aus der Literatur kennt, der kann getrost ohne Verify arbeiten. Selbstverständlich kann man weiterhin Programme überschreiben. Dann ist der Klammeraffe mit anzugeben. So überschreibt der Befehl SAVE "*@:name", 8 ein File, ohne die auf Diskette geschriebenen Blöcke zu prüfen, also ohne Verify. Hat der Computer alle Daten gesendet, wird im Gegensatz zur Original-SAVE-Routine nicht gewartet, bis das Laufwerk die Datei geschlossen hat. Dies macht sich besonders beim Überschreiben von Programmen bemerkbar. Die Floppystation arbeitet noch, während der Computer sich längst zurückgemeldet hat. Man darf die Diskette selbstverständlich nicht vor dem Erlöschen der roten LED aus dem Laufwerk nehmen.

Beim Speichern von Programmen mit dem Klammeraffen kommt die 1541 häufig ins »Schleudern«, wie Sie vielleicht aus eigener Erfahrung wissen. So kann es passieren, daß einige Programme nicht mehr geladen werden können. Löschen Sie daher ein Programm erst mit dem SCRATCH-Befehl und speichern Sie erst dann die neue Version ab.

Anstelle von RUN/STOP dient bei Hypra-Save 64 die RESTORE-Taste zum Abbrechen. Allerdings wird die Programmdatei dann nicht geschlossen und erscheint im Directory mit einem Stern. Möchte man die nicht geschlossene Datei löschen, so geht das nicht mit dem SCRATCH-, sondern nur mit dem VALIDATE-Befehl. Übrigens blinkt die rote LED nach dem Drücken von RESTORE. Liest man dann den Fehlerkanal aus, so erhält man die Meldung »51, OVERFLOW IN RECORD«, gefolgt von Track und Sektor des letzten Blocks. Diese Fehlermeldung weist sonst auf einen Übertragungsfehler hin, der von zu vielen Geräten am seriellen Bus herrühren kann.

Sollte man eine merkwürdige Fehlernummer wie zum Beispiel 61 oder 71 erhalten, hilft meist nur ein Aus- und Einschalten der Floppy. Nach STOP/RESTORE oder nach einem Reset ist der SAVE-Vektor zurückgesetzt. Hypra-Save läßt sich dann mit SYS 365 wieder aktivieren.

Das Programm verschiebt sich nach dem Starten. Der eine Teil von Hypra-Save liegt ab $0150 bis $0177, der andere von $D000 bis $D5FF im Speicher. Wird die SAVE-Routine aufgerufen, so wird über den geänderten SAVE-Vektor in die Initialisierungs-Routine nach $0150 gesprungen. Dort wird geprüft, ob die anzusprechende Gerätenummer größer als 3

ist. Wenn ja, blendet die Routine das RAM unter den I/O-Bausteinen ein. Dann kopiert es den Speicherbereich von $C000 bis $C5FF, der nach dem Speichern wieder zurückgeschrieben wird, nach $D600 bis $DBFF (RAM) und verschiebt die eigentlichen SAVE-Routinen von $D000 bis $D5FF (ROM) nach $C000 bis $C5FF. Denn die I/O-Bausteine können nicht vom darunterliegenden RAM angesprochen werden.

Durch diese Speicherbelegungstechnik wird kein normalerweise sichtbares RAM verändert und nur ein minimaler Bereich im Stack von der Hypra-Save-Routine belegt. Vom Programmteil bei $C000 wird das File zuerst wie gewohnt durch die normalen IEC-Busroutinen geöffnet. Dann testet Hypra-Save ob kein Fehler aufgetreten ist und die Daten gesendet werden können. Ist kein Fehler aufgetreten, so wird die SAVE-Routine für das Diskettenlaufwerk in das Floppy-RAM geschrieben und gestartet. Dieser Programmteil liegt von $0146 bis $01D9 im Floppy-RAM. Er enthält hauptsächlich die Transferroutine, mit der sofort nach dem Aufrufen das Diskettenstationsprogramm übertragen, in den Bereich von $0300 bis $04A0 geschrieben und dann gestartet wird.

Der wesentlich zur Geschwindigkeitserhöhung beitragende Datentransfer beruht auf der gleichzeitigen Übertragung von je zwei Bit und auf dem erheblich eingeschränkten Handshakebetrieb, mit dem beide Geräte ihre Bereitschaft zum Transfer signalisieren.

Die hohe SAVE-Geschwindigkeit wird aber auch vom Schreibalgorithmus erzielt, der die für das eventuell durchzuführende Verify benötigte Zeit erheblich verkürzt. Hierzu ein Beispiel, um die Arbeitsweise dieses Algorithmus zu verdeutlichen:

Wenn sich der Schreib-/Lesekopf in einer halben Diskettenumdrehung über dem Header des gewünschten Blocks befindet, die zu schreibenden Daten im Puffer stehen und codiert sind, so hat die Originalroutine den Pufferinhalt nach einer halben Umdrehung geschrieben und nach einer weiteren ganzen Umdrehung verifiziert. Diese Methode benötigt 1,5 Umdrehungen pro Block, also drei für zwei Blöcke.

Hypra-Save geht anders vor. Es verwendet zwei Puffer. Nach einer halben Umdrehung wird der erste, nach einer weiteren halben der zweite Block geschrieben. Es wird dann noch eine Umdrehung gebraucht, um beide Sektoren zu verifizieren. Man benötigt also nur zwei Rotationen für zwei Blöcke, der Zeitgewinn beträgt etwa 0,1 Sekunden pro Block. Sind sämtliche Bytes übertragen, so stellt der Computer wie erwähnt das RAM von $C000 bis $C5FF wieder her, springt in die Digitalisierungsroutine, schaltet die ursprüngliche Speicherbelegung ein und kehrt ins aufrufende Programm zurück. Das Laufwerk schließt währenddessen das File, wofür es mehr Zeit brauchen kann, wenn beim Überschreiben das alte Programm gelöscht werden muß.

(Martin Pfost/hm)

```
PROGRAMM : HYPRA-SAVE          0801 0D3F
-----------------------------------------

0801 : 0C 08 C1 07 9E 20 32 30  77
0809 : 36 32 00 00 00 78 A5 01  B5
0811 : 48 A9 34 85 01 A0 28 B9  15
0819 : 51 08 99 4F 01 88 D0 F7  46
0821 : A2 06 84 AC A9 D0 85 AD  0F
0829 : A9 A0 85 14 A9 08 85 15  21
0831 : B1 14 91 AC C8 D0 F9 E6  AF
0839 : 15 E6 AD CA D0 F2 20 6D  86
0841 : 01 68 85 01 58 A2 DA BD  79
0849 : A0 07 20 D2 FF E8 D0 F7  4A
0851 : 60 A5 BA C9 04 B0 03 4C  D6
0859 : ED F5 78 A6 01 A9 34 85  6D
0861 : 01 8E 9F D4 20 4F D4 AE  59
0869 : 9F D4 86 01 58 60 A9 50  04
0871 : 8D 32 03 A9 01 8D 33 03  5D
0879 : 60 0D 0D 09 0E 55 4C 54  2A
0881 : 52 41 53 41 56 45 20 36  ED
0889 : 34 20 41 4B 54 49 56 0D  8A
0891 : 49 4E 49 54 20 3A 20 53  D9
0899 : 59 53 20 33 36 35 0D BA  C1
08A1 : 8E A0 D4 AD 19 03 8D A1  8D
08A9 : D4 AD 18 03 8D A2 D4 A9  4F
08B1 : 9C 8D 18 03 A9 C1 8D 19  8B
08B9 : 03 AD 9F D4 85 01 AD 00  2C
08C1 : DD 8D 1F 01 58 A9 61 85  B0
08C9 : B9 A2 2C A0 00 B1 BB C9  03
08D1 : 2A D0 0A A2 4C C6 B7 E6  E2
08D9 : BB D0 02 E6 BC 8E A5 C3  B8
08E1 : A9 C1 48 A9 75 48 A5 B7  52
08E9 : D0 03 4C 10 F7 20 D5 F3  10
08F1 : 20 8F F6 A5 BA 20 0C ED  04
08F9 : A5 B9 20 B9 ED A9 00 85  F1
0901 : 90 20 DD ED 20 DD ED 20  BF
0909 : FE ED A5 90 F0 02 18 60  B9
0911 : A9 1A A2 C2 85 AC 86 AD  FB
0919 : A9 46 A2 01 85 14 86 15  EB
0921 : A2 05 A5 BA 20 0C ED A9  74
0929 : 6F 20 B9 ED A0 FD B9 15  DF
0931 : C1 20 DD ED C8 D0 F7 A5  76
0939 : 14 20 DD ED A5 15 20 DD  D2
0941 : ED A9 1E 20 DD ED A0 00  5E
0949 : B1 AC 20 DD ED C8 C0 1E  78
0951 : 90 F6 20 FE ED A5 AC 18  33
0959 : 69 1E 85 AC 90 02 E6 AD  D8
0961 : A5 14 18 69 1E 85 14 90  C3
0969 : 02 E6 15 CA D0 B4 A5 BA  3C
0971 : 20 0C ED A9 6F 20 B9 ED  03
0979 : A0 FB B9 1A C1 20 DD ED  39
0981 : C8 D0 F7 20 FE ED 78 A2  3A
0989 : 00 8E 00 DD AD 11 D0 29  85
0991 : EF 8D 11 D0 BD AE C2 9D  3D
0999 : 00 C5 E8 D0 F7 AD 00 DD  79
09A1 : 30 FB 20 FE C1 A2 00 BD  63
09A9 : AE C3 9D 00 C5 E8 D0 F7  77
09B1 : 20 FE C1 20 8E FB A2 01  1A
09B9 : A5 AC 9D 01 C5 E8 A5 AD  D1
09C1 : 9D 01 C5 E8 20 D1 FC B0  53
09C9 : 38 A5 AD C9 C0 90 1D C9  11
09D1 : C6 B0 19 69 16 85 AD A9  FA
09D9 : 34 85 01 B1 AC 48 AD 9F  49
09E1 : D4 85 01 A5 AD E9 15 85  F6
09E9 : AD 68 B0 02 B1 AC 9D 01  30
09F1 : C5 E8 20 DB FC E0 FF 90  A6
09F9 : CB 20 D1 FC B0 03 A9 FF  B2
0A01 : 2C A9 00 48 8D 00 C5 8E  18
0A09 : 01 C5 20 FE C1 A2 01 68  DB
0A11 : D0 B2 18 A9 00 48 AD 11  91
0A19 : D0 09 10 8D 11 D0 AD 1F  B0
0A21 : 01 8D 00 DD 78 A9 34 85  55
0A29 : 01 AD A2 D4 8D 18 03 AD  45
0A31 : A1 D4 8D 19 03 68 AE A0  32
0A39 : D4 9A 60 38 AD 1F 01 09  63
0A41 : 10 8D 1F 01 B0 CD B9 00  60
0A49 : C5 85 95 A2 00 2C 00 DD  A8
0A51 : 50 FB 2C 00 DD 10 05 E8  EE
0A59 : D0 F8 F0 B6 A9 10 8D 00  0A
0A61 : DD AA 66 95 6A 66 95 6A  E5
0A69 : 4A 4A 8D 00 DD 8A 66 95  33
0A71 : 6A 66 95 6A 4A 4A 8D 00  EE
0A79 : DD 8A 66 95 6A 66 95 6A  ED
0A81 : 4A 4A 8D 00 DD 8A 66 95  4B
0A89 : 6A 66 95 6A 4A 4A 8D 00  06
0A91 : DD EA EA EA A9 00 8D 00  CC
0A99 : DD C8 D0 AA 60 A0 00 98  A0
0AA1 : 59 00 C5 C8 D0 FA 85 14  A8
0AA9 : 20 A7 C1 88 A5 14 4C AA  A0
0AB1 : C1 4D 2D 57 4D 2D 45 B7  12
0AB9 : 01 A0 00 84 11 A9 02 8D  1C
0AC1 : 00 18 A9 04 2C 00 18 F0  BD
0AC9 : FB A9 00 8D 00 18 A2 03  9C
0AD1 : CA D0 FD A2 0A AD 00 18  15
0AD9 : 4A 6A 4A 66 85 0A 0A 66  55
0AE1 : 85 AD 00 18 4A 6A 4A 66  2E
0AE9 : 85 0A 0A 66 85 AD 00 18  B9
0AF1 : 4A 6A 4A 66 85 0A 0A 66  6D
0AF9 : 85 AD 00 18 8E 00 18 4A  36
0B01 : 6A 4A 66 85 0A 0A 66 85  70
0B09 : A5 85 91 30 45 11 85 11  F0
0B11 : C8 D0 AA 88 60 85 31 20  34
0B19 : 46 01 B1 30 48 20 4A 01  03
0B21 : 68 91 30 A5 11 F0 ED 4C  FB
0B29 : 43 E8 78 A9 0A 8D 00 18  71
0B31 : A2 00 88 D0 FD CA D0 FA  7F
0B39 : A9 0A 85 69 A9 00 85 30  87
0B41 : A9 03 20 A2 01 A9 04 20  76
0B49 : A2 01 4C 3D 04 A5 00 A2  D9
0B51 : 01 86 00 29 02 F0 10 A6  F0
0B59 : 98 86 32 20 2E 04 20 AF  A8
0B61 : 03 20 03 04 4C 69 F9 A2  F3
0B69 : 08 20 35 03 A2 0A 20 35  94
0B71 : 03 A2 08 20 F5 03 A2 0A  E2
0B79 : 20 F5 03 A5 8C 30 E8 4C  90
0B81 : 69 F9 86 32 86 98 A5 8C  AC
0B89 : 10 61 B5 83 D0 5D BD 31  79
0B91 : 04 20 A2 01 C8 A6 32 A9  4C
0B99 : 80 95 83 A5 80 95 00 A5  79
0BA1 : 81 95 01 B1 30 F0 1C 20  9F
0BA9 : 21 F1 A6 82 F6 B5 D0 02  21
0BB1 : F6 BB A0 00 A5 80 91 30  B2
0BB9 : C8 A5 81 91 30 A5 80 C5  A4
0BC1 : 22 F0 02 84 8C A6 32 A9  87
0BC9 : 00 85 30 85 33 85 2E 85  6C
0BD1 : 36 85 0C 85 50 A9 BB 85  CA
0BD9 : 34 BD 31 04 85 2F 20 E9  DF
0BE1 : F5 85 3A BD 32 04 20 A3  EA
0BE9 : F7 A6 32 B5 83 F0 52 20  C0
0BF1 : 2E 04 AD 00 1C 29 10 D0  7A
0BF9 : 03 4C 81 F5 20 10 F5 A2  E1
0C01 : 09 50 FE B8 CA D0 FA A9  7B
0C09 : FF 8D 03 1C AD 0C 1C 29  11
0C11 : 1F 09 C0 8D 0C 1C A9 FF  DF
0C19 : A2 05 8D 01 1C B8 50 FE  88
0C21 : B8 CA D0 FA A0 BB B1 0C  98
0C29 : 50 FE B8 8D 01 1C C8 D0  8E
0C31 : F5 B1 30 50 FE B8 8D 01  03
0C39 : 1C C8 D0 F5 50 FE 4C 00  DA
0C41 : FE 60 86 32 2C 24 04 86  58
0C49 : 98 B5 83 F0 F4 20 2E 04  CC
0C51 : 20 0A F5 A0 BB B1 0C 50  22
0C59 : FE B8 4D 01 1C D0 19 C8  65
0C61 : D0 F3 B1 30 50 FE B8 4D  18
0C69 : 01 1C D0 0C C8 C0 FD D0  5A
0C71 : F1 A6 32 A9 00 95 83 60  F3
0C79 : 4C C5 F6 BD 31 04 85 31  C9
0C81 : BD 32 04 85 0D 60 05 01  F3
0C89 : 06 04 58 20 19 F1 A9 84  7C
0C91 : D5 A7 F0 05 95 A7 20 42  B2
0C99 : D0 A9 40 8D F9 02 A9 01  58
0CA1 : 85 83 20 07 D1 90 03 4C  17
0CA9 : F8 CF 20 3E DE F6 B5 A9  28
0CB1 : 00 85 8B 85 8D A9 80 85  3B
0CB9 : 8C A5 80 85 06 A9 E0 85  25
0CC1 : 00 A5 00 30 FC F0 24 C9  15
0CC9 : 01 F0 EA A5 18 85 06 A5  C3
0CD1 : 19 85 07 A2 00 A9 B0 20  13
0CD9 : 7D D5 20 99 D5 A9 E2 20  F3
0CE1 : 7D D5 20 99 D5 A5 8C D0  E2
0CE9 : CC F0 CE 4C 23 DB A0 C0  80
0CF1 : A2 D6 20 6A D4 A0 D0 A2  2F
0CF9 : C0 20 6A D4 20 00 C0 48  94
0D01 : A0 D6 A2 C0 20 6A D4 68  46
0D09 : 60 AD FA FF 48 AD FB FF  E0
0D11 : 48 A9 FB 8D FA FF A9 D3  DC
0D19 : 8D FB FF 84 AD 86 15 A0  D9
0D21 : 00 84 AC 84 14 A2 06 B1  F1
0D29 : AC 91 14 C8 D0 F9 E6 AD  90
0D31 : E6 15 CA D0 F2 68 8D FB  0F
0D39 : FF 68 8D FA FF 60 B2 4C  95
```

MSE-Listing zu Hypra-Save. Beachten Sie bitte die Eingabehinweise auf Seite 18.