SoftwareService

Franzis'

Der CP/M-CCP

Franzis-Software-Service
Franzis-Verlag GmbH
Karlstraße 37–41
8000 München 2

Wilfried Gröhle
Programmierservice
Falcenstraße 34
8501 Dietzel
Tel.

p019

Der neue CCP enspricht den CP/M-Konventionen (Länge 800H, Startadresse BDOS-Untergrenze minus 800H) und sollte deshalb auf allen CP/M-Maschinen laufen. Die absolute Startadresse ist durch Änderung des .PHASE-Befehls und Neuassemblierung der Quelle problemlos zu ändern. Das Laden des Programms auf die Systemspuren muß individuell gelöst werden, z.B. mit SYSGEN. Bei den meisten CP/M-Maschinen beginnt der CCP auf Spur 0, Sektor 1. Da der neue CCP die gleiche Länge hat wie der alte, kann ein Überschreiben des restlichen Betriebssystems prinzipiell nicht auftreten.

Der Befehlspuffer des neuen CCP beginnt um drei Byte "eher" als beim alten, da ein nie benutzter Sprungbefehl entfernt wurde! Diese Verschiebung ist für solche Benutzer wesentlich, die den Puffer zwecks Autoload, Menüprogramm o.Ä. bereits systemmäßig mit einem Kommando füllen.

Die Änderungen im Einzelnen:

1) Residente Kommandos:

- DIR: liefert neben den Dateinamen auch die Information, ob die einzelnen Dateien schreibgeschützt sind (Ausgabe "R/O") oder nicht.
- ERA: Bei Angabe einer Dateigruppenbezeichnung werden alle nicht-schreibgeschützten zu dieser Gruppe gehörenden Dateien gelöscht, ohne daß der CCP bei der ersten R/O-Datei mit BDOS-Error "aussteigt". Findet der CCP keine einzige passender R/W-Datei, wohl aber Datei(en) vom Typ R/O, so meldet er sich mit "File(s) R/O".
- SAVE: Keine Änderung
- TYPE: Angabe von Dateigruppen ist zulässig. In diesem Fall gibt der CCP alle zu dieser Gruppe passenden Dateien mit Titel "File (drive):(filename) (typ)" auf die Konsole aus.
- REN: geht nur noch auf Dateien vom Typ DIR und R/W. Dateigruppenbezeichnungen sind zulässig (bei Dateiname-alt), falls zu der angegebenen Gruppe nur eine einzige Datei existiert.
- USER: keine Änderung.

2) Behandlung von .SUB-Dateien

Bei vorhandenem SUBMIT.COM können .SUB-Dateien wie .COM-Dateien behandelt werden (auch bezüglich Abkürzungen, s.u.), d.h. die Kommandos SUBMIT XYZ TTY: und XYZ TTY: haben die gleiche Wirkung, vorausgesetzt, es existiert keine .COM-Datei gleichen Namens (hier: XYZ.COM). In diesem Fall hätte die .COM-Datei Vorrang; das Programm würde geladen und normal ausgeführt.

3) Suche auf Laufwerke A

Wird zu einem eingegebenen Kommando die entsprechende .COM-Datei nicht auf dem aktuellen Laufwerk gefunden, so wird zusätzlich auf Laufwerk A weitergesucht, vorausgesetzt, daß

- ein Laufwerk nicht explizit angegeben wurde (z.B. B:DDT) und
- das aktuelle Laufwerk nicht ohnehin schon "A" ist.

Das hat u.a. den Vorteil, daß man auf "A" das Betriebssystem, den Editor, Compiler, Assembler etc. hat, evtl. mit Hardware-Schreibschutz, auf "B" die eigenen Programmentwicklungen, und man von B aus sämtliche Programme auf A ohne das lästige "A:" aufrufen kann (z.B. B ED DAT.PAS mit ED.COM auf Laufwerk A und DAT-PAS auf Laufwerk B). Das Ganze ist natürlich auf Laufwerk C, D usw. ausdehnbar, falls vorhanden.

Dies impliziert, daß in Laufwerk A immer eine Floppy eingelegt sein muß. Da es bei vielen CP/M-Rechnern ohnehin zwingend notwendig ist, auf "A" eine Floppy mitlaufen zu lassen, auch wenn nur auf "B" gearbeitet wird (Warmstart, Disk-Reset...), wurde diese Zusatzvorschrift in Kauf genommen.

4) Abkürzungen

Alle Kommandos dürfen abgekürzt werden, solange auf gleicher Hierarchiestufe (s.u.) Eindeutigkeit besteht. Dies soll anhand mehrerer Beispiele erläutert werden.

Es existieren auf Laufwerk A die Dateien ED.COM, DDT.COM, ERROR.COM und SAG.COM, sowie auf "B" die Dateien RENNE.COM, RENNE2.COM, S.COM und SAGEN.SUB.

Das aktuelle Laufwerk sei zunächst A. Es enthalte außerdem die CP/M-Datei SUBMIT.COM.

"DIR" läßt sich beispielsweise zu "DI" abkürzen; es existiert auf "A" keine Datei mit Typ .COM, die mit "DI" anfängt. Die Abkürzung "D" wäre jedoch nicht mehr eindeutig, da der DDT.COM auch mit "D" beginnt. Diese Eingabe würde folglich bemängelt.

"A:D" ginge aber! Wie schon in der alten CCP-Version sagt die explizite Angabe des Laufwerks dem CCP, daß es sich um ein externes Kommando handelt. DIR scheidet aus; in diesem Fall würde der DDT gestartet.

"B:D" gäbe Fehler. Es existiert keine passende Datei auf Laufwerk B.

"ED" läßt sich auch nicht weiter abkürzen ("ERA", "ERROR"); auch der Trick mit dem Laufwerk funktioniert hier nicht ("A:E" paßt auf "ED.COM" und auf "ERROR.COM" - Fehler!).

Sei nun das aktuelle Laufwerk B.

"ERA" läßt sich zu "ER" oder "E" abkürzen; auf "B" befindet sich keine Datei, die mit "E" beginnt. "ERR" ginge jedoch auf Laufwerk A; ERROR.COM würde ausgeführt.

"REN" läßt sich nicht abkürzen ("RENNE", "RENNE2"), ist aber eindeutig, denn das Kommando ist voll ausgeschrieben. In diesem Fall ist es egal, ob weitere Kommandos den gleichen Anfang haben. Ebenso läßt sich "RENNE" nicht abkürzen, wohl aber in der ausgeschriebenen Form normal starten.

"SUBMIT SAGEN" ist (s.o.) zunächst auf "SAGEN" abkürzbar. Auch "SAGE" wird noch als "SUBMIT SAGEN" interpretiert.
"SAG" jedoch geht auf A:SAG.COM und wird deshalb nicht als Fehler bemängelt.
"SA" hingegen geht auf das residente Kommando "SAVE", das gegenüber .SUB-Dateien Vorrang hat und deswegen auch nicht bemängelt wird.
"S" geht auf S.COM und wird ebenfalls nicht als Fehler bemängelt. Also Aufpassen!

"SUBMIT SAGE" ist dagegen unzulässig! Kommandozusätze dürfen <u>nicht abgekürzt</u> werden (wesentlich bei "ERA")!

5) Zusammenfassung der Hierarchie:

1	Residentes Kommando	(unabgekürzt)
2	Externes Kommando auf aktuellem Laufwerk	(unabgekürzt)
3	Externes Kommando auf Laufwerk A	(unabgekürzt)
4	SUBMIT-Datei auf aktuellem Laufwerk	(unabgekürzt)
5	Residentes Kommando	(abgekürzt)
6)	Externes Kommando auf aktuellem Laufwerk	(abgekürzt)
7)	Externes Kommando auf Laufwerk A	(abgekürzt)
8)	SUBMIT-Datei auf aktuellem Laufwerk	(abgekürzt)

Fazit: Bis auf das Umbenennen von SYS-Dateien sind alle Möglichkeiten des alten CCP auch im neuen enthalten.

Reinhard Drube

```
 1:          ASEG
 2:          ORG     100H
 3:          .Z80
 4:          TITLE         C   C   P       (07.07.1983)
 5:
 6:
 7: ;;;
```

```
 51:          JP      Z,RAUS
 52:          INC     HL
 53:          LD      A,(HL)
 54:          CP      CR
 55:          JP      Z,RAUS
 56:          CP      LF
 57:          JP      Z,RAUS
 58:          CP      'N'
 59:          JP      Z,RAUS
 60:          CP      'n'
 61:          JP      Z,RAUS
 62:          CP      'y'
 63:          JP      Z,OK
 64:          CP      'Y'
 65:          JR      Z,OK
 66:          LD      DE,WRTQTXT
 67:          JR      MSGSCH          ; ...NICHT VERSTANDEN!
 68:
 69: OK:      LD      HL,(BOOT+1)     ; STARTADRESSE BIOS+3
 70:          LD      DE,24
 71:          ADD     HL,DE
 72:          LD      (BSELDSK+1),HL
 73:          LD      DE,3
 74:          ADD     HL,DE
 75:          LD      (BSETTRK+1),HL
 76:          ADD     HL,DE
 77:          LD      (BSETSEC+1),HL
 78:          ADD     HL,DE
 79:          LD      (BSETDMA+1),HL
 80:          ADD     HL,DE
 81:          ADD     HL,DE
 82:          LD      (BWRITE+1),HL   ; DER REIHE NACH BENOETIGTE SPRUNGTABELLE KOPIEREN
 83:
 84: DRVSCH:  LD      DE,DRVQTXT
 85:          CALL    MSGOUT
 86:          CALL    KONEIN
 87:          LD      HL,BUF+1
 88:          LD      A,(HL)
 89:          DEC     A
 90:          JR      NZ,DRVSCH
 91:          INC     HL
 92:          LD      A,(HL)
 93:          SUB     "A"
 94:          JR      C,DRVSCH
 95:          LD      C,A
 96:          LD      B,0
 97:          CALL    BSELDSK         ; LAUFWERK AUSWAEHLEN
 98:          LD      A,H
 99:          OR      L
100:          JP      Z,NODSK         ; FALSCHES LAUFWERK
```

```
101:
102:          LD      DE,KOPFTXT
103:          CALL    MSGOUT
104:
105: SPURSCH:LD       DE,SPURQTXT
106:          CALL    MSGOUT
107:          CALL    KONEIN
108:          LD      HL,BUF+1
109:          LD      A,(HL)
110:          OR      A
111:          JR      NZ,SPURW
112:          INC     HL
113:          LD      (HL),"0"
114:          DEC     HL
115:          INC     A
116: SPURW:   DEC     A
117:          JR      NZ,SPURSCH
118:          INC     HL
119:          LD      A,(HL)
120:          SUB     "0"
121:          JR      C,SPURSCH
122:          CP      10
123:          JR      NC,SPURSCH
124:          LD      C,A
125:          LD      B,0
126:          CALL    BSETTRK          ; SPUR AUSWAEHLEN
127:
128: SECSCH: LD       DE,SECTQTXT
129:          CALL    MSGOUT
130:          CALL    KONEIN
131:          LD      HL,BUF+1
132:          LD      B,(HL)           ; LAENGE AKTUELLER PUFFER
133:          INC     HL
134:          INC     B
135:          DEC     B                ; <B> = 0 ?
136:          LD      A,2              ; --> VOREINSTELLUNG = 2
137:          JR      Z,STELLSEC       ; --> VOREINSTELLUNG TRITT IN KRAFT
138:
139:          LD      C,0
140: DEZSCH:  LD      A,(HL)
141:          SUB     "0"
142:          JR      C,SECSCH
143:          CP      10
144:          JR      NC,SECSCH
145:          LD      E,A
146:          LD      A,C
147:          AND     0E0H             ; WENN = 0 --> NOCH MIT 8 MULTIPLIZIERBAR
148:          JR      NZ,SECSCH
149:          LD      A,C
150:          RLCA
```

```
151:          RLCA
152:          RLCA                      ; <A> := <A> * 8
153:          ADD     A,C
154:          JR      C,SECSCH
155:          ADD     A,C               ; <A> := <A> * 10
156:          JR      C,SECSCH
157:          ADD     A,E               ; <A> := <A> * 10 + NEUE STELLE
158:          JR      C,SECSCH
159:          LD      C,A
160:          DJNZ    DEZSCH
161:
162: STELLSEC:LD      (ANFSEC),A        ; EINZUSTELLENDER STARTSEKTOR
163:          LD      HL,CCP
164:          LD      A,CCPLNG/128+1    ; ...AUFRUNDEN
165:
166: WRTSCH:  LD      C,L
167:          LD      B,H
168:          PUSH    BC                ; <BC> = AKTUELLE DMA-ADRESSE
169:          PUSH    AF
170:          CALL    BSETDMA
171:          POP     AF
172:          PUSH    AF                ; <A> = SEKTORENZAEHLER
173:          LD      B,A
174:          LD      A,CCPLNG/128+1
175:          SUB     B
176:          LD      HL,ANFSEC
177:          ADD     A,(HL)
178:          LD      C,A
179:          LD      B,0               ; <BC> = AKTUELL EINZUSTELLENDER SEKTOR
180:          CALL    BSETSEC
181:          LD      C,1               ;ALLE PUFFER SOFORT SCHREIBEN
182:          CALL    BWRITE            ; SEKTOR SCHREIBEN
183:          POP     AF
184:          POP     HL
185:          LD      BC,80H
186:          ADD     HL,BC
187:          DEC     A
188:          JR      NZ,WRTSCH
189:          JP      RAUS
190:
191:
192: BUF:     DB      80H,0             ; MAXIMALE PUFFERLAENGE = 128 BYTE
193:          DS      80H,0FFH
194:
195: KONEIN:  LD      DE,BUF
196:          LD      C,10
197:          JP      URBDOS
198:
199: MSGOUT:  LD      C,9
200:          JP      URBDOS
```

```
201:
202: KONFTXT:DB      CR,LF,"CCP ab Adresse 1000H im Arbeitsspeicher.",CR,LF
203: WRTQTXT:DB      CR,LF,"Soll der CCP sofort auf die Systemspuren geschrieben werden (Y/N
?$"
204: ENDTXT: DB      CR,LF,LF,"CCP noch im Arbeitsspeicher ab Adresse 1000H."
205:         DB      CR,LF,LF,"Bearbeitung beendet.",CR,LF,"$"
206: SPURQTXT:DB     CR,LF,LF,"Auf welche Systemspur soll der CCP geschrieben werden",CR,LF
207:         DB      "(leere Eingabe wird als Null interpretiert)?$"
208: SECTQTXT:DB     CR,LF,LF,"Ab welchem logischen Sektor soll der CCP geschrieben werden"
209:         DB      CR,LF,"(leere Eingabe wird als 2 interpretiert)?$"
210: DRVQTXT:DB      CR,LF,LF,LF,"Auf welches Laufwerk (A,B,C,...) soll geschrieben werden?$
211: KOPFTXT:DB      CR,LF,LF,LF,"Die folgenden Angaben koennen leer beantwortet werden;"
212:         DB      CR,LF,"es wird dann die Voreinstellung (Spur 0, Sektoren 2 mit 17"
213:         DB      CR,LF,"angenommen.",CR,LF,LF,"$"
214: NODTXT: DB      CR,LF,LF,LF,"****** Fehler: Gewaehltes Laufwerk nicht ansprechbar! ****
CR,LF,"$"
215: ZUTXT:  DB      CR,LF,LF,LF,"****** Fehler: Systemuntergrenze liegt im Ladebereich des
******",CR,LF,LF,LF,"$"
216:
217:
218:         ; DIE FOLGENDEN ADRESSEN WERDEN MIT DEN AKTUELLEN BIOS-ADRESSEN
219:         ; BEIM PROGRAMMSTART UEBERSCHRIEBEN.
220: BSETDMA:JP      0
221: BWRITE: JP      0
222: BSETSEC:JP      0
223: BSETTRK:JP      0
224: BSELDSK:JP      0
225:
226: ANFSEC: DB      0
227:
228:
229:
230: NODSK:  LD      DE,NODTXT
231:         CALL    MSGOUT
232:
233: RAUS:   LD      DE,ENDTXT
234:         CALL    MSGOUT
235:         JP      BOOT
236:
237: ZUTIEF: LD      DE,ZUTXT
238:         CALL    MSGOUT
239:         JP      BOOT
240:
241:
242:         DS      30
243: STACK:  DS      0
244:
245: GRUND:  .PHASE  CCP
246: CCPA:   JP      ANFANG
247:         JP      CLRBUF
248: MAXPUF: DB      7FH              ; MAXIMALE BEFEHLSPUFFERLAENGE
249: AKTPUF: DB      0                ; AKTUELLE BEFEHLSPUFFERLAENGE
250: BEFPUF: DB      '
```

44 Blanke

```
251:            DB          '                   (C) 19.11.1983, Reinhard Drube
252:            DS          40,20H                ; PUFFER MIT BLANKS FUELLEN
253: BEFPUFPOI:DW           BEFPUF
254:
255: PUTFILE:LD             BC,FILETXT
256:            CALL        PUTSKIPMSG
257:            CALL        PUTDISK
258:            CALL        PUTNAME
259:            CALL        CRLF
260: CRLF:      LD          A,0DH                 ; CR
261:            CALL        CONOUT
262:            LD          A,0AH                 ; LINE-FEED
263: CONOUT:    PUSH        BC                    ; CONOUT RETTET REGISTER <HL>, <BC>
264:            AND         7FH
265:            LD          E,A
266:            LD          C,2
267:            CALL        HBDOS
268:            POP         BC
269:            RET
270:
271: RESDSK:    LD          C,0DH
272: HBDOS:     PUSH        HL
273:            CALL        URBDOS                ; ALLE BDOS-AUFRUFE IM CCP RETTEN REGISTER <HL>
274:            POP         HL
275:            OR          A
276:            RET
277:
278: PUTNAME:LD             A,':'
279:            CALL        CONOUT
280:            CALL        SENDSPACE
281:            LD          B,11
282:            LD          HL,FCB
283: PUTSCH:    INC         HL
284:            LD          A,(HL)
285:            CALL        CONOUT
286:            LD          A,B
287:            CP          4                     ; FILENAME KOMPLETT --> FILETYP
288:            JR          NZ,$+5
289:            CALL        SENDSPACE             ; NACH FILENAMEN BLANK SENDEN
290:            DJNZ        PUTSCH
291: SENDSPACE:LD           A,' '
292:            JR          CONOUT
293:
294: PUTSKIPMSG:CALL CRLF
295: PUTMSG:    LD          A,(BC)                ; AUSGABE VON <<BC>>,<<BC>+1>,... BIS <<BC>+N> = 0
296:            OR          A
297:            RET         Z
298:            INC         BC
299:            CALL        CONOUT
300:            JR          PUTMSG
```

```
301:
302: SELDFTDRV:LD     A,(AKTDISK)
303: SELDSK: LD       E,A
304:         LD       C,0EH
305:         JR       HBDOS
306:
307: OPCCPFILE:LD     DE,CCPFCB
308: OPENFILE:XOR     A
309:         PUSH     DE
310:         POP      IX
311:         LD       (IX+32),A        ; CR AUF NULL
312:         LD       (IX+12),A        ; EX AUF NULL
313:         LD       C,0FH
314: BDOSCALL:CALL    HBDOS
315:         INC      A
316:         RET                       ; BEI OPEN + CLOSE MIT Z-FLAG = 1 ZURUECK, FALLS FEHLER
SONST Z=0
317:
318: CLOSEFILE:LD     C,10H
319:         JR       BDOSCALL
320:
321: FCBKOP: LD       DE,FCB           ; <HL> ZEIGT AUF QUELL-FCB
322:         LD       BC,32
323:         LDIR
324:         RET
325:
326: DELFILE:LD       C,13H
327:         JR       HBDOS
328:
329: RDSEQ:  LD       C,14H
330:         JR       HBDOS
331:
332: WRTSEQ: LD       C,15H
333:         JR       HBDOS
334:
335: MAKEFILE:LD      C,16H
336:         JR       BDOSCALL
337:
338: RENAME: LD       C,17H
339:         JR       HBDOS
340:
341: RETUSRNR:LD      E,0FFH
342: USRCOD: LD       C,20H            ; <E>=0FFH <--> GET USER CODE, SONST SET USER-CODE AUF

343:         JR       HBDOS
344:
345: RETCDS: LD       C,19H
346:         JR       HBDOS
347:
348: STDDMA: LD       DE,DMA
349: SETDMA: LD       C,1AH
350:         JR       USRCOD+2
```

```
351:
352: SCFIRST:LD      DE,CCPFCB
353:         LD      C,11H
354:         JR      SNEXT+2
355: SNEXT:  LD      C,12H
356:         CALL    BDOSCALL
357:         RET     Z                  ; KEINE WEITERE DATEI GEFUNDEN
358:         PUSH    HL
359:         DEC     A
360:         RRCA
361:         RRCA
362:         RRCA
363:         ADD     A,DMA              ; DMA + 32 * <A>
364:         LD      L,A
365:         LD      H,0                ; <HL> = DMA + 32 * <A>
366:         CALL    FCBKOP
367:         LD      A,(CCPFCB)
368:         LD      (FCB),A            ; LAUFWERK EXPLIZIT EINSETZEN
369:         POP     HL
370:         RET
371:
372: GETFIRST:XOR    A
373:         LD      (DIRZ),A
374:         JR      SCFIRST
375: GETNEXT:CALL    SCFIRST
376:         RET     Z                  ; FALLS KEINE DATEI GEFUNDEN
377:         LD      IX,DIRZ            ; ZAEHLER FUER BEREITS GELESENE/GELOESCHTE FILES
378:         INC     (IX)               ; ...ERHOEHEN...
379:         LD      A,(IX)             ; ...UND LADEN.
380:         SUB     (HL)               ; BEI ERA: FALLS DATEIEN BEREITS GELOESCHT WURDEN
381:         JR      Z,GETEND
382:         LD      B,A
383: SNSCHL: PUSH    BC
384:         CALL    SNEXT              ; (DIRZ)-<HL> FILES UEBERLESEN
385:         POP     BC
386:         RET     Z                  ; BEI FEHLER MIT Z=1 ZURUECK
387:         DJNZ    SNSCHL
388: GETEND: OR      H                  ; <HL> = ADRESSE VON FCB+... <> 0
389:         RET                        ; MIT Z=0 ZURUECK
390:
391: READIFPOS:LD    C,0BH              ; STATUS?
392:         CALL    HBDOS
393:         RET     Z                  ; KEIN ZEICHEN STEHT ZUR EINGABE AN.
394:         LD      C,1
395:         JR      WRTSEQ+2           ; LESEN SONST.
396:
397:
398:         ;-----------------------------------------------------------
399:         ; UP:   KONVERTIERT KLEINBUCHSTABEN --> GROSSBUCHSTABEN.
400:         ;-----------------------------------------------------------
```

```
401: KLEINGROSS:CP    'a'                 ; KONVERTIERT KLEINBUCHSTABEN NACH GROSS
402:          RET     C
403:          CP      '('                 ; 7B
404:          RET     NC
405:          AND     '_'                 ; 5F
406:          RET
407:
408:
409: GETCOMAND:LD     A,(SUBMERK)
410:          OR      A
411:          JR      Z,NIXSUBMIT
412:          LD      DE,SUBMITFCB
413:          CALL    OPENFILE            ; SUBMIT-FILE EROEFFNEN
414:          JR      NZ,SUBEXT
415:          LD      HL,SUBMITFCB
416:          LD      (HL),0              ; AUF AKTUELLEM LAUFWERK SUCHEN
417:          EX      DE,HL
418:          CALL    OPENFILE
419:          JR      Z,NIXSUBMIT         ; ...GIBT'S NICHT!
420: SUBEXT:  LD      A,(SUBMITFCB+15);   RC = AUGENBLICKLICHE DATEIGROESSE + 1
421:          DEC     A
422:          LD      (SUBMITFCB+32),A;   LIES LETZTEN SATZ ($$$.SUB IST VON HOECHSTER
423:                                      ; ZU NIEDERWERTIGERER SATZNUMMER ZU LESEN!!!)
424:          LD      DE,SUBMITFCB
425:          CALL    RDSEQ               ; LETZTEN AKTUELLEN DATEISATZ LESEN ( = NAECHSTER)
426:          JR      NZ,NIXSUBMIT        ; LESEFEHLER!!
427:          LD      DE,AKTPUF
428:          LD      HL,DMA
429:          PUSH    HL
430:          POP     BC
431:          LDIR                        ; KOPIERE 128 BYTE VON 80H NACH AKTPUF (=DURCH FILE-SA
SIMULIERTE
432:          LD      HL,SUBMITFCB+14     ; KONSOLEINGABE)
433:          LD      (HL),0              ; S2 WIEDER AUF NULL SETZEN (ALS WAERE NOCH NICHT GELE
N)
434:          INC     HL
435:          DEC     (HL)                ; RC := RC -1, DAMIT WIRD DIE DATEI UM EINEN...
436:          LD      DE,SUBMITFCB        ; ...SATZ BEIM NECHSTEN "CLOSE-FILE" KLEINER...
437:          CALL    CLOSEFILE           ; ...WAS NACH ALLEN GELESENEN SAETZEN ZU EINEM...
438:          JR      Z,NIXSUBMIT         ; ...GEWOLLTEN LESEFEHLER FUEHRT.
439:                                      ; JEDER SATZ EINER $$$.SUB-DATEI ENTSPRICHT
440:                                      ; GENAU EINEM KONSOLKOMMANDO; IM RESTLICHEN
441:                                      ; PUFFER STEHT "SCHMUTZ".
442:          LD      BC,BEFPUF
443:          CALL    PUTMSG
444:          CALL    READIFPOS           ; ABBRUCH VON "SUBMIT" DURCH TASTENANSCHLAG
445:          JR      Z,CONVTOUPPER       ; WENN KEINE KONSOLEINGABE ANSTEHT
446:          JR      KILLSUB
447:
448: NIXSUBMIT:CALL   KILLSUBFILE
449:          LD      C,0AH
450:          LD      DE,MAXPUF
```

```
451:            CALL    HBDOS           ; READ CONSOLE BUFFER = LIES KOMMANDO VON KONSOLE EI
452: CONVTOUPPER:LD     HL,AKTPUF
453:            LD      B,(HL)
454:            INC     B
455:            JR      CONVIN
456: CONVSCHL:LD        A,(HL)
457:            CALL    KLEINGROSS
458:            LD      (HL),A
459: CONVIN:    INC     HL
460:            DJNZ    CONVSCHL
461:            LD      (HL),B          ; LETZTES BYTE NACH EINGABE AUF 0
462: BEFINIT:LD         HL,BEFPUF
463:            LD      (BEFPUFPOI),HL  ; ZEIGER AUF PUFFERANFANG
464:            RET
465:
466:
467: KILLSUBFILE:LD     HL,SUBMERK      ; (SUBMERK) = 0 <--> KEINE SUBMIT-DATEI VORHANDEN
468:            XOR     A
469:            CP      (HL)
470:            RET     Z
471:            LD      (HL),A
472:            LD      DE,SUBMITFCB
473:            JP      DELFILE         ; SUBMIT-FILE LOESCHEN
474:
475: LAUFWA:
476:            LD      (HL),1          ; <HL> = POINTER AUF CCPFCB
477:            CALL    SCFIRST
478:            CALL    NZ,NEXTTEST     ;PRUEFEN, OB WEITERE DATEI DER BE-
479:                                    ;SCHREIBUNG ENTSPRICHT. FALLS JA - FEHLER
480:            RET     NZ
481:            LD      (HL),0          ;ZURUECK AUF VOREINGEST. LW.
482:            RET
483:
484:
485: NEXTTEST:
486:            CALL    SNEXT
487:            JR      NZ,MELDEERROR   ;FEHLER, FALLS NOCH KEINE DATEI GEFUNDEN
488:            XOR     A
489:            DEC     A               ;A=FF, Z-FLAG AUF "NZ"
490:            RET
491:
492:
493:            ;-----------------------------------------------------------
494:            ; UP:   GIBT IM FEHLERFALL BEFEHLSPUFFER MIT FRAGEZEICHEN
495:            ;       WIEDER AUS, BRICHT EVTL. SUBMIT-FOLGEN AB.
496:            ;-----------------------------------------------------------
497: MELDEERROR:CALL    CRLF            ; SENDE CR UND LINE-FEED
498:            LD      BC,BEFPUF
499:            CALL    PUTMSG
500:            LD      A,'?'
```

```
501:          CALL     CONOUT
502:          CALL     CRLF              ; SENDE CR UND LINE-FEED
503: KILLSUB:CALL      KILLSUBFILE
504:          JP       PUTPROMPTNG
505:
506:
507:
508: TRENNZ: DB        '=_.:;<>'
509: CHECKINPUT:LD     A,(DE)            ; KEHRT MIT Z=1 ZURUECK, FALLS TRENNZEICHEN ODER PUFFEI
NDE
510:          OR       A
511:          RET      Z
512:          CP       ' '
513:          RET      Z
514:          JR       C,MELDEERROR      ; STEUERZEICHEN ABWUERGEN
515:          PUSH     HL
516:          PUSH     BC
517:          LD       HL,TRENNZ
518:          LD       BC,7
519:          CPIR
520:          POP      BC
521:          POP      HL
522:          RET
523:
524:
525: GETNUMBER:CALL    SCANINPUT
526:          LD       A,(LAUFWCODE)
527:          OR       A
528:          JR       NZ,MELDEERROR
529:          LD       HL,CCPFCB+1
530:          LD       BC,0B00H          ; <C> = 0, <B> = 11
531: DEZIMBIN:LD       A,(HL)            ; ERSTE EINGABE (SAVE/USER) MUSS ZIFFERNEINGABE ENTHAL
N
532:          CP       ' '
533:          JR       Z,IFSPACE
534:          INC      HL
535:          SUB      '0'               ; 30
536:          CP       0AH
537:          JR       NC,MELDEERROR     ; NICHT DEZIMAL
538:          LD       D,A
539:          LD       A,C
540:          AND      0E0H
541:          JR       NZ,MELDEERROR     ; NICHT MEHR MIT 8 MULTIPLIZIERBAR
542:          LD       A,C
543:          RLCA
544:          RLCA
545:          RLCA
546:          ADD      A,C
547:          JR       C,MELDEERROR
548:          ADD      A,C
549:          JR       C,MELDEERROR
550:          ADD      A,D
```

```
551:            JR      C,MELDEERROR
552:            LD      C,A                 ; <C> := 10 * <C> + EINGABE
553:            DJNZ    DEZIMBIN
554:            RET
555: IFSPACE:LD         A,(HL)              ; LIEST RESTLICHE ERSTEINGABE, OB STELLE <> BLANK VORHAN
DEN
556:            CP      ' '
557:            JR      NZ,MELDEERROR
558:            INC     HL
559:            DJNZ    IFSPACE
560:            LD      A,C
561:            RET                         ; KEHRT MIT BINAERWERT VON EINGEG. ZIFFERNFOLGE IN <A> Z
URUECK.
562:
563:
564: SUCHINPUT:LD       A,(DE)              ; SUCHT DEN BEFEHLSPUFFER NACH ERSTEM ZEICHEN <> BLANK A
B
565:            OR      A
566:            RET     Z                   ; EINGABEENDE ERREICHT
567:            CP      ' '
568:            RET     NZ
569:            INC     DE
570:            JR      SUCHINPUT
571:
572:
573: SCFILE:    CALL    CHECKINPUT
574:            JR      Z,FILLBLANK
575:            INC     HL
576:            CP      '*'                 ; DATEIGRUPPENBEZEICHNUNG?
577:            JR      NZ,KEINSTERN
578:            LD      (HL),'?'            ; AUS  *  MACH  ???...
579:            JR      $+4
580: KEINSTERN:LD       (HL),A
581:            INC     DE
582:            DJNZ    SCFILE
583: IGNORREST:CALL     CHECKINPUT          ; TRENNZEICHEN, PUFFERENDE...?
584:            RET     Z
585:            INC     DE
586:            JR      IGNORREST
587: FILLBLANK:INC      HL
588:            LD      (HL),' '            ; RESTLICHEN FILE-TYP MIT BLANKS AUFFUELLEN
589:            DJNZ    FILLBLANK
590:            RET
591: SCANINPUT:LD       DE,0
592: SCANSCHL:LD        HL,CCPFCB
593:            ADD     HL,DE
594:            PUSH    HL
595:            XOR     A
596:            LD      (LAUFWCODE),A
597:            LD      DE,(BEFPUFPOI)
598:            CALL    SUCHINPUT
599:                                        ; <DE> = POINTER AUF CCPFCB+...
600:            LD      A,(DE)              ; ERSTER EINGEGEBENER CHARACTER
```

CCP-0503.MAC

```
601:            OR      A
602:            JR      Z,NOEXPLDRV     ; --> EINGABE WAR LEER.
603:            SUB     "A"-1           ; AUS "A" WIRD 1, AUS "B" WIRD 2 USW USW.
604:            LD      B,A             ; <B> := LAUFWERK-CODE
605:            INC     DE
606:            LD      A,(DE)
607:            CP      ':'             ; KOMMT NACH LAUFWERK DER OBLIGATORISCHE DOPPELPUNKT?
608:            JR      Z,DISKEXPL      ; JA!
609:            DEC     DE              ; SONST: POINTER ZURUECK AUF EINGABEANFANG
610: NOEXPLDRV:XOR      A
611:            JR      SCANFILENAME
612: DISKEXPL:LD        A,B
613:            LD      (LAUFWCODE),A
614:            INC     DE
615: SCANFILENAME:LD    B,8             ; LAENGE DATEIBEZEICHNUNG
616:            LD      (HL),A          ; IN FCB DAS AKTUELLE LAUFWERK EINSETZEN
617:            CALL    SCFILE
618:            LD      B,3
619:            CP      '.'
620:            JR      Z,POINT
621:            CALL    FILLBLANK       ; KEIN FILE-TYP ANGEGEBEN
622:            JR      DREIBT00
623: POINT:     INC     DE
624:            CALL    SCFILE
625: DREIBT00:LD        B,3
626:            INC     HL              ; FCB+12,13,14 AUF 0 SETZEN
627:            LD      (HL),0
628:            DJNZ    $-3
629:            LD      (BEFPUFPOI),DE
630:            POP     HL              ; <HL> = POINTER AUF CCPFCB + .... (SIEHE OBEN)
631:            LD      BC,0B00H        ; <C> AUF NULL, <B> AUF 11
632:            LD      A,'?'
633: GRUPPSCHL:INC      HL
634:            CP      (HL)
635:            JR      NZ,$+3
636:            INC     C
637:            DJNZ    GRUPPSCHL
638:            LD      A,C
639:            OR      A
640:            RET                     ; Z=1, FALLS DATEI EINDEUTIG BESTIMMT, SONST Z=0
641:
642:
643: RESIDENT:DB        'DIR '
644:            DB      'REN '
645:            DB      'SAVE'
646:            DB      'ERA '
647:            DB      'TYPE'
648:            DB      'USER'
649: ASKRESID:LD        DE,RESIDENT
650:            LD      C,0
```

```
651: CPRESI: LD      A,C              ; UEBERPRUEFT, OB RESIDENTE KOMMANDOS VOLIEGEN
652:         CP      6
653:         RET     NC
654:         LD      HL,CCPFCB+1
655:         LD      B,4              ; LAENGE DER RESIDENTEN KOMMANDOS
656: SCANRESID:LD    A,(DE)
657:         CP      (HL)
658:         JR      Z,RESIDOK
659:         LD      A,(ZWEITER)
660:         OR      A
661:         JR      Z,TESTNEXT
662:         LD      A,'?'
663:         CP      (HL)             ; ABKUERZUNG?
664:         JR      NZ,TESTNEXT      ; NEIN --> NAECHSTES RESID. KOMMANDO
665:         PUSH    BC
666:         CALL    SCFIRST          ; IST ABKUERZUNG EINDEUTIG?
667:         POP     BC
668: JMERROR:JP      NZ,MELDEERROR    ; NEIN --> FEHLER UND RAUS
669:         LD      A,C
670:         RET
671: RESIDOK:INC     DE
672:         INC     HL
673:         DJNZ    SCANRESID
674:         LD      A,(HL)
675:         CP      ' '              ; BESTAND KOMMANDO NUR AUS VIER BUCHSTABEN?
676:         JR      NZ,CPNEXTRESID
677:         LD      A,C              ; DIR = 0, REN = 1, SAVE = 2    USW USW
678:         RET                      ; KEHRT MIT # RESIDENTES KOMMANDO ZURUECK.
679: TESTNEXT:INC    DE
680:         DJNZ    TESTNEXT
681: CPNEXTRESID:INC C
682:         JR      CPRESI
683:
684:
685:
686: CLRBUF: XOR     A
687:         LD      (AKTPUF),A       ; PUFFER AUF "LEER" SETZEN.
688: ANFANG: LD      SP,CSTACK
689:         PUSH    BC               ; <C> = USER-NR  +  DRIVE-NR  AUS SPEICHERZELLE 0004
690:         LD      A,C
691:         RRA
692:         RRA
693:         RRA
694:         RRA
695:         AND     0FH
696:         LD      E,A
697:         CALL    USRCOD           ; SET USER 0...15
698:         CALL    RESDSK
699:         CALL    RESDSK
700: ; ZWEITER DISKRESET HAT SICH DOCH ALS NOTWENDIG ERWIESEN.
```

CLRBUF

warmboot --> ~~ANFANG~~

kaltboot --> Anfang.

```
701: ; ES GIBT CP/M SYSTEME, BEI DENEN DER ERSTE RESET ANDERS ALS
702: ; ALLE FOLGENDEN BEHANDELT WIRD .
703: ; ZUM TESTEN, OB NOTWENDIG, KANN MAN EIN SUB-FILE AUF A ERZEUGEN,
704: ; DAS MAN DANN VON B AUS ALS KOMMANDO AUFRUFT. STEIGT DER
705: ; RECHNER DANN BEI EINEM DISKRESET NICHT AUS, IST ANZUNEHMEN,
706: ; DASS DER ZEITE WEGFALLEN KANN.
707:
708:          LD      DE,SUBMITFCB
709:          LD      C,11H
710:          CALL    BDOSCALL          ; SUCHE $$$.SUB AUF LAUFWERK A
711:          POP     BC
712:          PUSH    AF
713:          LD      A,C
714:          AND     0FH
715:          LD      (AKTDISK),A
716:          CALL    NZ,SELDSK
717:          POP     AF
718:          LD      A,1               ; CODE FUER LAUFWERK A IM FCB
719:          JR      NZ,LDVAR          ; FALLS $$$.SUB AUF <A> EXISTIERTE
720:          LD      DE,SUBMITFCB
721:          LD      C,11H
722:          CALL    BDOSCALL
723:          JR      Z,LDVAR           ; KEINE $$$.SUB AUF AKTUELLEM LAUFW.
724:          LD      A,(AKTDISK)
725:          INC     A
726: LDVAR:   LD      (SUBMERK),A
727:          LD      (SUBMITFCB),A
728:          LD      A,(AKTPUF)
729:          OR      A
730:          JR      NZ,CCPPUVOLL
731: PUTPROMPTNG:LD   SP,CSTACK
732:          CALL    CRLF              ; SENDE CR UND LINE-FEED
733:          LD      HL,CURDRV
734:          LD      A,(HL)
735:          AND     0F0H
736:          LD      B,A
737:          LD      A,(AKTDISK)
738:          OR      B
739:          LD      (HL),A
740:          CALL    RETCDS            ; RETURN CURRENT DISK (0 = A, 1 = B, ETC...)
741:          ADD     A,'A'
742:          CALL    CONOUT
743:          LD      A,'>'
744:          CALL    CONOUT
745:          CALL    GETCOMAND
746: CCPPUVOLL:CALL   STDDMA
747:          CALL    SCANINPUT         ; SCHREIBT IN CCPFCB DEN ZU LADENDEN FILE-NAMEN.
748:          JP      NZ,JMERROR        ; FALLS "?" IM DATEINAMEN AUFTRATEN
749:          XOR     A
750:          LD      (ZWEITER),A
```

```
751: ZWEITERLAUF:LD   A,(LAUFWCODE)
752:          OR      A
753:          JP      NZ,EXTERN
754:          CALL    ASKRESID
755:          LD      HL,TABELLE
756:          LD      E,A
757:          LD      D,0
758:          ADD     HL,DE
759:          ADD     HL,DE
760:          LD      A,(HL)
761:          INC     HL
762:          LD      H,(HL)
763:          LD      L,A
764:          JP      (HL)
765:
766: TABELLE:DW       DIR
767:          DW      REN
768:          DW      SAVE
769:          DW      ERA
770:          DW      TYPE
771:          DW      USER
772:          DW      EXTERN
773:
774:
775: ERREAD: LD       BC,RDRTXT
776:          JR      NOFILE+3
777: RDRTXT: DB       'Read Error',0
778:
779: NOFILE: LD       BC,NOFTXT
780:          JP      PUTSKIPMSG
781: NOFTXT: DB       'No File',0
782:
783:
784: DIR:    CALL     SCANINPUT        ; SCHREIBT IN (LAUFWCODE) DAS EINGEG. LAUFWERK, SONST 0
785:          LD      HL,CCPFCB+1
786:          LD      A,(HL)
787:          CP      ' '
788:          JR      NZ,FILEEXPL      ; FALLS NICHT ALLE DATEIEN AUSGE-"DIR"-T WERDEN SOLLEN
789:          LD      B,0BH
790:          LD      (HL),'?'         ; CCPFCB TOTAL MIT '??????????' FUELLEN
791:          INC     HL
792:          DJNZ    $-3
793: FILEEXPL:LD      E,0
794:          PUSH    DE
795:          CALL    SCFIRST          ; ERSTE DATEI SUCHEN (DIRECTORY DURCHLESEN)
796:          CALL    Z,NOFILE         ; DISK LEER.
797:          JR      Z,DIRENDE        ; ...UND RAUS
798: DIRSCHL:LD       HL,FCB+10
799:          BIT     7,(HL)           ; $SYS-FILE?
800:          JR      NZ,SYSFILE
```


CCP-0603.MAC ✓

```
801:            DEC     HL
802:            BIT     7,(HL)          ; $R/O-FILE?
803:            LD      BC,BLANKS
804:            JR      Z,RWRITE
805:            LD      BC,RONL
806: RWRITE:    POP     DE
807:            LD      A,E
808:            INC     E
809:            PUSH    DE
810:            PUSH    BC
811:            AND     3               ; VIER FILES PRO AUSGEGEBENER ZEILE
812:            JR      NZ,PUTBLK
813:            CALL    CRLF
814:            CALL    PUTDISK
815: PUTBLK:    CALL    PUTNAME
816:            POP     BC
817:            CALL    PUTMSG
818: SYSFILE:   CALL    READIFPOS       ; ABBRUCH DURCH KONSOLEINGABE?
819:            JR      NZ,DIRENDE
820:            CALL    SNEXT
821:            JR      NZ,DIRSCHL
822: DIRENDE:   POP     DE
823:            JP      CSCNINPUT       ;  R A U S !
824: PUTDISK:   LD      A,(CCPFCB)
825:            OR      A
826:            JR      NZ,PUTW
827:            CALL    RETCDS
828:            INC     A
829: PUTW:      ADD     A,"A"-1
830:            CALL    CONOUT
831:            RET
832: FILETXT:   DB      CR,LF,LF,"File"
833: BLANKS:    DB      "    ",0
834:
835:
836: REN:       CALL    SCANINPUT
837:            JP      NZ,RENFEHL      ; FALLS "?" IM DATEINAMEN AUFTRAT
838:            LD      A,(LAUFWCODE)
839:            PUSH    AF
840:            CALL    SCFIRST
841:            JR      NZ,GIBTSSCHON
842:            LD      HL,CCPFCB
843:            LD      DE,CCPFCB+16    ; FCB NACH FCB2 KOPIEREN
844:            LD      BC,16
845:            LDIR
846:            LD      DE,(BEFPUFPOI)
847:            CALL    SUCHINPUT
848:            CP      '='
849:            JR      Z,ISTGLEICH
850:            CP      '_'
```

```
851:            JR      NZ,RENFEHL
852: ISTGLEICH:INC      DE
853:            LD      (BEFPUFPOI),DE
854:            CALL    SCANINPUT       ; ALTEN FILE-NAMEN BESORGEN
855:            POP     BC              ; <B> = DRIVE-NEU
856:            LD      HL,LAUFWCODE
857:            LD      A,(HL)
858:            OR      A
859:            JR      Z,DEFAULT       ; LAUFWERK NICHT EXPLIZIT
860:            CP      B
861:            LD      (HL),B
862:            JR      NZ,RENFEHL      ; KEINE EINHEITLICHE LAUFWERKANGABE
863: DEFAULT:LD         (HL),B
864:            LD      A,B
865:            LD      (CCPFCB),A
866:            CALL    SCFIRST
867:            JR      Z,HAMWANICH
868:            LD      HL,FCB+1
869:            LD      DE,CCPFCB+1
870:            LD      BC,15
871:            LDIR
872:            CALL    SNEXT
873:            JR      NZ,RENFEHL
874:            LD      HL,FCB+9
875:            BIT     7,(HL)          ; $R/O ?
876:            JR      NZ,RENFEHL
877:            INC     HL
878:            BIT     7,(HL)          ; $SYS ?
879:            JR      NZ,RENFEHL
880:            LD      DE,CCPFCB
881:            CALL    RENAME
882:            JR      $+5
883: HAMWANICH:CALL     NOFILE
884:            JR      SAVEOK+3
885: RENFEHL:JP         MELDEERROR
886: GIBTSSCHON:LD      BC,FILEEXISTXT
887:            CALL    PUTSKIPMSG
888:            JR      HAMWANICH+3
889: FILEEXISTXT:DB     'File exists',0
890:
891:
892: SAVE:      CALL    GETNUMBER       ; <A> = ANZAHL 1/4-KILOBYTES
893:            PUSH    AF
894:            CALL    SCANINPUT       ; FILE-NAMEN BESORGEN
895:            JR      NZ,RENFEHL
896:            LD      DE,CCPFCB
897:            PUSH    DE
898:            CALL    DELFILE         ; ALTEN GLEICHNAMIGEN FILE LOESCHEN
899:            POP     DE
900:            CALL    MAKEFILE        ; ...UND NEU ERZEUGEN
```

```
901:            JR      Z,DISKFULL
902:            XOR     A
903:            LD      (CCPFCB+32),A
904:            POP     AF
905:            LD      L,A
906:            LD      H,0
907:            ADD     HL,HL           ; PRO SATZ 1/8 KILOBYTE
908:            LD      DE,TBASE
909: SAVESCHL:LD        A,H             ; <HL> = ZAEHLER DER ZU SCHREIBENDEN DATEISAETZE
910:            OR      L
911:            JR      Z,SAVEFERTIG
912:            DEC     HL
913:            PUSH    HL
914:            LD      HL,DMA
915:            ADD     HL,DE
916:            CALL    SETDMA
917:            LD      DE,CCPFCB
918:            CALL    WRTSEQ
919:            EX      DE,HL           ; <DE> = DMA+80H
920:            POP     HL
921:            JR      NZ,DISKFULL
922:            JR      SAVESCHL
923: SAVEFERTIG:LD      DE,CCPFCB
924:            CALL    CLOSEFILE
925:            JR      NZ,SAVEOK
926: DISKFULL:LD        BC,NOSPACETXT
927:            CALL    PUTSKIPMSG
928: SAVEOK: CALL       STDDMA
929:            JP      CSCNINPUT
930: NOSPACETXT:DB      'No Space',0
931:
932:
933:
934: ERA:       CALL    SCANINPUT       ; KEHRT MIT # "?" IN <A> ZURUECK.
935:            CP      11              ; 11 FRAGEZEICHEN <--> ALLE DATEIEN.
936:            JR      NZ,NICHTALLE
937:            LD      BC,ALLKILLQU
938:            CALL    PUTSKIPMSG
939:            CALL    GETCOMAND
940:            LD      HL,AKTPUF
941:            DEC     (HL)
942: JPUTPRO:JP         NZ,PUTPROMPTNG  ; WENN EINGABE AUS MEHR ALS EINEM BUCHSTABEN BESTAND
943:            INC     HL
944:            LD      A,(HL)
945:            CP      'Y'
946:            JR      NZ,JPUTPRO      ; WENN EINGABE <> "Y"
947:            INC     HL
948:            LD      (BEFPUFPOI),HL  ; POINTER AUF PUFFERENDE --> LEERER, GELESENER PUFFE
949: NICHTALLE:LD       HL,RONLY
950:            LD      (HL),0
```

```
951:            CALL    GETFIRST
952:            JR      NZ,GEFUNDEN
953:            CALL    NOFILE
954:            JR      ERAENDE
955: GEFUNDEN:LD        A,(FCB+9)
956:            RLA
957:            JR      C,SGNXT         ; $R/O
958:            LD      DE,FCB
959:            CALL    DELFILE
960:            INC     (HL)
961: SGNXT:     CALL    GETNEXT
962:            JR      NZ,GEFUNDEN
963:            LD      A,(HL)
964:            OR      A
965:            JR      NZ,ERAENDE      ; WENN NICHT ALLE BETRACHTETEN FILES R/O
966:            LD      BC,RONLYMSG
967:            CALL    PUTSKIPMSG
968: ERAENDE:JP         CSCNINPUT
969: RONLY:     DB      0
970: RONLYMSG:DB        'File(s) '
971: RONL:      DB      'R/O ',0
972: ALLKILLQU:DB       'All (Y/N)?',0
973:
974:
975: TYPE:      CALL    SCANINPUT
976:            PUSH    AF
977:            CALL    GETFIRST        ; ERSTE DATEI IN FCB SCHREIBEN
978:            JR      Z,OPERROR       ; KEINE DATEI GEFUNDEN
979:            POP     AF
980:            JR      Z,TYPEW
981: TY2:       CALL    PUTFILE
982: TYPEW:     LD      DE,FCB
983:            CALL    OPENFILE
984:            JR      Z,OPERROR
985:            CALL    CRLF            ; SENDE CR UND LINE-FEED
986: TYPELES:LD         DE,FCB
987:            CALL    RDSEQ
988:            JR      NZ,LESEFEHL
989:            LD      HL,DMA
990: TYPESCHL:LD        A,H
991:            OR      A               ; <HL> KLEINER ALS 100H?
992:            JR      NZ,TYPELES
993:            LD      A,(HL)
994:            INC     HL
995:            CP      1AH             ; END OF FILE?
996:            JR      Z,LESEFEHL
997:            CALL    CONOUT
998:            CALL    READIFPOS       ; ABBRUCH DURCH KONSOLEINGABE
999:            JR      NZ,ERAENDE      ; RAUS!
1000:           JR      TYPESCHL
```

```
1001: LESEFEHL:LD      HL,TYPE-1        ; POINTER AUF WERT "NULL"
1002:          CALL    GETNEXT
1003:          JR      Z,ERAENDE
1004:          JR      TY2
1005:          CALL    ERREAD
1006: OPERROR:CALL     SELDFTDRV
1007:          JP      MELDEERROR       ; GIBT FEHLERHAFTES FILE WIEDER AUF KONSOLE AUS
1008:
1009:
1010:
1011: USER:    CALL    GETNUMBER
1012:          CP      16               ; USER-NR. 0...15
1013:          JR      NC,OPERROR+3
1014:          LD      E,A
1015:          LD      A,(CCPFCB+1)
1016:          CP      ' '
1017:          JR      Z,OPERROR+3
1018:          CALL    USRCOD
1019:          JP      CSCNINPUT
1020:
1021:
1022: EXTERN:  LD      A,(ZWEITER)
1023:          OR      A                ; ZWEITER DURCHLAUF (MIT ABKUERZUNG)?
1024:          JR      NZ,GRUPPE
1025:          LD      A,(CCPFCB+1)
1026:          CP      ' '
1027:          JR      NZ,NAME
1028:          LD      A,(LAUFWCODE)
1029:          OR      A
1030:          JR      Z,NAME-3         ;  R  A  U  S  !  !
1031:          DEC     A
1032:          LD      (AKTDISK),A      ; EINGABE WAR  A>B:  ODER SO AEHNLICH
1033:          CALL    SELDSK
1034:          JP      CSCNINPUT        ;  R  A  U  S  !  !
1035:
1036: NAME:    LD      DE,CCPFCB+9
1037:          LD      A,(DE)
1038:          CP      ' '
1039:          JP      NZ,RENFEHL       ;.TYP "COM" IST VOREINSTELLUNG UND DARF NICHT ANGEGE
WERDEN
1040:          LD      (ZWEITER),A      ; FUER ZWEITEN DURCHLAUF <> 0 SETZEN
1041:          CALL    FCBTEST          ;SCHAUEN, OB PASSENDE DATEIEN VORHANDEN
1042:          LD      B,8
1043: SCHL:    INC     HL
1044:          LD      A,(HL)
1045:          CP      ' '
1046:          JR      NZ,$+4
1047:          LD      (HL),'?'
1048:          DJNZ    SCHL
1049:          CALL    GETCOM
1050:          JP      ZWEITERLAUF
```

```
1051:
1052:
1053: GRUPPE: CALL     FCBTEST
1054:         JP       MELDEERROR
1055:
1056:
1057: FCBTEST:LD       HL,CCPFCB
1058:         CALL     GETCOM
1059:         CALL     SCFIRST
1060:         CALL     NZ,NEXTTEST
1061:         JR       NZ,GIBTSOPEN
1062:         CALL     ASKLW
1063:         CALL     NZ,LAUFWA
1064:         JR       NZ,GIBTSOPEN
1065:         CALL     GETSUB
1066:         CALL     SCFIRST
1067:         CALL     NZ,NEXTTEST
1068:         JR       NZ,SUBMITINS
1069:         CALL     ASKLW
1070:         CALL     NZ,LAUFWA
1071:         RET      Z
1072: SUBMITINS:POP    HL
1073:         LD       HL,(BEFPUFPOI)
1074:         LD       DE,BEFPUF+17
1075:         LD       BC,6EH
1076:         LD       A,L
1077:         DEC      A
1078:         SUB      E
1079:         JR       C,LDDROK
1080:         JR       Z,LDDRWEI
1081:         LD       C,A
1082:         PUSH     HL
1083:         POP      DE
1084:         DEC      DE
1085:         JR       DOLDDR
1086:
1087: LDDROK:
1088:         ADD      HL,BC
1089:         LD       DE,BEFPUF+7FH
1090:         LD       C,6FH
1091: DOLDDR:
1092:         LDDR
1093: LDDRWEI:
1094:         LD       HL,FCB+8
1095:         LD       C,8
1096:         LDDR
1097:         LD       A,':'
1098:         LD       (DE),A
1099:         LD       A,(HL)
1100:         OR       A
```

```
1101:            JR      NZ,DRVOK
1102:            LD      A,(AKTDISK)
1103:            INC     A
1104: DRVOK:     ADD     A,'A'-1
1105:            DEC     DE
1106:            LD      (DE),A
1107:            DEC     DE
1108:            LD      HL,SUBTXT+6
1109:            LD      C,7
1110:            LDDR
1111:            CALL    BEFINIT
1112:            JP      CCPPUVOLL
1113:
1114: GIBTSOPEN:POP      DE
1115:            EX      DE,HL
1116:            INC     DE
1117:            LD      HL,FCB+1
1118:            LD      BC,8
1119:            LDIR
1120:            CALL    OPCCPFILE
1121:            LD      DE,TBASE
1122: LOADSCHL:PUSH      DE
1123:            CALL    SETDMA
1124:            LD      DE,CCPFCB
1125:            CALL    RDSEQ
1126:            JR      NZ,ENDOFFILE
1127:            POP     HL
1128:            LD      DE,80H
1129:            ADD     HL,DE           ; DMA UM 80H "NACH OBEN"
1130:            LD      DE,CCP          ; OBERGRENZE LADBARES PROGRAMM = CCP-ANFANG
1131:            PUSH    HL
1132:            SBC     HL,DE           ; OBERGRENZE UEBERSCHRITTEN?
1133:            POP     DE
1134:            JR      NC,ZUGROSS
1135:            JR      LOADSCHL
1136: ENDOFFILE:POP      HL
1137:            DEC     A
1138:            JR      NZ,ZUGROSS      ; DIRECTORY-CODE = 1  <--)  END OF FILE
1139:            LD      HL,(BEFPUFPOI)
1140:            PUSH    HL
1141:            CALL    SCANINPUT       ; WEITERE EINGABEN VERBRATEN UND FUER BENUTZERPROGRAMM
UFBEREITEN
1142:            LD      DE,16
1143:            CALL    SCANSCHL        ; INPUT WIRD AUF CCPFCB+16 GESCHRIEBEN (ZWEITES FCB)
1144:            XOR     A
1145:            LD      (CCPFCB+32),A
1146:            LD      HL,CCPFCB
1147:            CALL    FCBKOP
1148:            LD      (DE),A          ; 33-TES BYTE ABSPEICHERN (=0)
1149:            POP     HL
1150:            LD      B,0
```

```
1151:          LD       DE,DMA+1
1152: BEFTODMA:LD       A,(HL)
1153:          LD       (DE),A
1154:          OR       A
1155:          JR       Z,ALLKOPIERT
1156:          INC      B
1157:          INC      HL
1158:          INC      DE
1159:          JR       BEFTODMA
1160:
1161: ALLKOPIERT:LD     A,B
1162:          LD       (DMA),A          ; ERSTES BYTE GIBT ANZAHL DEFINIERTER TEXTZEICHEN AB DI
+1 AN.
1163:          CALL     CRLF             ; SENDE CR UND LINE-FEED
1164:          CALL     STDDMA
1165:          CALL     TBASE            ; AUFRUF BENUTZERPROGRAMM
1166:          LD       SP,CSTACK
1167:          CALL     SELDFTDRV
1168: JPUTPR:  JP       PUTPROMPTNG
1169: ZUGROSS:LD        BC,BADLOADMSG
1170:          CALL     PUTSKIPMSG
1171: CSCNINPUT:CALL    SCANINPUT
1172:          LD       HL,LAUFWCODE
1173:          OR       (HL)
1174:          JP       NZ,MELDEERROR
1175:          JR       JPUTPR
1176:
1177: ASKLW:   LD       A,(LAUFWCODE)
1178:          OR       A
1179:          JR       NZ,ASKLW2
1180:          LD       A,(AKTDISK)
1181:          OR       A
1182:          RET
1183: ASKLW2:  XOR      A                ; ...UND UMGEKEHRT
1184:          RET
1185:
1186:
1187: GETCOM:  LD       DE,"OC"          ; REIHENFOLGE VON "C" UBD "O" !!!!
1188:          LD       A,"M"
1189:          JR       PUTTYP
1190: GETSUB:  LD       DE,"US"
1191:          LD       A,"B"
1192: PUTTYP:  LD       (CCPFCB+9),DE
1193:          LD       (CCPFCB+11),A
1194:          RET
1195:
1196:
1197: SUBTXT:  DB       "SUBMIT "
1198: BADLOADMSG:DB     'Bad Load',0
1199:
1200:
```

```
1201:          DS       30
1202: CSTACK:  DS       0
1203:
1204: SUBMERK:DB        0
1205: SUBMITFCB:DB      0,'$$$      SUB'
1206:          DS       21,0
1207: CCPFCB:  DS       33,0
1208: AKTDISK:DB        0
1209: LAUFWCODE:DS      2,0
1210: ZWEITER:DB        0                   ; =0 BEI ERSTEM DURCHLAUF (KEINE ABK.), <>0 SONST
1211: DIRZ:    DB       0
1212: ENDE:    END
```

1984

© Franzis-Verlag GmbH, München

Bearbeitet vom Franzis-Software-Service
Für den Inhalt verantwortlich: Dipl.-Inform. Jürgen Plate
Sämtliche Rechte, besonders das Übersetzungsrecht, an Text und Bildern vorbehalten. Fotomechanische Vervielfältigung nur mit Genehmigung des Verlages. Jeder Nachdruck - auch auszugsweise - und jegliche Wiedergabe der Bilder sind verboten.

Druck: Franzis-Druck GmbH, Karlstraße 35, 8000 München 2
Printed in Germany. Imprimé en Allemagne.