

```

;      TERMINAL.ASM   AUGUST 27, 1981
;
;      THIS PROGRAM WAS ORIGINALLY DEVELOPED BY THE SMALL SYSTEMS
;      GROUP OF THE INSTITUTE OF COMPUTER SCIENCE, UNIVERSITY OF GUELPH,
;      COPYRIGHT 1979 BY DON GENNER.
;      ADAPTED TO RUN IN A MACHINE INDEPENDENT CP/M ENVIRONMENT
;      BY JOHN WILSON.
;      MODIFIED BY J. LAW, DEPT. OF PHYSICS, UNIVERSITY OF GUELPH,
;      WITH DON GENNER'S HELP.
;      EXTENSIVELY MODIFIED TO BE MORE NEARLY HOST INDEPENDENT,
;      TO RECEIVE INTO A BUFFER, TO SEND BY LOGICAL DISK SECTOR,
;      EXPANDED LOCAL MODE, TO WORK WITH STANDARD BDOS, AND MANY OTHERS
;      BY GEORGE CORLISS, DEPT. OF MATH., MARQUETTE UNIVERSITY.
;      MODIFIED FOR SLOW PRINTER BY J. GLENN BROOKSHEAR, MARQUETTE
;      UNIVERSITY.

```

```

;      DOCUMENTATION AND INSTRUCTIONS ARE IN FILE TERMINAL.DOC
;
;-----
;

```

```

BASE      EQU      4200H
USER      EQU      BASE+100H
FCB       EQU      BASE+005CH
STACK    EQU      USER+0D00H
BUFF     EQU      USER+0E00H
PRTBUF   EQU      USER+0F00H
CPM      EQU      BASE  B
BDOS     EQU      BASE+0005H
;
CTLA     EQU      01H
CTLB     EQU      02H
CTLC     EQU      03H
CTLE     EQU      05H
BEL      EQU      07H      ;CTRL G
BS       EQU      08H      ;BACKSPACE = CTRL H
TAB      EQU      09H      ;CTRL I
LF       EQU      0AH      ;LINE FEED = CTRL J
FF       EQU      0CH      ;FORM FEED = CTRL L
CR       EQU      0DH      ;CTRL M
CTLP     EQU      10H
XON      EQU      11H      ;CTRL Q
XOFF     EQU      13H      ; CTRL S
CTLY     EQU      19H      ;CTRL Y
EOD      EQU      1AH      ;CP/M END OF FILE = CTRL Z
ESC      EQU      1BH      ;ESCAPE
DEL      EQU      7FH

```

```

OVFLBF EQU USER+0F00H

```

```

;----- BEGIN
;
;      ORG      USER
;      JMP      ENTER
HOSTON   DB      01H
HEAD     DW      PRTBUF
TAIL     DW      PRTBUF
BAKLOG   DB      00
STOPLMT EQU      0C0H
STRTLMT EQU      20H
;

```

remove FF from stack

SELECTION

<i>A</i>	<i>1</i>
<i>B</i>	<i>128</i>
<i>C (DEFAULT)</i>	<i>256</i>

FOR MACHINES WITH

- 32K RAM*
- 48K RAM*
- 64K RAM*

; BEGIN MAIN PROGRAM

TERMINAL:

IXI D,TMESS
CALL STRINGF
JMP TERM2

TMESS DB CR,LF,'TERMINAL MODE',CR,LF,LF,' \$'

TERM2 CALL CONIO ;ANYTHING FROM TERMINAL?
JZ SGET ;NO - MOVE ON
MOV E,A ;YES - LOAD E

; PROCESS SPECIAL CONTROL CHARACTERS

; P TOGGLE PRINTER ON/OFF
; A ENTER LOCAL MODE
; B SET BELL TO RING ON NEXT HOST CHAR
; C RETURN TO CP/M
; E TOGGLE CRT ECHO ON/OFF
; ESC FORCE SEND NEXT CHAR TO HOST

CPI CTLP
JZ SWCHPT ; IF P
CPI CTLA
JZ LOCAL ; IF A
CPI CTLB
JZ SWBEL ; IS B
CPI CTLC
JZ EXIT ; IF C
CPI CTLE
JZ SWECHO ; IF E
CPI ESC
JZ FORSND ; IF ESCAPE
CPI BADBS
JZ TRADE ; IF WRONG BS/DEL FOR HOST

push
; SPUT CALL D PUNOUTF ;SEND TO HOST
pop D LDA ECHO ;SHOULD CHARACTERS BE SENT TO CRT?
CPI 0 ;0 - YES, 1 - NO
JNZ SGET

DPUTB CALL PRNT ;SEND TO CONSOLE
SGET CALL RDRINPF ;ANYTHING FROM HOST?
JZ CKPRT ;NO. CHECK PRINT BUFFER
CALL PRNT ;ECHO IT
LDA BELWAK ;IS BELL SWITCH SET?
CPI 0
JZ CKPRT ;IF NOT SET
MVI E,BEL ;RING BELL
CALL CONOUTF

XRA A ;RESET BELL SWITCH

CKPRT LDA BAKLOG ;ANYTHING IN PRINT BUFFER?

CPI 00
JZ TERM2 ;NO. TRY TERMINAL
CALL PRTSTAT ;PRINTER BUSY?

CPI 00
JZ TERM2 ;YES. TRY TERMINAL

LHLD HEAD ;SEND CHARACTER TO PRINTER
MOV E,M ;AND SHIFT HEAD POINTER.

INR L
SHLD HEAD
CALL LSTOUTF
LDA BAKLOG ;IF BACKLOG HAS DROPPED

```

DCR      A                ;TO START LIMIT THEN TELL
STA      BAKLOG           ;HOST TO START SENDING AGAIN.
CPI      STRTLMT
JNZ      TERM2
CALL     STRTHST
JMP      TERM2           ;CHECK ON TERMINAL

```

```

;
;----- END OF MAIN PROGRAM
;
;

```

```

CONIO    MVI      E,0FFH  ;DIRECT CONSOLE INPUT
         CALL     DIRCONF ;(A) = CHAR OR 00 IF NO CHAR
         OR      A        ;FORCE FLAG
         RET

```

```

;
PRNT     PUSH     H        ;PRINT CHAR (A) AT CONSOLE
         PUSH     D
         MOV      E,A
         CPI      HSTEOL  ;END OF LINE?
         JZ       PRNT2
         PUSH     D
         CALL     CONOUTF
         POP      D
         LDA      AUXON
         CPI      0
         JNZ     PRNT1    ;AND PERHAPS AT THE PRINTER
         CALL     LIST
PRNT1    POP      D
         POP      H
         RET

```

```

;
PRNT2    LDA      AUXON
         CPI      0
         JNZ     PRNT3
         LXI     D,CRLF
         CALL    STRINGF
         MVI     E,CR
         CALL    LIST
         MVI     E,LF
         CALL    LIST
         JMP     PRNT1

```

```

PRNT3    LXI     D,CRLF
         CALL    STRINGF
         JMP     PRNT1
CRLF     DB      CR,LF,'$'

```

```

;
; PLACE CONTENTS OF E REGISTER IN PRINT BUFFER.
; IF BACKLOG IS THEN COH, TELL HOST TO WAIT.
;
; CAUTION*** HOST WAIT ROUTINE USES E REGISTER.
;

```

```

LIST     LHLD    TAIL
         MOV     M,E
         INR    L
         SHLD   TAIL
         LDA    BAKLOG
         INR    A
         STA    BAKLOG
         CPI    STOPLMT
         RNZ
         CALL   STOPHST
         RET

```

```

STOPHST  LDA     Hoston  ;IS HOST ALREADY STOPPED?
         CPI     00
         RZ      ;YES. RETURN
         LXI     D,HSTOP ;NO. THEN STOP IT.

```

```

CALL MSGHOST
MVI A,00
STA HOSTON
RET
STRTHST LDA HOSTON ;IS HOST ALREADY STARTED?
CPI 00
RNZ ;YES. RETURN
LXI D,HGO ;NO. THEN START IT.
CALL MSGHOST
MVI A,01
STA HOSTON
RET

```

```

;
;
;----- SPECIAL CONTROL CODE HANDLERS
;

```

```

SWCHPT LDA AUXON
MOV B,A
MVI A,1
SUB B
STA AUXON
JMP SGET

```

```

;
AUXON DB 1

```

```

;
SWBEL MVI A,1 ;SET BELL SWITCH
STA BELWAK
JMP SGET

```

```

;
BELWAK DB 0

```

```

;
SWECHO LDA ECHO
MOV B,A
MVI A,1
SUB B
STA ECHO
JMP SGET

```

~~XRI~~ 01H

```

;
ECHO DB 1 ;1 - NO ECHO, 0 - ECHO TO CRT

```

```

;
FORSND CALL CONINPF ;FORCE SEND NEXT CHAR TO HOST
MOV E,A
JMP SPUT

```

```

;
LOCAL LXI D,LMESS ;DISPLAY 'LOCAL MODE'
CALL STRINGF
CALL CONINPF
CPI CR
JZ HELP
CPI CTLC
JZ EXIT
ANI 5FH ;CONVERT TO UPPER CASE
SUI 'A' ;SET UP FOR JUMP TABLE
JM NOTYET
CPI LAH
JP NOTYET ;IF ASCII 'Z'
LXI H,IFA
MOV E,A ;LB OF INDEX
MVI D,0 ;HB OF INDEX
DAD D ;ADD INDEX TO BASE OF JUMP TABLE
DAD D ; TWICE BECAUSE 2 BYTES/WORD
INX H ;ADDRESS HIGH BYTE FIRST
MOV D,M ;HB OF JUMP ADDRESS
DCX H
MOV E,M ;LB OF JUMP ADDRESS
XCHG ;JUMP ADDRESS TO HL

```

```

SHLD DUMMY+1 ;FILL IN JUMP ADDRESS
DUMMY JMP DUMMY ;JUMP ADDRESS IS SUPPLIED ABOVE
;
; TABLE OF JUMP ADDRESSES:
;
; A B C D E F G
IFA DW NOTYET,NOTYET,EXIT,DSKDIR,TERMINAL,NOTYET,NOTYET
; H I J K L M N O
; DW HELP,NOTYET,NOTYET,NOTYET,LOGON,NOTYET,NOTYET,OFF
; P Q R S T U V
; DW PRINTR,NOTYET,RECEIVE,SEND,TERMINAL,NOTYET,VAXHST
; W X Y Z
; DW WAIT,NOTYET,NOTYET,NOTYET
;
;
NOTYET LXI D,YETMSG
CALL STRINGF
JMP LOCAL
;
LMESS DB CR,LF,'LOCAL MODE $'
YETMSG DB BEL,CR,LF,'NOT IMPLEMENTED YET.$'
FMESS DB CR,LF,'FILE NAME? $'
;
;
TRADE MVI E,GOODBS ;SUBSTITUTE GOOD BS/DEL CHAR
JMP SPUT ;FOR A BAD ONE
;
;
;----- SUBROUTINES FOR LOCAL MODE
;
DSKDIR JMP NOTYET
;
HELP LXI D,HLPMSG
CALL STRINGF
JMP LOCAL
;
HLPMSG DB FF,LF,LF,' OPTIONS AVAILABLE:',CR,LF,LF
DB TAB,'C',TAB,'EXIT TO CP/M (E)',CR,LF
DB TAB,'D',TAB,'DISK DIRECTORY (L)',CR,LF
DB TAB,'E',TAB,'EXIT TO TERMINAL MODE (E)',CR,LF
DB TAB,'H',TAB,'HELP (L)',CR,LF
DB TAB,'L',TAB,'SEND LOGON TO HOST (E)',CR,LF
DB TAB,'O',TAB,'SEND LOGOFF TO HOST (E)',CR,LF
DB TAB,'P',TAB,'TOGGLE PRINTER ON/OFF (L)',CR,LF
DB TAB,'R',TAB,'RECEIVE FILE FROM HOST (L)',CR,LF
DB TAB,'S',TAB,'SEND FILE TO HOST (L)',CR,LF
DB TAB,'T',TAB,'TERMINAL MODE (E)',CR,LF
DB TAB,'V',TAB,'CHANGE TO VAX-11 MODE (E)',CR,LF
DB TAB,'W',TAB,'WAIT - SET SENDING DELAY (L)',CR,LF
DB LF,TAB,'(L) COMMANDS RETURN TO LOCAL MODE',CR,LF
DB TAB,'(E) COMMANDS EXIT LOCAL MODE',CR,LF,LF,'$'
;
LOGON LXI D,CRLF
CALL STRINGF
LXI D,LONMSG
CALL MSGHOST
JMP TERM2
;
OFF LXI D,CRLF
CALL STRINGF
LXI D,OFFMSG
CALL MSGHOST
JMP TERM2
;
VAXHST MVI B,55H ;MOVE VAXMSG BLOCK TO
LXI H,VAXMSG ;HOSTBF BLOCK AND THEN
LXI D,HOSTBF ;GO TO TERM2.
MOVBYT MOV A,M

```

CHGBUF

DB TAB, 'B', TAB, 'CHANGE RECEIVING BUFFER SIZE', CR, LF

CHGBUF
GETOPT

```

LXI D, CHGB
CALL STRINGF
CALL CONIO
JZ GETOPT
CPI 'A'
JZ GOT A
CPI 'B'
JZ GOT B
MOVE A, 00
STA BFSIZE
JMP TERM2
MOVE A, 01
STA BSIZE
JMP TERM2

```

SIZE BY TYPING
SENDING SELECTIONS

```

XCHG      M,A
MOV       B
DCR       B
CPI       00
JZ        VAXH1
INX       D
INX       H
XCHG
JMP       MOVBYT
VAXH1    LXI       D,VAXMOD      ;INDICATE MODE IS VAX
        CALL      STRINGF
        JMP       TERM2
VAXMOD   DB        'VAX-11 MODE.',CR,LF,LF,'$'
;
PRINTR   LDA       AUXON        ;TOGGLE PRINTER OFF/ON
        MOV      B,A
        MVI      A,1 XRI 01
        SUB      B
        STA       AUXON
        JMP       LOCAL
;
;
; RECEIVE A FILE FROM HOST
; ESTABLISH STATE
; LOOP FOR EACH CHAR
; CHECK CONSOLE
; ESC -----
; GET CHAR FROM HOST
; NONE -----
; END OF FILE -----
; WRITE CHAR TO MEMORY
; MANAGE BUFFER
; COUNT CHARS IN SECTOR
; WHEN NECESSARY, WRITE SECTOR
; HOST WAIT
; DISK WRITE
; HOST CONTINUE
; CLOSE FILE
;
RECEIVE  MVI       A,01          ;TURN PRINTER OFF
        STA       AUXON
        CALL      GETFN         ;GET FILE NAME
        LXI       D,FCB
        CALL      OPENFIF
        CPI       0FFH
        JZ        REC2         ;IF FILE IS NEW
        LXI       D,EXISTM      ;FILE EXISTS.
        CALL      STRINGF       ;WARN
        LXI       D,FCB         ;AND DELETE OLD ONE
        CALL      DELETF
REC2     LXI       D,FCB
        CALL      MAKEFIF
        CPI       0FFH
        JNZ       REC7         ;IF MAKE WAS SUCCESSFUL
        LXI       D,FULLMSG
        CALL      STRINGF
        JMP       LOCAL
;
EXISTM   DB        BEL,CR,LF,'EXISTING FILE DELETED',CR,LF,'$'
FULLMSG  DB        BEL,CR,LF,'DIRECTORY FULL. ACTION CANCELLED.$'
;
REC7     LXI       D,CRMSG
        CALL      MSGHOST
REC4     CALL      RDRINPF ;WAIT FOR ECHO OF CR SENT TO START TRANSMISSION
        CPI       CR
        JNZ       REC4

```

GOT B MVI A, 8
 STA BFSIZE
 JMP TERM2
 SELECT R
 ON OF THE FOLL

CHGL DB
 SELECT BFSIZE DB
 00

```

RECEOD CALL CONIO ;LOOP FOR EACH CHAR
        CPI EOD ;TO TERMINATE TRANSMISSION
        JZ RECEOD ;IF ESCAPE, ELSE IGNOR
        CALL RDRINPF ;WAIT FOR CHAR
        JZ REC3 ;IF NOTHING WAS RECEIVED
        CPI HSTEOD
        JZ RECEOD ;IF END OF FILE
        PUSH PSW
        CALL PRNT ;ECHO TO CONSOL
        POP PSW
        PUSH PSW
        CALL WRTMEMF
        JNZ LOCAL ;IF WRITE WAS NOT SUCCESSFUL
        POP PSW
        CPI CR
        JNZ REC3
        MVI A,LF
        CALL WRTMEMF
        JNZ LOCAL
        JMP REC3

;
;
; RECEOD LXI D,HNOSND ;END OF TRANSMISSION. CLOSE FILES
        CALL MSGHOST
        MVI A,EOD ;CP/M END OF FILE
        CALL WRTMEMF
        JMP LOCAL

;
;
; SEND A FILE FROM TERMINAL TO HOST
; ESTABLISH STATE
; OPEN FILE
; SEND START SIGNAL
; LOOP FOR EACH LOGICAL SECTOR (128 BYTES)
; READ SECTOR FROM DISK
; LOOP FOR EACH OF 128 CHARS
; GET CHAR FROM MEMORY
; LF? -----
; CR? -----
; TAB? -----
; EOD? -----
; INVALID? ----
; SEND CHAR TO HOST
; IF RESPONSE, THEN XOFF? ----
; ELSE ECHO RESPONSE
; CHECK CONSOLE
; ESC -----
; EOD OR ESC: CLOSE FILES
;
; SEND MVI A,01 ;TURN PRINTER OFF
        STA AUXON
        CALL GETFN
        LXI D,FCB
        CALL OPENFIF ;OPEN SOURCE FILE
        CPI OFFH
        JZ SENDNOT ;IF FILE NOT FOUND
        LXI D,HREC ;SET HOST TO RECEIVE A FILE
        CALL MSGHOST

;
; SENDL1 LXI D,BUFF ;LOOP FOR EACH SECTOR
        PUSH D ;BUFF REAPPEARS IN HL
        CALL SETDMAF
        LXI D,FCB
        CALL READSQF
        ORA A

;
; JZ SENDL7 ;IF NOT END OF FILE
        PUSH B ;FILL STACK TO BE CLEARED

```



```

; IN SENDOED
PUSH B
JMP SENDEOD

;
SENDL7 MVI B,129 ;INITIALIZE LOOP COUNTER
PUSH B

;
SENDL2 POP B ;LOOP FOR EACH CHAR
POP H
DCR B
XRA A
ADD B ;DONE WITH THIS SECTOR?
JZ SENDL1 ;GET NEXT SECTOR
MOV A,M ;GET CHAR FROM MEMORY
INX H
PUSH H
PUSH B
ANI 7FH ;GUARANTEE VALID ASCII CHAR
MOV E,A
CPI LF
JZ SENDLF
CPI CR
JZ SENDL3
CPI TAB
JZ SENDTAB
CPI EOD
JZ SENDEOD
CPI 20H ; 20H ARE CONTROL CODES
JM SENDINV
SENDL3 CALL SENDCHR ;RETURNS (A) = 0 IF EOD.
ORA A
JZ SENDEOD
JMP SENDL2 ;GET NEXT CHAR

;
SENDTAB JMP SENDL3 ;EXPAND TABS LATER

;
SENDINV LXI D,INVMSG ;INVALID CHAR IN FILE
CALL STRINGF ;WARN AND IGNOR
JMP SENDL2

;
INVMSG DB BEL,CR,LF,'INVALID CHARACTER IN FILE',CR,LF,'$'

;
SENDCHR LDA ECHO
CPI 0
JNZ SENDL8 ;IF NO ECHO TO CONSOLE BEFORE SEND
MOV A,E
CALL PRNT
SENDL8 CALL PUNOUTF ;SEND CHAR (E) TO HOST
LDA NDELAY ;NUMBER OF DELAY LOOPS
ORA A
JZ SENDL4 ;IF NO DELAY
CALL DELAYF
SENDL4 CALL RDRINPF ;RESPONSE?
ORA A
JZ CKPUN ;IF NOTHING RECEIVED
CPI XOFF ;XOFF MEANS WE MUST WAIT
JZ SENDOFF
CALL PRNT ;ECHO IT TO CONSOL
JMP SENDL4 ;MORE RESPONSE?
CKPUN CALL PUNSTAT ;KEEP WATCHING INPUT FROM
CPI 00 ;HOST UNTIL PUN: PORT
JZ SENDL4 ;IS CLEAR.
SENDL5 CALL CONIO ;CHECK FOR CONSOL INPUT
SUI EOD ;RETURN (A) = 0 IF Z
RET

;
SENDL6 LXI D,WAITMG ;TELL CONSOL WE ARE WAITING

```

```

CALL STRINGF
SENDOFF CALL RDRINPF ;HOST IS SLOW - WAIT FOR XON
CPI XON
JZ CKPUN
ORA A
JZ SENDOFF ;NOTHING RECEIVED, WAIT
CALL PRNT ;ELSE ECHO RECEIVED CHAR
JMP SENDOFF ;AND CONTINUE WAITING

;
; WAITMG DB CR,LF,'WAITING',CR,LF,'$'
;
; SENDLF JMP SENDL3 ;LF CAN BE HANDLED LATER
;
; SENDEOD LXI D,HCLOSE ;CLOSE HOST FILE
CALL MSGHOST
POP B
POP H ;CLEAR STACK
JMP LOCAL

;
; SENDNOT LXI D,FNFMSG ;FILE NOT FOUND TO SEND
CALL STRINGF
JMP LOCAL

;
; FNFMSG DB BEL,CR,LF,'FILE NOT FOUND',CR,LF,'$'
;
; WAIT LXI D,DELAYM ;SET COUNTER FOR NUMBER
CALL STRINGF ;DELAY LOOPS
CALL CONINPF ;USE THE ASCII CODE AS INDEX
STA NDELAY
JMP LOCAL

;
; DELAYM DB CR,LF,'ENTER DELAY COUNTER $'
; NDELAY DB 0,0
;
;
; GET A FILE NAME FROM THE OPERATOR
;
; STORE FILE NAME IN FCB
;
; GETFN LXI H,FCB
XRA A
STA FCB+12 ;ZERO CURRENT EXTENT
STA FCB+32 ;ZERO CURRENT RECORD
MVI M,0
INX H
MVI C,11
MVI A,' '
CALL FILLF ;BLANK NAME AREA
LXI D,FMESS
CALL STRINGF
LXI H,GETFNR
MVI C,14
MVI A,' '
CALL FILLF ;BLANK BUFFER AREA
LXI D,GETFNRB
CALL READCNF
LXI D,GETFNR
LXI H,FCB
INX D
LDAX D
DCX D
CPI ':'
JNZ GETNAM
LDAX D
ANI 0FH
CPI 3 ;VALID DISK NUMBER?

```

```

JP GETFN
MOV M,A
INX D
INX D
GETNAM INX H ;FILE NAME
PUSH H
MVI C,9
GETFNL1 LDAX D
CPI '.'
JZ GETFNL1F
CPI ''
RZ
CPI 'A'
JM GETFN3 ;IF NOT LOWER CASE
ANI 5FH ;CONVERT TO UPPER CASE
GETFN3 MOV M,A
INX D
INX H
DCR C
JNZ GETFNL1
GETFNL1F POP H
LXI B,8
DAD B
INX D
MVI C,3
GETFNL2 LDAX D ;EXTENSION
CPI ''
RZ
CPI 'A'
JM GETFN4 ;IF NOT LOWER CASE
ANI 5FH
GETFN4 MOV M,A
INX D
INX H
DCR C
JNZ GETFNL2
RET

;
GETFNRB DB 14,0,'1234567890123456'
GETFNR EQU GETFNRB+2
;
;
; MSGHOST: SEND MESSAGE TO THE HOST
; DE POINTS TO THE MESSAGE (MUST END IN A $)
;
MSGHOST PUSH D
MSGL1 POP D
LDAX D
INX D
CPI '$'
RZ ;END OF MESSAGE
PUSH D
MOV E,A
CALL PUNOUTF ;SEND CHAR TO HOST
CALL RDRINPF ;HAS HOST SENT ANYTHING BACK?
JZ MSGL1 ;IF NOTHING RECEIVED
CALL PRNT ;ECHO RECEIVED CHAR
JMP MSGL1

;
;
; FILLF FILL MEMORY WITH CHAR
;
; ENTRY: (A) CHAR TO BE FILLED
; (C) NUMBER OF LOCATIONS TO BE FILLED
; (HL) BEGINNING ADDRESS

```

```

MOV M,A
INX H
DCR C
JNZ FILLF
RET

```

```

;
;
; DELAYF: DELAY FOR A WHILE. (A) CONTAINS COUNTER
;

```

```

DELAYF DCR A
RZ
PUSH B
MVI B,20H
DEL1 DCR B
JZ DEL2
LHLD NDELAY
SHLD NDELAY
JMP DEL1
DEL2 POP B
JMP DELAYF

```

```

;
; WRTMEMF: HANDLE WRITING A CHAR IN REGISTER A TO MEMORY
; AND TO DISK.
; ASSUMES DISK FILE HAS ALREADY BEEN OPENED.
; IF (A) = EOD ( Z), DISK FILE IS CLOSED
;
; RETURNS (A) = 0 IF SUCCESSFUL, ≠ 0 OTHERWISE
;

```

```

WRTMEMF LHLD MEMADD
MOV M,A
INX H
SHLD MEMADD
CPI EOD
JZ MEMCLOS ;IF END OF FILE
LDA MEMCNT ;MEMORY COUNT 0 .. 128
ADI 1
CPI 80H
JZ SECFUL ;IF LOGICAL SECTOR IS FULL
STA MEMCNT
XRA A ;SUCCESS
RET
;
; SECFUL LDA BFSIZE
; LDA SECNT ;LOGICAL SECTOR FULL
; ADI 1 ;INCREMENT SECTOR COUNTER
; JZ 100H LOGICAL SECTORS ;IF 100H LOGICAL SECTORS
; STA SECNT ;RESET CHAR COUNTER
; STA MEMCNT
; RET

```

CMP B
~~*CPI BFSIZE*~~
← BUFFER FULL

```

;
; BFLMSG DB BEL,CR,LF,'SOURCE FILE IS TOO LARGE.'
; DB CR,LF,'DESTINATION FILE CLOSED.',CR,LF,'$'
;

```

```

;
; BUFFUL LXI D,HNOSND ;BUFFER IS FULL
; CALL MSCHOST STOPHST ;HOST MUST STOP SENDING
; LXI D,BFLMSG
; CALL STRINGF
;

```

```

MEMCLOS LDA SECNT ;WRITE MEMORY BUFFER TO DISK
MOV B,A ;AND CLOSE DESTINATION FILE
INR B ;(We counted sectors from 0)
LHLD BUFADD ;(B) COUNTS DOWN LOGICAL SECTORS
SHLD MEMADD ;RESET ADDRESS OF MEMORY BUFFER
MEML1 MVI A,80H ;INCREMENT MEMADD BY 80H
LHLD MEMADD ;BYTES WRITTEN PER SECTOR
MOV D,H

```

```

LDA BFFL
INC A
STA BFFL ← LXI OVFLBF
GETSTRG MVI B,0
GETL CALL RDRINPF
CPI 0
JNZ GOTIT
DCR B
JNZ GETL
JMP MEMCLOS
GOTIT MOV M,A
INX H
LDA OVCNT
INC A
STA OVCNT

```

set buffer full flag
Get string from host

```

MOV      E,L
ADD      E
MOV      L,A
JNC      MEML4
MVI      A,1
ADD      D
MOV      H,A
MEML4   SHLD  MEMADD
        PUSH B
        CALL SETDMAF
        LXI  D,FCB
        CALL WRITSQF
        POP  B
        ORA  A
        JNZ  MEML3           ;IF DISK FULL
        DCR  B
        XRA  A
        ADD  B
        JNZ  MEML1           ;IF NOT ALL SECTORS ARE WRITTEN

;
MEML2   LXI  D,FCB           ;DONE WRITING.  CLOSE FILE
        CALL CLOSEFF
        XRA  A           ;RESET ALL COUNTERS AND ADDRESSES
        STA  MEMCNT
        STA  SECNT
        LHL  BUFADD
        SHLD MEMADD
        ADI  1           ;FORCE FLAG
        RET

;
MEML3   LXI  D,LOSTMG       ;DISK FULL
        CALL STRINGF
        JMP  MEML2

;
LOSTMG  DB    BEL,CR,LF,'DISK FULL.  FILE CLOSED',CR,LF,'$'

;
MEMADD  DW    BUFF
BUFADD  DW    BUFF
MEMCNT  DB    0
SECNT   DB    0

;
;
;  FUNCTION 1:  CONSOLE INPUT
;
;  RETURN:  (A) = ASCII CHARACTER
;
CONINPF IN    350Q+5
        ANI  1
        JZ   CONINPF
        IN  350Q
        RET

;
;CONINPF MVI    C,1
;        JMP   BDOS

;
;
;  FUNCTION 2:  CONSOLE OUTPUT
;
;  ENTRY:  (E) = ASCII CHARACTER
;
CONOUTF MVI    C,2
        JMP   BDOS

;
;
;  FUNCTION 3:  READER INPUT
;

```

← BFFL DB 0
 OVCNT DB 0
 BFSIZE DB 00H ; initialized for 100H sectors

```

JMP GETSTRG

LDA  BFFL
CPI  00H
JZ   MTHL2

XRA  A
STA  BFFL
STA  SECNT
LHLD BUFADD
SHLD MEMADD
LDA  OVCNT
STA  MEMCNT
CPI  0
JZ   HOSTGO
LDA  OVCNT
MOV  B,A
LHLD MEMADD

XCHG
LHLD OVFBUF
LDAX H
STAX D
INX  H
INX  D
DCR  B
JNZ  MOVE
CALL STRHST
XRA  A
STA  OVCNT
RET

MOVE:
HOSTGO:

```

Handwritten notes:
 BFFL 00H MTHL2
 indicates buffer not full
 STA MEMCNT
 Reset address of memory buffer
 OVCNT MEMCNT
 HOSTGO OVCNT B, A MEMADD
 OVFBUF H D
 CPI EOD MEMCLOS
 JZ
 LDA MEMCNT
 INR A
 STA MEMCNT

```

;
;RDRINPF IN      330Q+5
;      ANI      1
;      RZ
;      IN      330Q
;      ANI      7FH
RDRINPF MVI      C,3
;      CALL     BDOS
;      ANI      7FH      ;STRIP PARITY BIT
;      RZ
;      CPI      DEL
;      JZ      IGNOR
;      CPI      BEL
;      JZ      IGNOR
;      CPI      LF
;      RNZ
IGNOR  XRA      A
;      RET
;
;
;
;      FUNCTION 4: PUNCH OUTPUT
;
;      ENTRY: (E) = ASCII CHARACTER
;
;PUNOUTF IN      330Q+5
;      ANI      20H
;      JZ      PUNOUTF
;      MOV      A,E
;      OUT      330Q
;      RET
PUNOUTF MVI      C,4
;      JMP      BDOS
;
;      PUNCH STATUS
;
;PUNSTAT IN      330Q+5
;      ANI      20H
;      RET
;
;
;      FUNCTION 5: LIST OUTPUT
;
;      ENTRY: (E) = ASCII CHARACTER
;
;LSTOUTF MVI      C,5
;      JMP      BDOS
;
;      CHECK PRINTER STATUS
;
;PRTSTAT IN      101Q
;      ANI      01H
;      RET
;
;
;      FUNCTION 6: DIRECT CONSOLE I/O
;
;      ENTRY: (E) = 0FFH FOR INPUT FROM CONSOLE
;              = ASCII CHAR FOR OUTPUT TO CONSOLE
;      RETURN: (A) = ASCII CHAR, IF ANY
;              = 00 IF NO CHAR WAS ENTERED
;
;      IF      BDOS VERSION DOES NOT SUPPORT FUNCTION 6
;DIRCONF MVI      C,11      ;GET CONSOLE STATUS
;      CALL     BDOS

```

```

; RZ ;NOTHING WAITING
; MVI C,1 ;CONSOLE READ (BETTER IF NO ECHO)
; CALL BDOS
; ENDIF
;
DIRCONF IN 350Q+5
ANI 1
RZ
IN 350Q
RET

;
;DIRCONF MVI C,6
; JMP BDOS
;
;
; FUNCTION 9: PRINT STRING
;
; ENTRY: (DE) = ADDRESS OF MESSAGE
; MESSAGE MUST END WITH '$'
;
STRINGF MVI C,9
JMP BDOS

;
;
; FUNCTION 10: READ CONSOLE BUFFER
;
; ENTRY: (DE) = BUFFER ADDRESS
; RETURN: ((DE)) = CONSOLE CHARACTERS
;
READCNF MVI C,10
JMP BDOS

;
;
; FUNCTION 15: OPEN FILE
;
; ENTRY: (DE) = FCB ADDRESS
; RETURN: (A) = 0 - 3 IF SUCCESS, OFFH OTHERWISE
;
OPENFIF MVI C,15
JMP BDOS

;
;
; FUNCTION 16: CLOSE FILE
;
; ENTRY: (DE) = FCB ADDRESS
; RETURN: (A) = 0 - 3 IF SUCCESS, OFFH OTHERWISE
;
CLOSEFF MVI C,16
JMP BDOS

;
;
; FUNCTION 19: DELETE FILE
;
; ENTRY: (DE) = FCB ADDRESS
; RETURN: (A) = 0 - 3 IF SUCCESS, OFFH OTHERWISE
;
DELETEF MVI C,19
JMP BDOS

;
;
; FUNCTION 20: READ SEQUENTIAL
;
; ENTRY: (DE) = FCB ADDRESS
; RETURN: (A) = 0 IF SUCCESS, ELSE NONZERO
;
READSQF MVI C,20

```

