```
10 ; ************************************************
20 ; *                                              *
30 ; *        Discmon - Utilities fuer den Discmon  *
40 ; *                                              *
50 ; *      1986 by              Holger Jurlat      *
60 ; *                           Kaffeetwiete 14    *
70 ; *                           2082 Tornesch      *
80 ; *                                              *
90 ; ************************************************
100 ;
110 ;    Startadresse festlegen
120 ;
130         ORG  #7000
140         ENT  $
150 ;
160 ;    Programmstart
170 ;
180 ;
190 ;    RSX einbiden
200 ;
210         LD   BC,RSX             ; Adresse der RSX Befehlstabelle
220         LD   HL,KERNAL          ; 4 Bytes fuer Kernal
230         JP   #BCD1              ; Routine KL LOG EXT
240 ;
250 ;    4 Bytes fuer Kernal
260 ;
270 KERNAL: DEFS 4
280 ;
290 ;    RSX - Befehlstabelle
300 ;
310 RSX:    DEFW TABLE              ; Adresse der Befehlsworte
320         JP   SECR               ; SECREAD - Routine aufrufen
330         JP   SECW               ; SECWRITE - Routine aufrufen
340         JP   HEXD               ; HEXDUMP - Routine aufrufen
350         JP   SEARCH             ; SUCH - ROUTINE
360         JP   BUFFDO             ; BUFFDOWN - Routine
370         JP   FLPAR              ; FILEPAR - Routine
380 ;
390 ;    Befehlsworte
400 ;
410 TABLE:  DEFM "SECREA"
420         DEFB "D"+128
430         DEFM "SECWRIT"
440         DEFB "E"+128
450         DEFM "HEXDUM"
460         DEFB "P"+128
470         DEFM "SEARC"
480         DEFB "H"+#80
490         DEFM "BUFFDOW"
500         DEFB "N"+#80
510         DEFM "FILEPA"
520         DEFB "R"+#80
530         DEFB 0                  ; Ende der Tabelle
540 ;
550 ;    Routine SECREAD
560 ;
570 SECR:   LD   B,#84              ; Komandonummer #84 fuer Sector lesen
580         JP   RW                 ; Read / Write Routine aufrufen
590 ;
600 ;    Routine SECWRITE
610 ;
620 SECW:   LD   B,#85              ; Komandonummer fuer Sector schreiben
630 ;
640 ;    Read / Write Routine
650 ;
660 RW:     CP   3                  ; 3 Parameter ?
670         RET  NZ                 ; wenn nicht, dann zurueck
680         LD   H,(IX+1)           ; Bufferadresse laden
690         LD   L,(IX)
700         PUSH HL                 ; Bufferadresse auf Stack
710         LD   E,(IX+2)           ; Sectonummer laden
720         LD   D,(IX+4)           ; Tracknummer laden
730         PUSH DE                 ; DE auf Stack
740 ;
750 ; Komandonummer festlegen
760 ;
770         LD   A,B
780         LD   (COMAND),A
790 ;
800 ; Startadresse der Routine suchen
810 ;
820         LD   HL,COMAND          ; Adresse nach HL
830         CALL #BCD4              ; KL - FIND - COMMAND aufrufen
840         JR   NC,END             ; Nicht gefunden
850         LD   (ROUADR),HL        ; Adresse sichern
860         LD   A,C                ; Romnummer nach A
870         LD   (ROUADR+2),A       ; Romnummer sichern
880 ;
890 ; Disc - Routine aufrufen
900 ;
910         POP  DE                 ; Track und Sector vom Stack holen
920         LD   C,E                ; Sector nach C
930         LD   A,(DRIVE)          ; Drivenummer laden
940         LD   E,A                ; Drivenummer nach A
950         POP  HL                 ; Bufferadresse vom Stack
960         RST  #0018              ; Routine aufrufen
970         DEFW ROUADR             ; Adresse 3 Byte Block der Routine
980         RET                     ; Programm beenden
990 ;
1000 ; Platz fuer Parameter
1010 ;
1020 DRIVE:  DEFB 0                 ; Fuer Drive 0 oder 1
1030 COMAND: DEFB 0                 ; Fuer Komandonummer #84 oder #85
1040 ROUADR: DEFS 3                 ; 3 Bytes fuer Adresse
1050 ;
1060 ; ENDE
1070 ;
1080 END:    POP  HL                ; Elemente vom Stack vernichten
1090         POP  HL
1100         RET
1110 ;
1120 ;    HEXDUMP - Routine
1130 ;
1140 HEXD:   CP   1                 ; 1 Parameter ?
1150         RET  NZ                ; Zurueck, wenn nicht
1160         CALL KOPFZ             ; Kopfzeile ausdrucken
1170         LD   H,(IX+1)          ; Bufferadresse nach HL
1180         LD   L,(IX)
1190         LD   B,0               ; Zeilenzahl in B
1200 HL1:    INC  B                 ; Zeilenzahl inkrementieren
1210         LD   A,B               ; Zeilenzahl in A
1220         CP   17                ; Ist A = 17 ?
1230         RET  Z                 ; Wenn gleich, dann Ende
1240 ;
1250 ;    Ausgabe einer Zeile mit 16 Bytes
1260 ;
1270 ; Zuerst die Adresse
1280 ;
1290         LD   A,H               ; Hi - Byte in Acku
1300         LD   D,96 - kann gestrichen werden ; Differenz zum Ascii
1310         AND  1                 ; Mit 01 verknuepfen
1320         ADD  A,48
1330         CALL ZEIAUS            ; ASCII - Zeichen ausgeben
1340         LD   A,L               ; Lo Byte in Acku
1350         CALL ZIFAUS            ; Ausgabe einer Ziffer
1360         LD   A,9               ; Ein Zeichenvorschub
1370         CALL ZEIAUS            ; Zeichenausgabe
1380 ;
1390 ; Nun die Hexziffern
1400 ;
1410         LD   C,0               ; Zeichenzaehler loeschen
1420 HL2:    LD   A,(HL)            ; Byte ausgeben
1430         CALL ZIFAUS
1440         LD   A,9               ; Zeichenvorschub
1450         CALL ZEIAUS            ; Zeichenvorschub
1460         INC  HL                ; Adresse = Adresse + 1
1470         INC  C                 ; Zeichenzaehler incrementieren
1480         LD   A,C               ; Zeichenzaehler nach A
1490         CP   16                ; Schon 16 Zeichen ?
```

```
1500            JR   NZ,HL2                ; wenn nicht, naechstes Zeichen
1510 ;
1520 ; Ascii - Ausgabe
1530 ;
1540            SCF
1550            CCF
1560 ;
1570            LD   DE,16
1580            SBC  HL,DE                ; 16 von HL abziehen
1590            LD   C,0                  ; Zeichenzaehler auf 0
1600 HL3:       LD   A,(HL)               ; Zeichen nach A
1610            CP   127                  ; Zeichen > 127 ?
1620            JP   P,INVAUS             ; Wenn > 127
1630            CALL AUS                  ; Ausgabe des Zeichens
1640 NEA:       INC  HL
1650            INC  C
1660            LD   A,C
1670            CP   16
1680            JR   NZ,HL3
1690 ; Zeilenende
1700 ;
1710 ; Ende der Zeile
1720 ;
1730            JR   HL1
1740 ;
1750 ; Ausgabe des Ascii - Zeichens
1760 ;
1770 AUS:       CP   32
1780            JP   M,HL4                ; JA
1790            JR   ZEIAUS
1800 ;
1810 HL4:       LD   A,"."                ; Punkt ausgeben
1820            JR   ZEIAUS
1830 ;
1840 ; ZIFAUS
1850 ;
1860 ZIFAUS:    LD   D,A                  ; Zeichen zwischenspeichern
1870            RRA                       ; 4 * Nach rechts
1880            RRA
1890            RRA
1900            RRA
1910            CALL HL6
1920            LD   A,D
1930 HL6:       AND  15                   ; Hi Nibble loeschen
1940            ADD  A,48                 ; ZIFFERN 0 - 9
1950            CP   58                   ; Ist die Zifer von 0 - 9 ?
1960            JP   M,ZEIAUS
1970            ADD  A,7                  ; Bereich von A - F
1980 ;
1990 ; ZEIAUS - Routie
2000 ;
2010 ZEIAUS:    PUSH AF                   ; Alle Register retten
2020            PUSH BC
2030            PUSH DE
2040            PUSH HL
2050            CALL #BB5A                ; Print Vektor
2060            POP  HL                   ; Alle Register wiederholen
2070            POP  DE
2080            POP  BC
2090            POP  AF
2100            RET                       ; Ende der Routine
2110 ;
2120 ; INVAUS
2130 ;
2140 INVAUS:    LD   E,A                  ; Zeichen retten
2150            LD   A,24                 ; Inverse Modus einschalten
2160            CALL ZEIAUS
2170            LD   A,E                  ; Zeichen wiederholen
2180            AND  127                  ; Auf 7 Bit bringen
2190            CALL AUS                  ; Ausgabe des Zeichens
2200            LD   A,24                 ; Inverse Modus loeschen
2210            CALL ZEIAUS
2220            JR   NEA                  ; Naechstes Zeichen
2230 ;
2240 ; Kopfzeile ausgeben
2250 ;
2260 KOPF2:     PUSH AF                   ; Alle Register auf Stack
2270            PUSH BC
2280            PUSH DE
2290            PUSH HL
2300            LD   A,(LAENGE)           ; Laenge nach B
2310            LD   B,A
2320            LD   HL,KOPFS             ; Anfang nach HL
2330 KPFL:      LD   A,(HL)               ; Zeichen nach A
2340            CALL ZEIAUS               ; ZEICHEN AUSGEBEN
2350            DEC  B                    ; DECREMENTIERE B
2360            LD   A,B                  ; B NACH A
2370            CP   0                    ; IST A=0 ?
2380            JP   Z,KPFE               ; SPRINGE WENN JA
2390            INC  HL                   ; HL INKREMENTIEREN
2400            JP   KPFL
2410 ;
2420 KPFE:      POP  HL                   ; ALLE REGISTER ZURUECK
2430            POP  DE
2440            POP  BC
2450            POP  AF
2460            RET                       ; ZURUECK ZUM HAUPTPROGRAMM
2470 ;.
2480 LAENGE:    DEFB 59
2490 ;
2500 KOPFS:     DEFM "Hex  0  1  2  3  4  5  6  7"
2510            DEFM "  8  9  A  B  C  D  E  F ASCII"
2520            DEFB 13,10
2530 ;
2540 ; ***************** Suchroutine ****************
2550 ;
2560 ;
2570 ; @SEARCH,START,LAENGE,$STRING$,%RESULT
2580 ;
2590 SEARCH:    CP   4                    ; 3 Parameter ?
2600            RET  NZ
2610 ;
2620            LD   H,(IX+7)             ; Startadresse holen
2630            LD   L,(IX+6)
2640            LD   (BUFADR),HL
2650 ;
2660            LD   H,(IX+5)             ; Laenge holen
2670            LD   L,(IX+4)
2680            LD   (BUFLEN),HL
2690 ;
2700            LD   H,(IX+3)             ; Stringdescriptor holen
2710            LD   L,(IX+2)
2720 ;
2730            LD   A,(HL)               ; Stringlaenge holen
2740            LD   (STRLEN),A
2750 ;
2760            INC  HL                   ; Stringadresse holen
2770            LD   c,(HL)
2780            INC  HL
2790            LD   b,(HL)
2800            LD   (STRADR),BC
2810 ;
2820            CALL SUCH                 ; Suchroutine aufrufen
2830 ;
2840            LD   B,(IX+1)             ; Adresse Ruecgabe
2850            LD   C,(IX)
2860 ;
2870            CP   1
2880            JP   Z,REGRUK
2890 ;
2900            LD   A,0
2910            LD   (BC),A
2920            INC  BC
2930            LD   (BC),A
2940            RET
2950 ;
2960 REGRUK:    LD   A,L
2970            LD   (BC),A
2980            LD   A,H
2990            INC  BC
3000            LD   (BC),A
3010            RET
3020 ;
3030 ; ***************** Haupt - Suchroutine ***************
3040 ;
3050 ;
3060 ; Parameter :
3070 ;
3080 BUFADR: DEFS 2                       ; Startadresse Buffer
3090 BUFLEN: DEFS 2                       ; Laenge des Buffers
3100 STRADR: DEFS 2                       ; Startadresse String
3110 STRLEN: DEFS 1                       ; Laenge des Strings
3120 ;
3130 SUCH:      LD   HL,(BUFLEN)          ; Laenge nach HL
3140            LD   A,(STRLEN)           ; Wahre Laenge ausrechnen
3150            LD   C,A
3160            LD   B,0
3170            SCF
3180            CCF
3190            SBC  HL,BC
3200            INC  HL
3210            LD   B,H                  ; Wahre Laege nach BC
3220            LD   C,L
3230            LD   HL,(STRADR)          ; Stringadresse nach HL
3240            LD   A,(HL)               ; 1. Element in Acku
3250            LD   HL,(BUFADR)          ; Anfangsadresse Buffer nach HL
3260 COMP:      CPIR                      ; Suchen
3270            CALL Z,FOUND              ; Bei Gleichheit
3280            JP   PO,SRET              ; Wenn Block durchsucht, dann SRET
3290            JR   COMP                 ; Weiter Suchen
3300 ;
3310 FOUND:     PUSH AF                   ; Register auf Stack
3320            PUSH BC
3330            PUSH HL
3340            LD   A,(STRLEN)           ; Stringlaenge nach BC
3350            LD   C,A
3360            LD   B,0
3370            DEC  C                    ; 2. Element vergleichen
3380            JR   Z,OK                 ; falls String nur 1 Element
3390            LD   DE,(STRADR)          ; Startadresse 2. Element
3400            INC  DE
3410 COMP1:     LD   A,(DE)               ; Zeichen aus String in A
3420            CPI                       ; vergleichen
3430            INC  DE                   ; Zeiger erhoehen
3440            JR   NZ,BACK              ; wenn a<>Speicher, dann Zurueck
3450            JP   PE,COMP1             ; Wenn BC <> 0
3460 ;
3470 OK:        POP  HL                   ; Adresse holen
3480            DEC  HL
3490            POP  BC                   ; Register holen
3500            POP  AF
3510            POP  DE                   ; RET - Adresse vernichten
3520            LD   A,1                  ; Wert gefunden
3530            RET                       ; Zum Hauptprograma
3540 ;
3550 BACK:      POP  HL                   ; Register holen
3560            POP  BC
3570            POP  AF
3580            RET
3590 ;
3600 SRET:      LD   A,0                  ; Wert nicht gefunden
3610            RET
3620 ;
3630 ; ***************** Buffdown - Routine ***************
3640 ;
3650 ; @BUFFDOWN,Startadresse%
3660 ;
3670 BUFFDD:    CP   1                    ; 1 Parameter ?
3680            RET  NZ                   ; Wenn nicht, dann zurueck
3690            LD   H,(IX+1)             ; Adresse nach HL
3700            LD   L,(IX)
3710            PUSH HL
3720            LD   BC,512               ; Laenge nach BC
3730            SCF
3740            CCF                       ; Carry geloescht
3750            SBC  HL,BC                ; Adresse 512 Bytes tiefer berechnen
3760            LD   D,H                  ;  und ach DE speichern
3770            LD   E,L
3780            POP  HL                   ; Orginaladresse nach HL
3790            LDIR                      ; Block uebertragen
3800            RET
3810 ;
3820 ; Fileparameter - Routine
3830 ;
3840 ; @FILEPAR,$FILENAME$
3850 ;
3860 ; Ruecgabeparameter :
3870 ;
3880 FLTYP:     DEFS 1                    ; Filetyp
3890 FLSTA:     DEFS 2                    ; Startadresse des Files
3900 FLLEN:     DEFS 2                    ; Laenge des Files
3910 FLEIN:     DEFS 2                    ; Einsprungsadresse des Files
3920 ;
3930 ; Am Ende des Programms befindet sich der
3940 ; Buffer fuer die Blocks.
3950 ;
3960 FLPAR:     CP   1                    ; 1. Parameter ?
3970            RET  NZ                   ; Wenn Ja, dann zurueck
3980            LD   D,(IX+1)             ; Adresse Descriptor nach DE
3990            LD   E,(IX)
4000            LD   A,(DE)               ; Laenge nach B
4010            LD   B,A
4020            INC  DE                   ; DE zeigt auf Stringadresse
4030            LD   A,(DE)               ; 1. Low - Byte laden
4040            LD   L,A
4050            INC  DE
4060            LD   A,(DE)               ; 2. Hi - Byte laden
4070            LD   H,A
4080            LD   DE,(#A751)           ; ZK OPENIN BUFFER nach DE
4090            CALL #BC77                ; DISC IN OPEN aufrufen
4100            LD   (FLTYP),A            ; Filetyp abspeichern
4110            LD   (FLSTA),DE           ; Startadresse abspeicern
4120            LD   (FLLEN),BC           ; Filelaenge abspeicern
4130            LD   BC,#1A               ; Byte #1A im Header enthaelt Einsprung
4140            ADD  HL,BC                ; Dazuaddieren
4150            LD   C,(HL)               ; 1. Low - Byte Einsprung laden
4160            INC  HL
4170            LD   B,(HL)               ; 2. Hi - Byte laden
4180            LD   (FLEIN),BC           ; Einsprung abspeichern
4190 ;
4200            LD   DE,FLBLK             ; Buffer fuer Bloecke nach DE
4210 LOOP1:     LD   HL,#A719             ; Adresse der Blocktabelle
4220            LD   B,16                 ; Anzahl Bloecke / Extend
4230 LOOP2:     LD   A,(HL)               ; Block nach A
4240            LD   (DE),A               ; In Blockbuffer speichern
4250            OR   A                    ; Flags setzen
4260            JP   Z,FLEND              ; Ist A = 0, dann Ende der Routine
4270            INC  DE                   ; Zeiger auf Buffer erhoehen
4280            INC  HL                   ; Zeiger auf naechsten Block
4290            DJNZ LOOP2                ; Naechster Block
4300            LD   HL,(#A729)           ; AMSDOS vortaeuschen,
4310            LD   BC,#80               ; dass alle Records zum
4320            ADD  HL,BC                ; Extend gelesen wurden.
4330            LD   (#A729),HL           ; (#A729)+#80
4340            LD   HL,0                 ; AMSDOS einen leeren
4350            LD   (#A768),HL           ; Openin-Buffer voetaeuschen
4360            LD   HL,(#A753)           ; Pointer auf letztes
4370            LD   BC,2047              ; Zeichen im Buffer
4380            ADD  HL,BC                ; setzen.
4390            LD   (#A753),HL           ; (#A753)+2047
4400            LD   (#A796),BC           ; Belesene Zeichen
4410            push de
4420            CALL #BCB0                ; DISC IN CHAR aufrufen
4430            pop  de
4440            JR   LOOP1                ; Mit naechstem Extend weiter
4450 ;
4460 FLEND:     CALL #BC7D                ; DISC IN ABADON
4470            RET
4480 ;
4490 FLBLK:     DEFS 180
```