

000.001

```
2  DEBUG  EQU  1          DEBUG MODE
3
4  ***   SYSTEM I/O HANDLER.
5  *
6  *   JG LETWIN, 10/77
7  *
8  *   COPYRIGHT HEATH COMPANY.
9
10
11 ***   THE SYSTEM I/O HANDLER HANDLES SYSTEM REQUESTS FOR
12 *   READS AND WRITES.
13 *
14 *   IF A MASS STORAGE DEVICE, THIS SIH DOES THE CORRECT
15 *   STORAGE MANAGEMENT. IF A SERIAL DEVICE, THE COMMAND IS PASSED
16 *   ONTO THE DEVICE DRIVER.
```

```
19  
20  
21 ** MACHINE INSTRUCTIONS  
22  
000.376 23 MI.CPI EQU 3760  
000.303 24 MI.JMP EQU 3030  
000.311 25 MI.RET EQU 3110  
26  
27  
28 ** SYSTEM SYMBOLS  
29  
000.000 30 XTEXT U8251
```

```

33X **      8251 USART BIT DEFINITIONS.
34X *
35X
36X **      PORT ADDRESSES
37X
000.000    38X UDR      EQU      0          DATA REGISTER IS EVEN
000.001    39X USR      EQU      1          STATUS REGISTER IS NEXT
40X
000.372    41X SC_UART EQU     3720         CONSOLE USART ADDRESS (IFF 8251)
42X
43X
44X **      MODE INSTRUCTION CONTROL BITS.
45X
000.100    46X UMI.1B  EQU     01000000B        1 STOP BIT
000.200    47X UMI.HB  EQU     10000000B        1 1/2 STOP BITS
000.300    48X UMI.2B  EQU     11000000B        2 STOP BITS
000.040    49X UMI.PE  EQU     00100000B        EVEN PARITY
000.020    50X UMI.PA  EQU     00010000B        USE PARITY
000.000    51X UMI.L5  EQU     00000000B        5 BIT CHARACTERS
000.004    52X UMI.L6  EQU     00000100B        6 BIT CHARACTERS
000.010    53X UMI.L7  EQU     00001000B        7 BIT CHARACTERS
000.014    54X UMI.L8  EQU     00001100B        8 BIT CHARACTERS
000.001    55X UMI.1X  EQU     00000001B        CLOCK X 1
000.002    56X UMI.16X EQU     00000010B        CLOCK X 16
000.003    57X UMI.64X EQU     00000011B        CLOCK X 64
58X
59X **      COMMAND INSTRUCTION BITS.
60X
000.100    61X UCI. IR  EQU     01000000B        INTERNAL RESET
000.040    62X UCI. RO  EQU     00100000B        READER-ON CONTROL FLAG
000.020    63X UCI. ER  EQU     00010000B        ERROR RESET
000.004    64X UCI. RE  EQU     00000100B        RECEIVE ENABLE
000.002    65X UCI. IE  EQU     00000010B        ENABLE INTERRUPTS FLAG
000.001    66X UCI. TE  EQU     00000001B        TRANSMIT ENABLE
67X
68X **      STATUS READ COMMAND BITS.
69X
000.040    70X USR.FE  EQU     00100000B        FRAMING ERROR
000.020    71X USR.OE  EQU     00010000B        OVERRUN ERROR
000.010    72X USR.PE  EQU     00001000B        PARITY ERROR
000.004    73X USR.TXE  EQU     00000100B        TRANSMITTER EMPTY
000.002    74X USR.RXR  EQU     00000010B        RECEIVER READY
000.001    75X USR.TXR  EQU     00000001B        TRANSMITTER READY
000.000    76      XTEXT  ASCII
77X
78X **      ASCII CHARACTER EQUIVALENCES.
79X
000.015    80X CR      EQU      13          CARRIAGE RETURN
000.012    81X LF      EQU      10          LINE FEED
000.200    82X NULL    EQU     2000         PAD CHARACTER
000.000    83X NUL2    EQU      0
000.007    84X BELL    EQU      7          BELL CHARACTER
000.177    85X RUBOUT  EQU     177Q         RUBOUT
000.010    86X BKSP    EQU     100          CTL-H
000.026    87X C.SYN   EQU     26Q          SYNC

```

000.002	88X C,STX	EQU	2	STX
000.047	89X QUOTE	EQU	47Q	
000.011	90X TAB	EQU	11Q	
000.033	91X ESC	EQU	33Q	
000.012	92X NL	EQU	12Q	NEW LINE (HDOS SYSTEMS)
000.212	93X ENL	EQU	NL+200Q	NL + END-OF-LINE-FLAG
000.014	94X FF	EQU	14Q	FORM FEED
000.001	95X CTLA	EQU	01Q	CTL-A
000.002	96X CTLB	EQU	02Q	CTL-B
000.003	97X CTLC	EQU	03Q	CTL-C
000.004	98X CTLD	EQU	04Q	CTL-D
000.017	99X CTLO	EQU	17Q	CTL-O
000.020	100X CTLP	EQU	20Q	CTL-P
000.021	101X CTLQ	EQU	21Q	CTL-Q
000.023	102X CTLS	EQU	23Q	CTL-S
000.032	103X CTLZ	EQU	32Q	CTL-Z
000.000	104	XTEXT	MTR	

107X ** MTR - PAM/8 EQUIVALENCES.

108X *
109X * THIS DECK CONTAINS SYMBOLIC DEFINITIONS USED TO
110X * MAKE USE OF THE PAM/8 CODE AND CONTROL BYTES.

112X ** IO PORTS

113X
000.360 114X IP.PAD EQU 3600 PAD INPUT PORT
000.360 115X OP.CTL EQU 3600 CONTROL OUTPUT PORT
000.360 116X OP.DIG EQU 3600 DIGIT SELECT OUTPUT PORT
000.361 117X OP.SEG EQU 3610 SEGMENT SELECT OUTPUT PORT

119X ** FRONT PANEL CONTROL BITS.

120X
000.020 121X CB.SSI EQU 00010000B SINGLE STEP INTERRUPT
000.040 122X CB.MTL EQU 00100000B MONITOR LIGHT
000.100 123X CB.CLI EQU 01000000B CLOCK INTERRUPT ENABLE
000.200 124X CB.SPK EQU 10000000B SPEAKER ENABLE

126X ** MONITOR MODE FLAGS.

127X
000.000 128X DM.MR EQU 0 MEMORY READ
000.001 129X DM.MW EQU 1 MEMORY WRITE
000.002 130X DM.RR EQU 2 REGISTER READ
000.003 131X DM.RW EQU 3 REGISTER WRITE

133X ** USER OPTION BITS.

134X *
135X * THESE BITS ARE SET IN CELL .MFLAG.
136X
000.200 137X UO.HLT EQU 10000000B DISABLE HALT PROCESSING
000.100 138X UO.NFR EQU CB.CLI NO REFRESH OF FRONT PANEL
000.002 139X UO.DDU EQU 00000010B DISABLE DISPLAY UPDATE
000.001 140X UO.CLK EQU 00000001B ALLOW PRIVATE INTERRUPT PROCESSING

142X ** MONITOR IDENTIFICATION FLAGS.

143X *
144X * THESE BYTES IDENTIFY THE ROM MONITOR.
145X * THEY ARE THE VARIOUS VALUES OF LOCATION .IDENT
146X
000.021 147X M.PAM8 EQU 0210 'LXI' INSTRUCTION AT 000.000 IN PAM-8
000.303 148X M.FOX EQU 3030 'JMP' INSTRUCTION AT 000.000 IN FOX ROM

150X ** ROUTINE ENTRY POINTS.

	151X *			
	152X			
000.000	153X .IDENT	EQU	0000A	IDENTIFICATION LOCATION
000.053	154X .DLY	EQU	0053A	DELAY
001.267	155X .LOAD	EQU	1267A	TAPE LOAD
001.374	156X .DUMP	EQU	1374A	TAPE DUMP
002.136	157X .ALARM	EQU	2136A	ALARM ROUTINE
002.140	158X .HORN	EQU	2140A	HORN
002.172	159X .CTC	EQU	2172A	CHECK TAPE CHECKSUM
002.205	160X .TPERR	EQU	2205A	TAPE ERROR ROUTINE
002.264	161X .PCHL	EQU	2264A	PCHL INSTRUCTION
002.265	162X .SRS	EQU	2265A	SCAN RECORD START
002.325	163X .RNP	EQU	2325A	READ NEXT PAIR
002.331	164X .RNB	EQU	2331A	READ NEXT BYTE
002.347	165X .CRC	EQU	2347A	CRC-16 CALCULATOR
003.017	166X .WNP	EQU	3017A	WRITE NEXT PAIR
003.024	167X .WNB	EQU	3024A	WRITE NEXT BYTE
003.122	168X .DOD	EQU	3122A	DECODE FOR OCTAL DISPLAY
003.260	169X .RCK	EQU	3260A	READ CONSOLE KEYS
003.356	170X .DODA	EQU	3356A	SEGMENT CODE TABLE

172X ** RAM CELLS USED BY H8MTR.

	173X *			
	174X			
040.000	175X .START	EQU	40000A	START DUMP ADDRESS
040.002	176X .IDWRK	EQU	40002A	IN OR OUT INSTRUCTION
040.005	177X .REGI	EQU	40005A	DISPLAYED REGISTER INDEX
040.006	178X .DSPROT	EQU	40006A	PERIOD FLAG BYTE
040.007	179X .DSPMOD	EQU	40007A	DISPLAY MODE
040.010	180X .MFLAG	EQU	40010A	USER OPTION BYTE
040.011	181X .CTLFLG	EQU	40011A	PANEL CONTROL BYTE
040.013	182X .ALEDS	EQU	40013A	ABUSS LEDES
040.021	183X .DLEDS	EQU	40021A	DRUSS LEDES
040.024	184X .ABUSS	EQU	40024A	ABUSS REGISTER
040.027	185X .CRCSUM	EQU	40027A	CRCSUM WORD
040.031	186X .TPERRX	EQU	40031A	TAPE ERROR EXIT VECTOR
040.033	187X .TICCNT	EQU	40033A	CLOCK TICK COUNTER
040.035	188X .REGPTR	EQU	40035A	REGISTER POINTER
040.037	189X .UIVEC	EQU	40037A	USER INTERRUPT VECTORS
000.000	190	XTEXT	BOODEF	

192X ** BOODEF - SPECIAL BOOT-HDOS INTERFACE DEFINITIONS.

	193X			
047.000	194X SB.ORG	EQU	47000A	ORG FOR LOAD OF INITIAL HDOS.SAV
014.000	195X SB.OVMX	EQU	14000A	SIZE OF HOLD AREA FOR SWAPPED USER CODE
	196X *			(=MAX SIZE OF HDOSOVL.SYS)
000.000	197	XTEXT	HDSROM	

```

199X **      HDOS M17 ROM ENTRY POINTS.
031.253      200X      ORG      31253A
031.253      201X DWRITE EQU      *
031.253      202X      DS      31256A-31253A
031.256      203X DREAD  EQU      *
031.256      204X      DS      31275A-31256A
031.275      205X S.READ  EQU      *
031.275      206X      DS      31321A-31266A
031.330      207X S.WRITE EQU      *
031.330      208X      DS      31325A-31311A
031.344      209X ERR.FND EQU      *
031.344      210X      DS      31331A-31325A
031.350      211X ERR.ILR EQU      *
031.350      212X      DS      31335A-31331A
031.354      213X CFF    EQU      *
031.354      214X      DS      31363A-31335A
032.002      215X DCA    EQU      *
032.002      216X      DS      32114A-31363A
032.133      217X FFB    EQU      *
032.133      218X      DS      32166A-32114A
032.205      219X FFL    EQU      *
032.205      220X      DS      32204A-32166A
032.223      221X *LDD   EQU      *
032.223      222X      DS      32372A-32204A+1
033.012      223X LDD   EQU      *
033.012      224X      DS      33135A-33002A
033.145      225X PDI    EQU      *
033.145      226X      DS      33154A-33124A
033.175      227X REL.   EQU      *
033.175      228X      DS      33156A-33154A
033.177      229X REL    EQU      *
033.177      230X      DS      33212A-33156A
033.233      231X TFE    EQU      *
033.233      232X      DS      33232A-33206A
033.257      233X RUC    EQU      *
033.257      234      XTEXT  FILDEF

```

```

236X **      FILDEF - FILE TYPE DEFINITIONS.
237X *
238X *      DB      377Q,FT,XXX
239X
240X
000.000      241X FT.ABS EQU      0      ABSOLUTE BINARY
000.001      242X FT.FIC EQU      1      POSITION INDEPENDANT CODE
000.002      243X FT.REL EQU      2      RELOCATABLE CODE
000.003      244X FT.BAC EQU      3      COMPILED BASIC CODE
033.257      245      XTEXT  MOSDEF

```

```

247X **      HOSDEF - DEFINE HDOS PARAMETER.
248X *
249X
250X
000.026     251X .VERS  ERU    1*12+6      VERSION 1.6
252X
000.377     253X .SYSCALL ERU    3770        SYSCALL INSTRUCTION
254X
000.000     255X
256X          ORG     0
257X
258X *      RESIDENT FUNCTIONS
259X
000.000     260X .EXIT  DS     1      EXIT (MUST BE FIRST)
000.001     261X .SCIN  DS     1      SCIN
000.002     262X .SCOUT DS     1      SCOUT
000.003     263X .PRINT DS     1      PRINT
000.004     264X .READ  DS     1      READ
000.005     265X .WRITE DS     1      WRITE
000.006     266X .CONSL DS     1      SET/CLEAR CONSOLE OPTIONS
000.007     267X .CLRCD  DS     1      CLEAR CONSOLE BUFFER
000.010     268X .LDADD  DS     1      LOAD AN OVERLAY
000.011     269X .VERS  DS     1      RETURN HDOS VERSION NUMBER
000.012     270X .SYSRES DS     1      PRECEDING FUNCTIONS ARE RESIDENT
271X
272X
273X *      *HDOSVOL0.SYS* FUNCTIONS
274X
000.040     275X          ORG     40A
276X
000.040     277X .LINK  DS     1      LINK (MUST BE FIRST)
000.041     278X .CTLCD  DS     1      CTL-C
000.042     279X .OPENR  DS     1      OPENR
000.043     280X .OPENW  DS     1      OPENW
000.044     281X .OPENU  DS     1      OPENU
000.045     282X .OPENC  DS     1      OPENC
000.046     283X .CLOSE  DS     1      CLOSE
000.047     284X .POSIT  DS     1      POSITION
000.050     285X .DELET  DS     1      DELETE
000.051     286X .RENAM  DS     1      RENAME
000.052     287X .SETTP  DS     1      SETTOP
000.053     288X .DECODE DS     1      NAME DECODE
000.054     289X .NAME   DS     1      GET FILE NAME FROM CHANNEL
000.055     290X .CLEAR  DS     1      CLEAR CHAN
000.056     291X .CLEARA DS     1      CLEAR ALL CHANS
000.057     292X .ERROR  DS     1      LOOKUP ERROR
000.060     293X .CHFLG  DS     1      CHANGE FLAGS
000.061     294X .DISMT  DS     1      FLAG SYSTEM DISK DISMOUNTED
000.062     295X .LOADD  DS     1      LOAD DEVICE DRIVER
296X
297X
298X *      *HDOSVOL1.SYS* FUNCTIONS
299X
000.200     300X          ORG     2000
301X
000.200     302X .MOUNT  DS     1      MOUNT (MUST BE FIRST)

```


000.201	303X	.DMOUN	DS	1	DISMOUNT
000.202	304X	.MONMS	DS	1	MOUNT/NO MESSAGE
000.203	305X	.DMNMS	DS	1	DISMOUNT/NO MESSAGE
000.204	306X	.RESET	DS	1	RESET = DISMOUNT/MOUNT OF UNIT
000.205	307		XTEXT	DEVDEF	

309X ** DEVICE TABLE ENTRYS.

	310X				
000.000	311X		ORG	0	
	312X				
000.000	313X	DEV.NAM	DS	2	DEVICE NAME
000.000	314X	DEV.EL	EQU	00000000B	END OF DEVICE LIST FLAG
000.001	315X	DEV.NU	EQU	00000001B	DEVICE ENTRY NOT IN USE
	316X				
000.002	317X	DEV.RES	DS	1	DRIVER RESIDENCE CODE
000.001	318X	DR.IM	EQU	00000001B	DRIVER IN MEMORY
000.002	319X	DR.PR	EQU	00000010B	DRIVER PERMINANTLY RESIDENT
	320X				
000.003	321X	DEV.JMP	DS	1	JMP TO PROCESSOR
000.004	322X	DEV.DDA	DS	2	DRIVER ADDRESS
000.006	323X	DEV.FLG	DS	1	FLAG BYTE
000.001	324X	DT.DD	EQU	00000001B	DIRECTORY DEVICE
000.002	325X	DT.CR	EQU	00000010B	CAPABLE OF READ OPERATION
000.004	326X	DT.CW	EQU	00000100B	CAPABLE OF WRITE OPERATION
	327X				
000.007	328X	DEV.SPG	DS	1	SECTORS PER GROUP THIS DEVICE
000.010	329X	DEV.MUM	DS	1	MOUNTED UNIT MASK
000.011	330X	DEV.MNU	DS	1	MAXIMUM NUMBER OF UNITS
000.012	331X	DEV.UNT	DS	2	ADDRESS OF UNIT SPECIFIC DATA TABLE
	332X				
000.014	333X	DEV.DVL	DS	2	DRIVER BYTE LENGTH
000.016	334X	DEV.DVG	DS	1	DRIVER ROUTINE GROUP ADDRESS
	335X				
000.017	336X	DEVELEN	EQU	*	DEVICE TABLE ENTRY LENGTH

338X ** UNIT SPECIFIC DEVICE DATA TABLE ENTRIES

	339X				
000.000	340X		ORG	0	
	341X				
000.000	342X	UNT.FLG	DS	1	UNIT SPECIFIC #DEV.FLG*
000.001	343X	UNT.GRT	DS	2	ADDRESS OF GROUP RESERVATION TABLE (IF DT.DD)
000.003	344X	UNT.GTS	DS	2	GRT SECTOR NUMBER
000.005	345X	UNT.FIS	DS	2	DIRECTORY FIRST SECTOR NUMBER
	346X				
000.007	347X	UNT.SIZ	EQU	*	SIZE OF UNIT SPECIFIC DATA TABLE PER UNIT
000.007	348		XTEXT	DIRDEF	

```

350X **    DIRECTORY ENTRY FORMAT.
351X
000.000   352X    ORG    0
353X
354X
000.377   355X DF.EMP  EQU    3770    FLAGS ENTRY EMPTY
000.376   356X DF.CLR  EQU    3760    FLAGS ENTRY EMPTY; REST OF DIR ALSO CLEAR
357X
000.000   358X DIR.NAM  DS      8        NAME
000.010   359X DIR.EXT  DS      3        EXTENSION
000.013   360X DIR.PRO  DS      1        PROJECT
000.014   361X DIR.VER  DS      1        VERSION
000.015   362X DIR.IDL  EQU    *        FILE IDENTIFICATION LENGTH
363X
000.015   364X DIR.CLU  DS      1        CLUSTER FACTOR
000.016   365X DIR.FLG  DS      1        FLAGS
000.017   366X      DS      1        RESERVED
000.020   367X DIR.FGN  DS      1        FIRST GROUP NUMBER
000.021   368X DIR.LGN  DS      1        LAST GROUP NUMBER
000.022   369X DIR.LSI  DS      1        LAST SECTOR INDEX (IN LAST GROUP)
000.023   370X DIR.CRD  DS      2        CREATION DATE
000.025   371X DIR.ALD  DS      2        LAST ALTERATION DATE
372X
000.027   373X DIRELEN EQU    *        DIRECTORY ENTRY LENGTH
000.027   374      XTEXT  DISDEF
    
```

```

376X **    DIRECTORY BLOCK FORMAT.
377X
000.000   378X    ORG    0
379X
000.000   380X DIS.ENT  EQU    *        FIRST ENTRY ADDRESS
000.000   381X      DS    22*DIRELEN    22 DIRECTORY ENTRIES PER BLOCK
001.372   382X      DS      1        0 BYTE = END OF ENTRIES IN THIS BLOCK
383X
001.373   384X    ORG    512-5        AT END OF BLOCK
001.373   385X DIS.ENL  DS      1        LENGTH OF EACH ENTRY (=DIRELEN)
001.374   386X DIS.SEC  DS      2        BLOCK # OF THIS BLOCK;
001.376   387X DIS.LNK  DS      2        BLOCK # OF NEXT BLOCK, =0 IF THIS IS LAST
002.000   388      XTEXT  IOCDEF
    
```

```

390X **    I/O CHANNEL DEFINITIONS.
391X
000.000   392X    ORG    0
393X
000.000   394X IOC.LNK  DS      2        ADDRESS OF NEXT CHANNEL; =0 IF LAST
000.002   395X IOC.DDA  DS      2        THREAD JUMP TO DEVICE DRIVER (VIA DEV TABLE)
396X
000.004   397X IOC.FLG  DS      1        FILE TYPE FLAGS
000.001   398X FT.BD   EQU    00000001B    =1 IF DIRECTORY DEVICE
000.002   399X FT.OR   EQU    00000010B    =1 IF OPEN FOR READ
000.004   400X FT.OW   EQU    00000100B    =1 IF OPEN FOR WRITE
000.010   401X FT.OU   EQU    00001000B    =1 IF OPEN FOR UPDATE
    
```

IOC

000.003	402X	IOC.SQL	ERU	*-IOC.DDA	LENGTH OF INFO FOR SEQUENTIAL FILE (FROM IOC)
	403X				
000.005	404X	IOC.BRT	DS	2	ADDRESS OF GROUP RESERVATION TABLE
000.007	405X	IOC.SPG	DS	1	SECTORS PER GROUP, THIS DEVICE
000.010	406X	IOC.CGN	DS	1	CURRENT GROUP NUMBER
000.011	407X	IOC.CSI	DS	1	CURRENT SECTOR INDEX (IN CURRENT GROUP)
000.012	408X	IOC.LGN	DS	1	LAST GROUP NUMBER
000.013	409X	IOC.LSI	DS	1	LAST SECTOR INDEX (IN LAST GROUP)
000.010	410X	IOC.BRL	ERU	*-IOC.FLG	LENGTH OF INFO NORMALLY COPIED BACK TO THE CHANNEL TABLE
	411X	*			
000.014	412X	IOC.DTA	DS	2	DEVICE TABLE ADDRESS FOR THIS DEVICE
000.016	413X	IOC.DES	DS	2	SECTOR NUMBER OF DIRECTORY ENTRY
000.020	414X	IOC.DEV	DS	2	DEVICE CODE
000.022	415X	IOC.UNI	DS	1	UNIT NUMBER (0-9)
000.021	416X	IOC.DIL	ERU	*-IOC.DDA	LENGTH OF INFO FOR DIRECTORY FILE (FROM IOC)
	417X				
000.023	418X	IOC.DIR	DS	DIRELEN	DIRECTORY ENTRY
	419X				
000.052	420X	IOCELEN	ERU	*	IOC ENTRY LENGTH
	421X				
000.001	422X	IOCCTD	ERU	1	INDEX OF USER CHANNEL #0 IN CHANTAB (FIRST = 0)
000.052	423	XTEXT	DDDEF		

425X ** DEVICE DRIVER COMMUNICATION FLAGS:

	426X	*			
	427X				
000.000	428X		ORG	0	
	429X				
000.000	430X	DC.REA	DS	1	READ
000.001	431X	DC.WRI	DS	1	WRITE
000.002	432X	DC.RER	DS	1	READ REGARDLESS
000.003	433X	DC.OPR	DS	1	OPEN FOR READ
000.004	434X	DC.OPW	DS	1	OPEN FOR WRITE
000.005	435X	DC.OPU	DS	1	OPEN FOR UPDATE
000.006	436X	DC.CLO	DS	1	CLOSE
000.007	437X	DC.ABT	DS	1	ABORT
000.010	438X	DC.MOU	DS	1	MOUNT DEVICE
000.011	439X	DC.LOD	DS	1	LOAD DEVICE DRIVER
000.012	440X	DC.MAX	DS	1	MAXIMUM ENTRY INDEX
000.013	441	XTEXT	ECDEF		

443X ** ERROR CODE DEFINITIONS:

	444X				
000.000	445X		ORG	0	
000.000	446X		DS	1	NO ERROR #0
000.001	447X	EC.EOF	DS	1	END OF FILE
000.002	448X	EC.EOM	DS	1	END OF MEDIA
000.003	449X	EC.ILC	DS	1	ILLEGAL SYSCALL CODE
000.004	450X	EC.CNA	DS	1	CHANNEL NOT AVAILABLE
000.005	451X	EC.DNS	DS	1	DEVICE NOT SUITABLE
000.006	452X	EC.IDN	DS	1	ILLEGAL DEVICE NAME
000.007	453X	EC.IFN	DS	1	ILLEGAL FILE NAME

000.010	454X	EC.NRD	DS	1	NO ROOM FOR DEVICE DRIVER
000.011	455X	EC.FNO	DS	1	CHANNEL NOT OPEN
000.012	456X	EC.YLR	DS	1	ILLEGAL REQUEST
000.013	457X	EC.FUC	DS	1	FILE USAGE CONFLICT
000.014	458X	EC.FNF	DS	1	FILE NAME NOT FOUND
000.015	459X	EC.UND	DS	1	UNKNOWN DEVICE
000.016	460X	EC.ICN	DS	1	ILLEGAL CHANNEL NUMBER
000.017	461X	EC.DIF	DS	1	DIRECTORY FULL
000.020	462X	EC.IFC	DS	1	ILLEGAL FILE CONTENTS
000.021	463X	EC.NEM	DS	1	NOT ENOUGH MEMORY
000.022	464X	EC.RF	DS	1	READ FAILURE
000.023	465X	EC.WF	DS	1	WRITE FAILURE
000.024	466X	EC.WPV	DS	1	WRITE PROTECTION VIOLATION
000.025	467X	EC.WP	DS	1	DISK WRITE PROTECTED
000.026	468X	EC.FAP	DS	1	FILE ALREADY PRESENT
000.027	469X	EC.DBA	DS	1	DEVICE DRIVER ABORT
000.030	470X	EC.FL	DS	1	FILE LOCKED
000.031	471X	EC.FAO	DS	1	FILE ALREADY OPEN
000.032	472X	EC.IS	DS	1	ILLEGAL SWITCH
000.033	473X	EC.UUN	DS	1	UNKNOWN UNIT NUMBER
000.034	474X	EC.FNR	DS	1	FILE NAME REQUIRED
000.035	475X	EC.DIW	DS	1	DEVICE IS NOT WRITABLE (OR WRITE LOCKED)
000.036	476X	EC.UNA	DS	1	UNIT NOT AVAILABLE
000.037	477X	EC.ILV	DS	1	ILLEGAL VALUE
000.040	478X	EC.ILO	DS	1	ILLEGAL OPTION
000.041	479X	EC.VPM	DS	1	VOLUME PRESENTLY MOUNTED ON DEVICE
000.042	480X	EC.NVN	DS	1	NO VOLUME PRESENTLY MOUNTED
000.043	481X	EC.FOD	DS	1	FILE OPEN ON DEVICE
000.044	482X	EC.NFM	DS	1	NO PROVISIONS MADE FOR REMOUNTING MORE DISKS
000.045	483X	EC.DNI	DS	1	DISK NOT INITIALIZED
000.046	484X	EC.DNR	DS	1	DISK IS NOT READABLE
000.047	485X	EC.DSC	DS	1	DISK STRUCTURE IS CORRUPT
000.050	486X	EC.NCV	DS	1	NOT CORRECT VERSION OF HDOS
000.051	487X	EC.NOS	DS	1	NO OPERATING SYSTEM MOUNTED
000.052	488X	EC.IOI	DS	1	ILLEGAL OVERLAY INDEX
000.053	489X	EC.OTL	DS	1	OVERLAY TOO LARGE
000.054	490		XTEXT	DDFDEF	

492X ** DIRECTORY DEVICE FORMAT DEFINITION.

493X *

494X

495X

000.002 496X HOS.SPG EQU 2 2 SECTORS PER GROUP REQUIRED FOR NOW

497X

000.000 498X ORG 0

000.000 499X DDF.B00 DS 9 2K ROOT PROGRAM

000.011 500X DDF.B0L EQU * LENGTH OF ROOT

000.011 501X DDF.LAB DS 1 LABEL SECTOR

000.012 502X DDF.RGT DS 2 RESERVED GROUP TABLE

000.014 503X DDF.USR DS 0 BEGINNING OF OPEN SPACE

000.014 504 XTEXT LABDEF

506X ** DISK LABEL SECTOR FORMATS.

000.000	507X				
000.000	508X	ORG	0		
000.001	509X	LAB.SER DS	1	SERIAL NUMBER OF VOLUME	
000.003	510X	LAB.IND DS	2	INITIALIZATION DATE	
000.005	511X	LAB.DIS DS	2	SECTOR NUMBER OF 1ST DIRECTORY SECTOR	
000.007	512X	LAB.GRT DS	2	INDEX OF GRT SECTOR	
	513X	LAB.SPG DS	1	SECTORS PER GROUP	
	514X				
000.000	515X	LAB.DAT EQU	0	DATA VOLUME ONLY	
000.001	516X	LAB.SYS EQU	1	SYSTEM VOLUME	
000.002	517X	LAB.NOD EQU	2	=> LAB.NOD MEANS VOLUME HAS NO DIRECTORY	
	518X				
000.010	519X	LAB.VLT DS	1	VOLUME TYPE	
000.011	520X	LAB.VER DS	1	VERSION OF INIT17 THAT INITED DISK	
000.012	521X	DS	7	UNUSED	
000.021	522X	LAB.LAB DS	60	LABEL	
000.074	523X	LAB.LBL EQU	*-LAB.LAB	LABEL LENGTH	
000.115	524	XTEXT	ABSDEF		

526X ** ABS FORMAT EQUIVALENCES.

000.000	527X				
	528X	ORG	0		
	529X				
000.000	530X	ABS.ID DS	1	377Q = BINARY FILE FLAG	
000.001	531X	DS	1	FILE TYPE (FT.ABS)	
000.002	532X	ABS.LDA DS	2	LOAD ADDRESS	
000.004	533X	ABS.LEN DS	2	LENGTH OF ENTIRE RECORD	
000.006	534X	ABS.ENT DS	2	ENTRY POINT	
	535X				
000.010	536X	ABS.COD DS	0	CODE STARTS HERE	
000.010	537	XTEXT	PICDEF		

539X ** PIC FORMAT EQUIVALENCES.

000.000	540X				
	541X	ORG	0		
	542X				
000.000	543X	PIC.ID DS	1	377Q = BINARY FILE FLAG	
000.001	544X	DS	1	FILE TYPE (FT.PIC)	
000.002	545X	PIC.LEN DS	2	LENGTH OF ENTIRE RECORD	
000.004	546X	PIC.PTR DS	2	INDEX OF START OF PIC TABLE	
	547X				
000.006	548X	PIC.COD DS	0	CODE STARTS HERE	
000.006	549	XTEXT	DIFDEF		

551X ** DIRECTORY FILE FLAGS.
552X
000.200 553X DIF.SYS EQU 1000000B SYSTEM FILE
000.100 554X DIF.LOC EQU 0100000B LOCKED FOR CHANGE
000.040 555X DIF.WP EQU 0010000B WRITE PROTECTED
000.020 556X DIF.CNT EQU 0001000B CONTIGUOUS FILE
557X
000.006 558 XTEXT NAMDEF

560X ** SYSTEM FILE NAME CONVENTIONS
561X *
562X * RGT .SYS RESERVED GROUP TABLE (1 SECTOR)
563X * GRT .SYS GROUP RESERVATION TABLE (1 SECTOR)
564X * DIRECT .SYS DIRECTORY
565X * HOS .SYS SYSTEM IMAGE PROGRAM FOR SYSTEM
566X
000.006 567 XTEXT MTRDEF

569X ** HDOS MONITOR PRIVATE RAM AREA DEFINITIONS.
570X
000.000 571X ORG 0
000.000 572X M.SYSM DS 1 SYSCALL ITERATION COUNT
000.001 573X M.SALD DS 1 STAND-ALONE FLAG
000.002 574X M.CSLC DS 1 LINES IN CONSOLE BUFFER
000.003 575X M.CPRE DS 1 CONSOLE PREVIOUS CHARACTER
000.004 576X M.CRUB DS 1 CONSOLE RUBOUT FLAG
000.005 577X M.CINT DS 1 CONSOLE INTERRUPT FLAG
000.006 578X M.CIN DS 2 CONSOLE CB IN POINTER
000.010 579X M.COUT DS 2 CONSOLE CB OUT POINTER
000.012 580X M.CFWA DS 2 CONSOLE CB FWA POINTER
000.014 581X M.CLWA DS 2 CONSOLE CB LWA POINTER
000.016 582X M.CDLY DS 1 CONSOLE PAD CHARACTER COUNT
000.017 583X M.CDCA DS 2 ADDRESS OF CHARACTER BEING PADDED

000.021

586

XTEXT HOSEQU

588X ** HDOS SYSTEM EQUIVALENCES.

589X *

590X

024.000 591X S.BRT0 EQU 24000A SYSTEM AREA FOR GRT0
025.000 592X S.BRT1 EQU 25000A SYSTEM AREA FOR GRT1
026.000 593X S.BRT2 EQU 26000A SYSTEM AREA FOR GRT2

594X

030.000 595X ROMBOOT EQU 30000A ROM BOOT ENTRY

596X

040.100 597X ORG 40100A FREE SPACE FROM PAM-8

598X

040.100 599X DS 8 JUMP TO SYSTEM EXIT

040.110 600X D.CON DS 16 DISK CONSTANTS

040.130 601X SYDD EQU * SYSTEM DISK ENTRY POINT

040.130 602X D.VEC DS 24*3 SYSTEM ROM ENTRY VECTORS

040.240 603X D.RAM DS 31 SYSTEM ROM WORK AREA

040.277 604X S.VAL DS 36 SYSTEM VALUES

040.343 605X S.INT DS 115 SYSTEM INTERNAL WORK AREAS

041.126 606X DS 16

041.146 607X S.SOVR DS 2 STACK OVERFLOW WARNING

041.150 608X DS 42200A-* SYSTEM STACK

001.032 609X STACKL EQU *-S.SOVR STACK SIZE

610X

042.200 611X STACK EQU * LWA+1 SYSTEM STACK

042.200 612X USERFWA EQU * USER FWA

613

042.200 614 XTEXT ESVAL

616X ** S.VAL - SYSTEM VALUE DEFINITIONS.

617X *

618X *

THESE VALUES ARE SET AND MAINTAINED BY THE SYSTEM.

619X *

620X *

THE DECK HOSEQU MUST BE MODIFIED WHEN THIS IS MODIFIED.

621X

622X

040.277 623X ORG S.VAL

624X

040.277 625X S.DATE DS 9 SYSTEM DATE (IN ASCII)

040.310 626X S.DATC DS 2 CODED DATE

040.312 627X S.TIME DS 4 TIME FROM MIDNIGHT (IN TICS)

040.316 628X S.HIMEM DS 2 HARDWARE HIGH MEMORY ADDRESS+1

629X

040.320 630X S.SYSH DS 2 FWA RESIDENT SYSTEM

631X

040.322 632X S.USRM DS 2 LWA USER MEMORY

633X

040.324 634X S.OMAX DS 2 MAX OVERLAY SIZE FOR SYSTEM

635X

```

636X
637X **      THE FOLLOWING FIVE CELLS SHOULD BE MODIFIED/READ ONLY VIA THE .CONSL SYSCALL
638X
000.200     639X CSL.ECH EQU      10000000B      SUPPRESS ECHO
000.002     640X CSL.WRP EQU      00000010B      WRAP LINES AT WIDTH
000.001     641X CSL.CHR EQU      00000001B      OPERATE IN CHARACTER MODE
642X
000.000     643X I.CSLMD EQU      0              S.CSLMD IS FIRST BYTE
040.326     644X S.CSLMD DS      1              CONSOLE MODE
645X
000.200     646X CTP.BKS EQU      10000000B      TERMINAL PROCESSES BACKSPACES
000.040     647X CTP.MLI EQU      00100000B      MAP LOWER CASE TO UPPER ON INPUT
000.020     648X CTP.MLO EQU      00010000B      MAP LOWER CASE TO UPPER ON OUTPUT
000.010     649X CTP.2SB EQU      00001000B      TERMINAL NEEDS TWO STOP BITS
000.002     650X CTP.BKM EQU      00000010B      MAP BKSP (UPON INPUT) TO RUBOUT
000.001     651X CTP.TAB EQU      00000001B      TERMINAL SUPPORTS TAB CHARACTERS
652X
000.001     653X I.CONTY EQU      1              S.CONTY IS 2ND BYTE
000.000     654X          ERRNZ *-S.CSLMD-I.CONTY
040.327     655X S.CONTY DS      1              CONSOLE TYPE FLAGS
000.002     656X I.CUSOR EQU      2              S.CUSOR IS 3RD BYTE
000.000     657X          ERRNZ *-S.CSLMD-I.CUSOR
040.330     658X S.CUSOR DS      1              CURRENT CURSOR POSITION
000.003     659X I.CONWI EQU      3              S.CONWI IS 4TH BYTE
000.000     660X          ERRNZ *-S.CSLMD-I.CONWI
040.331     661X S.CONWI DS      1              CONSOLE WIDTH
662X
000.001     663X CO.FLG EQU      00000001B      CTL-O FLAG
000.200     664X CS.FLG EQU      10000000B      CTL-S FLAG
665X
000.004     666X I.CONFL EQU      4              S.CONFL IS 5TH BYTE
000.000     667X          ERRNZ *-S.CSLMD-I.CONFL
040.332     668X S.CONFL DS      1              CONSOLE FLAGS
669X
040.333     670X S.CAADR DS      2              ADDRESS FOR ABORT PROCESSING (>256 IF VALID)
040.335     671X S.CCTAB DS      6              ADDR FOR CTL-A, CTL-B, CTL-C PROCESSING
672
040.343     673          XTEXT      ESINT

675X **      S.INT - SYSTEM INTERNAL WORKAREA DEFINITIONS.
676X *
677X *      THESE CELLS ARE REFERENCED BY OVERLAYS AND MAIN CODE, AND
678X *      MUST THEREFORE RESIDE IN FIXED LOW MEMORY.
679X
040.343     680X
681X          ORG      S.INT
682X
683X **      CONSOLE STATUS FLAGS
684X
040.343     685X S.CDB  DS      1              CONSOLE DESCRIPTOR BYTE
000.000     686X CDB.H85 EQU      00000000B
000.001     687X CDB.H84 EQU      00000001B      =0 IF H8-5, *1 IF H8-4
040.344     688X S.BAUD DS      2              [0-14] H8-4 BAUD RATE, =0 IF H8-5

```


Address	Label	Type	Value	Description
689X *			[15]	=1 IF BAUD RATE => 2 STOP BITS
690X				
691X **	TABLE ADDRESS WORDS			
692X				
040.346	693X S.DLINK	DS	2	ADDRESS OF DATA IN HDOS CODE
040.350	694X S.OFWA	DS	2	FWA OVERLAY TABLE
040.352	695X S.CFWA	DS	2	FWA CHANNEL TABLE
040.354	696X S.DFWA	DS	2	FWA DEVICE TABLE
040.356	697X S.RFWA	DS	2	FWA RESIDENT HDOS CODE
698X				
699X **	DEVICE DRIVER DELAYED LOAD FLAGS			
700X				
040.360	701X S.DDLDA	DS	2	DRIVER LOAD ADDRESS (HIGH BYTE=0 IF NO LOAD PENDING)
040.362	702X S.DDLEN	DS	2	CODE LENGTH IN BYTES
040.364	703X S.DDGRP	DS	1	GROUP NUMBER FOR DRIVER
040.365	704X	DS	1	HOLD PLACE
	705X *S.DDSEC	DS	2	SECTOR NUMBER FOR DRIVER (* OBSOLETE ! *)
040.366	706X S.DDDTA	DS	2	DEVICE'S ADDRESS IN DEVLST +DEV.RES
040.370	707X S.DDOPC	DS	1	OPEN OPCODE PENDING
708X				
709X **	OVERLAY MANAGEMENT FLAGS			
710X				
000.001	711X OVL.IN	EQU	00000001B	IN MEMORY
000.002	712X OVL.RES	EQU	00000010B	PERMINANTLY RESIDENT
000.014	713X OVL.NUM	EQU	00001100B	OVERLAY NUMBER MASK
000.200	714X OVL.UCS	EQU	10000000B	USER CODE SWAPPED FOR OVERLAY
715X				
040.371	716X S.OVLFL	DS	1	OVERLAY FLAG
040.372	717X S.UCSF	DS	2	FWA SWAPPED USER CODE
040.374	718X S.UCSL	DS	2	LENGTH SWAPPED USER CODE
040.376	719X S.OVLS	DS	2	SIZE OF OVERLAY CODE
041.000	720X S.OVLE	DS	2	ENTRY POINT OF OVERLAY CODE
721X				
041.002	722X S.SSN	DS	2	SWAP AREA SECTOR NUMBER
041.004	723X S.OSN	DS	2	OVERLAY SECTOR NUMBER
724X				
725X *	SYSCALL PROCESSING WORK AREAS			
726X				
041.006	727X S.CACC	DS	1	(ACC) UPON SYSCALL
041.007	728X S.CODE	DS	1	SYSCALL INDEX IN PROGRESS
729X				
730X *	JUMPS TO ROUTINES IN RESIDENT HDOS CODE			
731X				
041.010	732X S.JUMPS	DS	0	START OF DUMP VECTORS
041.010	733X S.SDD	DS	3	JUMP TO STAND-IN DEVICE DRIVER
041.013	734X S.FASER	DS	3	JUMP TO FATERR (FATAL SYSTEM ERROR)
041.016	735X S.DIREA	DS	3	JUMP TO DIREAD (DISK FILE READ)
041.021	736X S.FCI	DS	3	JUMP TO FCI (FETCH CHANNEL INFO)
041.024	737X S.SCI	DS	3	JUMP TO SCI (STORE CHANNEL INFO)
041.027	738X S.GUP	DS	3	JUMP TO GUP (GET UNIT POINTER)
739X				
041.032	740X S.MOUNT	DS	1	<>0 IF THE SYSTEM DISK IS MOUNTED
041.033	741X S.DCS	DS	1	DEFAULT CLUSTER SIZE-1
742X				
041.034	743X S.BOOTF	DS	1	BOOT FLAGS
000.001	744X BOOT.P	EQU	00000001B	EXECUTE PROLOGUE UPON BOOTUP

```

745X
746X *      STACK VALUE SAVED FOR OVERLAY SYSCALLS
747X
041.035    748X S.OVSTK DS      2      VALUE OF SP UPON SYSCALLS USING OVERLAY
749X
041.037    750X      DS      1      RESERVED

752X **    ACTIVE I/O AREA.
753X *
754X *      THE AIO.XXX AREA CONTAINS INFORMATION ABOUT THE I/O OPERATION
755X *      CURRENTLY BEING PERFORMED. THE INFORMATION IS OBTAINED FROM
756X *      THE CHANNEL TABLE, AND WILL BE RESTORED THERE WHEN DONE.
757X *
758X *      NORMALLY, THE AIO.XXX INFORMATION WOULD BE OBTAINED DIRECTLY
759X *      FROM VARIOUS SYSTEM TABLES VIA POINTER REGISTERS. SINCE THE
760X *      8080 HAS NO GOOD INDEXED ADDRESSING, THE DATA IS MANUALLY
761X *      COPIED INTO THE AIO.XXX CELLS BEFORE PROCESSING, AND
762X *      BACKDATED AFTER PROCESSING.
763X
041.040    764X AIO.VEC DS      3      JUMP INSTRUCTION
041.041    765X AIO.DDA EQU     *-2     DEVICE DRIVER ADDRESS
041.043    766X AIO.FLG DS      1      FLAG BYTE
041.044    767X AIO.GRT DS      2      ADDRESS OF GROUP RESERV TABLE
041.046    768X AIO.SPG DS      1      SECTORS PER GROUP
041.047    769X AIO.CGN DS      1      CURRENT GROUP NUMBER
041.050    770X AIO.CSI DS      1      CURRENT SECTOR INDEX
041.051    771X AIO.LGN DS      1      LAST GROUP NUMBER
041.052    772X AIO.LSI DS      1      LAST SECTOR INDEX
041.053    773X AIO.DTA DS      2      DEVICE TABLE ADDRESS
041.055    774X AIO.DES DS      2      DIRECTORY SECTOR
041.057    775X AIO.DEV DS      2      DEVICE CODE
041.061    776X AIO.UNI DS      1      UNIT NUMBER (0-9)
777X
041.062    778X AIO.DIR DS      DIRELEN  DIRECTORY ENTRY
779X
041.111    780X AIO.CNT DS      1      SECTOR COUNT
041.112    781X AIO.EOM DS      1      END OF MEDIA FLAG
041.113    782X AIO.EOF DS      1      END OF FILE FLAG
041.114    783X AIO.TFP DS      2      TEMP FILE POINTERS
041.116    784X AIO.CHA DS      2      ADDRESS OF CHANNEL BLOCK (IOC.DDA)

041.120    786X S.SCR DS      2      SYSTEM SCRATCH AREA ADDRESS

```

```

789          CODE    PIC          POSITION INDEPENDANT CODE
790
791
792 ***      HDOSOVL.SYS - HDOS OVERLAY PROCESSOR.
793 *
794 *      HDOSOVL.SYS PROCESSES THE SYSCALL FUNCTIONS WHICH ARE RARELY
795 *      ISSUED, SUCH AS FILE OPENS AND CLOSES, ETC.
796 *
797 *      ENTRY    (SP)    = CODE
798 *              (SP+2)  = (HL)
799 *              (SP+6)  = (RET)
800 *              (S.CACC) = USER (ACC)
801 *      EXIT    TO (SP+6)
802 *              'C' CLEAR IF OK
803 *              'C' SET IF ERROR
804 *              (A) = ERROR CODE
805 *      USES    ALL
806
807
808 000.006 361      808 HDOSOVL POP    PSW          (A) = CODE
809 000.007 326 040 809          SUI    .LINK
810 000.011 322 021 000 810          JNC    HOSOVL2      COMMAND IS IN RANGE
811
812 *      COMMAND ERROR
813
814 000.014 341      814 HDOSOVL1 POP    H          RESTORE USER HL
815 000.015 076 003 815 ERRILC MVI    A,EC.ILC      ILLEGAL CODE
816 000.017 067      816          STC
817 000.020 311      817          RET          EXIT WITH ERROR
818
819 000.021 376 023 819 HDOSOVL2 CPI    HOSVECL      SEE IF IN RANGE
820 000.023 322 014 000 820          JNC    HOSOVL1      NOT IN RANGE
821 000.026 041 046 000 821          LXI    H,HOSVEC
822 000.031 207      822          ADD    A          (A) = 2*CODE
823 000.032 315 101 030 823          CALL  $DADA.
824 000.035 176      824          MOV    A,M
825 000.036 043      825          INX    H
826 000.037 146      826          MOV    H,M
827 000.040 157      827          MOV    L,A
828 000.041 343      828          XTHL          ((SP)) = PROCESSOR ADDRESS, (HL) = USER HL
829 000.042 072 006 041 829          LDA    S.CACC      (A) = USER ACC
830 000.045 311      830          RET          ENTER PROCESSOR
831
832 000.046          832 HDOSVEC DS    0          JUMP VECTORS
833
834 000.046 352 002 834          DW    LINK
835 000.000          835          ERRNZ *-HOSVEC/2+.LINK-.CTLC
836 000.050 242 003 836          DW    SCTL
837 000.000          837          ERRNZ *-HOSVEC/2+.LINK-.OPENR
838 000.052 114 000 838          DW    OPENR
839 000.000          839          ERRNZ *-HOSVEC/2+.LINK-.OPENW
840 000.054 215 000 840          DW    OPENW
841 000.000          841          ERRNZ *-HOSVEC/2+.LINK-.OPENU
842 000.056 340 000 842          DW    OPENU
843 000.000          843          ERRNZ *-HOSVEC/2+.LINK-.OPENC
844 000.060 020 001 844          DW    OPENC

```

000.000		845	ERRNZ	*-HOSVEC/2+.LINK-.CLOSE
000.062	213 001	846	DW	CLOSE
000.000		847	ERRNZ	*-HOSVEC/2+.LINK-.POSIT
000.064	126 002	848	DW	POSIT
000.000		849	ERRNZ	*-HOSVEC/2+.LINK-.DELET
000.066	357 001	850	DW	DELETE
000.000		851	ERRNZ	*-HOSVEC/2+.LINK-.RENAM
000.070	025 002	852	DW	RENAME
000.000		853	ERRNZ	*-HOSVEC/2+.LINK-.SETTP
000.072	270 003	854	DW	SETTOP
000.000		855	ERRNZ	*-HOSVEC/2+.LINK-.DECODE
000.074	241 002	856	DW	DECODE
000.000		857	ERRNZ	*-HOSVEC/2+.LINK-.NAME
000.076	300 002	858	DW	NAME
000.000		859	ERRNZ	*-HOSVEC/2+.LINK-.CLEAR
000.100	351 003	860	DW	CLEAR
000.000		861	ERRNZ	*-HOSVEC/2+.LINK-.CLEARA
000.102	013 004	862	DW	CLRALL
000.000		863	ERRNZ	*-HOSVEC/2+.LINK-.ERROR
000.104	041 004	864	DW	ERROR
000.000		865	ERRNZ	*-HOSVEC/2+.LINK-.CHFLG
000.106	263 004	866	DW	CHFLG
000.000		867	ERRNZ	*-HOSVEC/2+.LINK-.DISMT
000.110	334 004	868	DW	DMOUNT
000.000		869	ERRNZ	*-HOSVEC/2+.LINK-.LOADD
000.112	014 005	870	DW	LOADD
		871		
000.023		872	HOSVECL EQU	*-HOSVEC/2 MAX FUNCTION INDEX

```

875
876 *** OPENR - OPENR SYSCALL PROCESSOR.
877 *
878 * OPENR IS CALLED TO OPEN A CHANNEL FOR READ.
879 *
880 * THE CALLER SUPPLIES A FILE NAME, A DEFAULT BLOCK FOR THE DEVICE
881 * AND EXTENSION, AND A CHANNEL NUMBER.
882 *
883 * DEFAULT BLOCK FORMAT:
884 *
885 * DB 'DDD' DEFAULT DEVICE
886 * DB 'XXX' DEFAULT EXTENSION
887 *
888 * ENTRY (DE) = DEFAULT BLOCK ADDRESS
889 * (HL) = NAME ADDRESS
890 * (A) = CHANNEL NUMBER
891 * EXIT 'C' CLEAR IF OK
892 * (HL) = ADVANCED PAST FILE NAME
893 * 'C' SET IF ERROR
894 * (A) = ERROR CODE
895 * USES ALL
896
897
000.114 898 OPENR EQU *
000.114 305 899 PUSH B SAVE (BC)
000.115 315.150.000 900 CALL OPENR1 PROCESS OPEN
901
902 ** OPENR, OPENW, OPENU RETURN HERE
903 *
904 * (A) = EXIT CODE.
905
000.120 332.143.000 906 OPENX JC OPENEX ERROR EXIT
000.123 315.040.041 907 CALL AIO.VEC OPEN DEVICE CODE
000.126 052.116.041 908 LHLB AIO.CHA
000.131 021.041.041 909 LXI D,AIO.DDA
000.134 001.050.000 910 LXI B,IOCELEN-IOC,DDA
000.137 315.252.030 911 CALL $MOVE RESTORE TO IO CHANNEL
000.142 247 912 ANA A CLEAR CARRY
000.143 301 913 OPENEX POP B
000.144 041.000.000 914 LXI H,0 (HL) POINTS TO NEXT FILE (SET BY DFA)
000.145 915 OPENHL EQU *-2
000.147 311 916 RET
917
918 * PROCESS OPENR
919
000.150 315.221.005 920 OPENR1 CALL DFC DECODE FILE AND CHANNEL
000.153 330 921 RC ERROR
000.154 346.002 922 ANI DT,CR SEE IF CAPABLE OF READING
000.156 076.005 923 MVI A,EC.DNS
000.160 067 924 STC
000.161 310 925 RZ DEVICE NOT READ TYPE
926
927 * MUST FIND DIRECTORY ENTRY
928
000.162 176 929 MOV A,M (A) = TYPE
000.163 346.003 930 ANI FT,DD+FT,OR OPEN

```

```
000.165 167          931      MOV      M,A          SET TUPE
000.166 037          932      RAR
000.000          933      ERRNZ  FT,DD-1
000.167 322 212 000  934      JNC      OPENR2      NOT DIRECTORY DEVICE
000.172 315 076 005  935      CALL    CFC          CHECK FOR CONFLICT IF DIRECTORY
000.175 346 004      936      ANI      FT,OW      CANT HAVE ANY WRITES
000.177 076 013      937      MVI      A,EC,FUC   FILE USAGE CONFLICT
000.201 067          938      STC
000.202 300          939      RNZ
000.203 315 041 007  940      CALL    LDE.        HAVE CONFLICT
000.206 330          941      RC          LOCATE DIRECTORY ENTRY
000.207 315 157 005  942      CALL    CFC          IF ERROR
000.212 076 003      943  OPENR2 MVI      A,DC,OPR  COPY FILE INFO FROM DIRECTORY ENTRY
000.214 311          944      RET          SET CODE
```

```

947 *** OPENW - OPEN FILE FOR WRITE
948 *
949 * OPENW IS CALLED TO OPEN A CHANNEL FOR WRITE.
950 *
951 * THE FILE IS ENTERED IN THE CHANNEL TABLE, BUT NOT ON THE
952 * DISK. IT WILL BE ENTERED IN THE DIRECTORY AT CLOSE TIME.
953 *
954 * THE CALLER SUPPLIES A FILE NAME, A DEFAULT BLOCK FOR THE DEVICE
955 * AND EXTENSION, AND A CHANNEL NUMBER.
956 *
957 * DEFAULT BLOCK FORMAT:
958 *
959 * DB 'DDD' DEFAULT DEVICE
960 * DB 'XXX' DEFAULT EXTENSION
961 *
962 * ENTRY (DE) = DEFAULT BLOCK ADDRESS
963 * (HL) = NAME ADDRESS
964 * (A) = CHANNEL NUMBER
965 * EXIT 'C' CLEAR IF OK
966 * (HL) = ADVANCED PAST FILE NAME
967 * 'C' SET IF ERROR
968 * (A) = ERROR CODE
969 * USES ALL
970 *
971 *
972 *
000.215 973 OPENW EGU *
000.215 305 974 PUSH B SAVE (BC)
000.216 315 224 000 975 CALL OPENW1 PERFORM OPEN
000.221 303 120 000 976 JMP OPENX FINISH IN COMMON CODE
977 *
978 * PROCESS OPENW
979 *
000.224 315 221 005 980 OPENW1 CALL DFC DECODE FILE AND CHANNEL
000.227 330 981 RC ERROR
000.230 346 004 982 ANI DT,CW
000.232 076 035 983 MVI A,EC,DIW DEVICE IS NOT WRITABLE
000.234 067 984 STC
000.235 310 985 RZ NOT SUITABLE
000.236 176 986 MOV A,M (A) = FLAG BYTE
000.237 346 007 987 ANI FT,OW+FT,DD+FT,OR SET OPEN FOR WRITE (AND MAYBE READ)
000.241 167 988 MOV M,A
000.000 989 ERRNZ FT,DD-1
000.242 037 990 RAR
000.243 322 335 000 991 JNC OPENW3 NOT DIRECTORY TYPE
000.246 315 212 005 992 CALL CFP CHECK FOR POSSESSION
000.251 330 993 RC IF ERROR
000.252 315 041 007 994 CALL LDE. LOCATE ENTRY IN DIRECTORY
000.255 332 273 000 995 JC OPENW2 NO ENTRY IN DIRECTORY
996 *
997 * HAVE EXISTING ENTRY IN DIRECTORY. SEE IF WE HAVE WRITE PERMISSION,
998 * SINCE WE WILL DELETE THIS ENTRY WHEN WE CLOSE.
999 *
000.260 021 016 000 1000 LXI D,DIR,FLG
000.263 031 1001 DAD D (HL) = ADDRESS OF DIR,FLG IN ENTRY
000.264 176 1002 MOV A,M (A) = DIR,FLG

```

```

000.265 346 040 1003 ANI DIF.WP
000.267 076 024 1004 MVI A,EC.MPV ASSUME WRITE PROTECT VIOLATION
000.271 067 1005 STC
000.272 300 1006 RNZ VIOLATION
1007
1008 * ALLOCATE AND LINK THE FILE'S FIRST GROUP
1009
000.273 072 033 041 1010 OPENW2 LDA S.DCS (A) = SYSTEM DEFAULT CLUSTER SIZE
000.276 062 077 041 1011 STA AIO.DIR+DIR.CLU SET CLUSTER SIZE
1012
000.301 016 000 1013 MVI C,0 NEEDENT RE CONTIGUOUS
000.303 121 1014 MOV D,C NO BLOCK PREFERENCES
000.304 315 133 032 1015 CALL FFB FIND FREE BLOCK
1016
000.307 076 002 1017 MVI A,EC.EOM
000.311 330 1018 RC NO FREE BLOCK TO HAVE
000.312 176 1019 MOV A,M (A) = NEXT BLOCK IN FREE CHAIN
000.313 066 000 1020 MVI M,0 THIS BLOCK IS LAST (AND FIRST) FOR THE FILE
000.315 125 1021 MOV D,L (D) = INDEX OF NEW BLOCK
000.316 153 1022 MOV L,E (HL) = ADDRESS OF PREVIOUS BLOCK
000.317 167 1023 MOV M,A UNCHAIN ALLOCATED BLOCK FROM FREE LIST
000.320 152 1024 MOV L,D (L) = GROUP INDEX
000.321 046 000 1025 MVI H,0 (H) = SECTOR INDEX
000.323 042 047 041 1026 SHLD AIO.CGN
000.000 1027 ERRNZ AIO.CSI-AIO.CGN-1
000.326 042 051 041 1028 SHLD AIO.LGN SET FIRST=LAST
000.000 1029 ERRNZ AIO.LSI-AIO.LGN-1
000.331 175 1030 MOV A,L
000.332 062 102 041 1031 STA AIO.DIR+DIR.FGN SET FIRST GROUP NUMBER
1032
000.335 076 004 1033 OPENW3 MVI A,DC.OPW SET OPENW OPERATION
000.337 311 1034 RET EXIT
  
```



```

1037 *** OPENU - OPEN FILE FOR UPDATE.
1038 *
1039 * OPENU IS CALLED TO OPEN A CHANNEL FOR UPDATE.
1040 *
1041 * UPDATE IS JUST LIKE READ, BUT THE FILE MAY BE WRITTEN ALSO.
1042 *
1043 * THE CALLER SUPPLIES A FILE NAME, A DEFAULT BLOCK FOR THE DEVICE
1044 * AND EXTENSION, AND A CHANNEL NUMBER.
1045 *
1046 * DEFAULT BLOCK FORMAG:
1047 *
1048 * DB 'DDD' DEFAULT DEVICE
1049 * DB 'XXX' DEFAULT EXTENSION
1050 *
1051 * ENTRY (DE) = DEFAULT BLOCK ADDRESS
1052 * (HL) = NAME ADDRESS
1053 * (A) = CHANNEL NUMBER
1054 * EXIT 'C' CLEAR IF OK
1055 * (HL) = ADVANCED PAST FILE NAME
1056 * 'C' SET IF ERROR
1057 * (A) = ERROR CODE
1058 * USES ALL
1059 *
1060
000.340 1061 OPENU EQU *
000.340 305 1062 PUSH B SAVE (BC)
000.341 315 347 000 1063 CALL OPENU1 PROCESS OPEN
000.344 303 120 000 1064 JMP OPENX FINISH IN COMMON CODE
1065
1066 * PROCESS OPENU
1067
000.347 315 221 005 1068 OPENU1 CALL DFC DECODE FILE AND CHANNEL
000.352 330 1069 RC ERROR
000.353 057 1070 CMA
000.354 348 007 1071 ANI DT,DD+DT,CR+DT,CW MUST BE DIRECTORY/READ/WRITE
000.356 067 1072 STC
000.357 076 005 1073 MVI A,EC,DNS
000.361 300 1074 RNZ DEVICE DOESNT MEET ALL REQUIREMENTS.
1075
1076 * FIND DIRECTORY ENTRY
1077
000.362 176 1078 MOV A,M
000.363 348 007 1079 ANI FT,OR+FT,OW+FT,DD
000.365 366 010 1080 ORI FT,OU SET OPEN FOR UPDATE
000.367 167 1081 MOV M,A
000.370 315 212 005 1082 CALL CFP CHECK FOR POSESSION
000.373 330 1083 RC FILE USAGE CONFLICT
000.374 315 041 007 1084 CALL LDE, FIND ENTRY
000.377 330 1085 RC ERROR
001.000 315 157 005 1086 CALL CFP COPY FILE INFO FROM DIRECTORY ENTRY
1087
1088 * SEE IF WE HAVE WRITE PERMISSION
1089
001.003 072 100 041 1090 LDA AIO,DIR+DIR,FLG
001.006 348 040 1091 ANI DIF,WP
001.010 076 024 1092 MVI A,EC,WPU ASSUME WRITE PERMISSION VIOLATION

```

001.012	087	1093	STC		
001.013	300	1094	RNZ		IN ERROR
001.014	247	1095	ANA	A	CLEAR CARRY
001.015	076 005	1096	MVI	A,DC.OPU	
001.017	311	1097	RET		EXIT TO COMMON CODE

```

1100 *** OPENC - OPEN CONTIGUOUS FILE FOR WRITE
1101 *
1102 * OPENC IS CALLED TO OPEN A CHANNEL FOR WRITE.
1103 *
1104 * THE FILE IS ENTERED IN THE CHANNEL TABLE, BUT NOT ON THE
1105 * DISK. IT WILL BE ENTERED IN THE DIRECTORY AT CLOSE TIME.
1106 *
1107 * THE CALLER SUPPLIES A FILE NAME, A DEFAULT BLOCK FOR THE DEVICE
1108 * AND EXTENSION, AND A CHANNEL NUMBER.
1109 *
1110 * DEFAULT BLOCK FORMAG:
1111 *
1112 * DB 'DDD' DEFAULT DEVICE
1113 * DB 'XXX' DEFAULT EXTENSION
1114 *
1115 * ENTRY (C) = SECTOR COUNT
1116 * (DE) = DEFAULT BLOCK ADDRESS
1117 * (HL) = NAME ADDRESS
1118 * (A) = CHANNEL NUMBER
1119 * EXIT (C) CLEAR IF OK
1120 * (HL) = ADVANCED PAST FILE NAME
1121 * (C) SET IF ERROR
1122 * (A) = ERROR CODE
1123 * USES ALL
1124 *
1125
1126
001.020 1127 OPENC EQU *
001.020 305 1128 PUSH B SAVE (BC)
001.021 315 027 001 1129 CALL OPENC1 PERFORM OPEN
001.024 303 120 000 1130 JMP OPENX FINISH IN COMMON CODE
1131
1132 * PROCESS OPENC
1133
001.027 305 1134 OPENC1 PUSH B SAVE COUNT
001.030 315 221 005 1135 CALL DEC DECODE FILE AND CHANNEL
001.033 301 1136 POP B
001.034 330 1137 RC ERROR
001.035 346 004 1138 ANI DT,CW
001.037 076 035 1139 MVI A,EC,DIW DEVICE IS NOT WRITABLE
001.041 067 1140 STC
001.042 310 1141 RZ NOT SUITABLE
001.043 176 1142 MOV A,M (A) = FLAG BYTE
001.044 346 007 1143 ANI FT,DW+FT,DD+FI,OR SET OPEN FOR READ AND WRITE
001.046 167 1144 MOV M,A
000.000 1145 ER&NZ FT,DD-1
001.047 037 1146 RAR
001.050 322 210 001 1147 JNC OPENC? NOT DIRECTORY TYPE
001.053 171 1148 MOV A,C
001.054 247 1149 ANA A
001.055 302 061 001 1150 JNZ OPENC2 NOT 0
001.060 074 1151 INR A FORCE AT LEAST 1
001.061 062 114 001 1152 OPENC2 STA OPENCA SAVE COUNT
001.064 315 212 005 1153 CALL CFP CHECK FOR POSSESSION
001.067 330 1154 RC IF ERROR
001.070 315 041 007 1155 CALL LBE LOCATE ENTRY IN DIRECTORY
  
```

```

001.073 076 026 1156 MVI A,EC.FAP FILE ALREADY PRESENT
001.075 077 1157 CMC
001.076 330 1158 RC IF PRESENT
1159
1160 * READY TO TRY TO FIND THE CONTINUOUS AREA.
1161
001.077 052 044 041 1162 LHL0 AIO.BRT
001.102 125 1163 MOV D,L (D) = PREVIOUS BLOCK INDEX
001.103 156 1164 MOV L,M (HL) = ADDRESS IF FIRST FREE BLOCK
1165
1166 * AM AT HEAD OF A NEW CONTIGUOUS GROUP OF FREE BLOCKS
1167 * (MAY BE ONLY 1 BLOCK, CONTIGUOUS WITH ITSELF!)
1168
001.104 175 1169 OPENC3 MOV A,L SEE IF NO MORE BLOCKS
001.105 247 1170 ANA A
001.106 076 002 1171 MVI A,EC.EDM ASSUME EDM
001.110 067 1172 STC
001.111 310 1173 RZ EDM
001.112 325 1174 PUSH B SAVE INDEX OF GROUP BEFORE THIS ONE
001.113 006 000 1175 MVI B,0 (B) = GROUP COUNT
001.114 1176 OPENC4 EQU *-1
1177
1178 * FOLLOW BLOCKS IN CHAIN, LOOKING FOR GAP
1179
001.115 005 1180 OPENC4 DCR B
001.116 312 135 001 1181 JZ OPENC5 GOTEM
001.121 175 1182 MOV A,L
001.122 125 1183 MOV D,L (D) = INDEX OF LAST BLOCK
001.123 156 1184 MOV L,M FOLLOW LINK
001.124 074 1185 INR A
001.125 275 1186 CMP L
001.126 312 115 001 1187 JE OPENC4 STILL CONTIGOUS
1188
1189 * BREAK IN CONTINUITY, START COUNTING OVER AGAIN
1190
001.131 301 1191 POP B DISCARD OLD ADDRESS
001.132 303 104 001 1192 JMP OPENC3 TRY AGAIN
1193
1194 * GOT THE BLOCKS WE NEED.
1195 *
1196 * (L) = LAST BLOCK
1197 * ((SP)) = INDEX OF BLOCK BEFORE FIRST
1198 *
1199 * WE MUST
1200 *
1201 * 1) REMOVE THIS BLOCK OF GROUPS BY LINKING
1202 * THE GROUP BEFORE IT TO THE GROUP AFTER IT
1203 * 2) TERMINATE THIS CHAIN OF GROUPS BY ZEROING THE LAST LINK
1204 * 3) SETUP THE FILE POINTERS IN THE AIO.XXX FIELD
1205
001.135 321 1206 OPENC5 POP D (D) = INDEX OF GROUP BEFORE FIRST IN BLOCK
001.136 106 1207 MOV B,M (B) = FIRST FREE GROUP AFTER BLOCK
001.137 066 000 1208 MVI M,0 CLEAR CHAIN IN LAST GROUP OF BLOCK
001.141 135 1209 MOV E,L (E) = LAST GROUP NUMBER IN BLOCK
001.142 152 1210 MOV L,D (L) = LAST FREE BLOCK BEFORE GROUP
001.143 176 1211 MOV A,M (A) = FIRST GROUP OF BLOCK

```

OPENC - OPEN CONTIGUOUS FILE

14:14:50 16-MAY-80

```
001.144 160 1212 MOV M,B CHAIN FREE AROUND GROUP
001.145 062 102 041 1213 STA AIO,DIR+DIR.FGN SET FIRST GROUP
001.150 056 001 1214 MVI L,1
001.152 165 1215 MOV M,L FLAG CHANGE IN GRT
001.153 157 1216 MOV L,A
001.154 046 000 1217 MVI H,0
001.156 042 047 041 1218 SHLD AIO,CGN SET CURRENT GROUP AND INDEX
001.161 153 1219 MOV L,E (L) = LAST GROUP IN BLOCK
001.162 072 046 041 1220 LDA AIO,SPG
001.165 147 1221 MOV H,A
001.166 042 051 041 1222 SHLD AIO,LGN SET LAST GROUP AND SECTOR
001.171 042 103 041 1223 SHLD AIO,DIR+DIR.LGN
001.174 076 003 1224 MVI A,3
001.176 062 077 041 1225 STA AIO,DIR+DIR.CLU SET CLUSTER SIZE
001.201 041 100 041 1226 LXI H,AIO,DIR+DIR.FLG
001.204 176 1227 MOV A,M (A) = FLAG
001.205 366 020 1228 ORI DIF,CNT FLAG CONTIGUOUS
001.207 167 1229 MOV M,A REPLACE
1230
1231 * OPEN COMPLETE, SET CODE AND EXIT
1232
001.210 076 004 1233 OPENC? MVI A,DC.OPW SET OPENC OPERATION
001.212 311 1234 RET EXIT
```

```

1237 *** CLOSE - PROCESS CLOSE SYSCALL.
1238 *
1239 * CLOSE PROCESSING DEPENDS UPON THE FILE AND DEVICE TYPE.
1240 *
1241 * FOR A WRITE/DIRECTORY TYPE, THE DIRECTORY IS SEARCHED FOR A
1242 * PREVIOUS ENTRY. IF FOUND, IT IS DELETED. THE NEW ENTRY IS THEN
1243 * INSERTED
1244 *
1245 * FOR A UPDATE/DIRECTORY TYPE, THE PREVIOUS ENTRY IS UPDATED.
1246 *
1247 * FOR ALL FILES, THE DRIVER IS CALLED WITH THE DC,CLO FUNCTION.
1248 * THE CHANNEL IS RELEASED.
1249 *
1250 * ENTRY (A) = CHANNEL #
1251 * EXIT 'C' CLEAR IF OK
1252 * 'C' SET IF ERROR
1253 * (A) = CODE
1254 * USES ALL
1255 *
1256 *
001,213 1257 CLOSE EQU *
001,213 315 245 006 1258 CALL FCC FETCH COMPLETE CHANNEL
001,216 330 1259 RC ERROR
001,217 247 1260 ANA A
001,220 312 344 031 1261 JZ ERR,FND FILE NOT OPEN
001,223 037 1262 RAR
000,000 1263 ERKNZ FT,DD-1
001,224 322 343 001 1264 JNC CLOSE8 IS NOT DIRECTORY FILE
001,227 346 002 1265 ANI FT,OW/2
001,231 312 343 001 1266 JZ CLOSE8 IS READ
1267 *
1268 * FILE IS DIRECTORY FILE, AND WAS WRITTEN TO
1269 *
001,234 176 1270 MOV A,M (A) = AIO,FLG
001,235 346 010 1271 ANI FT,OU
001,237 041 266 001 1272 LXI H,CLOSE2 ASSUME WAS UPDATE TYPE
001,242 302 250 001 1273 JNZ CLOSE1 IS UPDATE
001,245 041 275 001 1274 LXI H,CLOSE3 IS OPEN FOR WRITE
001,250 345 1275 CLOSE1 PUSH H SAVE PROCESS AS RETURN ADDRESS
001,251 052 055 041 1276 LHLD AIO,DES (HL) = DIRECTORY SECTOR #
001,254 001 015 000 1277 LXI B,DIRIDL (BC) = DIRECTORY NAME LENGTH
001,257 174 1278 MOV A,M
001,260 265 1279 ORA L
001,261 067 1280 STC ASSUME NO PREVIOUS ENTRY
001,262 302 057 007 1281 JNZ LDE,, HAVE PREVIOUS ENTRY
001,265 311 1282 RET EXIT TO CLOSE2 OR CLOSE3
1283 *
1284 * AM OPEN FOR UPDATE
1285 *
001,266 334 013 041 1286 CLOSE2 CC S,FASER FATAL ERROR! COULDN'T FIND IT
001,271 353 1287 XCHG (DE) = ENTRY ADDRESS
001,272 303 321 001 1288 JMP CLOSE7 UPDATE DIRECTORY ENTRY
1289 *
1290 * AM OPEN FOR WRITE. DELETE ANY EXISTING FILE OF THAT NAME
1291 *
001,275 332 306 001 1292 CLOSE3 JC CLOSE4 NONE YET EXIST

```

001.300	315	353	007	1293		CALL	RBF	RETURN BLOCKS TO FREE POOL
001.303	303	312	001	1294		JMP	CLOSE6	SET CREATION DATE AND UPDATE DIRECTORY
				1295				
				1296	*			FIND EMPTY SPOT FOR THE FILE NAME
				1297				
001.306	315	360	006	1298	CLOSE4	CALL	FOE	FIND OPEN ENTRY
001.311	330			1299		RC		DIRECTORY FULL UP
001.312	353			1300	CLOSE6	XCHG		
001.313	052	310	040	1301		LHLD	S,DATC	
001.316	042	105	041	1302		SHLD	AIO,DIR+DIR,CRD	SET CREATION DATE
				1303				
				1304	*			UPDATE DIRECTORY ENTRY FOR WRITTEN FILE
				1305	*			
				1306	*			(DE) = ADDRESS OF ENTRY IN DIRECTORY SECTOR
				1307				
001.321	052	310	040	1308	CLOSE7	LHLD	S,DATC	
001.324	042	107	041	1309		SHLD	AIO,DIR+DIR,ALD	SET ALTERATION DATE
001.327	353			1310		XCHG		(HL) = ADDRESS
001.330	315	156	010	1311		CALL	UDE	UPDATE DIRECTORY ENTRY
001.333	315	111	010	1312		CALL	SGT	SAVE GRT TABLE
001.336	330			1313		RC		ERROR
001.337	315	177	010	1314		CALL	UBS	UPDATE BIRECTORY SECTOR
001.342	330			1315		RC		ERROR
				1316				
				1317	*			CLOSE I/O CHANNEL
				1318				
001.343	052	116	041	1319	CLOSE8	LHLD	AIO,CHA	
000.000				1320		ERRNZ	IOC,FLG-IOC,DDA-2	
001.346	043			1321		INX	H	
001.347	043			1322		INX	H	(HL) = ADDRESS OF IOC,FLG
001.350	066	000		1323		MVI	M,0	CLEAR
				1324				
				1325	*			CALL CLOSE IN DEVICE DRIVER
				1326				
001.352	076	006		1327		MVI	A,DC,CLO	
001.354	303	040	041	1328		JMP	AIO,VEC	CLOSE AND RETURN

DELETE - PROCESS DELETE FUNCTION

14:14:52 16-MAY-80

```

1331 **      DELETE - PROCESS DELETE COMMAND.
1332 *
1333 *      ENTRY (HL) = NAME STRING
1334 *      (DE) = DEFAULT BLOCK
1335 *      EXIT 'C' CLEAR IF OK
1336 *      'C' SET IF ERROR
1337 *      (A) = CODE
1338 *      USES ALL
1339
1340
001.357      1341 DELETE EQU *
001.357 315 371 007 1342 CALL LFD LOCATE FILE IN DIRECTORY
001.362 330      1343 RC ERROR OF SOME KIND
1344
1345 *      SEE IF FILE IS WRITE PROTECTED
1346
001.363 353      1347 XCHG (DE) = DIRECTORY ENTRY ADDRESS
001.364 041 016 000 1348 LXI H,DIR.FLG
001.367 031      1349 DAD D (HL) = #DIR.FLG
001.370 176      1350 MOV A,M
001.371 346 040 1351 ANI DIF.WP SEE IF WRITE PROTECT
001.373 076 024 1352 MVI A,EC.WPV ASSUME WRITE PROTECT VIOLATION
001.375 067      1353 STC
001.376 300      1354 RNZ IS PROTECTED
001.377 072 043 041 1355 LDA AID.FLG
002.002 346 004 1356 ANI FT.OW
002.004 076 035 1357 MVI A,EC.DIW DEVICE IS NOT WRITABLE
002.006 067      1358 STC
002.007 310      1359 RZ
002.010 353      1360 XCHG (HL) = ADDRESS OF DIR ENTRY
1361
1362 *      DELETE FILE
1363
002.011 066 377 1364 MVI M,DF.EMP FLAG EMPTY
002.013 315 353 007 1365 CALL RBF RETURN BLOCKS TO FREE POOL
002.016 315 177 010 1366 CALL UDS UPDATE DIRECTORY SECTOR
002.021 330      1367 RC ERROR
002.022 303 111 010 1368 JMP SGT SAVE GRT TABLE, AND EXIT
    
```



```

1371 *** RENAME - PROCESS RENAME FUNCTION.
1372 *
1373 * RENAME RENAMES A FILE ON A DIRECTORY DEVICE.
1374 *
1375 * * NOTE * RENAME DOES NOT CHECK TO SEE IF THE NEW NAME ALREADY
1376 * EXISTS, THIS IS CURRENTLY THE RESPONSIBILITY OF THE CALLER !
1377 *
1378 * ENTRY (HL) = NAME STRING
1379 * (DE) = DEFAULT BLOCK
1380 * (BC) = NEW NAME STRING
1381 * EXIT 'C' CLEAR IF OK
1382 * 'C' SET IF ERROR
1383 * (A) = CODE
1384 * USES ALL
1385
1386
002.025 315 371 007 1387 RENAME CALL LFD LOCATE FILE IN DIRECTORY
002.030 330 1388 RC FILE NOT FOUND
1389
1390 * SEE IF DEVICE IS READ AND WRITABLE
1391
002.031 072 043 041 1392 LDA AIO.FLG
002.034 057 1393 CMA
002.035 346 007 1394 ANI DT,DD+DT,CR+DT,CW
002.037 076 035 1395 MVI A,EC.DIW DEVICE NOT WRITABLE
002.041 067 1396 STC
002.042 300 1397 RNZ WE KNOW IS DISK, AND ERGO READABLE, MUST BE WRITE PROT
1398
1399 * HAVE FOUND DIRECTORY ENTRY. CRACK NEW NAME.
1400
002.043 345 1401 PUSH H SAVE ENTRY ADDRESS
002.044 021 016 000 1402 LXI D,DIR.FLG
002.047 031 1403 DAD D (HL) = ADDRESS OF FLAG BYTE
002.050 176 1404 MOV A,M
002.051 346 140 1405 ANI DIF,LOC+DIF,WP DONT RENAME IF LOCKED OR WRITE PROTECT
002.053 312 063 002 1406 JZ RENAME1 NOT LOCKED OR WRITE PROTECT
002.056 076 024 1407 MVI A,EC.WPV ASSUME FILE WRITE PROTECTED
002.060 067 1408 STC
002.061 341 1409 POP H CLEAN STACK
002.062 311 1410 RET
1411
002.063 052 060 041 1412 RENAME1 LHLD AIO.UNI-1 (H) = AIO.UNI
002.066 345 1413 PUSH H SAVE CURRENT UNIT
002.067 140 1414 MOV H,B
002.070 151 1415 MOV L,C (HL) = NEW NAME ADDRESS
002.071 021 320 031 1416 LXI D,$ZEROS
002.074 315 305 005 1417 CALL DFD DECODE FILE DESCRIPTOR
002.077 301 1418 POP B (B) = UNIT NUMBER
002.100 341 1419 POP H (HL) = ADDRESS OF ENTRY IN DIRECTORY
002.101 330 1420 RC ERROR
002.102 076 007 1421 MVI A,EC.IFN ERROR CODE, ILLEGAL FILE NAME
002.104 067 1422 STC
002.105 310 1423 RZ NULL FILE NAME
002.106 170 1424 MOV A,B (A) = UNIT NUMBER
002.107 062 061 041 1425 STA AIO.UNI SET ORIGINAL UNIT NUMBER
000.000 1426 ERRNZ DIR.NAM ASSUMED FIRST OFF

```

RENAME - PROCESS RENAME FUNCTION

14:14:56 16-MAY-80

002.112	021	062	041	1427	LXI	D,AIO.DIR+DIR.NAM	
002.115	001	015	000	1428	LXI	B,DIRIDL	
002.120	315	252	030	1429	CALL	\$MOVE	MOVE IN NEW NAME
002.123	303	177	010	1430	JMP	UDS	UPDATE DIRECTORY SECTOR AND RETURN

```

1433 *** POSIT - POSITION FILE.
1434 *
1435 * LXI B,POSITION
1436 * MVI A,CHANNEL NUMBER
1437 * SYSCALL POSIT
1438 *
1439 * ENTRY (A) = CHANNEL NUMBER
1440 * (BC) = SECTOR NUMBER TO POSITION BEFORE
1441 * EXIT 'C' CLEAR IF OK
1442 * 'C' SET IF ERROR
1443 * (A) = ERROR CODE
1444 * (A) = EC.EOF IF OFF END
1445 * (BC) = SECTORS UNSKIPPED (REMAINDER OF COUNT)
1446 * FILE POSITIONED AT EOF
1447 * USES ALL
1448 *
1449 *
002.126 1450 POSIT EQU *
002.126 305 1451 PUSH B SAVE SECTOR #
002.127 315 245 006 1452 CALL FCC FETCH COMPLETE CHANNEL INFORMATION
002.132 330 1453 RC ERROR
002.133 247 1454 ANA A
002.134 312 344 031 1455 JZ ERR.FND FILE NOT OPEN
000.000 1456 ERRNZ FT.DD-1
002.137 037 1457 RAR
002.140 077 1458 CMC 'C' SET IF NOT DIRECTORY DEVICE
002.141 076 005 1459 MVI A,EC.DNS
002.143 330 1460 RC DEVICE NOT SUITABLE
002.144 052 102 041 1461 LHLD AIO.DIR+DIR.FGN
002.147 046 000 1462 MVI H,0
002.151 042 047 041 1463 SHLD AIO.CGN REWIND
000.000 1464 ERRNZ AIO.CSI-AIO.CGN-1
002.154 301 1465 POP B (BC) = SECTOR COUNT
1466 *
002.155 170 1467 POSIT2 MOV A,B
002.156 261 1468 ORA C
002.157 312 024 041 1469 JZ S.SCI ALL DONE
002.162 315 201 002 1470 CALL AFP ADVANCE FILE POINTER
002.165 353 1471 XCHG
002.166 332 024 041 1472 JC S.SCI AT EOF
002.171 353 1473 XCHG
002.172 013 1474 DCX B MORE TO GO
002.173 303 155 002 1475 JMP POSIT2 ADVANCE ANOTHER ONE

```

```

1477 ** AFP - ADVANCE FILE POINTER.
1478 *
1479 * AFP ADVANCES THE ACTIVE I/O POINTER ONE SECTOR.
1480 *
1481 * ENTRY NONE
1482 * EXIT 'C' CLEAR IF ADVANCED
1483 * 'C' SET IF EOF
1484 * (A) = EC.EOF
1485 * USES A,F,H,L

```

```
1486  
1487  
002.176 315 205 032 1488 AFF1 CALL FPL FOLLOW FORWARD LINK  
1489  
002.201 041 047 041 1490 AFF LXI H,AIO.CGN  
002.204 072 051 041 1491 LDA AIO.LGN  
002.207 276 1492 CMP M SEE IF AM ON LAST GROUP  
000.000 1493 ERRNZ AIO.CSI-AIO.CGN-1  
002.210 043 1494 INX H (HL) = #AIO.CSI  
002.211 302 227 002 1495 JNE AFF2 NOT YET  
002.214 072 052 041 1496 LDA AIO.LSI  
002.217 276 1497 CMP M SEE IF ALSO ON LAST SECTOR  
002.220 302 227 002 1498 JNE AFF2 NOT YET  
002.223 067 1499 STC  
002.224 076 001 1500 MVI A,EC,EOF  
002.226 311 1501 RET AT END OF FILE  
1502  
1503 * NOT AT END OF FILE  
1504  
002.227 072 046 041 1505 AFF2 LDA AIO.SPG  
002.232 276 1506 CMP M  
002.233 312 176 002 1507 JE AFF1 AT END OF GROUP  
002.236 064 1508 INR M ADVANCE  
002.237 247 1509 ANA A CLEAR CARRY  
002.240 311 1510 RET
```

```

1513 *** DECODE - PROCESS DECODE SYSCALL.
1514 *
1515 * DECODE DECODES THE SUPPLIED FILE NAME INTO A BLOCK IN THE FORM:
1516 *
1517 * DB FLAG DEVICE FLAG
1518 * DS 2 DEVICE NAME
1519 * DS 1 DEVICE UNIT
1520 * DS 8 FILE NAME
1521 * DS 3 FILE EXTENSION
1522 * DS 1 PROJECT
1523 * DS 1 VERSION
1524 * DS 2 DEVICE TABLE POINTER
1525 *
1526 * ENTRY (BC) = LIST FWA
1527 * (DE) = DEFAULT TABLE
1528 * (HL) = NAME ADDRESS
1529 * EXIT NONE
1530 * USES ALL
1531
1532
002,241 1533 DECODE EQU *
002,241 305 1534 PUSH B SAVE (BC)
002,242 315 234 005 1535 CALL DFA DECODE FILENAME INTO AIO,XXX
002,245 301 1536 POP B
002,246 330 1537 RC ERROR
002,247 140 1538 MOV H,B
002,250 151 1539 MOV L,C (HL) = ADDRESS OF BLOCK
002,251 072 043 041 1540 LDA AIO,FLG
002,254 167 1541 MOV M,A STORE
002,255 043 1542 INX H
000,000 1543 ERRNZ AIO,DIR-AIO,UNI-1
000,000 1544 ERRNZ AIO,UNI-AIO,DEV-2
002,256 001 020 000 1545 LXI B,DIRIDL+3
002,261 021 057 041 1546 LXI D,AIO,DEV
002,264 315 252 030 1547 CALL *MOVE MOVE IN STUFF
002,267 353 1548 XCHG
002,270 052 053 041 1549 LHLD AIO,DTA
002,273 353 1550 XCHG
002,274 163 1551 MOV M,E
002,275 043 1552 INX H
002,276 162 1553 MOV M,D SET DEV TABLE ADDRESS
002,277 311 1554 RET
  
```

```

1557 *** NAME - PROCESS NAME SYSCALL.
1558 *
1559 * THE NAME SYSCALL RETURNS THE DEVICE, FILE NAME, AND
1560 * FILE EXTENSION OF AN OPEN CHANNEL.
1561 *
1562 * THE INFORMATION IS OBTAINED FROM THE CHANNEL TABLE, WHICH WAS
1563 * SETUP UPON FILE OPEN.
1564 *
1565 * ENTRY (A) = CHANNEL NUMBER
1566 * (DE) = ADDRESS FOR DEVICE AND EXTENSION (DEFAULT BLOCK FORMAT)
1567 * (HL) = ADDRESS FOR NAME (8 CHARACTERS, FOLLOWED BY 00)
1568 * EXIT 'C' CLEAR IF OK
1569 * 'C' SET IF ERROR
1570 * (A) = ERROR CODE
1571 * USES ALL
1572
1573
002.300 345 1574 NAME PUSH H
002.301 325 1575 PUSH D
002.302 315 245 006 1576 CALL FCC FETCH COMPLETE CHANNEL
002.305 322 312 002 1577 JNC NAME1 NO ERROR
002.310 321 1578 POP D
002.311 311 1579 RET RETURN WITH ERROR
1580
002.312 341 1581 NAME1 POP H (HL) = ADDRESS FOR DEVICE, ETC
002.313 021 057 041 1582 LXI D,AIO.DEV
002.316 001 003 000 1583 LXI B,3
002.321 315 252 030 1584 CALL $MOVE MOVE IN DEVICE CODE
002.324 053 1585 DCX H
002.325 176 1586 MOV A,M
002.326 306 060 1587 ADI '0' MAKE INTO DIGIT
002.330 167 1588 MOV M,A
002.331 043 1589 INX H
002.332 343 1590 XTHL (HL) = ADDRESS FOR NAME
002.333 001 010 000 1591 LXI B,8
000.000 1592 ERRNZ AIO.DIR+DIR.NAM-AIO.DEV-3
002.336 315 252 030 1593 CALL $MOVE MOVE IN NAME
002.341 066 000 1594 MVI M,0 FLAG END OF NAME
002.343 341 1595 POP H (HL) = ADDRESS FOR EXTENSION
002.344 001 003 000 1596 LXI B,3
000.000 1597 ERRNZ DIR.EXT-DIR.NAM-B
002.347 303 252 030 1598 JMP $MOVE MOVE EXTENSION, AND EXIT

```

```

1601 *** LINK - PROCESS LINK SYSCALL.
1602 *
1603 * LINK LOADS IN AND RUNS ANOTHER PROGRAM. THE OPEN FILES,
1604 * SYSTEM TABLES, AND STACK ARE NOT DISTURBED.
1605 *
1606 * ENTRY (HL) = ADDRESS OF PROGRAM FILE DESCRIPTOR
1607 * EXIT TO LINKED PROGRAM, IF OK
1608 * (A) UNCHANGED
1609 * (SP) = VALUE AT 'LINK' SYSCALL
1610 * TO CALLER IF ERROR
1611 * 'C' SET
1612 * (A) = ERROR CODE
1613 * USES ALL
1614 *
1615 *
002.352 1616 LINK EQU *
002.352 315 234 005 1617 CALL DFA DECODE FILE INTO AIO.XXX
002.355 330 1618 RC ERROR
002.356 057 1619 CMA SEE IF DIRECTOR, FOR READ
002.357 346 003 1620 ANI FT,BD+FT,OR
002.361 076 005 1621 MVI A,EC,DNS ASSUME NOT SUITABLE
002.363 067 1622 STC
002.364 300 1623 RNZ NOT SUITABLE
002.365 066 003 1624 MVI M,FT,DR+FT,OR SET OPEN FOR READ
002.367 315 041 007 1625 CALL LDE. LOCATE DIRECTORY ENTRY
002.372 330 1626 RC ERROR
002.373 315 157 005 1627 CALL CFD COPY FILE INFO FROM DIRECTORY ENTRY
1628 *
1629 * READ 1ST SECTOR TO GET LOAD ADDRESS AND LENGTH INFORMATION.
1630 *
002.376 001 000 001 1631 LXI B,256
003.001 353 1632 XCHG /79.11.6C/
003.002 052 120 041 1633 LHLD S,SCR /79.11.6C/
003.005 353 1634 XCHG DE = SECTOR SCRATCH AREA /79.11.6C/
003.006 315 016 041 1635 CALL S,DIREA READ IT
003.011 330 1636 RC ERROR
1637 *
1638 * SEE IF ABS FILE
1639 *
003.012 052 120 041 1640 LHLD S,SCR /79.11.6C/
003.015 176 1641 MOV A,M /79.11.6C/
000.000 1642 ERRNZ ABS,ID /79.11.6C/
003.016 074 1643 INK A
003.017 076 020 1644 MVI A,EC,IFC ILLEGAL FILE CONTENTS
003.021 067 1645 STC
003.022 300 1646 RNZ FILE IS NOT BINARY TYPE
1647 *
003.023 043 1648 INX H /79.11.6C/
003.024 176 1649 MOV A,M /79.11.6C/
000.000 1650 ERRNZ ABS,LDA-ABS,ID-2 TRY TO CATCH ERROR? /79.11.6C/
003.025 376 000 1651 CPI FT,ABS /79.11.6C/
003.027 076 020 1652 MVI A,EC,IFC
003.031 067 1653 STC ASSUME NOT FT,ABS
003.032 300 1654 RNE NOT ABS
1655 *
1656 * SEE IF ENOUGH ROOM

```

```

1657
003.033 043 1658 INX H /79.11.GC/
003.034 315 310 010 1659 CALL ILDEHL DE = LOAD ADDRESS /79.11.GC/
000.000 1660 ERRNZ ABS.LDA-ABS.ID-2 /79.11.GC/
1661
003.037 325 1662 PUSH D /79.11.GC/
003.040 315 310 010 1663 CALL ILDEHL /79.11.GC/
000.000 1664 ERRNZ ABS.LEN-ABS.LDA-2 /79.11.GC/
003.043 353 1665 XCHG HL = LOAD LENGTH /79.11.GC/
003.044 321 1666 POP D
1667
003.045 031 1668 DAD D (HL) = LWA+1
003.046 353 1669 XCHG (DE) = LWA+1
003.047 041 047 003 1670 LXI H,* MUSENT OVERLAY LINK
003.052 315 224 030 1671 CALL $CHL COMPLEMENT (HL)
003.055 031 1672 DAD D
003.056 076 021 1673 MVI A,EC,NEM ASSUME NOT ENOUGH MEM
003.060 330 1674 RC NOT ENOUGH MEM
003.061 353 1675 XCHG (HL) = LOAD LWA
003.062 315 270 003 1676 CALL SETTOP /80.01.GC/
003.065 330 1677 RC /80.01.GC/
1678
1679 * ENOUGH MEMORY FOR LOAD. COPY 1ST PARTIAL SECTOR INTO PLACE
1680
003.066 001 370 000 1681 LXI B,256-ABS.COD /79.11.GC/
003.071 052 120 041 1682 LHLD S,SCR /79.11.GC/
003.074 315 234 030 1683 CALL $INDL /79.11.GC/
003.077 002 000 1684 DW ABS.LDA
003.101 325 1685 PUSH D SAVE LOAD ADDRESS /79.11.GC/
003.102 021 010 000 1686 LXI D,ABS.COD /79.11.GC/
003.105 031 1687 DAD D /79.11.GC/
003.106 353 1688 XCHG DE = FWA OF CODE /79.11.GC/
003.107 341 1689 POP H HL = LOAD ADDRESS /79.11.GC/
1690
003.110 315 252 030 1691 CALL $MOVE
1692
1693 * COMPUTE SECTOR COUNT FOR REMAINDER OF LOAD
1694
003.113 345 1695 PUSH H SAVE LOAD ADDRESS /79.11.GC/
003.114 052 120 041 1696 LHLD S,SCR /79.11.GC/
003.117 315 234 030 1697 CALL $INDL /79.11.GC/
003.122 004 000 1698 DW ABS.LEN DE = LENGTH /79.11.GC/
003.124 041 010 000 1699 LXI H,ABS.COD /79.11.GC/
003.127 031 1700 DAD D ADD BYTES FOR ABS HEADER /79.11.GC/
003.130 016 000 1701 MVI C,0
003.132 175 1702 MOV A,L
003.133 306 377 1703 ADI 3770
003.135 174 1704 MOV A,H
003.136 211 1705 ADC C
003.137 107 1706 MOV B,A (BC) = SECTOR COUNT*256
003.140 005 1707 DCR B COUNT SECTOR READ
003.141 321 1708 POP D RESTORE LOAD ADDRESS /79.11.GC/
003.142 315 016 041 1709 CALL S,DIREA READ IT
1710
1711 * UPDATE CHANNEL TABLE SO CHANNEL 3770 IS THE FILE LOADED
1712

```



```

003.145 001 050 000 1713 LXI B,IOCELEN-IOC.DDA
003.150 021 041 041 1714 LXI D,AIO.DDA
003.153 052 352 040 1715 LHLD S,CFWA
000.000 1716 ERRNZ IOCCTD-1 ASSUME FIRST IN CHANTAB IS 377Q
000.000 1717 ERRNZ IOC.DDA-2
003.156 043 1718 INX H
003.157 043 1719 INX H (HL) = #IOC.DDA IN CHANNEL
003.160 315 252 030 1720 CALL $MOVE MOVE INFO INTO CHANNEL
1721
1722 * SETUP SYSTEM FOR TRANSFER OF CONTROL.
1723 *
1724 * 1) CLEAR CONTROL CHARACTER PROCESS TABLE
1725 * 2) CLEAR SYSMODE FLAG
1726 * 3) CLEAR USER CODE SWAPPED FLAGS
1727
003.163 041 333 040 1728 LXI H,S.CAADR
003.166 006 010 1729 MVI B,8
003.170 315 212 031 1730 CALL $ZERO CLEAR VECTOR TABLE, AND PENDING VECTOR
003.173 052 346 040 1731 LHLD S,DLINK (HL) = HIGHMEM POINTER
000.000 1732 ERRNZ M,SYSM CLEAR SYSMODE
003.176 066 000 1733 MVI M,0
003.200 021 005 000 1734 LXI D,M.CINT
003.203 031 1735 DAD D
003.204 066 000 1736 MVI M,0 CLEAR SCINTFL
003.206 072 371 040 1737 LDA S,OVLFL
003.211 346 177 1738 ANI 377Q-OVL,UCS
003.213 062 371 040 1739 STA S,OVLFL CLEAR USER CODE SWAPPED
003.216 315 222 031 1740 CALL $WDR WRITE DISABLE RAM
000.001 1741 IF DEBUG
1742 LDA 40077A ** DEBUG **
1743 ANA A
1744 JNZ 160000A ENTER MBUG
1745 ENDIF
003.221 052 035 041 1746 LHLD S,OVSTK
003.224 371 1747 SPHL RESTORE STACK
1748
003.225 052 120 041 1749 LHLD S,SCR /79.11.GC/
003.230 315 234 030 1750 CALL $INDL /79.11.GC/
003.233 006 000 1751 DW ABS,ENT /79.11.GC/
003.235 353 1752 XCHG HL = ENTRY POINT /79.11.GC/
1753
003.236 072 006 041 1754 LDA S,CACC RESTORE (A)
1755
003.241 351 1756 PCHL ENTER USER CODE

```

SCTLC - SET CONTROL CHARACTER ADDRESS

14:15:02 16-MAY-80

```

1759 *** SCTLC - SET CONTROL CHARACTER ADDRESS
1760 *
1761 * THE .CTLG SYSCALL IS USED TO SETUP HANDLING FOR
1762 * THE CONTROL CHARACTERS CTL-A, CTL-B, AND CTL-C.
1763 *
1764 * A SEPERATE ADDRESS IS SPECIFIABLE FOR EACH CHARACTER, IF
1765 * AN ADDRESS OF 0 IS SPECIFIED, PROCESSING OF THAT CHARACTER
1766 * IS SUSPENDED.
1767 *
1768 * THE PROCESS ADDRESS MUST BE > 255A.
1769 *
1770 * ENTRY (A) = CONTROL CHARACTER WHOSE PROCESS ADDRESS IS
1771 * TO CHANGE (CTL-A, CTL-B, OR CTL-C)
1772 * (HL) = NEW ADDRESS (=0 TO CLEAR PROCESSING)
1773 * EXIT 'C' CLEAR IF OK
1774 * 'C' SET IF ERROR
1775 * (A) = ERROR CODE
1776 * USES A,F,H,L
1777
1778
003,242 1779 SCTLC EQU *
003,242 075 1780 DCR A
003,243 372 015 000 1781 JM ERRILC ILLEGAL CODE
003,246 376 003 1782 CPI 3
003,250 322 015 000 1783 JNC ERRILC ILLEGAL CODE
003,253 353 1784 XCHG (R0) = PROCESS ADDRESS
003,254 041 335 040 1785 LXI H,S,CCTAB
003,257 207 1786 ADD A (A) = 2*INDEX
003,260 315 101 030 1787 CALL *DADA. (HL) = ADDRESS FOR ADDRESS STORAGEE
003,263 163 1788 MOV M,E
003,264 043 1789 INX H
003,265 162 1790 MOV M,D
003,266 247 1791 ANA A CLEAR CARRY
003,267 311 1792 RET
    
```

```

1795 *** SETUP - SET TOP OF USER MEMORY.
1796 *
1797 * SETUP IS CALLED TO NOTIFY THE SYSTEM OF A NEW MEMORY LIMIT
1798 * ADDRESS. IF NECESSARY, THE OVL WILL BE CLEARED.
1799 *
1800 * ENTRY (HL) = NEW ADDRESS
1801 * EXIT 'C' CLEAR IF OK
1802 * 'C' SET IF TOO HIGH
1803 * (HL) = MAXIMUM HEIGHT
1804 * USES ALL
1805
1806
003.270 1807 SETUP EQU *
003.270 353 1808 XCHG (DE) = NEW TOP
003.271 052 320 040 1809 LHL S.SYSM
003.274 053 1810 ICX H
003.275 053 1811 DCX H
003.276 175 1812 MOV A,L COMPARE S.SYSM TO SETUP
003.277 223 1813 SUB E
003.300 174 1814 MOV A,H
003.301 232 1815 SBB D
003.302 076 021 1816 MVI A,EC.NEM ASSUME NOT ENOUGH
003.304 330 1817 RC CANT HAVE IT
003.305 353 1818 XCHG
003.306 042 322 040 1819 SHLD S.USRM CAN HAVE IT
003.311 353 1820 XCHG
1821
1822 * SEE IF MUST UN-RESIDE THE OVL
1823
003.312 052 322 040 1824 LHL S.UCSF (HL) = FWA OF OVL LOAD
003.315 175 1825 MOV A,L
003.316 223 1826 SUB E
003.317 174 1827 MOV A,H
003.320 232 1828 SBB D (A) = OVLADR-USRTOP
003.321 041 371 040 1829 LXI H,S.OVLFL (HL) = #S.OVLFL
003.324 332 334 003 1830 JC SETTOP1 NOT ENOUGH ROOM TO RESIDE OVERLAY
1831
1832 * PLENTY OF ROOM TO RESIDE OVERLAY.
1833
003.327 176 1834 MOV A,M
003.330 346 177 1835 ANI 3770-OVL.UCS DONT RELOAD ANY SWAPPED USER CODE
003.332 167 1836 MOV M,A
003.333 311 1837 RET EXIT
1838
1839 * NOT ENOUGH ROOM TO RESIDE OVERLAY
1840
003.334 072 032 041 1841 SETTOP1 LIA S.MOUNT
003.337 247 1842 ANA A
003.340 076 021 1843 MVI A,EC.NEM
003.342 067 1844 STC
003.343 310 1845 RZ NOT ENOUGH MEMORY IF DISMOUNTED
003.344 176 1846 MOV A,M
003.345 346 376 1847 ANI 3770-OVL.IN CLEAR OVERLAY IN MEMORY FLAG
003.347 167 1848 MOV M,A CLEAR IN
003.350 311 1849 RET
  
```

```

1852 *** CLEAR - CLEAR I/O CHANNEL.
1853 *
1854 * CLEAR IS CALLED TO CLEAR AN I/O CHANNEL. IF THE CHANNEL IS CLOSED,
1855 * NO ACTION IS PERFORMED. IF THE CHANNEL IS OPEN, IT IS
1856 * FLAGGED CLOSED. THE RESULTS OF THIS OPERATION DEPEND UPON THE TYPE
1857 * OF FILE:
1858 *
1859 * OPEN FOR ACTION
1860 *
1861 * READ SAME AS .CLOSE
1862 * WRITE FILE IS FORGOTTEN. ANY WRITTEN DISK BLOCKS
1863 * ARE RESTORED TO THE FREE POOL.
1864 * UPDATE REPLACED SECTORS REMAIN REPLACED. APPENDED SECTORS
1865 * ARE LOST UNTIL NEXT BOOT. FILE STAYS AT PREVIOUS LENGTH.
1866 * WRITEC SAME AS WRITE
1867 *
1868 * THE DEVICE DRIVER IS NOT INFORMED OF THE CLOSING.
1869 *
1870 * DB SYSCALL, .CLEAR
1871 *
1872 * ENTRY (A) = CHANNEL NUMBER
1873 * EXIT 'C' CLEAR IF OK
1874 * 'C' SET IF ERROR
1875 * (A) = ERROR CODE
1876 * USES ALL
1877 *
003.351 1878
003.351 315 021 041 1879 CLEAR EQU *
003.354 330 1880 CALL S.FCI FETCH CHANNEL INFO
003.355 052 116 041 1881 RC ERROR
1882 LHL D AIO.CHA
1883
1884
1885 ** CLEAR1 - CLEAR CHANNEL.
1886 *
1887 * ENTRY (HL) = IOC.DDA ADDRESS
1888 * EXIT NONE
1889 * USES ALL
1890
000.000 1891 ERRNZ IOC.FLG-IOC.DDA-2
003.360 043 1892 CLEAR1 INX H
003.361 043 1893 INX H (HL) = #IOC.FLG ADDRESS
003.362 176 1894 MOV A,H
1895
1896 * IF FILE IS DIRECTORY DEVICE, OPEN FOR WRITE BUT NOT UPDATE, CLEAR BLOCKS
1897
003.363 356 010 1898 XRI FT,DU MUST NOT BE FOR UPDATE
003.365 057 1899 CMA
003.366 346 017 1900 ANI FT,OR+FT,OW+FT,DD+FT,OU FT.DD & FT.OW & FT.OR & FT.OU
003.370 066 000 1901 MVI M,0 CLEAR FLAGS
003.372 300 1902 RNZ NOT WRITE-ONLY MASS STORAGE, EXIT
1903
1904 * IS WRITE ONLY MASS STORAGE FILE, CLEAR SPACE
1905
000.000 1906 ERRNZ IOC.BRT-IOC.FLG-1
003.373 043 1907 INX H (HL) = #IOC.BRT

```

003.374	124	1908	MOV	D,H	
003.375	135	1909	MOV	E,L	(DE) = #IOC.GRT
003.376	315 211 030	1910	CALL	\$HLIHL	(HL) = GRT ADDRESS
004.001	345	1911	PUSH	H	SAVE
004.002	041 036 000	1912	LXI	H,IOC.DIR+DIR.FGN-IOC.GRT	
004.005	031	1913	DAD	D	(HL) = ADDRESS OF FIRST GROUP NUMBER
004.006	176	1914	MOV	A,M	(A) = FIRST GROUP INDEX
004.007	341	1915	POP	H	(HL) = GRT ADDRESS
004.010	303 322 006	1916	JMP	FBC	FREE GROUP CHAIN AND EXIT

```

1919 *** CLRALL - CLEAR ALL CHANNELS.
1920 *
1921 * CLRALL PERFORMS THE .CLEAR ACTION FOR ALL EXISTING CHANNELS,
1922 * EXCEPT CHANNEL 3770, THE LOAD IMAGE CHANNEL.
1923 *
1924 * DB SYSCALL, CLRALL
1925 *
1926 * ENTRY NONE
1927 * EXIT NONE
1928 * USES ALL
1929
1930
004.013 1931 CLRALL EQU *
004.013 052 352 040 1932 LHL D S,CFWA
000.000 1933 ERRNZ IOCCTD-1
004.016 315 211 030 1934 CALL $HLIHL NEED TO CHAIN PAST 1 CHANNEL
1935 POINT TO USER CHANNEL #0
004.021 174 1936 CLRALL1 MOV A,H
004.022 265 1937 ORA L
004.023 310 1938 RZ END OF CHANNELS
004.024 345 1939 PUSH H SAVE ADDRESS
000.000 1940 ERRNZ IOC.DDA-IOC.LNK-2
004.025 043 1941 INX H
004.026 043 1942 INX H POINT TO IOC.DDA
004.027 315 360 003 1943 CALL CLEAR1 CLEAR IT
004.032 341 1944 POP H (HL) = CHANNEL LINK ADDRESS
004.033 315 211 030 1945 CALL $HLIHL (HL) = ADDRESS OF NEXT ONE
004.036 303 021 004 1946 JMP CLRALL1
  
```

```

1949 *** ERROR - PRINT ERROR MESSAGE.
1950 *
1951 * ERROR IS CALLED TO PRINT AN ERROR MESSAGE.
1952 *
1953 * THE HDOS SYSTEM RETURNS ERROR CODE NUMBERS WHEN IT DETECTS
1954 * AN ERROR. THE ERROR FUNCTION MAY BE USED TO TYPE AN ALPHABETIC
1955 * EXPLANATION OF THE ERROR.
1956 *
1957 * THE ERRORS ARE STORED IN THE FILE 'ERRORMSG.SYS' ON THE SYSTEM
1958 * DISK, ONE MESSAGE PER LINE. THE LINES LOOK LIKE!
1959 *
1960 * NNTEXT
1961 *
1962 * FOR EXAMPLE,
1963 *
1964 * 001END OF MEDIA
1965 *
1966 * IF THE ERROR MESSAGE FILE CANNOT BE READ, OR THE MESSAGE DOES
1967 * NOT APPEAR, THE ERROR IS TYPED AS
1968 *
1969 * 'SYSTEM ERROR # NNN'
1970 *
1971 * ENTRY (A) = ERROR CODE
1972 * (H) = TRAILING CHARACTER (TYPED AFTER MESSAGE)
1973 * EXIT NONE
1974 * USES ALL
1975 *
1976 *
004.041 1977 ERROR EQU *
004.041 345 1978 PUSH H SAVE FINAL CHARACTER
1979 *
1980 * CRACK ERROR CODE
1981 *
004.042 117 1982 MOV C,A
004.043 006 000 1983 MVI B,0 (BC) = CODE
004.045 041 211 004 1984 LXI H,ERRORB
004.050 076 003 1985 MVI A,3
004.052 315 157 031 1986 CALL $UDD UNPACK DECIMAL DIGITS
004.055 315 074 004 1987 CALL ERR0.5
004.060 322 070 004 1988 JNC ERR4 CRACKED MESSAGE OK
1989 *
1990 * COULDNT GET MESSAGE
1991 *
004.063 041 173 004 1992 ERR3 LXI H,ERRORA
004.066 377 003 1993 DB SYSCALL, PRINT
1994 *
1995 * PRINT FINAL CHARACTER
1996 *
004.070 361 1997 ERR4 POP PSW (A) = CODE
004.071 377 002 1998 DB SYSCALL, SCOUT
004.073 311 1999 RET
2000 *
2001 * OPEN ERROR MESSAGE FILE
2002 *
004.074 2003 ERR0.5 EQU *
2004 * LDA S.MOUNT

```

```

2005 * SUI 1 'C' SET IF 0 /79.08.BC/
2006 * RC ERR3 SYSTEM DISMOUNTED: NO ERROR FILE /79.06.BC/
004.074 041 215 004 2007 LXI H,ERRDRC
004.077 315 234 005 2008 CALL DFA
004.102 330 2009 RC ERR3 FILE ERROR
004.103 315 041 007 2010 CALL LDE. LOCATE DIRECTORY ENTRY
004.106 330 2011 RC ERR3 FILE ERROR
004.107 315 157 005 2012 CALL CFD COPY FILE INFO FROM DIRECTORY ENTRY
2013
2014 * READ SECTORS FOR ERROR MESSAGE
2015
004.112 052 120 041 2016 LHLD S,SCR SECTOR SCRATCH AREA /79.11.GC/
004.115 055 2017 DCR L FORCE IMMEDIATE READ /79.11.GC/
2018
004.116 006 003 2019 ERRO MVI B,3
004.120 021 211 004 2020 LXI D,ERRDRB
2021
004.123 315 236 004 2022 ERR1 CALL RTB READ TEXT BYTE
004.126 330 2023 RC ERR3 END OF FILE
004.127 032 2024 LDAX D
004.130 276 2025 CMP M
004.131 302 156 004 2026 JNE ERR2,5 NOT RIGHT MESSAGE NUMBER
004.134 023 2027 INX D
004.135 005 2028 DCR B
004.136 302 123 004 2029 JNZ ERR1 KEEP TRYING
2030
2031 * GOT ERROR MESSAGE. TYPE IT
2032
004.141 315 236 004 2033 ERR2 CALL RTB READ MESSAGE BYTE
004.144 330 2034 RC ERR3 ERROR
004.145 176 2035 MOV A,M
004.146 376 012 2036 CPI NL SEE IF NEW LINE
004.150 310 2037 RE ERR4 END OF MESSAGE
004.151 377 002 2038 DB SYSCALL,SCDUT
004.153 303 141 004 2039 JMP ERR2
2040
2041 * THIS IS NOT THE RIGHT MESSAGE. READ THE NEXT
2042
004.156 315 236 004 2043 ERR2,5 CALL RTB
004.161 330 2044 RC ERR3 MESSAGE NOT FOUND
004.162 176 2045 MOV A,M (A) = MESSAGE CHARACTER
004.163 376 012 2046 CPI NL
004.165 302 156 004 2047 JNE ERR2,5 READ TILL NEXT LINE
004.170 303 116 004 2048 JMP ERRO TRY THIS MESSAGE
2049
004.173 007 077 060 2050 ERRORA DB BELL,'?02 SYS ERR #'
2051
004.211 112 107 114 2052 ERRORB DB 'JGL' MESSAGE NUMBER
004.214 240 2053 DB '+2000'
2054
004.215 123 131 060 2055 ERRDRC DB 'SYO:ERRORMSG.SYS';0
  
```



```
2057 ** RTB - READ TEXT BYTE
2058 *
2059 * RTB LOCATES THE NEXT BYTE OF THE FILE BEING READ.
2060 *
2061 * ENTRY (HL) = BUFFER POINTER
2062 * EXIT 'C' CLEAR IF BOT IT
2063 * (HL) = ADDRESS
2064 * 'C' SET IF ERROR
2065 * USES A,F,L
2066
004.236 247 2067 RTB ANA A CLEAR CARRY
004.237 054 2068 INR L
004.240 300 2069 RNZ GOT MORE
004.241 305 2070 PUSH B
004.242 325 2071 PUSH D
004.243 001 000 001 2072 LXI B,256
004.246 353 2073 XCHG /79.11.GC/
004.247 052 120 041 2074 LHLD S,SCR /79.11.GC/
004.252 353 2075 XCHG DE = SCRATCH ADDRESS /79.11.GC/
004.253 325 2076 PUSH D
004.254 315 016 041 2077 CALL S,DIREA
004.257 341 2078 POP H (HL) = POINTER
004.260 321 2079 POP D
004.261 301 2080 POP B
004.262 311 2081 RET
```

```

2085 ** CHFLG - CHANGE FILE FLAGS.
2086 *
2087 * CHFLG IS CALLED TO CHANGE THE FILE DESCRIPTION FLAGS
2088 * FOR A MASS STORAGE FILE. ONLY CERTAIN FLAGS MAY BE
2089 * CHANGED:
2090 *
2091 * FLAG BIT MEANING
2092 *
2093 * DIF.SYS 2000 IS SYSTEM FILE
2094 * DIF.WP 0400 IS WRITE PROTECTED
2095 * DIF.LOC 1000 LOCKED FOR CHANGE (SETABLE ONLY)
2096 *
2097 * CHFLG WILL REFUSE THE OPERATION IF THE DIF.LOC BIT IS SET.
2098 *
2099 * ENTRY (B) = NEW BIT VALUES
2100 * (C) = CHANGE MASK (BIT SET FOR EVERY BIT TO REPLACE FROM (B))
2101 * (DE) = DEFAULT BLOCK ADDRESS
2102 * (HL) = FILE NAME
2103 * EXIT 'C' CLEAR, CHANGE DONE
2104 * 'C' SET, ERROR
2105 * (A) = ERROR CODE
2106 * USES ALL
2107
004.263 315 371 007 2109 CHFLG CALL LFD LOCATE FILE IN DIRECTORY
004.266 330 2110 RC IF ERROR
2111
004.267 072 043 041 2112 LDA AIO.FLG /79.12.GC/
004.272 346 004 2113 ANI DT.CW /79.12.GC/
004.274 076 025 2114 MVI A,EC.WP ASSUME DISKETTE IS WRITE PROTECTED /79.12.GC/
004.276 067 2115 STC /79.12.GC/
004.277 310 2116 RZ NOT CAPABLE OF WRITE /79.12.GC/
2117
004.300 171 2118 MOV A,C
000.001 2119 IF DEBUG
2120 ANI DIF.SYS+DIF.WP+DIF.LOC ** DEBUG ** ALLOW ALL
2121 ELSE
004.301 346 240 2122 ANI DIF.SYS+DIF.WP ALLOW ONLY SYS AND WP
2123 ENDIF
004.303 057 2124 CMA
004.304 117 2125 MOV C,A (C) = ~MASK
004.305 057 2126 CMA
004.306 366 100 2127 ORI DIF.LOC ALLOW HIM TO LOCK
004.310 240 2128 ANA B CLEAR UNWANTED BITS
004.311 021 016 000 2129 LXI D,DIR.FLG
004.314 031 2130 DAD D (HL) = ADDRESS OF DIR.FLG BYTE
004.315 107 2131 MOV B,A
004.316 176 2132 MOV A,M (A) = FLAG BYTE
004.317 346 100 2133 ANI DIF.LOC
004.321 076 030 2134 MVI A,EC.FL ASSUME FILE LOCKED
004.323 067 2135 STC
000.001 2136 IF DEBUG
2137 ELSE
004.324 300 2138 RNZ LOCKED
2139 ENDIF
004.325 176 2140 MOV A,M

```

CHFLG

.....
004.326 241 2141
004.327 260 2142
004.330 167 2143
004.331 303 177 010 2144
.....

ANA C
ORA R
MOV M,A
JMP UDS

.....
CLEAR BITS
SET BITS
REPLACE
UPDATE DIRECTORY SECTOR
.....

```

2148 *** DMOUNT - FLAG SYSTEM DISK DISMOUNTED.
2149 *
2150 * THE DMOUNT FUNCTION IS USED WHEN THE SYSTEM DISK IS ABOUT TO BE
2151 * DISMOUNTED. ANY HDOS FUNCTIONS WHICH REQUIRE SYSTEM FILES
2152 * WILL BE TREATED AS FATAL SYSTEM ERRORS.
2153 *
2154 * IF THE ACCUMULATOR IS NON-ZERO UPON THE DMOUNT CALL,
2155 * THE OVERLAY WILL BE CLEARED FROM MEMORY. SYSTEME CALLS
2156 * REQUIRING THE OVERLAY WILL THEN BECOME ILLEGAL.
2157 *
2158 * ENTRY (A) = 0 IF TO RETAIN OVERLAY
2159 * (A) <> 0 IF TO DISCARD OVERLAY
2160 * EXIT S.SYSM = LWA OF FREE SPACE FOR USER
2161 * (HL) = (S.SYSM)
2162 * USES ALL
2163
004.334 2164
004.334 041 371 040 2165 DMOUNT EQU *
004.337 247 2166 LXI H,S,OVLFL
004.340 076 000 2167 ANA A
004.342 062 032 041 2168 MVI A,0
004.345 312 357 004 2169 STA S,MOUNT FLAG DISK UNMOUNTED
004.350 176 2170 JZ DMOUNT1 AM TO RETAIN OVERLAY
004.351 346 374 2171 MOV A,M (A) = S,OVLFL
004.353 167 2172 ANI 3770-OVL,IN-OVL,RES
004.354 303 372 004 2173 MOV M,A CLEAR PRESENSE
2174 JMP DMOUNT2 FINISH UP
2175
2176 * LEAVE OVERLAYS IN. SEE IF USER CODE SWAPPED
2177
004.357 176 2178 DMOUNT1 MOV A,M
000.000 2179 ERRNZ OVL,UCS-2000
004.360 027 2180 RAL
004.361 076 021 2181 MVI A,FC,NEM
004.363 330 2182 RC NOT ENOUGH MEMORY
004.364 052 372 040 2183 LHLD S,UCSF
004.367 042 320 040 2184 SHLD S,SYSM SET MEMORY AT START OF OVERLAY
2185
2186 * FLAG DEVICES DISMOUNTED
2187
004.372 052 354 040 2188 DMOUNT2 LHLD S,DFWA
004.375 021 010 000 2189 LXI D,DEV,MUM
005.000 031 2190 DAD D (HL) = #DEV,MUM FOR DEVICE
005.001 066 000 2191 MVI M,0 CLEAR MOUNTED
005.003 156 2192 MOV L,M
005.004 145 2193 MOV H,L (HL) = 0
005.005 042 324 040 2194 SHLD S,OMAX OVERLAY SIZE NO LONGER PERTINANT
005.010 052 320 040 2195 LHLD S,SYSM (HL) FOR EXIT
005.013 311 2196 RET
    
```

```
2198 *** LOADD - LOAD DEVICE DRIVER
2199 *
2200 * LOADD LOADS THE SPECIFIED DEVICE DRIVER.
2201 *
2202 *
2203 * ENTRY: (HL) = DEVICE DRIVER DESCRIPTOR STRING
2204 *
2205 * EXIT: NONE
2206 *
2207 * USES: ALL
2208 *
2209
005.014 2210 LOADD EQU *
005.014 021 031 005 2211 LXI D,LOAA
005.017 315 234 005 2212 CALL DFA DECODE FILE DESCRIPTOR INTO ACTIVE
005.022 330 2213 RC
005.023 076 011 2214 MVI A,DC.LOD
005.025 062 370 040 2215 STA S.DDOPC
005.030 311 2216 RET
2217
005.031 000 000 000 2218 LOAA DB 0,0,0,0,0 NULL DEFAULT DEVICE
```

```

2222 **      CAC - CLEAR ACTIVE CHANNEL.
2223 *
2224 *      CAC CLEARS OUT THE ACTIVE I/O BLOCK.
2225 *
2226 *      BYTES AIO.DDA TO AND INCLUDING AIO.YFF ARE ZEROED.
2227 *
2228 *      AIO.CHA IS LEFT AS IT IS.
2229 *
2230 *      ENTRY  NONE
2231 *      EXIT   NONE
2232 *      USES   A
2233
2234
005.037 345      2235 CAC  PUSH   H
005.040 041 041 041 2236      LXI   H,AIO.DDA
005.043 305      2237      PUSH  B
005.044 006 055  2238      MVI   B,AIO.CHA-AIO.DDA      (B) = LEN
005.046 315 212 031 2239      CALL  $ZERO
005.051 301      2240      POP   B
005.052 341      2241      POP   H      RESTORE (HL)
005.053 311      2242      RET

2244 **      CDU - CHECK DEVICE UNIT.
2245 *
2246 *      CDU CHECKS THE THIRD DEVICE SPECIFICATION CHARACTER. IF IT IS NULL,
2247 *      THE DEVICE UNIT IS 0. IF IT IS NON-NULL, IT MUST BE
2248 *      A DIGIT SPECIFYING THE DEVICE UNIT.
2249 *
2250 *      ENTRY  AIO.UNI = UNIT CODE
2251 *      EXIT   'C' CLEAR IF OK
2252 *           AIO.UNI = UNIT NUMBER
2253 *           'C' SET IF ERROR
2254 *           (A) = ERROR CODE
2255 *      USES   A,F,H,L
2256
2257
005.054 041 061 041 2258 CDU  LXI   H,AIO.UNI
005.057 176      2259      MOV   A,M
005.060 247      2260      ANA   A
005.061 310      2261      RZ           IS 0
005.062 326 060  2262      SUI   '0'
005.064 332 073 005 2263      JC   CDU1      ERROR
005.067 167      2264      MOV   M,A
005.070 376 010  2265      CPI   8
005.072 077      2266      CMC           'C' IF TOO LARGE
005.073 076 006  2267 CDU1 MVI   A,EC.IDN      ERROR CODE FOR ILLEGAL DEVICE NAME
005.075 311      2268      RET           'C' SET IF ERROR

```

```

2270 **      CFC - CHECK FILE CONFLICTS.
2271 *
2272 *      CFC CHECKS TO SEE IF A FILE TO BE OPENED CONFLICTS WITH ANOTHER
2273 *      FILE CURRENTLY OPEN.
2274 *
2275 *      A FILE OPEN FOR READ MAY HAVE MANY READERS,
2276 *      A FILE OPEN FOR WRITE (OR UPDATE) MAY ONLY HAVE ONE
2277 *      USER.
2278 *
2279 *      ENTRY  NONE
2280 *      EXIT   (A) = LOGICAL OR OF IOC.FLG OF ALL FILES WITH
2281 *             THE SAME NAME AND DEVICE
2282 *      USES   ALL
2283
2284
005.076 006 000 2285 CFC   MVI   B,0           (B) = FLAG ACCUMULATOR
005.100 052 352 040 2286       LHLD  S,CFWA        (HL) = FWA CHANNEL TABLE
2287
2288 *      SEE IF THIS CHANNEL USES SAME NAME AND DEVICE.
2289
005.103 136      2290 CFC1  MOV   E,M
005.104 043      2291       INX   H
005.105 124      2292       MOV   D,M
005.106 043      2293       INX   H
000.000          2294       ERRNZ  IOC.DDA-IOC.LNK-2      (HL) = #IOC.DDA
005.107 325      2295       PUSH  D           SAVE NEXT ADDRESS
005.110 021 041 041 2296       LXI   D,AIO.DDA
000.000          2297       ERRNZ  IOC.DDA-2      (HL) = #IOC.DDA OF ENTRY
005.113 016 002 2298       MVI   C,2
005.115 315 060 030 2299       CALL  $COMP      COMPARE
005.120 302 147 005 2300       JNE   CFC2      NO MATCH
000.000          2301       ERRNZ  IOC.FLG-IOC.DDA-2      (HL) = IOC.FLG OF ENTRY
005.123 345      2302       PUSH  H           SAVE
005.124 021 016 000 2303       LXI   D,IOC.UNI-IOC.DDA-2
000.000          2304       ERRNZ  IOC.DIR-IOC.UNI-1
000.000          2305       ERRNZ  AIO.DIR-AIO.UNI-1      COMPARE UNIT #'S AND NAMES
005.127 031      2306       DAD   D           (HL) = IOC.DIR POINTER
005.130 021 041 041 2307       LXI   D,AIO.UNI
005.133 016 016 2308       MVI   C,DIRIDL+1
005.135 315 060 030 2309       CALL  $COMP      COMPARE UNIT NUMBERS AND NAMES
005.140 341      2310       POP   H
005.141 302 147 005 2311       JNE   CFC2      NO MATCH
2312
2313 *      HAVE MATCH
2314
005.144 170      2315       MOV   A,B
005.145 266      2316       ORA   M           OR FLAG
005.146 107      2317       MOV   B,A
2318
005.147 341      2319 CFC2  POP   H           (HL) = NEXT ADDRESS
005.150 174      2320       MOV   A,H
005.151 265      2321       ORA   L           SEE IF MORE CHANNELS
005.152 302 103 005 2322       JNZ   CFC1      MORE
005.155 170      2323       MOV   A,B           (A) = FLAGS
005.156 311      2324       RET

```

```

2326 **   CFD - COPY FILE INFORMATION FROM DIRECTORY ENTRY.
2327 *
2328 *   CFD COPIES A DIRECTORY ENTRY INTO THE AIO.DIR FIELD.
2329 *
2330 *   THE FILE POINTERS (AIO.CGN, AIO.CSI, AIO.LGN, AIO.LSI)
2331 *   ARE SETUP.
2332 *
2333 *   ENTRY (HL) = ADDRESS OF ENTRY IN SECSCR
2334 *   EXIT (HL) = ADDRESS OF ENTRY IN SECSCR
2335 *   USES ALL
2336
2337
005.157 345 2338 CFD   PUSH   H           SAVE (HL)
005.160 353 2339       XCHG          (DE) = ADDRESS OF ENTRY
005.161 041 062 041 2340       LXI     H,AIO.DIR
005.164 001 027 000 2341       LXI     B,DIRELEM
005.167 315 252 030 2342       CALL   $MOVE          MOVE INTO LIST
005.172 052 103 041 2343       LHL   AIO.DIR+DIR.LGN
005.175 042 051 041 2344       SHLD  AIO.LGN         SET LGN, LSI
005.200 052 102 041 2345       LHL   AIO.DIR+DIR.FGN
005.203 257 2346       XRA   A           CLEAR 'C'
005.204 147 2347       MOV   H,A         SET AIO.CSI=0
005.205 042 047 041 2348       SHLD  AIO.CGN         STORE
005.210 341 2349       POP  H           RESTORE (HL)
005.211 311 2350       RET

```

```

2352 **   CFP - CHECK FOR POSESSION.
2353 *
2354 *   CFP IS CALLED TO SEE IF THE ACTIVE I/O TABLE IS THE SOLE USER
2355 *   OF THE SPECIFIED FILE. IF THE FILE IS OPENED UNDER ANY CHANNEL,
2356 *   AN ERROR IS FLAGGED.
2357 *
2358 *   ENTRY NONE
2359 *   EXIT 'C' CLEAR IF NO USERS
2360 *   'C' SET IF USERS
2361 *   (A) = CODE
2362 *   USES ALL
2363
2364
005.212 315 076 005 2365 CFP   CALL   CFC           CHECK FOR CONFLICT
000.000 2366       ERRNZ FT,DD-1
005.215 037 2367       RAR          'C' SET IF ANY DIRECTORY USERS
005.216 076 013 2368       MVI   A,EC.FUC     FILE USAGE CONFLICT
005.220 311 2369       RET

```



```

2371 **      DFC - DECODE FILE AND CHANNEL
2372 *
2373 *      DFC LOCATES THE SPECIFIED CHANNEL TABLE ENTRY, AND PREPARES
2374 *      THE ACTIVE CHANNEL WITH FILE NAME AND DEVICE INFORMATION.
2375 *
2376 *      THE SPECIFIED CHANNEL NUMBER IS CHECKED FOR AVAILABILITY,
2377 *      THE FILE NAME IS CRACKED INTO THE AIO,XXX FIELDS, AND
2378 *      THE DEVICE DRIVER INFORMATION IS LOCATED.
2379 *
2380 *      ENTRY  (A) = CHANNEL NUMBER TO OPEN
2381 *             (DE) = DEFAULT BLOCK ADDRESS
2382 *             (HL) = FILE NAME IN ASCII
2383 *      EXIT   'C' CLEAR IF OK
2384 *             AIO.FLG = DEVICE TYPE FLAGS
2385 *             (HL) = #AIO.FLG
2386 *             (A) = (AIO.FLG)
2387 *             'C' SET IF ERROR
2388 *             (A) = ERROR CODE
2389 *      USES   ALL
2390 *
2391 *
005.221 345      2392 DFC  PUSH  H          SAVE TEXT ADDRESS
005.222 315 021 041 2393  CALL  S.FCI     FETCH CHANNEL INFORMATION
005.225 341      2394  POP   H          (HL) = TEXT ADDRESS
005.226 330      2395  RC    H          ERROR IN CHANNEL NUMBER
005.227 247      2396  ANA   A          'Z' CLEAR IF IN USE
005.230 076 004  2397  MVI   A,EC.CNA
005.232 067      2398  STC
005.233 300      2399  RNZ
2400 *      JMP    DFA          CHANNEL NOT AVAILABLE
                                DECODE FILE INFO INTO ACTIVE CHANNEL

2402 **      DFA - DECODE FILE INFORMATION INTO ACTIVE CHANNEL.
2403 *
2404 *      IT CRACKS THE FILE NAME INTO THE AIO,DIR FIELDS, AND
2405 *      DECODES DEVICE INFORMATION, AND LOADS THE DEVICE DRIVER.
2406 *
2407 *      ENTRY  (DE) = DEFAULT BLOCK ADDRESS
2408 *             (HL) = FILE NAME IN ASCII
2409 *      EXIT   'C' CLEAR IF OK
2410 *             (HL) = #AIO.FLG
2411 *             (A) = AIO.FLG
2412 *             'C' SET IF ERROR
2413 *             (A) = ERROR CODE
2414 *      USES   ALL
2415 *
2416 *
005.234 315 037 005 2417 DFA  CALL  CAC          CLEAR ACTIVE CHANNEL
005.237 315 305 005 2418  CALL  DFD          DECODE FILE DESCRIPTOR
005.242 330      2419  RC    H          ERROR
005.243 042 145 000 2420  SHLD  OPENHL     SAVE POINTER
005.246 315 054 005 2421  CALL  CDU          CHECK DEVICE UNIT
005.251 330      2422  RC    H          ERROR
005.252 315 165 007 2423  CALL  LDI          LOCATE DEVICE INFORMATION
005.255 330      2424  RC    H          IF ERROR

```

```

2425
2426 *      GOT NAME, DEVICE INFO, ETC. POINT TO FLAG BYTE.
2427
005.256 041 043 041 2428 LXI  H,AIO.FLG
005.261 176          2429 MOV  A,M
005.262 346 001     2430 ANI  DT,DD
005.264 312 303 005 2431 JZ   DFA1          NOT DIRECTORY DEVICE
2432
005.267 072 062 041 2433 LDA  AIO.DIR+DIR.NAM
005.272 376 000     2434 CPI  0
005.274 302 303 005 2435 JNZ  DFA1          NOT NULL-NAME
2436
005.277 076 007     2437 MVI  A,EC.IFN      ERROR CODE, ILLEGAL FILE NAME
005.301 067          2438 STC
005.302 311          2439 RET
2440
005.303 176          2441 DFA1 MOV  A,M      (A) = (AIO.FLG)
005.304 311          2442 RET
2443

2444 **     DFD - DECODE FILE DESCRIPTION.
2445 *
2446 *     DFD CRACKS AN ALPHANUMERIC FILE DESCRIPTION; OF THE FORM
2447 *
2448 *     DEV:NAME.EXT
2449 *
2450 *     ENTRY (DE) = POINT TO DEFAULT BLOCK
2451 *           (HL) = POINTER TO TEXT
2452 *     EXIT  'C' SET IF ERROR
2453 *           (A) = ERROR CODE
2454 *           'C' CLEAR IF OK
2455 *           (HL) = POINTS PAST FILE NAME
2456 *           'Z' SET IF NULL NAME
2457 *           'Z' CLEAR IF NON-NULL
2458 *           AIO.DIR.NAM = NAME
2459 *           AIO.DIR.EXT = EXTENSION
2460 *           AIO.DEV = DEVICE CODE
2461 *           AIO.UNI = UNIT NUMBER (ASCII DIGIT)
2462 *     USES  ALL
2463
005.305 345          2464
2465 DFD  PUSH  H
2466
2467 *     SET DEFAULTS IN AIO.***
2468
005.306 041 057 041 2469 LXI  H,AIO.DEV
005.311 001 003 000 2470 LXI  B,3
005.314 315 252 030 2471 CALL $MOVE          SET DEFALUT DEVICE
005.317 001 003 000 2472 LXI  B,3
005.322 041 072 041 2473 LXI  H,AIO.DIR+DIR.EXT
005.325 315 252 030 2474 CALL $MOVE          SET DEFAULT EXTENSION
005.330 341          2475 POP  H
005.331 315 272 010 2476 CALL $SOK           SKIP BLANKS
005.334 315 261 010 2477 CALL $MCU           MAP CHARACTER TO UPPER CASE

```

```

005.337 006 000 2478 MVI B,0
005.341 376 056 2479 CPI '!'
005.343 312 360 005 2480 JE DFD1 HAVE NAME
005.346 376 101 2481 CPI 'A'
005.350 332 076 006 2482 JC DFD4 NOT NAME
005.353 376 133 2483 CPI 'Z'
005.355 322 076 006 2484 JNC DFD4 NOT NAME
2485
2486 * HAVE ALPHA STRING. CRACK IT
2487
005.360 315 145 006 2488 DFD1 CALL DNT DECODE NEXT TOKEN
005.363 332 141 006 2489 JC DFD5 ERROR
005.366 376 072 2490 CPI '!'
005.370 302 023 006 2491 JNE DFD2 NOT DEVICE
2492
2493 * HAVE EXPLICIT DEVICE
2494
005.373 043 2495 INX H SKIP '!'
005.374 076 003 2496 MVI A,3
005.376 271 2497 CMP C
005.377 332 141 006 2498 JC DFD5 TOO MANY CHARACTERS
006.002 001 003 000 2499 LXI B,3
006.005 345 2500 PUSH H SAVE (HL)
006.006 041 057 041 2501 LXI H,AIO.DEV
006.011 315 252 030 2502 CALL $MOVE SET EXPLICIT DEVICE
006.014 341 2503 POP H
006.015 315 145 006 2504 CALL DNT DECODE NEXT TOKEN
006.020 332 141 006 2505 JC DFD5 ERROR
2506
2507 * DECODE NAME
2508
006.023 001 010 000 2509 DFD2 LXI B,8 (BC) = COUNT
006.026 345 2510 PUSH H SAVE TEXT ADDR
006.027 041 042 041 2511 LXI H,AIO.DIR+DIR.NAM
006.032 315 252 030 2512 CALL $MOVE MOVE IN NAME
006.035 341 2513 POP H
006.036 176 2514 MOV A,M (A) = DELIMITER
006.037 376 056 2515 CPI '!'
006.041 302 074 006 2516 JNE DFD3 NOT EXTENSION
2517
2518 * HAVE EXPLICIT EXTENSION
2519
006.044 043 2520 INX H
006.045 315 145 006 2521 CALL DNT
006.050 332 141 006 2522 JC DFD5 ERROR
006.053 076 003 2523 MVI A,3
006.055 271 2524 CMP C
006.056 332 141 006 2525 JC DFD5 TOO LONG
006.061 001 003 000 2526 LXI B,3
006.064 345 2527 PUSH H SAVE TEXT POINTER
006.065 041 072 041 2528 LXI H,AIO.DIR+DIR.EXT
006.070 315 252 030 2529 CALL $MOVE MOVE EXTENSION
006.073 341 2530 POP H
2531
2532 * DONE WITH NAME. MUST HAVE LEGIT DELIMITER
2533

```

```

006.074 006 001 2534 DFD3 MVI B,1 (B) = NAME PRESENT FLAG
2535
2536 * END OF NAME. EXIT
2537 * (B) = 0 IF NULL, (B) <> 0 IF NON-NULL
2538
006.076 115 2539 DFD4 MOV C,L (C) = #ADDR
006.077 315 272 010 2540 CALL #SOB SKIP BLANKS
006.102 171 2541 MOV A,C
006.103 225 2542 SUB L SEE IF ANY BLANKS
006.104 247 2543 ANA A 'Z' CLEAR IF BLANKS
006.105 176 2544 MOV A,M (A) = CHARACTER
006.106 314 237 010 2545 CZ #CFD CHECK FOR LEGAL DELIMITER
006.111 330 2546 RC ERROR
006.112 170 2547 MOV A,B
006.113 247 2548 ANA A SET 'Z' IF NULL
006.114 310 2549 RZ IF NULL FILE NAME
2550
006.115 072 062 041 2551 LDA AIO.DIR+DIR.NAM
006.120 376 000 2552 CPI 0
006.122 310 2553 RZ IF NULL FILE NAME
006.123 376 101 2554 CPI 'A'
006.125 332 141 006 2555 JC DFD5 NOT ALPHA CHAR.
006.130 376 133 2556 CPI 'Z'+1
006.132 322 141 006 2557 JNC DFD5 NOT ALPHA CHAR.
2558
006.135 076 001 2559 MVI A,1
006.137 247 2560 ANA A CLEAR 'Z' AND 'C' FLAG
006.140 311 2561 RET
2562
2563
2564 * ERROR
2565
006.141 076 007 2566 DFD5 MVI A,EC.IFN ILLEGAL FILE NAME
006.143 067 2567 STC
006.144 311 2568 RET

2570 ** DNT - DECODE NEXT TOKEN.
2571 *
2572 * DNT COPIES THE NEXT ALPHANUMERIC FIELD INTO A ZERO-FILLED WORK AREA,
2573 * THE CHARACTERS ARE ALL MAPPED TO UPPER CASE.
2574 *
2575 * ENTRY (HL) = TEXT POINTER
2576 * EXIT 'C' SET IF ERROR
2577 * 'C' CLEAR IF OK
2578 * (A) = DELIMITER CHARACTER
2579 * (HL) UPDATED TO DELIMITER CHARACTER
2580 * (DNTA) = STRING
2581 * (C) = LENGTH
2582 * (DE) = #DNTA
2583 * USES ALL
2584
006.145 021 234 006 2586 DNT LXI D,DNTA

```

```

006.150 016 011 2587      MOVI  C,9      (C) = SIZE OF DNTA
006.152 101 2588      MOV   B,C      (B) = MAX ALLOWED +1
006.153 257 2589      XRA   A
006.154 022 2590 DNT1  STAX  D      ZERO BUFFER
006.155 023 2591      INX   D
006.156 015 2592      DCR  C
006.157 302 154 006 2593      JNZ  DNT1
006.162 021 234 006 2594      LXI  D,DNTA
2595
2596 *      COPY CHARACTERS
2597
006.165 174 2598 DNT2  MOV   A,M
006.166 315 261 010 2599      CALL $MCU      MAP CHARACTER TO UPPER CASE
006.171 376 060 2600      CPI  '0'
006.173 332 227 006 2601      JC   DNT4      NOT ALPHANUMERIC
006.176 376 072 2602      CPI  '9'+1
006.200 332 215 006 2603      JC   DNT3      NUMERIC
006.203 376 101 2604      CPI  'A'
006.205 332 227 006 2605      JC   DNT4      DELIMITER
006.210 376 133 2606      CPI  'Z'+1
006.212 322 227 006 2607      JNC  DNT4      DELIMITER
2608
2609 *      HAVE GOOD CHARACTER
2610
006.215 022 2611 DNT3  STAX  D      STORE CHAR
006.216 023 2612      INX  D
006.217 043 2613      INX  H
006.220 014 2614      INR  C      COUNT
006.221 005 2615      DCR  B      LIMIT DECREMENT
006.222 302 165 006 2616      JNZ  DNT2      NOT OVERFLOW
2617
2618 *      OVERFLOW
2619
006.225 067 2620      STC      FLAG ERR
006.226 311 2621      RET
2622
2623 *      END OF STRING
2624
006.227 247 2625 DNT4  ANA   A      CLEAR 'C'
006.230 021 234 006 2626      LXI  D,DNTA      SET POINTER
006.233 311 2627      RET
2628
006.234 115 115 055 2629 DNTA  DB   'MM-DDD-YY'  WORK AREA (VALUE SHOWN IS MEANINGLESS
2630 *      SIMPLY A ? BYTE WORK AREA)
2631
2632 **     FCC - FETCH COMPLETE CHANNEL INFORMATION.
2633 *
2634 *     FCC FETCHES THE ENTIRE CHANNEL AREA INTO THE AIO. FIELD.
2635 *
2636 *     ENTRY (A) = CHANNEL #
2637 *     EXIT  'C' CLEAR IF OK
2638 *     (HL) = #AIO.FLG
2639 *     (A) = AIO.FLG

```

```

2640 *          'C' SET IF ERROR
2641 *          (A) = CODE
2642 *          USES  ALL
2643
2644
006.245 315 021 041 2645 FCC  CALL  S,FCI          FETCH CHANNEL INFO
006.250 330          2646      RC          ERROR
006.251 345          2647      PUSH  H          SAVE (HL)
006.252 001 027 000 2648      LXI  B,DIRELEN
006.255 052 116 041 2649      LHLD  AIO,CHA
006.260 021 021 000 2650      LXI  D,IOC.DIR-IOC.DDA
006.263 031          2651      DAD   D
006.264 353          2652      XCHG                    (HL) = ADDRESS OF IOC.DIR
006.265 041 062 041 2653      LXI  H,AIO.DIR
006.270 315 252 030 2654      CALL  $MOVE          COPY
006.273 341          2655      POP   H
006.274 176          2656      MOV   A,M          (A) = (AIO.FLG)
006.275 311          2657      RET

```

```

2659 **        FDB - FIND DIRECTORY FIRST BLOCK.
2660 *
2661 *          FDB RETURNS THE SECTOR NUMBER OF THE DIRECTORY'S FIRST BLOCK
2662 *          ON THIS DEVICE.
2663 *
2664 *
2665 *          ENTRY  AIO.UNI = UNIT NUMBER
2666 *                  DEVICE ASSUMED BY:
2667 *          EXIT  (HL) = SECTOR ADDRESS
2668 *          USES  A,F,D,E,H,L
2669
2670
006.276 052 053 041 2671 FDB  LHLD  AIO.DTA
006.301 021 012 000 2672      LXI  D,DEV.UNT
006.304 031          2673      DAD   D          HL = UNIT TABLE
006.305 072 061 041 2674      LDA  AIO.UNI
006.310 315 027 041 2675      CALL  S,GUP
006.313 315 234 030 2676      CALL  $INRL          DE = FIRST DIRECTORY SECTOR
006.316 005 000 2677      DW   UNT,DIS
006.320 353          2678      XCHG
006.321 311          2679      RET

```

```

2681 **        FGC - FREE GROUP CHAIN.
2682 *
2683 *          FGC UNCHAINS A LIST OF GROUPS AND ENTERS THEM INTO THE
2684 *          FREE CHAIN. THIS CAUSES THE SPACE TO BE RETURNED TO THE FREE
2685 *          POOL. NOTE THAT THE FREE CHAIN IS KEPT IN ORDER.
2686 *
2687 *          ENTRY  (A) = 1ST GROUP NUMBER
2688 *                  (HL) = GRP ADDRESS
2689 *          EXIT  NONE

```

```

2690 *      USES      A,F,D,E,H,L
2691
2692
006.322 043 2693 FGC      INX      H
006.323 066 001 2694      MVI      M,1      FLAG CHANGE
006.325 247 2695 FGC0     ANA      A
006.326 310 2696      RZ
006.327 157 2697      MOV      L,A      NO MORE IN CHAIN
006.330 176 2698      MOV      A,M      (A) = 2ND GROUP IN CHAIN TO FREE
006.331 365 2699      PUSH     PSW      SAVE FOR NEXT PASS
006.332 125 2700      MOV      D,L      (D) = # OF GROUP TO FREE
2701
2702 *      SCAN FREE CHAIN FOR THE RIGHT SPOT
2703
006.333 257 2704      XRA      A      (A) = 0
006.334 157 2705 FGC1     MOV      L,A      (L) = INDEX OF NEXT GROUP
006.335 176 2706      MOV      A,M      FOLLOW FREE CHAIN
006.336 247 2707      ANA      A
006.337 312 351 006 2708      JZ      FGC2     AT END, MUST PUT NEW ONE HERE
006.342 272 2709      CMP      D
006.343 314 013 041 2710      CE      S,FASER   GROUP IS ALREADY FREE!
006.346 332 334 006 2711      JC      FGC1     NOT FAR ENOUGH YET
2712
2713 *      FOUND THE PROPER SPOT.
2714 *
2715 *      (L) = PRECEDING GROUP #
2716 *      (D) = INDEX OF GROUP TO FREE
2717
006.351 162 2718 FGC2     MOV      M,D      POINT TO FREED BYTE
006.352 152 2719      MOV      L,D
006.353 167 2720      MOV      M,A      FREE POINTS TO NEXT
006.354 361 2721      POP      PSW
006.355 303 325 006 2722      JMP      FGC0     DO NEXT
2723
2724 **     FDE - FIND OPEN DIRECTORY ENTRY.
2725 *
2726 *      FDE IS CALLED TO LOCATE AN OPEN DIRECTORY ENTRY.
2727 *
2728 *      ENTRY      A10,XXX SETUP
2729 *      EXIT      'C' CLEAR, ENTRY FOUND
2730 *      SEC.SCR = DIRECTORY SECTOR
2731 *      (HL) = ADDRESS OF SEC.SCR LOCATION
2732 *      'C' SET, DIRECTORY FULL
2733 *      (A) = ERROR CODE
2734 *      USES      ALL
2735
2736
006.360 315 276 006 2737 FDE      CALL     FDB      FIND DIRECTORY BLOCK
2738
2739 *      READ ANOTHER SECTOR
2740
006.363 001 000 002 2741 FDE1     LXI      B,512
006.366 353 2742      XCHG

```

```

006.367 052 120 041 2743 LHLB S,SCR /79.11.GC/
006.372 353 2744 XCHG DE = SECTOR SCRATCH ADDRESS /79.11.GC/
006.373 315 256 031 2745 CALL DREAD READ SECTOR
006.376 330 2746 RC RETURN IF ERROR
2747
2748 * SCAN SECTOR FOR EMPTY SPOT
2749
006.377 052 120 041 2750 LHLB S,SCR /79.11.GC/
006.000 2751 ERRNZ DIS,ENT /79.11.GC/
2752
007.002 176 2753 FOE2 MOV A,H
007.003 247 2754 ANA A
007.004 370 2755 RM GET EMPTY SPOT
007.005 312 017 007 2756 JZ FOE3 END OF CONTENTS
007.010 021 027 000 2757 LXI D,DIRELEN
007.013 031 2758 DAD D
007.014 303 002 007 2759 JMP FOE2 TRY NEXT ENTRY
2760
2761 * SECTOR IS FULL UP, READ ANOTHER
2762
007.017 052 120 041 2763 FOE3 LHLB S,SCR /79.11.GC/
007.022 315 234 030 2764 CALL $INDL /79.11.GC/
007.025 376 001 2765 DW DIS,LNK /79.11.GC/
007.027 353 2766 XCHG HL = NEXT SECTOR OF DIR. /79.11.GC/
007.030 174 2767 MOV A,H
007.031 265 2768 ORA L
007.032 302 363 006 2769 JNZ FOE1 READ ANOTHER
2770
007.035 076 017 2771 MVI A,EC.DIF DIRECTORY FULL
007.037 067 2772 STC FLAG ERROR
007.040 311 2773 RET

2775 ** LDE - LOCATE DIRECTORY ENTRY.
2776 *
2777 * LDE LOCATES A DIRECTORY ENTRY CORRESPONDING TO THE AID,DIR,ENTRY.
2778 *
2779 * ENTRY (BC) = NUMBER OF CHARACTERS TO MATCH ON.
2780 * EXIT 'C' CLEAR IF FOUND
2781 * AID,DES,SETUP
2782 * (HL) = ADDRESS OF DIRECTORY ENTRY IN SECSR
2783 * 'C' SET IF NOT FOUND
2784 * (A) = CODE
2785 * USES ALL
2786
2787
007.041 001 015 000 2788 LDE LXI B,DIRINDL ENTRY FOR FULL NAME COMPARE
007.044 072 062 041 2789 LDE LDA AID,DIR+DIR,NAM
007.047 247 2790 ANA A
007.050 076 034 2791 MVI A,EC.FNR ASSUME FILE NAME MISSING
007.052 067 2792 STC
007.053 310 2793 RZ FILE NAME REQUIRED
007.054 315 276 006 2794 CALL FDB FIND DIRECTORY BLOCK SECTOR NUMBER
2795

```


LDE

```

2796 ** ENTRY FOR (HL) = SECTOR NUMBER TO START WITH
2797
007.057 305 2798 LDE., PUSH B SAVE COUNT
007.060 001 000 002 2799 LXI B,512
007.063 353 2800 XCHG /79.11.GC/
007.064 052 120 041 2801 LHLD S,SCR /79.11.GC/
007.067 353 2802 XCHG /79.11.GC/
007.070 042 055 041 2803 SHLD AIO,DES ASSUME WILL FIND IN THIS BLOCK
007.073 315 256 031 2804 CALL DREAD READ FRM DEVICE
007.076 301 2805 POP B RESTORE (BC)
007.077 330 2806 RC RETURN IF ERROR
2807
2808 * SCAN SECTOR FOR INFO
2809
007.100 052 120 041 2810 LHLD S,SCR /79.11.GC/
000.000 2811 ERRNZ DIS,ENT /79.11.GC/
2812
2813 * COMPARE
2814
007.103 021 062 041 2815 LDE3 LXI D,AIO,DIR
007.106 176 2816 MOV A,M
000.000 2817 ERRNZ DF,EMP-3770
007.107 074 2818 INR A
007.110 312 127 007 2819 JZ LDE5 EMPTY SPOT
000.000 2820 ERRNZ DF,CLR-3760
007.113 074 2821 INR A
007.114 312 161 007 2822 JZ LDE6 NO MORE FILES IN DIRECTORY
007.117 305 2823 PUSH B SAVE COPY OF (BC)
007.120 345 2824 PUSH H SAVE ADDRESS
007.121 315 060 030 2825 CALL %COMP COMPARE
007.124 341 2826 POP H
007.125 301 2827 POP B (BC) = COMPARE COUNT
007.126 310 2828 RE GOT MATCH
007.127 021 027 000 2829 LDE5 LXI D,DIRELEN MISSED, SCAN TO NEXT ENTRY
007.132 031 2830 DAD D
007.133 176 2831 MOV A,M
007.134 247 2832 ANA A
007.135 302 103 007 2833 JNZ LDE3 MORE IN SECTOR
2834
2835 * DIDNT FIND IT IN THIS SECTOR, TRY NEXT
2836
007.140 052 120 041 2837 LHLD S,SCR
007.143 315 234 030 2838 CALL %INDL
007.146 376 001 2839 DW DIS,LNK
007.150 353 2840 XCHG HL = NEXT DIRECTORY SECTOR
007.151 042 055 041 2841 SHLD AIO,DES SET POSSIBLE SECTOR INDEX
007.154 174 2842 MOV A,H
007.155 265 2843 ORA L
007.156 302 057 007 2844 JNZ LDE., HAVE MORE SECTORS
007.161 076 014 2845 LDE6 MVI A,EC,FNF FILE NOT FOUND
007.163 067 2846 STC
007.164 311 2847 RET

```

```

2849 **      LDI - LOCATE DEVICE INFORMATION.
2850 *
2851 *      LDI FINDS A DEVICE IN THE DEVICE TABLE; ENTERS THE PROPER
2852 *      INFO IN THE AIO TABLE, AND LOADS THE DEVICE DRIVER, IF NECESSARY.
2853 *
2854 *      ENTRY  AIO.DEV = DEVICE CODE
2855 *      EXIT   'C' CLEAR IF OK
2856 *      AIO.DDA, AIO.FLG, AIO.SPG, AIO.GRT AND AIO.DTA SETUP
2857 *      'C' SET IF ERROR
2858 *      (A) = CODE
2859 *      USES  ALL
2860
2861
007.165 052 354 040 2862 LDI  LHL  S.DFWA
007.170 353          2863 XCHG          (DEV) = FWA DEVICE LIST
2864
007.171 353          2865 LDI1 XCHG
007.172 042 053 041 2866 SHLD  AIO.DTA  SET DEVICE TABLE ADDRESS
007.175 353          2867 XCHG
007.176 032          2868 LDAX  D
007.177 247          2869 ANA  A
007.200 076 015     2870 MVI  A,EC.UND
007.202 067          2871 STC
000.000            2872 ERRNZ  DV.EL  END OF LIST
007.203 310          2873 RZ      UNKNOWN DEVICE
007.204 052 057 041 2874 LHL  AIO.DEV
007.207 032          2875 LDAX  D
007.210 023          2876 INX  D
007.211 376 001     2877 CFI  DV.NU
007.213 312 227 007 2878 JE  LD12  DEVICE ENTRY NOT USED
007.216 275          2879 CMP  L    COMPARE 1ST CHAR
007.217 302 227 007 2880 JNE  LD12  NOT THIS ONE
007.222 032          2881 LDAX  D
007.223 274          2882 CMP  H
007.224 312 237 007 2883 JE  LD13  GOT DEVICE
2884
2885 *      MISSED DEVICE. TRY NEXT ENTRY
2886
007.227 041 016 000 2887 LD12 LXI  H,DEVLEN-1
007.232 031          2888 DAD  H
007.233 353          2889 XCHG
007.234 303 171 007 2890 JMP  LD11  TRY AGAIN
2891
2892 *      GOT DEVICE. SEE IF WE MUST LOAD IT
2893
007.237 023          2894 LD13 INX  D
000.000            2895 ERRNZ  DEV.RES-2
007.240 032          2896 LDAX  D
007.241 346 001     2897 ANI  DR.IM
007.243 314 020 010 2898 CZ  RDL  REQUEST DEVICE DRIVER, IF NOT PRESENT
007.246 330          2899 RC      ERROR GETTING DRIVER
007.247 023          2900 INX  D
000.000            2901 ERRNZ  DEV.JMP-3  (HL) = #DEV.JMP
007.250 353          2902 XCHG
007.251 042 041 041 2903 SHLD  AIO.DDA  SET DEVICE DRIVER LINK ADDRESS
2904

```

```

007.254 021 004 000 2905 LXI D,DEV.SPG-DEV.JMP
007.257 031 2906 DAD D (HL) = DEV.SPG
007.260 176 2907 MOV A,M
007.261 062 046 041 2908 STA AIO.SPG SET UP SECTORS/GROUP
2909
2910 * CHECK LEGALITY OF UNIT NUMBER
2911
007.264 021 002 000 2912 LXI D,DEV.MNU-DEV.SPG
007.267 031 2913 DAD D
007.270 072 061 041 2914 LDA AIO.UNI
007.273 276 2915 CMP M 'C' SET IF AIO.UNI < DEV.MNU
007.274 077 2916 CMC
007.275 076 033 2917 MVI A,EC.UUN ASSUME UNKNOWN UNIT NUMBER
007.277 330 2918 RC 'C' IF AIO.UNI >= DEV.MNU
2919
007.300 072 061 041 2920 LIA AIO.UNI
007.303 107 2921 MOV B,A
007.304 257 2922 XRA A
007.305 315 220 010 2923 CALL BITS A = MASK TO CHECK IF UNIT IS MOUNTED
007.310 053 2924 DCX H
000.000 2925 ERRNZ DEV.MNU-DEV.MUM-1
007.311 246 2926 ANA M
007.312 076 033 2927 MVI A,EC.UUN ASSUME UNKNOWN UNIT NUMBER
007.314 067 2928 STC
007.315 310 2929 RZ UNIT WAS NOT MOUNTED
2930
2931 * SET UP AIO.FLG, AIO.GRT
2932
007.316 021 002 000 2933 LXI D,DEV.UNT-DEV.MUM
007.321 031 2934 DAD D
007.322 072 061 041 2935 LDA AIO.UNI
007.325 315 027 041 2936 CALL S.GUP
007.330 315 315 010 2937 CALL *INDB
007.333 000 000 2938 DW UNT.FLG
007.335 062 043 041 2939 STA AIO.FLG
007.340 315 234 030 2940 CALL *INDL
007.343 001 000 2941 DW UNT.GRT
007.345 353 2942 XCHG
007.346 042 044 041 2943 SHLD AIO.GRT
007.351 247 2944 ANA A CLEAR CARRY
007.352 311 2945 RET

2947 ** RBF - RETURN BLOCKS TO FREE POOL.
2948 *
2949 * RBF RETURNS THE BLOCKS BELONGING TO A FILE TO THE
2950 * DISKS FREE POOL.
2951 *
2952 * ENTRY (HL) = ADDRESS OF ENTRY IN DIRECTORY SECTOR
2953 * EXIT NONE
2954 * USES A,F,D,E
2955
007.353 345 2957 RBF PUSH H SAVE ADDRESS OF ENTRY

```

OVL SUBROUTINES

RBF

14:15:23 16-MAY-80

```

007.354 021 020 000 2958 LXI D,DIR.FGN
007.357 031 2959 DAD D
007.360 178 2960 MOV A,M (A) = FIRST GROUP NUMBER
007.361 052 044 041 2961 LHLD AIG.GRT
007.364 315 322 006 2962 CALL FGC FREE GROUP CHAIN
007.367 341 2963 POP H (HL) = POINTER TO DIRECTORY ENTRY
007.370 311 2964 RET
    
```

```

2966 ** LFD - LOCATE FILE IN DIRECTORY.
2967 *
2968 * LFD IS CALLED TO LOCATE A NAMED FILE IN IT'S
2969 * DEVICES DIRECTORY
2970 *
2971 * ENTRY (DE) = DEFAULT BLOCK
2972 * (HL) = FILE NAME ADDRESS
2973 * EXIT 'C' CLEAR, GOT ENTRY
2974 * (HL) = ADDRESS OF FILE IN DIRECTORY BLOCK (IN SECSR)
2975 * 'C' SET, ERROR
2976 * (A) = ERROR CODE
2977 * USES A,F,D,E,H,L
2978
2979
    
```

```

007.371 305 2980 LFD PUSH B
007.372 315 377 007 2981 CALL LFD1 FIND IT
007.375 301 2982 POP B
007.376 311 2983 RET
2984
007.377 315 234 005 2985 LFDI CALL DFA DECODE FILENAME INTO AIG.
010.002 330 2986 RC ERROR
000.000 2987 ERRNZ FT.DD-1
010.003 037 2988 RAR
010.004 077 2989 CMC 'C' SET IF NOT DIRECTORY
010.005 076 005 2990 MVI A,ED.DNS
010.007 330 2991 RC DEVICE NOT SUITABLE
010.010 315 212 005 2992 CALL CFP CHECK FOR POSSESSION
010.013 330 2993 RC CANT DO IT NOW
010.014 315 041 007 2994 CALL LDE. LOCATE DIRECTORY ENTRY
010.017 311 2995 RET FOUND OR NOT
    
```

```

2997 ** RDL - REQUEST DEVICE DRIVER.
2998 *
2999 * RDL SETS A REQUEST FOR THE LOADING OF A DEVICE DRIVER.
3000 *
3001 * THE DRIVER IS LOADED INTO MEMORY JUST BELOW #9.SYSM*.
3002 *
3003 * ENTRY (DE) = #DEV.RES
3004 * EXIT 'C' SET IF ERROR
3005 * (A) = ERROR CODE
3006 * 'C' CLEAR IF OK
3007 * DEVLST POINTERS SET
    
```

RDL

```

3008 *      USES      A,F,B,C,H,L
3009
3010
010.020 325 3011 RDL  PUSH  D      SAVE (DE)
010.021 353 3012      XCHG      (HL) = #DEV.RES
010.022 042 366 040 3013      SHLD  S,DDTA      SET DEVICE TABLE ADDRESS (OF DEV.RES)
010.025 021 012 000 3014      LXI   D,DEV.DVL-DEV.RES
010.030 031 3015      DAD   D      (HL) = ADDRESS OF LENGTH
010.031 136 3016      MOV   E,M
010.032 043 3017      INX   H
010.033 126 3018      MOV   D,M      (DE) = LEN OF DRIVER
000.000 3019      ERKZ  DEV,DVG-DEV,DVL-2
010.034 043 3020      INX   H
010.035 176 3021      MOV   A,M      (A) = (DEV,DVG)
010.036 062 364 040 3022      STA  S,DDGRP      SET GROUP FOR FILE
010.041 052 320 040 3023      LHL  S,SYSM
010.044 175 3024      MOV   A,L      COMPUTE LOAD ADDRESS
010.045 223 3025      SUB  E
010.046 157 3026      MOV   L,A
010.047 174 3027      MOV   A,H
010.050 232 3028      SBB  D
010.051 147 3029      MOV   H,A      (HL) = LOAD ADDRESS
010.052 076 010 3030      MVI  A,EC.NRD      NO ROOM FOR DRIVER
010.054 332 107 010 3031      JC   RDL1      ERROR
3032
3033 *      SEE IF THIS IS ABOVE THE USER HIMEM.
3034
010.057 353 3035      XCHG      (DE) = NEW S,SYSM
010.060 042 362 040 3036      SHLD  S,DDLEN      SET LENGTH OF LOAD
010.063 052 322 040 3037      LHL  S,USRM
010.066 043 3038      INX   H
010.067 173 3039      MOV   A,E
010.070 225 3040      SUB  L
010.071 172 3041      MOV   A,D
010.072 234 3042      SBB  H
010.073 076 010 3043      MVI  A,EC.NRD      NO ROOM
010.075 332 107 010 3044      JC   RDL1      (HL) = NEW S,SYSM
010.100 353 3045      XCHG      (HL) = NEW S,SYSM
010.101 042 320 040 3046      SHLD  S,SYSM
010.104 042 360 040 3047      SHLD  S,DDLDA      SET LOAD ADDR
010.107 321 3048 RDL1  POP  D      RESTORE (DE)
010.110 311 3049      RET

```

```

3051 **     SGT - SAVE GROUP TABLE.
3052 *
3053 *     SGT UPDATES THE GROUP TABLE ONTO THE DISK, IF IT HAS BEEN
3054 *     ALTERED.
3055 *
3056 *     ENTRY  AIO,GRT = TABLE ADDRESS
3057 *     EXIT  AIO,GRT FLAGGED UPDATED
3058 *     USES  ALL
3059
3060

```

010.111	052 044 041	3061	SGT	LHLD	AIO.GRT	
010.114	043	3062		INX	H	
010.115	176	3063		MOU	A,M	
010.116	247	3064		ANA	A	
010.117	310	3065		RZ		NOT CHANGED
		3066				
010.120	066 000	3067		MUI	H,0	FLAG NOT CHANGED
010.122	353	3068		XCHG		
010.123	033	3069		DCX	D	(DE) = GRT ADDRESS
		3070				
010.124	052 053 041	3071		LHLD	AIO.DTA	(HL) = DEVICE TABLE ADDRESS
010.127	001 012 000	3072		LXI	B,DEV.UNT	
010.132	011	3073		DAD	B	HL = ADDRESS OF DEV.UNT TABLE
010.133	072 061 041	3074		LDA	AIO.UNI	
010.136	315 027 041	3075		CALL	S.GUP	
010.141	001 003 000	3076		LXI	B,UNT.GTS	
010.144	011	3077		DAD	B	HL = ADDRESS OF DEV.UNT ENTRY
010.145	315 211 030	3078		CALL	%HLIHL	HL = ADDRESS OF GRT
		3079				
010.150	001 000 001	3080		LXI	B,256	SET I/O COUNT
010.153	303 253 031	3081		JMP	DWRITE	UPDATE GRT

3083 ** UDE - UPDATE DIRECTORY ENTRY.
 3084 *
 3085 * UDE UPDATES THE DIRECTORY ENTRY IN THE SECTOR BUFFER
 3086 * FROM AIO,XXX
 3087 *
 3088 * ENTRY SECSCR = DIRECTORY SECTOR
 3089 * (HL) = ADDRESS OF ENTRY IN SECSCR
 3090 * EXIT NONE
 3091 * USES ALL

		3092				
		3093				
010.156	345	3094	UDE	PUSH	H	SAVE (HL)
010.157	052 051 041	3095		LHLD	AIO.LGN	(L) = AIO.LGN
000.000		3096		ERRNZ	AIO.LSI-AIO.LGN-1	(H) = AIO.LSI
010.162	042 103 041	3097		SHLD	AIO.DIR+DIR.LGN	SET DIR.LGN
000.000		3098		ERRNZ	DIR.LSI-DIR.LGN-1	SET DIR.LSI
010.165	341	3099		POP	H	RESTORE (HL)
		3100				
010.166	001 027 000	3101		LXI	B,DIRELEN	
010.171	021 062 041	3102		LXI	B,AIO.DIR	
010.174	303 252 030	3103		JMP	%MOVE	UPDATE IN DIR


```

010.220      3125      XTEXT  BITS
.....
3127X **      BITS - BIT SET
3128X *
3129X *      BITS SETS THE SPECIFIED BIT IN THE ACCUMULATOR.
3130X *
3131X *      ENTRY: A = ORIGINAL A
3132X *      B = NUMBER OF BIT TO SET ( 7=HIGH, ..., 0=LOW )
3133X *
3134X *      EXIT: A = ORIGINAL A WITH BIT(B) SET
3135X *
3136X *      USES: PSW
3137X *
3138X
010.220 305      3139X BITS  PUSH  B
3140X
010.221 365      3141X      PUSH  PSW
010.222 076 200  3142X      MVI   A,10000000B
010.224 004      3143X      INR   B
010.225 007      3144X BITS1  RLC
010.226 005      3145X      DCR   B
010.227 302 225 010 3146X      JNZ   BITS1
3147X
010.232 117      3148X      MOV   C,A
010.233 361      3149X      POP   PSW
010.234 261      3150X      ORA   C
3151X
010.235 301      3152X      POP   BC
010.236 311      3153X      RET
010.237          3154      XTEXT  CFB
.....
3156X **      $CFD - CHECK FILE DELIMITER,
3157X *
3158X *      $CFD CHECKS AN ASCII CHARACTER TO SEE IF IT IS A LEGAL FILE
3159X *      NAME DELIMITER. LEGAL DELIMITERS ARE
3160X *
3161X *      , = / <BLANK> <00>
3162X *
3163X *      ENTRY (A) = CHARACTER
3164X *      EXIT  'C' CLEAR IF OK
3165X *      'E' SET IF ERROR
3166X *      (A) = ERROR CODE
3167X *      USES  A,F
3168X
3169X
010.237 247      3170X $CFD  ANA   A
010.240 310      3171X      RZ
010.241 376 054  3172X      CPI   ','
010.243 310      3173X      RE
010.244 376 075  3174X      CPI   '='

```



```

010.246 310 3175X RE IS =
010.247 376 057 3176X CPI ' '
010.251 310 3177X RE IS /
010.252 376 040 3178X CPI ' '
010.254 310 3179X RE IS ' '
010.255 076 007 3180X MVI A,EC.IFN ILLEGAL FILE NAME
010.257 067 3181X STC
010.260 311 3182X RET
010.261 3183 XTEXT MCU

```

```

3185X ** MCU - MAP LOWER CASE TO UPPER CASE.
3186X *
3187X * MCU MAPS A LOWER CASE ALPHABETIC TO UPPER
3188X * CASE.
3189X *
3190X * ENTRY (A) = CHARACTER
3191X * EXIT (A) = CHARACTER RESULT
3192X * USES A,F
3193X
3194X

```

```

010.261 376 141 3195X $MCU CPI 'a'
010.263 330 3196X RC NOT LOWER CASE
010.264 376 173 3197X CPI 'z'+1
010.266 320 3198X RNC NOT LOWER CASE
010.267 326 040 3199X SUI 'a'-'A'
010.271 311 3200X RET
010.272 3201 XTEXT SOB

```

```

3203X ** $SOB - SKIP OVER BLANKS.
3204X *
3205X * $SOB IS CALLED TO SKIP AN ARBITRARILY LONG STRING OF BLANKS AND TABS.
3206X *
3207X * ENTRY (HL) = FWA OF (POSSIBLE) BLANK STRING
3208X * EXIT (HL) = LWA+1 OF BLANK STRING (UNCHANGED IF NO BLANKS)
3209X * (A) = FIRST NON-BLANK, NON-TAB CHARACTER EEN
3210X * USES A,F,H,L
3211X
3212X

```

```

010.272 053 3213X $SOB DCX H PRE-DECREMENT
010.273 043 3214X $SOB1 INX H
010.274 176 3215X MOV A,M
010.275 376 040 3216X CPI ' '
010.277 312 273 010 3217X JE $SOB1 GOT BLANK
010.302 376 011 3218X CPI TAB
010.304 312 273 010 3219X JE $SOB1 GOT TAB
010.307 311 3220X RET
010.310 3221 XTEXT ILBEHL

```

```

3223X **      ILDEHL - INDEXED LOAD OF DE FROM HL
3224X *
3225X *      'DE' GET THE FULL WORD VALUE POINTED TO BY 'HL', AND 'HL' IS
3226X *      INCREMENTED BY TWO.
3227X *
3228X *      ENTRY:  HL      = ADDRESS OF FULL WORD VALUE
3229X *
3230X *      EXIT:   DE      = (HL)
3231X *      HL      = 'HL' + 2
3232X *
3233X *      USES:   DE
3234X *
3235X
010.310 136    3236X ILDEHL MOV    E,M
010.311 043    3237X      INX    H
010.312 126    3238X      MOV    D,M
010.313 043    3239X      INX    H
010.314 311    3240X      RET
010.315        3241X XTEXT  INDL

```

```

3243X **      $INDL - INDEXED LOAD.
3244X *
3245X *      $INDL LOADS DE WITH THE TWO BYTES AT (HL)+DISPLACMENT
3246X *
3247X *      THIS ACTS AS AN INDEXED FULL WORD LOAD.
3248X *
3249X *      (DE) = ( (HL) + DSPLACEMENT )
3250X *
3251X *      ENTRY  ((RET)) = DISPLACMENT (FULL WORD)
3252X *      (HL) = TABLE ADDRESS
3253X *      EXIT   TO (RET+2)
3254X *      USES   A,F,D,E
3255X
3256X
030.234        3257X $INDL  EQU    30234A      IN H17 ROM
010.315        3258X XTEXT  INDX

```

```

3260X **      $INDLB - INDEXED LOAD BYTE
3261X *
3262X *      BYTE INDEXED LOAD PRIMITIVE
3263X *
3264X *      ENTRY:  HL      = BASE ADDRESS
3265X *      (RET)   = FULL WORD RELOCATION
3266X *
3267X *      EXIT:   A        = ( HL + (RET) )
3268X *
3269X *      USES:   A
3270X *
3271X
010.315 353    3272X $INDLB XCHG      DE = BASE

```

COMMON DECKS

*INDLE

14:15:42 14-MAY-80

```

010.316 343      3273X      XTHL
010.317 325      3274X      PUSH    D
010.320 305      3275X      PUSH    B
                  3276X
010.321 116      3277X      MOV     C,M
010.322 043      3278X      INX    H
010.323 106      3279X      MOV     B,M      BC = OFFSET
010.324 043      3280X      INX    H          HL = .RET.
                  3281X
010.325 353      3282X      XCHG
010.326 011      3283X      DAD    B          HL = BASE + OFFSET
010.327 176      3284X      MOV     A,M      A = ( BASE + OFFSET )
010.330 353      3285X      XCHG            HL = .RET.
                  3286X
010.331 301      3287X      POP    B          RESTORE .BC.
010.332 321      3288X      POP    D          RESTORE BASE
010.333 343      3289X      XTHL            HL = .DE. ; (SP) = .RET.
010.334 353      3290X      XCHG            DE = .DE. ; HL = BASE
010.335 311      3291X      RET
    
```

```

3293X **      *INDS - INDEXED STORE
3294X *
3295X *      INDEXED STORE PRIMITIVE.
3296X *
3297X *      ENTRY: HL = BASE ADDRESS
3298X *          DE = VALUE TO STORE
3299X *
3300X *      EXIT: ( HL + (RET) ) = DE
3301X *
3302X *      USES: NONE
3303X *
3304X
    
```

```

010.336 315 020 011 3305X *INDS CALL    XCHGBC
010.341 343      3306X      XTHL            SAVE .BC.
010.342 325      3307X      PUSH    D
010.343 315 310 010 3308X      CALL    ILDEHL  DE = OFFSET
010.346 315 020 011 3309X      CALL    XCHGBC  BC = .RET.
010.351 353      3310X      XCHG            DE = BASE ; HL = OFFSET
010.352 031      3311X      DAD    B          HL = BASE + OFFSET
010.353 353      3312X      XCHG
010.354 343      3313X      XTHL            SAVE BASE
010.355 353      3314X      XCHG            DE = VALUE
010.356 315 013 011 3315X      CALL    ISDEHL
010.361 341      3316X      POP    H          HL = BASE
010.362 315 020 011 3317X      CALL    XCHGBC
010.365 343      3318X      XTHL            RESTORE .BC.
010.366 315 020 011 3319X      CALL    XCHGBC
010.371 311      3320X      RET
    
```

```

3322X ** *INDSB - INDEXED BYTE STORE
3323X *
3324X * INDEXED BYTE STORE.
3325X *
3326X * ENTRY: A = VALUE TO STORE
3327X * HL = BASE ADDRESS
3328X * (RET) = OFFSET
3329X *
3330X * EXIT: NONE
3331X *
3332X * USES: PSW
3333X *
3334X
010.372 353 3335X *INDSB XCHG DE = BASE
010.373 343 3336X XTHL SAVE .DE.
010.374 325 3337X PUSH D SAVE .BASE
010.375 305 3338X PUSH B SAVE .BC.
3339X
010.376 116 3340X MOV C,M
010.377 043 3341X INX H
011.000 106 3342X MOV B,M BC = OFFSET
011.001 043 3343X INX H HL = .RET.
3344X
011.002 353 3345X XCHG HL = BASE
011.003 011 3346X DAD B HL = BASE + OFFSET
011.004 167 3347X MOV M,A ( BASE + OFFSET ) = A
011.005 353 3348X XCHG
3349X
011.006 301 3350X POP B RESTORE .BC.
011.007 321 3351X POP D RESTORE .BASE
011.010 343 3352X XTHL HL = .DE. ; (SP) = .RET.
011.011 353 3353X XCHG DE = .DE. ; HL = BASE
011.012 311 3354X RET
011.013 3355 XTEXT ISDEHL

```

```

3357X ** ISDEHL - INDEXED STORE OF DE AT HL
3358X *
3359X * STORE 'DE' AT THE ADDRESS POINTED TO BY 'HL', AND INCREMENT 'HL'
3360X * BY 2.
3361X *
3362X * ENTRY: DE = VALUE
3363X * HL = ADDRESS OF VALUE
3364X *
3365X * EXIT: (HL) = DE
3366X * HL = HL + 2
3367X *
3368X * USES: HL
3369X *
3370X
011.013 163 3371X ISDEHL MOV M,E
011.014 043 3372X INX H
011.015 162 3373X MOV M,D
011.016 043 3374X INX H

```

011.017 311
011.020

3375X RET
3376 XTEXT WER

3378X ** \$WER - WRITE ENABLE RAM.
3379X *
3380X * \$WER IS CALLED TO ENABLE WRITING TO THE H17 CONTROLLER'S
3381X * RAM AREA.
3382X *
3383X * ENTRY NONE
3384X * EXIT NONE
3385X * USES NONE
3386X
3387X

031.241 3388X \$WER EQU 31241A IN H17 ROM

3390X ** \$WDR - WRITE DISABLE RAM.
3391X *
3392X * \$WDR IS CALLED TO DISABLE WRITING TO THE H17 CONTROLLER'S
3393X * RAM AREA.
3394X *
3395X * ENTRY NONE
3396X * EXIT NONE
3397X * USES NONE
3398X
3399X

031.222 3400X \$WDR EQU 31222A IN H17 ROM
011.020 3401 XTEXT DADA2

3403X ** \$DADA - ADD (O,A) TO (H,L)
3404X *
3405X * ENTRY NONE
3406X * EXIT (HL) = (HL) + (OA)
3407X * USES O,F,H,L
3408X
3409X

030.101 3410X \$DADA EQU 30101A IN H17 ROM
011.020 3411 XTEXT CHL

```

3413X **      *CHL - COMPLEMENT (HL),
3414X *
3415X *      (HL) = -(HL)          TWO'S COMPLEMENT
3416X *
3417X *      ENTRY  NONE
3418X *      EXIT   NONE
3419X *      USES   A,F,H,L
3420X
3421X
030.224      3422X *CHL  EQU   30224A      IN H17 ROM
011.020      3423      XTEXT  MOVE

3425X **      *MOVE - MOVE DATA
3426X *
3427X *      *MOVE MOVES A BLOCK OF BYTES TO A NEW MEMORY ADDRESS.
3428X *      IF THE MOVE IS TO A LOWER ADDRESS, THE BYTES ARE MOVED FROM
3429X *      FIRST TO LAST.
3430X *
3431X *      IF THE MOVE IS TO A HIGHER ADDRESS, THE BYTES ARE MOVED FROM
3432X *      LAST TO FIRST.
3433X *
3434X *      THIS IS DONE SO THAT AN OVERLAPED MOVE WILL NOT 'RIPPLE'.
3435X *
3436X *      ENTRY  (BC) = COUNT
3437X *             (DE) = FROM
3438X *             (HL) = TO
3439X *      EXIT  MOVED
3440X *             (DE) = ADDRESS OF NEXT FROM BYTE
3441X *             (HL) = ADDRESS OF NEXT *TD* BYTE
3442X *             'C' CLEAR
3443X *      USES  ALL
3444X
3445X
030.252      3446X *MOVE  EQU   30252A      IN H17 ROM
011.020      3447      XTEXT  XCHGBC

3449X **      XCHGBC - XCHG BC
3450X *
3451X *      EXCHANGE THE 'BC' REGISTER PAIR WITH THE 'HL' REGISTER PAIR.
3452X *
3453X *      ENTRY:  BC      = ORIGINAL BC
3454X *             HL      = ORIGINAL HL
3455X *
3456X *      EXIT:   BC      = ORIGINAL HL
3457X *             HL      = ORIGINAL BC
3458X *
3459X *      USES:   BC,HL
3460X *
3461X
011.020 365  3462X XCHGBC PUSH  PSW

```

```

011.021 170      3463X      MOV      A,B
011.022 104      3464X      MOV      B,H
011.023 147      3465X      MOV      H,A
011.024 171      3466X      MOV      A,C
011.025 115      3467X      MOV      C,L
011.026 157      3468X      MOV      L,A
011.027 361      3469X      POP      PSW
011.030 311      3470X      RET
011.031          3471      XTEXT   ZEROS

```

```

          3473X **      B CONSTANT ZERO BYTES.
          3474X
031.320          3475X $ZEROS EQU      31320A      IN H17 ROM
011.031          3476      XTEXT   ZERO

```

```

          3478X **      $ZERO - ZERO MEMORY.
          3479X *
          3480X *      $ZERO ZEROS A BLOCK OF MEMORY.
          3481X *
          3482X *      ENTRY   (HL) = ADDRESS
          3483X *              (B) = COUNT
          3484X *      EXIT    (A) = 0
          3485X *      USES   A,B,F,H,L
          3486X
          3487X
031.212          3488X $ZERO EQU      31212A      IN H17 ROM
011.031          3489      XTEXT   UDD

```

```

          3491X **      $UDD - UNPACK DECIMAL DIGITS.
          3492X *
          3493X *      UDD CONVERTS A 16 BIT VALUE INTO A SPECIFIED NUMBER OF
          3494X *      DECIMAL DIGITS. THE RESULT IS ZERO FILLED.
          3495X *
          3496X *      ENTRY   (B,C) = ADDRESS VALUE
          3497X *              (A) = DIGIT COUNT
          3498X *              (H,L) = MEMORY ADDRESS
          3499X *      EXIT    (HL) = (HL) + (A)
          3500X *      USES   ALL
          3501X
          3502X
031.157          3503X $UDD EQU      31157A      IN H17 ROM
011.031          3504      XTEXT   COMP

```

```

3506X ** $COMP - COMPARE TWO CHARACTER STRINGS.
3507X *
3508X * $COMP COMPARES TWO BYTE STRINGS.
3509X *
3510X * ENTRY (C) = COMPARE COUNT
3511X * (DE) = FWA OF STRING #1
3512X * (HL) = FWA OF STRING #2
3513X * EXIT 'Z' CLEAR, IS MIS-MATCH
3514X * (C) = LENGTH REMAINING
3515X * (DE) = ADDRESS OF MISMATCH IN STRING#1
3516X * (HL) = ADDRESS OF MISMATCH IN STRING #2
3517X * 'C' SET, HAVE MATCH
3518X * (C) = 0
3519X * (DE) = (DE) + (OC)
3520X * (HL) = (HL) + (OC)
3521X * USES A>F,C>D,E>H,L
3522X
3523X
030.060 3524X $CDMP EQU 30060A IN H17 ROM
011.031 3525 XTEXT HLIHL

```

```

3527X ** $HLIHL - LOAD HL INDIRECT THROUGH HL.
3528X *
3529X * (HL) = ((HL))
3530X *
3531X * ENTRY NONE
3532X * EXIT NONE
3533X * USES A>H,L
3534X
030.211 3535X $HLIHL EQU 30211A IN H17 ROM

```

```

3537 *****
3538 *
3539 * NOTE: THIS OVERLAY, AND ITS RELOCATION TABLE MUST USE LESS THAN
3540 * SB.DVMX BYTES.
3541 *
3542 * THE SIZE OF THE RELOCATION TABLE CANNOT BE KNOWN AT ASSEMBLY TIME,
3543 * SO THE '450' FIGURE USED BELOW IS APPROX, AND MUST BE WATCHED!
3544 *
3545 *****
3546
011.031 114 122 3547 DW 'RL' DUMMY AREA FOR UNWANTED RELOCATIONS
011.033 000 000 000 3548 DB 0,0,0,0,0,0,0,0 PATCH AREA
011.043 000 000 000 3549 DB 0,0,0,0,0,0,0,0
011.053 000 000 000 3550 DB 0,0,0,0,0,0,0,0
011.063 000 000 000 3551 DB 0,0,0,0,0,0,0,0
011.073 000 000 000 3552 DB 0,0,0,0,0,0,0,0
011.103 000 000 000 3553 DB 0,0,0,0,0,0,0,0
3554
376.333 3555 ERRPL *+400-SB.DVMX TOO LARGE

```


011.113 012 000 024
000 027 000
046 000 050
000 052 000
054 000 056
000 060 000
062 000 064
000 066 000
070 000 072
000 074 000
076 000 100
000 102 000
104 000 106
000 110 000
112 000 116
000 121 000
151 000 170
000 173 000
204 000 210
000 217 000
222 000 225
000 244 000
247 000 253
000 256 000
342 000 345
000 350 000
371 000 375
000 001 001
022 001 025
001 031 001
051 001 056
001 062 001
065 001 071
001 117 001
127 001 133
001 214 001
225 001 232
001 240 001
243 001 246
001 263 001
273 001 276
001 301 001
304 001 307
001 331 001
334 001 340
001 360 001
014 002 017
002 023 002
026 002 054
002 075 002
124 002 130

3556
3557
3558
3559
3560
3561

LON B

END

002 163 002
174 002 212
002 221 002
234 002 243
002 303 002
306 002 353
002 370 002
374 002 035
003 041 003
050 003 063
003 244 003
251 003 325
003 011 004
030 004 037
004 046 004
056 004 061
004 064 004
075 004 100
004 104 004
110 004 121
004 124 004
132 004 137
004 142 004
154 004 157
004 166 004
171 004 264
004 332 004
346 004 355
004 015 005
020 005 065
005 121 005
142 005 153
005 213 005
235 005 240
005 244 005
247 005 253
005 265 005
275 005 332
005 335 005
344 005 351
005 356 005
361 005 364
005 371 005
000 006 016
006 021 006
042 006 046
006 051 006
057 006 100
006 107 006
126 006 133
006 146 006
160 006 163
006 167 006
174 006 201
006 206 006
213 006 223

006 231 006
340 006 347
006 356 006
361 006 006
007 015 007
033 007 055
007 111 007
115 007 136
007 157 007
214 007 220
007 225 007
235 007 244
007 306 007
331 007 365
007 373 007
000 010 011
010 015 010
055 010 076
010 230 010
300 010 305
010 337 010
344 010 347
010 357 010
363 010 367
010 000 000

ASSEMBLY COMPLETE
3561 STATEMENTS
0 ERRORS DETECTED
10870 BYTES FREE

NDOS SYSTEM OVERLAYS
CROSS REFERENCE TABLE

XREF V1.1
PAGE 85

.NAME	000054	289L	857																		
.OPENC	000045	282L	843																		
.OPENR	000042	279L	837																		
.OPENU	000044	281L	841																		
.OPENW	000043	280L	839																		
.PCHL	002264	161E																			
.POSIT	000047	284L	847																		
.PRINT	000003	263L	1993																		
.RCK	003260	169E																			
.READ	000004	264L																			
.REGI	040005	177E																			
.REGPTR	040035	188E																			
.RENAM	000051	286L	851																		
.RESET	000204	306L																			
.RNB	002331	164E																			
.RNP	002325	163E																			
.SCIN	000001	261L																			
.SCOUT	000002	262L	1998	2038																	
.SETTP	000052	287L	853																		
.SRS	002265	162E																			
.START	040000	175E																			
.SYSRES	000012	270L																			
.TICCNT	040033	187E																			
.TPERR	002205	160E																			
.TPERRX	040031	186E																			
.UIVEC	040037	189E																			
.VERS	000011	269L																			
.WNB	003024	167E																			
.WNP	003017	166E																			
.WRITE	000005	265L																			
ABS.COD	000010	536L	1681	1686	1699																
ABS.ENT	000006	534L	1751																		
ABS.ID	000000	530L	1642	1650	1660																
ABS.LDA	000002	532L	1650	1660	1664	1684															
ABS.LEN	000004	533L	1664	1698																	
AFP	002201	1470	1490L																		
AFP1	002176	1488L	1507																		
AFP2	002227	1495	1498	1505L																	
AIO.CGN	041047	769L	1026	1027	1218	1463	1464	1490	1493	2348											
AIO.CHA	041116	784L	908	1319	1882	2238	2649														
AIO.CNT	041111	780L																			
AIO.CSI	041050	770L	1027	1464	1493																
AIO.DDA	041041	765E	909	1714	2236	2238	2296	2903													
AIO.DES	041055	774L	1276	2803	2841																
AIO.DEV	041057	775L	1544	1546	1582	1592	2468	2501	2874												
AIO.DIR	041062	778L	1011	1031	1090	1213	1223	1225	1226	1302	1309	1427	1461								
		1543	1592	2305	2340	2343	2345	2433	2473	2511	2528	2551	2653	2789							
		2815	3097	3102																	
AIO.DTA	041053	773L	1549	2671	2866	3071															
AIO.EOF	041113	782L																			
AIO.EOM	041112	781L																			
AIO.FLG	041043	766L	1355	1392	1540	2112	2428	2939													
AIO.GRT	041044	767L	1162	2943	2961	3061															
AIO.LGN	041051	771L	1028	1029	1222	1491	2344	3095	3096												
AIO.LSI	041052	772L	1029	1496	3096																
AIO.SPG	041046	768L	1220	1505	2908																
AIO.TFP	041114	783L																			
AIO.UNI	041061	776L	1412	1425	1543	1544	2258	2305	2307	2674	2914	2920	2935								

HDOS SYSTEM OVERLAYS
CROSS REFERENCE TABLE

XREF V1.1
PAGE 86

3074						
AID.VEC	041040	764L	907	1328		
BELL	000007	84E	2050			
BITS	010220	2923	3139L			
BITS1	010225	3144L	3146			
BKSP	000010	86E				
BOOT.P	000001	744E				
C.STX	000002	88E				
C.SYN	000026	87E				
CAC	005037	2235L	2417			
CB.CLI	000100	123E	138			
CB.MTL	000040	122E				
CB.SPK	000200	124E				
CB.SSI	000020	121E				
CDB.H84	000001	687E				
CDB.H85	000000	686E				
CDU	005054	2258L	2421			
CDU1	005073	2263	2267L			
CFC	005076	935	2285L	2365		
CFC1	005103	2290L	2322			
CFC2	005147	2300	2311	2319L		
CFD	005157	942	1086	1627	2012	2338L
CFE	031354	213E				
CFP	005212	992	1082	1153	2365L	2992
CHFLG	004263	866	2109L			
CLEAR	003351	860	1879E			
CLEAR1	003360	1892L	1943			
CLOSE	001213	846	1257E			
CLOSE1	001250	1273	1275L			
CLOSE2	001266	1272	1286L			
CLOSE3	001275	1274	1292L			
CLOSE4	001306	1292	1298L			
CLOSE6	001312	1294	1300L			
CLOSE7	001321	1288	1308L			
CLOSE8	001343	1264	1266	1319L		
CLRALL	004013	862	1931E			
CLRALL1	004021	1936L	1946			
CD.FLG	000001	663E				
CR	000015	80E				
CS.FLG	000200	644E				
CSL.CHR	000001	641E				
CSL.ECH	000200	639E				
CSL.WRP	000002	640E				
CTLA	000001	95E				
CTLR	000002	96E				
CTLC	000003	97E				
CTLD	000004	98E				
CTLO	000017	99E				
CTLP	000020	100E				
CTLG	000021	101E				
CTLS	000023	102E				
CTLZ	000032	103E				
CTP.2SB	000010	649E				
CTP.BKM	000002	650E				
CTP.BKS	000200	646E				
CTP.HLI	000040	647E				
CTP.HLO	000020	648E				
CTP.TAB	000001	651E				

MDOS SYSTEM OVERLAYS
CROSS REFERENCE TABLE

XREF V1.1
PAGE 87

D.COM	040110	600L							
D.RAM	040240	603L							
D.VEC	040130	602L							
DC.ABT	000007	437L							
DC.CLO	000006	436L	1327						
DC.LOD	000011	439L	2214						
DC.MAX	000012	440L							
DC.MOU	000010	438L							
DC.DPR	000003	433L	943						
DC.DPU	000005	435L	1096						
DC.DPW	000004	434L	1033	1233					
DC.REA	000000	430L							
DC.RER	000002	432L							
DC.WRI	000001	431L							
DCA	032002	215E							
DDF.BOL	000011	500E							
DDF.BOO	000000	497L							
DDF.LAB	000011	501L							
DDF.RGT	000012	502L							
DDF.USR	000014	503L							
DEBUG	000001	2E	1741	2119	2136				
DECODE	002241	856	1533E						
DELETE	001357	850	1341E						
DEV.BDA	000004	322L							
DEV.DVR	000014	334L	3019						
DEV.DVL	000014	333L	3014	3019					
DEV.FLB	000006	323L							
DEV.JMP	000003	321L	2901	2905					
DEV.MNU	000011	330L	2912	2925					
DEV.MUM	000010	329L	2189	2925	2933				
DEV.NAM	000000	313L							
DEV.RES	000002	317L	2895	3014					
DEV.SFB	000007	328L	2905	2912					
DEV.UNT	000012	331L	2672	2933	3072				
DEVELEN	000017	334E	2887						
DF.CLR	000376	356E	2820						
DF.EMP	000377	355E	1364	2817					
DFA	005234	1535	1617	2008	2212	2417L	2985		
DFA1	005303	2431	2435	2441L					
DFC	005221	920	980	1068	1135	2392L			
DFD	005308	1417	2418	2465L					
DFD1	005360	2480	2488L						
DFD2	006023	2491	2509L						
DFD3	006074	2516	2534L						
DFD4	006076	2482	2484	2539L					
DFD5	006141	2489	2498	2505	2522	2525	2555	2557	2566L
DIF.CNT	000020	556E	1228						
DIF.LOC	000100	554E	1405	2127	2133				
DIF.SYS	000200	553E	2122						
DIF.WP	000040	555E	1003	1091	1351	1405	2122		
DIR.ALD	000025	371L	1309						
DIR.CLU	000015	364L	1011	1225					
DIR.CRD	000023	370L	1302						
DIR.EXT	000010	359L	1597	2473	2528				
DIR.FGN	000020	367L	1031	1213	1461	1912	2345	2958	
DIR.FLG	000016	365L	1000	1090	1226	1348	1402	2129	
DIR.LGN	000021	368L	1223	2343	3097	3098			
DIR.LSI	000022	369L	3098						

..... HDOS SYSTEM OVERLAYS
 CROSS REFERENCE TABLE

XREF 01.1
 PAGE 91

LOAA	005031	2211	2218L		
LOADD	005014	870	2210E		
M.CDCA	000017	583L			
M.CDLY	000016	582L			
M.CFWA	000012	580L			
M.CIN	000006	578L			
M.CINT	000005	577L	1734		
M.CLWA	000014	581L			
M.COUT	000010	579L			
M.CPRE	000003	575L			
M.CRUB	000004	576L			
M.CSLC	000002	574L			
M.FDX	000303	148E			
M.FAMS	000021	147E			
M.SALO	000001	573L			
M.SYSM	000000	572L	1732		
MI.CPI	000376	23E			
MI.JMP	000303	24E			
MI.RET	000311	25E			
NAME	002300	858	1574L		
NAME1	002312	1577	1581L		
NL	000012	92E	93	2036	2046
NUL2	000000	83E			
NULL	000200	82E			
OP.CTL	000360	115E			
OP.DIG	000360	116E			
OP.SEG	000361	117E			
OPENC	001020	844	1127E		
OPENC1	001027	1129	1134L		
OPENC2	001061	1150	1152L		
OPENC3	001104	1169L	1192		
OPENC4	001115	1180L	1187		
OPENC5	001135	1181	1206L		
OPENC9	001210	1147	1233L		
OPENCA	001114	1152	1176E		
OPENEX	000143	906	913L		
OPENHL	000145	915E	2420		
OPENR	000114	838	898E		
OPENR1	000150	900	920L		
OPENR2	000212	934	943L		
OPENU	000340	842	1061E		
OPENU1	000347	1063	1068L		
OPENW	000215	840	973E		
OPENW1	000224	975	980L		
OPENW2	000273	995	1010L		
OPENW3	000335	991	1033L		
OPENX	000120	906L	976	1064	1130
QVL.IN	000001	711E	1847	2172	
QVL.NUM	000014	713E			
QVL.RES	000002	712E	2172		
QVL.UCS	000200	714E	1738	1835	2179
FBI	033145	225E			
PIC.COD	000006	548L			
PIC.ID	000000	543L			
PIC.LEN	000002	545L			
PIC.PTR	000004	546L			
POSIT	002124	648	1450E		
POSIT2	002155	1467L	1475		

DDOS SYSTEM OVERLAYS
CROSS REFERENCE TABLE

XREF VI.1
PAGE 93

S.SDD	041010	733L					
S.SOUR	041144	607L	609				
S.SSN	041002	722L					
S.SYSM	040320	630L	1809	2184	2195	3023	3046
S.TIME	040312	627L					
S.UCSF	040372	717L	1824	2183			
S.UCSL	040374	718L					
S.USRM	040322	632L	1819	3037			
S.VAL	040277	604L	623				
S.WRITE	031330	207E					
SB.ORG	047000	194E					
SB.OVMX	014000	195E	3555				
SC.UART	000372	41E					
SCTLC	003242	836	1779E				
SETTOP	003270	854	1476	1807E			
SETTOP1	003334	1830	1841L				
SGT	010111	1312	1368	3061L			
STACK	042200	611E					
STACKL	001032	609E					
SYDD	040130	601E					
SYSCALL	000377	253E	1993	1998	2038		
TAB	000011	90E	3218				
TFE	033233	231E					
UCI.ER	000020	63E					
UCI.IE	000002	65E					
UCI.IR	000100	61E					
UCI.RE	000004	64E					
UCI.RO	000040	62E					
UCI.TE	000001	66E					
UDE	010156	1311	3094L				
UDR	000000	38E					
UDS	010177	1314	1366	1430	2144	3116L	
UMI.16X	000002	56E					
UMI.1B	000100	46E					
UMI.1X	000001	55E					
UMI.2B	000300	48E					
UMI.64X	000003	57E					
UMI.HB	000200	47E					
UMI.L5	000000	51E					
UMI.L6	000004	52E					
UMI.L7	000010	53E					
UMI.L8	000014	54E					
UMI.PA	000020	50E					
UMI.PE	000040	49E					
UNT.BIS	000005	345L	2677				
UNT.FLG	000000	342L	2938				
UNT.GRT	000001	343L	2941				
UNT.GTS	000003	344L	3076				
UNT.SIZ	000007	347E					
UD.CLK	000001	140E					
UD.DDU	000002	139E					
UD.HLT	000200	137E					
UD.NFR	000100	138E					
USERFWA	042200	612E					
USR	000001	39E					
USR.FE	000040	70E					
USR.OE	000020	71E					
USR.PE	000010	72E					

