

```

TITLE 'ZCPR Version 1.0'
;MARCH 25,1984 MVD.      NOV 15,1985 CHANGED TO 59K
; TO ASSEMBLE:  MAC ZCPR.ASM $+S
; CP/M Z80 COMMAND PROCESSOR REPLACEMENT (CPR) VERSION 1.0
; CCPZ CREATED AND CUSTOMIZED FOR ARIES-II BY RLC
; FURTHER MODIFIED BY RGF AS V2.0
; FURTHER MODIFIED BY RLC AS V2.1
; FURTHER MODIFIED BY KBP AS V2.2
; FURTHER MODIFIED BY RLC AS V2.4 (V2.3 SKIPPED)
; FURTHER MODIFIED BY RLC AS V2.5
; FURTHER MODIFIED BY RLC AS V2.6
; FURTHUR MODIFIED BY SBB AS V2.7
; FURTHER MODIFIED BY RLC AS V2.8
; FURTHER MODIFIED BY RLC AS V2.9
; FURTHER MODIFIED BY RLC AS V3.0
; FURTHER MODIFIED BY RLC AS V3.1
; FURTHER MODIFIED BY RLC AS V4.0
; ZCPR VERSION 1.0 CREATED FROM CCPZ VERSION 4.0 BY RLC IN
;   A COORDINATED EFFORT WITH CCP-GROUP
;
; ZCPR IS A GROUP EFFORT BY CCP-GROUP, WHOSE ACTIVE MEMBERSHIP

```

INVOLVED

```

; IN THIS PROJECT CONSISTS OF THE FOLLOWING:
;   RLC - RICHARD CONN
;   RGF - RON FOWLER
;   KBP - KEITH PETERSON
;   FJW - FRANK WANCHO
; THE FOLLOWING INDIVIDUAL ALSO PROVIDED A CONTRIBUTION:
;   SBB - STEVE BOGOLUB
;
;
;***** STRUCTURE NOTES *****
;
; THIS CPR IS DIVIDED INTO A NUMBER OF MAJOR SECTIONS.  THE

```

FOLLOWING
ROUTINES

```

; IS AN OUTLINE OF THESE SECTIONS AND THE NAMES OF THE MAJOR
; LOCATED THEREIN.
;
; SECTION      FUNCTION/ROUTINES
; -----
;  --         OPENING COMMENTS, EQUATES, AND MACRO DEFINITIONS
;
;  0         JMP TABLE INTO CPR
;
;  1         BUFFERS
;
;  2         CPR STARTING MODULES
;           CPR1  CPR   RESTRT      RSTCPR      RCPRNL
;           PRNNF
;
;  3         UTILITIES
;           CRLF  CONOUT      CONIN  LCOUT  LSTOUT
;           READF READ  BDOSEB PRINTC      PRINT
;           GETDRV  DEFDMA      DMASET  RESET  BDOSJP
;           LOGIN OPENF  OPEN  GRBDOS      CLOSE
;           SEARF SEAR1  SEARN  SUBKIL      DELETE
;           RESETUSR GETUSR  SETUSR

```

```

;
; 4      CPR UTILITIES
;      SETUD SETU0D      UCASE REDBUF      CNVBUF
;      BREAK USRNUM      ERROR SDELM ADVAN
;      SBLANK      ADDAH NUMBER      NUMERR      HEXNUM
;      DIRPTR      SLOGIN      DLOGIN      COMLOG      SCANNER
;      CMDSER
;
; 5      CPR-RESIDENT COMMANDS AND FUNCTIONS
; 5A      DIR      DIRPR FILLQ
; 5B      ERA
; 5C      LIST
; 5D      TYPE      PAGER
; 5E      SAVE
; 5F      REN
; 5G      USER
; 5H      DFU
; 5I      JUMP
; 5J      GO
; 5K      COM      CALLPROG      ERRLOG      ERRJMP
; 5L      GET      MEMLOAD      PRNLE
;
;
0000 =      FALSE      EQU      0
FFFF =      TRUE      EQU      NOT FALSE
;
; CUSTOMIZATION EQUATES
;
; THE FOLLOWING EQUATES MAY BE USED TO CUSTOMIZE THIS CPR FOR THE
USER'S
; SYSTEM AND INTEGRATION TECHNIQUE. THE FOLLOWING CONSTANTS ARE
PROVIDED:
;
; REL - TRUE IF INTEGRATION IS TO BE DONE VIA MOVCPM
; - FALSE IF INTEGRATION IS TO BE DONE VIA DDT AND SYSGEN
;
; BASE - BASE ADDRESS OF USER'S CP/M SYSTEM (NORMALLY 0 FOR DR
VERSION)
; THIS EQUATE ALLOWS EASY MODIFICATION BY NON-STANDARD
CP/M (EG,H89)
;
; CPRLOC - BASE PAGE ADDRESS OF CPR; THIS VALUE CAN BE OBTAINED
BY RUNNING
; THE BDOSLOC PROGRAM ON YOUR SYSTEM, OR BY SETTING THE
; MSIZE AND BIOSEX EQUATES TO THE SYSTEM MEMORY SIZE IN
; K-BYTES AND THE "EXTRA" MEMORY REQUIRED BY YOUR BIOS
; IN K-BYTES. BIOSEX IS ZERO IF YOUR BIOS IS NORMAL SIZE,
; AND CAN BE NEGATIVE IF YOUR BIOS IS IN PROM OR IN
; NON-CONTIGUOUS MEMORY.
;
; RAS - REMOTE-ACCESS SYSTEM; SETTING THIS EQUATE TO TRUE
DISABLES
; CERTAIN CPR COMMANDS THAT ARE CONSIDERED HARMFUL IN A REMOTE-
; ACCESS ENVIRONMENT; USE UNDER REMOTE-ACCESS SYSTEMS (RBBS) FOR
; SECURITY PURPOSES
;
0000 =      REL      EQU      FALSE ;SET TO TRUE FOR MOVCPM INTEGRATION
;
0000 =      BASE      EQU      0      ;BASE OF CP/M SYSTEM (SET FOR STANDARD CP/M)
;
; IF REL
CPRLOC EQU 0 ;MOVCPM IMAGE

```


IN DRIVE

```
; OR ON DRIVE A:.  
; A FINAL DEFINITION OF THE INDIRECT COMMAND FILE ($$$$.SUB OR SUBMIT  
; FILE) IS PRESENTED AS FOLLOWS:  
; "AN INDIRECT COMMAND FILE IS ONE WHICH CONTAINS  
; A SERIES OF COMMANDS EXACTLY AS THEY WOULD BE  
; ENTERED FROM A CP/M CONSOLE. THE SUBMIT COMMAND  
; (OR SUB COMMAND) READS THIS FILES AND TRANSFORMS  
; IT FOR PROCESSING BY THE ZCPR (THE $$$$.SUB FILE).  
; ZCPR WILL THEN EXECUTE THE COMMANDS INDICATED  
; EXACTLY AS IF THEY WERE TYPED AT THE CONSOLE."
```

```

; HENCE, TO PERMIT THIS TO HAPPEN, THE $$$$.SUB FILE MUST ALWAYS
; BE PRESENT ON A SPECIFIC DRIVE, AND A: IS THE CHOICE FOR SAID
DRIVE.

; WITH THIS FACILITY ENGAGED AS SUCH, INDIRECT COMMAND FILES LIKE:
;     DIR
;     A:
;     DIR
; CAN BE EXECUTED, EVEN THOUGH THE CURRENTLY LOGGED-IN DRIVE IS
CHANGED
; DURING EXECUTION. IF THE $$$$.SUB FILE WAS PRESENT ON THE
CURRENTLY
; LOGGED-IN DRIVE, THE ABOVE SERIES OF COMMANDS WOULD NOT WORK SINCE
THE
; ZCPR WOULD BE LOOKING FOR $$$$.SUB ON THE LOGGED-IN DRIVE, AND
SWITCHING
; LOGGED-IN DRIVES WITHOUT MOVING THE $$$$.SUB FILE AS WELL WOULD
CAUSE
; PROCESSING TO ABORT.
;
FFFF = SUBA EQU TRUE ; SET TO TRUE TO HAVE $$$$.SUB ALWAYS ON A:
; SET TO FALSE TO HAVE $$$$.SUB ON THE LOGGED-IN DRIVE
;
; THE FOLLOWING FLAG ENABLES EXTENDED PROCESSING FOR USER-PROGRAM
SUPPLIED
; COMMAND LINES. THIS IS FOR COMMAND LEVEL 3 OF ZCPR. UNDER THE
CCPZ VERSION
; 4.0 PHILOSOPHY, THREE COMMAND LEVELS EXIST:
; (1) THAT COMMAND ISSUED BY THE USER FROM HIS CONSOLE AT THE '>'
PROMPT
; (2) THAT COMMAND ISSUED BY A $$$$.SUB FILE AT THE '$' PROMPT
; (3) THAT COMMAND ISSUED BY A USER PROGRAM BY PLACING THE COMMAND
INTO
; CIBUFF AND SETTING THE CHARACTER COUNT IN CBUFF
; SETTING CLEVEL3 TO TRUE ENABLES EXTENDED PROCESSING OF THE THIRD
LEVEL OF
; ZCPR COMMAND. ALL THE USER PROGRAM NEED DO IS TO STORE THE
COMMAND LINE AND
; SET THE CHARACTER COUNT; ZCPR WILL INITIALIZE THE POINTERS
PROPERLY, STORE
; THE ENDING ZERO PROPERLY, AND CAPITALIZE THE COMMAND LINE FOR
PROCESSING.
; ONCE THE COMMAND LINE IS PROPERLY STORED, THE USER EXECUTES THE
COMMAND LINE
; BY REENTERING THE ZCPR THROUGH CPRLOC [NOTE: THE C REGISTER MUST
CONTAIN
; A VALID USER/DISK FLAG (SEE LOCATION 4) AT THIS TIME.]
;
FFFF = CLEVEL3 EQU TRUE ;ENABLE COMMAND LEVEL 3 PROCESSING
;
;
;*** TERMINAL AND 'TYPE' CUSTOMIZATION EQUATES
;
0018 = NLines EQU 24 ;NUMBER OF LINES ON CRT SCREEN
FFFF = WIDE EQU TRUE ;TRUE IF WIDE DIR DISPLAY
007C = FENCE EQU '|' ;SEP CHAR BETWEEN DIR FILES
;
FFFF = PGDFLT EQU TRUE ;SET TO FALSE TO DISABLE PAGING BY
DEFAULT
0050 = PGDFLG EQU 'P' ;FOR TYPE COMMAND: PAGE OR NOT (DEP ON
PGDFLT)
; THIS FLAG REVERSES THE DEFAULT EFFECT
;
000F = MAXUSR EQU 15 ;MAXIMUM USER NUMBER ACCESSABLE

```

```
0041 =      ;
            SYSFLG EQU  'A'          ;FOR DIR COMMAND: LIST $SYS AND $DIR
            ;
0053 =      ;
            SOFLG  EQU  'S'          ;FOR DIR COMMAND: LIST $SYS FILES ONLY
            ;
0000 =      ;
            SUPRES EQU  FALSE        ;SUPRESSES USER # REPORT FOR USER 0
            ;
0000 =      ;
            DEFUSR EQU  0            ;DEFAULT USER NUMBER FOR COM FILES
            ;
0024 =      ;
            SPRMPT EQU  '$'         ;CPR PROMPT INDICATING SUBMIT COMMAND
003E =      ;
            CPRMPT EQU  '>'         ;CPR PROMPT INDICATING USER COMMAND
```

```

;
0048 = NUMBASE EQU 'H' ;CHARACTER USED TO SWITCH FROM DEFAULT
; NUMBER BASE
;
0053 = SECTFLG EQU 'S' ;OPTION CHAR FOR SAVE COMMAND TO SAVE
SECTORS
;
; END OF CUSTOMIZATION SECTION
;
000D = CR EQU 0DH
000A = LF EQU 0AH
0009 = TAB EQU 09H
;
0000 = WBOOT EQU BASE+0000H ;CP/M WARM BOOT ADDRESS
0004 = UDFLAG EQU BASE+0004H ;USER NUM IN HIGH NYBBLE, DISK IN
LOW
0005 = BDOS EQU BASE+0005H ;BDOS FUNCTION CALL ENTRY PT
005C = TFCB EQU BASE+005CH ;DEFAULT FCB BUFFER
0080 = TBUFF EQU BASE+0080H ;DEFAULT DISK I/O BUFFER
0100 = TPA EQU BASE+0100H ;BASE OF TPA
;
;
; MACROS TO PROVIDE Z80 EXTENSIONS
; MACROS INCLUDE:
;
$-MACRO ;FIRST TURN OFF THE EXPANSIONS
;
; JR - JUMP RELATIVE
; JRC - JUMP RELATIVE IF CARRY
; JRNC - JUMP RELATIVE IF NO CARRY
; JRZ - JUMP RELATIVE IF ZERO
; JRNZ - JUMP RELATIVE IF NO ZERO
; DJNZ - DECREMENT B AND JUMP RELATIVE IF NO ZERO
; LDIR - MOV @HL TO @DE FOR COUNT IN BC
; LXXD - LOAD DOUBLE REG DIRECT
; SXXD - STORE DOUBLE REG DIRECT
;
;
;
; @GENDD MACRO USED FOR CHECKING AND GENERATING
; 8-BIT JUMP RELATIVE DISPLACEMENTS
;
@GENDD MACRO ?DD ;;USED FOR CHECKING RANGE OF 8-BIT DISPLACEMENTS
IF (?DD GT 7FH) AND (?DD LT 0FF80H)
DB 100H ;DISPLACEMENT RANGE ERROR ON JUMP RELATIVE
ELSE
DB ?DD
ENDIF
ENDM
;
;
; Z80 MACRO EXTENSIONS
;
JR MACRO ?N ;;JUMP RELATIVE
DB 18H
@GENDD ?N-$-1
ENDM
;

```

```

JRC      MACRO ?N      ;;JUMP RELATIVE ON CARRY
  DB      38H
  @GENDD  ?N-$-1
  ENDM
;
JRNC     MACRO ?N      ;;JUMP RELATIVE ON NO CARRY
  DB      30H
  @GENDD  ?N-$-1
  ENDM
;
JRZ      MACRO ?N      ;;JUMP RELATIVE ON ZERO
  DB      28H
  @GENDD  ?N-$-1
  ENDM
;
JRNZ     MACRO ?N      ;;JUMP RELATIVE ON NO ZERO
  DB      20H
  @GENDD  ?N-$-1
  ENDM
;
DJNZ     MACRO ?N      ;;DECREMENT B AND JUMP RELATIVE ON NO ZERO
  DB      10H
  @GENDD  ?N-$-1
  ENDM
;
LDIR     MACRO          ;;LDIR
  DB      0EDH,0B0H
  ENDM
;
LDED     MACRO ?N      ;;LOAD DE DIRECT
  DB      0EDH,05BH
  DW      ?N
  ENDM
;
LBCD     MACRO ?N      ;;LOAD BC DIRECT
  DB      0EDH,4BH
  DW      ?N
  ENDM
;
SDED     MACRO ?N      ;;STORE DE DIRECT
  DB      0EDH,53H
  DW      ?N
  ENDM
;
SBCD     MACRO ?N      ;;STORE BC DIRECT
  DB      0EDH,43H
  DW      ?N
  ENDM
;
; END OF Z80 MACRO EXTENSIONS
;
;
;**** SECTION 0 ****
;
D000    ORG    CPRLOC
;

```



```

; ENTRY POINTS INTO ZCPR
; IF THE ZCPR IS ENTERED AT LOCATION CPRLOC (AT THE JMP TO CPR),
THEN
; THE DEFAULT COMMAND IN CIBUFF WILL BE PROCESSED. IF THE ZCPR IS
ENTERED
; AT LOCATION CPRLOC+3 (AT THE JMP TO CPR1), THEN THE DEFAULT
COMMAND IN
; CIBUFF WILL NOT BE PROCESSED.
; NOTE: ENTRY INTO ZCPR IN THIS WAY IS PERMITTED UNDER ZCPR
VERSION 4.0,
; BUT IN ORDER FOR THIS TO WORK, CIBUFF AND CBUFF MUST BE
INITIALIZED PROPERLY
; AND THE C REGISTER MUST CONTAIN A VALID USER/DISK FLAG (SEE
LOCATION 4: THE
; MOST SIGNIFICANT NYBBLE CONTAINS THE USER NUMBER AND THE LEAST
SIGNIFICANT
; NYBBLE CONTAINS THE DISK NUMBER).
; SOME USER PROGRAMS (SUCH AS SYNONYM3) ATTEMPT TO USE THE
DEFAULT
; COMMAND FACILITY. UNDER THE ORIGINAL CPR, IT WAS NECESSARY TO
INITIALIZE
; THE POINTER AFTER THE RESERVED SPACE FOR THE COMMAND BUFFER TO
POINT TO
; THE FIRST BYTE OF THE COMMAND BUFFER. UNDER VERSION 4.X OF ZCPR,
THIS IS
; NO LONGER THE CASE. THE CIBPTR (COMMAND INPUT BUFFER POINTER) IS
LOCATED
; TO BE COMPATABLE WITH SUCH PROGRAMS (PROVIDED THEY DETERMINE THE
BUFFER
; LENGTH FROM THE BYTE AT MBUFF [CPRLOC + 6]), BUT UNDER VERSION 4.X
OF ZCPR
; THIS IS NO LONGER NECESSARY. ZCPR VERSION 4.X AUTOMATICALLY
INITIALIZES
; THIS BUFFER POINTER IN ALL CASES.
;
ENTRY:
D000 C305D1 JMP CPR ; PROCESS POTENTIAL DEFAULT COMMAND
D003 C301D1 JMP CPR1 ; DO NOT PROCESS POTENTIAL DEFAULT COMMAND
;
;**** SECTION 1 ****
; BUFFERS ET AL
;
; INPUT COMMAND LINE AND DEFAULT COMMAND
; THE COMMAND LINE TO BE EXECUTED IS STORED HERE. THIS COMMAND
LINE
; IS GENERATED IN ONE OF THREE WAYS:
; (1) BY THE USER ENTERING IT THROUGH THE BDOS READLN FUNCTION AT
; THE DU> PROMPT [USER INPUT FROM KEYBOARD]
; (2) BY THE SUBMIT FILE FACILITY PLACING IT THERE FROM A $$$$.SUB
; FILE
; (3) BY AN EXTERNAL PROGRAM OR USER PLACING THE REQUIRED COMMAND
; INTO THIS BUFFER
; IN ALL CASES, THE COMMAND LINE IS PLACED INTO THE BUFFER
STARTING AT
; CIBUFF. THIS COMMAND LINE IS TERMINATED BY THE LAST CHARACTER
(NOT CARRIAGE
; RETURN), AND A CHARACTER COUNT OF ALL CHARACTERS IN THE COMMAND
LINE
; UP TO AND INCLUDING THE LAST CHARACTER IS PLACED INTO LOCATION
CBUFF
; (IMMEDIATELY BEFORE THE COMMAND LINE AT CIBUFF). THE PLACED
COMMAND LINE
; IS THEN PARSED, INTERPRETED, AND THE INDICATED COMMAND IS

```

EXECUTED.

COMMAND

IS

PTR).

ENOUGH

IS NOT

THIS FOR

LENGTH.

BE

0050 =

D006 50

```
; IF CLEVEL3 IS PERMITTED, A TERMINATING ZERO IS PLACED AFTER THE
; (OTHERWISE THE USER PROGRAM HAS TO PLACE THIS ZERO) AND THE CIBPTR
; PROPERLY INITIALIZED (OTHERWISE THE USER PROGRAM HAS TO INIT THIS
; IF THE COMMAND IS PLACED BY A USER PROGRAM, ENTERING AT CPRLOC IS
; TO HAVE THE COMMAND PROCESSED. AGAIN, UNDER CCPZ VERSION 4.0, IT
; NECESSARY TO STORE THE POINTER TO CIBUFF IN CIBPTR; ZCPR WILL DO
; THE CALLING PROGRAM IF CLEVEL3 IS MADE TRUE.
; WARNING: THE COMMAND LINE MUST NOT EXCEED BUFLLEN CHARACTERS IN
; FOR USER PROGRAMS WHICH LOAD THIS COMMAND, THE VALUE OF BUFLLEN CAN
; OBTAINED BY EXAMINING THE BYTE AT MBUFF (CPRLOC + 6).
;
BUFLLEN EQU 80 ;MAXIMUM BUFFER LENGTH
MBUFF:
DB BUFLLEN ;MAXIMUM BUFFER LENGTH
```

```

CIBUFF:
D007 00      DB      0          ;NUMBER OF VALID CHARS IN COMMAND LINE
CIBUFF:
D008 2020202020 DB      '          ' ;DEFAULT (COLD BOOT) COMMAND
CIBUF:
D017 00      DB      0          ;COMMAND STRING TERMINATOR
D018        DS      BUFLen-(%-CIBUFF)+1 ;TOTAL IS 'BUFLen' BYTES
;
CIBPTR:
D059 08D0    DW      CIBUFF      ;POINTER TO COMMAND INPUT BUFFER
CIPTR:
D05B 17D0    DW      CIBUF      ;CURRENT POINTER
;
D05D        DS      26          ;STACK AREA
D077 =      STACK EQU      $      ;TOP OF STACK
;
; FILE TYPE FOR COMMAND
;
COMMSG:
D077 434F4D  DB      'COM'
;
; SUBMIT FILE CONTROL BLOCK
;
SUBFCB:
D07A 01      IF      SUBA          ;IF $$$ .SUB ON A:
DB      1          ;DISK NAME SET TO DEFAULT TO DRIVE A:
ENDIF
;
IF      NOT SUBA      ;IF $$$ .SUB ON CURRENT DRIVE
DB      0          ;DISK NAME SET TO DEFAULT TO CURRENT DRIVE
ENDIF
;
D07B 242424  DB      '$$$'          ;FILE NAME
D07E 2020202020 DB      '          '
D083 535542  DB      'SUB'          ;FILE TYPE
D086 00      DB      0          ;EXTENT NUMBER
D087 00      DB      0          ;S1
SUBFS2:
D088        DS      1          ;S2
SUBFRC:
D089        DS      1          ;RECORD COUNT
D08A        DS      16         ;DISK GROUP MAP
SUBFCR:
D09A        DS      1          ;CURRENT RECORD NUMBER
;
; COMMAND FILE CONTROL BLOCK
;
FCBDN:
D09B        DS      1          ;DISK NAME
FCBFN:
D09C        DS      8          ;FILE NAME
FCBFT:
D0A4        DS      3          ;FILE TYPE
D0A7        DS      1          ;EXTENT NUMBER
D0A8        DS      2          ;S1 AND S2
D0AA        DS      1          ;RECORD COUNT

```

```

FCBDM:
D0AB      DS      16          ;DISK GROUP MAP
FCBCR:
D0BB      DS      1          ;CURRENT RECORD NUMBER
;
; OTHER BUFFERS
;
PAGCNT:
D0BC 16   DB      N LINES-2   ;LINES LEFT ON PAGE
CHRCNT:
D0BD 00   DB      0           ;CHAR COUNT FOR TYPE
QMCNT:
D0BE 00   DB      0           ;QUESTION MARK COUNT FOR FCB TOKEN SCANNER
;
; CPR BUILT-IN COMMAND TABLE
;
0004 =    NCHARS EQU      4           ;NUMBER OF CHARS/COMMAND
;
; CPR COMMAND NAME TABLE
; EACH TABLE ENTRY IS COMPOSED OF THE 4-BYTE COMMAND AND 2-BYTE
ADDRESS
;
CMDTBL:
D0BF 44495220 DB      'DIR '
D0C3 81D4     DW      DIR
D0C5 4C495354 DB      'LIST'
D0C9 5AD5     DW      LIST
D0CB 54595045 DB      'TYPE'
D0CF 5ED5     DW      TYPE
D0D1 55534552 DB      'USER'
D0D5 BDD6     DW      USER
D0D7 44465520 DB      'DFU '
D0DB C7D6     DW      DFU
;
IF      NOT RAS          ;FOR NON-RAS
D0DD 474F2020 DB      'GO '
D0E1 D4D6     DW      GO
D0E3 45524120 DB      'ERA '
D0E7 32D5     DW      ERA
D0E9 53415645 DB      'SAVE'
D0ED F7D5     DW      SAVE
D0EF 52454E20 DB      'REN '
D0F3 79D6     DW      REN
D0F5 47455420 DB      'GET '
D0F9 7ED7     DW      GET
D0FB 4A554D50 DB      'JUMP'
D0FF CFD6     DW      JUMP
ENDIF
;
000B =    NCMNDS EQU      ($-CMDTBL)/(NCHARS+2)
;
;
;**** SECTION 2 ****
; CPR STARTING POINTS
;
; START CPR AND DON'T PROCESS DEFAULT COMMAND STORED
;

```

```

CPR1:
D101 AF      XRA   A           ;SET NO DEFAULT COMMAND
D102 3207D0  STA   CBUFF
;
; START CPR AND POSSIBLY PROCESS DEFAULT COMMAND
;
; NOTE ON MODIFICATION BY RGF: BDOS RETURNS 0FFH IN
; ACCUMULATOR WHENEVER IT LOGS IN A DIRECTORY, IF ANY
; FILE NAME CONTAINS A '$' IN IT. THIS IS NOW USED AS
; A CLUE TO DETERMINE WHETHER OR NOT TO DO A SEARCH
; FOR SUBMIT FILE, IN ORDER TO ELIMINATE WASTEFUL SEARCHES.
;
CPR:
D105 3177D0  LXI   SP,STACK   ;RESET STACK
D108 C5      PUSH  B
D109 79      MOV   A,C           ;C=USER/DISK NUMBER (SEE LOC 4)
D10A 1F      RAR                   ;EXTRACT USER NUMBER
D10B 1F      RAR
D10C 1F      RAR
D10D 1F      RAR
D10E E60F   ANI   0FH
D110 5F      MOV   E,A           ;SET USER NUMBER
D111 CD45D2  CALL  SETUSR
D114 CD08D2  CALL  RESET           ;RESET DISK SYSTEM
D117 322AD1  STA   RNGSUB           ;SAVE SUBMIT CLUE FROM DRIVE A:
D11A C1      POP   B
D11B 79      MOV   A,C           ;C=USER/DISK NUMBER (SEE LOC 4)
D11C E60F   ANI   0FH           ;EXTRACT DEFAULT DISK DRIVE
D11E 3259D2  STA   TDRIVE          ;SET IT
D121        JRZ   NOLOG          ;SKIP IF 0...ALREADY LOGGED
D123 CD0DD2  CALL  LOGIN           ;LOG IN DEFAULT DISK
;
; IF $$$$.SUB IS ON CURRENT DRIVE
;   STA   RNGSUB           ;BDOS '$' CLUE
;   ENDDIF
;
NOLOG:
D126 117AD0  LXI   D,SUBFCB       ;CHECK FOR $$$$.SUB ON CURRENT DISK
D12A =      RNGSUB EQU   $+1       ;POINTER FOR IN-THE-CODE MODIFICATION
D129 3E00   MVI   A,0           ;2ND BYTE (IMMEDIATE ARG) IS THE RNGSUB FLAG
D12B B7      ORA   A           ;SET FLAGS ON CLUE
D12C 2F      CMA                   ;PREPARE FOR COMING 'CMA'
D12D C427D2  CNZ   SEAR1
D130 2F      CMA           ;0FFH IS RETURNED IF NO $$$$.SUB, SO COMPLEMENT
D131 322AD1  STA   RNGSUB           ;SET FLAG (0=NO $$$$.SUB)
D134 3A07D0  LDA   CBUFF           ;EXECUTE DEFAULT COMMAND?
D137 B7      ORA   A           ;0=NO
D138        JRNZ  RS1
;
; PROMPT USER AND INPUT COMMAND LINE FROM HIM
;
;
RESTR1:
D13A 3177D0  LXI   SP,STACK   ;RESET STACK
;
; PRINT PROMPT (DU>)
;

```

```

D13D CDA7D1      CALL  CRLF          ;PRINT PROMPT
D140 CDFDD1      CALL  GETDRV         ;CURRENT DRIVE IS PART OF PROMPT
D143 C641        ADI   'A'          ;CONVERT TO ASCII A-P
D145 CDAED1      CALL  CONOUT        ;
D148 CD43D2      CALL  GETUSR         ;GET USER NUMBER
;
; IF SUPRES          ;IF SUPPRESSING USR # REPORT FOR USR 0
ORA  A
JRZ  RS000
ENDIF
;
D14B FE0A        CPI   10           ;USER < 10?
D14D             JRC   RS00
D14F D60A        SUI   10           ;SUBTRACT 10 FROM IT
D151 F5          PUSH  PSW          ;SAVE IT
D152 3E31        MVI  A,'1'        ;OUTPUT 10'S DIGIT
D154 CDAED1      CALL  CONOUT
D157 F1          POP   PSW
RS00:
D158 C630        ADI   '0'          ;OUTPUT 1'S DIGIT (CONVERT TO ASCII)
D15A CDAED1      CALL  CONOUT
;
; READ INPUT LINE FROM USER OR $$$$.SUB
;
RS000:
D15D CD67D2      CALL  REDBUF         ;INPUT COMMAND LINE FROM USER (OR $$$$.SUB)
;
; PROCESS INPUT LINE
;
RS1:
;
; IF CLEVEL3        ;IF THIRD COMMAND LEVEL IS PERMITTED
D160 CDC8D2      CALL  CNVBUF         ;CAPITALIZE COMMAND LINE, PLACE ENDING 0,
; AND SET CIBPTR VALUE
ENDIF
;
D163 CD01D2      CALL  DEFDMA          ;SET TBUF TO DMA ADDRESS
D166 CDFDD1      CALL  GETDRV         ;GET DEFAULT DRIVE NUMBER
D169 3259D2      STA  TDRIVE         ;SET IT
D16C CDE0D3      CALL  SCANNER        ;PARSE COMMAND NAME FROM COMMAND LINE
D16F C4EFD2      CNZ  ERROR          ;ERROR IF COMMAND NAME CONTAINS A '?'
D172 1188D1      LXI  D,RSTCPR       ;PUT RETURN ADDRESS OF COMMAND
D175 D5          PUSH  D             ;ON THE STACK
D176 3AD7D3      LDA  TEMPDR         ;IS COMMAND OF FORM 'D:COMMAND'?
D179 B7          ORA  A             ;NZ=YES
D17A C2D9D6      JNZ  COM            ; IMMEDIATELY
D17D CD5FD4      CALL  CMDSER        ;SCAN FOR CPR-RESIDENT COMMAND
D180 C2D9D6      JNZ  COM            ;NOT CPR-RESIDENT
D183 7E          MOV  A,M           ;FOUND IT: GET LOW-ORDER PART
D184 23          INX  H             ;GET HIGH-ORDER PART
D185 66          MOV  H,M           ;STORE HIGH
D186 6F          MOV  L,A           ;STORE LOW
D187 E9          PCHL              ;EXECUTE CPR ROUTINE
;
; ENTRY POINT FOR RESTARTING CPR AND LOGGING IN DEFAULT DRIVE
;

```

```

RSTCPR:
D188 CDCCD3      CALL  DLOGIN          ;LOG IN DEFAULT DRIVE
;
; ENTRY POINT FOR RESTARTING CPR WITHOUT LOGGING IN DEFAULT DRIVE
;
RCPRNL:
D18B CDE0D3      CALL  SCANER          ;EXTRACT NEXT TOKEN FROM COMMAND LINE
D18E 3A9CD0      LDA   FCBFN          ;GET FIRST CHAR OF TOKEN
D191 D620        SUI   ' '          ;ANY CHAR?
D193 21D7D3      LXI   H,TEMPDR
D196 B6          ORA   M
D197 C2EFD2      JNZ   ERROR
D19A             JR    RESTRT
;
; NO FILE ERROR MESSAGE
;
PRNNF:
D19C CDE5D1      CALL  PRINTC          ;NO FILE MESSAGE
D19F 4E6F204669 DB   'No Fil','e'+80H
D1A6 C9          RET
;
;**** SECTION 3 ****
; I/O UTILITIES
;
; OUTPUT CHAR IN REG A TO CONSOLE AND DON'T CHANGE BC
;
; OUTPUT <CRLF>
;
CRLF:
D1A7 3E0D        MVI   A,CR
D1A9 CDAED1      CALL  CONOUT
D1AC 3E0A        MVI   A,LF ;FALL THRU TO CONOUT
;
CONOUT:
D1AE C5          PUSH  B
D1AF 0E02        MVI   C,02H
;
OUTPUT:
D1B1 5F          MOV   E,A
D1B2 E5          PUSH  H
D1B3 CD0500      CALL  BDOS
D1B6 E1          POP   H
D1B7 C1          POP   B
D1B8 C9          RET
;
CONIN:
D1B9 0E01        MVI   C,01H ;GET CHAR FROM CON: WITH ECHO
D1BB CDDED1      CALL  BDOSB
D1BE C35ED2      JMP   UCASE ;CAPITALIZE
;
LCOUT:
D1C1 F5          PUSH  PSW ;OUTPUT CHAR TO CON: OR LST: DEP ON PRFLG
D1C3 =           PRFLG EQU $+1 ;POINTER FOR IN-THE-CODE MODIFICATION
D1C2 3E00        MVI   A,0 ;2ND BYTE (IMMEDIATE ARG) IS THE PRINT FLAG
D1C4 B7          ORA   A ;0=TYPE
D1C5             JRZ   LC1

```

```

D1C7 F1          POP   PSW   ;GET CHAR
                ;
                ; OUTPUT CHAR IN REG A TO LIST DEVICE
                ;
                LSTOUT:
D1C8 C5          PUSH   B
D1C9 0E05        MVI    C,05H
D1CB            JR     OUTPUT
                LC1:
D1CD F1          POP   PSW   ;GET CHAR
D1CE F5          PUSH   PSW
D1CF CDAED1      CALL   CONOUT   ;OUTPUT TO CON:
D1D2 F1          POP   PSW
D1D3 FE0A        CPI    LF    ;CHECK FOR PAGING
D1D5 CADED5      JZ     PAGER
D1D8 C9          RET
                ;
                READF:
D1D9 119BD0      LXI    D,FCBDN ;FALL THRU TO READ
                READ:
D1DC 0E14        MVI    C,14H ;FALL THRU TO BDOSB
                ;
                ; CALL BDOS AND SAVE BC
                ;
                BDOSB:
D1DE C5          PUSH   B
D1DF CD0500      CALL   BDOS
D1E2 C1          POP   B
D1E3 B7          ORA   A
D1E4 C9          RET
                ;
                ; PRINT STRING (ENDING IN 0) PTED TO BY RET ADR;START WITH <CRLF>
                ;
                PRINTC:
D1E5 F5          PUSH   PSW           ;SAVE FLAGS
D1E6 CDA7D1      CALL   CRLF          ;NEW LINE
D1E9 F1          POP   PSW
                ;
                PRINT:
D1EA E3          XTHL           ;GET PTR TO STRING
D1EB F5          PUSH   PSW           ;SAVE FLAGS
D1EC CDF2D1      CALL   PRIN1          ;PRINT STRING
D1EF F1          POP   PSW           ;GET FLAGS
D1F0 E3          XTHL           ;RESTORE HL AND RET ADR
D1F1 C9          RET
                ;
                ; PRINT STRING (ENDING IN 0) PTED TO BY HL
                ;
                PRIN1:
D1F2 7E          MOV    A,M           ;GET NEXT BYTE
D1F3 CDAED1      CALL   CONOUT          ;PRINT CHAR
D1F6 7E          MOV    A,M           ;GET NEXT BYTE AGAIN FOR TEST
D1F7 23          INX   H             ;PT TO NEXT BYTE
D1F8 B7          ORA   A             ;SET FLAGS
D1F9 C8          RZ                ;DONE IF ZERO
D1FA F8          RM                ;DONE IF MSB SET

```



```

D1FB          JR    PRIN1
              ;
              ; BDOS FUNCTION ROUTINES
              ;
              ;
              ; RETURN NUMBER OF CURRENT DISK IN A
              ;
              GETDRV:
D1FD 0E19     MVI    C,19H
D1FF          JR    BDOSJP
              ;
              ; SET 80H AS DMA ADDRESS
              ;
              DEFDMA:
D201 118000   LXI    D,TBUFF      ;80H=TBUFF
              DMASET:
D204 0E1A     MVI    C,1AH
D206          JR    BDOSJP
              ;
              RESET:
D208 0E0D     MVI    C,0DH
              BDOSJP:
D20A C30500   JMP    BDOS
              ;
              LOGIN:
D20D 5F       MOV    E,A
D20E 0E0E     MVI    C,0EH
D210          JR    BDOSJP      ;SAVE SOME CODE SPACE
              ;
              OPENF:
D212 AF       XRA    A
D213 32BBD0   STA    FCBCR
D216 119BD0   LXI    D,FCBDN ;FALL THRU TO OPEN
              ;
              OPEN:
D219 0E0F     MVI    C,0FH ;FALL THRU TO GRBDOS
              ;
              GRBDOS:
D21B CD0500   CALL   BDOS
D21E 3C       INR    A      ;SET ZERO FLAG FOR ERROR RETURN
D21F C9       RET
              ;
              CLOSE:
D220 0E10     MVI    C,10H
D222          JR    GRBDOS
              ;
              SEARF:
D224 119BD0   LXI    D,FCBDN ;SPECIFY FCB
              SEAR1:
D227 0E11     MVI    C,11H
D229          JR    GRBDOS
              ;
              SEARN:
D22B 0E12     MVI    C,12H
D22D          JR    GRBDOS
              ;

```

```

; CHECK FOR SUBMIT FILE IN EXECUTION AND ABORT IT IF SO
;
SUBKIL:
D22F 212AD1    LXI  H,RNGSUB    ;CHECK FOR SUBMIT FILE IN EXECUTION
D232 7E        MOV  A,M
D233 B7        ORA  A          ;0=NO
D234 C8        RZ
D235 3600      MVI  M,0          ;ABORT SUBMIT FILE
D237 117AD0    LXI  D,SUBFCB    ;DELETE $$$ .SUB
;
DELETE:
D23A 0E13      MVI  C,13H
D23C           JR   BDOSJP    ;SAVE MORE SPACE
;
; RESET USER NUMBER IF CHANGED
;
RESETUSR:
D23F =         TMPUSR EQU  $+1          ;POINTER FOR IN-THE-CODE MODIFICATION
D23E 3E00      MVI  A,0          ;2ND BYTE (IMMEDIATE ARG) IS TMPUSR
D240 5F        MOV  E,A          ;PLACE IN E
D241           JR   SETUSR    ;THEN GO SET USER
GETUSR:
D243 1EFF      MVI  E,0FFH        ;GET CURRENT USER NUMBER
SETUSR:
D245 0E20      MVI  C,20H        ;SET USER NUMBER TO VALUE IN E (GET IF E=FFH)
D247           JR   BDOSJP    ;MORE SPACE SAVING
;
; END OF BDOS FUNCTIONS
;
;
;**** SECTION 4 ****
; CPR UTILITIES
;
; SET USER/DISK FLAG TO CURRENT USER AND DEFAULT DISK
;
SETUD:
D249 CD43D2    CALL  GETUSR          ;GET NUMBER OF CURRENT USER
D24C 87        ADD  A          ;PLACE IT IN HIGH NYBBLE
D24D 87        ADD  A
D24E 87        ADD  A
D24F 87        ADD  A
D250 2159D2    LXI  H,TDRIVE    ;MASK IN DEFAULT DRIVE NUMBER (LOW NYBBLE)
D253 B6        ORA  M          ;MASK IN
D254 320400    STA  UDFLAG        ;SET USER/DISK NUMBER
D257 C9        RET
;
; SET USER/DISK FLAG TO USER 0 AND DEFAULT DISK
;
SETU0D:
D259 =         TDRIVE EQU  $+1          ;POINTER FOR IN-THE-CODE MODIFICATION
D258 3E00      MVI  A,0          ;2ND BYTE (IMMEDIATE ARG) IS TDRIVE
D25A 320400    STA  UDFLAG        ;SET USER/DISK NUMBER
D25D C9        RET
;
; CONVERT CHAR IN A TO UPPER CASE
;

```

```

UCASE:
D25E FE61      CPI    61H          ;LOWER-CASE A
D260 D8        RC
D261 FE7B      CPI    7BH          ;GREATER THAN LOWER-CASE Z?
D263 D0        RNC
D264 E65F      ANI    5FH          ;CAPITALIZE
D266 C9        RET

;
; INPUT NEXT COMMAND TO CPR
; THIS ROUTINE DETERMINES IF A SUBMIT FILE IS BEING PROCESSED
; AND EXTRACTS THE COMMAND LINE FROM IT IF SO OR FROM THE USER'S
CONSOLE

;
REDBUF:
D267 3A2AD1    LDA    RNGSUB              ;SUBMIT FILE CURRENTLY IN EXECUTION?
D26A B7        ORA    A                ;0=NO
D26B           JRZ    RB1              ;GET LINE FROM CONSOLE IF NOT
D26D 117AD0    LXI    D,SUBFCB         ;OPEN $$$$.SUB
D270 D5        PUSH   D                ;SAVE DE
D271 CD19D2    CALL   OPEN
D274 D1        POP    D                ;RESTORE DE
D275           JRZ    RB1              ;ERASE $$$$.SUB IF END OF FILE AND GET CMND
D277 3A89D0    LDA    SUBFRC           ;GET VALUE OF LAST RECORD IN FILE
D27A 3D        DCR    A                ;PT TO NEXT TO LAST RECORD
D27B 329AD0    STA    SUBFCR           ;SAVE NEW VALUE OF LAST RECORD IN $$$$.SUB
D27E CDDCD1    CALL   READ              ;DE=SUBFCB
D281           JRNZ   RB1              ;ABORT $$$$.SUB IF ERROR IN READING LAST REC
D283 1107D0    LXI    D,CBUFF          ;COPY LAST RECORD (NEXT SUBMIT CMND) TO CBUFF
D286 218000    LXI    H,TBUFF          ; FROM TBUFF
D289 015000    LXI    B,BUFLEN         ;NUMBER OF BYTES
D28C           LDIR
D28E 2188D0    LXI    H,SUBFS2          ;PT TO S2 OF $$$$.SUB FCB
D291 3600      MVI    M,0                ;SET S2 TO ZERO
D293 23        INX    H                ;PT TO RECORD COUNT
D294 35        DCR    M                ;DECREMENT RECORD COUNT OF $$$$.SUB
D295 117AD0    LXI    D,SUBFCB         ;CLOSE $$$$.SUB
D298 CD20D2    CALL   CLOSE
D29B           JRZ    RB1              ;ABORT $$$$.SUB IF ERROR
D29D 3E24      MVI    A,SPRMPT          ;PRINT SUBMIT PROMPT
D29F CDAED1    CALL   CONOUT
D2A2 2108D0    LXI    H,CIBUFF          ;PRINT COMMAND LINE FROM $$$$.SUB
D2A5 CDF2D1    CALL   PRIN1
D2A8 CDDDED2  CALL   BREAK              ;CHECK FOR ABORT (ANY CHAR)

;
; IF CLEVEL3 ;IF THIRD COMMAND LEVEL IS PERMITTED
D2AB C8        RZ
;IF <NULL> (NO ABORT), RETURN TO CALLER AND RUN
ENDIF

;
; IF NOT CLEVEL3 ;IF THIRD COMMAND LEVEL IS NOT PERMITTED
JRZ    CNVBUF          ;IF <NULL> (NO ABORT), CAPITALIZE COMMAND
ENDIF

;
D2AC CD2FD2    CALL   SUBKIL              ;KILL $$$$.SUB IF ABORT
D2AF C33AD1    JMP    RESTRT              ;RESTART CPR

;
; INPUT COMMAND LINE FROM USER CONSOLE
;

```

```

RB1:
D2B2 CD2FD2    CALL  SUBKIL                ;ERASE $$$ .SUB IF PRESENT
D2B5 CD49D2    CALL  SETUD                 ;SET USER AND DISK
D2B8 3E3E      MVI   A,CPRMPT         ;PRINT PROMPT
D2BA CDAED1    CALL  CONOUT
D2BD 0E0A      MVI   C,0AH           ;READ COMMAND LINE FROM USER
D2BF 1106D0    LXI   D,MBUFF
D2C2 CD0500    CALL  BDOS

;
D2C5 C358D2    IF    CLEVEL3             ;IF THIRD COMMAND LEVEL IS PERMITTED
                JMP    SETU0D           ;SET CURRENT DISK NUMBER IN LOWER PARAMS
                ENDIF

;
                IF    NOT CLEVEL3       ;IF THIRD COMMAND LEVEL IS NOT PERMITTED
                CALL  SETU0D           ;SET CURRENT DISK NUMBER IF LOWER PARAMS
                ; AND FALL THRU TO CNVBUF
                ENDIF

;
; CAPITALIZE STRING (ENDING IN 0) IN CBUFF AND SET PTR FOR PARSING
;
CNVBUF:
D2C8 2107D0    LXI   H,CBUFF         ;PT TO USER'S COMMAND
D2CB 46        MOV   B,M             ;CHAR COUNT IN B
D2CC 04        INR   B               ;ADD 1 IN CASE OF ZERO

CB1:
D2CD 23        INX   H               ;PT TO 1ST VALID CHAR
D2CE 7E        MOV   A,M             ;CAPITALIZE COMMAND CHAR
D2CF CD5ED2    CALL  UCASE
D2D2 77        MOV   M,A
D2D3          DJNZ  CB1             ;CONTINUE TO END OF COMMAND LINE

CB2:
D2D5 3600      MVI   M,0             ;STORE ENDING <NULL>
D2D7 2108D0    LXI   H,CIBUFF        ;SET COMMAND LINE PTR TO 1ST CHAR
D2DA 2259D0    SHLD  CIBPTR
D2DD C9        RET

;
; CHECK FOR ANY CHAR FROM USER CONSOLE;RET W/ZERO SET IF NONE
;
BREAK:
D2DE D5        PUSH  D               ;SAVE DE
D2DF 0E0B      MVI   C,11           ;CSTS CHECK
D2E1 CDDED1    CALL  BDOSB
D2E4 C4B9D1    CNZ   CONIN          ;GET INPUT CHAR

BRKBK:
D2E7 D1        POP   D
D2E8 C9        RET

;
; GET THE REQUESTED USER NUMBER FROM THE COMMAND LINE AND VALIDATE
IT.

;
USRNUM:
D2E9 CD3BD3    CALL  NUMBER
D2EC FE10      CPI   MAXUSR+1
D2EE D8        RC

;
; INVALID COMMAND -- PRINT IT
;

```

```

ERROR:
D2EF CDA7D1      CALL  CRLF          ;NEW LINE
D2F2 2A5BD0      LHLD  CIPTR         ;PT TO BEGINNING OF COMMAND LINE
ERR2:
D2F5 7E          MOV   A,M           ;GET CHAR
D2F6 FE21        CPI   ' '+1       ;SIMPLE '?' IF <SP> OR LESS
D2F8             JRC   ERR1
D2FA E5          PUSH  H           ;SAVE PTR TO ERROR COMMAND CHAR
D2FB CDAED1      CALL  CONOUT        ;PRINT COMMAND CHAR
D2FE E1          POP   H           ;GET PTR
D2FF 23          INX   H           ;PT TO NEXT
D300             JR    ERR2        ;CONTINUE

ERR1:
D302 CDEAD1      CALL  PRINT        ;PRINT '?'
D305 BF          DB    '?' + 80H
D306 CD2FD2      CALL  SUBKIL        ;TERMINATE ACTIVE $$$ .SUB IF ANY
D309 C33AD1      JMP   RESTRT        ;RESTART CPR
;
; CHECK TO SEE IF DE PTS TO DELIMITER; IF SO, RET W/ZERO FLAG SET
;
SDELM:
D30C 1A          LDAX  D
D30D B7          ORA   A           ;0=DELIMITER
D30E C8          RZ
D30F FE20        CPI   ' '         ;ERROR IF < <SP>
D311             JRC   ERROR
D313 C8          RZ           ;<SP>=DELIMITER
D314 FE3D        CPI   '='        ;' '=DELIMITER
D316 C8          RZ
D317 FE5F        CPI   5FH        ;UNDERSCORE=DELIMITER
D319 C8          RZ
D31A FE2E        CPI   '.'        ;'.' =DELIMITER
D31C C8          RZ
D31D FE3A        CPI   ':'        ;':' =DELIMITER
D31F C8          RZ
D320 FE3B        CPI   ';'        ;';' =DELIMITER
D322 C8          RZ
D323 FE3C        CPI   '<'        ;'<' =DELIMITER
D325 C8          RZ
D326 FE3E        CPI   '>'        ;'>' =DELIMITER
D328 C9          RET
;
; ADVANCE INPUT PTR TO FIRST NON-BLANK AND FALL THROUGH TO SBLANK
;
ADVAN:
D329             LDED  CIBPTR
;
; SKIP STRING PTED TO BY DE (STRING ENDS IN 0) UNTIL END OF STRING
; OR NON-BLANK ENCOUNTERED (BEGINNING OF TOKEN)
;
SBLANK:
D32D 1A          LDAX  D
D32E B7          ORA   A
D32F C8          RZ
D330 FE20        CPI   ' '
D332 C0          RNZ

```

```

D333 13      INX   D
D334          JR    SBLANK
;
; ADD A TO HL (HL=HL+A)
;
ADDAH:
D336 85      ADD   L
D337 6F      MOV   L,A
D338 D0      RNC
D339 24      INR   H
D33A C9      RET
;
; EXTRACT DECIMAL NUMBER FROM COMMAND LINE
; RETURN WITH VALUE IN REG A;ALL REGISTERS MAY BE AFFECTED
;
NUMBER:
D33B CDE0D3  CALL  SCANNER          ;PARSE NUMBER AND PLACE IN FCBFN
D33E 21A6D0  LXI   H,FCBFN+10      ;PT TO END OF TOKEN FOR CONVERSION
D341 060B    MVI   B,11          ;11 CHARS MAX
;
; CHECK FOR SUFFIX FOR HEXADECIMAL NUMBER
;
NUMS:
D343 7E      MOV   A,M          ;GET CHARS FROM END, SEARCHING FOR SUFFIX
D344 2B      DCX   H          ;BACK UP
D345 FE20    CPI   ' '          ;SPACE?
D347          JRNZ  NUMS1        ;CHECK FOR SUFFIX
D349          DJNZ  NUMS         ;COUNT DOWN
D34B          JR    NUM0         ;BY DEFAULT, PROCESS
NUMS1:
D34D FE48    CPI   NUMBASE        ;CHECK AGAINST BASE SWITCH FLAG
D34F          JRZ   HNUM0
;
; PROCESS DECIMAL NUMBER
;
NUM0:
D351 219CD0  LXI   H,FCBFN          ;PT TO BEGINNING OF TOKEN
D354 010011  LXI   B,1100H          ;C=ACCUMULATED VALUE, B=CHAR COUNT
; (C=0, B=11)
NUM1:
D357 7E      MOV   A,M          ;GET CHAR
D358 FE20    CPI   ' '          ;DONE IF <SP>
D35A          JRZ   NUM2
D35C 23      INX   H          ;PT TO NEXT CHAR
D35D D630    SUI   '0'          ;CONVERT TO BINARY (ASCII 0-9 TO BINARY)
D35F FE0A    CPI   10          ;ERROR IF >= 10
D361          JRNC  NUMERR
D363 57      MOV   D,A          ;DIGIT IN D
D364 79      MOV   A,C          ;NEW VALUE = OLD VALUE * 10
D365 07      RLC
D366 07      RLC
D367 07      RLC
D368 81      ADD   C          ;CHECK FOR RANGE ERROR
D369          JRC   NUMERR
D36B 81      ADD   C          ;CHECK FOR RANGE ERROR
D36C          JRC   NUMERR

```

```

D36E 82      ADD    D          ;NEW VALUE = OLD VALUE * 10 + DIGIT
D36F        JRC    NUMERR      ;CHECK FOR RANGE ERROR
D371 4F      MOV    C,A        ;SET NEW VALUE
D372        DJNZ  NUM1        ;COUNT DOWN
;
; RETURN FROM NUMBER
;
NUM2:
D374 79      MOV    A,C        ;GET ACCUMULATED VALUE
D375 C9      RET
;
; NUMBER ERROR ROUTINE FOR SPACE CONSERVATION
;
NUMERR:
D376 C3EFD2  JMP    ERROR        ;USE ERROR ROUTINE - THIS IS RELATIVE PT
;
; EXTRACT HEXADECIMAL NUMBER FROM COMMAND LINE
; RETURN WITH VALUE IN REG A; ALL REGISTERS MAY BE AFFECTED
;
HEXNUM:
D379 CDE0D3  CALL  SCANER          ;PARSE NUMBER AND PLACE IN FCBFN
HNUM0:
D37C 219CD0  LXI    H,FCBFN      ;PT TO TOKEN FOR CONVERSION
D37F 110000  LXI    D,0                ;DE=ACCUMULATED VALUE
D382 060B    MVI    B,11          ;B=CHAR COUNT
HNUM1:
D384 7E      MOV    A,M          ;GET CHAR
D385 FE20    CPI    ' '          ;DONE?
D387        JRZ    HNUM3        ;RETURN IF SO
D389 FE48    CPI    NUMBASE      ;DONE IF NUMBASE SUFFIX
D38B        JRZ    HNUM3
D38D D630    SUI    '0'          ;CONVERT TO BINARY
D38F        JRC    NUMERR      ;RETURN AND DONE IF ERROR
D391 FE0A    CPI    10          ;0-9?
D393        JRC    HNUM2
D395 D607    SUI    7          ;A-F?
D397 FE10    CPI    10H        ;ERROR?
D399        JRNC  NUMERR
HNUM2:
D39B 23      INX    H          ;PT TO NEXT CHAR
D39C 4F      MOV    C,A        ;DIGIT IN C
D39D 7A      MOV    A,D        ;GET ACCUMULATED VALUE
D39E 07      RLC                ;EXCHANGE NYBBLES
D39F 07      RLC
D3A0 07      RLC
D3A1 07      RLC
D3A2 E6F0    ANI    0F0H        ;MASK OUT LOW NYBBLE
D3A4 57      MOV    D,A
D3A5 7B      MOV    A,E        ;SWITCH LOW-ORDER NYBBLES
D3A6 07      RLC
D3A7 07      RLC
D3A8 07      RLC
D3A9 07      RLC
D3AA 5F      MOV    E,A        ;HIGH NYBBLE OF E=NEW HIGH OF E,
; LOW NYBBLE OF E=NEW LOW OF D
D3AB E60F    ANI    0FH        ;GET NEW LOW OF D

```

```

D3AD B2      ORA   D           ;MASK IN HIGH OF D
D3AE 57      MOV   D,A         ;NEW HIGH BYTE IN D
D3AF 7B      MOV   A,E
D3B0 E6F0    ANI   0F0H        ;MASK OUT LOW OF E
D3B2 B1      ORA   C           ;MASK IN NEW LOW
D3B3 5F      MOV   E,A         ;NEW LOW BYTE IN E
D3B4         DJNZ  HNUM1       ;COUNT DOWN
;
; RETURN FROM HEXNUM
;
HNUM3:
D3B6 EB      XCHG                ;RETURNED VALUE IN HL
D3B7 7D      MOV   A,L         ;LOW-ORDER BYTE IN A
D3B8 C9      RET
;
; PT TO DIRECTORY ENTRY IN TBUFF WHOSE OFFSET IS SPECIFIED BY A AND
C
;
DIRPTR:
D3B9 218000  LXI   H,TBUFF      ;PT TO TEMP BUFFER
D3BC 81      ADD   C           ;PT TO 1ST BYTE OF DIR ENTRY
D3BD CD36D3  CALL  ADDAH        ;PT TO DESIRED BYTE IN DIR ENTRY
D3C0 7E      MOV   A,M         ;GET DESIRED BYTE
D3C1 C9      RET
;
; CHECK FOR SPECIFIED DRIVE AND LOG IT IN IF NOT DEFAULT
;
SLOGIN:
D3C2 AF      XRA   A           ;SET FCBDN FOR DEFAULT DRIVE
D3C3 329BD0  STA  FCBDN
D3C6 CDD6D3  CALL  COMLOG        ;CHECK DRIVE
D3C9 C8      RZ
D3CA         JR    DLOG5       ;DO LOGIN OTHERWISE
;
; CHECK FOR SPECIFIED DRIVE AND LOG IN DEFAULT DRIVE IF SPECIFIED<>
DEFAULT
;
DLOGIN:
D3CC CDD6D3  CALL  COMLOG        ;CHECK DRIVE
D3CF C8      RZ           ;ABORT IF SAME
D3D0 3A59D2  LDA  TDRIVE        ;LOG IN DEFAULT DRIVE
;
D3D3 C30DD2  DLOG5: JMP  LOGIN
;
; ROUTINE COMMON TO BOTH LOGIN ROUTINES; ON EXIT, Z SET MEANS ABORT
;
COMLOG:
D3D7 =      TEMPDR EQU  $+1      ;POINTER FOR IN-THE-CODE MODIFICATION
D3D6 3E00    MVI  A,0           ;2ND BYTE (IMMEDIATE ARG) IS TEMPDR
D3D8 B7      ORA  A           ;0=NO
D3D9 C8      RZ
D3DA 3D      DCR  A           ;COMPARE IT AGAINST DEFAULT
D3DB 2159D2  LXI  H,TDRIVE
D3DE BE      CMP  M
D3DF C9      RET           ;ABORT IF SAME
;
; EXTRACT TOKEN FROM COMMAND LINE AND PLACE IT INTO FCBDN;
; FORMAT FCBDN FCB IF TOKEN RESEMBLES FILE NAME AND TYPE
(FILENAME.TYP);

```



```

; ON INPUT, CIBPTR PTS TO CHAR AT WHICH TO START SCAN;
; ON OUTPUT, CIBPTR PTS TO CHAR AT WHICH TO CONTINUE AND ZERO FLAG
IS RESET
; IF '?' IS IN TOKEN
;
; ENTRY POINTS:
; SCANNER - LOAD TOKEN INTO FIRST FCB
; SCANX - LOAD TOKEN INTO FCB PTED TO BY HL
;
SCANNER:
D3E0 219BD0 LXI H,FCBDN ;POINT TO FCBDN
SCANX:
D3E3 AF XRA A ;SET TEMPORARY DRIVE NUMBER TO DEFAULT
D3E4 32D7D3 STA TEMPDR
D3E7 CD29D3 CALL ADVAN ;SKIP TO NON-BLANK OR END OF LINE
D3EA SDED CIPTR ;SET PTR TO NON-BLANK OR END OF LINE
D3EE 1A LDAX D ;END OF LINE?
D3EF B7 ORA A ;0=YES
D3F0 JRZ SCAN2
D3F2 DE40 SBI 'A'-1 ;CONVERT POSSIBLE DRIVE SPEC TO NUMBER
D3F4 47 MOV B,A ;STORE NUMBER (A:=0, B:=1, ETC) IN B
D3F5 13 INX D ;PT TO NEXT CHAR
D3F6 1A LDAX D ;SEE IF IT IS A COLON (:)
D3F7 FE3A CPI ':'
D3F9 JRZ SCAN3 ;YES, WE HAVE A DRIVE SPEC
D3FB 1B DCX D ;NO, BACK UP PTR TO FIRST NON-BLANK CHAR
SCAN2:
D3FC 3A59D2 LDA TDRIVE ;SET 1ST BYTE OF FCBDN AS DEFAULT DRIVE
D3FF 77 MOV M,A
D400 JR SCAN4
SCAN3:
D402 78 MOV A,B ;WE HAVE A DRIVE SPEC
D403 32D7D3 STA TEMPDR ;SET TEMPORARY DRIVE
D406 70 MOV M,B ;SET 1ST BYTE OF FCBDN AS SPECIFIED DRIVE
D407 13 INX D ;PT TO BYTE AFTER ':'
;
; EXTRACT FILENAME FROM POSSIBLE FILENAME.TYP
;
SCAN4:
D408 AF XRA A ;A=0
D409 32BED0 STA QMCNT ;INIT COUNT OF NUMBER OF QUESTION MARKS IN FCB
D40C 0608 MVI B,8 ;MAX OF 8 CHARS IN FILE NAME
D40E CD30D4 CALL SCANF ;FILL FCB FILE NAME
;
; EXTRACT FILE TYPE FROM POSSIBLE FILENAME.TYP
;
D411 0603 MVI B,3 ;PREPARE TO EXTRACT TYPE
D413 FE2E CPI '.' ;IF (DE) DELIMITER IS A '.', WE HAVE A TYPE
D415 JRNZ SCAN15 ;FILL FILE TYPE BYTES WITH <SP>
D417 13 INX D ;PT TO CHAR IN COMMAND LINE AFTER '.'
D418 CD30D4 CALL SCANF ;FILL FCB FILE TYPE
D41B JR SCAN16 ;SKIP TO NEXT PROCESSING
SCAN15:
D41D CD51D4 CALL SCANF4 ;SPACE FILL
;
; FILL IN EX, S1, S2, AND RC WITH ZEROES
;

```

```

SCAN16:
D420 0604      MVI   B,4           ;4 BYTES
SCAN17:
D422 23        INX   H           ;PT TO NEXT BYTE IN FCBDN
D423 3600      MVI   M,0
D425           DJNZ  SCAN17
;
; SCAN COMPLETE -- DE PTS TO DELIMITER BYTE AFTER TOKEN
;
D427          SDED  CIBPTR
;
; SET ZERO FLAG TO INDICATE PRESENCE OF '?' IN FILENAME.TYP
;
D42B 3ABED0    LDA   QMCNT        ;GET NUMBER OF QUESTION MARKS
D42E B7        ORA   A           ;SET ZERO FLAG TO INDICATE ANY '?'
D42F C9        RET
;
; SCANF -- SCAN TOKEN PTED TO BY DE FOR A MAX OF B BYTES; PLACE IT
INTO
OF
; FILE NAME FIELD PTED TO BY HL; EXPAND AND INTERPRET WILD CARDS
; '*' AND '?'; ON EXIT, DE PTS TO TERMINATING DELIMITER
;
SCANF:
D430 CD0CD3    CALL  SDELM        ;DONE IF DELIMITER ENCOUNTERED - <SP> FILL
D433           JRZ   SCANF4
D435 23        INX   H           ;PT TO NEXT BYTE IN FCBDN
D436 FE2A      CPI   '*'         ;IS (DE) A WILD CARD?
D438           JRNZ  SCANF1        ;CONTINUE IF NOT
D43A 363F      MVI   M,'?'       ;PLACE '?' IN FCBDN AND DON'T ADVANCE DE IF SO
D43C CD57D4    CALL  SCQ         ;SCANNER COUNT QUESTION MARKS
D43F           JR    SCANF2
SCANF1:
D441 77        MOV   M,A         ;STORE FILENAME CHAR IN FCBDN
D442 13        INX   D           ;PT TO NEXT CHAR IN COMMAND LINE
D443 FE3F      CPI   '?'        ;CHECK FOR QUESTION MARK (WILD)
D445 CC57D4    CZ    SCQ         ;SCANNER COUNT QUESTION MARKS
SCANF2:
D448           DJNZ  SCANF        ;DECREMENT CHAR COUNT UNTIL 8 ELAPSED
SCANF3:
D44A CD0CD3    CALL  SDELM        ;8 CHARS OR MORE - SKIP UNTIL DELIMITER
D44D C8        RZ             ;ZERO FLAG SET IF DELIMITER FOUND
D44E 13        INX   D           ;PT TO NEXT CHAR IN COMMAND LINE
D44F           JR    SCANF3
;
; FILL MEMORY POINTED TO BY HL WITH SPACES FOR B BYTES
;
SCANF4:
D451 23        INX   H           ;PT TO NEXT BYTE IN FCBDN
D452 3620      MVI   M,' '       ;FILL FILENAME PART WITH <SP>
D454           DJNZ  SCANF4
D456 C9        RET
;
; INCREMENT QUESTION MARK COUNT FOR SCANNER
; THIS ROUTINE INCREMENTS THE COUNT OF THE NUMBER OF QUESTION
MARKS IN
; THE CURRENT FCB ENTRY
;
SCQ:

```

```

D457 3ABED0    LDA    QMCNT    ;GET COUNT
D45A 3C        INR    A        ;INCREMENT
D45B 32BED0    STA    QMCNT    ;PUT COUNT
D45E C9        RET

;
; CMDTBL (COMMAND TABLE) SCANNER
; ON RETURN, HL PTS TO ADDRESS OF COMMAND IF CPR-RESIDENT
; ON RETURN, ZERO FLAG SET MEANS CPR-RESIDENT COMMAND
;
;
CMDSER:
D45F 21BFD0    LXI    H,CMDTBL    ;PT TO COMMAND TABLE
D462 0E0B      MVI    C,NCMNDS    ;SET COMMAND COUNTER

CMS1:
D464 119CD0    LXI    D,FCBFN    ;PT TO STORED COMMAND NAME
D467 0604      MVI    B,NCHARS    ;NUMBER OF CHARS/COMMAND (8 MAX)

CMS2:
D469 1A        LDAX   D        ;COMPARE AGAINST TABLE ENTRY
D46A BE        CMP    M
D46B           JRNZ   CMS3    ;NO MATCH
D46D 13        INX    D        ;PT TO NEXT CHAR
D46E 23        INX    H
D46F           DJNZ   CMS2    ;COUNT DOWN
D471 1A        LDAX   D        ;NEXT CHAR IN INPUT COMMAND MUST BE <SP>
D472 FE20      CPI    ' '
D474           JRNZ   CMS4
D476 C9        RET        ;COMMAND IS CPR-RESIDENT (ZERO FLAG SET)

CMS3:
D477 23        INX    H        ;SKIP TO NEXT COMMAND TABLE ENTRY
D478           DJNZ   CMS3

CMS4:
D47A 23        INX    H        ;SKIP ADDRESS
D47B 23        INX    H
D47C 0D        DCR    C        ;DECREMENT TABLE ENTRY NUMBER
D47D           JRNZ   CMS1
D47F 0C        INR    C        ;CLEAR ZERO FLAG
D480 C9        RET        ;COMMAND IS DISK-RESIDENT (ZERO FLAG CLEAR)

;
;**** SECTION 5 ****
; CPR-RESIDENT COMMANDS
;
;
;SECTION 5A
;COMMAND: DIR
;FUNCTION: TO DISPLAY A DIRECTORY OF THE FILES ON DISK
;FORMS:
; DIR <AFN> DISPLAYS THE DIR FILES
; DIR <AFN> S DISPLAYS THE SYS FILES
; DIR <AFN> A DISPLAY BOTH DIR AND SYS FILES
;
;
DIR:
D481 3E80      MVI    A,80H    ;SET SYSTEM BIT EXAMINATION
D483 F5        PUSH   PSW
D484 CDE0D3    CALL   SCANER    ;EXTRACT POSSIBLE D:FILENAME.TYP TOKEN
D487 CDC2D3    CALL   SLOGIN    ;LOG IN DRIVE IF NECESSARY
D48A 219CD0    LXI    H,FCBFN    ;MAKE FCB WILD (ALL '?') IF NO FILENAME.TYP
D48D 7E        MOV    A,M    ;GET FIRST CHAR OF FILENAME.TYP

```

```

D48E FE20      CPI      ' '          ;IF <SP>, ALL WILD
D490 CC2AD5    CZ        FILLQ
D493 CD29D3    CALL     ADVAN          ;LOOK AT NEXT INPUT CHAR
D496 0600      MVI      B,0          ;SYS TOKEN DEFAULT
D498           JRZ      DIR2          ;JUMP; THERE ISN'T ONE
D49A FE41      CPI      SYSFLG        ;SYSTEM FLAG SPECIFIER?
D49C           JRZ      GOTSYS        ;GOT SYSTEM SPECIFIER
D49E FE53      CPI      SOFLG        ;SYS ONLY?
D4A0           JRNZ     DIR2
D4A2 0680      MVI      B,80H        ;FLAG SYS ONLY
GOTSYS:
D4A4 13        INX      D
D4A5           SDED     CIBPTR
D4A9 FE53      CPI      SOFLG        ;SYS ONLY SPEC?
D4AB           JRZ      DIR2          ;THEN LEAVE BIT SPEC UNCHANGED
D4AD F1        POP      PSW          ;GET FLAG
D4AE AF        XRA      A           ;SET NO SYSTEM BIT EXAMINATION
D4AF F5        PUSH     PSW
DIR2:
D4B0 F1        POP      PSW          ;GET FLAG
DIR2A:
           ;DROP INTO DIRPR TO PRINT DIRECTORY
           ; THEN RESTART CPR
;
; DIRECTORY PRINT ROUTINE; ON ENTRY, MSB OF A IS 1 (80H) IF SYSTEM
FILES EXCL
;
DIRPR:
D4B1 57        MOV      D,A          ;STORE SYSTEM FLAG IN D
D4B2 1E00      MVI      E,0          ;SET COLUMN COUNTER TO ZERO
D4B4 D5        PUSH     D           ;SAVE COLUMN COUNTER (E) AND SYSTEM FLAG (D)
D4B5 78        MOV      A,B          ;SYS ONLY SPECIFIER
D4B6 32D1D4    STA      SYSTST
D4B9 CD24D2    CALL     SEARF          ;SEARCH FOR SPECIFIED FILE (FIRST OCCURRANCE)
D4BC CC9CD1    CZ        PRNNF        ;PRINT NO FILE MSG;REG A NOT CHANGED
;
; ENTRY SELECTION LOOP; ON ENTRY, A=OFFSET FROM SEARF OR SEARN
;
DIR3:
D4BF           JRZ      DIR11        ;DONE IF ZERO FLAG SET
D4C1 3D        DCR      A           ;ADJUST TO RETURNED VALUE
D4C2 0F        RRC
D4C3 0F        RRC          ;CONVERT NUMBER TO OFFSET INTO TBUFF
D4C4 0F        RRC
D4C5 E660      ANI      60H
D4C7 4F        MOV      C,A          ;OFFSET INTO TBUFF IN C (C=OFFSET TO ENTRY)
D4C8 3E0A      MVI      A,10        ;ADD 10 TO PT TO SYSTEM FILE ATTRIBUTE BIT
D4CA CDB9D3    CALL     DIRPTR
D4CD D1        POP      D           ;GET SYSTEM BIT MASK FROM D
D4CE D5        PUSH     D
D4CF A2        ANA      D           ;MASK FOR SYSTEM BIT
D4D1 =         SYSTST EQU      $+1    ;POINTER TO IN-THE-CODE BUFFER SYSTST
D4D0 FE00      CPI      0
D4D2           JRNZ     DIR10
D4D4 D1        POP      D           ;GET ENTRY COUNT (= <CR> COUNTER)
D4D5 7B        MOV      A,E          ;ADD 1 TO IT
D4D6 1C        INR      E

```

```

D4D7 D5      PUSH D          ;SAVE IT
D4D8 E603    ANI 03H        ;OUTPUT <CRLF> IF 4 ENTRIES PRINTED IN LINE
D4DA F5      PUSH PSW
D4DB        JRNZ DIR4
D4DD CDA7D1  CALL CRLF       ;NEW LINE
D4E0        JR DIR5

DIR4:
D4E2 CDEAD1  CALL PRINT
;
; IF WIDE
D4E5 2020    DB ' '          ;2 SPACES
D4E7 7C      DB FENCE        ;THEN FENCE CHAR
D4E8 20A0    DB ' ',' '+80H ;THEN 2 MORE SPACES
ENDIF
;
; IF NOT WIDE
DB ' '          ;SPACE
DB FENCE        ;THEN FENCE CHAR
DB ' '+80H      ;THEN SPACE
ENDIF
;
DIR5:
D4EA 0601    MVI B,01H      ;PT TO 1ST BYTE OF FILE NAME
DIR6:
D4EC 78      MOV A,B        ;A=OFFSET
D4ED CDB9D3  CALL DIRPTR       ;HL NOW PTS TO 1ST BYTE OF FILE NAME
D4F0 E67F    ANI 7FH        ;MASK OUT MSB
D4F2 FE20    CPI ' '        ;NO FILE NAME?
D4F4        JRNZ DIR8       ;PRINT FILE NAME IF PRESENT
D4F6 F1      POP PSW
D4F7 F5      PUSH PSW
D4F8 FE03    CPI 03H
D4FA        JRNZ DIR7
D4FC 3E09    MVI A,09H      ;PT TO 1ST BYTE OF FILE TYPE
D4FE CDB9D3  CALL DIRPTR       ;HL NOW PTS TO 1ST BYTE OF FILE TYPE
D501 E67F    ANI 7FH        ;MASK OUT MSB
D503 FE20    CPI ' '        ;NO FILE TYPE?
D505        JRZ DIR9        ;CONTINUE IF SO
DIR7:
D507 3E20    MVI A,' '      ;OUTPUT <SP>
DIR8:
D509 CDAED1  CALL CONOUT       ;PRINT CHAR
D50C 04      INR B          ;INCR CHAR COUNT
D50D 78      MOV A,B
D50E FE0C    CPI 12         ;END OF FILENAME.TYP?
D510        JRNZ DIR9       ;CONTINUE IF SO
D512 FE09    CPI 09H        ;END IF FILENAME ONLY?
D514        JRNZ DIR6       ;PRINT TYP IF SO
D516 3E2E    MVI A,'.'      ;PRINT DOT BETWEEN FILE NAME AND TYPE
D518 CDAED1  CALL CONOUT
D51B        JR DIR6
DIR9:
D51D F1      POP PSW
DIR10:
D51E CDDDED2 CALL BREAK        ;CHECK FOR ABORT
D521        JRNZ DIR11

```

```

D523 CD2BD2      CALL  SEARN      ;SEARCH FOR NEXT FILE
D526             JR      DIR3      ;CONTINUE

DIR11:
D528 D1         POP      D          ;RESTORE STACK
D529 C9         RET

;
; FILL FCB @HL WITH '?'
;
FILLQ:
D52A 060B      MVI      B,11        ;NUMBER OF CHARS IN FN & FT
FQLP:
D52C 363F      MVI      M,'?'        ;STORE '?'
D52E 23        INX      H
D52F           DJNZ     FQLP
D531 C9        RET

;
;SECTION 5B
;COMMAND: ERA
;FUNCTION:  ERASE FILES
;FORMS:
; ERA <AFN>  ERASE SPECIFIED FILES AND PRINT THEIR NAMES
;
; IF      NOT RAS          ;NOT FOR REMOTE-ACCESS SYSTEM
;
ERA:
D532 CDE0D3    CALL  SCANER      ;PARSE FILE SPECIFICATION
D535 FE0B      CPI      11          ;ALL WILD (ALL FILES = 11 '?')?
D537           JRNZ     ERA1        ;IF NOT, THEN DO ERASES
D539 CDE5D1    CALL  PRINTC
D53C 416C6CBF  DB      'all','?'+80H
D540 CDB9D1    CALL  CONIN      ;GET REPLY
D543 FE59      CPI      'Y'        ;YES?
D545 C23AD1    JNZ      RESTRT      ;RESTART CPR IF NOT
D548 CDA7D1    CALL  CRLF        ;NEW LINE

ERA1:
D54B CDC2D3    CALL  SLOGIN      ;LOG IN SELECTED DISK IF ANY
D54E AF        XRA      A          ;PRINT ALL FILES (EXAMINE SYSTEM BIT)
D54F 47        MOV      B,A        ;NO SYS-ONLY OPT TO DIRPR
D550 CDB1D4    CALL  DIRPR      ;PRINT DIRECTORY OF ERASED FILES
D553 119BD0    LXI      D,FCBDN      ;DELETE FILE SPECIFIED
D556 CD3AD2    CALL  DELETE
D559 C9        RET          ;REENTER CPR

;
; ENDIF          ;RAS
;
;SECTION 5C
;COMMAND: LIST
;FUNCTION:  PRINT OUT SPECIFIED FILE ON THE LST: DEVICE
;FORMS:
; LIST <UFN>  PRINT FILE (NO PAGING)
;
LIST:
D55A 3EFF      MVI      A,0FFH      ;TURN ON PRINTER FLAG
D55C           JR      TYPE0

;
;SECTION 5D

```

```

;COMMAND: TYPE
;FUNCTION: PRINT OUT SPECIFIED FILE ON THE CON: DEVICE
;FORMS:
; TYPE <UFN> PRINT FILE
; TYPE <UFN> P PRINT FILE WITH PAGING FLAG
;
TYPE:
D55E AF XRA A ;TURN OFF PRINTER FLAG
;
; ENTRY POINT FOR CPR LIST FUNCTION (LIST)
;
TYPE0:
D55F 32C3D1 STA PRFLG ;SET FLAG
D562 CDE0D3 CALL SCANER ;EXTRACT FILENAME.TYP TOKEN
D565 C2EFD2 JNZ ERROR ;ERROR IF ANY QUESTION MARKS
D568 CD29D3 CALL ADVAN ;GET PGDFLG IF IT'S THERE
D56B 32E8D5 STA PGFLG ;SAVE IT AS A FLAG
D56E JRZ NOSLAS ;JUMP IF INPUT ENDED
D570 13 INX D ;PUT NEW BUF POINTER
D571 EB XCHG
D572 2259D0 SHLD CIBPTR
NOSLAS:
D575 CDC2D3 CALL SLOGIN ;LOG IN SELECTED DISK IF ANY
D578 CD12D2 CALL OPENF ;OPEN SELECTED FILE
D57B CADBD5 JZ TYPE4 ;ABORT IF ERROR
D57E CDA7D1 CALL CRLF ;NEW LINE
D581 3E17 MVI A,NLINES-1 ;SET LINE COUNT
D583 32BCD0 STA PAGCNT
D586 21BDD0 LXI H,CHRCNT ;SET CHAR POSITION/COUNT
D589 36FF MVI M,0FFH ;EMPTY LINE
D58B 0600 MVI B,0 ;SET TAB CHAR COUNTER
TYPE1:
D58D 21BDD0 LXI H,CHRCNT ;PT TO CHAR POSITION/COUNT
D590 7E MOV A,M ;END OF BUFFER?
D591 FE80 CPI 80H
D593 JRC TYPE2
D595 E5 PUSH H ;READ NEXT BLOCK
D596 CDD9D1 CALL READF
D599 E1 POP H
D59A JRNZ TYPE3 ;ERROR?
D59C AF XRA A ;RESET COUNT
D59D 77 MOV M,A
TYPE2:
D59E 34 INR M ;INCREMENT CHAR COUNT
D59F 218000 LXI H,TBUFF ;PT TO BUFFER
D5A2 CD36D3 CALL ADDAH ;COMPUTE ADDRESS OF NEXT CHAR FROM OFFSET
D5A5 7E MOV A,M ;GET NEXT CHAR
D5A6 E67F ANI 7FH ;MASK OUT MSB
D5A8 FE1A CPI 1AH ;END OF FILE (^Z)?
D5AA C8 RZ ;RESTART CPR IF SO
;
; OUTPUT CHAR TO CON: OR LST: DEVICE WITH TABULATION
;
D5AB FE0D CPI CR ;RESET TAB COUNT?
D5AD JRZ TABRST
D5AF FE0A CPI LF ;RESET TAB COUNT?

```

```

D5B1          JRZ   TABRST
D5B3 FE09     CPI   TAB           ;TAB?
D5B5          JRZ   LTAB
D5B7 CDC1D1   CALL  LCOUT         ;OUTPUT CHAR
D5BA 04       INR   B             ;INCREMENT CHAR COUNT
D5BB          JR   TYPE2L

TABRST:
D5BD CDC1D1   CALL  LCOUT         ;OUTPUT <CR> OR <LF>
D5C0 0600     MVI   B,0           ;RESET TAB COUNTER
D5C2          JR   TYPE2L

LTAB:
D5C4 3E20     MVI   A,' '         ;<SP>
D5C6 CDC1D1   CALL  LCOUT
D5C9 04       INR   B             ;INCR POS COUNT
D5CA 78       MOV   A,B
D5CB E607     ANI   7
D5CD          JRNZ  LTAB

;
; CONTINUE PROCESSING
;
TYPE2L:
D5CF CDDDED2  CALL  BREAK           ;CHECK FOR ABORT
D5D2          JRZ   TYPE1         ;CONTINUE IF NO CHAR
D5D4 FE03     CPI   'C'-'@'       ;^C?
D5D6 C8       RZ                 ;RESTART IF SO
D5D7          JR   TYPE1

TYPE3:
D5D9 3D       DCR   A             ;NO ERROR?
D5DA C8       RZ                 ;RESTART CPR

TYPE4:
D5DB C378D7   JMP   ERRLOG

;
; PAGING ROUTINES
; PAGER COUNTS DOWN LINES AND PAUSES FOR INPUT (DIRECT) IF COUNT
EXPIRES

; PAGSET SETS LINES/PAGE COUNT
;
PAGER:
D5DE E5       PUSH  H
D5DF 21BCD0   LXI   H,PAGCNT       ;COUNT DOWN
D5E2 35       DCR   M
D5E3          JRNZ  PGBAK         ;JUMP IF NOT END OF PAGE
D5E5 3616     MVI   M,NLINES-2    ;REFILL COUNTER

;
D5E8 =        PGFLG EQU $+1        ;POINTER TO IN-THE-CODE BUFFER PGFLG
D5E7 3E00     MVI   A,0           ;0 MAY BE CHANGED BY PGFLG EQUATE
D5E9 FE50     CPI   PGDFLG        ;PAGE DEFAULT OVERRIDE OPTION WANTED?

;
IF PGDFLT          ;IF PAGING IS DEFAULT
D5EB          JRZ   PGBAK         ; PGDFLG MEANS NO PAGING, PLEASE
ELSE              ;IF PAGING NOT DEFAULT
D5EB          JRNZ  PGBAK         ; PGDFLG MEANS PLEASE PAGINATE
ENDIF

;
D5ED CDB9D1   CALL  CONIN          ;GET CHAR TO CONTINUE
D5F0 FE03     CPI   'C'-'@'       ;^C
D5F2 CA88D1   JZ    RSTCPR         ;RESTART CPR

```



```

PGBAK:
D5F5 E1      POP      H          ;RESTORE HL
D5F6 C9      RET

;
;SECTION 5E
;COMMAND: SAVE
;FUNCTION: TO SAVE THE CONTENTS OF THE TPA ONTO DISK AS A FILE
;FORMS:
; SAVE <NUMBER OF PAGES> <UFN>
;          SAVE SPECIFIED NUMBER OF PAGES (START AT 100H)
;          FROM TPA INTO SPECIFIED FILE; <NUMBER OF
;          PAGES> IS IN DEC
; SAVE <NUMBER OF SECTORS> <UFN> S
;          LIKE SAVE ABOVE, BUT NUMERIC ARGUMENT SPECIFIES
;          NUMBER OF SECTORS RATHER THAN PAGES
;
; IF      NOT RAS          ;NOT FOR REMOTE-ACCESS SYSTEM
;
SAVE:
D5F7 CD3BD3  CALL    NUMBER          ;EXTRACT NUMBER FROM COMMAND LINE
D5FA 6F      MOV     L,A          ;HL=PAGE COUNT
D5FB 2600    MVI     H,0
D5FD E5      PUSH   H          ;SAVE PAGE COUNT
D5FE CD4AD6  CALL    EXTEST          ;TEST FOR EXISTENCE OF FILE AND ABORT IF
SO
D601 0E16    MVI     C,16H        ;BDOS MAKE FILE
D603 CD1BD2  CALL    GRBDOS
D606 E1      POP     H          ;GET PAGE COUNT
D607         JRZ    SAVE3        ;ERROR?
D609 AF      XRA     A          ;SET RECORD COUNT FIELD OF NEW FILE'S FCB
D60A 32BBD0  STA     FCBCR
D60D CD29D3  CALL    ADVAN          ;LOOK FOR 'S' FOR SECTOR OPTION
D610 13      INX     D          ;PT TO AFTER 'S' TOKEN
D611 FE53    CPI     SECTFLG
D613         JRZ    SAVE0
D615 1B      DCX     D          ;NO 'S' TOKEN, SO BACK UP
D616 29      DAD     H          ;DOUBLE IT FOR HL=SECTOR (128 BYTES) COUNT

SAVE0:
D617         SDED   CIBPTR          ;SET PTR TO BAD TOKEN OR AFTER GOOD TOKEN
D61B 110001  LXI     D,TPA          ;PT TO START OF SAVE AREA (TPA)

SAVE1:
D61E 7C      MOV     A,H          ;DONE WITH SAVE?
D61F B5      ORA     L          ;HL=0 IF SO
D620         JRZ    SAVE2
D622 2B      DCX     H          ;COUNT DOWN ON SECTORS
D623 E5      PUSH   H          ;SAVE PTR TO BLOCK TO SAVE
D624 218000  LXI     H,128          ;128 BYTES PER SECTOR
D627 19      DAD     D          ;PT TO NEXT SECTOR
D628 E5      PUSH   H          ;SAVE ON STACK
D629 CD04D2  CALL    DMASET          ;SET DMA ADDRESS FOR WRITE (ADDRESS IN DE)
D62C 119BD0  LXI     D,FCBDN        ;WRITE SECTOR
D62F 0E15    MVI     C,15H        ;BDOS WRITE SECTOR
D631 CDD1    CALL    BDOSB          ;SAVE BC
D634 D1      POP     D          ;GET PTR TO NEXT SECTOR IN DE
D635 E1      POP     H          ;GET SECTOR COUNT
D636         JRNZ   SAVE3        ;WRITE ERROR?
D638         JR     SAVE1        ;CONTINUE

```

```

SAVE2:
D63A 119BD0      LXI   D,FCBDN      ;CLOSE SAVED FILE
D63D CD20D2      CALL  CLOSE
D640 3C          INR   A          ;ERROR?
D641             JRNZ  SAVE4

SAVE3:
D643 CDDED7      CALL  PRNLE        ;PRINT 'NO SPACE' ERROR

SAVE4:
D646 CD01D2      CALL  DEFDMA       ;SET DMA TO 0080
D649 C9          RET          ;RESTART CPR
;
; TEST FILE IN FCB FOR EXISTENCE, ASK USER TO DELETE IF SO, AND
ABORT IF HE
; CHOSSES NOT TO
;
EXTTEST:
D64A CDE0D3      CALL  SCANER       ;EXTRACT FILE NAME
D64D C2EFD2      JNZ   ERROR       ;'?' IS NOT PERMITTED
D650 CDC2D3      CALL  SLOGIN       ;LOG IN SELECTED DISK
D653 CD24D2      CALL  SEARF        ;LOOK FOR SPECIFIED FILE
D656 119BD0      LXI   D,FCBDN      ;PT TO FILE FCB
D659 C8          RZ          ;OK IF NOT FOUND
D65A D5          PUSH  D          ;SAVE PTR TO FCB
D65B CDE5D1      CALL  PRINTC
D65E 44656C6574 DB   'Delete File','?'+80H
D66A CDB9D1      CALL  CONIN        ;GET RESPONSE
D66D D1          POP   D          ;GET PTR TO FCB
D66E FE59        CPI   'Y'         ;KEY ON YES
D670 C288D1      JNZ   RSTCPR       ;RESTART IF NO
D673 D5          PUSH  D          ;SAVE PTR TO FCB
D674 CD3AD2      CALL  DELETE       ;DELETE FILE
D677 D1          POP   D          ;GET PTR TO FCB
D678 C9          RET
;
; ENDIF          ;RAS
;
;SECTION 5F
;COMMAND: REN
;FUNCTION: TO CHANGE THE NAME OF AN EXISTING FILE
;FORMS:
; REN <NEW UFN>=<OLD UFN> PERFORM FUNCTION
;
; IF NOT RAS          ;NOT FOR REMOTE-ACCESS SYSTEM
;
REN:
D679 CD4AD6      CALL  EXTEST       ;TEST FOR FILE EXISTENCE AND RETURN
; IF FILE DOESN'T EXIST; ABORT IF IT DOES
D67C 3AD7D3      LDA   TEMPDR       ;SAVE CURRENT DEFAULT DISK
D67F F5          PUSH  PSW          ;SAVE ON STACK

REN0:
D680 219BD0      LXI   H,FCBDN      ;SAVE NEW FILE NAME
D683 11ABD0      LXI   D,FCBDM
D686 011000      LXI   B,16         ;16 BYTES
D689             LDIR
D68B CD29D3      CALL  ADVAN        ;ADVANCE CIBPTR
D68E FE3D        CPI   '='         ;'=' OK
D690             JRNZ  REN4

```

```

REN1:
D692 EB      XCHG          ;PT TO CHAR AFTER '=' IN HL
D693 23      INX          H
D694 2259D0  SHLD  CIBPTR          ;SAVE PTR TO OLD FILE NAME
D697 CDE0D3  CALL  SCANNER          ;EXTRACT FILENAME.TYP TOKEN
D69A        JRNZ  REN4          ;ERROR IF ANY '?'
D69C F1      POP  PSW          ;GET OLD DEFAULT DRIVE
D69D 47      MOV  B,A          ;SAVE IT
D69E 21D7D3  LXI  H,TEMPDR      ;COMPARE IT AGAINST CURRENT DEFAULT DRIVE
D6A1 7E      MOV  A,M          ;MATCH?
D6A2 B7      ORA  A
D6A3        JRZ  REN2
D6A5 B8      CMP  B          ;CHECK FOR DRIVE ERROR
D6A6 70      MOV  M,B
D6A7        JRNZ  REN4

REN2:
D6A9 70      MOV  M,B
D6AA AF      XRA  A
D6AB 329BD0  STA  FCBDN          ;SET DEFAULT DRIVE
D6AE 119BD0  LXI  D,FCBDN        ;RENAME FILE
D6B1 0E17    MVI  C,17H          ;BDOS RENAME FCT
D6B3 CD1BD2  CALL  GRBDOS
D6B6 C0      RNZ

REN3:
D6B7 CD9CD1  CALL  PRNNF          ;PRINT NO FILE MSG

REN4:
D6BA C378D7  JMP  ERRLOG

;
;   ENDIF          ;RAS
;
;SECTION 5G
;COMMAND: USER
;FUNCTION:  CHANGE CURRENT USER NUMBER
;FORMS:
; USER <UNUM> SELECT SPECIFIED USER NUMBER;<UNUM> IS IN DEC
;
USER:
D6BD CDE9D2  CALL  USRNUM          ;EXTRACT USER NUMBER FROM COMMAND LINE
D6C0 5F      MOV  E,A          ;PLACE USER NUMBER IN E
D6C1 CD45D2  CALL  SETUSR          ;SET SPECIFIED USER

RSTJMP:
D6C4 C38BD1  JMP  RCPRNL          ;RESTART CPR
;
;SECTION 5H
;COMMAND: DFU
;FUNCTION:  SET THE DEFAULT USER NUMBER FOR THE COMMAND/FILE SCANNER
;          (MEMLOAD)
;FORMS:
; DFU <UNUM> SELECT DEFAULT USER NUMBER;<UNUM> IS IN DEC
;
DFU:
D6C7 CDE9D2  CALL  USRNUM          ;GET USER NUMBER
D6CA 32A6D7  STA  DFUSR          ;PUT IT AWAY
D6CD        JR   RSTJMP          ;RESTART CPR (NO DEFAULT LOGIN)
;
;SECTION 5I

```

```

;COMMAND: JUMP
;FUNCTION: TO CALL THE PROGRAM (SUBROUTINE) AT THE SPECIFIED
ADDRESS
; WITHOUT LOADING FROM DISK
;FORMS:
; JUMP <ADR> CALL AT <ADR>;<ADR> IS IN HEX
;
; IF NOT RAS ;NOT FOR REMOTE-ACCESS SYSTEM
;
JUMP:
D6CF CD79D3 CALL HEXNUM ;GET LOAD ADDRESS IN HL
D6D2 JR CALLPROG ;PERFORM CALL
;
; ENDF ;RAS
;
;SECTION 5J
;COMMAND: GO
;FUNCTION: TO CALL THE PROGRAM IN THE TPA WITHOUT LOADING
; LOADING FROM DISK. SAME AS JUMP 100H, BUT MUCH
; MORE CONVENIENT, ESPECIALLY WHEN USED WITH
; PARAMETERS FOR PROGRAMS LIKE STAT. ALSO CAN BE
; ALLOWED ON REMOTE-ACCESS SYSTEMS WITH NO PROBLEMS.
;
;FORM:
; GO <PARAMETERS LIKE FOR COMMAND>
;
; IF NOT RAS ;ONLY IF RAS
;
D6D4 210001 GO: LXI H,TPA ;ALWAYS TO TPA
D6D7 JR CALLPROG ;PERFORM CALL
;
; ENDF ;END OF GO FOR RAS
;
;SECTION 5K
;COMMAND: COM FILE PROCESSING
;FUNCTION: TO LOAD THE SPECIFIED COM FILE FROM DISK AND EXECUTE IT
;FORMS:
; <COMMAND>
;
COM:
D6D9 3A9CD0 LDA FCBFN ;ANY COMMAND?
D6DC FE20 CPI ' ' ;' ' MEANS COMMAND WAS 'D:' TO SWITCH
D6DE JRNZ COM1 ;NOT <SP>, SO MUST BE TRANSIENT OR ERROR
D6E0 3AD7D3 LDA TEMPDR ;LOOK FOR DRIVE SPEC
D6E3 B7 ORA A ;IF ZERO, JUST BLANK
D6E4 CA8BD1 JZ RCPRNL
D6E7 3D DCR A ;ADJUST FOR LOG IN
D6E8 3259D2 STA TDRIVE ;SET DEFAULT DRIVE
D6EB CD58D2 CALL SETU0D ;SET DRIVE WITH USER 0
D6EE CD0DD2 CALL LOGIN ;LOG IN DRIVE
D6F1 C38BD1 JMP RCPRNL ;RESTART CPR
COM1:
D6F4 3AA4D0 LDA FCBFT ;FILE TYPE MUST BE BLANK
D6F7 FE20 CPI ' '
D6F9 C2EFD2 JNZ ERROR
D6FC 2177D0 LXI H,COMMSG ;PLACE DEFAULT FILE TYPE (COM) INTO FCB
D6FF 11A4D0 LXI D,FCBFT ;COPY INTO FILE TYPE

```

```

D702 010300      LXI   B,3           ;3 BYTES
D705              LDIR
D707 210001      LXI   H,TPA         ;SET EXECUTION/LOAD ADDRESS
D70A E5          PUSH  H           ;SAVE FOR EXECUTION
D70B CD88D7      CALL  MEMLOAD        ;LOAD MEMORY WITH FILE SPECIFIED IN CMD
LINE
D70E E1          POP   H           ;GET EXECUTION ADDRESS
D70F C0          RNZ                ;RETURN (ABORT) IF LOAD ERROR
;
; CALLPROG IS THE ENTRY POINT FOR THE EXECUTION OF THE LOADED
; PROGRAM;ON ENTRY TO THIS ROUTINE, HL MUST CONTAIN THE EXECUTION
; ADDRESS OF THE PROGRAM (SUBROUTINE) TO EXECUTE
;
CALLPROG:
D710 2266D7      SHLD  EXECADR        ;PERFORM IN-LINE CODE MODIFICATION
D713 CDCCD3      CALL  DLOGIN         ;LOG IN DEFAULT DRIVE
D716 CDE0D3      CALL  SCANER          ;SEARCH COMMAND LINE FOR NEXT TOKEN
D719 21D7D3      LXI   H,TEMPDR      ;SAVE PTR TO DRIVE SPEC
D71C E5          PUSH  H
D71D 7E          MOV   A,M           ;SET DRIVE SPEC
D71E 329BD0      STA   FCBDN
D721 21ABD0      LXI   H,FCBDN+10H ;PT TO 2ND FILE NAME
D724 CDE3D3      CALL  SCANX          ;SCAN FOR IT AND LOAD IT INTO FCBDN+16
D727 E1          POP   H           ;SET UP DRIVE SPECS
D728 7E          MOV   A,M
D729 32ABD0      STA   FCBDM
D72C AF          XRA   A
D72D 32BBD0      STA   FCBCR
D730 115C00      LXI   D,TFCB          ;COPY TO DEFAULT FCB
D733 219BD0      LXI   H,FCBDN        ;FROM FCBDN
D736 012100      LXI   B,33          ;SET UP DEFAULT FCB
D739              LDIR
D73B 2108D0      LXI   H,CIBUFF
COM4:
D73E 7E          MOV   A,M           ;SKIP TO END OF 2ND FILE NAME
D73F B7          ORA   A           ;END OF LINE?
D740              JRZ   COM5
D742 FE20        CPI   ' '          ;END OF TOKEN?
D744              JRZ   COM5
D746 23          INX   H
D747              JR    COM4
;
; LOAD COMMAND LINE INTO TBUFF
;
COM5:
D749 0600        MVI   B,0           ;SET CHAR COUNT
D74B 118100      LXI   D,TBUFF+1       ;PT TO CHAR POS
COM6:
D74E 7E          MOV   A,M           ;COPY COMMAND LINE TO TBUFF
D74F 12          STAX  D
D750 B7          ORA   A           ;DONE IF ZERO
D751              JRZ   COM7
D753 04          INR   B           ;INCR CHAR COUNT
D754 23          INX   H           ;PT TO NEXT
D755 13          INX   D
D756              JR    COM6
;

```

```

; RUN LOADED TRANSIENT PROGRAM
;
COM7:
D758 78      MOV   A,B           ;SAVE CHAR COUNT
D759 328000  STA   TBUF                    ;
D75C CDA7D1  CALL  CRLF                    ;NEW LINE
D75F CD01D2  CALL  DEFDMA                   ;SET DMA TO 0080
D762 CD49D2  CALL  SETUD                    ;SET USER/DISK
;
; EXECUTION (CALL) OF PROGRAM (SUBROUTINE) OCCURS HERE
;
D766 =      EXECADR EQU   $+1           ;CHANGE ADDRESS FOR IN-LINE CODE
MODIFICATION
D765 CD0001  CALL  TPA                    ;CALL TRANSIENT
D768 CD01D2  CALL  DEFDMA                   ;SET DMA TO 0080, IN CASE
;PROG CHANGED IT ON US
D76B CD58D2  CALL  SETU0D                   ;SET USER 0/DISK
D76E CD0DD2  CALL  LOGIN                    ;LOGIN DISK
D771 C33AD1  JMP   RESTRT                   ;RESTART CPR
;
; TRANSIENT LOAD ERROR
;
COM8:
D774 E1      POP   H                    ;CLEAR RETURN ADDRESS
D775 CD3ED2  CALL  RESETUSR                ;RESET CURRENT USER NUMBER
; RESET MUST BE DONE BEFORE LOGIN
ERRLOG:
D778 CDCCD3  CALL  DLOGIN                    ;LOG IN DEFAULT DISK
ERRJMP:
D77B C3EFD2  JMP   ERROR
;
;SECTION 5L
;COMMAND: GET
;FUNCTION: TO LOAD THE SPECIFIED FILE FROM DISK TO THE SPECIFIED
ADDRESS
;FORMS:
; GET <ADR> <UFN> LOAD THE SPECIFIED FILE AT THE SPECIFIED PAGE;
; <ADR> IS IN HEX
;
; IF NOT RAS ;NOT FOR REMOTE-ACCESS SYSTEM
;
GET:
D77E CD79D3  CALL  HEXNUM                    ;GET LOAD ADDRESS IN HL
D781 E5      PUSH  H                    ;SAVE ADDRESS
D782 CDE0D3  CALL  SCANNER                    ;GET FILE NAME
D785 E1      POP   H                    ;RESTORE ADDRESS
D786        JRNZ  ERRJMP                ;MUST BE UNAMBIGUOUS
;
; FALL THRU TO MEMLOAD
;
; ENDIF ;RAS
;
; LOAD MEMORY WITH THE FILE WHOSE NAME IS SPECIFIED IN THE COMMAND
LINE
; ON INPUT, HL CONTAINS STARTING ADDRESS TO LOAD
;
MEMLOAD:
D788 CD91D7  CALL  MLOAD                    ;USER MEMORY LOAD SUBROUTINE
D78B F5      PUSH  PSW                    ;SAVE RETURN STATUS

```

```

D78C CD3ED2      CALL  RESETUSR      ;RESET USER NUMBER
D78F F1          POP    PSW          ;GET RETURN STATUS
D790 C9          RET

;
; MEMORY LOAD SUBROUTINE
; EXIT POINTS ARE A SIMPLE RETURN WITH THE ZERO FLAG SET IF NO
ERROR,
; A SIMPLE RETURN WITH THE ZERO FLAG RESET (NZ) IF MEMORY FULL, OR A
JMP TO
; COM8 IF COM FILE NOT FOUND
;
MLOAD:
D791 22C1D7      SHLD  LOADADR          ;SET LOAD ADDRESS
D794 CD43D2      CALL  GETUSR          ;GET CURRENT USER NUMBER
D797 323FD2      STA   TMPUSR          ;SAVE IT FOR LATER
D79A 32A8D7      STA   TSELUSR       ;TEMP USER TO SELECT
;
; MLA IS A REENTRY POINT FOR A NON-STANDARD CP/M MODIFICATION
; THIS IS THE RETURN POINT FOR WHEN THE .COM (OR GET) FILE IS NOT
FOUND THE
; FIRST TIME, DRIVE A: IS SELECTED FOR A SECOND ATTEMPT
;
MLA:
D79D CDC2D3      CALL  SLOGIN          ;LOG IN SPECIFIED DRIVE IF ANY
D7A0 CD12D2      CALL  OPENF          ;OPEN COMMAND.COM FILE
D7A3              JRNZ  MLA1          ;FILE FOUND - LOAD IT
;
; ERROR ROUTINE TO SELECT USER 0 IF ALL ELSE FAILS
;
D7A6 =           DFUSR  EQU   $+1          ;MARK IN-THE-CODE VARIABLE
D7A5 3E00        MVI   A,DEFUSR       ;GET DEFAULT USER
D7A8 =           TSELUSR EQU   $+1          ;MARK IN-THE-CODE VARIABLE
D7A7 FE00        CPI   DEFUSR          ;SAME?
D7A9              JRZ   MLA0          ;JUMP IF
D7AB 32A8D7      STA   TSELUSR       ;ELSE PUT DOWN NEW ONE
D7AE 5F          MOV   E,A
D7AF CD45D2      CALL  SETUSR          ;GO SET NEW USER NUMBER
D7B2              JR    MLA          ;AND TRY AGAIN
;
; ERROR ROUTINE TO SELECT DRIVE A: IF DEFAULT WAS ORIGINALLY
SELECTED
;
MLA0:
D7B4 21D7D3      LXI   H,TEMPDR       ;GET DRIVE FROM CURRENT COMMAND
D7B7 AF          XRA   A              ;A=0
D7B8 B6          ORA   M
D7B9 C274D7      JNZ   COM8          ;ERROR IF ALREADY DISK A:
D7BC 3601        MVI   M,1          ;SELECT DRIVE A:
D7BE              JR    MLA
;
; FILE FOUND -- PROCEED WITH LOAD
;
MLA1:
D7C1 =           LOADADR EQU   $+1          ;MEMORY LOAD ADDRESS (IN-LINE CODE MOD)
D7C0 210001      LXI   H,TPA          ;SET START ADDRESS OF MEMORY LOAD
ML2:
D7C3 3ECF        MVI   A,ENTRY/256-1    ;GET HIGH-ORDER ADR OF JUST BELOW CPR
D7C5 BC          CMP   H              ;ARE WE GOING TO OVERWRITE THE CPR?
D7C6              JRC   PRNLE         ;ERROR IF SO

```

```

D7C8 E5      PUSH  H           ;SAVE ADDRESS OF NEXT SECTOR
D7C9 EB      XCHG             ;... IN DE
D7CA CD04D2  CALL  DMASET        ;SET DMA ADDRESS FOR LOAD
D7CD 119BD0  LXI   D,FCBDN     ;READ NEXT SECTOR
D7D0 CDDCD1  CALL  READ
D7D3 E1      POP   H           ;GET ADDRESS OF NEXT SECTOR
D7D4         JRNZ  ML3         ;READ ERROR OR EOF?
D7D6 118000  LXI   D,128       ;MOVE 128 BYTES PER SECTOR
D7D9 19      DAD   D           ;PT TO NEXT SECTOR IN HL
D7DA         JR    ML2

;
ML3:
D7DC 3D      DCR   A           ;LOAD COMPLETE
D7DD C8      RZ              ;OK IF ZERO, ELSE FALL THRU TO PRNLE

;
; LOAD ERROR
;
PRNLE:
D7DE CDE5D1  CALL  PRINTC
D7E1 46756CEC DB   'Fu1','1'+80H
D7E5 3E01    MVI   A,1         ;SET NON-ZERO TO INDICATE ERROR
D7E7 B7      ORA   A           ;SET FLAG
D7E8 C9      RET

;
D7E9         END

```


D336 ADDAH D329 ADVAN 0000 BASE 0005 BDOS D1DE BDOSEB
D20A BDOSJP 0000 BIOSEX D2DE BREAK D2E7 BRKKBK 0050 BUFLLEN
D710 CALLPROG D2CD CB1 D2D5 CB2 D007 CBUFF D0BD CHRCNT
D059 CIBPTR D008 CIBUFF D017 CIBUF D05B CIPTR FFFF CLEVEL3
D220 CLOSE D45F CMDSER D0BF CMDTBL D464 CMS1 D469 CMS2
D477 CMS3 D47A CMS4 D2C8 CNVBUF D6F4 COM1 D73E COM4
D749 COM5 D74E COM6 D758 COM7 D774 COM8 D3D6 COMLOG
D077 COMMSG D6D9 COM D1B9 CONIN D1AE CONOUT D101 CPR1
D000 CPRLOC 003E CPRMPT 3980 CPRR D105 CPR 000D CR
D1A7 CRLF D201 DEFDMA 0000 DEFUSR D23A DELETE D6C7 DFU
D7A6 DFUSR D481 DIR D51E DIR10 D528 DIR11 D4B0 DIR2
D4B1 DIR2A D4BF DIR3 D4E2 DIR4 D4EA DIR5 D4EC DIR6
D507 DIR7 D509 DIR8 D51D DIR9 D4B1 DIRPR D3B9 DIRPTR
D3D3 DLOG5 D3CC DLOGIN D204 DMASET D000 ENTRY D532 ERA
D54B ERA1 D302 ERR1 D2F5 ERR2 D77B ERRJMP D778 ERRLOG
D2EF ERROR D766 EXECADR D64A EXTEST 0000 FALSE D0BB FCBCR
D0AB FCBDM D09B FCBDN D09C FCBFN D0A4 FCBFT 007C FENCE
D52A FILLQ D52C FQLP D77E GET D1FD GETDRV D243 GETUSR
D6D4 GO D4A4 GOTSYS D21B GRBDOS D379 HEXNUM D37C HNUM0
D384 HNUM1 D39B HNUM2 D3B6 HNUM3 D6CF JUMP D1CD LC1
D1C1 LCOUT 000A LF D55A LIST D7C1 LOADADR D20D LOGIN
D1C8 LSTOUT D5C4 LTAB 000F MAXUSR D006 MBUFF D788 MEMLOAD
D7C3 ML2 D7DC ML3 D79D MLA D7B4 MLA0 D7C0 MLA1
D791 MLOAD 003B MSIZE 0004 NCHARS 000B NCMNDS 0018 NLINES
D126 NOLOG D575 NOSLAS D351 NUM0 D357 NUM1 D374 NUM2
0048 NUMBASE D33B NUMBER D376 NUMERR D343 NUMS D34D NUMS1
D212 OPENF D219 OPEN D1B1 OUTPUT D0BC PAGCNT D5DE PAGER
D5F5 PGBAK 0050 PGDFLG FFFF PGDFLT D5E8 PGFLG D1C3 PRFLG
D1F2 PRIN1 D1E5 PRINTC D1EA PRINT D7DE PRNLE D19C PRNNF
D0BE QMCNT 0000 RAS D2B2 RB1 D18B RCPRNL D1D9 READF
D1DC READ D267 REDBUF 0000 REL D679 REN D680 REN0
D692 REN1 D6A9 REN2 D6B7 REN3 D6BA REN4 D208 RESET
D23E RESETUSR D13A RESTRT D12A RNGSUB D158 RS00 D15D RS000
D160 RS1 D188 RSTCPR D6C4 RSTJMP D5F7 SAVE D617 SAVE0
D61E SAVE1 D63A SAVE2 D643 SAVE3 D646 SAVE4 D32D SBLANK
D41D SCAN15 D420 SCAN16 D422 SCAN17 D3FC SCAN2 D402 SCAN3
D408 SCAN4 D3E0 SCANER D430 SCANF D441 SCANF1 D448 SCANF2
D44A SCANF3 D451 SCANF4 D3E3 SCANX D457 SCQ D30C SDELM
D227 SEAR1 D224 SEARF D22B SEARN 0053 SECTFLG D258 SETU0D
D249 SETUD D245 SETUSR D3C2 SLOGIN 0053 SOFLG 0024 SPRMPT
D077 STACK FFFF SUBA D07A SUBFCB D09A SUBFCR D089 SUBFCR
D088 SUBFS2 D22F SUBKIL 0000 SUPRES 0041 SYSFLG D4D1 SYSTST
0009 TAB D5BD TABRST 0080 TBUFF D259 TDRIVE D3D7 TEMPDR
005C TFCB D23F TMPUSR 0100 TPA FFFF TRUE D7A8 TSELUSR
D55E TYPE D55F TYPE0 D58D TYPE1 D59E TYPE2 D5CF TYPE2L
D5D9 TYPE3 D5DB TYPE4 D25E UCASE 0004 UDFLAG D6BD USER
D2E9 USRNUM 0000 WBOOT FFFF WIDE