

```

;
; 'PASCAL' FIRST READS IN AND SCANS THE DIRECTORY FOR THE 8080/
; Z80 INTERPRETER (SYSTEM.MICRO). THEN THE INTERPRETER IS
; LOADED INTO MEMORY AND STARTED AT THE SYSTEM.PASCAL BOOT VECTOR.
;
; THIS PROGRAM PROVIDED COURTESY OF NORTHWEST MICROCOMPUTER SYSTEMS, INC.
;
; ASEG
;
; MODIFIED
; 10/11/78 BY WINK SAVILLE TO READ IN ANY SIZE INTERPETER
;
; BOOT EQU 0H ;LOCATION OF CP/M BOOT VECTOR
; BDOS EQU 5H ;CP/M ENTRY VECTOR
; TPA EQU 100H ;START OF USER AREA
;
; RDCON EQU 1
; WRBUF EQU 9
;
; NBLOCKS EQU 32 ;MAXIMUM # OF BLOCKS FOR THE
; ;INTERPETER. USED TO COMPUTE START
; START EQU TPA+NBLOCKS*512 ;LOCATION OF THIS PROGRAM
; INTERP$BASE EQU TPA ;FIRST LOC USED BY THE INTERPRETER
; PBEGIN EQU INTERP$BASE+100H ;ENTRY TO THE PASCAL BOOTER
; FIRSTSP EQU INTERP$BASE+103H ;PASCAL INITIAL STACK POINTER
; DENTSZ EQU 1AH ;DIR ENTRY SIZE IN BYTES
; DTITLE EQU 06H ;OFFSET TO ENTRY TITLE
; DIRTOP EQU PBEGIN ;TOP OF TEMP RAM DISK DIRECTORY
; CONMOD EQU 0F3D5H ;added by P. Schorn
;
; CR EQU 0DH
; LF EQU 0AH
; EOM EQU '$'
;
;
; ORG TPA
;
;
; JMP START
;
;
; ORG START
;
; LXI SP,100H
; MAIN: CALL INIT$IO ;LET'S BOOT SYSTEM.MICRO
; ;RESET I/O SYSTEM
;
; removed by P. Schorn as UCSD disk is already attached
; CALL REQUEST$DISK ;GET PASCAL DISK ON DRIVE A
; MVI C,0 ; THEN SELECT THE DRIVE
; CALL SELDSK
;
; LXI B,DIRTOP ;READ THE DIRECTORY INTO DIRTOP
; CALL READ$DIR
;
; LXI H,DIRTOP ;SET THE DIRECTORY ENTRY POINTER
; LXI D,DENTSZ ; TO FIRST ENTRY AFTER THE VOLUME NAME
; DAD D
; SHLD DENTP
;
; CALL FIND$INTERP ;THEN FIND THE INTERPRETER
; CALL SAY$LOADING ;GOT IT SAY WHAT WE'RE UP TO
; CALL READ$INTERP ; AND READ IT IN
;
; LHLD BOOT+1 ;NOTE: LOC 2 MUST HAVE CURRENT BIOS PAGE
; MVI L,0 ; FOR PROPER SYSTEM OPERATION
; SHLD FIRSTSP
;
; JMP PBEGIN ;START BOOTING SYSTEM.PASCAL
;
;
; INIT$IO: ;INITIALIZE SYSTEM
; LDA CONMOD ;check location CONMOD
; CPI (NOP) ;do we have patched already with NOP?
; RZ ;yes, already done
; CPI (RET) ;there should be a RET statement
; JNZ UNEXP ;print error message if not found
; MVI A,(NOP) ; enable the code that inserts '['
; STA CONMOD ; after escape, added by P. Schorn
; RET ;THAT'S IT
; UNEXP: MVI C,WRBUF ;ask user to check BIOS for correct location
; LXI D,UNEMSG ; of CONMOD
; CALL BDOS ;print error message
; JMP BOOT ;reboot
; UNEMSG: DB CR,LF,'unexpected code found - cannot initialize. ',CR,LF
; DB 'Check BIOS and fix CONMOD in PASCAL.ASM',CR,LF,EOM
;
; REQUEST$DISK: ;ASK FOR PASCAL
; MVI C,WRBUF
; LXI D,DSKMSG
; CALL BDOS
; RD$LOOP: ;THEN WAIT FOR A CR
; MVI C,RDCON
; CALL BDOS
; CPI CR
; JNZ RD$LOOP
; RET
;
; DSKMSG: DB CR,LF,'INSERT PASAL DISK IN DRIVE A, THEN TYPE RETURN',EOM
;
; READ$DIR: ;READ DIRECTORY'S 4 BLOCKS TO BUFFER
; ;BUFFER ADDRESS IS ALREADY IN BC-REG
; MVI E,4 ;DIR IS 4 BLOCKS LONG
; LXI H,2 ;AND STARTS AT BLOCK #2
; CALL SYSRD ;GO GET IT
; RET
;
;
; FIND$INTERP: ;FIND 'SYSTEM.MICRO'
; MVI C,77 ;STOP AFTER THE 77'TH ENTRY
; LHLD DENTP ;GET STARTING ENTRY
; FI$SCH$LP: ;
; LXI D,DTITLE ;ADVANCE TO TITLE STRING
; DAD D
; LXI D,SYSTLE ;SET DE-REG TO COMPARISON STRING
; MVI B,LENGTH+1 ;COMPARISON LENGTH
; FI$CMP$LP: ;START COMPARING
; LDAX D
; CMP M
; JNZ FI$CONT ;IT'S NOT THIS ONE
; INX D ;HEY, WE'VE STILL GOT A CHANCE

```

```

        INX      H
        DCR      B
        JNZ      FI$CMP$LP      ;IS THIS THE END OF THE STRING
        JMP      FI$FOUND
FI$CONT: LHL      DENTP          ;ON TO THE NEXT ENTRY
        LXI      D,DENTSZ
        DAD      D
        SHLD     DENTP
        DCR      C
        JNZ      FI$SCH$LP      ;WAIT, IS THERE ANY DIR LEFT?
;
        MVI      C,WRBUF        ;NO INTERP THERE
        LXI      D,NOTFNDMSG
        CALL     BDOS
;
;
REBOOT: MVI      C,WRBUF        ;TRY TO REBOOT CP/M
        LXI      D,REBOOTMSG
        CALL     BDOS
;
        MVI      C,RDCON
        CALL     BDOS          ;WAIT FOR ANY CHAR
        JMP      BOOT
;
FI$FOUND:                ;WE'VE GOT IT
        RET
;
NOTFNDMSG: DB      CR,LF,'INTERPRETER NOT FOUND',CR,LF,EOM
REBOOTMSG: DB      CR,LF,'REBOOTING CP/M',EOM
;
LENGTH EQU 12             ;TITLE LENGTH
SYSTLE: DB      LENGTH,'SYSTEM.MICRO'
;
;
SAY$LOADING:                ;WE'RE GOING TO LOAD THE INTERPRETER
        MVI      C,WRBUF
        LXI      D,LOADINGMSG
        CALL     BDOS
        RET
;
LOADINGMSG: DB      CR,LF,'LOADING... ',EOM
;
;
READ$INTERP:                ;PUT INTERP IN ITS PLACE
        LHL      DENTP          ;GET STARTING BLOCK
        MOV      E,M            ; INTO HL-REG
        INX      H
        MOV      D,M
;
; COMPUTE THE LENGTH OF THE INTERPETER
        PUSH     D              ;SAVE FIRST BLOCK ON STACK
                                ;TAKE 2'S COMPLIMENT OF FIRST BLOCK
        MOV      A,E
        CMA
        MOV      E,A
        MOV      A,D
        CMA
        MOV      D,A
        INX      D              ;DE=2'S COMP OF FIRST BLOCK
        PUSH     D              ;SAVE ON THE STACK
;
; GET NEXT AVAIL BLOCK
        INX      H
        MOV      E,M
        INX      H
        MOV      D,M
        XCHG
;
        POP      D              ;HL=NXT BLOCK,DE=-(FIRST BLOCK)
                                ;SO HL+DE=LENGTH OF SYSTEM.MICRO
        DAD      D
        XCHG      H
        POP      H              ;HL=LENGTH
                                ;DE=LENGTH
                                ;HL=FIRST BLOCK , DE=LENGTH
;
; CHECK THAT WE WON'T OVERWRITE OURSELVES
        LXI      B,NBLOCKS
        MOV      A,C
        SUB      E
        MOV      A,B
        SBB      D
        JNC      OK            ;JIF OK
                                ;ELSE TELL OPERATOR AND REBOOT
        LXI      D,INTERP$TO$LARGE
        MVI      C,WRBUF
        CALL     BDOS          ;PRINT THE MESSAGE
        JMP      REBOOT        ;REBOOT CPM
;
INTERP$TO$LARGE:
        DB      CR,LF,'INTERPETER TO LARGE IT WILL OVER WRITE'
        DB      CR,LF,'THIS PROGRAM. REASSEMBLE THIS PROGRAM'
        DB      CR,LF,'WITH A HIGHER STARTING ADDRESS',CR,LF,EOM
;
;
OK:     LXI      B,INTERP$BASE  ;AND SET IT LOAD POINT
        CALL     SYSRD          ;THEN READ IT
        RET
;
;
SYSRD:  PUSH     D              ;READ BLOCKS FROM PASCAL DISKETTE
        PUSH     H              ;SAVE BLOCK COUNT
        CALL     READ$RX        ;AND BLOCK NUMBER
        POP      H              ;BUFFER IS ADVANCED BY 512 BYTES
        POP      D
        INX      H              ;ADVANCE TO NEXT BLOCK
        DCR      E              ;BUT, BEFORE WE GO ON
        JNZ      SYSRD          ; SEE IF WE'RE DONE
        RET
;
;
READ$RX: DAD      H              ;READ A PASCAL BLOCK
        DAD      H              ;THERE ARE 4 IBM SECTORS TO A PASCAL BLOCK
        MVI      E,4            ; SO MULT LOGICAL BLOCK BY 4 TO GET 1ST SEC
;
RR$LP:  PUSH     B              ;THIS GETS CONFUSING
        PUSH     D              ;SET BUFFER ADDRESS
        PUSH     H
        CALL     SETDMA
        POP      H
        PUSH     H
        CALL     MAP            ;MAP CONVERTS LOGICAL SECTOR IN HL-REG
        MOV      C,H            ; INTO PHYSICAL TRACK, H-REG, SECTOR, L-REG
        PUSH     H
        CALL     SETTRK

```

```

POP      H
MOV      C,L
CALL     SETSEC
CALL     READ          ;AND READ THE DATA
POP      H
POP      D
POP      B
PUSH     H              ;ADVANCE THE BUFFER ADDRESS
LXI      H,128
DAD      B
MOV      B,H
MOV      C,L
POP      H
INX      H              ;ADVANCE BLOCK COUNT
DCR      E              ;THEN SEE IF WE CONTINUE
JNZ      RR$LP
RET      ;LEAVE, WHEN DONE
;
;
DENTP:   DS      2
;
HOME:    LHL      BOOT+1          ;HOME SELECTED DISK TO TRACK 00
          MVI      L,18H
          PCHL
;
SELDSK:  LHL      BOOT+1          ;SELECT DISK (C-REG)
          MVI      L,1BH
          PCHL
;
SETTRK:  LHL      BOOT+1          ;SET TRACK (C-REG)
          MVI      L,1EH
          PCHL
;
SETSEC:  LHL      BOOT+1          ;SET SECTOR (C-REG)
          MVI      L,21H
          PCHL
;
SETDMA:  LHL      BOOT+1          ;SET DATA TRANSFER ADDRESS (BC-REG)
          MVI      L,24H
          PCHL
;
READ:    LHL      BOOT+1          ;READ A SECTOR TO THE DATA AREA
          MVI      L,27H
          PCHL
;
;
MAP:     ;TURN LSN INTO IBM TRACK-SECTOR
;
NOTE:    TRACK 00 IS NOT USED SO BLOCK 0
          IS AT TRACK 01 SECTOR 1
;
ON ENTRY: HL-REG HAS LOGICAL SECTOR NO.
ON EXIT:  H-REG HAS PHYSICAL TRACK
          L-REG HAS PHYSICAL SECTOR
;
;
PUSH     B
PUSH     D
;
CALL     DIV26
MOV      A,L
ADD      A,A
MOV      B,A
MVI      A,12
CMP      L
JNC      MAPC
INR      B
MAPC:    MOV      C,E
          XRA      A
          MOV      D,A
          MOV      H,A
          MOV      L,B
          MVI      A,6
MAP$LOOP: DAD      D
          DCR      A
          JNZ      MAP$LOOP
          PUSH     B
          CALL     DIV26
          POP      B
          INR      L
          MOV      H,C
          INR      H
          POP      D
          POP      B
          RET
;
;
DIV26:   LXI      B,-26
          MVI      E,0FFH
DIVL:    INR      E
          DAD      B
          MOV      A,H
          ORA      A
          JP       DIVL
          LXI      B,26
          DAD      B
          RET
;
;
END      START

```