

```

2
3
4 ***   SYSGEN IS A CUT-DOWN FROM PIP.
5 *
6 ***   PIP - PERIPHERAL INTERCHANGE PROGRAM.
7 *
8 *     J.B. LETWIN, 11/1977 FOR *HEATH* COMPANY
9 *
10 *    COPYRIGHT 1977, 1979 BY HEATH COMPANY
11 *
12 *    G. Chandler, 9/78      Maintenance release
13 *                          79/04      Issue --.04.--
14 *                          79/11      Issue --.05.--
15 *

```

```

17 ***   USE:
18 *
19 *     no commands may be entered to sysgen. SYSGEN
20 *     contains an internal default command.

```

22 \*\* SYSTEM EQUIVALENCES

```

23
24 000.000 CN.SOU EQU 0 SOURCE CHANNEL NUMBER
25 000.001 CN.DES EQU 1 DESTINATION CHANNEL NUMBER
26 000.002 CN.DIR EQU 2 DIRECTORY CHANNEL NUMBER
27

```

28 \*\* PROGRAM ERROR CODES

```

29
30 000.200 PEC.DF EQU 2000 DEVICE FORMAT ERROR
31 000.201 PEC.DNC EQU 2010 DEVICES NOT CONSISTANT
32 000.202 PEC.RSE EQU 2020 RENAME SPECIFICATION ERROR
33 000.203 PEC.TFI EQU 2030 TARGET FILE ILLEGAL
34 000.204 PEC.CS EQU 2040 CONTRADICTIONARY SWITCHES
35 000.205 PEC.IUW EQU 2050 ILLEGAL USE OF WILDCARD
36 000.206 PEC.IDF EQU 2060 ILLEGAL DESTINATION FILE FORMAT
37

```

```

000.000 38 XTEXT U8250

```

40X \*\* 8250 UART CONTROL AND BIT DEFINITIONS.

```

41X
42X 000.350 SC.ACE EQU 3500 SYSTEM CONSOLE PORT IF 8250 ACE
43X 000.156 AC.DLY EQU 110 220 MIL. SEC. DELAY FOR 8250
44X
45X 000.000 UR.RBR EQU 0 RECEIVER BUFFER REGISTER (READ ONLY)
46X
47X 000.000 UR.THR EQU 0 TRANSMITTER HOLDING REGISTER (WRITE ONLY)
48X

```

000.000	49X UR.DLL EQU	0	DIVISOR LATCH (LEAST SIGNIFICANT)
	50X		
000.001	51X UR.DLM EQU	1	DIVISOR LATCH (MOST SIGNIFICANT)
	52X		
000.001	53X UR.IER EQU	1	INTERRUPT ENABLE REGISTER
000.001	54X UC.EDA EQU	00000001B	ENABLE RECEIVED DATA AVAILABLE INTERRUPT
000.002	55X UC.TRE EQU	00000010B	ENABLE TRANSMIT HOLD REGISTER EMPTY INTERRUPT
000.004	56X UC.RSI EQU	00000100B	ENABLE RECEIVE STATUS INTERRUPT
000.010	57X UC.HSI EQU	00001000B	ENABLE MODEM STATUS INTERRUPT
	58X		
000.002	59X UR.IIR EQU	2	INTERRUPT IDENTIFICATION REGISTER
000.001	60X UC.IIP EQU	00000001B	INVERTED INTERRUPT PENDING (0 MEANS PENDING)
000.006	61X UC.IID EQU	00000110B	INTERRUPT ID
	62X		
000.003	63X UR.LCR EQU	3	LINE CONTROL REGISTER
000.000	64X UC.5BW EQU	00000000B	5 BIT WORDS
000.001	65X UC.6BW EQU	00000001B	6 BIT WORDS
000.002	66X UC.7BW EQU	00000010B	7 BIT WORDS
000.003	67X UC.8BW EQU	00000011B	8 BIT WORDS
000.004	68X UC.2SB EQU	00000100B	TWO STOP BITS SELECTED
000.010	69X UC.PEN EQU	00001000B	PARITY COMPUTATION ENABLED
000.020	70X UC.EPS EQU	00010000B	EVEN PARITY SELECT
000.040	71X UC.SKP EQU	00100000B	STICK PARITY
000.100	72X UC.SB EQU	01000000B	SET BREAK
000.200	73X UC.DLA EQU	10000000B	DIVISOR LATCH ACCESS
	74X		
000.004	75X UR.MCR EQU	4	MODEM CONTROL REGISTER
000.001	76X UC.DTR EQU	00000001B	DATA TERMINAL READY
000.002	77X UC.RTS EQU	00000010B	REQUEST TO SEND
000.004	78X UC.OU1 EQU	00000100B	OUT 1
000.010	79X UC.OU2 EQU	00001000B	OUT 2
000.020	80X UC.LOD EQU	00010000B	LOOP
	81X		
000.005	82X UR.LSR EQU	5	LINE STATUS REGISTER
000.001	83X UC.DR EQU	00000001B	DATA READY
000.002	84X UC.OR EQU	00000010B	OVERRUN
000.004	85X UC.FE EQU	00000100B	PARITY ERROR
000.010	86X UC.FR EQU	00001000B	FRAMING ERROR
000.020	87X UC.BI EQU	00010000B	BREAK INTERRUPT
000.040	88X UC.THE EQU	00100000B	TRANSMITTER HOLDING REGISTER EMPTY
000.100	89X UC.TSE EQU	01000000B	TRANSMITTER SHIFT REGISTER EMPTY
	90X		
000.006	91X UR.MSR EQU	6	MODEM STATUS REGISTER
000.001	92X UC.DCS EQU	00000001B	DELTA CLEAR TO SEND
000.002	93X UC.DDR EQU	00000010B	DELTA DATA SET READY
000.004	94X UC.TER EQU	00000100B	TRAILING EDGE OF RING
000.010	95X UC.DRL EQU	00001000B	DELTA RECEIVE LINE SIGNAL DETECT
000.020	96X UC.CTS EQU	00010000B	CLEAR TO SEND
000.040	97X UC.DSR EQU	00100000B	DATA SET READY
000.100	98X UC.RI EQU	01000000B	RING INDICATOR
000.200	99X UC.RLS EQU	10000000B	RECEIVED LINE SIGNAL DETECT
000.000	100 XTEXT	U8251	

```

103X **      8251 USART BIT DEFINITIONS.
104X *
105X
106X **      PORT ADDRESSES
107X
000.000     108X UDR   EQU   0           DATA REGISTER IS EVEN
000.001     109X USR   EQU   1           STATUS REGISTER IS NEXT
110X
000.372     111X SC.USART EQU  3720        CONSOLE USART ADDRESS (IFF 8251)
112X
113X
114X **      MODE INSTRUCTION CONTROL BITS.
115X
000.100     116X UMI.1B  EQU  01000000B    1 STOP BIT
000.200     117X UMI.HB  EQU  10000000B    1 1/2 STOP BITS
000.300     118X UMI.2B  EQU  11000000B    2 STOP BITS
000.040     119X UMI.PE  EQU  00100000B    EVEN PARITY
000.020     120X UMI.PA  EQU  00010000B    USE PARITY
000.000     121X UMI.L5  EQU  00000000B    5 BIT CHARACTERS
000.004     122X UMI.L6  EQU  00000100B    6 BIT CHARACTERS
000.010     123X UMI.L7  EQU  00001000B    7 BIT CHARACTERS
000.014     124X UMI.L8  EQU  00001100B    8 BIT CHARACTERS
000.001     125X UMI.1X  EQU  00000001B    CLOCK X 1
000.002     126X UMI.16X EQU  00000010B    CLOCK X 16
000.003     127X UMI.64X EQU  00000011B    CLOCK X 64
128X
129X **      COMMAND INSTRUCTION BITS.
130X
000.100     131X UCI.IR  EQU  01000000B    INTERNAL RESET
000.040     132X UCI.RQ  EQU  00100000B    READER-ON CONTROL FLAG
000.020     133X UCI.ER  EQU  00010000B    ERROR RESET
000.004     134X UCI.RE  EQU  00000100B    RECEIVE ENABLE
000.002     135X UCI.IE  EQU  00000010B    ENABLE INTERRUPTS FLAG
000.001     136X UCI.TE  EQU  00000001B    TRANSMIT ENABLE
137X
138X **      STATUS READ COMMAND BITS.
139X
000.040     140X USR.FE  EQU  00100000B    FRAMING ERROR
000.020     141X USR.OE  EQU  00010000B    OVERRUN ERROR
000.010     142X USR.PE  EQU  00001000B    PARITY ERROR
000.004     143X USR.TXE  EQU  00000100B    TRANSMITTER EMPTY
000.002     144X USR.RXR  EQU  00000010B    RECEIVER READY
000.001     145X USR.TXR  EQU  00000001B    TRANSMITTER READY
000.000     146      XTEXT  DIRDEF

148X **      DIRECTORY ENTRY FORMAT.
149X
000.000     150X      ORG    0
151X
152X
000.377     153X DF.EMP  EQU  3770        FLAGS ENTRY EMPTY
000.376     154X DF.CLR  EQU  3760        FLAGS ENTRY EMPTY, REST OF DIR ALSO CLEAR
155X
000.000     156X DIR.NAM IS  8          NAME

```

000.010	157X	DIR.EXT	DS	3	EXTENSION
000.013	158X	DIR.PRO	DS	1	PROJECT
000.014	159X	DIR.VER	DS	1	VERSION
000.015	160X	DIRIDL	EQU	*	FILE IDENTIFICATION LENGTH
	161X				
000.015	162X	DIR.CLU	DS	1	CLUSTER FACTOR
000.016	163X	DIR.FLG	DS	1	FLAGS
000.017	164X		DS	1	RESERVED
000.020	165X	DIR.FGN	DS	1	FIRST GROUP NUMBER
000.021	166X	DIR.LGN	DS	1	LAST GROUP NUMBER
000.022	167X	DIR.LSI	DS	1	LAST SECTOR INDEX (IN LAST GROUP)
000.023	168X	DIR.CRD	DS	2	CREATION DATE
000.025	169X	DIR.ALD	DS	2	LAST ALTERATION DATE
	170X				
000.027	171X	DIRELEN	EQU	*	DIRECTORY ENTRY LENGTH
000.027	172		XTEXT	DIFDEF	

174X \*\* DIRECTORY FILE FLAGS.

	175X				
000.200	176X	DIF.SYS	EQU	10000000B	SYSTEM FILE
000.100	177X	DIF.LOC	EQU	01000000B	LOCKED FOR CHANGE
000.040	178X	DIF.WP	EQU	00100000B	WRITE PROTECTED
000.020	179X	DIF.CNT	EQU	00010000B	CONTIGUOUS FILE
	180X				
000.027	181		XTEXT	DEVDEF	

183X \*\* DEVICE TABLE ENTRIES.

	184X				
000.000	185X		ORG	0	
	186X				
000.000	187X	DEV.NAM	DS	2	DEVICE NAME
000.000	188X	DV.EL	EQU	00000000B	END OF DEVICE LIST FLAG
000.001	189X	DV.NU	EQU	00000001B	DEVICE ENTRY NOT IN USE
	190X				
000.002	191X	DEV.RES	DS	1	DRIVER RESIDENSE CODE
000.001	192X	DR.IM	EQU	00000001B	DRIVER IN MEMORY
000.002	193X	DR.FR	EQU	00000010B	DRIVER PERMINANTLY RESIDENT
	194X				
000.003	195X	DEV.JMP	DS	1	JMP TO PROCESSOR
000.004	196X	DEV.IDA	DS	2	DRIVER ADDRESS
000.006	197X	DEV.FLG	DS	1	FLAG BYTE
000.001	198X	DT.DD	EQU	00000001B	DIRECTORY DEVICE
000.002	199X	DT.CR	EQU	00000010B	CAPABLE OF READ OPERATION
000.004	200X	DT.CW	EQU	00000100B	CAPABLE OF WRITE OPERATION
	201X				
000.007	202X	DEV.SPG	DS	1	SECTORS PER GROUP THIS DEVICE
000.010	203X	DEV.MUM	DS	1	MOUNTED UNIT MASK
000.011	204X	DEV.MNU	DS	1	MAXIMUM NUMBER OF UNITS
000.012	205X	DEV.UNT	DS	2	ADDRESS OF UNIT SPECIFIC DATA TABLE
	206X				

DEV

000.014	207X	DEV.DVL DS	2	DRIVER BYTE LENGTH
000.016	208X	DEV.DVG DS	1	DRIVER ROUTINE GROUP ADDRESS
	209X			
000.017	210X	DEVELEN EQU	*	DEVICE TABLE ENTRY LENGTH
	212X	**		UNIT SPECIFIC DEVICE DATA TABLE ENTRIES
	213X			
000.000	214X	ORG	0	
	215X			
000.000	216X	UNT.FLG DS	1	UNIT SPECIFIC *DEV.FLG*
000.001	217X	UNT.GRT DS	2	ADDRESS OF GROUP RESERVATION TABLE (IF DT.DD)
000.003	218X	UNT.GTS DS	2	GRT SECTOR NUMBER
000.005	219X	UNT.DIS DS	2	DIRECTORY FIRST SECTOR NUMBER
	220X			
000.007	221X	UNT.SIZ EQU	*	SIZE OF UNIT SPECIFIC DATA TABLE PER UNIT
000.007	222	XTEXT	IOCDEF	
	224X	**		I/O CHANNEL DEFINITIONS.
	225X			
000.000	226X	ORG	0	
	227X			
000.000	228X	IOC.LNK DS	2	ADDRESS OF NEXT CHANNEL; =0 IF LAST
000.002	229X	IOC.DDA DS	2	THREAD JUMP TO DEVICE DRIVER (VIA DEV.TABLE)
	230X			
000.004	231X	IOC.FLG DS	1	FILE TYPE FLAGS
000.001	232X	FT.BB EQU	00000001B	=1 IF DIRECTORY DEVICE
000.002	233X	FT.OR EQU	00000010B	=1 IF OPEN FOR READ
000.004	234X	FT.OW EQU	00000100B	=1 IF OPEN FOR WRITE
000.010	235X	FT.OU EQU	00001000B	=1 IF OPEN FOR UPDATE
000.003	236X	IOC.SQL EQU	*-IOC.DDA	LENGTH OF INFO FOR SEQUENTIAL FILE (FROM IOC)
	237X			
000.005	238X	IOC.GRT DS	2	ADDRESS OF GROUP RESERVATION TABLE
000.007	239X	IOC.SPG DS	1	SECTORS PER GROUP, THIS DEVICE
000.010	240X	IOC.CGN DS	1	CURRENT GROUP NUMBER
000.011	241X	IOC.CSI DS	1	CURRENT SECTOR INDEX (IN CURRENT GROUP)
000.012	242X	IOC.LGN DS	1	LAST GROUP NUMBER
000.013	243X	IOC.LSI DS	1	LAST SECTOR INDEX (IN LAST GROUP)
000.010	244X	IOC.DRL EQU	*-IOC.FLG	LENGTH OF INFO NORMALLY COPIED BACK TO THE CHANNEL TABLE
	245X	*		
000.014	246X	IOC.DTA DS	2	DEVICE TABLE ADDRESS FOR THIS DEVICE
000.016	247X	IOC.DES DS	2	SECTOR NUMBER OF DIRECTORY ENTRY
000.020	248X	IOC.DEV DS	2	DEVICE CODE
000.022	249X	IOC.UNI DS	1	UNIT NUMBER (0-9)
000.021	250X	IOC.DIL EQU	*-IOC.DDA	LENGTH OF INFO FOR DIRECTORY FILE (FROM IOC)
	251X			
000.023	252X	IOC.DIR DS	DIRELEN	DIRECTORY ENTRY
	253X			
000.052	254X	IOCELEN EQU	*	IOC ENTRY LENGTH
	255X			
000.001	256X	IOCCTD EQU	1	INDEX OF USER CHANNEL #0 IN CHANTAB (FIRST = 0)
000.052	257	XTEXT	DISDEF	

```

259X **      DIRECTORY BLOCK FORMAT.
260X
000.000     261X      ORG      0
262X
000.000     263X DIS.ENT EQU      *      FIRST ENTRY ADDRESS
000.000     264X      DS      22*DIRELEN  22 DIRECTORY ENTRYS PER BLOCK
001.372     265X      DS      1          0 BYTE = END OF ENTRYS IN THIS BLOCK
266X
001.373     267X      ORG      $12-5     AT END OF BLOCK
001.373     268X DIS.ENL DS      1          LENGTH OF EACH ENTRY (=DIRELEN)
001.374     269X DIS.SEC DS      2          BLOCK # OF THIS BLOCK,
001.376     270X DIS.LNK DS      2          BLOCK # OF NEXT BLOCK, =0 IF THIS IS LAST
002.000     271      XTEXT  FBDEF
    
```

273X \*\* FILE BLOCK DEFINITIONS.

```

274X
000.000     275X      ORG      0
000.000     276X FB.CHA DS      1          CHANNEL NUMBER
000.001     277X FB.FLG DS      1          FLAGS
000.002     278X FB.FWA DS      2          BUFFER FWA
000.004     279X FB.PTR DS      2          BUFFER POINTER
000.006     280X FB.LIM DS      2          LIMIT OF DATA IN BUFFER (READ OPERATIONS)
000.010     281X FB.LWA DS      2          LWA OF BUFFER
000.012     282X FB.NAM DS      4+8+4+1   NAME OF FILE
000.021     283X FB.NAML EQU      *-FB.NAM
000.033     284X FBENL EQU      *          ENTRY LENGTH
000.033     285      XTEXT  ECDEF
    
```

287X \*\* ERROR CODE DEFINITIONS.

```

288X
000.000     289X      ORG      0
000.000     290X      DS      1          NO ERROR #0
000.001     291X EC.EOF DS      1          END OF FILE
000.002     292X EC.EDM DS      1          END OF MEDIA
000.003     293X EC.ILC DS      1          ILLEGAL SYSCALL CODE
000.004     294X EC.CNA DS      1          CHANNEL NOT AVAILABLE
000.005     295X EC.DNS DS      1          DEVICE NOT SUITABLE
000.006     296X EC.IDN DS      1          ILLEGAL DEVICE NAME
000.007     297X EC.IFN DS      1          ILLEGAL FILE NAME
000.010     298X EC.NRD DS      1          NO ROOM FOR DEVICE DRIVER
000.011     299X EC.FNO DS      1          CHANNEL NOT OPEN
000.012     300X EC.ILR DS      1          ILLEGAL REQUEST
000.013     301X EC.FUC DS      1          FILE USAGE CONFLICT
000.014     302X EC.FNF DS      1          FILE NAME NOT FOUND
000.015     303X EC.UND DS      1          UNKNOWN DEVICE
000.016     304X EC.ICN DS      1          ILLEGAL CHANNEL NUMBER
000.017     305X EC.DIF DS      1          DIRECTORY FULL
000.020     306X EC.IFC DS      1          ILLEGAL FILE CONTENTS
000.021     307X EC.NEM DS      1          NOT ENOUGH MEMORY
000.022     308X EC.RF  DS      1          READ FAILURE
000.023     309X EC.WF  DS      1          WRITE FAILURE
    
```

000.024	310X	EC.WPV	DS	1	WRITE PROTECTION VIOLATION
000.025	311X	EC.WF	DS	1	DISK WRITE PROTECTED
000.026	312X	EC.FAP	DS	1	FILE ALREADY PRESENT
000.027	313X	EC.DDA	DS	1	DEVICE DRIVER ABORT
000.030	314X	EC.FL	DS	1	FILE LOCKED
000.031	315X	EC.FAD	DS	1	FILE ALREADY OPEN
000.032	316X	EC.IS	DS	1	ILLEGAL SWITCH
000.033	317X	EC.UUN	DS	1	UNKNOWN UNIT NUMBER
000.034	318X	EC.FNR	DS	1	FILE NAME REQUIRED
000.035	319X	EC.DIW	DS	1	DEVICE IS NOT WRITABLE (OR WRITE LOCKED)
000.036	320X	EC.UNA	DS	1	UNIT NOT AVAILABLE
000.037	321X	EC.ILV	DS	1	ILLEGAL VALUE
000.040	322X	EC.ILD	DS	1	ILLEGAL OPTION
000.041	323X	EC.VPM	DS	1	VOLUME PRESENTLY MOUNTED ON DEVICE
000.042	324X	EC.NVM	DS	1	NO VOLUME PRESENTLY MOUNTED
000.043	325X	EC.FOD	DS	1	FILE OPEN ON DEVICE
000.044	326X	EC.NPM	DS	1	NO PROVISIONS MADE FOR REMOUNTING MORE DISKS
000.045	327X	EC.DNI	DS	1	DISK NOT INITIALIZED
000.046	328X	EC.DNR	DS	1	DISK IS NOT READABLE
000.047	329X	EC.DSC	DS	1	DISK STRUCTURE IS CORRUPT
000.050	330X	EC.NCV	DS	1	NOT CORRECT VERSION OF HDOS
000.051	331X	EC.NOS	DS	1	NO OPERATING SYSTEM MOUNTED
000.052	332X	EC.IOI	DS	1	ILLEGAL OVERLAY INDEX
000.053	333X	EC.OTL	DS	1	OVERLAY TOO LARGE
000.054	334	XTEXT	OVLDEF		

336X \*\* OVERLAY TABLE ENTRIES.

	337X				
000.000	338X	ORG		0	
	339X				
000.000	340X	OVL.COD	DS	2	FIRST SECTOR OF OVERLAY CODE
000.002	341X	OVL.SIZ	DS	2	OVERLAY SIZE
000.004	342X	OVL.ENT	DS	2	OVERLAY ENTRY POINT
000.006	343X	OVL.FLB	DS	1	OVERLAY FLAG BYTE
000.007	344X		DS	1	DUMMY BYTE TO ROUND TABLE SIZE UP TO 8
000.010	345X	OVL.ENS	EGU	*	OVERLAY ENTRY SIZE
	346X				
	347X	*			OVERLAY INDICES
	348X				
000.000	349X	ORG		0	
	350X				
000.000	351X	OVL0	DS	1	
000.001	352X	OVL1	DS	1	
000.002	353	XTEXT	HOSEGU		

355X \*\* HDOS SYSTEM EQUIVALENCES.

	356X *			
	357X			
024.000	358X S.GRT0	EQU	24000A	SYSTEM AREA FOR GRT0
025.000	359X S.GRT1	EQU	25000A	SYSTEM AREA FOR GRT1
026.000	360X S.GRT2	EQU	26000A	SYSTEM AREA FOR GRT2
	361X			
030.000	362X ROMBOOT	EQU	30000A	ROM BOOT ENTRY
	363X			
040.100	364X	ORG	40100A	FREE SPACE FROM PAM-8
	365X			
040.100	366X	DS	8	JUMP TO SYSTEM EXIT
040.110	367X D.CON	DS	16	DISK CONSTANTS
040.130	368X SYDD	EQU	*	SYSTEM DISK ENTRY POINT
040.130	369X D.VEC	DS	24*3	SYSTEM ROM ENTRY VECTORS
040.240	370X D.RAM	DS	31	SYSTEM ROM WORK AREA
040.277	371X S.VAL	DS	36	SYSTEM VALUES
040.343	372X S.INT	DS	115	SYSTEM INTERNAL WORK AREAS
041.126	373X	DS	16	
041.146	374X S.SOVR	DS	2	STACK OVERFLOW WARNING
041.150	375X	DS	42200A-*	SYSTEM STACK
001.032	376X STACKL	EQU	*-S.SOVR	STACK SIZE
	377X			
042.200	378X STACK	EQU	*	LWA+1 SYSTEM STACK
042.200	379X USERFWA	EQU	*	USER FWA
042.200	380	XTEXT	HOSDEF	

382X \*\* HOSDEF - DEFINE HOS PARAMETER.

	383X *			
	384X			
	385X			
000.026	386X VERS	EQU	1*16+6	VERSION 1.6
	387X			
000.377	388X SYSCALL	EQU	3770	SYSCALL INSTRUCTION
	389X			
	390X			
000.000	391X	ORG	0	
	392X			

393X \* RESIDENT FUNCTIONS

	394X			
000.000	395X .EXIT	DS	1	EXIT (MUST BE FIRST)
000.001	396X .SCIN	DS	1	SCIN
000.002	397X .SCOUT	DS	1	SCOUT
000.003	398X .PRINT	DS	1	PRINT
000.004	399X .READ	DS	1	READ
000.005	400X .WRITE	DS	1	WRITE
000.006	401X .CONSL	DS	1	SET/CLEAR CONSOLE OPTIONS
000.007	402X .CLRCD	DS	1	CLEAR CONSOLE BUFFER
000.010	403X .LOADO	DS	1	LOAD AN OVERLAY
000.011	404X .VERS	DS	1	RETURN HDOS VERSION NUMBER
000.012	405X .SYSRES	DS	1	PRECEDING FUNCTIONS ARE RESIDENT

406X  
 407X  
 408X \* \*HDOSQVLO.SYS\* FUNCTIONS



```

.....
000.040      409X
.....
000.040      410X      ORG      40A
.....
000.040      411X
.....
000.040      412X .LINK DS      1      LINK (MUST BE FIRST)
000.041      413X .CTLC DS      1      CTL-C
000.042      414X .OPENR DS      1      OPENR
000.043      415X .OPENW DS      1      OPENW
000.044      416X .OPENU DS      1      OPENU
000.045      417X .OPENC DS      1      OPENC
000.046      418X .CLOSE DS      1      CLOSE
000.047      419X .POSIT DS      1      POSITION
000.050      420X .DELET DS      1      DELETE
000.051      421X .RENAM DS      1      RENAME
000.052      422X .SETTP DS      1      SETTOP
000.053      423X .DECODE DS      1      NAME DECODE
000.054      424X .NAME DS      1      GET FILE NAME FROM CHANNEL
000.055      425X .CLEAR DS      1      CLEAR CHAN
000.056      426X .CLEARA DS      1      CLEAR ALL CHANS
000.057      427X .ERROR DS      1      LOOKUP ERROR
000.060      428X .CHFLG DS      1      CHANGE FLAGS
000.061      429X .DISMT DS      1      FLAG SYSTEM DISK DISMOUNTED
000.062      430X .LDADD DS      1      LOAD DEVICE DRIVER
.....
431X
432X
433X *      *HDOSVLI.SYS* FUNCTIONS
.....
000.200      434X
.....
000.200      435X      ORG      2000
.....
000.200      436X
.....
000.200      437X .MOUNT DS      1      MOUNT (MUST BE FIRST)
000.201      438X .DMOUN DS      1      DISMOUNT
000.202      439X .MONMS DS      1      MOUNT/NO MESSAGE
000.203      440X .DMNMS DS      1      DISMOUNT/NO MESSAGE
000.204      441X .RESET DS      1      RESET = DISMOUNT/MOUNT OF UNIT
000.205      442      XYEXT  ASCII
.....

```

444X \*\* ASCII CHARACTER EQUIVALENCES.

```

.....
000.015      445X
.....
000.012      446X CR      EQU      13      CARRIAGE RETURN
.....
000.200      447X LF      EQU      10      LINE FEED
000.000      448X NULL   EQU      2000    PAD CHARACTER
000.007      449X NUL2   EQU      0       0
.....
000.007      450X BELL   EQU      7       BELL CHARACTER
.....
000.177      451X RUBOUT  EQU      177R    RUBOUT
.....
000.010      452X BKSP   EQU      100     CTL-H
000.026      453X C.SYN  EQU      260     SYNC
000.002      454X C.STX  EQU      2       STX
000.047      455X QUOTE  EQU      470     QUOTE
000.011      456X TAB   EQU      110     TAB
.....
000.033      457X ESC   EQU      330     ESC
000.012      458X NL    EQU      120     NL
000.212      459X ENL   EQU      NL+2000  NEW LINE (HDOS SYSTEMS)
.....
000.014      460X FF    EQU      140     FORM FEED
000.001      461X CTLA   EQU      010     CTL-A
000.002      462X CTLB  EQU      020     CTL-B
000.003      463X CTLC  EQU      030     CTL-C
.....

```

000.004	464X	CTLD	EQU	040	CTL-D
000.017	465X	CTL0	EQU	170	CTL-0
000.020	466X	CTLP	EQU	200	CTL-P
000.021	467X	CTLQ	EQU	210	CTL-Q
000.023	468X	CTLS	EQU	230	CTL-S
000.032	469X	CTLZ	EQU	320	CTL-Z
000.205	470		XTEXT	EDRAM	
	472X	**		EDRAM - DISK RAM WORKAREA DEFINITION.	
	473X	*			
	474X	*		ZEROED UPON BOOTING UP.	
	475X	*			
	476X	*		HOSEQU MUST BE CHANGED WHEN THIS DECK IS CHANGED.	
	477X				
	478X				
040.240	479X		ORG	D.RAM	
	480X				
040.240	481X	D.TT	DS	1	TARGET TRACK (CURRENT OPERATION)
040.241	482X	D.TS	DS	1	TARGET SECTOR (CURRENT OPERATION)
	483X				
040.242	484X	D.DVCTL	DS	1	DEVICE CONTROL BYTE
	485X				
040.243	486X	D.DLYMD	DS	1	MOTOR ON DELAY COUNT
040.244	487X	D.DLYHS	DS	1	HEAD SETTLE DELAY COUNTER
	488X				
040.245	489X	D.TRKPT	DS	2	ADDRESS IN D.DRVTB FOR TRACK NUMBER
040.247	490X	D.VOLPT	DS	2	ADDRESS IN D.DRVTB FOR VOLUME NUMBER
	491X				
040.251	492X	D.DRVTB	DS	2*4	TRACK NUMBER AND VOLUME NUMBER FOR 4 DRIVES
	493X				
040.261	494X	D.HECNT	DS	1	HARD ERROR COUNT
040.262	495X	D.SECNT	DS	2	SOFT ERROR COUNT
040.264	496X	D.OECNT	DS	1	OPERATION ERROR COUNT
	497X				
	498X	*		GLOBAL DISK ERROR COUNTERS	
	499X				
040.265	500X	D.ERR	DS	0	BEGINNING OF ERROR BLOCK
040.265	501X	D.E.MDS	DS	1	MISSING DATA SYNC
040.266	502X	D.E.HSY	DS	1	MISSING HEADER SYNC
040.267	503X	D.E.CHK	DS	1	DATA CHECKSUM
040.270	504X	D.E.HCK	DS	1	HEADER CHECKSUM
040.271	505X	D.E.VOL	DS	1	WRONG VOLUME NUMBER
040.272	506X	D.E.TRK	DS	1	BAD TRACK SEEK
040.273	507X	D.ERRL	DS	0	LIMIT OF ERROR COUNTERS
	508X				
	509X	*		I/O OPERATION COUNTS	
	510X				
040.273	511X	D.OPR	DS	2	
040.275	512X	D.OPW	DS	2	
	513X				
000.037	514X	D.RAML	EQU	*-D.RAM	
040.277	515		XTEXT	ESINT	

```

517X **      S.INT - SYSTEM INTERNAL WORKAREA DEFINITIONS.
518X *
519X *      THESE CELLS ARE REFERENCED BY OVERLAYS AND MAIN CODE, AND
520X *      MUST THEREFORE RESIDE IN FIXED LOW MEMORY.
521X
522X
040.343     523X      ORG      S.INT
524X
525X **      CONSOLE STATUS FLAGS
526X
040.343     527X S.CDB   DS      1      CONSOLE DESCRIPTOR BYTE
000.000     528X CDB.H85 EQU    00000000B
000.001     529X CDB.H84 EQU    00000001B      =0 IF H8-5, =1 IF H8-4
040.344     530X S.BAUD  DS      2      [0-14] H8-4 BAUD RATE, =0 IF H8-5
531X *      [15]      =1 IF BAUD RATE => 2 STOP BITS
532X
533X **      TABLE ADDRESS WORDS
534X
040.346     535X S.DLINK DS      2      ADDRESS OF DATA IN HDOS CODE
040.350     536X S.DFWA  DS      2      FWA OVERLAY TABLE
040.352     537X S.CFWA  DS      2      FWA CHANNEL TABLE
040.354     538X S.DFWA  DS      2      FWA DEVICE TABLE
040.356     539X S.RFWA  DS      2      FWA RESIDENT HDOS CODE
540X
541X **      DEVICE DRIVER DELAYED LOAD FLAGS
542X
040.360     543X S.DDLDA DS      2      DRIVER LOAD ADDRESS (HIGH BYTE=0 IF NO LOAD PENDING)
040.362     544X S.DDLEN DS      2      CODE LENGTH IN BYTES
040.364     545X S.DDGRP DS      1      GROUP NUMBER FOR DRIVER
040.365     546X      DS      1      HOLD PLACE
547X *S.DDSEC DS      2      SECTOR NUMBER FOR DRIVER (* OBSOLETE ! *)
040.366     548X S.DDDTA DS      2      DEVICE'S ADDRESS IN DEVLST +DEV.RES
040.370     549X S.DDOFC DS      1      OPEN OF CODE PENDING
550X
551X **      OVERLAY MANAGEMENT FLAGS
552X
000.001     553X OVL.IN  EQU    00000001B      IN MEMORY
000.002     554X OVL.RES EQU    00000010B      PERMINANTLY RESIDENT
000.014     555X OVL.NUM EQU    00001100B      OVERLAY NUMBER MASK
000.200     556X OVL.UCS EQU    10000000B      USER CODE SWAPPED FOR OVERLAY
557X
040.371     558X S.OVLFL DS      1      OVERLAY FLAG
040.372     559X S.UCSF  DS      2      FWA SWAPPED USER CODE
040.374     560X S.UCSL  DS      2      LENGTH SWAPPED USER CODE
040.376     561X S.OVLS  DS      2      SIZE OF OVERLAY CODE
041.000     562X S.OVLE  DS      2      ENTRY POINT OF OVERLAY CODE
563X
041.002     564X S.SSN   DS      2      SWAP AREA SECTOR NUMBER
041.004     565X S.OSN   DS      2      OVERLAY SECTOR NUMBER
566X
567X *      SYSCALL PROCESSING WORK AREAS
568X
041.006     569X S.CACC  DS      1      (ACC) UPON SYSCALL
041.007     570X S.CODE  DS      1      SYSCALL INDEX IN PROGRESS
571X
572X *      JUMPS TO ROUTINES IN RESIDENT HDOS CODE

```

	573X				
041.010	574X	S.JUMPS DS	0		START OF DUMP VECTORS
041.010	575X	S.SDD DS	3		JUMP TO STAND-IN DEVICE DRIVER
041.013	576X	S.FASER DS	3		JUMP TO FATERR (FATAL SYSTEM ERROR)
041.016	577X	S.DIREA DS	3		JUMP TO DIREAD (DISK FILE READ)
041.021	578X	S.FCI DS	3		JUMP TO FCI (FETCH CHANNEL INFO)
041.024	579X	S.SCI DS	3		JUMP TO SCI (STORE CHANNEL INFO)
041.027	580X	S.GUP DS	3		JUMP TO GUP (GET UNIT POINTER)
	581X				
041.032	582X	S.MOUNT DS	1		<>0 IF THE SYSTEM DISK IS MOUNTED
041.033	583X	S.DCS DS	1		DEFAULT CLUSTER SIZE-1
	584X				
041.034	585X	S.BOOTF DS	1		BOOT FLAGS
000.001	586X	BOOT.F EQU	00000001B		EXECUTE PROLOGUE UPON BOOTUP
	587X				
	588X	*		STACK VALUE SAVED FOR OVERLAY SYSCALLS	
	589X				
041.035	590X	S.OVSTK DS	2		VALUE OF SP UPON SYSCALLS USING OVERLAY
	591X				
041.037	592X	DS	1		RESERVED
	594X	**		ACTIVE I/O AREA.	
	595X	*			
	596X	*		THE AIO.XXX AREA CONTAINS INFORMATION ABOUT THE I/O OPERATION	
	597X	*		CURRENTLY BEING PERFORMED. THE INFORMATION IS OBTAINED FROM	
	598X	*		THE CHANNEL TABLE, AND WILL BE RESTORED THERE WHEN DONE.	
	599X	*			
	600X	*		NORMALLY, THE AIO.XXX INFORMATION WOULD BE OBTAINED DIRECTLY	
	601X	*		FROM VARIOUS SYSTEM TABLES VIA POINTER REGISTERS. SINCE THE	
	602X	*		BO80 HAS NO GOOD INDEXED ADDRESSING, THE DATA IS MANUALLY	
	603X	*		COPIED INTO THE AIO.XXX CELLS BEFORE PROCESSING, AND	
	604X	*		BACKDATED AFTER PROCESSING.	
	605X				
041.040	606X	AIO.VEC DS	3		JUMP INSTRUCTION
041.041	607X	AIO.DDA EQU	*-2		DEVICE DRIVER ADDRESS
041.043	608X	AIO.FLG DS	1		FLAG BYTE
041.044	609X	AIO.GRT DS	2		ADDRESS OF GROUP RESERV TABLE
041.046	610X	AIO.SPG DS	1		SECTORS PER GROUP
041.047	611X	AIO.CGN DS	1		CURRENT GROUP NUMBER
041.050	612X	AIO.CSI DS	1		CURRENT SECTOR INDEX
041.051	613X	AIO.LGN DS	1		LAST GROUP NUMBER
041.052	614X	AIO.LSI DS	1		LAST SECTOR INDEX
041.053	615X	AIO.DTA DS	2		DEVICE TABLE ADDRESS
041.055	616X	AIO.DES DS	2		DIRECTORY SECTOR
041.057	617X	AIO.DEV DS	2		DEVICE CODE
041.061	618X	AIO.UNI DS	1		UNIT NUMBER (0-9)
	619X				
041.062	620X	AIO.DIR DS	DIRELEN		DIRECTORY ENTRY
	621X				
041.111	622X	AIO.CNT DS	1		SECTOR COUNT
041.112	623X	AIO.EOM DS	1		END OF MEDIA FLAG
041.113	624X	AIO.EOF DS	1		END OF FILE FLAG
041.114	625X	AIO.TFP DS	2		TEMP FILE POINTERS

041.116 626X AIO.CHA DS 2 ADDRESS OF CHANNEL BLOCK (IOC.DDA)

041.120 628X S.SCR DS 2 SYSTEM SCRATCH AREA ADDRESS  
041.122 629 XTEXT ESVAL

631X \*\* S.VAL - SYSTEM VALUE DEFINITIONS.

632X \*

633X \* THESE VALUES ARE SET AND MAINTAINED BY THE SYSTEM.

634X \*

635X \* THE DECK HOSEQU MUST BE MODIFIED WHEN THIS IS MODIFIED.

636X

637X

040.277 638X ORG S.VAL

639X

040.277 640X S.DATE DS 9 SYSTEM DATE (IN ASCII)

040.310 641X S.DATC DS 2 CODED DATE

040.312 642X S.TIME DS 4 TIME FROM MIDNIGHT (IN TICS)

040.316 643X S.HIMEM DS 2 HARDWARE HIGH MEMORY ADDRESS+1

644X

040.320 645X S.SYSM DS 2 FWA RESIDENT SYSTEM

646X

040.322 647X S.USRM DS 2 LWA USER MEMORY

648X

040.324 649X S.OMAX DS 2 MAX OVERLAY SIZE FOR SYSTEM

650X

651X

652X \*\* THE FOLLOWING FIVE CELLS SHOULD BE MODIFIED/READ ONLY VIA THE .CONSL SYSCALL

653X

000.200 654X CSL.ECH EQU 10000000B SUPPRESS ECHO

000.002 655X CSL.WRP EQU 00000010B WRAP LINES AT WIDTH

000.001 656X CSL.CHR EQU 00000001B OPERATE IN CHARACTER MODE

657X

000.000 658X I.CSLMD EQU 0 S.CSLMD IS FIRST BYTE

040.326 659X S.CSLMD DS 1 CONSOLE MODE

660X

000.200 661X CTP.BKS EQU 10000000B TERMINAL PROCESSES BACKSPACES

000.040 662X CTP.MLI EQU 00100000B MAP LOWER CASE TO UPPER ON INPUT

000.020 663X CTP.MLO EQU 00010000B MAP LOWER CASE TO UPPER ON OUTPUT

000.010 664X CTP.2SB EQU 00001000B TERMINAL NEEDS TWO STOP BITS

000.002 665X CTP.BRM EQU 00000010B MAP BKSP (UPON INPUT) TO RUBOUT

000.001 666X CTP.TAB EQU 00000001B TERMINAL SUPPORTS TAB CHARACTERS

667X

000.001 668X I.CONTY EQU 1 S.CONTY IS 2ND BYTE

000.000 669X ERRNZ \*-S.CSLMD-I.CONTY

040.327 670X S.CONTY DS 1 CONSOLE TYPE FLAGS

000.002 671X I.CUSOR EQU 2 S.CUSOR IS 3RD BYTE

000.000 672X ERRNZ \*-S.CSLMD-I.CUSOR

040.330 673X S.CUSOR DS 1 CURRENT CURSOR POSITION

000.003 674X I.CONWI EQU 3 S.CONWI IS 4TH BYTE

000.000 675X ERRNZ \*-S.CSLMD-I.CONWI

040,331	676X	S.CONWI	DS	1	CONSOLE WIDTH
	677X				
000,001	678X	CO.FLG	EQU	00000001B	CTL-O FLAG
000,200	679X	CS.FLG	EQU	10000000B	CTL-S FLAG
	680X				
000,004	681X	I.CONFL	EQU	4	S.CONFL IS 5TH BYTE
000,000	682X	ERRNZ	*-S,CSLMD-I,CONFL		
040,332	683X	S.CONFL	DS	1	CONSOLE FLAGS
	684X				
040,333	685X	S.CAADR	DS	2	ADDRESS FOR ABORT PROCESSING (>256 IF VALID)
040,335	686X	S.CCTAB	DS	6	ADDR FOR CTL-A, CTL-B, CTL-C PROCESSING
040,343	687	XTEXT	DDDEF		

689X \*\* DEVICE DRIVER COMMUNICATION FLAGS.

	690X	*			
	691X				
000,000	692X	DRG	DS	0	
	693X				
000,000	694X	DC.REA	DS	1	READ
000,001	695X	DC.WRI	DS	1	WRITE
000,002	696X	DC.RER	DS	1	READ REGARDLESS
000,003	697X	DC.OPR	DS	1	OPEN FOR READ
000,004	698X	DC.OPW	DS	1	OPEN FOR WRITE
000,005	699X	DC.OPU	DS	1	OPEN FOR UPDATE
000,006	700X	DC.CLO	DS	1	CLOSE
000,007	701X	DC.ART	DS	1	ABORT
000,010	702X	DC.MOU	DS	1	MOUNT DEVICE
000,011	703X	DC.LOD	DS	1	LOAD DEVICE DRIVER
000,012	704X	DC.MAX	DS	1	MAXIMUM ENTRY INDEX
000,013	705	XTEXT	MTR		

708X \*\* MTR - PAM/8 EQUIVALENCES.

709X \*  
710X \* THIS DECK CONTAINS SYMBOLIC DEFINITIONS USED TO  
711X \* MAKE USE OF THE PAM/8 CODE AND CONTROL BYTES.

713X \*\* IO PORTS

714X  
000.360 715X IP.PAD EQU 3600 PAD INPUT PORT  
000.360 716X OP.CTL EQU 3600 CONTROL OUTPUT PORT  
000.360 717X OP.DIB EQU 3600 DIGIT SELECT OUTPUT PORT  
000.361 718X OP.SEG EQU 3610 SEGMENT SELECT OUTPUT PORT

720X \*\* FRONT PANEL CONTROL BITS.

721X  
000.020 722X CB.SSI EQU 00010000B SINGLE STEP INTERRUPT  
000.040 723X CB.MTL EQU 00100000B MONITOR LIGHT  
000.100 724X CB.CLI EQU 01000000B CLOCK INTERRUPT ENABLE  
000.200 725X CB.SPK EQU 10000000B SPEAKER ENABLE

727X \*\* MONITOR MODE FLAGS.

728X  
000.000 729X DM.MR EQU 0 MEMORY READ  
000.001 730X DM.MW EQU 1 MEMORY WRITE  
000.002 731X DM.RR EQU 2 REGISTER READ  
000.003 732X DM.RW EQU 3 REGISTER WRITE

734X \*\* USER OPTION BITS.

735X \*  
736X \* THESE BITS ARE SET IN CELL .MFLAG.  
737X  
000.200 738X UO.HLT EQU 10000000B DISABLE HALT PROCESSING  
000.100 739X UO.NFR EQU CB.CLI NO REFRESH OF FRONT PANEL  
000.002 740X UO.DDU EQU 00000010B DISABLE DISPLAY UPDATE  
000.001 741X UO.CLK EQU 00000001B ALLOW PRIVATE INTERRUPT PROCESSING

743X \*\* MONITOR IDENTIFICATION FLAGS

744X \*  
745X \* THESE BYTES IDENTIFY THE ROM MONITOR.  
746X \* THEY ARE THE VARIOUS VALUES OF LOCATION .IDENT  
747X  
000.021 748X M.PAMB EQU 0210 'LXI' INSTRUCTION AT 000.000 IN PAM-8  
000.303 749X M.FOX EQU 3030 'JMP' INSTRUCTION AT 000.000 IN FOX ROM

751X \*\* ROUTINE ENTRY POINTS.

Address	Label	Equation	Value	Description
000.000	754X .IDENT	EQU	0000A	IDENTIFICATION LOCATION
000.053	755X .DLY	EQU	0053A	DELAY
001.267	756X .LOAD	EQU	1267A	TAPE LOAD
001.374	757X .DUMP	EQU	1374A	TAPE DUMP
002.136	758X .ALARM	EQU	2136A	ALARM ROUTINE
002.140	759X .HORN	EQU	2140A	HORN
002.172	760X .CTC	EQU	2172A	CHECK TAPE CHECKSUM
002.205	761X .TPERR	EQU	2205A	TAPE ERROR ROUTINE
002.264	762X .FCHL	EQU	2264A	FCHL INSTRUCTION
002.265	763X .SRS	EQU	2265A	SCAN RECORD START
002.325	764X .RNP	EQU	2325A	READ NEXT PAIR
002.331	765X .RNB	EQU	2331A	READ NEXT BYTE
002.347	766X .CRC	EQU	2347A	CRC-16 CALCULATOR
003.017	767X .WNP	EQU	3017A	WRITE NEXT PAIR
003.024	768X .WNB	EQU	3024A	WRITE NEXT BYTE
003.122	769X .DOB	EQU	3122A	DECODE FOR OCTAL DISPLAY
003.260	770X .RCK	EQU	3260A	READ CONSOLE KEYS
003.356	771X .DODA	EQU	3356A	SEGMENT CODE TABLE

773X \*\* RAM CELLS USED BY HBMT.

Address	Label	Equation	Value	Description
040.000	776X .START	EQU	40000A	START DUMP ADDRESS
040.002	777X .IOWRK	EQU	40002A	IN OR OUT INSTRUCTION
040.005	778X .REGI	EQU	40005A	DISPLAYED REGISTER INDEX
040.006	779X .BSPROT	EQU	40006A	PERIOD FLAG BYTE
040.007	780X .ISPHOD	EQU	40007A	DISPLAY MODE
040.010	781X .MFLAG	EQU	40010A	USER OPTION BYTE
040.011	782X .CTLFLG	EQU	40011A	PANEL CONTROL BYTE
040.013	783X .ALEDS	EQU	40013A	ABUSS LEDS
040.021	784X .DLEDS	EQU	40021A	DBUSS LEDS
040.024	785X .ABUSS	EQU	40024A	ABUSS REGISTER
040.027	786X .CRCSUM	EQU	40027A	CRCSUM WORD
040.031	787X .TPERRX	EQU	40031A	TAPE ERROR EXIT VECTOR
040.033	788X .TICNT	EQU	40033A	CLOCK TICK COUNTER
040.035	789X .REGPTR	EQU	40035A	REGISTER POINTER
040.037	790X .UIVEC	EQU	40037A	USER INTERRUPT VECTORS
000.013	791	XTEXT	DDFDEF	

793X \*\* DIRECTORY DEVICE FORMAT DEFINITION.

Address	Label	Equation	Value	Description
000.002	797X HDS.SPG	EQU	2	2 SECTORS PER GROUP REQUIRED FOR NOW
000.000	798X			
000.000	799X	ORG	0	
000.000	800X DDF.BOD	DS	9	2K BOOT PROGRAM
000.011	801X DDF.BOL	EQU	*	LENGTH OF BOOT
000.011	802X DDF.LAB	DS	1	LABEL SECTOR



000.012	803X	DDF.RGT	DS	2	RESERVED GROUP TABLE
000.014	804X	DDF.USR	DS	0	BEGINNING OF OPEN SPACE
000.014	805	XTEXT	LABDEF		
807X ** DISK LABEL SECTOR FORMATS.					
808X					
000.000	809X	DRG		0	
000.000	810X	LAB.SER	DS	1	SERIAL NUMBER OF VOLUME
000.001	811X	LAB.IND	DS	2	INITIALIZATION DATE
000.003	812X	LAB.DIS	DS	2	SECTOR NUMBER OF 1ST DIRECTORY SECTOR
000.005	813X	LAB.GRT	DS	2	INDEX OF GRT SECTOR
000.007	814X	LAB.SPG	DS	1	SECTORS PER GROUP
815X					
000.000	816X	LAB.DAT	EQU	0	DATA VOLUME ONLY
000.001	817X	LAB.SYS	EQU	1	SYSTEM VOLUME
000.002	818X	LAB.NOD	EQU	2	=> LAB.NOD MEANS VOLUME HAS NO DIRECTORY
819X					
000.010	820X	LAB.VLT	DS	1	VOLUME TYPE
000.011	821X	LAB.VER	DS	1	VERSION OF INIT17 THAT INITED DISK
000.012	822X		DS	7	UNUSED
000.021	823X	LAB.LAB	DS	60	LABEL
000.074	824X	LAB.LBL	EQU		*-LAB.LAB LABEL LENGTH
000.115	825	XTEXT	FILDEF		

827X \*\* FILDEF - FILE TYPE DEFINITIONS.

828X *					
	829X	*	DB	3770,FT.XXX	
830X					
831X					
000.000	832X	FT.ABS	EQU	0	ABSOLUTE BINARY
000.001	833X	FT.PIC	EQU	1	POSITION INDEPENDANT CODE
000.002	834X	FT.REL	EQU	2	RELOCATABLE CODE
000.003	835X	FT.BAC	EQU	3	COMPILED BASIC CODE
000.115	836	XTEXT	ABSDEF		

838X \*\* ABS FORMAT EQUIVALENCES.

839X					
000.000	840X		DRG	0	
841X					
000.000	842X	ABS.ID	DS	1	3770 = BINARY FILE FLAG
000.001	843X		DS	1	FILE TYPE (FT.ABS)
000.002	844X	ABS.LDA	DS	2	LOAD ADDRESS
000.004	845X	ABS.LEN	DS	2	LENGTH OF ENTIRE RECORD
000.006	846X	ABS.ENT	DS	2	ENTRY POINT
847X					
000.010	848X	ABS.COD	DS	0	CODE STARTS HERE

```

042.170          851      ORG      USERFWA-ABS.CDD
042.170  377 000      852      DB      3770,FT.ABS
042.172  200 042      853      DW      USERFWA      LOAD ADDRESS
042.174  202 014      854      DW      MEML-USERFWA  SIZE
042.176  254 055      855      DW      ENTRY      ENTRY
                   856
042.200          857 PYP      EQU      *
                   858
                   859 *      COMMAND INTERPRETATION COMES HERE
                   860
042.200  061 200 042 861 START LXI      SP,STACK      CLEAN STACK
                   862
                   863 *      CLEAR CHANNELS AND FILE BUFFER
                   864
042.203  377 056      865      DB      SYSCALL, .CLEARA CLEAR CHANNELS
                   866
                   867 *      CLEAR DYNAMIC BUFFERS
                   868
042.205  041 000 000 869      LXI      H,0
042.210  042 111 055 870      SHLD   BUFSIZ      EMPTY BUFFER
042.213  042 146 055 871      SHLD   NAMTLEN     CLEAR NAMTAB
042.216  042 150 055 872      SHLD   NAMTHAX     CLEAR NAMTAB AREA
042.221  041 327 055 873      LXI      H,BUFF
042.224  042 107 055 874      SHLD   BUFPTR      SET BUFFER AGAINST END OF NAMTAB
                   875
                   876 *      INPUT COMMAND LINE
                   877
042.227  315 066 043 878      CALL   OCOPY      COPY FILES
                   879
                   880 *      SET SYSGENED FLAG IN LABEL
                   881
042.232  041 157 046 882      LXI      H,MNDA
042.235  377 203      883      DB      SYSCALL, .DMNMS
042.237  332 007 050 884      JC      ERROR
                   885
042.242  056 000      886      MVI      L,0
042.244  076 010      887      MVI      A,DC.MOU
042.246  315 130 040 888      CALL   SYDD      MOUNT DISK AS 0
042.251  041 011 000 889      LXI      H,DDF.LAB
042.254  001 000 001 890      LXI      B,256
042.257  315 241 031 891      CALL   $WER      WRITE ENABLE RAM
042.262  021 000 027 892      LXI      D,LABEL
042.265  076 000      893      MVI      A,DC.REA
042.267  315 130 040 894      CALL   SYDD      READ LABEL
042.272  332 007 050 895      JC      ERROR      BAD TROUBLE
042.275  076 001      896      MVI      A,LAB.SYS
042.277  062 010 027 897      STA      LABEL+LAB.ULT SET VOLUME TYPE
042.302  021 000 027 898      LXI      D,LABEL
042.305  041 011 000 899      LXI      H,DDF.LAB
042.310  001 000 001 900      LXI      B,256
042.313  076 001      901      MVI      A,DC.WRI
042.315  315 130 040 902      CALL   SYDD      WRITE LABEL BACK
042.320  332 007 050 903      JC      ERROR      BAD TROUBLE
042.323  257          904      XRA      A
042.324  303 334 042 905      JMP      EXIT.      GRACEFUL EXIT
                   906

```

```
907 ** NO RESTARTING ALLOWED
908
042.327 909 RESTART EQU *
910
042.327 303 332 042 911 JMP EXIT EXIT
912
913 * CTL-D HIT
914
042.332 076 001 915 EXIT MUI A,I FLAG ABORT
042.334 377 000 916 EXIT DB SYSCALL,.EXIT EXIT TO *HDOS*
```

  

```
918 ** CCHIT - CTL-C HIT
919 *
920 * ENTRY FROM SYSTEM
921
922
042.336 315 136 031 923 CCHIT CALL $TYPTX
042.341 136 303 924 DB 'C'+2000
042.343 303 332 042 925 JMP EXIT BOOT IT
926
927
928 * COMMAND LINE
929
042.346 052 056 052 930 LINE DB /*.*=*,SYS,PIP,ABS,SET,ABS,FLAGS,ONECOPY,
043.016 052 056 104 931 DB /*.DVD,SYSHelp.DOC,HELP',',0
043.046 932 DS 16
```

```

936 *** SYSGEN - COPY FILES BETWEEN TWO VOLUMES, WITH ONLY ONE
937 * DRIVE.
938 *
939 * (AND FOR MY NEXT TRICK...)
940 *
941 * OPECOPY COPIES FILES BETWEEN TWO VOLUMES BY ALTERNATING BETWEEN
942 * TWO PHASES, THE READ PHASE AND THE WRITE PHASE.
943 *
944 * READ PHASE:
945 *
946 * DURING THE READ PHASE, THE SOURCE DISK IS MOUNTED, SOURCE FILES ARE
947 * OPENED IN THE ORDER OF THEIR APPEARANCE. FOR EACH OPENED
948 * FILE, A 'FILE DESCRIPTOR NODE' *FDN* IS ADDED TO THE ACTIVE
949 * CHAIN, THEN, AS MUCH AS THE FILE AS POSSIBLE IS READ INTO MEMORY.
950 *
951 * THE PROCESS CONTINUES UNTIL
952 * 1) THERE IS NO MORE FREE RAM
953 * 2) OR, THERE ARE NO MORE FILE DESCRIPTOR NODES IN THE FREE CHAIN
954 * 3) OR, THERE ARE NO MORE FILES IN NAMTAB (INPUT FILE LIST)
955 *
956 *
957 * WRITE PHASE
958 *
959 * DURING THE WRITE PHASE, THE DESTINATION DISK IS MOUNTED, THE NODES
960 * ARE TAKEN FROM THE ACTIVE CHAIN, AND PROCESSED. IF THE FILE HAD
961 * BEEN PARTIALLY WRITTEN THE LAST PASS, IT IS RE-OPENED AND POSITIONED,
962 * IF THERE IS NOT MORE DATA TO READ FOR A PROCESSED
963 * NODE, IT IS REMOVED, AND THE CORRESPONDING ENTRY IN NAMTAB IS DELETED.
964 *
965 * WRITE PHASE CONTINUES UNTIL
966 *
967 * 1) THERE ARE NO MORE FILE NODES IN THE ACTIVE LIST
968 * 2) OR, THE FIRST (AND ONLY) ENTRY IN THE LIST HAS NO
969 * MORE DATA IN MEMORY, BUT HAS NOT BEEN COMPLETELY READ.
970 *
971 * OCOPY EXITS WITH THE DESTINATION DISK MOUNTED.
972 *
973 *
043.066 974 COPY EQU * CALLED 'COPY' BY MAINLINE CODE
043.066 975 OCOPY EQU *
043.066 315 314 045 976 CALL IFL INITIALIZE FDN LISTS
043.071 257 977 XRA A
043.072 062 276 054 978 STA VOLFLAG FLAG SOURCE VOLUME MOUNTED
043.075 072 252 040 979 LDA D,DRVTR+1
043.100 062 271 054 980 STA VOLSER SET VOLUME SERIAL NUMBER
043.103 315 040 051 981 CALL DDF DECODE DESTINATION FILE
043.106 332 007 050 982 JC ERROR ERROR
043.111 062 301 043 983 STA OCOPIA SAVE DESTINATION TYPE
043.114 257 984 XRA A ALLOW *.*
043.115 315 233 050 985 CALL BSL BUILD SOURCE FILE LIST
043.120 332 007 050 986 JC ERROR
043.123 315 076 054 987 CALL $MOVEL
043.126 021 000 988 DW OCOPIYL
043.130 125 055 989 DW DESTFB+FB.NAM
043.132 303 043 990 DW OCOPIYD SAVE WILDCARD DESTINATION
043.134 315 100 052 991 CALL EBM EXPAND BUFFER TO MAX
  
```



```

1049 **      RPH = READ PHASE.
1050 *
1051 *      RPH HANDLES THE READ PHASE OF THE COPY PROCESS.
1052 *
1053 *      IT IS ENTERED WITH THE NAMTAB AND FDN TABLE SETUP, AND
1054 *      WITH THE SOURCE DISK MOUNTED.
1055 *
1056 *      READ PHASE:
1057 *
1058 *      DURING THE READ PHASE, THE SOURCE DISK IS MOUNTED. SOURCE FILES ARE
1059 *      OPENED IN THE ORDER OF THEIR APPEARANCE. FOR EACH OPENED
1060 *      FILE, A FILE DESCRIPTOR NODE *FDN* IS ADDED TO THE ACTIVE
1061 *      CHAIN. THEN, AS MUCH AS THE FILE AS POSSIBLE IS READ INTO MEMORY.
1062 *
1063 *      THE PROCESS CONTINUES UNTIL
1064 *          1) THERE IS NO MORE FREE RAM
1065 *          2) OR, THERE ARE NO MORE FILE DESCRIPTOR NODES IN THE FREE CHAIN
1066 *          3) OR, THERE ARE NO MORE FILES IN NAMTAB (INPUT FILE LIST)
1067 *
1068 *      ENTRY  NONE
1069 *      EXIT   NONE
1070 *      USES  ALL
1071
1072
043.375 1073 RPH  EQU  *
1074
1075
1076 *      SEE IF ANY MEMORY TO HAVE
1077
043.375 315 306 045 1078 CALL  CBR      COMPUTE BUFFER ROOM
044.000 310          1079 RZ      NONE
1080
1081 *      SEE IF WE NEED TO READ SOME MORE INTO A PART-COPIED FILE
1082
044.001 041 147 054 1083 LXI  H,FINHEAD
044.004 156          1084 MOV  L,M      (HL) = ADDRESS IF FIRST NODE
044.005 175          1085 MOV  A,L
044.006 247          1086 ANA  A
044.007 312 024 044 1087 JZ   RPH1     IS NO FIRST NODE, ERGO NO FILE
044.012 043          1088 INX  H
000.000          1089 ERRNZ FDN,STA-1
044.013 176          1090 MOV  A,M      (A) = .STA
044.014 346 002     1091 ANI  ST,OPR
044.016 021 327 055 1092 LXI  D,NAMTAB
044.021 302 117 044 1093 JNZ  RPH2.5   FILE IS INCOMPLETELY READ
1094
1095 *      SEE IF ANY FREE FILE DESCRIPTOR NODES TO USE
1096
044.024 072 146 054 1097 RPH1  LDA  FDNFRE
044.027 247          1098 ANA  A
044.030 310          1099 RZ      NO MORE
1100
1101 *      SEE IF THERE IS A FILE IN NAMTAB WITHOUT AN ENTRY IN FDNLIST.
1102 *      SINCE THE FIRST ENTRY IN FDNLIST CORRESPONDS TO THE FIRST IN
1103 *      NAMTAB, ETC., WE'LL JUST RUN DOWN FDNLIST UNTIL THE END, AND
1104 *      THE NEXT NAMTAB FILE WILL BE THE ONE WE WANT...

```

RPH

```

1105
044.031 001 021 000 1106 LXI B,FB.NAML (BC) = ENTRY SIZE IN NAMTAB
044.034 021 357 377 1107 LXI D,-FB.NAML (DE) = POINTER INTO NAMTAB
044.037 041 147 054 1108 LXI H,FDNHEAD
044.042 175 1109 MOV A,L START WITH FDNHEAD
044.043 157 1110 RPH2 MOV L,A FOLLOW LINK
044.044 176 1111 MOV A,M (A) = NEXT NODE
044.045 353 1112 XCHG
044.046 011 1113 DAD B ADVANCE POINTER INTO NAMTAB
044.047 353 1114 XCHG
044.050 247 1115 ANA A
044.051 302 043 044 1116 JNZ RPH2 LINK SOME MORE
044.054 345 1117 PUSH H (HL) = ADDRESS OF LAST NODE
044.055 052 146 055 1118 LHLI NAMTLEN
044.060 315 216 030 1119 CALL $CDEHL SEE IF HAVE ACCOUNTED FOR ALL NAMTAB ENTRIES
044.063 341 1120 POP H
044.064 310 1121 RE FILES ALL USED UP
1122
1123 * HAVE ROOM FOR DATA, HAVE A NODE FOR THE FILE COUNTS, AND
1124 * HAVE A FILE NAME, ALL SET FOR BUSINESS.
1125 *
1126 * (DE) = INDEX INTO NAMTAB FOR FILE
1127 * (HL) = NODE ADDRESS OF LAST ENTRY IN LIST
1128 *
1129 * CHAIN THE FIRST FREE NODE ONTO THE END OF THE LIST
1130
044.065 072 146 054 1131 LDA FDNFRE
044.070 167 1132 MOV M,A CHAIN TO NEW END NODE
044.071 157 1133 MOV L,A
044.072 176 1134 MOV A,M (A) = NEXT NODE IN FREE CHAIN
044.073 062 146 054 1135 STA FDNFRE
044.076 006 012 1136 MVI B,FDNLEN
044.100 345 1137 PUSH H SAVE NODE ADDRESS
044.101 315 212 031 1138 CALL $ZERO ZERO ENTIRE NODE, ENCLUING CHAIN (AT END, NOW)
044.104 001 327 055 1139 LXI B,NAMTAB
044.107 353 1140 XCHG
044.110 011 1141 DAD B (HL) = ADDRESS OF NAMTAB ENTRY
044.111 042 152 055 1142 SHLD NAMPTR POINTER TO CURRENT NAMTAB ENTRY
044.114 353 1143 XCHG
044.115 341 1144 POP H
000.000 1145 ERRNZ FDN.STA-1
044.116 043 1146 INX H (HL) = ADDR OF FDN.STA OF NODE
1147
1148 * READY TO OPEN FILE
1149 *
1150 * (DE) = NAMTAB ENTRY ADDRESS
1151 * (HL) = #FDN.STA OF ENTRY
1152
044.117 345 1153 RPH2.5 PUSH H SAVE ADDRESS
044.120 353 1154 XCHG
044.121 257 1155 XRA A
000.000 1156 ERRNZ CN,SOU (A) = SOURCE CHANNEL NUMBER
044.122 377 042 1157 DB SYSCALL,OPEN OPEN
044.124 332 153 047 1158 JC NAMERR ERROR
044.127 321 1159 POP D
044.130 032 1160 LDAX D (A) = FDN.STA

```

SYSGEN SUBROUTINES

RPH

16:07:59 16-MAY-80

```

044.131 346 002 1161 ANI ST,OPR
044.133 325 1162 PUSH D SAVE ADDRESS
044.134 302 225 044 1163 JNZ RPH3 ALREADY OPENED IN PREVIOUS PASSES
1164
1165 * FIRST TIME THIS FILE HAS BEEN OPENED. SEE IF CONTIGUOUS
1166
044.137 041 302 043 1167 LXI H,OCOPYC
044.142 064 1168 INR M
044.143 032 1169 LBAX D
044.144 366 002 1170 OKI ST,OPR SET OPEN FOR READ
044.146 022 1171 STAX D
044.147 325 1172 PUSH D SAVE #FDN,STA
044.150 052 352 040 1173 LHLD S,CFWA (HL) = CHANNEL 0 FWA
000.000 1174 ERRNZ IOCCTD-1 MUST SKIP A CHANNEL FOR USER #0
044.153 315 211 030 1175 CALL $HLIHL (HL) = #USER CHANNEL 0
000.000 1176 ERRNZ CN,SOU ASSUME WE WANT CHANNEL 0
044.156 315 234 030 1177 CALL $INDL
044.161 041 000 1178 DW IOC,DIR+DIR,FLG
044.163 173 1179 MOV A,E (A) = DIR,FLG
044.164 321 1180 POP D (DE) = #FDN,STA
000.000 1181 ERRNZ FDN,FLG-FDN,STA-1
044.165 023 1182 INX D (DE) = FDN,FLG
044.166 022 1183 STAX D SAVE FILE FLAGS
044.167 346 020 1184 ANI DIF,CNT
044.171 312 225 044 1185 JZ RPH3 NOT CONTIG
1186
1187 * IS CONTIG. GET FILE SIZE
1188
044.174 315 234 030 1189 CALL $INDL
044.177 005 000 1190 DW IOC,GRT
044.201 325 1191 PUSH D SAVE GRT ADDRESS
044.202 315 234 030 1192 CALL $INDL
044.205 043 000 1193 DW IOC,DIR+DIR,FGN (E) = DIR,FGN
044.207 173 1194 MOV A,E
044.210 341 1195 POP H (HL) = GRT TABLE ADDRESS
044.211 315 276 050 1196 CALL CFS, COMPUTE BLOCK SIZE
044.214 341 1197 POP H (HL) = ADDRESS OF FDN,STA
044.215 345 1198 PUSH H
044.216 176 1199 MOV A,M (A) = FDN,STA
044.217 366 020 1200 ORI ST,CNT FLAG CONTIG
044.221 167 1201 MOV M,A
000.000 1202 ERRNZ FDN,SIZ-FDN,STA-2
044.222 043 1203 INX H
044.223 043 1204 INX H (HL) = #FDN,SIZ
044.224 163 1205 MOV M,E SET BLOCK COUNT
1206
1207 * READY TO READ DATA. POSITION FILE (IN CASE SOME WAS READ IN
1208 * PREVIOUS PASSES) AND COMPUTE THE MAX POSSIBLE READ COUNT
1209 *
1210 * ((SP)) = ADDRESS OF FDN,STA FOR NODE
1211
044.225 341 1212 RPH3 POP H (HL) = ADDRESS OF FDN,STA
044.226 345 1213 PUSH H
044.227 315 234 030 1214 CALL $INDL
044.232 003 000 1215 DW FDN,AMR-FDN,STA (DE) = AMOUNT READ (IN SECTORS)
044.234 102 1216 MOV B,D

```



RPH

```

044.235 113 1217 MOV C,E (BC) = AMOUNT READ
044.236 076 000 1218 MVI A,CN,SOU
044.240 377 047 1219 DB SYSCALL,POSIT POSIT
044.242 332 205 047 1220 JC IERR3 POSIT BLEW UP
044.245 315 306 045 1221 CALL CBR COMPUTE BUFFER ROOM
044.250 353 1222 XCHG (D) = POINTER/256, (E) = LIMIT/256
044.251 341 1223 POP H (HL) = #FDN,STA
044.252 001 007 000 1224 LXI B,FDN,ADR-FDN,STA
044.255 011 1225 DAD B (HL) = #FDN,ADR
044.256 162 1226 MOV M,D SET ADDRESS/256
044.257 345 1227 PUSH H SAVE #FDN,ADR
044.260 036 000 1228 MVI E,0 (DE) = ADDRESS
044.262 107 1229 MOV B,A (B) = SECTORS OF RAM AVAILABLE
044.263 113 1230 MOV C,E (C) = 0
044.264 305 1231 PUSH B SAVE TRY COUNT
044.265 076 000 1232 MVI A,CN,SOU
044.267 377 004 1233 DB SYSCALL,READ READ THE STUFF
1234
1235 * COMPUTE THE AMOUNT READ (IN CASE OF EOF)
1236
044.271 321 1237 POP D (DE) = TRY COUNT
044.272 322 317 044 1238 JNC RPH4 GOT ALL WE TRYED
044.275 376 001 1239 CPI EC,EOF
044.277 302 153 047 1240 JNE NAMERR NOT JUST EOF, GOT TROUBLES
044.302 172 1241 MOV A,B
044.303 220 1242 SUB B REMOVE AMOUNT WE DIDNT GET
044.304 127 1243 MOV D,A
044.305 341 1244 POP H (HL) = #FDN,ADR
044.306 345 1245 PUSH H
044.307 001 371 377 1246 LXI B,FDN,STA-FDN,ADR
044.312 011 1247 DAD B
044.313 176 1248 MOV A,M (A) = FDN,STA
044.314 346 375 1249 ANI 3770-ST,OPR EOF, NOT OPEN FOR READ ANYMORE
044.316 167 1250 MOV M,A POST READ COMPLETE FOR THIS GUY
1251
1252 * STORE RESULTS OF READ IN NODE
1253 *
1254 * (D) = SECTORS READ
1255 * ((SP)) = #FDN,ADR
1256
044.317 341 1257 RPH4 POP H (HL) = #FDN,ADR
044.320 043 1258 INX H
000.000 1259 ERRNZ FDN,AIM-FDN,ADR-1 (HL) = ADDRESS IF AMOUNT IN MEMORY BYTE
044.321 162 1260 MOV M,D STORE SECTORS IN MEMORY COUNT
044.322 001 373 377 1261 LXI B,FDN,AMR-FDN,AIM
044.325 011 1262 DAD B (HL) = #FDN,AMR (AMOUNT READ)
044.326 176 1263 MOV A,M (A) = AMOUNT READ BEFORE
044.327 202 1264 ADD D ADD NEW AMOUNT
044.330 167 1265 MOV M,A
044.331 043 1266 INX H
044.332 176 1267 MOV A,M
044.333 316 000 1268 ACI 0 PROPIGATE FOR VERY LARGE FILES
044.335 167 1269 MOV M,A
044.336 041 273 054 1270 LXI H,ORUFPTR
044.341 176 1271 MOV A,M
044.342 202 1272 ADD D ADVANCE FREE RAM POINTER BY AMOUNT READ

```

```

044.343 167      1273      MOV      M,A
044.344 076 000  1274      MVI      A,CN,SOU
044.346 377 046  1275      DB       SYSCALL,,CLOSE  CLOSE FILE
044.350 303 375 043 1276      JMP      RPH          SEE IF MORE TO READ
  
```

```

1278 **      WPH = WRITE PHASE.
1279 *
1280 *      WPH HANDLES THE WRITE PHASE PROCESSING. IT IS ENTERED WITH
1281 *      THE FDN CHAIN SETUP, THE NAMTAB SETUP, AND
1282 *      THE DESTINATION DISK MOUNTED.
1283 *
1284 *
1285 *      WRITE PHASE
1286 *
1287 *      DURING THE WRITE PHASE, THE DESTINATION DISK IS MOUNTED. THE NODES
1288 *      ARE TAKEN FROM THE ACTIVE CHAIN, AND PROCESSED. IF THE FILE HAD
1289 *      BEEN PARTIALLY WRITTEN THE LAST PASS, IT IS RE-OPENED AND POSITIONED.
1290 *      IF THERE IS NOT MORE DATA TO READ FOR A PROCESSED
1291 *      NODE, IT IS REMOVED, AND THE CORRESPONDING ENTRY IN NAMTAB IS DELETED.
1292 *
1293 *      WRITE PHASE CONTINUES UNTIL
1294 *
1295 *      1) THERE ARE NO MORE FILE NODES IN THE ACTIVE LIST
1296 *      2) OR, THE FIRST (AND ONLY) ENTRY IN THE LIST HAS NO
1297 *      MORE DATA IN MEMORY, BUT HAS NOT BEEN COMPLETELY READ.
  
```

```

1298 *
1299 *      ENTRY  NONE
1300 *      EXIT   NONE
1301 *      USES   ALL
1302
1303
044.353      1304 WPH      EQU      *
1305
1306 *      SEE IF MORE TO WRITE
1307
044.353 041 147 054 1308      LXI      H,FDNHEAD
044.356 156      1309      MOV      L,H
044.357 175      1310      MOV      A,L          (A) = FIRST NODE INDEX
044.360 247      1311      ANA      A
044.361 310      1312      RZ              NO MORE
044.362 315 234 030 1313      CALL   $INDL
044.365 011 000  1314      DW      FDN,AIM      (E) = AMOUNT IN MEMORY FOR THIS GUY
044.367 173      1315      MOV      A,E
044.370 247      1316      ANA      A
044.371 302 006 045 1317      JNZ     WPHO          GOT DATA
1318
1319 *      NO DATA IN NODE. IF STILL READING, RETURN FOR MORE
1320
044.374 043      1321      INX      H
044.375 176      1322      MOV      A,M
044.376 053      1323      DCX      H
044.377 346 002  1324      ANI      ST,OPR
045.001 300      1325      RNZ              STILL READING, GET MORE
  
```

```

045.002 353 1326 XCHG (DE) = ADDRESS
045.003 303 230 045 1327 JMP WPH4 REMOVE NODE, AM DONE WITH FILE
1328
1329 * HAVE DATA TO WRITE, SEE IF WE HAVE OPENED THIS FILE BEFORE.
1330 * OR IF THIS IS THE FIRST TIME
1331
045.006 345 1332 WPH0 PUSH H SAVE NODE POINTER
045.007 043 1333 INX H
000.000 1334 ERRNZ FDN,STA-1
045.010 176 1335 MOV A,M (A) = FDN,STA
045.011 346 001 1336 ANI ST,OPW
045.013 302 123 045 1337 JNZ WPH2 OPENED BEFORE
000.000 1338 ERRNZ ST,OPW-1
045.016 064 1339 INR M SET '1' BIT
1340
1341 * BUILD NAME INTO DESTFB
1342
045.017 345 1343 PUSH H SAVE NODE ADDRESS
045.020 001 303 043 1344 LXI B,OCOPYD
045.023 021 327 055 1345 LXI D,NAFTAR
045.026 041 125 055 1346 LXI H,DESTFB+FB,NAM
045.031 315 201 053 1347 CALL MWN MERGE WILDCARD NAME
045.034 341 1348 POP H
1349
1350 * IS 1ST TIME FOR THIS FILE, IF CONTIGUOUS FLAG, OPEN THE FILE
1351 * FOR CONTIGUOUS
1352
045.035 176 1353 MOV A,M (A) = FLAG BYTE
045.036 346 020 1354 ANI ST,CNT
045.040 302 060 045 1355 JNZ WPH1 IS CONTIG
045.043 041 125 055 1356 LXI H,DESTFB+FB,NAM
045.046 076 001 1357 MVI A,CN,DES
045.050 377 043 1358 DB SYSCALL,.OPENW JUST OPEN FOR WRITE
045.052 332 165 047 1359 JC DESTERR ERROR
045.055 303 155 045 1360 JMP WPH3 WRITE THE DATA
1361
1362 * IS CONTIG FILE, OPEN IN CONTIG MODE
1363
045.060 043 1364 WPH1 INX H
045.061 043 1365 INX H (HL) = #FDN,STA
000.000 1366 ERRNZ FDN,SIZ-FDN,STA-2
045.062 116 1367 MOV C,M (C) = COUNT (IN BLOCKS)
045.063 006 000 1368 MVI B,0
045.065 041 125 055 1369 LXI H,DESTFB+FB,NAM
045.070 076 001 1370 MVI A,CN,DES
045.072 305 1371 PUSH B SAVE COUNT
045.073 377 050 1372 DB SYSCALL,.DELET DELETE OLD ONE
045.075 322 105 045 1373 JNC WPH1.5 DELETED
045.100 376 014 1374 CPI EC,FNF
045.102 302 007 050 1375 JNE ERROR MUST BE WRITE PROTECTED, OR SOMETHING...
045.105 301 1376 WPH1.5 POP B (BC) = COUNT
045.106 041 125 055 1377 LXI H,DESTFB+FB,NAM
045.111 076 001 1378 MVI A,CN,DES
045.113 377 045 1379 DB SYSCALL,.OPENC OPEN CONTIG
045.115 332 165 047 1380 JC DESTERR
045.120 303 155 045 1381 JMP WPH3

```

```

1382
1383 * THIS FILE HAS ALREADY BEEN PARTIALLY WRITTEN. OPEN IN UPDATE MODE
1384 * SO WE CAN EXTEND IT.
1385
045.123 041 125 055 1386 WPH2 LXI H,DESTFB+FB.NAM
045.126 076 001 1387 MVI A,CN.DES
045.130 377 044 1388 DB SYSCALL,,OPENU OPEN FOR UPDATE
045.132 332 165 047 1389 JC DESTERR PROBLEMS
045.135 341 1390 POP H
045.136 345 1391 PUSH H (HL) = #FDN.STA
045.137 315 234 030 1392 CALL $INDL
045.142 006 000 1393 DW FDN.AMW (DE) = AMOUNT WRITTEN
045.144 102 1394 MOV B,D
045.145 113 1395 MOV C,E (BC) = SECTORS WRITTEN
045.146 076 001 1396 MVI A,CN.DES
045.150 377 047 1397 DB SYSCALL,,POSIT POSITION FOR EXTEND
045.152 332 173 047 1398 JC IERR1 COULDN'T GET THERE!
1399
1400 * FILE OPEN AND POSITIONED. WRITE DATA
1401
045.155 341 1402 WPH3 POP H
045.156 345 1403 PUSH H (HL) = #FDN.LNK
045.157 315 234 030 1404 CALL $INDL
045.162 010 000 1405 DW FDN.ADR (E) = ADDR/256, (D) = CNT/256
045.164 102 1406 MOV B,D
045.165 123 1407 MOV D,E
045.166 036 000 1408 MVI E,0 (DE) = ADDRESS
045.170 113 1409 MOV C,E (BC) = COUNT
045.171 076 001 1410 MVI A,CN.DES
045.173 305 1411 PUSH B SAVE WRITE COUNT
045.174 377 005 1412 DB SYSCALL,,WRITE WRITE IT
045.176 332 165 047 1413 JC DESTERR PROBABLY OUT OF ROOM
045.201 076 001 1414 MVI A,CN.DES
045.203 377 046 1415 DB SYSCALL,,CLOSE CLOSE IT
045.205 332 165 047 1416 JC DESTERR
045.210 301 1417 POP B (B) = SECTORS WRITTEN
045.211 341 1418 POP H
045.212 345 1419 PUSH H (HL) = #FDN.LNK
045.213 021 006 000 1420 LXI D,FDN.AMW-FDN.LNK
045.216 031 1421 DAD D (HL) = FDN.AMW
045.217 176 1422 MOV A,M
045.220 200 1423 ADD B
045.221 167 1424 MOV M,A
045.222 043 1425 INX H
045.223 176 1426 MOV A,M
045.224 316 000 1427 ACI 0 INCREMENT AMOUNT WRITTEN
045.226 167 1428 MOV M,A
1429
1430 * CLEAR 'IN MEMORY' COUNT IN NODE. IF THE FILE HAS NO MORE TO
1431 * READ, REMOVE IT FROM THE CHAIN AND NANTAB
1432
045.227 321 1433 POP D (DE) = FDN.LNK
045.230 041 011 000 1434 WPH4 LXI H,FDN.AIM
045.233 031 1435 DAD D
045.234 066 000 1436 MVI M,0 CLEAR AMOUNT IN MEMORY
045.236 353 1437 XCHG (HL) = FDN.LNK

```

```

045.237 043          1438      INX      H
000.000            1439      ERRNZ   FDN,STA-FDN,LNK-1
045.240 176          1440      MOV     A,M          (A) = FDN,STA
045.241 346 002     1441      ANI     ST,DPR
045.243 300          1442      RNZ     STILL READING, AM DONE FOR THIS PHASE
000.000            1443      ERRNZ   FDN,FLG-FDN,STA-1
045.244 043          1444      INX     H          (HL) = #FDN,FLG
045.245 104          1445      MOV     B,M          (B) = FILE FLAGS
045.246 305          1446      PUSH   B          SAVE
  
```

1447  
 1448 \* UNLINK NODE FROM LIST  
 1449

```

045.247 053          1450      DCX     H
045.250 053          1451      DCX     H
000.000            1452      ERRNZ   FDN,LNK-FDN,FLG+2      (HL) = #FDN,LNK
045.251 176          1453      MOV     A,M
045.252 062 147 054 1454      STA   FDNHEAD      UNLINK FROM ACTIVE LIST
045.255 072 146 054 1455      LDA   FDNFRE
045.260 167          1456      MOV     M,A          PUT THIS GUY ON HEAD OF FREE LIST
045.261 175          1457      MOV     A,L
045.262 062 146 054 1458      STA   FDNFRE
045.265 315 255 053 1459      CALL  REN          REMOVE ENTRY FROM NAMTAB
  
```

1460  
 1461 \* FILE IS COMPLETED. NOW WE CAN  
 1462 \* SET SPECIAL FLAGS: SWL  
 1463

```

045.270 301          1464      POP     B          (B) = FLAGS
045.271 016 377      1465      MVI     C,3770      SET AS MANY AS ALLOWED
045.273 041 125 055 1466      LXI     H,DESTFB+FB,NAM
045.276 377 060      1467      DB
045.300 332 165 047 1468      JC     DESTERR
045.303 303 353 044 1469      JMP     WPH          TRY TO WRITE THE NEXT GUY
  
```

1471 \*\* CBR - COMPUTE BUFFER ROOM.  
 1472 \*  
 1473 \* CBR COMPUTES THE NUMBER OF SECTORS WORTH OF RAM  
 1474 \* STILL FREE.  
 1475 \*  
 1476 \* ENTRY NONE  
 1477 \* EXIT (A) = SECTORS OF RAM FREE  
 1478 \* Z: SET IFF (A) = 0  
 1479 \* (H) = BUFPTR/256  
 1480 \* (L) = DBUFLIM/256  
 1481 \* USES A,F  
 1482  
 1483

```

045.306 052 272 054 1484 CBR  LHLD  DBUFLIM
000.000            1485      ERRNZ  DBUFFTR-DBUFLIM-1
045.311 175          1486      MOV     A,L
045.312 224          1487      SUB     H
045.313 311          1488      RET
  
```

```

1490 **      IFL - INITIALIZE FDN LIST.
1491 *
1492 *      IFL CHAINS ALL THE FDN NODES TO THE FREE LIST. THIS
1493 *      CLEANUP IS NECESSARY IN CASE A CTL-C OR SOMETHING
1494 *      LEFT THE LIST GARBAGED.
1495 *
1496 *      ENTRY  NONE
1497 *      EXIT   NONE
1498 *      USES  ALL
1499
1500
045.314 041 150 054 1501 IFL LXI  H,FDN,1
045.317 175          1502      MOV  A,L          (A) = FIRST LINK
045.320 062 146 054 1503      STA  FDNFRE
045.323 257          1504      XRA  A
045.324 062 147 054 1505      STA  FDNHEAD  NONE IN LIST
045.327 006 007     1506      MVI  B,FDNCNT-1 (B) = NUMBER OF NODES-1
045.331 076 012     1507 IFL1 MVI  A,FDNELEN
045.333 205          1508      ADD  L          (A) = #ADDR OF NEXT NODE
045.334 167          1509      MOV  M,A          SET LINK
045.335 157          1510      MOV  L,A          FORWARD TO NEXT LINK
045.336 005          1511      DCR  B
045.337 302 331 045 1512      JNZ  IFL1        MORE TO GO
045.342 066 000     1513      MVI  M,0          LAST ONE CHAINS NOWHERE
045.344 311          1514      RET

1516 **      MAD - MOUNT ALTERNATE DISK.
1517 *
1518 *      MAD DISMOUNTS THE CURRENT DISK, HAS THE USER INSERT THE
1519 *      OTHER DISK, AND MOUNTS IT.
1520 *
1521 *      ENTRY  (B) = FRONT PANEL LED PATTERN
1522 *            (DE) = PROMPT PATTERNS FOR PANEL AND CONSOLE
1523 *      EXIT  (HL) = #VOLFLAG
1524 *            (LABEL?) = LABEL SECTOR
1525 *      USES  ALL
1526
1527
045.345          1528 MAD  EQU  *
1529
1530 *      DISMOUNT CURRENT DISK
1531
045.345 325          1532      PUSH D
045.346 305          1533      PUSH B          SAVE ENTRY PARAMETERS IN CASE OF RETRY
045.347 325          1534      PUSH D
045.350 305          1535      PUSH B          SAVE ENTRY PARAMETERS OVER SYDD CALL
045.351 041 157 046 1536      LXI  B,MNDA      DEVICE SPECIFICATION
045.354 377 203     1537      DB   SYSCALL,.DMNMS DISMOUNT WITHOUT MESSAGE
045.356 332 007 050 1538      JC   ERROR      IF ERROR
1539
1540 *      SETUP PROMPT ON FP LEDS AND CONSOLE FOR NEW DISK
1541
045.361 363          1542 MAD0 DI

```

```

045.362 041 243 040 1543 LXI H,D,DLYMD
045.365 176 1544 MOV A,M
045.366 247 1545 ANA A
045.367 312 374 045 1546 JZ MAD1 DISK ALREADY STOPPED
045.372 066 001 1547 MVI M,1 STOP DISK VERY SOON
045.374 373 1548 MAD1 EI
045.375 076 203 1549 MVI A,00,BDU+00,CLK+00,HLY
045.377 062 010 040 1550 STA .MFLAG HALT DISPLAY UPDATE
046.002 041 013 040 1551 LXI H,.ALEDS
046.005 076 011 1552 MVI A,9
046.007 301 1553 POP B (B) = PERIOD PATTERN
046.010 160 1554 MAD2 MOV M,B SET PATTERN
046.011 043 1555 INX H
046.012 075 1556 DCR A
046.013 302 010 046 1557 JNZ MAD2 IF MORE TO BLANK
046.016 041 016 040 1558 LXI H,.ALEDS+3
046.021 001 003 000 1559 LXI B,3
046.024 321 1560 POP D (DE) = PROMPT LIST
046.025 315 252 030 1561 CALL $MOVE MOVE IN PROMPT PATTERN
046.030 353 1562 XCHG (HL) = PATTERN
046.031 377 003 1563 DB SYSCALL,.PRINT CONSOLE PROMPT
046.033 315 136 031 1564 CALL $TYPTX
046.036 207 1565 DB BELL+200R BEEP CONSOLE, TOO
046.037 076 144 1566 MVI A,100
046.041 315 140 002 1567 CALL .HORN BEEP A WARNING
1568
1569 * WAIT FOR SIGNAL THAT NEW DISK IS IN
1570
046.044 377 001 1571 MAD3 DB SYSCALL,.SCIN
046.046 322 057 046 1572 JNC MAD4 GOT A CHARACTER
046.051 333 360 1573 IN IP,PAD
046.053 074 1574 INR A
046.054 312 044 046 1575 JZ MAD3 NO REPLY THERE, EITHER
1576
1577 * GOT REPLY. GOBBLE EXTRA CHARACTERS FROM CONSOLE
1578
046.057 377 001 1579 MAD4 DB SYSCALL,.SCIN
046.061 322 057 046 1580 JNC MAD4
1581
1582 * READ NEW DISK'S LABEL
1583
046.064 315 164 046 1584 CALL GETLAB GET LABEL
046.067 332 007 050 1585 JC ERROR
1586
1587 * SEE IF LABEL CHANGED FROM BEFORE
1588
046.072 301 1589 POP B
046.073 321 1590 POP D RESTORE ENTRY PARAMETERS
046.074 041 271 054 1591 LXI H,VOLSER
046.077 072 000 027 1592 LDA LABEL+LAB.SER
046.102 276 1593 CMP H
046.103 302 115 046 1594 JNE MAD4.5 IS THE RIGHT DISK
046.106 325 1595 PUSH D SAVE AS AT THE BEGINNING
046.107 305 1596 PUSH B
046.110 325 1597 PUSH D SAVE FOR RETRY
046.111 305 1598 PUSH B
  
```

```

046.112 303 361 045 1599 JMP MAD0 TRY AGAIN
1600
046.115 167 1601 MAD4.5 MOV M,A SET NEW SERIAL
046.116 041 270 054 1602 LXI H,VOLFLAG
046.121 176 1603 MOV A,M
046.122 057 1604 CMA
046.123 167 1605 MOV M,A COMPLEMENT VOLUME FLAG
1606
1607 * ERASE FRONT PANEL DISPLAY
1608
046.124 041 013 040 1609 LXI H,ALEDS
046.127 076 011 1610 MVI A,9
046.131 160 1611 MAD5 MOV M,B SET TO PATTERN
046.132 043 1612 INX H
046.133 075 1613 DCR A
046.134 302 131 046 1614 JNZ MAD5
046.137 315 143 046 1615 CALL MND MOUNT NEW DISK
046.142 311 1616 RET
  
```

```

1618 ** MND - MOUNT SYSTEM DISK.
1619 *
1620 * MND MOUNTS A NEW DISK INTO 'SY' UNIT 'UNIT'
1621 *
1622 *
1623 * THE LABEL MUST ALREADY HAVE BEEN READ INTO 'LABEL'
1624 *
1625 * ENTRY NONE
1626 *
1627 * EXIT LABEL = LABEL SECTOR
1628 *
1629 * USES ALL
1630 *
1631
046.143 041 157 046 1632 MND LXI H,MNDA DEVICE SPECIFICATION
046.146 377 202 1633 DB SYSCALL,MONMS MOUNT WITHOUT MESSAGE
046.150 332 007 050 1634 JC ERROR IF ERROR
046.153 315 164 046 1635 CALL GETLAB GET LABEL
046.156 311 1636 RET
1637
046.157 123 131 060 1638 MNDA DB 'SY01',0
  
```

```

1640 ** GETLAB - GET LABEL
1641 *
1642 * GETLAB GETS THE LABEL FROM THE DISK
1643 *
1644 * ENTRY NONE
1645 *
1646 * EXIT LABEL IN LABEL
1647 * (PSW) = 'C' CLEAR IF NO ERROR
1648 * = 'C' SET IF ERROR
  
```



```

1649 *
1650 *      USES      ALL
1651 *
1652
046.164 041 011 000 1653 GETLAB LXI      H,DDF.LAB
046.167 021 000 027 1654 LXI      D,LABEL
046.172 001 000 001 1655 LXI      B,256
046.175 315 241 031 1656 CALL     $WER          WRITE ENABLE RAM
046.200 076 002      1657 MVI      A,DC.KER     READ REGARDLESS
046.202 315 130 040 1658 CALL     SYDD
046.205 311      1659 RET

1661 **      RDD - REQUIRE DATA DISK.
1662 *
1663 *      RDD CHECKS THE VOLUME TYPE TO MAKE SURE THAT IT IS A VALID
1664 *      DATA DISK.
1665 *
1666 *      ENTRY     NONE
1667 *      EXIT      TO CALLER IF OK
1668 *      TO EXIT IF BAD
1669 *      USES      ALL
1670
1671
046.206 315 164 046 1672 RDD     CALL     GETLAB     READ NEW DISK'S LABEL
046.211 332 007 050 1673 JC      ERROR
046.214 072 010 027 1674 LDA     LABEL+LAB,ULT  (A) = VOLUME TYPE
000.000      1675 ERRNZ   LAB.DAT
046.217 247      1676 ANA     A
046.220 310      1677 RZ
000.000      1678 ERRNZ   LAB.SYS-1     IS DATA DISK, OK
046.221 075      1679 BCR     A             SEE IF SYSTEM DISK
046.222 302 302 046 1680 JNZ     RDD1         DISK NOT EVEN INITIALIZED
046.225 315 136 031 1681 CALL     $TYPTX
046.230 012 007 124 1682 DB     NL,BELL,'This Disk Has Already Been SYSGENed.',ENL
046.277 303 332 042 1683 JMP     EXIT
1684
1685 *      DISK IS NOT PROPERLY INITIALIZED.
1686 *      (THIS CODE MAY BE ENTERED FROM OTHER ROUTINES)
1687
046.302 315 136 031 1688 RDD1   CALL     $TYPTX
046.305 012 007 124 1689 DB     NL,BELL,'This Disk Must be Re-Initialized Before It Can Be
046.371 123 131 123 1690 DB     'SYSGENed.',ENL
047.003 303 332 042 1691 JMP     EXIT

```



```

1716 **      ERROR PROCESSING ROUTINES
1717 *

1719 ***      NAMERR - FILE TYPE ERROR, OCCURED ON FILE WHOSE NAME
1720 *          IS NEXT UP IN NAMTAB.
1721 *
1722 *          PROCESS VIA $FERROR
1723
047.153 052 152 055 1724 NAMERR LHL D  NAMTPTK
047.154 001 366 377 1725      LXI  B,-FB,NAM
047.161 011          1726      DAD  B
047.162 303 274 054 1727      JMP  $FERROR

1729 **      ERROR ON FILE IN DESTFB
1730
047.165 041 113 055 1731 DESTERR LXI  H,DESTFB
047.170 303 274 054 1732      JMP  $FERROR

1734 **      INTERNAL ERRORS, SHOULD NOT OCCOUR,
1735
047.173 076 061 1736 IERR1  MVI  A,'1'
047.175 303 212 047 1737      JMP  INTERR
1738
047.200 076 062 1739 IERR2  MVI  A,'2'
047.202 303 212 047 1740      JMP  INTERR
047.205 076 063 1741 IERR3  MVI  A,'3'
047.207 303 212 047 1742      JMP  INTERR
1743
047.212 365 1744
1745 INTERR PUSH  PSW          SAVE CODE
047.213 315 136 031 1746      CALL $TYPTX
047.216 007 012 123 1747      DB  BELL,NL,'SYSGEN Internal Error ','$'+2000
047.247 361 1748      POP  PSW
047.250 315 127 054 1749      CALL $WCHAR
047.253 315 136 031 1750      CALL $TYPTX
047.254 012 124 150 1751      DB  NL,'This Error Should not Occur, Contact HEATH Technical'
047.343 012 103 157 1752      DB  NL,'Correspondence for Assistance.',NL
050.003 076 001 1753      MVI  A,1
050.005 377 000 1754      DB  SYSCALL,.EXIT          ABORT

```

```

1756 ** ERROR - GENERAL AND SYNTAX ERRORS NOT DIRECTLY ASSOCIATED
1757 * WITH A VALID FILE NAME.
1758
1759
050.007 365 1760 ERROR PUSH PSW SAVE CODE
050.010 315 136 031 1761 CALL $TYPTX
050.013 007 105 122 1762 DB BELL,'ERROR #','+2000
050.024 361 1763 POP PSW
050.025 247 1764 ANA A
050.026 372 040 050 1765 JM ERROR1 IS PRODUCT ERROR
050.031 046 012 1766 MVI H,NL USE NL AS MESSAGE TRAIL CHAR
050.033 377 057 1767 DB SYSCALL,'ERROR LOOK UP SYSTEM ERROR
050.035 303 327 042 1768 JMP RESTART
1769
1770 * IS PRODUCT ERROR
1771
050.040 041 100 050 1772 ERROR1 LXI H,ERRORA
050.043 276 1773 ERROR2 CMP H
050.044 043 1774 INX H
050.045 302 043 050 1775 JNE ERROR2 FIND ERROR MESSAGE
050.050 315 136 031 1776 CALL $TYPTX
050.053 007 123 131 1777 DB BELL,'SYSGEN Error #','+2000
050.073 377 003 1778 DB SYSCALL,'PRINT PRINT MESSAGE
050.075 303 327 042 1779 JMP RESTART
1780
050.100 1781 ERRORA DS 0 ERROR MESSAGES
050.100 200 060 061 1782 DB PEC.DF,'01',ENL
050.104 201 060 062 1783 DB PEC.INC,'02',ENL
050.110 202 060 063 1784 DB PEC.RSE,'03',ENL
050.114 203 060 064 1785 DB PEC.TFI,'04',ENL
050.120 204 060 065 1786 DB PEC.CS,'05',ENL
050.124 205 060 066 1787 DB PEC.IUW,'06',ENL
050.130 206 060 067 1788 DB PEC.IDF,'07',ENL
050.134 207 060 070 1789 DB 2070,'08',ENL
    
```

AEN

```

1793 **      AEN - ADD ENTRY TO 'NAMTAB'
1794 *
1795 *      AEN EXPANDS THE FILE INFO IN PIO.XXX INTO A FILE DESCRIPTOR
1796 *      AND ENTERS IT IN THE NAMTAB TABLE.
1797 *
1798 *      ENTRY  NONE
1799 *      EXIT   'C' SET IF WILDCARD
1800 *      USES   ALL
1801
1802
050.140 041 212 050 1803 AEN  LXI   H,AENA
050.143 315 124 052 1804      CALL  CDA          CONVERT DIRECTORY FORMAT TO ASCII FORMAT
050.146 326 001      1805      SUI   1          'C' SET IF WILDCARD
050.150 365      1806      PUSH  PSW          SAVE FLAG
050.151 052 146 055 1807      LHLD  NAMTLEN
050.154 001 021 000 1808      LXI   B,FB.NAML
050.157 011      1809      DAD   B          INCREASE SIZE
050.160 042 146 055 1810      SHLD  NAMTLEN
050.163 353      1811      XCHG          (DE) = NEW LENGTH
050.164 052 150 055 1812      LHLD  NAMTMAX
050.167 175      1813      MOV   A,L          SEE IF WILL OVERFLOW
050.170 223      1814      SUB   E
050.171 174      1815      MOV   A,H
050.172 232      1816      SBB   D
050.173 334 127 053 1817      CC   INA          INCREASE NAMTAB ALLOCATION
050.176 041 306 055 1818      LXI   H,NAMTAB-FB.NAML
050.201 031      1819      DAD   D          (HL) = *TO* ADDRESS
050.202 021 212 050 1820      LXI   D,AENA          (DE) = *FROM* ADDRESS
050.205 315 252 030 1821      CALL  $MOVE          MOVE ENTRY IN
050.210 361      1822      POP  PSW          (PSW) = WILDCARD FLAG
050.211 311      1823      RET
1824
050.212      1825 AENA  DS    FB.NAML

```

```

1827 **      BSL - BUILD SOURCE FILE LIST.
1828 *
1829 *      BSL CRACKS THE LIST OF THE SOURCE FILES FROM THE COMMAND LINE AND
1830 *      BUILDS THEM INTO THE NAMTAB MANAGED TABLE.
1831 *      WILD CARDS ENCOUNTERED ARE EXPANDED.
1832 *
1833 *      ENTRY  (A) 0 IF TO ASK ABOUT *.* USE
1834 *      EXIT   'C' CLEAR IF OK
1835 *           'C' SET IF ERROR
1836 *           (A) = CODE
1837 *      USES   ALL
1838
1839
050.233 062 275 050 1840 BSL  STA   BSLA          SAVE ASK FLAG
050.236 315 161 053 1841      CALL  LSN          LOCATE SOURCE NAME
1842
1843 *      GO THROUGH SOURCE LIST CRACKING NAMES
1844
050.241 176      1845 BSL1  MOV   A,M

```

BSL

```

050.242 247 1846 ANA A
050.243 310 1847 RZ
050.244 021 101 055 1848 LXI B,DEFAULT ALL DONE
050.247 315 107 051 1849 CALL CAD CONVERT ASCII NAME TO DIRECTORY FORMAT
050.252 330 1850 RC ERROR
050.253 315 340 053 1851 CALL SNB SET NEW DEFAULTS
050.256 345 1852 PUSH H SAVE LINE ADDRESS
050.257 315 215 052 1853 CALL EWS EXPAND WILDCARD SPECIFICATION
050.262 332 265 050 1854 JC IF ERROR
050.265 341 1855 BSL2 POP H RESTORE LINE ADDRESS
050.266 330 1856 RC USER REFUSED *.*
050.267 315 323 053 1857 CALL SFS SKIP FILE SEPERATOR (BLANKS AND/OR COMMA)
050.272 303 241 050 1858 JMP BSL1 DO MORE
1859
050.275 000 1860 BSLA DB 0 <>0 IF TO CHECK FOR *.*
  
```

```

1862 ** CFS - COMPUTE FILE SIZE
1863 *
1864 * CFS COMPUTES THE SIZE OF A FILE. THE DEVICE'S GRT MUST BE IN
1865 * THE 'GRT' BUFFER.
1866 *
1867 * ENTRY (A) = FIRST GROUP NUMBER
1868 * EXIT (DE) = SIZE
1869 * USES ALL
1870
1871
050.276 021 000 000 1872 CFS LXI D,0
050.301 247 1873 CFS1 ANA A
050.302 310 1874 RZ ALL DONE
050.303 157 1875 MOV L,A
050.304 176 1876 MOV A,M (A) = NEXT GRT
050.305 023 1877 INX D
050.306 303 301 050 1878 JMP CFS1 TRY AGAIN
  
```

```

1880 ** CSF - CHECK FOR SPECIAL FILE.
1881 *
1882 * CSF CHECKS TO SEE IF THE FILE NAME (IN DIRECTORY FORMAT)
1883 * SUPPLIED MATCHES ONE OF A LIST OF 'NOT-TO-BE-PROCESSED'
1884 * FILES. THE LIST IS:
1885 *
1886 * RGT.SYS
1887 * RGT.SYS
1888 * DIRECT.SYS
1889 *
1890 * ENTRY (DE) = ADDRESS OF DIRECTORY BLOCK
1891 * EXIT 'Z' SET IF MATCH
1892 * 'Z' CLEAR OTHERSIZE
1893 * USES A,F
1894
1895
  
```

050.311	305		1896	CSF	PUSH	B		
050.312	325		1897		PUSH	D		
050.313	345		1898		PUSH	H	SAVE POINTERS	
			1899					
050.314	041	352	050	1900	LXI	H,CSFA	(A) = START OF LIST	
050.317	325			1901	CSF1	PUSH	D	SAVE NAME
050.320	345			1902		PUSH	H	SAVE LIST ADDRESS
050.321	016	015		1903		MVI	C,DIRIDL	
050.323	315	060	030	1904		CALL	\$COMP	SEE IF MATCH
050.324	341			1905		POP	H	
050.327	321			1906		POP	D	
050.330	312	346	050	1907		JE	CSF2	GOT MATCH
050.333	076	015		1908		MVI	A,DIRIDL	
050.335	315	101	030	1909		CALL	\$DADA.	POINT TO NEXT ENTRY
050.340	176			1910		MOV	A,H	
050.341	247			1911		ANA	A	
050.342	302	317	050	1912		JNZ	CSF1	MORE TO CHECK
				1913				
				1914	*		NO MATCH	
				1915				
050.345	074			1916		INR	A	CLEAR 'Z'
050.346	341			1917	CSF2	POP	H	
050.347	321			1918		POP	B	RESTORE REGS
050.350	301			1919		POP	B	
050.351	311			1920		RET		
				1921				
050.352	107	122	124	1922	CSFA	DB	'GRT',0,0,0,0,0,'SYS',0,0	GRT.SYS
000.000				1923		ERKNZ	*-CSFA-DIRIDL	ENTRYS MUST BE 'DIRIDL' LONG
050.367	122	107	124	1924		DB	'RGT',0,0,0,0,0,'SYS',0,0	RGT.SYS
051.004	104	111	122	1925		DB	'DIRECT',0,0,'SYS',0,0	
051.021	000			1926		DB	0	END OF TABLE
				1928	**		CWM - CHECK WILDCARD MATCH.	
				1929	*			
				1930	*		CWM CHECKS TO SEE IF A WILDCARDED FIELD MATCHES A NON-WILDCARDED	
				1931	*		FIELD.	
				1932	*			
				1933	*	ENTRY	(DE) = ADDRESS OF WC NAME	
				1934	*		(HL) = ADDRESS OF NON/WC NAME	
				1935	*		(B) = NUMBER OF CHARACTERS TO CHECK	
				1936	*	EXIT	'Z' SET IF MATCH	
				1937	*		(HL) = (HL)+(B)	
				1938	*		(DE) = (DE) - (B)	
				1939	*		'Z' CLEAR IF NO MATCH	
				1940	*	USES	A,F,B,D,E,H,L	
				1941				
				1942				
051.022	032			1943	CWM	LBAX	D	
051.023	247			1944		ANA	A	
051.024	372	031	051	1945		JM	CWM1	IS MATCH
051.027	276			1946		CMF	M	
051.030	300			1947		RNE		NO MATCH
051.031	023			1948	CWM1	INX	D	

051.032	043		1949		INX	H	ADVANCE ADDRESSES
051.033	005		1950		DCR	B	
051.034	302	022	051	1951	JNZ	CWM	GO FOR MORE
051.037	311		1952		RET		GOT MATCH
			1954	**			DDF - DECODE DESTINATION FILE.
			1955	*			
			1956	*			DDF DECODES THE DESTINATION FILE NAME FROM THE COMMAND LINE.
			1957	*			
			1958	*			IF NO DESTINATION NAME IS SPECIFIED, IT DEFAULTS TO
			1959	*			
			1960	*			KB:PIPDST.JGL
			1961	*			
			1962	*	ENTRY	NONE	
			1963	*	EXIT	'C' CLEAR IF OK	
			1964	*			(A) = 0 IF NAME HAS WILDCARDS
			1965	*			(A) = 1 IF NO WILDCARD USED
			1966	*			DESTFB+FB.NAM CONTAINS A COMPLETE DESTINATION FILE NAME
			1967	*			(HL) = COMMAND LINE POINTER UPDATED
			1968	*			'C' SET IF ERROR
			1969	*			(A) = CODE
			1970	*	USES	ALL	
			1971				
			1972				
051.040	041	346	042	1973	DDF	LXI	H,LINE
				1974			
				1975	*		(HL) = ADDRESS FOR NAME
				1976			
051.043	021	101	055	1977	DDF2	LXI	D,DEFAULT
051.046	315	107	051	1978	CALL	CAD	CONVERT ASCII NAME TO DIRECTORY FORMAT
051.051	330			1979	RC		ERROR
051.052	176			1980	MOV	A,M	
051.053	376	075		1981	CFI	'='	
051.055	076	206		1982	MVI	A,PEC.IDF	ASSUME ILLEGAL DESTINATION FORMAT
051.057	067			1983	STC		
051.060	300			1984	RNE		MUST HAVE '='
				1985			
				1986	*		HAVE NAME DECODED. EXPAND INTO DESTFB+FB.NAM
				1987			
051.061	041	125	055	1988	LXI	H,DESTFB+FB.NAM	
051.064	303	124	052	1989	JMP	CDA	CONVERT DIRECTORY FORMAT TO ASCII
				1990			
051.067	124	124	072	1991	DDFA	DB	'TT:RIFBEST.JGL=' ,0



```

1993 ** CAD - CONVERT ASCII FILE NAME INTO DIRECTORY FORMAT.
1994 *
1995 * CAD CRACKS AN ALPHANUMERIC FILE DESCRIPTION, OF THE FORM
1996 *
1997 * DEV:NAME.EXT
1998 *
1999 * INTO THE PIO.XXX FIELDS.
2000 *
2001 * THE DEFAULT BLOCK DETERMINES THE VALUES FOR THE DEVICE AND EXTENSION
2002 * FIELDS, IF THEY ARE UNSPECIFIED, IF *CAD* IS ENTERED
2003 * AT *CAD*, AN UNSPECIFIED NAME FIELD IS RETURNED AS ZERO BYTES.
2004 * IF ENTERED AT *CAD.*, AN UNSPECIFIED NAME FIELD IS
2005 * RETURNED AS 2000 (MATCH-ONE) BYTES.
2006 *
2007 * ENTRY (DE) = POINT TO DEFAULT BLOCK
2008 * (HL) = POINTER TO TEXT
2009 * EXIT 'C' SET IF ERROR
2010 * (A) = ERROR CODE
2011 * 'C' CLEAR IF OK
2012 * (HL) = POINTS PAST FILE NAME
2013 * 'Z' SET IF NULL NAME
2014 * 'Z' CLEAR IF NON-NULL
2015 * PIO.DIR.NAM = NAME
2016 * PIO.DIR.EXT = EXTENSION
2017 * PIO.DEV = DEVICE CODE
2018 * PIO.UNI = UNIT NUMBER (ASCII DIGIT)
2019 * USES ALL
2020
2021
2022 CAD XRA A SET TO NULLS
051.107 257
051.110 303 115 051 2023 JMF CADO
2024
051.113 076 200 2025 CAD. MVI A,2000
051.115 345 2026 CADO PUSH H
051.116 062 354 051 2027 STA CADA SAVE DEFAULT VALUE
2028
2029 * SET DEFAULTS IN PIO.XXX
2030
051.121 041 275 055 2031 LXI H,PIO.DEV
051.124 001 003 000 2032 LXI B,3
051.127 315 252 030 2033 CALL $MOVE SET DEFALUT DEVICE
051.132 001 003 000 2034 LXI B,3
051.135 041 310 055 2035 LXI H,PIO.DIR+DIR.EXT
051.140 315 252 030 2036 CALL $MOVE SET DEFAULT EXTENSION
051.143 341 2037 POP H
051.144 315 370 053 2038 CALL $SOB SKIP BLANKS
051.147 006 000 2039 MVI B,0
051.151 376 077 2040 CFI '?'
051.153 312 202 051 2041 JE CAD1 IS '?'
051.156 376 052 2042 CFI '*'
051.160 312 202 051 2043 JE CAD1 IS '*'
051.163 376 056 2044 CFI '/'
051.165 312 202 051 2045 JE CAD1 IS '/'
051.170 376 101 2046 CFI 'A'
051.172 332 342 051 2047 JC CADA NOT NAME
051.175 376 133 2048 CFI 'Z'+1

```

```

051.177 322 342 051 2049 JNC CAD4 NOT NAME
2050
2051 * HAVE ALPHA STRING. CRACK IT
2052
051.202 315 355 051 2053 CAD1 CALL DNT DECODE NEXT TOKEN
051.205 332 350 051 2054 JC CAD5 ERROR
051.210 376 072 2055 CPI ' '
051.212 302 245 051 2056 JNE CAD2 NOT DEVICE
2057
2058 * HAVE EXPLICIT DEVICE
2059
051.215 043 2060 INX H SKIP ':'
051.216 076 003 2061 MVI A,3
051.220 271 2062 CMP C
051.221 332 350 051 2063 JC CAD5 TOO MANY CHARACTERS
051.224 001 003 000 2064 LXI B,3
051.227 345 2065 PUSH H SAVE (HL)
051.230 041 275 055 2066 LXI H,PIO.DEV
051.233 315 252 030 2067 CALL $MOVE SET EXPLICIT DEVICE
051.236 341 2068 POP H
051.237 315 355 051 2069 CALL DNT DECODE NEXT TOKEN
051.242 332 350 051 2070 JC CAD5 ERROR
2071
2072 * DECODE NAME
2073
051.245 001 010 000 2074 CAD2 LXI B,8 (BC) = COUNT
051.250 345 2075 PUSH H SAVE TEXT ADDR
2076
2077 * SEE IF NAME IS UNSPECIFIED
2078
051.251 041 300 055 2079 LXI H,PIO.DIR+DIR.NAM
051.254 345 2080 PUSH H SAVE ADDRESS OF DIR.NAM
051.255 315 252 030 2081 CALL $MOVE MOVE IN NAME
051.260 341 2082 POP H (HL) = #PIO.DIR+DIR.NAM
051.261 176 2083 MOV A,M
051.262 247 2084 ANA A
051.263 302 301 051 2085 JNZ CAD2.6 IS SPECIFIED
051.266 072 354 051 2086 LDA CAD4 (A) = FILL CHARACTER
051.271 016 010 2087 MVI C,8 (C) = COUNT
051.273 167 2088 CAD2.4 MOV M,A
051.274 043 2089 INX H
051.275 015 2090 DCR C
051.276 302 273 051 2091 JNZ CAD2.4
051.301 341 2092 CAD2.6 POP H (A) = DELIMITER
051.302 176 2093 MOV A,M
051.303 376 056 2094 CPI ' '
051.305 302 340 051 2095 JNE CAD3 NOT EXTENSION
2096
2097 * HAVE EXPLICIT EXTENSION
2098
051.310 043 2099 INX H
051.311 315 355 051 2100 CALL DNT
051.314 332 350 051 2101 JC CAD5 ERROR
051.317 076 003 2102 MVI A,3
051.321 271 2103 CMP C
051.322 332 350 051 2104 JC CAD5 TOO LONG

```

```

051.325 001 003 000 2105 LXI B,3
051.330 345 2106 PUSH H SAVE TEXT POINTER
051.331 041 310 055 2107 LXI H,P10.DIR+DIR.EXT
051.334 315 252 030 2108 CALL #MOVE MOVE EXTENSION
051.337 341 2109 POP H
2110
2111 * DONE WITH NAME. MUST HAVE LEGIT DELIMITER
2112
051.340 006 001 2113 CAD3 MVI B,1 (B) = NAME PRESENT FLAG
2114
2115 * END OF NAME. EXIT
2116 * (B) = 0 IF NULL, (B) <> 0 IF NON-NULL
2117
051.342 315 370 053 2118 CAD4 CALL $SOB SKIP BLANKS
051.345 170 2119 MOV A,B
051.346 247 2120 ANA A SET 'Z' IF NULL
051.347 311 2121 RET
2122
2123 * ERROR
2124
051.350 076 007 2125 CAD5 MVI A,EC.IFN ILLEGAL FILE NAME
051.352 067 2126 STC
051.353 311 2127 RET
2128
051.354 000 2129 CADA DB 0 FILL CHARACTER FOR OMITTED NAME FIELD
2130
2131 ** DNT - DECODE NEXT TOKEN,
2132 *
2133 * DNT COPIES THE NEXT ALPHANUMERIC FIELD INTO A ZERO-FILLED WORK AREA.
2134 *
2135 * ENTRY (HL) = TEXT POINTER
2136 * EXIT 'C' SET IF ERROR
2137 * 'C' CLEAR IF OK
2138 * (A) = DELIMITER CHARACTER
2139 * (HL) UPDATED TO DELIMITER CHARACTER
2140 * (DNTA) = STRING
2141 * (C) = LENGTH
2142 * (DE) = #DNTA
2143 * USES ALL
2144
2145
051.355 021 067 052 2146 DNT LXI D,DNTA
051.360 016 011 2147 MVI C,9 (C) = SIZE OF DNTA
051.362 101 2148 MOV B,C (B) = MAX ALLOWED +1
051.363 257 2149 XRA A
051.364 022 2150 DNT1 STAX D ZERO BUFFER
051.365 023 2151 INX D
051.366 015 2152 DCR C
051.367 302 364 051 2153 JNZ DNT1
051.372 021 067 052 2154 LXI D,DNTA
2155
2156 * COPY CHARACTERS
2157

```

```

051.375 176      2158 DNT2  MOV  A,H
051.376 376 077  2159      CPI  '?'
052.000 076 200  2160      MVI  A,2000
052.002 312 037 052 2161      JE   DNT3      IS MATCHONE
052.005 176      2162      MOV  A,H
052.006 376 052  2163      CPI  '*'
052.010 312 051 052 2164      JE   DNT5      IS WILDCARD
052.013 376 040  2165      CPI  '0'
052.015 332 062 052 2166      JC   DNT4      NOT ALPHANUMERIC
052.020 376 072  2167      CPI  '9'+1
052.022 332 037 052 2168      JC   DNT3      NUMERIC
052.025 376 101  2169      CPI  'A'
052.027 332 062 052 2170      JC   DNT4      DELIMITER
052.032 376 133  2171      CPI  'Z'+1
052.034 322 062 052 2172      JNC  DNT4      DELIMITER
                2173
                2174 *   HAVE GOOD CHARACTER
                2175
052.037 022      2176 DNT3  STAX  D      STORE CHAR
052.040 023      2177      INX  D
052.041 043      2178      INX  H
052.042 014      2179      INR  C      COUNT
052.043 005      2180      DCR  B      LIMIT DECREMENT
052.044 302 375 051 2181      JNZ  DNT2      NOT OVERFLOW
                2182
                2183 *   OVERFLOW
                2184
052.047 067      2185      STC          FLAG ERR
052.050 311      2186      RET
                2187
                2188 *   IS '*' WILDCARD
                2189
052.051 076 200  2190 DNT5  MVI  A,2000
052.053 022      2191      STAX D
052.054 023      2192      INX  D
052.055 005      2193      DCR  B
052.056 302 051 052 2194      JNZ  DNT5      FILL WITH MATCH ONE
052.061 043      2195      INX  H      SKIP '*'
                2196
                2197 *   END OF STRING
                2198
052.062 247      2199 DNT4  ANA  A      CLEAR 'C'
052.063 021 067 052 2200      LXI  D,DNT4    SET POINTER
052.066 311      2201      RET
                2202
052.067      2203 DNTA  DS   9      WORK AREA

```

```

2205 **      EBM - EXPAND BUFFER TO MAXIMUM.
2206 *
2207 *      EBM IS CALLED TO EXPAND THE BUFFER 'BUF' TO THE MAXIMUM SIZE.
2208 *      WHICH DOES NOT REQUIRE THE OVERLAYING OF THE SYSTEM.
2209 *
2210 *      ENTRY  NONE
2211 *      EXIT   (BUFSIZ) = BUFFER SIZE (MULTIPLE OF 256)
2212 *      USES  ALL
2213 *
2214 *
052.100 052 320 040 2215 EBM  LHLD  S,SYSM
052.103 021 366 377 2216      LXI  D,-10
052.106 031          2217      DAD  D          THROW IN SOME SLOP
052.107 377 052     2218      DB   SYSCALL,SETTP
052.111 332 173 047 2219      JC   IERR1      NOT ENOUGH MEMORY
052.114 052 322 040 2220      LHLD  S,USRM
2221 *
052.117 174        2222      MOV  A,H          (A) = LIMIT/256
052.120 062 272 054 2223      STA  OBUFLIM     SET LIMIT
052.123 311        2224      RET

2226 **      CDA - CONVERT DIRECTORY FORMAT TO ASCII.
2227 *
2228 *      CDA COPIES A DIRECTORY ENTRY FROM PIO,XXX TO A TARGET FIELD.
2229 *      THE DEVICE SPECIFICATION (IN PIO.DEV AND PIO.UNI) IS ALSO ENCODED.
2230 *      THE TARGET FIELD IS LEFT IN THE FORM:
2231 *
2232 *      DEV:NAME,XXX <00>
2233 *
2234 *      ENTRY  (HL) = FWA NAME FIELD
2235 *      EXIT   (A) = 0, HAVE WILDCARD
2236 *           = 1, NO WILDCARDS USED
2237 *           'C' CLEAR
2238 *      USES  ALL
2239 *
2240 *
052.124 001 000 003 2241 CDA  LXI  B,3*256      (B) = CHARACTER COUNT, (C) = WILDCARD FLAG
052.127 021 275 055 2242      LXI  D,PIO.DEV
052.132 315 170 052 2243      CALL CDAS      COPY IT
052.135 066 072     2244      MVI  M,'.'
052.137 043          2245      INX  H
052.140 006 010     2246      MVI  B,8
052.142 021 300 055 2247      LXI  D,PIO.DIR+DIR.NAM
052.145 315 170 052 2248      CALL CDAS      COPY IT
052.150 066 056     2249      MVI  M,'.'
052.152 043          2250      INX  H
052.153 006 003     2251      MVI  B,3
000.000          2252      ERRNZ DIR,EXT-DIR,NAM-8
052.155 315 170 052 2253      CALL CDAS      COPY IT
052.160 066 000     2254      MVI  M,0          FLAG END OF NAME
052.162 171          2255      MOV  A,C          (A) (BIT 7) = 1 IF WILDCARDS
052.163 007          2256      RLC
052.164 057          2257      CMA

```

052.165 346 001 2258 ANI 1 =0 IF WILDCARD  
 052.167 311 2259 RET

2261 \*\* CDAS - CONVERT DIRECTORY FIELD TO ASCII.  
 2262 \*  
 2263 \* ZEROS ARE IGNORED; 2000 WILDCARDS ARE MAPPED TO '??'  
 2264 \*  
 2265 \* ENTRY (DE) = FROM  
 2266 \* (HL) = TO  
 2267 \* (B) = COUNT  
 2268 \* (C) = ORA ACCUMULATOR  
 2269 \* EXIT (DE) ADVANCED  
 2270 \* (HL) = (HL)+(B)  
 2271 \* (C) = (C).OR. (FROM CHARACTERS PROCESSED)  
 2272 \* USES ALL  
 2273 \*  
 2274 \*

052.170 032 2275 CDA5 LDAX D (A) = CHARACTER  
 052.171 261 2276 ORA C  
 052.172 117 2277 MOV C,A  
 052.173 032 2278 LDAX D  
 052.174 023 2279 INX D  
 052.175 247 2280 ANA A  
 052.176 312 210 052 2281 JZ CDA7 IS 00  
 052.201 362 206 052 2282 JP CDA6 NOT 2000  
 052.204 076 077 2283 MVI A,'?'  
 052.206 167 2284 CDA6 MOV M,A  
 052.207 043 2285 INX H INCREMENT TO  
 052.210 005 2286 CDA7 DCR B  
 052.211 302 170 052 2287 JNZ CDA5 IF MORE TO GO  
 052.214 311 2288 RET

2290 \*\* EWS - EXPAND WILDCARD SPECIFICATION.  
 2291 \*  
 2292 \* EWS ENTERS THE FILE NAME IN PIO.XXX INTO THE MANAGED TABLE  
 2293 \* NAMTAB. IF THE FILE NAME CONTAINS WILDCARDS, THE DIRECTORY  
 2294 \* IS READ FOR ELIGIBLE FILES.  
 2295 \*  
 2296 \* ENTRY PIO.XXX = FILE NAME  
 2297 \* EXIT 'C' CLEAR IF OK  
 2298 \* 'C' SET IF ERROR  
 2299 \* USES ALL  
 2300 \*  
 2301 \*

052.215 315 140 050 2302 EWS CALL AEN TRY TO ENTER IT  
 052.220 320 2303 RNC NO WILDCARDS, AM DONE  
 2304 \*  
 2305 \* IS WILDCARD, LOOK UP DEVICE TYPE  
 2306 \*  
 052.221 052 146 055 2307 LHLI NAMTLEN  
 052.224 021 306 055 2308 LXI D,NAMTAB-FB.NAML  
 052.227 031 2309 DAD D (HL) = ADDRESS OF LAST ENTRY

```

052.230 315 107 051 2310 CALL CAD CONVERT ASCII NAME TO DIRECTORY FORMAT
052.233 052 146 055 2311 LMLD NANTLEN
052.234 021 357 377 2312 LXI D,-FB.NAML
052.241 031 2313 DAD D
052.242 042 146 055 2314 SHLD NANTLEN REMOVE WILDCARD FROM TABLE
052.245 315 076 054 2315 CALL $MOVEL
052.250 003 000 275 2316 DW 3,PID.DEV,DIRNAM SET DIRECTORY NAME IN XXX:DIRECT.SYS
052.256 315 076 054 2317 CALL $MOVEL
052.261 013 000 300 2318 DW B+3,PID.DIR+DIR.NAM,EWSC SAVE WILDCARD PATTERN
052.267 001 056 053 2319 LXI B,EWSB
052.272 041 062 055 2320 LXI H,DIRNAM
052.275 377 053 2321 DB SYSCALL,.DECODE GET INFORMATION ABOUT DEVICE
052.277 330 2322 RC ERROR
052.300 072 056 053 2323 LDA EWSB SEE IF A DIRECTORY DEVICE
052.303 346 001 2324 ANI DT,DB
052.305 076 005 2325 MVI A,EC.DNS ASSUME DEVICE NOT SUITABLE
052.307 067 2326 STC
052.310 310 2327 RZ ERROR
2328
2329 * IS DIRECTORY DEVICE, OPEN DIRECTORY
2330
052.311 041 062 055 2331 LXI H,DIRNAM
052.314 076 002 2332 MVI A,CN.DIR
052.316 377 042 2333 DB SYSCALL,.OPENR
052.320 076 200 2334 MVI A,PEC.DF
052.322 330 2335 RC DEVICE FORMAT FAILURE
2336
2337 * READ DIRECTORY ENTRYS FOR MATCH
2338
052.323 052 120 041 2339 EWS1 LMLD DIRWRKP /79.12.GC/
052.326 353 2340 XCHG DE = POINTER TO THE SCRATCH /79.12.GC/
052.327 001 000 002 2341 LXI R,512
052.332 076 002 2342 MVI A,CN.DIR
052.334 325 2343 PUSH B SAVE ADDRESS
052.335 377 004 2344 DB SYSCALL,.READ READ BLOCK
052.337 341 2345 POP H (HL) = DIRECTORY ADDRESS
052.340 332 043 053 2346 JC EWS7 ALL DONE
2347
2348 * LOOK AT DIRECTORY BLOCK FOR MATCHES
2349
052.343 345 2350 PUSH H
052.344 052 120 041 2351 LMLD DIRWRKP /79.12.GC/
052.347 021 373 001 2352 LXI D,DIS.ENL /79.12.GC/
052.352 031 2353 DAD D /79.12.GC/
052.353 116 2354 MOV C,H C = LENGTH /79.12.GC/
052.354 341 2355 POP H /79.12.GC/
2356
2357 * CHECK NEXT ENTRY
2358
052.355 176 2359 EWS3 MOV A,H (A) = 1ST CHAR THIS ENTRY
052.356 247 2360 ANA A
052.357 312 323 052 2361 JZ EWS1 END OF BLOCK
000.000 2362 ERRNZ DF.EMP-3770
052.362 074 2363 INR A
052.363 312 035 053 2364 JZ EWS6 ENTRY EMPTY
000.000 2365 ERRNZ DF.CLR-3760

```

```

052.366 074      2366      INR      A
052.367 312 043 053 2367      JZ      EWS7      END OF LIST
052.372 345      2368      PUSH     M
052.373 021 114 053 2369      LXI     D,EWSC
052.376 006 013      2370      MVI     B,B+3
053.000 315 022 051 2371      CALL    CWM      CHECK WILDCARD MATCH
053.003 302 034 053 2372      JNZ     EWS4      NO MATCH
                2373
                2374 *      HAVE MATCH. ADD TO LIST
                2375
053.006 321      2376      POP     D      (DE) = FROM
053.007 325      2377      PUSH     D
053.010 315 311 050 2378      CALL    CSF      CHECK FOR SPECIAL FILE
053.013 312 034 053 2379      JZ      EWS4      IS SPECIAL FILE, DONT ENTER
053.016 305      2380      PUSH     B      SAVE (C)
053.017 001 013 000 2381      LXI     B,B+3
053.022 041 300 055 2382      LXI     H,PIO.DIR+DIR.NAM
053.025 315 252 030 2383      CALL    $MOVE
053.030 315 140 050 2384      CALL    AEN      ADD TO TABLE
053.033 301      2385      POP     B      RESTORE (C)
                2386
                2387 *      LOOKUP NEXT ENTRY
                2388
053.034 341      2389      EWS4    POP     H
053.035 006 000      2390      EWS6    MVI     R,0
053.037 011      2391      DAD     B      POINT TO NEXT
053.040 303 355 052 2392      JMP     EWS3
                2393
                2394 *      ALL DONE. CLOSE DIRECTORY FILE
                2395
053.043 076 002      2396      EWS7    MVI     A,CN.DIR
053.045 377 046      2397      DB     SYSCALL,CLOSE
053.047 311      2398      RET
                2399
053.050 123 131 060 2400      EWSA    DB     'SYO',2000,2000,2000
                2401
053.056      2402      EWSB    DS     30
                2403
053.114      2404      EWSC    DS     B+3      WILDCARD PATTERN FOR DIRECTORY SEARCH
                2405
                2406 **      INA - INCREASE NAMTAB ALLOCATION.
                2407 *
                2408 *      INA IS CALLED TO INCREASE THE NAMTAB ALLOCATION. THE
                2409 *      BUFFER AREA IS MOVED UP TO MAKE ROOM.
                2410 *
                2411 *      ENTRY    NONE
                2412 *      EXIT     NONE
                2413 *      USES    A,F,H,L
                2414
053.127 041 151 055 2415      INA    LXI     H,NAMTMAX+1
053.132 064      2416      INR     M      INCREMENT LENGTH
053.133 041 110 055 2417      LXI     H,BUFFPTR+1
053.136 064      2418      INR     M      MOVE BUFFER
    
```



```

053.137 052 111 055 2419 LHL D BUFSIZ
053.142 174 2420 MOV A,H
053.143 265 2421 ORA L
053.144 076 021 2422 MVI A,EC,NEM FLAG OUT OF MEMORY IF BUFFER NOT EMPTY
053.146 302 007 050 2423 JNZ ERROR
053.151 305 2424 PUSH B
053.152 325 2425 PUSH D
053.153 315 302 053 2426 CALL SKE NOTIFY SYSTEM
053.156 321 2427 POP D
053.157 301 2428 POP B
053.160 311 2429 RET
  
```

```

2431 ** LSN - LOCATE SOURCE NAME
2432 *
2433 * LSN SCANS THE COMMAND LINE FOR THE FIRST SOURCE FILE NAME.
2434 *
2435 * ENTRY NONE
2436 * EXIT (HL) = 1ST FILE NAME FWA
2437 * USES A,F,H,L
2438
053.161 041 346 042 2439 LSN LXI H,LINE
053.164 176 2440 LSN1 MOV A,H
053.165 043 2441 INX H
053.166 376 075 2442 CPI '='
053.170 310 2443 RE GOT IT
053.171 247 2444 ANA A
053.172 302 164 053 2445 JNZ LSN1 MORE LINE
053.175 041 346 042 2446 LXI H,LINE IS NO =
053.200 311 2447 RET
  
```

```

2449 ** MWN - MERGE WILDCARD NAMES.
2450 *
2451 * MWN MERGES A COMPLETELY SPECIFIED FILENAME WITH A WILDCARDED COMPLETELY
2452 * SPECIFIED FILE NAME.
2453 *
2454 * BOTH FILE NAMES SHOULD HAVE THE SAME DEVICE SPECIFICATION.
2455 *
2456 * FILE NAME FORMAT:
2457 *
2458 * DEV:NAMEXXXX.EXT 00
2459 *
2460 * ENTRY (BC) = ADDRESS OF WILDCARDED ASCII NAME
2461 * (DE) = ADDRESS OF NON-WC ASCII NAME
2462 * (HL) = ADDRESS FOR RESULTANT ASCII NAME
2463 * EXIT NONE
2464 * USES ALL
2465
053.201 345 2467 MWN PUSH H SAVE TARGET ADDRESS
053.202 305 2468 MWN PUSH B SAVE WC PATTERN
  
```

MWN

```

053.203 353      2469      XCHG      (HL) = MASTER NAME
053.204 315 107 051 2470      CALL      CAD      CONVERT TO DIRECTORY FORMAT
053.207 315 076 054 2471      CALL      $MOVE1
053.212 013 000 300 2472      DW      B+3,PIO.DIR,MWNA      (MWNA) = DECODED MASTER
053.220 341      2473      POP      H      (HL) = WC PATTERN
053.221 315 107 051 2474      CALL      CAD      (PIO.DIR) = WC PATTERN
053.224 021 254 055 2475      LXI      D,MWNA      (DE) = MASTER PATTERN
053.227 041 300 055 2476      LXI      H,PIO.DIR      (DE) = WC PATTERN ADDRESS
053.232 016 013      2477      MVI      C,B+3      MERGE NAME AND EXTENSION
                                2478
                                2479 *      MERGE NAMES
                                2480
053.234 176      2481 MWN1      MOV      A,H      (A) = WC PATTERN
053.235 247      2482      ANA      A
053.236 362 242 053 2483      JP      MWN2
053.241 032      2484      LDAX   B      IS MATCH CHARACTER, USE MASTER INSTEAD
053.242 167      2485 MWN2      MOV      H,A      STORE CHARACTER
053.243 023      2486      INX   D
053.244 043      2487      INX   H
053.245 015      2488      DCR   C
053.246 302 234 053 2489      JNZ   MWN1      MERGE TILL DONE
053.251 341      2490      POP   H      (HL) = TARGET ADDRESS
053.252 303 124 052 2491      JMP   CDA      CONVERT DIRECTORY FORMAT TO ASCII

                                2493 **      REN - REMOVE ENTRY FROM *NAMTAB*
                                2494 *
                                2495 *      REN REMOVES THE FIRST 'FB.NAML' BYTES FROM NAMTAB.
                                2496 *
                                2497 *      THE AMOUNT (FB.NAML) IS REMOVED FROM THE SIZE OF THE TABLE, THE
                                2498 *      TABLE IS NOT CHECKED FOR UNDERFLOW, THE CALLER MUST GUARANTEE THE
                                2499 *      PRESENCE OF AT LEAST FB.NAML BYTES IN NAMTAB.
                                2500 *
                                2501 *      ENTRY  NONE
                                2502 *      EXIT   NONE
                                2503 *      USES   ALL
                                2504
                                2505
053.255 052 146 055 2506 REN      LHL   NAMTLEN
053.260 021 357 377 2507      LXI   D,-FB.NAML
053.263 031      2508      DAD   D      REMOVE COUNT FROM LEN
053.264 042 146 055 2509      SHLD  NAMTLEN
053.267 104      2510      MOV   B,H
053.270 115      2511      MOV   C,L      (BC) = REMAINING LENGTH
053.271 021 350 055 2512      LXI   D,NAMTAB+FB.NAML      (DE) = START OF 2ND ENTRY
053.274 041 327 055 2513      LXI   H,NAMTAB
053.277 303 252 030 2514      JMP   $MOVE      MOVE DOWN AND RETURN

```

```

2516 **      SBE - SET BUFFER EMPTY.
2517 *
2518 *      THE SYSTEM IS NOTIFIED.
2519 *
2520 *      ENTRY  NONE
2521 *      EXIT   NONE
2522 *      USES   ALL
2523 *
2524 *
053.302 041 000 000 2525 SBE LXI    H,0
053.305 042 111 055 2526 SHLD  BUFSIZ
053.310 052 107 055 2527 LHLD  BUFPTR      (HL) = BUFFER FWA (AND LWA!)
053.313 043          2528 INX   H
053.314 043          2529 INX   H
053.315 377 052     2530 DB    SYSCALL,SETTF
053.317 320         2531 RNC
053.320 303 007 050 2532 JMP   ERROR      OK
                                NOT ENOUGH ROOM
  
```

```

2534 **      SFS - SKIP FILE SEPERATOR.
2535 *
2536 *      SFS IS CALLED TO SKIP OVER THE CHARACTERS SEPERATING ONE
2537 *      FILE NAME FROM ANOTHER ON THE LINE. THE FILES MAY BE SEPERATED
2538 *      BY BLANKS OR A COMMA ALONE, OR BY BLANKS WITH A COMMA. THE
2539 *      SYNTAX IS
2540 *
2541 *      <BLANKS> <,> <BLANKS>
2542 *
2543 *      ONE, TWO OR ALL THREE FIELDS MAY BE PRESENT.
2544 *
2545 *      ENTRY  (HL) = POINT TO START OF SEP FIELD
2546 *      EXIT   (HL) ADVANCED PAST SEPERATOR FIELD
2547 *      USES   A,F,H,L
2548 *
2549 *
053.323 318 370 053 2550 SFS  CALL  $SOB      SKIP BLANKS
053.326 176         2551 MOV   A,M
053.327 376 054     2552 CPI   ','
053.331 302 335 053 2553 JNE  SFS1      NOT ,
053.334 043         2554 INX   H        SKIP ,
053.335 303 370 053 2555 SFS1 JMP   $SOB      GET ANY MORE BLANKS AND EXIT
  
```

```

2557 **      SND - SET NEW DEFAULTS.
2558 *
2559 *      SND IS CALLED TO SET A NEW DEFAULT DEVICE AND EXTENSION
2560 *      IN THE 'DEFAULT' AREA.
2561 *
2562 *      ENTRY  PID.DEV = DEVICE CODE
2563 *            PID.UNI = UNIT #
2564 *            PID.DIR+DIR.EXT = EXTENSION
2565 *      EXIT  NONE
  
```

```
2566 *      USES  NONE
2567
2568
053.340 315 054 031 2569 SND  CALL  $SAVALL      SAVE REGS
000.000                               2570  ERRNZ  PIO.UNI-PIO.DEV-2
053.343 315 076 054 2571  CALL  $MOVEL
053.346 003 000      2572  IW     3
053.350 275 055      2573  IW     PIO.DEV
053.352 101 055      2574  IW     DEFAULT
053.354 315 076 054 2575  CALL  $MOVEL
053.357 003 000      2576  IW     3
053.361 310 055      2577  IW     PIO.DIR+DIR.EXT
053.363 104 055      2578  IW     DEFAULT+3
053.365 303 047 031 2579  JMP    $RSTALL      RETURN
```

053.370

2582

XTEXT COMP

2584X \*\* \*COMP - COMPARE TWO CHARACTER STRINGS.  
2585X \*  
2586X \* \*COMP COMPARES TWO BYTE STRINGS.  
2587X \*  
2588X \* ENTRY (C) = COMPARE COUNT  
2589X \* (DE) = FWA OF STRING #1  
2590X \* (HL) = FWA OF STRING #2  
2591X \* EXIT 'Z' CLEAR, IS MIS-MATCH  
2592X \* (C) = LENGTH REMAINING  
2593X \* (DE) = ADDRESS OF MISMATCH IN STRING#1  
2594X \* (HL) = ADDRESS OF MISMATCH IN STRING #2  
2595X \* 'C' SET, HAVE MATCH  
2596X \* (C) = 0  
2597X \* (DE) = (DE) + (0C)  
2598X \* (HL) = (HL) + (0C)  
2599X \* USES A,F,C,D,E,H,L

2600X

2601X

030.060

053.370

2602X \*COMP

EQU 30060A

IN H17 ROM

2603

XTEXT WER

2605X \*\* \*WER - WRITE ENABLE RAM.  
2606X \*  
2607X \* \*WER IS CALLED TO ENABLE WRITING TO THE H17 CONTROLLER'S  
2608X \* RAM AREA.  
2609X \*  
2610X \* ENTRY NONE  
2611X \* EXIT NONE  
2612X \* USES NONE  
2613X

2614X

031.241

2615X \*WER

EQU 31241A

IN H17 ROM

2617X \*\* \*WDR - WRITE DISABLE RAM.  
2618X \*  
2619X \* \*WDR IS CALLED TO DISABLE WRITING TO THE H17 CONTROLLER'S  
2620X \* RAM AREA.  
2621X \*  
2622X \* ENTRY NONE  
2623X \* EXIT NONE  
2624X \* USES NONE  
2625X

2626X

031.222

053.370

2627X \*WDR

EQU 31222A

IN H17 ROM

2628

XTEXT ZERO

```

2630X ** $ZERO - ZERO MEMORY
2631X *
2632X * $ZERO ZEROS A BLOCK OF MEMORY.
2633X *
2634X * ENTRY (HL) = ADDRESS
2635X * (B) = COUNT
2636X * EXIT (A) = 0
2637X * USES A,B,F,H,L
2638X
2639X
031.212 2640X $ZERO EQU 31212A IN H17 ROM
053.370 2641 XTEXT MOVE
    
```

```

2643X ** $MOVE - MOVE DATA
2644X *
2645X * $MOVE MOVES A BLOCK OF BYTES TO A NEW MEMORY ADDRESS.
2646X * IF THE MOVE IS TO A LOWER ADDRESS, THE BYTES ARE MOVED FROM
2647X * FIRST TO LAST.
2648X *
2649X * IF THE MOVE IS TO A HIGHER ADDRESS, THE BYTES ARE MOVED FROM
2650X * LAST TO FIRST.
2651X *
2652X * THIS IS DONE SO THAT AN OVERLAPED MOVE WILL NOT 'RIPPLE'.
2653X *
2654X * ENTRY (BC) = COUNT
2655X * (DE) = FROM
2656X * (HL) = TO
2657X * EXIT MOVED
2658X * (DE) = ADDRESS OF NEXT FROM BYTE
2659X * (HL) = ADDRESS OF NEXT *TO* BYTE
2660X * 'C' CLEAR
2661X * USES ALL
2662X
2663X
030.252 2664X $MOVE EQU 30252A IN H17 ROM
053.370 2665 XTEXT CHL
    
```

```

2667X ** $CHL - COMPLEMENT (HL).
2668X *
2669X * (HL) = -(HL) TWO'S COMPLEMENT
2670X *
2671X * ENTRY NONE
2672X * EXIT NONE
2673X * USES A,F,H,L
2674X
2675X
030.224 2676X $CHL EQU 30224A IN H17 ROM
053.370 2677 XTEXT SOB
    
```

\*SOB

```

2679X **      *SOB - SKIP OVER BLANKS.
2680X *
2681X *      *SOB IS CALLED TO SKIP AN ARBITRARILY LONG STRING OF BLANKS AND TABS.
2682X *
2683X *      ENTRY (HL) = FWA OF (POSSIBLE) BLANK STRING
2684X *      EXIT (HL) = LWA+1 OF BLANK STRING (UNCHANGED IF NO BLANKS)
2685X *      (A) = FIRST NON-BLANK, NON-TAB CHARACTER EEN
2686X *      USES A,F,H,L
2687X
2688X
053.370 053 2689X *SOB DCX H PRE-DECREMENT
053.371 043 2690X *SOB1 INX H
053.372 176 2691X MOV A,M
053.373 376 040 2692X CPI /
053.375 312 371 053 2693X JE $SOB1 GOT BLANK
054.000 376 011 2694X CPI TAB
054.002 312 371 053 2695X JE $SOB1 GOT TAB
054.005 311 2696X RET
054.006 2697 XTEXT DADA
    
```

```

2699X **      $DADA - PERFORM (H,L) = (H,L) + (0,A)
2700X *
2701X *      ENTRY (H,L) = BEFORE VALUE
2702X *      (A) = BEFORE VALUE
2703X *      EXIT (H,L) = (H,L) + (0,A)
2704X *      'C' SET IF OVERFLOW
2705X *      USES F,H,L
2706X
2707X
030.072 2708X $DADA EQU 30072A IN H17 ROM
054.006 2709 XTEXT TJMP
    
```

```

2711X **      $TJMP - TABLE JUMP.
2712X *
2713X *      USAGE
2714X *
2715X *      CALL $TJMP (A) = INDEX
2716X *      DW ADDR1
2717X *      .
2718X *      .
2719X *      .
2720X *      DW ADDRn
2721X *
2722X *      ENTRY (A) = INDEX
2723X *      EXIT TO PROCESSOR
2724X *      (A) = INDEX*2
2725X *      USES NONE.
2726X
2727X
031.061 2728X $TJMP EQU 31061A IN H17 ROM, (A) = INDEX*2
    
```

```

2729X
031.062 2730X $TJMP. EQU 31062A IN H17 ROM
054.006 2731 XTEXT CRLF

2733X ** *CRLF - TYPE CARRIAGE RETURN/ LINE FEED
2734X *
2735X * *CRLF IS USED TO GENERATE PADDED CRLF'S.
2736X *
2737X * ENTRY NONE
2738X * EXIT (A) = 0
2739X * USES A,F.
2740X
2741X
054.006 076 012 2742X $CRLF MVI A,NL
054.010 377 002 2743X DB SYSCALL,,SCOUT
054.012 257 2744X XRA A
054.013 311 2745X RET
054.014 2746 XTEXT TYPCH

2748X ** *TYPCH - TYPE SINGLE CHARACTER.
2749X *
2750X * ENTRY (RET) = CHARACTER
2751X * EXIT TO (RET)+1
2752X * (A) = CHARACTER TYPED
2753X
2754X
054.014 343 2755X $TYPCH XTHL (HL) = RETURN ADDRESS
054.015 176 2756X MOV A,M (A) = CHARACTER
054.016 043 2757X INX H
054.017 343 2758X XTHL RESTORE ADVANCED EXIT ADDRESS
2759X

2760X ** $TYPC - TYPE SINGLE CHARACTER.
2761X *
2762X * ENTRY (A) = CHARACTER
2763X * EXIT TO (RET)
2764X
054.020 377 002 2765X $TYPC. DB SYSCALL,,SCOUT
054.022 311 2766X RET
054.023 2767 XTEXT TYPT2

2769X ** $TYPTX - TYPE TEXT.
2770X *
2771X * $TYPTX IS CALLED TO TYPE A BLOCK OF TEXT ON THE SYSTEM CONSOLE.
2772X *
2773X * IMBEDDED ZERO BYTES INDICATE A CARRIAGE RETURN LINE FEED.
2774X * A BYTE WITH THE 2000 BIT SET IS THE LAST BYTE IN THE MESSAGE.
2775X *

```



```

2776X *      ENTRY  (RET) = TEXT
2777X *      EXIT   TO (RET+LENGTH)
2778X *      USES   A,F
2779X
2780X
031.136     2781X *TYPTX EQU   31136A      IN H17 ROM
2782X
031.144     2783X *TYPTX EQU   31144A      IN H17 ROM
054.023     2784      XTEXT  SAVALL

```

```

2786X **      *RSTALL - RESTORE ALL REGISTERS.
2787X *
2788X *      *RSTALL RESTORES ALL THE REGISTERS OFF THE STACK, AND
2789X *      RETURNS TO THE PREVIOUS CALLER.
2790X *
2791X *      ENTRY  (SP) = PSW
2792X *            (SP+2) = BC
2793X *            (SP+4) = DE
2794X *            (SP+6) = HL
2795X *            (SP+8) = RET
2796X *      EXIT   TO *RET*, REGISTERS RESTORED
2797X *      USES   ALL
2798X
2799X
031.047     2800X *RSTALL EQU   31047A      IN H17 ROM

```

```

2802X **      *SAVALL - SAVE ALL REGISTERS ON STACK.
2803X *
2804X *      *SAVALL SAVES ALL THE REGISTERS ON THE STACK.
2805X *
2806X *      ENTRY  NONE
2807X *      EXIT   (SP) = PSW
2808X *            (SP+2) = BC
2809X *            (SP+4) = DE
2810X *            (SP+6) = HL
2811X *      USES   H,L
2812X
2813X
031.054     2814X *SAVALL EQU   31054A      IN H17 ROM
054.023     2815      XTEXT  CDEHL

```

```

2817X **      *CDEHL - COMPARE (DE) TO (HL)
2818X *
2819X *      *CDEHL COMPARES (DE) TO (HL) FOR EQUALITY.
2820X *
2821X *      ENTRY  NONE
2822X *      EXIT   'Z' SET IF (DE) = (HL)
2823X *      USES   A,F
2824X

```

030.216 2825X  
054.023 2826X \$CDEHL EQU 30216A IN H17 ROM  
2827 XTEXT UDD

2829X \*\* \$UDD - UNPACK DECIMAL DIGITS.  
2830X \*  
2831X \* UDD CONVERTS A 16 BIT VALUE INTO A SPECIFIED NUMBER OF  
2832X \* DECIMAL DIGITS. THE RESULT IS ZERO FILLED.  
2833X \*  
2834X \* ENTRY (B;C) = ADDRESS VALUE  
2835X \* (A) = DIGIT COUNT  
2836X \* (H;L) = MEMORY ADDRESS  
2837X \* EXIT (HL) = (HL) + (A)  
2838X \* USES ALL  
2839X  
2840X

031.157 2841X \$UDD EQU 31157A IN H17 ROM  
054.023 2842 XTEXT DU66

2844X \*\* \$DU66 - UNSIGNED 16 / 16 DIVIDE.  
2845X \*  
2846X \* (HL) = (BC)/(DE)  
2847X \*  
2848X \* ENTRY (BC), (DE) PRESET  
2849X \* EXIT (HL) = RESULT  
2850X \* (DE) = REMAINDER  
2851X \* USES ALL  
2852X  
2853X

030.106 2854X \$DU66 EQU 30106A IN H17 ROM  
054.023 2855 XTEXT DADA2

2857X \*\* \$DADA. - ADD (D;A) TO (H;L)  
2858X \*  
2859X \* ENTRY NONE  
2860X \* EXIT (HL) = (HL) + (DA)  
2861X \* USES A,F,H;L  
2862X  
2863X

030.101 2864X \$DADA. EQU 30101A IN H17 ROM  
054.023 2865 XTEXT HLIHL

```

2867X **      $HLIHL - LOAD HL INDIRECT THROUGH HL.
2868X *
2869X *      (HL) = ((HL))
2870X *
2871X *      ENTRY  NONE
2872X *      EXIT   NONE
2873X *      USES   A,H,L
2874X *
030.211      2875X $HLIHL EQU   30211A      IN H17 ROM
054.023      2876      XTEXT  INDL

```

```

2878X **      $INDL - INDEXED LOAD.
2879X *
2880X *      $INDL LOADS DE WITH THE TWO BYTES AT (HL)+DISPLACMENT
2881X *
2882X *      THIS ACTS AS AN INDEXED FULL WORD LOAD.
2883X *
2884X *      (DE) = ( (HL) + DSPLACEMENT )
2885X *
2886X *      ENTRY  ((RET)) = DISPLACMENT (FULL WORD)
2887X *      (HL) = TABLE ADDRESS
2888X *      EXIT   TO (RET+2)
2889X *      USES   A,F,D,E
2890X *
030.234      2891X
054.023      2892X $INDL EQU   30234A      IN H17 ROM
2893      XTEXT  UPDN

```

```

2895X **      $UDDN - UNPACK DECIMAL DIGITS.
2896X *
2897X *      UDDN CONVERTS A 16 BIT VALUE INTO A SPECIFIED NUMBER OF
2898X *      DECIMAL DIGITS. THE RESULT IS NULL FILLED TO THE LEFT.
2899X *
2900X *      ENTRY  (B+C) = ADDRESS VALUE
2901X *      (A) = DIGIT COUNT
2902X *      (H:L) = MEMORY ADDRESS
2903X *      EXIT   (HL) = (HL) + (A)
2904X *      USES   ALL
2905X *
054.023      2906X
054.023      2907X $UDDN EQU   *
054.023 315 072 030 2908X CALL  $IADA
054.026 345      2909X PUSH  H          SAVE FINAL (H:L) VALUE
2910X
054.027 365      2911X UDDN1 PUSH  FSW
054.030 345      2912X PUSH  H
054.031 021 012 000 2913X LXI   D,10
054.034 315 106 030 2914X CALL  $DU66      (H:L) = VALUE/10
054.037 104      2915X MOV   B,H
054.040 115      2916X MOV   C,L      (BC) = QUOTIENT

```

\*UDDN

```

054.041 341      2917X      POP      H
054.042 076 060  2918X      MVI      A,'0'
054.044 203      2919X      ADD      E      ADD REMAINDER
054.045 053      2920X      DCX      H
054.046 167      2921X      MOV      M,A      STORE DIGIT
054.047 170      2922X      MOV      A,B
054.050 261      2923X      DRA      C
054.051 312 063 054 2924X      JZ       UDDN2      ALL ZEROS
054.054 361      2925X      POP      PSW
054.055 075      2926X      DCR      A
054.056 302 027 054 2927X      JNZ      UDDN1      IF MORE TO GO
2928X
2929X *      ALL DONE, EXIT
2930X
054.061 341      2931X UDDN1.5 POP      H      RESTORE H
054.062 311      2932X      RET      RETURN
2933X
2934X *      DIGITS LEADING THIS ONE ARE ZERO, STORE NULLS INSTEAD.
2935X
054.063 361      2936X UDDN2      POP      PSW
054.064 075      2937X UDDN3      DCR      A
054.065 312 061 054 2938X      JE       UDDN1.5      ALL DONE
054.070 053      2939X      DCX      H
054.071 066 000      2940X      MVI      M,0
054.073 303 064 054 2941X      JMP      UDDN3
054.076      2942      XTEXT  MOVEL

```

```

2944X **      $MOVEL - MOVE DATA
2945X *
2946X *      $MOVEL MOVES A BLOCK OF BYTES TO A NEW MEMORY ADDRESS.
2947X *      IF THE MOVE IS TO A LOWER ADDRESS, THE BYTES ARE MOVED FROM
2948X *      FIRST TO LAST.
2949X *
2950X *      IF THE MOVE IS TO A HIGHER ADDRESS, THE BYTES ARE MOVED FROM
2951X *      LAST TO FIRST.
2952X *
2953X *      THIS IS DONE SO THAT AN OVERLAPED MOVE WILL NOT 'RIPPLE'.
2954X *
2955X *      CALL  $MOVEL
2956X *      DW   COUNT
2957X *      DW   FROM
2958X *      DW   TO
2959X *
2960X *      ENTRY ((SP)) = RET
2961X *      (RET+0) = COUNT (WORD VALUE)
2962X *      (RET+2) = FROM
2963X *      (RET+4) = TO
2964X *      EXIT  TO (RET+6)
2965X *      (DE) = ADDRESS OF NEXT FROM BYTE
2966X *      (HL) = ADDRESS OF NEXT *TO* BYTE
2967X *      'C' CLEAR
2968X *      USES  ALL
2969X

```

```

2970X
054.076 341 2971X $MOVEL POP H (HL) = RET
054.077 116 2972X MOV C,M
054.100 043 2973X INX H
054.101 106 2974X MOV B,M (BC) = COUNT
054.102 043 2975X INX H
054.103 136 2976X MOV E,M
054.104 043 2977X INX H
054.105 126 2978X MOV D,M (DE) = FROM
054.106 043 2979X INX H
054.107 325 2980X PUSH D ((SP)) = FROM
054.110 136 2981X MOV E,M
054.111 043 2982X INX H
054.112 126 2983X MOV D,M (DE) = TO
054.113 043 2984X INX H
054.114 343 2985X XTHL ((SP)) = RET, (HL) = FROM
054.115 353 2986X XCHG (DE) = FROM, (HL) = TO
054.116 303 252 030 2987X JMP $MOVE MOVE IT
054.121 2988 XTEXT RCHAR

```

```

2990X ** $RCHAR - READ SINGLE CHARACTER FROM CONSOLE.
2991X *
2992X * ENTRY NONE
2993X * EXIT (A) = CHARACTER
2994X * USES A,F
2995X
2996X
054.121 377 001 2997X $RCHAR DB SYSCALL, SCIN
054.123 332 121 054 2998X JC $RCHAR NOT READY
054.126 311 2999X RET
3000X
054.127 377 002 3001X $WCHAR DB SYSCALL, SCOUT
054.131 311 3002X RET
054.132 3003 XTEXT TYPCC

```

```

3005X ** $TYPCC - TYPE A CHARACTER STRING BY COUNT.
3006X *
3007X * $TYPCC TYPES A STRING OF CHARACTERS, THE CALLER SUPPLIES
3008X * THE CHARACTER ADDRESS AND COUNT.
3009X *
3010X * ENTRY (HL) = ADDRESS
3011X * (A) = COUNT
3012X * EXIT (HL) = LAST CHARACTER ADDRESS+1
3013X * USES A,F,H,L
3014X
3015X
054.132 3016X $TYPCC EQU *
054.132 247 3017X ANA A
054.133 310 3018X RZ NOTHING TO TYPE
054.134 365 3019X PUSH PSW SAVE COUNT

```

COMMON DECKS

\$TYPCC

16:09:31 16-MAY-80

054.135	176	3020X	MOV	A,M	(A) = CHARACTER
054.136	043	3021X	INX	H	
054.137	377 002	3022X	DB	SYSCALL, SCOUT	
054.141	361	3023X	POP	PSW	
054.142	075	3024X	DCR	A	
054.143	303 132 054	3025X	JMP	\$TYPCC	

```

3028
3029
3030 **      FDN - FILE DESCRIPTOR NODES.
3031 *
3032 *      THESE NODES ARE USED TO KEEP TRACK OF FILES WHICH ARE BEING
3033 *      HELD IN MEMORY WHILE TRANSFERING.
3034
054.146 3035 FDN      DS      0      START OF TYPICAL NODE
000.000 3036 FDN.LNK EQU    *-FDN    LINK TO NEXT NODE IN CHAIN
054.146 3037      DS      1      ALL IN SAME PAGE, JUST KEEP PAGE INDEX
000.001 3038 FDN.STA EQU    *-FDN    STATUS BYTE
000.020 3039 ST.CNT EQU    DIF.CNT    IS CONTIGUOUS
000.002 3040 ST.OPR EQU    00000010B  IS BEING READ
000.001 3041 ST.OPW EQU    00000001B  OPEN FOR WRITE
054.147 3042      DS      1      STATUS BYTE
000.002 3043 FDN.FLG EQU    *-FDN    FLAG BITS SET ON SOURCE FILE
054.150 3044      DS      1
000.003 3045 FDN.SIZ EQU    *-FDN    TOTAL SIZE OF FILE (IF ST.CNT SET)
054.151 3046      DS      1      SIZE IN GROUPS
000.004 3047 FDN.AMR EQU    *-FDN    AMOUNT ALREADY READ
054.152 3048      DS      2      IN SECTORS
000.006 3049 FDN.AMW EQU    *-FDN    AMOUNT ALREADY WRITTEN
054.154 3050      DS      2      IN SECTORS
000.010 3051 FDN.ADR EQU    *-FDN    ADDRESS IN BUFFER
054.156 3052      DS      1      ADDRESS/256 (MUST BE EVEN PAGE)
000.011 3053 FDN.AIM EQU    *-FDN    AMOUNT IN MEMORY
054.157 3054      DS      1      IN SECTORS
000.012 3055 FDN.LEN EQU    *-FDN    ENTRY LENGTH
054.146 3056      ORG     FDN      ORG BACK OVER DEFINITION AREA
3057
3058 *
3059
3060
3061 **      TABLE. A LINK OF 0 IS A NULL LINK.
3062 *
3063 *      THE ENTIRE GROUP OF NODES MUST RESIDE
3064 *      IN THE SAME PAGE
3065
054.146 3066 FDN.FWA EQU    *      START OF NODES
3067
054.146 150 3068 FDN.FRE DB     #FDN.1    START OF FREE CHAIN
054.147 000 3069 FDN.HEAD DB     0      ACTIVE LIST NOW EMPTY
3070
054.150 3071 FDN.1  DS     0
054.150 162 3072      DB     #FDN.2    FDN.LNK
054.151 000 3073      DB     0      FDN.STA
054.152 000 3074      DB     0      FDN.FLG
054.153 000 3075      DB     0      FDN.SIZ
054.154 000 000 3076      DW     0      FDN.AMR
054.156 000 000 3077      DW     0      FDN.AMW
054.160 000 3078      DB     0      FDN.ADR
054.161 000 3079      DB     0      FDN.AIM
3080
054.162 3081 FDN.2  DS     0
054.162 174 3082      DB     #FDN.3    FDN.LNK
054.163 000 3083      DB     0      FDN.STA

```

054.164	000		3084	DB	0		FDN.FLG
054.165	000		3085	DB	0		FDN.SIZ
054.166	000	000	3086	DW	0		FDN.AMR
054.170	000	000	3087	DW	0		FDN.AMW
054.172	000		3088	DB	0		FDN.ADR
054.173	000		3089	DB	0		FDN.AIM
			3090				
054.174			3091	FDN.3	DS	0	
054.174	206		3092	DB	#FDN.4		FDN.LNK
054.175	000		3093	DB	0		FDN.STA
054.176	000		3094	DB	0		FDN.FLG
054.177	000		3095	DB	0		FDN.SIZ
054.200	000	000	3096	DW	0		FDN.AMR
054.202	000	000	3097	DW	0		FDN.AMW
054.204	000		3098	DB	0		FDN.ADR
054.205	000		3099	DB	0		FDN.AIM
			3100				
054.206			3101	FDN.4	DS	0	
054.206	220		3102	DB	#FDN.5		FDN.LNK
054.207	000		3103	DB	0		FDN.STA
054.210	000		3104	DB	0		FDN.FLG
054.211	000		3105	DB	0		FDN.SIZ
054.212	000	000	3106	DW	0		FDN.AMR
054.214	000	000	3107	DW	0		FDN.AMW
054.216	000		3108	DB	0		FDN.ADR
054.217	000		3109	DB	0		FDN.AIM
			3110				
054.220			3111	FDN.5	DS	0	
054.220	232		3112	DB	#FDN.6		FDN.LNK
054.221	000		3113	DB	0		FDN.STA
054.222	000		3114	DB	0		FDN.FLG
054.223	000		3115	DB	0		FDN.SIZ
054.224	000	000	3116	DW	0		FDN.AMR
054.226	000	000	3117	DW	0		FDN.AMW
054.230	000		3118	DB	0		FDN.ADR
054.231	000		3119	DB	0		FDN.AIM
			3120				
054.232			3121	FDN.6	DS	0	
054.232	244		3122	DB	#FDN.7		FDN.LNK
054.233	000		3123	DB	0		FDN.STA
054.234	000		3124	DB	0		FDN.FLG
054.235	000		3125	DB	0		FDN.SIZ
054.236	000	000	3126	DW	0		FDN.AMR
054.240	000	000	3127	DW	0		FDN.AMW
054.242	000		3128	DB	0		FDN.ADR
054.243	000		3129	DB	0		FDN.AIM
			3130				
054.244			3131	FDN.7	DS	0	
054.244	256		3132	DB	#FDN.8		FDN.LNK
054.245	000		3133	DB	0		FDN.STA
054.246	000		3134	DB	0		FDN.FLG
054.247	000		3135	DB	0		FDN.SIZ
054.250	000	000	3136	DW	0		FDN.AMR
054.252	000	000	3137	DW	0		FDN.AMW
054.254	000		3138	DB	0		FDN.ADR
054.255	000		3139	DB	0		FDN.AIM



			3140				
			3141	FDN,B	DS	0	
054.256			3142		DE	0	FDN.LNK
054.256	000		3143		DE	0	FDN.STA
054.257	000		3144		DE	0	FDN.FLG
054.260	000		3145		DB	0	FDN.SIZ
054.261	000		3146		IW	0	FDN.AMR
054.262	000 000		3147		DW	0	FDN.AMW
054.264	000 000		3148		DE	0	FDN.ADR
054.266	000		3149		DE	0	FDN.AIM
054.267	000		3150				
000.010			3151	FINDCNT	EQU	*-FDN,1/FDNELEN	NUMBER OF NODES
			3152				
000.054			3153		SET	*/256	
000.000			3154		ERRNZ	FDNFWA/256-	MUST BE ALL IN SAME PAGE
			3155				
054.270	000		3156	VOLFLAG	DB	0	=0 IF READING FROM SOURCE, =3770 IF WRITTING TO DEST
054.271	000		3157	VOLSER	DB	0	SERIAL NUMBER OF CURRENT DISK
			3158				
054.272	000		3159	ORUFLIM	DB	0	BUFFER LIMIT/256
054.273	000		3160	OBUFFPTR	DB	0	NEXT FREE PAGE IN BUFFER/256
			3161				
			3162				
054.274			3163	XTEXT	FERROR		HERE TO LET FDN BE IN ONE PAGE

			3165X	**			\$FERROR - PROCESS FILE ERRORS.
			3166X	*			
			3167X	*			\$FERROR IS CALLED TO COMPLAIN ABOUT AN ERROR ENCOUNTERED
			3168X	*			WHEN PROCESSING FILES.
			3169X	*			
			3170X	*	ENTRY	(A) = ERROR CODE	
			3171X	*		(HL) = ADDRESS OF FILE NAME - FB.NAM	
			3172X	*	EXIT	TO RESTART	
			3173X	*	USES	ALL	
			3174X				
			3175X				
054.274	365		3176X	\$FERROR	PUSH	PSW	SAVE CODE
054.275	315 136 031		3177X		CALL	\$TYPTX	
054.300	012 007 105		3178X		DB	NL,BELL,'ERROR ON FILE','+2000	
054.320	021 012 000		3179X		LXI	D,FB.NAM	
054.323	031		3180X		DAD	D	
			3181X				
			3182X	*		PRINT FILE NAME	
			3183X				
054.324	176		3184X	\$FERR1	MOV	A,M	
054.325	043		3185X		INX	H	ADVANCE MESSAGE
054.326	247		3186X		ANA	A	
054.327	312 340 054		3187X		JZ	\$FERR2	
054.332	315 127 054		3188X		CALL	\$WCHAR	
054.335	303 324 054		3189X		JMP	\$FERR1	
			3190X				
			3191X	*		TYPE ERROR MESSAGE	
			3192X				

054.340	315	136	031	3193X	\$FERR2	CALL	\$TYPTX	
054.343	040	055	240	3194X		DB	'-',''+200G	
054.346	046	012		3195X		MVI	H,NL	
054.350	361			3196X		POP	PSW	(A) = CODE
054.351	377	057		3197X		DB	SYSCALL,ERROR	
054.353	303	327	042	3198X		JMP	RESTART	EXIT
054.356				3199		XTEXT	DTB	

3201X \*\* \$DTB - DELETE TRAILING BLANKS.  
 3202X \*  
 3203X \* \$DTB DELETES THE TRAILING BLANKS FROM A CODED LINE.  
 3204X \*  
 3205X \* ENTRY (HL) = LINE FWA  
 3206X \* EXIT (AY) = LENGTH OF RESULT (ENCLUDING '00' TERMINATOR BYTE)  
 3207X \* USES A,F  
 3208X  
 3209X

054.356	325			3210X	\$DTB	PUSH	D	SAVE (DE)
054.357	124			3211X		MOV	D,H	
054.360	135			3212X		MOV	E,L	(DE) = FWA
054.361	033			3213X		DCX	D	(DE) = FWA-1
054.362	176			3214X	\$DTB1	MOV	A,M	
054.363	043			3215X		INX	H	
054.364	247			3216X		ANA	A	FIND END OF LINE
054.365	302	362	054	3217X		JNZ	\$DTB1	
054.370	053			3218X		DCX	H	(HL) = ADDRESS OF TERMINATING ZERO BYTE

3219X  
 3220X \* GOT END OF LINE. DELETE TRAILING BLANKS  
 3221X

054.371	053			3222X	\$DTB2	DCX	H	BACKUP ONE CHARACTER
054.372	315	216	030	3223X		CALL	\$CDEHL	
054.375	312	006	055	3224X		JE	\$DTB3	GDNE PAST FRONT OF LINE, MUST BE ALL BLANKS
055.000	176			3225X		MOV	A,M	
055.001	376	040		3226X		CPI	' '	
055.003	312	371	054	3227X		JE	\$DTB2	GOT BLANK

3228X  
 3229X \* HAVE TRIMED LINE. COMPUTE LENGTH  
 3230X

055.006	043			3231X	\$DTB3	INX	H	
055.007	066	000		3232X		MVI	M,0	TERMINATE LINE
055.011	175			3233X		MOV	A,L	
055.012	223			3234X		SUB	E	(A) = LENGTH +1 (FOR '00' BYTE)
055.013	353			3235X		XCHG		
055.014	043			3236X		INX	H	(HL) = LINE FWA
055.015	321			3237X		POP	D	RESTORE (DE)
055.016	311			3238X		RET		
000.001				3239	\$CMP*	EQU	1	
055.017				3240		XTEXT	TYPLN	

```

3242X **      $TYPLN - TYPE LINE.
3243X *
3244X *      $TYPLN IS CALLED TO TYPE A LINE OF TEXT. ZERO BYTES ARE
3245X *      TAKEN AS CRLF (WITH THE PROPER PADDING)
3246X *
3247X *      CALL      $TYPLN
3248X *      DB        N          BYTE COUNT OF FOLLOWING MESSAGE
3249X *      DB        'N-CHARACTER MESSAGE'
3250X *
3251X *      ENTRY    (RET) = TEXT COUNT
3252X *      (RET)+1 - (RET)+N = TEXT
3253X *      EXIT     TO (RET)+N+1
3254X *      USES     A,F
3255X *
3256X
3257X
055.017 343   3258X $TYPLN, XTHL      (HL) = COUNT ADDRESS
055.020 176   3259X      MOV      A,M          (A) = COUNT
055.021 043   3260X      INX      H          (HL) = TEXT ADDRESS
055.022 345   3261X      PUSH     H          SAVE TEXT FWA
055.023 315 072 030 3262X      CALL     $BADA        CALCULATE RETURN ADDRESS
055.026 343   3263X      XTHL      (HL) = TEXT ADDRE
055.027 315 035 055 3264X      CALL     $TYPL.        OUTPUT LINE
055.032 341   3265X      POP      H          (HL) = RETURN ADDRESS
055.033 343   3266X      XTHL      RESTORE (HL), SET RETURN ADDRESS
055.034 311   3267X      RET
3268X
3269X **      $TYPL. - TYPE LINE.
3270X *
3271X *      ENTRY    (HL) = ADDRESS
3272X *      (A) = COUNT
3273X *      EXIT     NONE
3274X *      USES     A,F,H,L
3275X
055.035      3276X $TYPL. EQU      *
055.035 247   3277X      ANA      A
055.036 310   3278X      RZ          NOTHING TO TYPE
055.037 365   3279X      PUSH     PSW        SAVE COUNT
055.040 176   3280X      MOV      A,M          (A) = CHARACTER
055.041 043   3281X      INX      H
055.042 247   3282X      ANA      A
000.001      3283X      IF      $CMP$        IF HAVE COMPRESSED SPACES
3284X      JM      TPL2        IS COMPRESSED SPACE
3285X      ENDF
055.043 314 006 054 3286X      CZ      $CRLF
055.046 315 020 054 3287X      CALL     $TYFC.        TYPE CHARACTER
055.051 361   3288X TPL1  POP      PSW
055.052 075   3289X      BCR      A
055.053 302 035 055 3290X      JNZ     $TYPL.
055.056 311   3291X      RET
000.001      3292X      IF      $CMP$        IF COMPRESSED TEXT
3293X
3294X *      HAVE COMPRESSED SPACE.
3295X
3296X TPL2  DCR      A
3297X      CP      $TYFCH        TYPE 00 IF CHARACTER WAS 2000

```

3298X	DB	0	
3299X	ANA	A	SET CODES
3300X	TPL3	JP	TPL1 ALL EXPANDED
3301X	PUSH	PSW	SAVE COUNT
3302X	CALL	\$TYPCH	
3303X	DB	/	
3304X	POP	PSW	
3305X	DCR	A	
3306X	JMP	TPL3	
3307X	ENDIF		

055.057	000	3310	COMAND	DB	0	COMMAND IN PROGRESS
055.060	000	3311	MODE	DB	0	<>0 IF LINE PASSED ON STACK
055.061	001	3312	SYSTEM	DB	1	/S FLAG (=0 IF /S SPECIFIED)
		3313				
055.062	130 130 130	3314	DIRNAM	DB	'XXX:DIRECT.SYS',0	DIRECTORY FILE NAME
		3315				
055.101	123 131 060	3316	DEFAULT	DB	'SY0',0,0,0	DEFAULT DEVICE AND EXTENSION
		3317				
055.107	327 055	3318	BUFFTR	DW	BUFF	POINTER TO START OF BUFFER
055.111	000 000	3319	BUFSIZ	DW	0	BUFFER LENGTH

3321 \*\* FILE BLOCKS

		3322				
055.113		3323	DESTFB	DS	0	DUMY BUFFER
055.113	310	3324		DB	200	ILLEGAL CHANNEL NUMBER
055.114	000	3325		DB	0	FLAGS
055.115	000 000	3326		DW	0	
055.117	000 000	3327		DW	0	
055.121	000 000	3328		DW	0	
055.123	000 000	3329		DW	0	END OF BLOCK
055.125		3330		DS	FB.NAML	NAME AREA

055.146	000 000	3332	NAMTLEN	DW	0	NAME TABLE POINTER
055.150	000 000	3333	NAMTMAX	DW	0	MAXIMUM SIZE OF NAME TABLE
055.152	000 000	3334	NAMTPTR	DW	0	POINTER TO ACTIVE ELEMENT IN NAMTAB
		3335				
		3336				
055.154		3337	PATCH	DS	64	

PRS - PRESET PROGRAM (OVERLAID BY BUFFERS).

PRS

16:09:46 16-MAY-80

```

3341 *** PRS - PRESET PIP PROGRAM.
3342 *
3343 * PRS IS CALLED TO PERFORM ONE-TIME-ONLY PRESETTING OF
3344 * THE PROGRAM ENVIRONMENT.
3345 *
3346 * THE CODE IS OVERLAID BY BUFFERS AND WORK AREAS WHEN PIP IS RUNNING.
3347 *
3348 * ENTRY NONE
3349 * EXIT NONE
3350 * USES ALL
3351
3352
055.254 3353 ENTRY EQU * INITIAL ENTRY POINT
055.254 377 011 3354 PRS DB SYSCALL,,VERS
055.256 332 160 056 3355 JC PRS1 NO ,VERS SYSTEM CALL
055.261 376 026 3356 CPI VERS
055.263 302 160 056 3357 JNZ PRS1
055.266 076 377 3358 MVI A,3770
055.270 377 046 3359 DB SYSCALL,,CLOSE CLOSE THE CHANNEL THAT WE CAME IN ON
055.272 041 327 055 3360 LXI H,RMEML (HL) = RUN-TIME HIGH MEMORY
055.275 377 052 3361 DB SYSCALL,,SETTP SET HI MEMORY
055.277 332 007 050 3362 JC ERROR
055.302 315 166 056 3363 CALL *DOS DISMOUNT OPERATING SYSTEM
055.305 332 007 050 3364 JC ERROR
055.310 257 3365 XRA A
055.311 062 033 041 3366 STA S,ICCS SET DEFAULT CLUSTER SIZE
055.314 041 336 042 3367 LXI H,CCHIT
055.317 076 003 3368 MVI A,CTLC
055.321 377 041 3369 DB SYSCALL,,CTLC SET CTL-C PROCESSING
055.323 315 136 031 3370 CALL $TYPTX
055.326 012 011 011 3371 DB NL,TAB,TAB,TAB,' ',SYSGEN'
055.343 012 011 011 3372 DB NL,TAB,TAB,TAB,'Version: ',VERS/16+0',',',VERS&OFH+0'
055.364 012 011 011 3373 DB NL,TAB,TAB,' ',issue: $50.05.00'
056.016 212 3374 DB ENL
056.017 315 136 031 3375 CALL $TYPTX
056.022 012 111 156 3376 DB NL,'Insert the System Distribution Source Disk. Hit RETURN when ready:',', '+2000
056.126 377 001 3377 PRS0 DB SYSCALL,,SCIN
056.130 332 126 056 3378 JC PRS0
3379
3380 * READ NEW DISK'S LABEL
3381
056.133 315 164 046 3382 CALL GETLAB READ NEW DISK'S LABEL
056.136 332 007 050 3383 JC ERROR
056.141 315 143 046 3384 CALL MND MOUNT NEW DISK
056.144 332 007 050 3385 JC ERROR
056.147 072 000 027 3386 LDA LABEL+LAB.SER
056.152 062 271 054 3387 STA VOLSER SET CURRENT VOLUME NUMBER
056.155 303 200 042 3388 JMP START START PROGRAM
3389
056.160 076 050 3390 PRS1 MVI A,EC.NCV NOT CORRECT VERSION
056.162 067 3391 STC
056.163 303 007 050 3392 JMP ERROR
3393
056.166 3394 XTEXT IOS

```

```

3396X ** $DOS - DISMOUNT OPERATING SYSTEM.
3397X *
3398X * $DOS DISMOUNTS SY2:, SY1: (IF MOUNTED), AND SY0: /79.11.6C/
3399X *
3400X * THE USER IS MESSAGED ABOUT THE DISKS, AND THE OPERATING
3401X * SYSTEM IS NOTIFIED.
3402X *
3403X *
3404X * ENTRY NONE
3405X *
3406X * EXIT (PSW) = 'C' CLEAR IF NO ERROR
3407X * 'C' SET IF ERROR
3408X * (A) = ERROR CODE
3409X *
3410X * USES ALL
3411X *
3412X *
056.166 315 136 031 3413X $DOS CALL $TYPTX
056.171 012 007 104 3414X DB NL,BELL,'Dismounting All Disks:',NL,ENL
3415X
056.223 076 000 3416X MVI A,DVLO
056.225 377 010 3417X DB SYSCALL,,LOAD0
056.227 330 3418X RC
056.230 076 001 3419X MVI A,DVLI
056.232 377 010 3420X DB SYSCALL,,LOAD0
056.234 330 3421X RC
3422X
056.235 041 375 056 3423X LXI H,DOSC
056.240 315 353 056 3424X CALL DOS.
056.243 330 3425X RC
056.244 041 370 056 3426X LXI H,DOSB
056.247 315 353 056 3427X CALL DOS.
056.252 330 3428X RC FATAL ERROR
056.253 041 363 056 3429X LXI H,DOSA
056.256 315 353 056 3430X CALL DOS.
056.261 330 3431X RC
3432X
056.262 315 136 031 3433X CALL $TYPTX
056.265 012 122 145 3434X DB NL,'Remove the Disk(s). Hit RETURN when ready:','+2000
056.341 315 121 054 3435X DOS1 CALL $RCHAR READ CHARACTER
056.344 376 012 3436X CFI NL
056.346 302 341 056 3437X JNE DOS1
056.351 247 3438X ANA A CLEAR CARRY
056.352 311 3439X RET
3440X
3441X * DISMOUNT A DEVICE WITHOUT REGARD TO WHETHER MOUNTED OR NOT
3442X *
056.353 377 201 3443X DOS1 DB SYSCALL,,DMOUN
056.355 320 3444X RNC
056.356 376 042 3445X CPI EC,NVM NO VOLUME MOUNTED ERROR NOT CONSIDERED FATAL
056.360 310 3446X RZ NOT FATAL, CARRY NOW CLEAR
056.361 067 3447X STC FLAG FATAL ERROR
056.362 311 3448X RET
3449X
056.363 123 131 060 3450X DOSA DB 'SY0:',0
056.370 123 131 061 3451X DOSB DB 'SY1:',0

```

056.375 123 131 062 3452X DOSC DB SY21,0

3453  
057.002 3454 MEML EQU \* MEMORY LENGTH



3457 \*\* THE FOLLOWING BUFFERS AND AREAS OVERLAY THE PRS CODE.  
3458  
055.254 3459 ORG FRS  
3460  
3461  
3462  
055.254 3463 MWNA DS FB.NAML MWN WORK AREA  
3464  
3465  
3466 \*\* \* \* NOTE \* \*  
3467 \* DIRWORK USES THE SYSTEM SCRATCH AREA, SECSCK, DIRWORK WILL NOT  
3468 \* BE PRESERVED DURING A SYSCALL !!  
3469  
027.000 3470 LABEL EQU S.GRT2+256 USE THE EXTRA GRT AREA  
3471  
041.120 3472 DIRWKP EQU S.SCR POINTER TO THE SCRATCH AREA  
  
3474 \*\* PID.XXX - IMAGE OF SYSTEM AID.XXX AREA  
3475 \*  
3476 \* THESE CELLS MIRROR THE SYSTEM AID.XXX AREA  
3477  
3478  
055.275 3479 PID.DEV DS 2 DEVICE CODE  
055.277 3480 PID.UNI DS 1 UNIT NUMBER (0-9)  
3481  
055.300 3482 PID.DIR DS DIRELEN DIRECTORY ENTRY  
3483  
3484  
055.327 3485 NAMTAB DS 0 NAME TABLE  
3486  
3487  
002.000 3488 BUFMINL EQU 512 MINIMUM SIZE FOR BUFFER (WHEN IN USE)  
055.327 3489 BUFF EQU \* BUFFER AREA STARTS AFTER NAMTAB  
3490  
055.327 3491 RMEML EQU \* INITIAL RUNNING MEMORY LENGTH  
3492  
3493  
3494  
055.327 3495 END  
ASSEMBLY COMPLETE  
3495 STATEMENTS  
0 ERRORS DETECTED  
9660 BYTES FREE



SYSGEN - GENERATE NEW SYSTEM  
CROSS REFERENCE TABLE

XREF V1.1  
PAGE 75

.DISMT	000061	429L				
.DLEDS	040021	784E				
.DLY	000053	755E				
.DMNMS	000203	440L	883	1537		
.DMOUN	000201	438L	3443			
.DOD	003122	769E				
.DODA	003356	771E				
.DSPMOD	040007	780E				
.DSPROT	040006	779E				
.DUMP	001374	757E				
.ERROR	000057	427L	1767	3197		
.EXIT	000000	395L	916	1754		
.HORN	002140	759E	1567			
.IDENT	000000	754E				
.IQWRK	040002	777E				
.LINK	000040	412L				
.LOAD	001267	756E				
.LOADD	000062	430L				
.LOADO	000010	403L	3417	3420		
.MFLAG	040010	781E	1550			
.MONMS	000202	439L	1633			
.MOUNT	000200	437L				
.NAME	000054	424L				
.OFENC	000045	417L	1379			
.OFENR	000042	414L	1157	2333		
.OFENU	000044	416L	1388			
.OPENW	000043	415L	1358			
.PCHL	002264	762E				
.POSIT	000047	419L	1219	1397		
.PRINT	000003	398L	1563	1778		
.RCK	003260	770E				
.READ	000004	399L	1233	2344		
.REGI	040005	778E				
.REGPTR	040035	789E				
.RENAM	000051	421L				
.RESET	000204	441L				
.RNB	002331	765E				
.RNF	002325	764E				
.SCIN	000001	396L	1571	1579	2997	3377
.SCOUT	000002	397L	2743	2765	3001	3022
.SETTP	000052	422L	2218	2530	3361	
.SRS	002265	763E				
.START	040000	776E				
.SYSRES	000012	405L				
.TICCNT	040033	788E				
.TPERR	002205	761E				
.TPERRX	040031	787E				
.UIVEC	040037	790E				
.VERS	000011	404L	3354			
.WNB	003024	768E				
.WNP	003017	767E				
.WRITE	000005	400L	1412			
ABS.COD	000010	848L	851			
ABS.ENT	000006	846L				
ABS.II	000000	842L				
ABS.LIA	000002	844L				
ABS.LEN	000004	845L				
AC.DLY	000156	43E				



SYSDEN - GENERATE NEW SYSTEM

XREF 01.1

CROSS REFERENCE TABLE

PAGE 77

CFS1	050301	1873L	1878						
CN.DES	000001	25E	1357	1370	1378	1387	1396	1410	1414
CN.DIR	000002	26E	2332	2342	2396				
CN.SOU	000000	24E	1156	1176	1218	1232	1274		
CO.FLG	000001	678E							
COMMAND	055057	3310L							
COPY	043066	974E							
CR	000015	446E							
CS.FLG	000200	679E							
CSF	050311	1896L	2378						
CSF1	050317	1901L	1912						
CSF2	050346	1907	1917L						
CSFA	050352	1900	1922L	1923					
CSL.CHR	000001	656E							
CSL.ECH	000200	654E							
CSL.WRF	000002	655E							
CTLA	000001	461E							
CTLB	000002	462E							
CTLC	000003	463E	3368						
CTLD	000004	464E							
CTLO	000017	465E							
CTLP	000020	466E							
CTLR	000021	467E							
CTLS	000023	468E							
CTLZ	000032	469E							
CTP.2SB	000010	664E							
CTP.BKM	000002	665E							
CTP.BKS	000200	661E							
CTP.MLI	000040	662E							
CTP.MLD	000020	663E							
CTP.TAB	000001	666E							
DWM	051022	1943L	1951	2371					
DWM1	051031	1945	1948L						
D.CON	040110	367L							
D.DLYHS	040244	487L							
D.DLYMD	040243	486L	1543						
D.DRVTR	040251	492L	979						
D.DVCTL	040242	484L							
D.E.CHS	040267	503L							
D.E.HCK	040270	504L							
D.E.HSY	040266	502L							
D.E.MDS	040265	501L							
D.E.TRK	040272	506L							
D.E.VOL	040271	505L							
D.ERR	040265	500L							
D.ERR1	040273	507L							
D.HECNT	040261	494L							
D.OECNT	040264	496L							
D.OPK	040273	511L							
D.OPW	040275	512L							
D.RAM	040240	370L	479	514					
D.RAML	000037	514E							
D.SECNT	040262	495L							
D.TRKFT	040245	489L							
D.TS	040241	482L							
D.TT	040240	481L							
D.VEC	040130	369L							
D.VOLPT	040247	490L							



SYSGEN - GENERATE NEW SYSTEM  
 CROSS REFERENCE TABLE

XREF V1.1  
 PAGE 79

DIS,SEC	001374	269L			
DM,MR	000000	729E			
DM,MM	000001	730E			
DM,RR	000002	731E			
DM,RW	000003	732E			
DNT	051355	2053	2069	2100	2146L
DNT1	051364	2150L	2153		
DNT2	051375	2158L	2181		
DNT3	052037	2161	2168	2176L	
DNT4	052062	2166	2170	2172	2199L
DNT5	052051	2164	2190L	2194	
DNTA	052067	2146	2154	2200	2203L
DOS,	056353	3424	3427	3430	3443L
DOS1	056341	3435L	3437		
DOSA	056363	3429	3450L		
DOSB	056370	3426	3451L		
DOSC	056375	3423	3452L		
DR,IM	000001	192E			
DR,PR	000002	193E			
DT,CR	000002	199E			
DT,CW	000004	200E			
DT,DP	000001	198E	2324		
DV,EL	000000	188E			
DV,NU	000001	189E			
ERM	052100	991	2215L		
EC,CNA	000004	294L			
EC,DDA	000027	313L			
EC,DIF	000017	305L			
EC,DIW	000035	319L			
EC,DNI	000045	327L			
EC,DNR	000046	328L			
EC,DNS	000005	295L	2325		
EC,DSC	000047	329L			
EC,EDF	000001	291L	1239		
EC,EOM	000002	292L			
EC,FAO	000031	315L			
EC,FAF	000026	312L			
EC,FL	000030	314L			
EC,FNF	000014	302L	1374		
EC,FNO	000011	299L			
EC,FNR	000034	318L			
EC,FOD	000043	325L			
EC,FUC	000013	301L			
EC,ICN	000016	304L			
EC,IDN	000006	296L			
EC,IFC	000020	306L			
EC,IFN	000007	297L	2125		
EC,ILC	000003	293L			
EC,ILO	000040	322L			
EC,ILR	000012	300L			
EC,ILV	000037	321L			
EC,IOI	000052	332L			
EC,IS	000032	316L			
EC,NCV	000050	330L	3390		
EC,NEM	000021	307L	2422		
EC,NOS	000051	331L			
EC,NPM	000044	326L			
EC,NRD	000010	298L			





SYSGEN - GENERATE NEW SYSTEM  
 CROSS REFERENCE TABLE

XREF V1.1  
 PAGE 81

FDNELEN	000012	1136	1507	3055E	3151			
FDNFRE	054146	1097	1131	1135	1455	1458	1503	3068L
FDNFWA	054146	3066E	3154					
FDNHEAD	054147	1014	1083	1108	1308	1454	1505	3069L
FF	000014	460E						
FT.ABS	000000	832E	852					
FT.BAC	000003	835E						
FT.BD	000001	232E						
FT.DR	000002	233E						
FT.OU	000010	235E						
FT.DW	000004	234E						
FT.FIC	000001	833E						
FT.REL	000002	834E						
GETLAB	046164	1584	1635	1653L	1672	1703	1382	
HOS.SPG	000002	797E						
I.CONFL	000004	681E	682					
I.CONTY	000001	668E	669					
I.CONWI	000003	674E	675					
I.CSLMD	000000	658E						
I.CUSOR	000002	671E	672					
IERR1	047173	1398	1736L	2219				
IERR2	047200	1739L						
IERR3	047205	1220	1741L					
IFL	045314	976	1501L					
IFL1	045331	1507L	1512					
INA	053127	1817	2415L					
INTERR	047212	1737	1740	1742	1745L			
IOC.CGN	000010	240L						
IOC.CSI	000011	241L						
IOC.DDA	000002	229L	236	250				
IOC.DES	000016	247L						
IOC.DEV	000020	248L						
IOC.DIL	000021	250E						
IOC.DIR	000023	252L	1178	1193				
IOC.DRL	000010	244E						
IOC.DTA	000014	246L						
IOC.FLG	000004	231L	244					
IOC.GRT	000005	238L	1190					
IOC.LGN	000012	242L						
IOC.LNK	000000	228L						
IOC.LSI	000013	243L						
IOC.SPG	000007	239L						
IOC.SQL	000003	236E						
IOC.UNI	000022	249L						
IOCTD	000001	256E	1174					
IOCELEN	000052	254E						
IP.FAD	000360	715E	1573					
LAB.BAT	000000	816E	1675					
LAB.DIS	000003	812L						
LAB.GRT	000005	813L						
LAB.IND	000001	811L						
LAB.LAB	000021	823L	824					
LAB.LEL	000074	824E						
LAB.NOD	000002	818E						
LAB.SER	000000	810L	1592	3386				
LAB.SPG	000007	814L						
LAB.SYS	000001	817E	896	1678	1706			
LAB.VER	000011	821L						







SYSGEN - GENERATE NEW SYSTEM  
CROSS REFERENCE TABLE

XREF V1.1  
PAGE 85

UC,DR	000001	83E	
UC,DRL	000010	95E	
UC,DSR	000040	97E	
UC,DTR	000001	76E	
UC,EDA	000001	54E	
UC,EPS	000020	70E	
UC,FE	000010	86E	
UC,IID	000006	61E	
UC,IIF	000001	60E	
UC,LDD	000020	80E	
UC,MSI	000010	57E	
UC,OR	000002	84E	
UC,QU1	000004	78E	
UC,QU2	000010	79E	
UC,PE	000004	85E	
UC,PEN	000010	69E	
UC,RI	000100	98E	
UC,RLS	000200	99E	
UC,RSI	000004	56E	
UC,RTS	000002	77E	
UC,SB	000100	72E	
UC,SKP	000040	71E	
UC,TER	000004	94E	
UC,THE	000040	88E	
UC,TRE	000002	55E	
UC,TSE	000100	89E	
UCI,ER	000020	133E	
UCI,IE	000002	135E	
UCI,IR	000100	131E	
UCI,RE	000004	134E	
UCI,RD	000040	132E	
UCI,TE	000001	136E	
UDDN1	054027	2911L	2927
UDDN1.5	054061	2931L	2938
UDDN2	054063	2924	2936L
UDDN3	054064	2937L	2941
UDR	000000	108E	
UMI,16X	000002	126E	
UMI,1R	000100	116E	
UMI,1X	000001	125E	
UMI,2R	000300	118E	
UMI,64X	000003	127E	
UMI,HR	000200	117E	
UMI,L5	000000	121E	
UMI,L6	000004	122E	
UMI,L7	000010	123E	
UMI,L8	000014	124E	
UMI,PA	000020	120E	
UMI,PE	000040	119E	
UNT,DIS	000005	219L	
UNT,FLG	000000	216L	
UNT,GRT	000001	217L	
UNT,GTS	000003	218L	
UNT,SIZ	000007	221E	
UD,CLK	000001	741E	1549
UD,DDU	000002	740E	1549
UD,HLT	000200	738E	1549
UD,NFR	000100	739E	

.....CROSS REFERENCE TABLE.....

UR.DLL	000000	49E			
UR.DLM	000001	51E			
UR.IER	000001	53E			
UR.IIR	000002	59E			
UR.LCR	000003	63E			
UR.LSR	000005	82E			
UR.MCR	000004	75E			
UR.MSR	000002	91E			
UR.RBR	000000	45E			
UR.THR	000000	47E			
USERFWA	042200	379E	851	853	854
USR	000001	109E			
USR.FE	000040	140E			
USR.OE	000020	141E			
USR.FE	000010	142E			
USR.RXR	000002	144E			
USR.TXE	000004	143E			
USR.TXR	000001	145E			
VERS	000026	386E	3356	3372	3372
VOLFLAG	054270	978	998	1006	1602
VOLSER	054271	980	1591	3157L	3156L
WPH	044353	1013	1304E	1469	
WPH0	045006	1317	1332L		
WPH1	045060	1355	1364L		
WPH1.5	045105	1373	1376L		
WPH2	045123	1337	1386L		
WPH3	045155	1360	1381	1402L	
WPH4	045230	1327	1434L		

.....18484 BYTES FREE.....