

DIGITAL RESEARCH

Post Office Box 579, Pacific Grove, California 93950, (408) 373-3403

CONSOLE COMMAND PROCESSOR (CCP)

CP/M VERSION _____

COPYRIGHT © 1976

DIGITAL RESEARCH

P. O. BOX 579

PACIFIC GROVE, CA. 93950

SER. # _____

CCP

```

0000 = BIAS EQU 000H
        ASSEMBLY LANGUAGE VERSION. OF THE CP/M CONSOLE COMMAND PROCESSOR
        VERSION. 1.1 OCTOBER, 1976
        COPYRIGHT (C) 1976
        DIGITAL RESEARCH

0000 = FALSE EQU 0000H
0001 = TRUE EQU NOT FALSE
0002 = TESTING EQU FALSE ; TRUE IF DEBUGGING
0003 = CCPLOC EQU BIAS ; BASE ADDRESS OF CCP
0004 = BDOSL EQU 900H+BIAS ; BASE OF DOS

        IF TESTING
        ORG 100H
        MVI C,0
TRAN EQU 01000H ; TRANSIENT PROGRAM AREA BASE (TPA)
TRANM EQU 1400H ; LAST TRANSIENT LOCATION. + 1
        ENDDIF

        IF NOT TESTING
        ORG CCPLOC
        EQU 100H
        EQU $
        ENDDIF

0000 C3E702 JMP CCPSTART
0003 434F505952 DB 'COPYRIGHT (C) 1976, DIGITAL RESEARCH'

0004 = DISKA EQU 0004H ; DISK ADDRESS FOR CURRENT DISK
0005 = BDOS EQU 0005H ; PRIMARY BDOS ENTRY POINT
0006 = BUFF EQU 0006H ; DEFAULT BUFFER
0007 = FCB EQU 0007H ; DEFAULT FILE CONTROL BLOCK

0001 = RCHARF EQU 1 ; READ CHARACTER FUNCTION
0002 = PCHARF EQU 2 ; PRINT CHARACTER FUNCTION
0003 = PBUF EQU 9 ; PRINT BUFFER FUNCTION
0004 = RBUF EQU 10 ; READ BUFFER FUNCTION
0005 = BREAKF EQU 11 ; BREAK KEY FUNCTION
0006 = LIFTF EQU 12 ; LIFT HEAD FUNCTION. (SHUGART SA3900 ONLY)
0007 = INITF EQU 13 ; INITIALIZE BDOS FUNCTION
0008 = SELF EQU 14 ; SELECT DISK FUNCTION
0009 = OPENF EQU 15 ; OPEN FILE FUNCTION
0010 = CLOSEF EQU 16 ; CLOSE FILE FUNCTION
0011 = SEARF EQU 17 ; SEARCH FOR FILE FUNCTION
0012 = SEARNF EQU 18 ; SEARCH FOR NEXT FILE FUNCTION
0013 = DELF EQU 19 ; DELETE FILE FUNCTION
0014 = DREADF EQU 20 ; DISK READ FUNCTION
0015 = DWRTF EQU 21 ; DISK WRITE FUNCTION
0016 = MAKEF EQU 22 ; FILE MAKE FUNCTION
0017 = RENF EQU 23 ; RENAME FILE FUNCTION

```

CP/M VERSION _____
 COPYRIGHT © 1976
 DIGITAL RESEARCH
 P. O. BOX 579
 PACIFIC GROVE, CA. 93950
 SER. # _____

```

0018 = LOGF EQU 24 ; RETURN. LOGIN VECTOR
0019 = CSELF EQU 25 ; RETURN. CURRENTLY SELECTED DRIVE NUMBER
001A = DMAF EQU 26 ; SET DMA ADDRESS

000D = CR EQU 13 ; CARRIAGE RETURN
000A = LF EQU 10 ; LINE FEED
000F = LA EQU 5FH ; LEFT ARROW
001A = EOFIL EQU 1AH ; END OF FILE
0002 = NDISKS EQU 2 ; NUMBER OF DISKS

        UTILITY PROCEDURES
PRINTCHAR: MOV E,A! MVI C,PCHARF! JMP BDOS
PRINTBC: ; PRINT CHARACTER, BUT SAVE B,C REGISTERS
        PUSH B! CALL PRINTCHAR! POP B! RET

        CRLF: MVI A,CR! CALL PRINTCHAR
        MVI A,LF! JMP PRINTCHAR

        PRINT: ; PRINT STRING STARTING AT B,C UNTIL NEXT DOLLAR SYMBOL
        PUSH B! CALL CRLF! POP B! MVI C,PBUFF! JMP BDOS

        INITIALIZE: MVI C,INITF! JMP BDOS

        SELECT: MOV E,A! MVI C,SELF! JMP BDOS

        OPEN: ; OPEN THE FILE GIVEN. BY D,E
        MVI C,OPENF! CALL BDOS! STA DCNT! INR A! RET

        OPENC: ; OPEN. COMFCB
        XRA A! STA CONREC ; CLEAR NEXT RECORD TO READ
        LXI D,COMFCB! JMP OPEN

        CLOSE: ; CLOSE THE FILE GIVEN. BY D,E
        MVI C,CLOSEF! CALL BDOS! STA DCNT! INR A! RET

        SEARCH: ; SEARCH FOR THE FILE GIVEN BY D,E
        MVI C,SEARF! CALL BDOS! STA DCNT! INR A! RET

        SEARCHN: ; SEARCH FOR THE NEXT OCCURRENCE OF THE FILE GIVEN. BY D,E
        MVI C,SEARNF! CALL BDOS! STA DCNT! INR A! RET

        SEARCHCOM: ; SEARCH FOR COMFCB FILE
        LXI D,COMFCB! JMP SEARCH

        DELETE: ; DELETE THE FILE GIVEN. BY D,E
        MVI C,DELF! JMP BDOS

        DISKREAD: ; READ THE NEXT RECORD FROM THE FILE GIVEN. BY D,E
        MVI C,DREADF! CALL BDOS! ORA A! RET

        DISKREADC:

```

CP/M VERSION _____
 COPYRIGHT © 1976
 DIGITAL RESEARCH
 P. O. BOX 579
 PACIFIC GROVE, CA. 93950
 SER. # _____

```

> 0096 112007C38F      ;READ THE COMFCB FILE
                        LXI D,COMFCB! JMP DISKREAD
>
> DISKWRITE:
> ;WRITE THE NEXT RECORD TO THE FILE GIVEN. BY D,E
> 009C 0E15CD0500      MVI C,DURITF! CALL BDOS! ORA A! RET
>
> MAKE:
> ;CREATE THE FILE GIVEN. BY D,E
> 00A3 0E16CD0500      MVI C,MAKEFI! CALL BDOS! STA DCHT! RET
>
> RENAM:
> ;RENAME THE FILE GIVEN BY D,E
> 00AC 0E17C30500      MVI C,RENFI! JMP BDOS
>
> TRANSLATE:
> ;TRANSLATE CHARACTER INH REGISTER A TO UPPER CASE
> 00B1 FE61D0          CPI 61H! RC ;RETURN. IF BELOW LOWER CASE A
> 00B4 FE70D0          CPI 70H! RNC ;RETURN IF ABOVE LOWER CASE Z
> 00B7 E65FC9          ANI 5FH! RET ;TRANSLATED TO UPPER CASE
>
> READCOM:
> ;READ THE NEXT COMMAND INTO THE COMMAND BUFFER
> ;CHECK FOR SUBMIT FILE
> 00BA 3AFE06B7CA      LDA SUBMIT! ORA A! JZ NOSUB
>
> ;SCANNING A SUBMIT FILE
> ;CHANGE DRIVES TO OPENH AND READ THE FILE
> 00C1 3AC007B73E      LDA CDISK! ORA A! MVI A,0! CNZ SELECT
> 00CA 3A0E073D      LDA SUBRC! DCR A ;READ LAST RECORD(S) FIRST
> 00CE 321F07          STA SUBCR ;CURRENT RECORD TO READ
> 00D1 11FF06CD8F      LXI D,SUBFCB! CALL DISKREAD ;END OF FILE IF LAST RECORD
> 00D7 C20D01          JNZ NOSUB
>
> ;DISK READ IS OK, TRANSFER TO COMBUF
> 00DA 1142072180      LXI D,COMLEN! LXI H,BUFF! MVI B,120! CALL MOVE0
> ;LINE IS TRANSFERRED, CLOSE THE FILE WITH A
> ;DELETED RECORD
> 00E5 210E0735      LXI H,SUBRC! DCR H ;ONE LESS RECORD
> 00E9 11FF06CD66      LXI D,SUBFCB! CALL CLOSE! JZ NOSUB
> ;CLOSE WENT OK, RETURN. TO ORIGINAL DRIVE
> 00F2 3AC007B7C4      LDA CDISK! ORA A! CNZ SELECT
> 00F9 0E09114307      MVI C,PBUFF! LXI D,COMBUF! CALL BDOS
> 0101 CD3301CA18      CALL BREAK*KEY! JZ N0READ
> 0107 CD5501C3FC      CALL DEL$SUB! JMP CCP ;BREAK KEY DEPRESSED
>
> NOSUB: ;NO SUBMIT FILE! CALL DEL$SUB
> ;TRANSLATE TO UPPER CASE, STORE ZERO AT END
> 010D CD5901          MVI C,PBUFF! LXI D,MAXLEN! CALL BDOS
> 0110 0E0A114107      N0READ: ;ENTER HERE FROM SUBMIT FILE
> ;SET THE LAST CHARACTER TO ZERO FOR LATER SCANS
> 0116 21420746      LXI H,COMLEN! MOV B,M ;LENGTH IS IN. B
> 011C 2370B7          READCOM0: INX H! MOV A,B! ORA A ;END OF SCAN?
> 011F CA2B017E      JZ READCOM1! MOV A,M ;GET CHARACTER AND TRANSLATE
> 0123 CDB1007705      CALL TRANSLATE! MOV M,A! DCR B! JMP READCOM0
>
> READCOM1: ;END OF SCAN, H,L ADDRESS END OF COMMAND
> 012B 77              MOV M,A ;STORE A ZERO
> 012C 21430722C3      LXI H,COMBUF! SHLD COMADDR ;READY TO SCANN TO ZERO
> 0132 C9              RET
>
> BREAK*KEY:
> ;CHECK FOR A CHARACTER READY AT THE CONSOLE
> 0133 0E0BCD0500      MVI C,BREAKFI! CALL BDOS
>

```

```

0138 B7C0          ORA A! RZ
013A 0E01CD0500      MVI C,RCHARFI! CALL BDOS ;CHARACTER CLEARED
013F B7C9          ORA A! RET
>
> LIFTHEAD:
> ;LIFT THE HEAD ON THE SA3900
> 0141 0E0CC30500      MVI C,LIFTFI! JMP BDOS
>
> CSELECT:
> ;GET THE CURRENTLY SELECTED DRIVE NUMBER TO REG-A
> 0146 0E19C30500      MVI C,CSELF! JMP BDOS
>
> MOVE:
> ;MOVE DATA FROM D,E TO H,L FOR C BYTES
> 014B 1A7713230D      LDAX D! MOV M,A! INX D! INX H! DCR C! JNZ MOVE! RET
>
> SETDMA:
> ;SET DMA ADDRESS TO D,E
> 0154 0E1AC30500      MVI C,DMAFI! JMP BDOS
>
> DEL$SUB:
> ;DELETE THE SUBMIT FILE, AND SET SUBMIT FLAG TO FALSE
> 0159 21FE067EB7      LXI H,SUBMIT! MOV A,M! ORA A! RZ ;RETURN. IF NO SUB FILE
> 015F 3600          MVI M,0 ;SUBMIT FLAG IS SET TO FALSE
> 0161 AFCD4C00      XRA A! CALL SELECT ;ON. DRIVE A TO ERASE FILE
> 0165 11FF06CD8A      LXI D,SUBFCB! CALL DELETE
> 016B 3AC007C34C      LDA CDISK! JMP SELECT ;BACK TO ORIGINAL DRIVE
>
> SERIALIZE:
> ;CHECK SERIALIZATION
> 0171 11A0022100      LXI D,SERIAL! LXI H,BDOSL! MVI B,6 ;CHECK SIX BYTES
> 0179 1ABEC24403      SER0: LDAX D! CMP M! JNZ BADSERIAL
> 017E 132305C279      INX D! INX H! DCR B! JNZ SER0
> 0184 C9              RET ;SERIAL NUMBER IS OK
>
> COMERR:
> ;ERROR IN. COMMAND STRING STARTING AT POSITION
> ;'STADDR' AND ENDING WITH FIRST DELIMITER
> 0185 CD3300          CALL CRLF ;SPACE TO NEXT LINE
> 0188 2AC507          LHLD STADDR ;H,L ADDRESS FIRST TO PRINT
> COMERR0: ;PRINT CHARACTERS UNTIL BLANK OR ZERO
> 018B 7EFE20CA9E      MOV A,M! CPI ' '! JZ COMERR1; NOT BLANK
> 0191 B7CA9E01      ORA A! JZ COMERR1; NOT ZERO, SO PRINT IT
> 0195 E5CD2700E1      PUSH H! CALL PRINTCHAR! POP H! INX H
> 019D C38B01          JMP COMERR0; FOR ANOTHER CHARACTER
> COMERR1: ;PRINT QUESTION. MARK,AND DELETE SUB FILE
> 019E 3E3FCD2700      MVI A,'?'! CALL PRINTCHAR
> 01A3 CD3300CD59      CALL CRLF! CALL DEL$SUB
> 01A9 C3FC02          JMP CCP ;RESTART WITH NEXT COMMAND
>
> ;FCB SCAN. AND FILL SUBROUTINE (ENTRY IS AT FILLFCB BELOW)
> ;FILL THE COMFCB, INDEXED BY A (0 OR 16)
> ;SUBROUTINES
> DELIM: ;LOOK FOR A DELIMITER
> 01AC 1AB7C0          LDAX D! ORA A! RZ ;NOT THE LAST ELEMENT
> 01AF FE20DA501      CPI ' '! JC COMERR ;NON. GRAPHIC
> 01B4 C0              RZ ;TREAT BLANK AS DELIMITER
> 01B5 FE3DC0          CPI '='! RZ
> 01B8 FE5FC0          CPI LAI! RZ ;LEFT ARROW
> 01BB FE2EC0          CPI '.'! RZ
> 01BE FE3AC0          CPI '!'! RZ

```

```

01C1 FE3BC8      CPI '!' RZ
01C4 FE3CC8      CPI '<' RZ
01C7 FE3EC8      CPI '>' RZ
01CA C9          RET      ;DELIMITER NOT FOUND

DEBLANK: ;DEBLANK THE INPUT LINE
LDAX D! ORA A! RZ ;TREAT END OF LINE AS BLANK
CPI '!' RNZ! INX D! JMP DEBLANK

ADDH: ;ADD A TO H.L
ADD L! MOV L.A! RNC
INR H! RET

FILLFCB:
;EQUIVALENT TO FILLFCB(8)
MVI A,0

01DA 3E00

FILLFCB:
LXI H,COMFCB! CALL ADDH! PUSH H! PUSH H ;FCB RESCANED AT END
XRA A! STA SDISK ;CLEAR SELECTED DISK (IN. CASE A....)
LHLD COMADDR! XCHG ;COMMAND ADDRESS IN. D.E
CALL DEBLANK ;TO FIRST NON-BLANK CHARACTER
XCHG! SHLD STADDR ;IN. CASE OF ERRORS
XCHG! POP H ;D.E HAS COMMAND, H.L HAS FCB ADDRESS
;LOOK FOR PRECEDING FILE NAME A. B. ...
LDAX D! ORA A! JZ SETCUR0 ;USE CURRENT DISK IF EMPTY COMMAND
SBI 'A'-1! MOV B,A ;DISK NAME HELD IN. B IF ;FOLLOWS
INX D! LDAX D! CPI '!' JZ SETDSK ;SET DISK NAME IF ;
;
SETCUR: ;SET CURRENT DISK
DCX D ;BACK TO FIRST CHARACTER OF COMMAND
SETCUR0:
LDA CDISK! MOV M,A! JMP SETNAME
;
SETDSK: ;SET DISK TO NAME IN. REGISTER B
MOV A,B! STA SDISK ;MARK AS DISK SELECTED
MOV M,B! INX D ;PAST THE ;
;
SETNAME: ;SET THE FILE NAME FIELD
MVI B,8 ;FILE NAME LENGTH (MAX)
SETNAM0: CALL DELIM! JZ PADNAME ;NOT A DELIMITER
INX H! CPI '*'! JNZ SETNAM1 ;MUST BE ?'S
MVI M, '?'! JMP SETNAM2 ;TO DEC COUNT
;
SETNAM1: MOV M,A ;STORE CHARACTER TO FCB! INX D
SETNAM2: DCR B ;COUNT DOWN. LENGTH! JNZ SETNAM0
;
;END OF NAME, TRUNCATE REMAINDER
TRNAME: CALL DELIM! JZ SETTY ;SET TYPE FIELD IF DELIMITER
INX D! JMP TRNAME
;
PADNAME: INX H! MVI M, ' '! DCR B! JNZ PADNAME
;
SETTY: ;SET THE TYPE FIELD
MVI B,3! CPI '!' JNZ PADTY ;SKIP THE TYPE FIELD IF NO .
INX D ;PAST THE . TO THE FILE TYPE FIELD
SETTY0: ;SET THE FIELD FROM THE COMMAND BUFFER
CALL DELIM! JZ PADTY! INX H! CPI '*'! JNZ SETTY1
MVI M, '?'! ;SINCE * SPECIFIED! JMP SETTY2 5

```

```

0255 7713
0257 05C24402

```

```

025B CDAC01CA6C
0265 23362005C2

```

```

026C 0603
026E 23360005C2
0275 EB22C377

```

```

0279 E1010B00
027D 237FE3FC2
0284 04
0285 0DC27D02

```

```

0289 78B7C9

```

INTVEC:

```

028C 44495220
0290 45524120
0294 54595045
0298 53415645
029C 52454E20
0005 =
02A0 0000000000

```

INTRINSIC:

```

02A6 218C020E00
02AB 79FE05D0
02AF 112107
02B2 0604
02B4 1ABE
02B6 C2C702
02B9 132305
02BC C2B402

```

```

02BF 1AFE20C2CC
02C5 79C9

```

```

02C7 2305C2C702

```

```

02CC 0C
02CD C3AB02

```

```

SETTY1: ;NOT A *, SO COPY TO TYPE FIELD
MOV M,A! INX D
SETTY2: ;DECREMENT COUNT AND GO AGAIN
DCR B! JNZ SETTY0
;
;END OF TYPE FIELD, TRUNCATE
TRTYP: ;TRUNCATE TYPE FIELD
CALL DELIM! JZ EFILL! INX D! JMP TRTYP
;
PADTY: ;PAD THE TYPE FIELD WITH BLANKS
INX H! MVI M, ' '! DCR B! JNZ PADTY
;
EFILL: ;END OF THE FILENAME/FILETYPE FILL, SAVE COMMAND ADDRESS:
;FILL THE REMAINING FIELDS FOR THE FCB
MVI B,3
EFILL0: INX H! MVI M,0! DCR B! JNZ EFILL0
XCHG! SHLD COMADDR ;SET NEW STARTING POINT
;
;RECOVER THE START ADDRESS OF THE FCB AND COUNT ?'S
POP H! LXI B,11 ;B=0, C=8+3
SCHQ: INX H! MOV A,M! CPI '?'! JNZ SCHQ0
;? FOUND, COUNT IT IN. B! INR B
SCHQ0: DCR C! JNZ SCHQ
;
;NUMBER OF ?'S INH C, MOVE TO A AND RETURN. WITH FLAGS
MOV A,B! ORA A! RET

```

```

;INTRINSIC FUNCTION. NAMES (ALL ARE FOUR CHARACTERS)
DB 'DIR '
DB 'ERA '
DB 'TYPE'
DB 'SAVE'
DB 'REN '
INTLEN EQU ($-INTVEC)/4 ;INTRINSIC FUNCTION. LENGTH
SERIAL: DB 0,0,0,0,0,0

```

```

;LOOK FOR INTRINSIC FUNCTIONS (COMFCB HAS BEEN FILLED)
LXI H,INTVEC! MVI C,0 ;C COUNTS INTRINSICS AS SCANNED
INTRIN0: MOV A,C! CPI INTLEN. ;DONE WITH SCAN?!! RNC
;NO, MORE TO SCAN
LXI D,COMFCB+1 ;BEGINNING OF NAME
MVI B,4 ;LENGTH OF MATCH IS IN. B
INTRIN1: LDAX D! CMP M ;MATCH?
JNZ INTRIN2 ;SKIP IF NO MATCH
INX D! INX H! DCR B
JNZ INTRIN1 ;LOOP WHILE MATCHING
;
;COMPLETE MATCH ON. NAME, CHECK FOR BLANK IN. FCB
LDAX D! CPI ' '! JNZ INTRIN3 ;OTHERWISE MATCHED
MOV A,C! RET ;WITH INTRINSIC NUMBER IN. A
;
INTRIN2: ;MISMATCH, MOVE TO END OF INTRINSIC
INX H! DCR B! JNZ INTRIN2
;
INTRIN3: ;TRY NEXT INTRINSIC
INR C ;TO NEXT INTRINSIC NUMBER
JMP INTRINA ;FOR ANOTHER ROUND

```

CHECKSUB:

```

;CHECK FOR A SUBMIT FILE (REGISTER A HAS DISK NAME)
ORA A! JNZ SETSUB0 ;CLEAR SUBMIT IF NOT DISK A
;OPENH SUBMIT FILE - IF POSSIBLE
LXI D,SUBFCB! CALL OPENH JZ SETSUB0
;SUBMIT FILE OPENED, SET SUBMIT TO TRUE
MVI A,0FFH! JMP SETSUB

```

SETSUB0:

```

XRA A ;CLEAR SUBMIT VARIABLE
SETSUB. STA SUBMIT
RET

```

CCPSTART:

```

;ENTER HERE FROM BOOT LOADER
LXI SP,STACK! PUSH B! PUSH 0 ;SAVE INITIAL DISK NUMBER
CALL INITIALIZE! POP B ;RECOVERED DISK NUMBER TO C
PUSH PSU ;SAVE * FLAG FROM INITIALIZATION. (00 IF NO * FILE)
MOV A,C ;DISK NUMBER IN. ACCUMULATOR
CALL SELECT ;PROPER DISK IS SELECTED, NOW CHECK SUB FILES
POP PSU ;RECOVER * FILE FLAG FROM INITIALIZATION
POP B ;ORIGINAL DISK NUMBER IS IN. C
INR A ;00 BECOMES 01, FF BECOMES 00! ORA C ;OR DISK NUMBER
;ACCUMULATOR CONTAINS 00 IF ONN DISK 0, AND * FILE PRESENT
;OTHERWISE, THE ACCUMULATOR CONTAINS A NON-ZERO VALUE
CALL CHECKSUB

```

CCP:

```

;ENTER HERE ON. EACH COMMAND OR ERROR CONDITION
LXI SP,STACK
CALL CRLF ;PRINT D> PROMPT, WHERE D IS DISK NAME
CALL CSELECT! STA CDISK!
ADI 'A!' CALL PRINTCHAR
MVI A,'>'! CALL PRINTCHAR
LXI D,BUFF! CALL SETDMA ;DEFAULT DMA ADDRESS IS AT BUFF
CALL READCOM! CALL FILLFCB0
CNZ COMERR ;THE NAME CANNOT BE ANH AMBIGUOUS REFERENCE
LDA SDISK! ORA A! JNZ USERFUNC

```

;CHECK FOR AN. INTRINSIC FUNCTION

CALL INTRINSIC

```

LXI H,JMPTAB ;INDEX IS IN. THE ACCUMULATOR
MOV E,A! MVI D,0! DAD D! DAD D ;INDEX INN D,E
MOV A,M! INX H! MOV H,M! MOV L,A! PCHL
;PC CHANGES TO THE PROPER INTRINSIC OR USER FUNCTION
JMPTAB:

```

DW	DIRECT	;DIRECTORY SEARCH
DW	ERASE	;FILE ERASE
DW	TYPE	;TYPE FILE
DW	SAVE	;SAVE MEMORY IMAGE
DW	RENAME	;FILE RENAME
DW	USERFUNC	;USER-DEFINED FUNCTION

BADSERIAL:

```

LXI H,D1 OR (HLT SHL 0)
SHLD CCPLOC1 LXI H,CCPLOC1 PCHL

```

UTILITY SUBROUTINES FOR INTRINSIC HANDLERS

```

READERR: ;PRINT THE READ ERROR MESSAGE

```

```

034E 015403C33D
0354 5245414420

```

```

035F 016503C33D
0365 4E4F542046

```

```

036F 2121070100

```

```

0375 7EFE20CAA0

```

```

037B 23D630FE0A
0383 5778
0385 E6E0C285J1
038A 78
038B 070707
038E 80DA8501
0392 80DA8501
0396 82DA8501
039A 470DC27503
039F C9

```

```

03A0 7EFE20C285
03A6 230DC2A003
03AB 78C9

```

```

03AD 0603
03AF 7E122313
03B3 05C2AF03
03B7 C9

```

```

03B0 218000B1CD

```

```

03C1 AF322007
03C5 3AC907B7C8
03CA 3D21C807BE
03D0 C34C00

```

```

03D3 3AC907B7C8
03D8 3D21C807BE
03DE 3AC807C34C

```

```

03E4 CDDA01
03E7 CDC103
03EA 2121077E
03EE FE20C2FC03

```

```

LXI B,RDMSG! JMP PRINT
RDMSG: DB 'READ ERROR*'

```

NOFILE:

```

;PRINT NO FILE MESSAGE
LXI B,NOFMSG! JMP PRINT
NOFMSG: DB 'NOT FOUND*'

```

CONVERT:

```

;CONVERT THE BYTE VALUE IN. COMFCB TO BINARY
LXI H,COMFCB+1! LXI B,11 ;(B=0, C=11)
;VALUE ACCUMULATED INN B, C COUNTS NAME LENGTH TO ZERO
CONV0: MOV A,M! CPI '!' JZ CONV1
;MORE TO SCAN, CONVERT CHAR TO BINARY AND ADD
INX H! SUI '0!' CPI 10! JNC COMERR ;VALID?
MOVV D,A ;SAVE VALUE! MOV A,B ;MULT BY 10
ANI 1110#0000! JNZ COMERR
MOV A,B ;RECOVER VALUE
RLC! RLC! RLC ;*8
ADD B! JC COMERR
ADD B! JC COMERR ;*8+2 = *10
ADD D! JC COMERR ;+DIGIT
MOV B,A! DCR C! JNZ CONV0 ;FOR ANOTHER DIGIT
RET
CONV1: ;END OF DIGITS, CHECK FOR ALL BLANKS
MOV A,M! CPI '!' JNZ COMERR ;BLANKS?
INX H! DCR C! JNZ CONV1
MOV A,B ;RECOVER VALUE! RET

```

MOVENAME:

```

;MOVE 3 CHARACTERS FROM H,L TO D,E ADDRESSES
MVI B,3
MOVE0: MOV A,M! STAX D! INX H! INX D
DCR B! JNZ MOVE0
RET

```

ADDHCF:

```

;BUFF + A + C TO H,L FOLLOWED BY FETCH
LXI H,BUFF! ADD C! CALL ADDH! MOV A,M! RET

```

SETDISK:

```

;CHANGE DISKS FOR THIS COMMAND, IF REQUESTED
XRA A! STA COMFCB ;CLEAR DISK NAME FROM FCB
LDA SDISK! ORA A! RZ ;NO ACTION, IF NOT SPECIFIED
DCR A! LXI H,CDISK! CMP M! RZ ;ALREADY SELECTED
JMP SELECT

```

RESETDISK:

```

;RETURN. TO ORIGINAL DISK AFTER COMMAND
LDA SDISK! ORA A! RZ ;NO ACTION, IF NOT SELECTED
DCR A! LXI H,CDISK! CMP M! RZ ;SAME DISK
LDA CDISK! JMP SELECT

```

INDIVIDUAL INTRINSICS FOLLOW

DIRECT:

```

;DIRECTORY SEARCH
CALL FILLFCB0 ;COMFCB GETS FILE NAME
CALL SETDISK ;CHANGE DISK DRIVES IF REQUESTED
LXI H,COMFCB+1! MOV A,M ;MAY BE EMPTY REQUEST
CPI '!' JNZ DIR1 ;SKIP FILL OF ??? IF NOT BLANK
;SET COMFCB TO ALL ??? FOR CURRENT DISK

```

03F3 0608
03F5 363F2385C2

03FC CD9400
03FF CC5F03
0402 CA5C04
0405 CD3300

0408 CD4601C641
0410 3E3ACD2700
0415 3E20CD2700

041A 3AC707
041D 0F0F0FE660

0423 0601
0425 78CDB083

0429 FE20C23904
042E 3E09CD0803
0433 3E20BECA50

0439 CD2D00
043C 0478FE0CD2

0443 FE09C22504

0448 3E20CD2D00

0450 CD3301
0453 C25C04
0456 CD7A00C302

045C C3D906

ERASE.

045F CDDA01
0462 FE0B
0464 C28784

0467 019304CD3D
046D CDBA00
0470 21420735C2
0477 237EFE59C2

047E 2322C307
0482 212007363F

0487 CDC103
048A 112007CD8A
0490 C3D906

0493 414C4C2046

TYPE.

04A4 CDDA01C285
04AA CDC103CD5C
04B0 CAEE04

MVI B,11 ;LENGTH OF FILE ??????????
DIP0: MVI M,'?' ; INX H! DCR B! JNZ DIR0
;NOT A BLANK REQUEST, MUST BE IN. COMFCB
DIR1. CALL SEARCHCOM ;FIRST ONE HAS BEEN FOUND
CZ NOFILE ;NOT FOUND MESSAGE
DIR2. JZ ENDIR
CALL CRLF
;ANOTHER FCB FOUND, PRINT PROMPT
CALL CSLECT! ADI 'A'! CALL PRINTCHAR
MVI A,'!'! CALL PRINTCHAR
MVI A,'!'! CALL PRINTCHAR
;COMPUTE POSITION. OF NAME IN. BUFFER
LDA DCNT ;GET THE LOCATION. OF THE ELEMENT
RRC! RRC! RRC! ANI 11080000B! MOV C,A
;C CONTAINS BASE INDEX INTO BUFF FOR DIR ENTRY
MVI B,1 ;START WITH FIRST CHARACTER OF NAME
DIR3. MOV A,B! CALL ADDHCF ;BUFF+A+C FETCHED
;DELETE TRAILING BLANKS
CPI ' '! JNZ DIR4 ;CHECK FOR BLANK TYPE
MVI A,9! CALL ADDHCF ;FIRST CHAR OF TYPE
MVI A,'!'! CMP M! JZ DIR5 ;TERMINATE

DIR4.

CALL PRINTBC ;CHAR PRINTED
INR B! MOV A,B! CPI 12! JNC DIR5
;CHECK FOR BREAK BETWEEN NAMES
CPI 9! JNZ DIR3 ;FOR ANOTHER CHAR
;PRINT A BLANK BETWEEN. NAMES
MVI A,'!'! CALL PRINTBC! JMP DIR3

DIR5.

;END OF CURRENT ENTRY
CALL BREAKSKEY ;CHECK FOR INTERRUPT AT KEYBOARD
JNZ ENDIR ;ABORT DIRECTORY SEARCH
CALL SEARCHM! JMP DIR2 ;FOR ANOTHER ENTRY
ENDIR. ;END OF DIRECTORY SCAN
JMP RETCOM

ERASEFILE.

CALL SETDISK
LXI D,COMFCB! CALL DELETE
JMP RETCOM

ERMSG. DB

'ALL FILES (Y/N)?'

CALL FILLFCB! JNZ COMERR ;DON'T ALLOW ?'S IN. FILE NAME
CALL SETDISK! CALL OPENC ;OPEN. THE FILE
JZ TYPERR ;ZERO FLAG INDICATES NOT FOUND

04B3 CD330021CA

04B8 21CA077EFE
04C1 DACE04E5

04C5 CD9600E1
04C9 C2E704
04CC AF77

04CE 34
04CF 218000CDD5
04D5 7EFE1ACAD9
04DB CD2700
04DE CD3301C2D9
04E4 C3B004

04E7 3DCAD906
04EB CD4E03
04EE CDD303C385

04F4 CDDA01
04F7 3AC907B7C2
04FE CD6F03
0501 F5

0502 CDDA01
0505 C28501
0508 CDC103
050B 112007D5CD
0512 D1CDA300
0516 CD5C00
0519 CA4105
051C F1
051D 6F260029
0521 110001

0524 7CB5CA4705
0529 2B
052A E5
052B 21800019E5
0530 CD5401
0533 112007CD9C
0539 D1E1
053B C24105
053E C32405

0541 015905CD3D

0547 112007CD66
054D C25605
0550 016205CD3D
0556 C3D906
0559 4E4F205350
0562 43414E4E4F

;FILE OPENED, READ 'TIL EOF
CALL CRLF! LXI H,BPTR! MVI M,255 ;READ FIRST BUFFER
TYPE0. ;LOOP ON. BPTR
LXI H,BPTR! MOV A,M! CPI 128 ;END BUFFER
JC TYPE1! PUSH H ;CARRY IF B.1,...,127
;READ ANOTHER BUFFER FULL
CALL DISKREAD! POP H ;RECOVER ADDRESS OF BPTR
JZ TYPE0F ;HARD END OF FILE
XRA A! MOV M,A ;BPTR = 0
TYPE1. ;READ CHARACTER AT BPTR AND PRINT
INR M ;BPTR = BPTR + 1
LXI H,BUFF! CALL ADDH ;H/L ADDRESSES CHAR
MOV A,M! CPI EOF! JZ RETCOM
CALL PRINTCHAR
CALL BREAKSKEY! JNZ RETCOM ;ABORT IF BREAK
JMP TYPE0 ;FOR ANOTHER CHARACTER

TYPE0F. ;END OF FILE, CHECK FOR ERRORS
DCR A! JZ RETCOM
CALL READERR
TYPERR. CALL RESETDISK! JMP COMERR

CALL FILLFCB ;SHOULD BE NUMBER
LDA SDISK! ORA A! JNZ COMERR ;CANNOT BE PREFIXED
CALL CONVERT ;RESULTING VALUE IN. REGISTER A
PUSH PSW ;SAVE IT FOR LATER

;SHOULD BE FOLLOWED BY A FILE TO SAVE THE MEMORY IMAGE
CALL FILLFCB
JNZ COMERR ;CANNOT BE AMBIGUOUS
CALL SETDISK ;MAY BE A DISK CHANGE
LXI D,COMFCB! PUSH D! CALL DELETE ;EXISTING FILE REMOVE
POP D! CALL MAKE ;CREATE A NEW FILE ON. DISK

CALL OPENC ;OPEN THE FILE
JZ SAVERR ;NO DIRECTORY SPACE

POP PSW ;PAGES TO WRITE IS IN. A. CHANGE TO #SECTORS
MOV L,A! MVI H,B! DAD H!
LXI D,TRAN. ;H/L IS SECTOR COUNT, D.E IS LOAD ADDRESS

SAVE0. ;CHECK FOR SECTOR COUNT ZERO
MOV A,H! ORA L! JZ SAVE1 ;MAY BE COMPLETED
DCX H ;SECTOR COUNT = SECTOR COUNT - 1
PUSH H ;SAVE IT FOR NEXT TIME AROUND
LXI H,128! DAD D! PUSH H ;NEXT DMA ADDRESS SAVED
CALL SETDMA ;CURRENT DMA ADDRESS SET
LXI D,COMFCB! CALL DISKWRITE
POP D! POP H ;DMA ADDRESS, SECTOR COUNT
JNZ SAVERR ;MAY BE DISK FULL CASE
JMP SAVE0 ;FOR ANOTHER SECTOR

SAVERR. ;MUST BE FULL DISK
LXI B,FULLMSG! CALL PRINT
SAVE1. ;END OF DUMP, CLOSE THE FILE
LXI D,COMFCB! CALL CLOSE
JNZ SAVE2 ;OTHERWISE CANNOT CLOSE FILE
LXI B,CLOSMSC! CALL PRINT
JMP RETCOM
FULLMSG. DB 'NO SPACES'
CLOSMSC. DB 'CANNOT CLOSE'

RENAME.

```

055F CDDA01C285 ;RENAME A FILE ON A SPECIFIC DISK
0575 3AC907F5 CALL FILLFCB01 JNZ COMERR ;MUST BE UNAMBIGUOUS
0579 C0C103 LDA SDISK! PUSH PSW ;SAVE FOR LATER COMPARE
057C C02400 CALL SETDISK ;DISK SELECTED
057F C2D005 CALL SEARCHCOM ;IS NEW NAME ALREADY THERE?
JNZ RENERR3

;FILE DOESN'T EXIST, MOVE TO SECOND HALF OF FCB
LXI H,COMFCB! LXI D,COMFCB+16! MVI B,16! CALL MOVE0
;CHECK FOR = OR LEFT ARROW
LHLD COMADDR! XCHG! CALL DEBLANK
CPI '='! JZ REN1 ;OK IF =
CPI LA! JNZ RENERR2
REN1: XCHG! INX H! SHLD COMADDR ;PAST DELIMITER
;PROPER DELIMITER FOUND
CALL FILLFCB01 JNZ RENERR2
;CHECK FOR DRIVE CONFLICT
POP PSW! MOV B,A ;PREVIOUS DRIVE NUMBER
LXI H,SDISK! MOV A,M! ORA A! JZ REN2
;DRIVE NAME WAS SPECIFIED, SAME ONE?
CMP B! MOV M,B! JNZ RENERR2
REN2: MOV M,B ;STORE THE NAME IN CASE DRIVES SWITCHED
XRA A! STA COMFCB! CALL SEARCHCOM ;IS OLD FILE THERE?
JZ RENERR1
;EVERYTHING IS OK, RENAME THE FILE
LXI D,COMFCB! CALL RENAM
JMP RETCOM

RENERR1: ;NO FILE ON DISK
CALL HOFIL! JMP RETCOM
RENERR2: ;AMBIGUOUS REFERENCE/NAME CONFLICT
CALL RESETDISK! JMP COMERR
RENERR3: ;FILE ALREADY EXISTS
LXI B,RENMSG! CALL PRINT! JMP RETCOM
RENMSG: DB 'FILE EXISTS!'

USERFUNC:
05ED C07101 CALL SERIALIZE ;CHECK SERIALIZATION
;LOAD USER FUNCTION, AND SET UP FOR EXECUTION
LDA COMFCB+1! CPI '='! JNZ USER0
;NO FILE NAME, BUT MAY BE DISK SWITCH
LDA SDISK! ORA A! JZ ENDCOM ;NO DISK NAME IF 0
DCR A! STA CDISK! STA DISKA
CALL SELECT! JMP ENDCOM
;FILE NAME IS PRESENT
LXI D,COMFCB+9! LDAX D! CPI '='! JNZ COMERR ;TYPE ' '
PUSH D! CALL SETDISK! POP D! LXI H,COMTYPE ;.COM
CALL MOVENAME ;FILE TYPE IS SET TO .COM
CALL OPENC! JZ USERER
;FILE OPENED PROPERLY, READ IT INTO MEMORY
LXI H,TRAN. ;TRANSIENT PROGRAM BASE
LOAD0: PUSH H ;SAVE DMA ADDRESS
XCHG! CALL SETDMA
LXI D,COMFCB! CALL DISKREAD! JNZ LOAD1
;SECTOR LOADED, SET NEW DMA ADDRESS AND COMPARE
POP H! LXI D,128! DAD D
LXI D,TRANH ;HAS THE LOAD OVERFLOWED?
MOV A,L! SUB EI MOV A,M! SBB D! JNC LOADERR
JMP LOAD0 ;FOR ANOTHER SECTOR

```

```

0649 E13DC2C206
064E CDD303
0651 CDDA0121C9
0658 7E322007
065C 3E10CDDC01
0661 E17E323007
0666 AF324007
066A 115C002120

```

```

0675 214307
0678 7EB7CA8606
0682 23C37806

```

```

0686 0600118100
068B 7E1207CA97

```

```

0691 042313C388

```

```

0697 78320008
069B CD3300

```

```

069E 3AC0073204
06A4 CD41011180
06AD CD0001
06B0 31FE06
06B3 3AC007CD4C
06B9 C3FC02

```

```

06BC CDD303C385

```

```

06C2 01C006CD3D
06C8 C3D306
06CB 4C4F414420
06D6 434F4D

```

```

06D9 CDD303

```

```

06DC CDDA01
06DF 3A2107D620

```

```

06E8 C28501
06EB C3FC02

```

```

06EE

```

```

06FE 00
06FF 0024242420SUBFCB,
0708 5355420000
070E

```

```

LOAD1: POP H! DCR A! JNZ LOADERR ;END FILE IS 1
CALL RESETDISK ;BACK TO ORIGINAL DISK
CALL FILLFCB01 LXI H,SDISK! PUSH H
MOV A,M! STA COMFCB ;DRIVE NUMBER SET
MVI A,16! CALL FILLFCB ;MOVE ENTIRE FCB TO MEMO
POP H! MOV A,M! STA COMFCB+16
XRA A! STA COMREC ;RECORD NUMBER SET TO ZERO
LXI D,FCB! LXI H,COMFCB! MVI B,33! CALL MOVE0
;MOVE COMMAND LINE TO BUFF
LXI H,COMBUF

```

```

BMOVE0: MOV A,M! ORA A! JZ BMOVE1! CPI ' '! JZ BMOVE1
INX H! JMP BMOVE0 ;FOR ANOTHER SCAN
;FIRST BLANK POSITION, FOUND

```

```

BMOVE1: MVI B,0! LXI D,BUFF+1! ;READY FOR THE MOVE
BMOVE2: MOV A,M! STAX D! ORA A! JZ BMOVE3
;MORE TO MOVE
INR B! INX H! INX D! JMP BMOVE2

```

```

BMOVE3: ;B HAS CHARACTER COUNT
MOV A,B! STA BUFF
CALL CRLF
;NOW GO TO THE LOADED PROGRAM
LDA CDISK! STA DISKA ;CURRENT DISK AT DISKA
CALL LIFTHD! LXI D,BUFF! CALL SETDMA
CALL TRAN. ;GONE TO THE LOADED PROGRAM
LXI SP,STACK ;MAY COME BACK HERE
LDA CDISK! CALL SELECT
JMP CCP

```

```

USERER: ;ARRIVE HERE ON COMMAND ERROR
CALL RESETDISK! JMP COMERR

```

```

LOADERR: ;CANNOT LOAD THE PROGRAM
LXI B,LOADMSG! CALL PRINT
JMP RETCOM
LOADMSG: DB 'LOAD ERRORS!'
COMTYPE: DB 'COM' ;FOR COM FILES

```

```

RETCOM: ;RESET DISK BEFORE END OF COMMAND CHECK
CALL RESETDISK

```

```

ENDCOM: ;END OF INTRINSIC COMMAND
CALL FILLFCB01 ;TO CHECK FOR GARBAGE AT END OF LINE
LDA COMFCB+1! SUI ' '! LXI H,SDISK! ORA H
;TO IN. ACCUMULATOR IF NO DISK SELECTED, AND BLANK FCB
JNZ COMERR
JMP CCP

```

```

DATA AREAS
DS 16 ;8 LEVEL STACK

```

```

STACK,
;SUBMIT' FILE CONTROL BLOCK
SUBMIT: DB 0 ;00 IF NO SUBMIT FILE, FF IF SUBMITTING
DB 0,'$$$' ;FILE NAME IS $$$
DB 0 ;FILE TYPE IS SUB
SUBRC: DS 1 ;RECORD COUNT FILED

```

```

070F      DS      16      ;DISK MAP
071F      SUBCR:  DS      1      ;CURRENT RECORD TO READ
;
;      COMMAND FILE CONTROL BLOCK
0720      COMFCB: DS      32      ;FIELDS FILLED IN. LATER
0740      COMREC: DS      1      ;CURRENT RECORD TO READ/WRITE
;
;      COMMAND BUFFER
0741 7F    MAXLEN: DB      17      ;MAX BUFFER LENGTH
0742 00    COMLEN: DB      0      ;COMMAND LENGTH (FILLED IN. BY DDS)
0743      COMBUF: DS     128      ;COMMAND BUFFER
07C3      COMADDR: DS      2      ;ADDRESS OF NEXT TO SCAN
07C5      STADDR: DS      2      ;STARTING ADDRESS OF CURRENT FILLFCB REQUEST
;
07C7      DCNT:   DS      1      ;DISK DIRECTORY COUNT (USED FOR ERROR CODES)
07C8      CDISK:  DS      1      ;CURRENT DISK
07C9      SDISK:  DS      1      ;SELECTED DISK FOR CURRENT OPERATION
;NONE=0, A=1, B=2 ...
07CA      BPTR:   DS      1      ;BUFFER POINTER
07CB      END      CCPL0C

```