

0000		0001 *****
0000		0002 *
0000		0003 * ESFTOS VERSION 3.0 03/03/79
0000		0004 * THIS MONITOR ADDS ESF SUPPORT TO ANY SYSTEM
0000		0005 * THAT HAS A VDM/VTI/ OR VBI
0000		0006 *****
0000		0007 *
0000		0008 * SYSTEM EQUATES
0000		0009 *
0000		0010 *****
0000		0011 KDDR EQU 1
0000		0012 PPLDR EQU 2
0000		0013 XMITDR EQU 4
0000		0014 SERDR EQU 1
0000		0015 TTBE EQU 128
0000		0016 SEROK EQU 20H
0000	00	STATUS EQU 0FAH0000
0000		0017 SDATA EQU 0F0H
0000		0020 PDATA EQU 0FDH
0000	000	0022 DIWT EQU 0B014H
0000		0023 D1GET EQU 0B00BH
0000		0024 D1SAV EQU 0B00EH
0000		0025 D1C1 EQU 0B008H
0000		0026 D1CE1 EQU 0B011H
0000		0027 BLKOF EQU 7
0000		0028 * <i>6B</i>
0000		0029 * SYSTEM JUMP TABLE
0000		0030 *
0000		0031 *****
0000	F3	0032 START DI
0000	C3 38 02	0033 INITA JMP STRTA JUMP TO COMMAND INITIALIZE
0004	C3 5F 02	0034 RETRN JMP COMND SYSTEM REENTRY POINT
0007	C3 4B 07	0035 FOPEN JMP BOPEN JUMP TO FILE OPEN RTN
000A	C3 7B 07	0036 FCLOS JMP PCLOS JUMP TO FILE CLOSE RTN
000D	C3 CD 07	0037 RDBYT JMP RTBYT JUMP TO READ BYTE RTN
0012	C3 47 08	0038 WRBYT JMP WTBYT JUMP TO WRITE BYTE RTN
0013	C3 C3 06	0039 RDBLK JMP RTAPE JUMP TO BLOCK READ RTN
0016	C3 A3 06	0040 WRBLK JMP WTAPE JUMP TO BLOCK WRITE RTN
0019	3A 0F 0D	0041 SOUT LDA OPORT
001C	C3 AF 00	0042 AGUT JMP OUTPR JUMP TO OUTPUT RTN
001F	3A 0E 0D	0043 SINP LDA IPORT
0022	C3 A3 00	0044 AINP JMP INPR JUMP TO INPUT RTN
0025		0045 *****
0025	DB FA	0046 PPLIN IN STATUS GET PARALLEL STATUS
0027	2F	0047 CMA . COMPLEMENT ACCUM IF NEEDED
0028	E6 02	0048 ANI PPLDR SEE IF ANY THING WAITING
002A	C8	0049 RZ . RETURN IF NOT
002B	DB FD	0050 IN PDATA ELSE GET THE DATA
002D	C9	0051 RET . AND RETURN
002E		0052 *
002E	DB FA	0053 PPLOUT IN STATUS GET STATUS OF PPL PORT
0030	00	0054 NOP . COMPLEMENT ACCUM IF NEEDED
0031	E6 04	0055 ANI XMITDR OK TO TRANSMIT?
0033	C2 2E 00	0056 JNZ PPLOUT IF NOT LOOP TILL OK
0036	78	0057 MOV A,B DATA TO ACCUM
0037	D3 FD	0058 OUT PDATA AND PUT IT OUT
0039	C9	0059 RET . AND RETURN

003A DB FA 03 134 DT  
 003C 2F  
 003D E6 01  
 003F C8  
 0040 DB FC  
 0042 C9  
 0043 DB F5  
 0045 00  
 0046 E6 01  
 0048 C8  
 0049 DB F0  
 004E C9  
 004C DB F5  
 004E 2F  
 004F E6 20  
 0051 C2 4C 00  
 0054 78  
 0055 D3 F0  
 0057 C9  
 0058  
 0059 49  
 0059 28 00  
 005E C3 14 B0 F0  
 005E C3 0B B0 F0  
 0061 C3 0E B0 F0  
 0064 C3 08 B0 F0  
 0067 C3 11 B0 F0  
 006A C9  
 006B 45 53 46 50  
 41 54 43 48  
 45 53 46 50  
 41 54 43 48  
 45 53 46 50  
 41 54 43 48

0060 \*  
 0061 KBDIN IN STATUS GET KEYBOARD STATUS  
 0062 CMA . COMPLEMENT ACCUM IF NEEDED  
 0063 ANI KBDDR SEE IF ANY THING WAITING  
 0064 RZ . RETURN IF NOT  
 0065 IN KDATA ELSE GET THE CHARACTER  
 0066 RET . AND RETURN  
 0067 \*  
 0068 SERIN IN SSTAT GET SERIAL STATUS  
 0069 NOP . COMPLEMENT ACCUM IF NEEDED  
 0070 ANI SERDR SEE IF ANY THING WAITING  
 0071 RZ . RETURN IF NOT  
 0072 IN SDATA ELSE GET THE CHARACTER  
 0073 RET . AND RETURN  
 0074 \*  
 0075 SEROUT IN SSTAT GET SERIAL STATUS  
 0076 CMA . COMPLEMENT ACCUM IF NEEDED  
 0077 ANI SEROK SEE IF OK TO TRANSMIT  
 0078 JNZ SEROUT IF NOT LOOP TILL OK  
 0079 MOV A,B MOVE DATA TO ACCUM  
 0080 OUT SDATA AND PUT IT OUT  
 0081 RET . AND RETURN  
 0082 \*  
 0083 DB 49H  
 0084 DW EXEND-EWTP1  
 0085 EWTP1 JMP DIWT EXITS  
 0086 ESFG1 JMP D1GET TO  
 0087 ESFS1 JMP D1SAV ESF  
 0088 ESFC1 JMP D1C1 UTILITY  
 0089 ESCE1 JMP D1CE1 PACKAGE  
 0090 EXINT RET call EPAGE  
 0091 PATCH ASC 'ESFPATCHESFPATCHESFPATCH'  
 0092 EXEND EQU \$  
 0093 \*  
 0094 EWTP2 JMP NOXST EXITS  
 0095 ESFG2 JMP NOXST TO ESF  
 0096 ESFS2 JMP NOXST UTILITY  
 0097 ESFC2 JMP NOXST IF IT  
 0098 ESCE2 JMP NOXST EXISTS  
 0099 \*  
 0100 DB 21H  
 0101 WHERE DW START LOCATION OF ESFTOS  
 0102 DB 40H  
 0103 NOP  
 0104 VDM DW 0CC00H LOCATION OF VDM  
 0105 SCREND DB 0D0H FIRST BYTE AFTER VIDIO 1  
 0106 OMASK DB 0 HI ORDER BIT MASK  
 0107 \* USE 00 FOR VDM OR VBI AND 80 FOR VTI  
 0108 CURCH DB 080H CURSER CHAR  
 0109 \* USE 80 FOR VDM, 7F FOR VBI, AND FF FOR VTI  
 0110 \*  
 0111 \*\*\*\*\*

0083  
 0083  
 0083 C3 9E 00 14 F0  
 0086 C3 9E 00 0B F0  
 0089 C3 9E 00 0E F0  
 008C C3 9E 00 08 F0  
 008F C3 9E 00 11 F0  
 0091  
 0092 21  
 0093 00 00 A0  
 0095 40  
 0096 00  
 0097 00 00 EC  
 0099 00 F0  
 009A 00  
 009B 00 7F  
 009C  
 009C  
 009C

009C 3E 10	0112 NOXST MVI A,10H SET UP NO SECOND
009E 32 8B 0F	0113 STA FCBC UNIT EXISTS
00A1 37	0114 STC
00A2 C9	0115 RET
00A3	0116 INPR EQU # *PROCESS
00A3 E5	0117 PUSH H INPUT
00A4 21 8A 03	0118 LXI H,INTBL DEVICES
00A7	0119 IOPRC EQU # PROCESS
00A7 E4 03	0120 ANI 03H OUTPUT
00A9 07	0121 RLC DEVICES
00AA 85	0122 ADD L
00AB 6F	0123 MOV L,A
00AC C3 E6 02	0124 JMP DISPT
00AF E5	0125 OUTPR PUSH H
00B0 21 82 03	0126 LXI H,OUTBL
00B3 C3 A7 00	0127 JMP IOPRC
00B6	0128 VMDR EQU # VIDIO DRIVER
00B6 E5	0129 PUSH H ENTRY
00B7 D5	0130 PUSH D
00B8 C5	0131 PUSH B
00B9 3A 15 0D	0132 LDA ESCFL ESC SEG ?
00BC B7	0133 ORA A
00BD C2 E8 01	0134 JNZ ESCS JMP IF SO
00C0 73	0135 MOV A,B OP CHAR TO A
00C1 E6 7F	0136 ANI 07FH GET RID OF BIT 7
00C3 47	0137 MOV B,A CHAR BACK TO B
00C4 CA E3 00	0138 JZ GOBK RET IF NOT PRINTABLE
00C7 21 46 03	0139 LXI H,TBL POINT AT SPEC CHAR TBL
00CA CD E9 00	0140 CALL TSRCH SEARCH FOR SPEC CHAR
00CD	0141 GOBACK EQU #
00CD CD A1 01	0142 CALL VDADD CURR ADDR ON SCRNM
00D0 7E	0143 MOV A,M CHAR THERE TO A
00D1 32 10 0D	0144 STA SVCHR SAVE IT
00D4 3A 9B 00	0145 LDA CURCH GET CURSOR CHAR
00D7 B6	0146 ORA M COMBINE WITH CHAR ON SCRNM
00D8 77	0147 MOV M,A AND PUT ON SCREEN
00D9 2A 13 0D	0148 LHLD BOT BUMP
00DC 2C	0149 INR L CURR POS BY 1
00DD AF	0150 XRA A
00DE	0151 TIMER EQU # LOOP
00DE 2B	0152 DCX H TILL H=0
00DF BC	0153 CMP H
00E0 C2 DE 00	0154 JNZ TIMER
00E3	0155 GOBK EQU # RET TO CALLER
00E3 C1	0156 POP B RESTORE
00E4 D1	0157 POP D BC, DE, HL REGS
00E5 E1	0158 POP H
00E6 C9	0159 RET .
00E7	0160 NEXT EQU #
00E7 23	0161 INX H
00E8 23	0162 INX H
00E9	0163 TSRCH EQU # LOOK FOR
00E9 7E	0164 MOV A,M MATCH WITH
00EA B7	0165 ORA A CHAR IN A
00EB CA FB 00	0166 JZ CHAR JMP IF MATCH
00EE B8	0167 CMP B
00EF 23	0168 INX H #
00F0 C2 E7 00	0169 JNZ NEXT
00F3 E5	0170 PUSH H
00F4 CD BB 01	0171 CALL CREM DELETE CURSOR CHAR FROM SCREEN
00F7 E3	0172 XTHL .
00F8 C3 E6 02	0173 JMP DISPT

A = output port  
B = character to be output  
H = outtbl

ESC SEG ?  
OP CHAR ?  
CHAR ?

00FB		0174	CHAR EQU \$	
00FB 78		0175	MOV A,B	CHAR TO BE OP TO A
00FC FE 7F		0176	CPI 07FH	MINUS BIT 7
00FE C8		0177	RZ	RET IF RESULT IS ZERO
00FF		0178	OCHAR EQU \$	
00FF CD A1 01		0179	CALL VDADD	GET SCREEN ADDR
0102 3A 9A 00		0180	LDA OMASK	DETERMINE TYPE OF VIDIO
0105 B0		0181	ORA B	DEVICE
0106 77		0182	MOV M,A	AND OUTPUT BASED THEREON
0107 3A 11 0D		0183	LDA NCHAR	END OF LINE?
010A FE 3F		0184	CPI 03FH	
010C DA 39 01		0185	JC OK	JMP IF NOT
0A 12 0D	0186	LDA	LINE	BOTTOM OF SCREEN
0112 FE 0F		0187	CPI 00FH	
0114 C2 39 01		0188	JNZ OK	JMP IF NOT
0117 AF		0189	XRA A	
0118 32 11 0D		0190	STA NCHAR	
011B		0191	SROL EQU \$	SCROLL ENTIRE
011B 2A 97 00		0192	LHLD VDM	SCREEN
011E 54		0193	MOV D,H	ONE LINE
011F 1E 40		0194	MVI E,40H	
0121 C5		0195	PUSH B	
0122 06 03		0196	MVI B,3	
0124 0E C0		0197	MVI C,0C0H	
0126 1A		0198	SCLP LDAX D	
0127 77		0199	MOV M,A	
0128 23		0200	INX H	
0129 13		0201	INX D	
012A 0B		0202	DCX B	
012B 79		0203	MOV A,C	
012C B0		0204	ORA B	
012D C2 26 01		0205	JNZ SCLP	
0130 C1		0206	POP B	
0131 AF		0207	XRA A	
0132 4F		0208	MOV C,A	
0133 CD 79 01		0209	CALL CLIN1	
0136 C3 6F 01		0210	JMP ERAS3	
01	0211	OK	EQU \$	
0139 3A 11 0D		0212	LDA NCHAR	LINE POSITION
013C 3C		0213	INR A	PLUS 1
013D E6 3F		0214	ANI 03FH	END OF LINE?
013F 32 11 0D		0215	STA NCHAR	SAVE POSITION
0142 C0		0216	RNZ	RET IF NOT END OF LINE
0143		0217	PDOWN EQU \$	CURSOR
0143 3A 12 0D		0218	LDA LINE	DOWN ONE
0146 3C		0219	INR A	LINE
0147		0220	CURSC EQU \$	
0147 E6 0F		0221	ANI 00FH	SET BOTTOM OF PAGE IF NEEDED
0149		0222	CUR EQU \$	
0149 32 12 0D		0223	STA LINE	SAVE WHERE WE ARE AT
014C C9		0224	RET .	
014D		0225	PERSE EQU \$	
014D 2A 97 00		0226	LHLD VDM	GET VIDIO MEMORY ADDRESS
0A 9B 00	0227	LDA	CURCH	PLUS CURSOR CHAR
0153 F6 20		0228	ORI 20H	PLUS BLANKS TO
0155 77		0229	MOV M,A	SCREEN
0156 23		0230	INX H	

```

0157
0157 3A 9A 00
015A F6 20
015C 77
015D 23
015E 3A 99 00
0160 BC
0162 C2 57 01
0165 37
0166
0166 3E 00
0168 32 12 0D
016B 32 11 0D
016E D0
016F
016F 32 13 0D
0172 C9
0173
0173 CD A1 01
0176 3A 11 0D
0179
0179 FE 40
017B D0
017C C5
017D 3A 9A 00
0182 77
0183 C1
0184 23
0185 3C
0186 C3 79 01
0189
0189 3A 12 0D
018C 3D
018D C3 47 01
0190
0190 3A 11 0D
0193 3D
0194
0194 E6 3F
0196 32 11 0D
0199 C9
019A
019A 3A 11 0D
019D 3C
019E C3 94 01
01A1
01A1 3A 11 0D
01A4 4F
01A5
01A5 3A 12 0D

```

```

0231 ERAS1 EQU $ ERASE
0232 LDA OMASK PART
0233 ORI 020H OR
0234 MOV M,A ALL
0235 INX H OF
0236 LDA SCREND THE
0237 CMP H SCREEN
0238 JNZ ERAS1
0239 STC .
0240 PHOME EQU $ CURSOR
0241 MVI A,000H TO
0242 STA LINE HOME
0243 STA NCHAR POSITION
0244 RNC ON
0245 ERAS3 EQU $ THE
0246 STA BOT SCREEN
0247 RET
0248 CLINE EQU $
0249 CALL VDADD
0250 LDA NCHAR
0251 CLIN1 EQU $
0252 CPI 040H
0253 RNC
0254 PUSH PSW
0255 LDA OMASK
0256 ORI 020H
0257 MOV M,A
0258 POP PSW
0259 INX H
0260 INR A
0261 JMP CLIN1
0262 PUP EQU $ CURSOR
0263 LDA LINE UP
0264 DCR A 1
0265 JMP CURSC LINE
0266 PLEFT EQU $ CURSOR
0267 LDA NCHAR LEFT
0268 DCR A 1
0269 PCUR EQU $ CHAR
0270 ANI 03FH
0271 STA NCHAR
0272 RET .
0273 PRIT EQU $ CURSOR
0274 LDA NCHAR RIGHT
0275 INR A 1
0276 JMP PCUR CHAR
0277 VDADD EQU $ THIS ROUTINE DETERMINES
0278 LDA NCHAR WHERE THINGS
0279 MOV C,A ARE ON THE SCREEN
0280 VDAD2 EQU $
0281 LDA LINE

```

*PERCEE part of map*  
*2nd 12*

*LINE = Line on keyboard where cursor is  
NCHAR = Column on keyboard where cursor  
cursor character used for screen*

*cursor up one line*  
*cursor left one space*  
*cursor down one line*

*Bumps NCHAR RIGHT one space*

*THIS ROUTINE DETERMINES  
WHERE THINGS  
ARE ON THE SCREEN*

*V*

01A8				0282	VDAD EQU #	
01A8	6F			0283	MOV L,A	
01A9	3A	13	0D	0284	LDA BOT	
01AC	85			0285	ADD L	
01AD	0F			0286	RRC .	
01AE	0F			0287	RRC .	
01AF	6F			0288	MOV L,A	
01B0	E6	03		0289	ANI 003H	
01B2	C6	CC		0290	ADI 0CCH	"EC"
01B4	67			0291	MOV H,A	
01B5	7D			0292	MOV A,L	
01B6	E6	C0		0293	ANI 0C0H	
01B8	81			0294	ADD C	
01B9	6F			0295	MOV L,A	
01BA	C9			0296	RET .	
01BB				0297	CREM EQU #	THIS ROUTINE
01BB	CD	A1	01	0298	CALL VDADD	REMOVES THE
01BE	3A	10	0D	0299	LDA SVCHR	CURSOR FROM
01C1	77			0300	MOV M,A	THE SCREEN
01C2	C9			0301	RET .	
01C3				0302	PBACK EQU #	BACK UP ONE
01C3	CD	90	01	0303	CALL PLEFT	POSITION
01C6	CD	A1	01	0304	CALL VDADD	ON
01C9	3A	9A	00	0305	LDA OMASK	THE SCREEN
01CC	F6	20		0306	ORI 020H	
01CE	77			0307	MOV M,A	
01CF	C9			0308	RET .	
01D0				0309	PCR EQU #	
01D0	CD	73	01	0310	CALL CLINE	
01D3	C3	94	01	0311	JMP PCUR	
01D6				0312	PLF EQU #	
01D6	3A	12	0D	0313	LDA LINE	
01D7	3C			0314	INR A	
01DA	E6	0F		0315	ANI 00FH	
01DC	C2	49	01	0316	JNZ CUR	
01DF	C3	1B	01	0317	JMP SROL	
01E2				0318	PESC EQU #	
01E2	3E	FF		0319	MVI A,0FFH	SET ESCAPE SEQUENCE
01E4	32	15	0D	0320	STA ESCFL	IN FLAG AREA
01E7	C9			0321	RET .	
01E8				0322	ESCS EQU #	PROCESS THE
01E8	CD	BB	01	0323	CALL CREM	ESCAPE SEQUENCE
01EB	CD	F1	01	0324	CALL ESCSP	
01EE	C3	CD	00	0325	JMP GOBACK	
01F1				0326	ESCSP EQU #	
01F1	3A	15	0D	0327	LDA ESCFL	
01F4	FE	FF		0328	CPI 0FFH	
01F6	CA	19	02	0329	JZ SECOND	
01F9	21	15	0D	0330	LXI H,ESCFL	
01FE	FE	02		0331	MVI M,000H	
0200	DA	11	02	0332	CPI 002H	
0203	CA	15	02	0333	JC SETX	
0206	FE	08		0334	JZ SETY	
0208	CA	F9	05	0335	CPI 008H	
020B	FE	09		0336	JZ STSPD	
020D	DA	FF	00	0337	CPI 009H	
0210	C0			0338	JC OCHAR	
				0339	RNZ	

0211  
 0211 78  
 0212 C3 94 01  
 0215  
 0215 78  
 0216 C3 47 01  
 0219 78  
 021A FE 03  
 021C CA 2F 02  
 021F FE 04  
 0221 C2 2B 02  
 0224 44  
 0225 4D  
 0226  
 0226 E1  
 0227 D1  
 0228 C5  
 0229 E5  
 022A AF  
 022B  
 022B 32 15 0D  
 022E C9  
 022F  
 022F 21 11 0D  
 0232 46  
 0233 23  
 0234 4E  
 0235 C3 26 02  
 0238  
 0238 AF  
 0239 4F  
 023A 21 08 0D  
 023D  
 023D 77  
 023E 23  
 023F 0C  
 0240 C2 3D 02  
 0243 01 85 0F  
 0246 CD 4D 01  
 0249 3E 12  
 024B 32 0B 0F  
 024E CD 63 0B  
 0251  
 0251 CD 6A 00  
 0254 AF  
 0255 D3 C8  
 0257 D3 FE  
 0259 32 0F 0D  
 025C 32 0E 0D

0340 SETX EQU \$  
 0341 MOV A,B  
 0342 JMP PCUR  
 0343 SETY EQU \$  
 0344 MOV A,B  
 0345 JMP CURSC  
 0346 SECOND EQU \$  
 0347 MOV A,B  
 0348 CPI 003H  
 0349 JZ CURET  
 0350 CPI 004H  
 0351 JNZ ARET2  
 0352 MOV B,H  
 0353 MOV C,L  
 0354 ARET1 EQU \$  
 0355 POP H  
 0356 POP D  
 0357 PUSH B  
 0358 PUSH H  
 0359 XRA A  
 0360 ARET2 EQU \$  
 0361 STA ESCFL  
 0362 RET .  
 0363 CURET EQU \$  
 0364 LXI H,NCHAR  
 0365 MOV B,M  
 0366 INX H  
 0367 MOV C,M  
 0368 JMP ARET1  
 0369 STRTA EQU \$  
 0370 XRA A  
 0371 MOV C,A  
 0372 LXI H,SYSTRAM  
 0373 CLERA EQU \$  
 0374 MOV M,A  
 0375 INX H  
 0376 INR C  
 0377 JNZ CLERA  
 0378 LXI SP,SYSTP  
 0379 CALL PERSE  
 0380 MVI A,12H  
 0381 STA FCBR  
 0382 CALL ERRS  
 0383 COMN1 EQU \$  
 0384 CALL EXINT  
 0385 XRA A  
 0386 OUT 0C8H  
 0387 OUT 0FEH  
 0388 STA OPORT  
 0389 STA IPORT

CLEAR SYSTEM  
 SCRATCH PAD  
 AREA

*Zero the accumulator*  
*" " C Reg*  
*Load R & L with starting address of system*

*Initial...*

*zero to default*

*Perch -> clean screen, puts cursor at home, zero cursor line and set up counter*

25F			0390	COMND EQU #	
025F	01	85	0F	U	0391 LXI SP,SYSTP SET STACK POINTER
0262	3A	0F	0D		0392 LDA OPORT GET CURR OP PORT
0265	C5	YS		U	0393 PUSH PSW SAVE
0266	3E	45			0394 MVI A,'E' FORCE
0268	32	ED	0C		0395 STA RDVE READ OPERATIONS
026B	AF				0396 XRA A
026C	32	0F	0D		0397 STA OPORT RESET OUTPUT PORT
026F	32	00	00		0398 STA START RESET INPUT PORT
0272	CD	A4	03		0399 CALL PROMPT PROMPT CHAR TO SCREEN
0275	CD	82	02		0400 CALL GCLIN GET INPUT FROM USER
0278	C1			U	0401 POP PSW RECOVER OUTPUT PORT
0279	32	0F	0D		0402 STA OPORT AND RESAVE
027C	CD	B1	02		0403 CALL COPRC PROCESS COMMAND
027F	C3	5F	02		0404 JMP COMND AND LOOP
0282					0405 GCLIN EQU #
0282	CD	1F	00		0406 CALL SINP GET A CHAR
0285	CA	82	02		0407 JZ GCLIN LOOP TILL WE GET ONE
0288	E6	7F			0408 ANI 7FH GET RID OF BIT SEVEN
028A	FE	61			0409 CPI 61H FORCE UPPER CASE ASCII
028C	DA	96	02		0410 JC NOTLO
028F	FE	7B			0411 CPI 7BH
0291	D2	96	02		0412 JNC NOTLO
0294	E6	5F			0413 ANI 5FH
0296	E6	7F			0414 NOTLO ANI 07FH
0298	CA	51	02		0415 JZ COMN1
029B	47				0416 MOV B,A
029C	FE	0D			0417 CPI 00DH CR = END
029E	CA	73	01		0418 JZ CLINE
02A1	FE	0A			0419 CPI 00AH IGNORE LINE FEED
02A3	C8				0420 RZ
02A4	FE	7F			0421 CPI 07FH DELETE MEANS DEL A CHAR
02A6	C2	AB	02		0422 JNZ CONT
02A7	06	5F			0423 MVI B,05FH DO IT WITH A BACK SPACE
02AB					0424 CONT EQU #
02AB	CD	19	00		0425 CALL SOUT OUTPUT THE CHAR
02AE	C3	82	02		0426 JMP GCLIN AND GO GET NEXT CHAR
02B1					0427 COPRC EQU #
02B1	CD	BB	01		0428 CALL CREM GET RID OF CURSOR
02B4	0E	01			0429 MVI C,001H
02B6	CD	A5	01		0430 CALL VDADZ FIND COMMAND LINE
02B9	EB				0431 XCHG .
02BA	21	00	00		0432 LXI H,START WHERE ARE WE AT
02BD	E5				0433 PUSH H
02BE	CD	E3	03		0434 CALL SCHR
02C1	CA	00	00		0435 JZ ERR1
02C4	EB				0436 XCHG .
02C5	11	15	03		0437 LXI D,COMTAB POINT AT OUR COMMAND TABLE
02C8	3A	9A	00		0438 LDA OMASK GET ANY NEEDED MASK
02CB	47				0439 MOV B,A
02CC	CD	F7	02		0440 CALL FDCOM SCAN THE COMMAND TABLE
02CF	C2	E4	02		0441 JNZ DISP1 JMP IF FIND
02D2	CD	F4	02		0442 CALL FDCOU ELSE CHECK USERS COMMAND TABLE
02D5	CA	81	05		0443 JZ ERR2 NO HIT MEANS INVALID COMMAND
02D8	13				0444 INX D
02D9	EB				0445 XCHG
02DA	7E				0446 MOV A,M
02DB	23				0447 INX H
02DC	66				0448 MOV H,M
02DD	6F				0449 MOV L,A
02DE	E3				0450 XTHL
02DF	7D				0451 MOV A,L
02E0	C9				0452 RET

*Handwritten notes:*  
 GCLIN  
 GET INPUT FROM USER

*Handwritten note:*  
 ↓ process command line



02E1  
02E1 CA 01 05  
02E4 13  
02E5 EB  
02E6  
02E6 7E  
02E7 23  
02E7 66  
02E7 6F  
02EA D5  
02EB EB  
02EC 2A 93 00  
02EF 19  
02F0 D1  
02F1 E3  
02F2 7D  
02F3 C9  
02F4  
02F4 11 EE 0C  
02F7  
02F7 1A  
02F8 B7  
02F9 C8  
02FA B0  
02FB E5  
02FC BE  
02FD 13  
02FE C2 0B 03  
0301 23  
0302 1A  
0303 B0  
0304 BE  
0305 C2 0B 03  
0308 E1  
0309 B7  
030A C9  
030B  
030B 13  
030C 13  
030D 13  
030E E1  
030F C3 F7 02  
0312 49  
0313 7D 00

0453 DISP0 EQU \$  
0454 JZ ERR2  
0455 DISP1 INX D  
0456 XCHG .  
0457 DISPT EQU \$ DISPATCH ROUTINES  
0458  
0459  
0460 MOV H,M  
0461 MOV L,A  
0462 PUSH D  
0463 XCHG  
0464 LHLD WHERE  
0465 DAD D  
0466 POP D  
0467 XTHL .  
0468 MOV A,L  
0469 RET .  
0470 FDCOU EQU \$  
0471 LXI D,CUTAB POINT AT CUSTOM COMMAND TABLE  
0472 FDCOM EQU \$  
0473 LDAX D SEARCH TABLE ROUTINE  
0474 ORA A  
0475 RZ  
0476 ORA B  
0477 PUSH H  
0478 CMP M  
0479 INX D  
0480 JNZ NCOM  
0481 INX H  
0482 LDAX D  
0483 ORA B  
0484 CMP M  
0485 JNZ NCOM  
0486 POP H  
0487 ORA A  
0488 RET .  
0489 NCOM EQU \$  
0490 INX D  
0491 INX D  
0492 INX D  
0493 POP H  
0494 JMP FDCOM  
0495 DB 49H  
0496 DW TABEND-COMTAB

0315  
0315 43 45  
0317 3B 0A  
0319 44 5E  
031B 20 04  
031D 45 4E  
031F 21 05  
0321 45 58  
0323 5C 05  
0325 4C 4F  
0327 DE 09  
0329 53 54  
032B A5 0A  
032D 52 55  
032F DD 09  
0331 43 54  
0333 D5 0A  
0335 53 45  
0337 E5 05  
0339 43 55  
033B 16 06  
033D 42 55  
033F 04 0B  
0341 56 45  
0343 D5 09  
0345 00  
0346  
0346 0B  
0347 4D 01  
0349 17  
034A 89 01  
034C 1A  
034D 43 01  
034F 01  
0350 90 01  
0352 13  
0353 9A 01  
0355 0E  
0356 66 01  
0358 0D  
0359 D0 01  
035B 0A  
035C D6 01  
035E 5F  
035F C3 01  
0360 13  
0362 E2 01  
0364 00

0497 COMTAB EQU \$ COMMAND TABLE

0498 ASC 'CE'  
0499 DW CERT1-START  
0500 ASC 'DU'  
0501 DW DUMP-START  
0502 ASC 'EN'  
0503 DW ENTER-START  
0504 ASC 'EX'  
0505 DW EXEC-START  
0506 ASC 'LO'  
0507 DW LOAD1-START  
0508 ASC 'ST'  
0509 DW STOR1-START  
0510 ASC 'RU'  
0511 DW LOADX-START  
0512 ASC 'CT'  
0513 DW CTLG1-START  
0514 ASC 'SE'  
0515 DW SET-START  
0516 ASC 'CU'  
0517 DW CUSET-START  
0518 ASC 'BU'  
0519 DW BILD1-START  
0520 ASC 'VE'  
0521 DW VERIF-START  
0522 NOP .

0523 TBL EQU \$ VIDIO COMMAND TABLE

0524 DB 0BH ✓  
0525 DW PERSE-START  
0526 DB 17H ✓  
0527 DW PUP-START  
0528 DB 1AH ✓  
0529 DW PDOWN-START  
0530 DB 01H ✓  
0531 DW PLEFT-START  
0532 DB 13H ✓  
0533 DW PRIT-START  
0534 DB 0EH ✓  
0535 DW PHOME-START  
0536 DB 0DH ✓  
0537 DW PCR-START  
0538 DB 0AH ✓  
0539 DW PLF-START  
0540 DB 5FH ✓  
0541 DW PBACK-START  
0542 DB 1BH ✓  
0543 DW PESC-START  
0544 DB 00H

ERASE SCREEN & cursor Home

move cursor up

" " down

" " left

" " right

" " Home

Carriage Return

Line Feed

Backspace

ESC.

0365  
 0365 53 3D  
 0367 FA 05  
 0369 49 3D  
 036B FE 05  
 036D 4F 3D  
 0370 02 06  
 0371 44 3D  
 0373 12 06  
 0375 43 49  
 0377 06 06  
 0379 43 4F  
 037B 0A 06  
 037D 54 59  
 037F 0E 06  
 0381 00  
 0382  
 0382 B6 00  
 0384 4C 00  
 0386 2E 00  
 0388 99 03  
 038A  
 038A 3A 00  
 038C 41 00  
 038E 25 00  
 0390 92 03  
 0392  
 0392  
 0392 E5  
 0393 2A 08 0D  
 0396 C3 9D 03  
 0399  
 0399 E5  
 039A 2A 0A 0D  
 039D  
 039D 7D  
 039E B4  
 039F CA 51 02  
 03A2 E3  
 03A3 C9  
 03A4  
 03A4 CD AC 03  
 03A7 06 2A  
 03A9 C3 19 00  
 03AC  
 03AC 06 0A  
 03AE CD 19 00  
 03B1 06 0D  
 03B3 CD 19 00  
 03B6 3A 18 0D  
 03B9 4F  
 03BA  
 03BA 0D  
 03BB F3  
 03BC AF  
 03BD CD 1D 05  
 03C0 C3 BA 03

0545 SETAB EQU \$ SET COMMAND TABLE  
 0546 ASC 'S='  
 0547 DW DISPD-START  
 0548 ASC 'I='  
 0549 DW SETIN-START  
 0550 ASC 'O='  
 0551 DW SETOT-START  
 0552 ASC 'D='  
 0553 DW SETNU-START  
 0554 ASC 'CI'  
 0555 DW SETCI-START  
 0556 ASC 'CO'  
 0557 DW SETCO-START  
 0558 ASC 'TY'  
 0559 DW SETTY-START  
 0560 DB 00H  
 0561 OUTBL EQU \$ OUTPUT DEVICE ADDRESS OFFSETS  
 0562 DW VMDR-START  
 0563 DW SEROUT-START  
 0564 DW PPLOUT-START  
 0565 DW ERROT-START  
 0566 INTBL EQU \$ INPUT DEVICE ADDRESS OFFSETS  
 0567 DW KBDIN-START  
 0568 DW SERIN-START  
 0569 DW PPLIN-START  
 0570 DW ERRIT-START  
 0571 TABEND EQU \$  
 0572 ERRIT EQU \$  
 0573 PUSH H  
 0574 LHLD SYSRAM  
 0575 JMP ERROR1  
 0576 ERROT EQU \$  
 0577 PUSH H  
 0578 LHLD UOPRT  
 0579 ERROR1 EQU \$  
 0580 MOV A,L  
 0581 ORA H  
 0582 JZ COMN1  
 0583 XTHL .  
 0584 RET .  
 0585 PROMPT EQU \$ PROMPT CHAR PROCESSING  
 0586 CALL CRLF  
 0587 MVI B,'\*'  
 0588 JMP SOUT  
 0589 CRLF EQU \$ CARRAGE RETURN LINE FEED PROCESSING  
 0590 MVI B,00AH  
 0591 CALL SOUT  
 0592 MVI B,00DH  
 0593 CALL SOUT  
 0594 LDA NUCNT  
 0595 MOV C,A  
 0596 NULOT EQU \$  
 0597 DCR C  
 0598 RM  
 0599 XRA A  
 0600 CALL OUTH  
 0601 JMP NULOT

03C3		0602	PSCAN EQU \$	SCAN FOR PARAMETERS
03C3	CD CE 03	0603	CALL SBLK	
03C6	3E 01	0604	MVI A,001H	
03C8	C8	0605	RZ	
03C9	CD F7 03	0606	CALL SHEX	
03CC	7D	0607	MOV A,L	
03CD	C9	0608	RET .	
03CE	0E 0C	0609	SBLK EQU \$	SCAN FOR BLANKS
03D0		0610	MVI C,00CH	
03D0	1A	0611	SBLK1 EQU \$	
03D1	E6 7F	0612	LDAX D	
03D3	FE 20	0613	ANI 7FH	
03D5	CA E3 03	0614	CPI 020H	
03D8	13	0615	JZ SCHR	
03D9	FE 3D	0616	INX D	
03DB	CA E3 03	0617	CPI 03DH	
03DE	0D	0618	JZ SCHR	
03DF	C2 D0 03	0619	DCR C	
03E2	C9	0620	JNZ SBLK1	
03E3		0621	RET .	
03E3	0E 0A	0622	SCHR EQU \$	SCAN FOR NON BLANKS
03E5		0623	MVI C,00AH	
03E5	1A	0624	SCHR1 EQU \$	
03E5	E6 7F	0625	LDAX D	
03E6	E6 7F	0626	ANI 7FH	
03E8	FE 20	0627	CPI 020H	
03EA	C0	0628	RNZ	
03EB	13	0629	INX D	
03EC	0D	0630	DCR C	
03ED	C8	0631	RZ	
03EE	C3 E5 03	0632	JMP SCHR1	
03F1		0633	SCONV EQU \$	
03F1	CD CE 03	0634	CALL SBLK	
03F4	CA 80 05	0635	JZ ERR1	
03F7		0636	SHEX EQU \$	
03F7	26 00	0637	MVI H,0	
03F9	6C	0638	MOV L,H	
03FA		0639	SHE1 EQU \$	PROCESS HEX INPUT
03FA	1A	0640	LDAX D	
03FB	E6 7F	0641	ANI 7FH	
03FD	FE 20	0642	CPI 020H	
03FF	C8	0643	RZ	
0400	FE 2D	0644	CPI '-'	DASH IS OK FOR RANGE INPUT
0402	C8	0645	RZ	
0403	FE 2F	0646	CPI '/'	SLASH TERMINATES DATA INPUT
0406	29	0647	RZ	
0406	29	0648	DAD H	
0407	29	0649	DAD H	
0408	29	0650	DAD H	
0409	29	0651	DAD H	
040A	CD 16 04	0652	CALL HCOV1	
040D	D2 80 05	0653	JNC ERR1	
0410	85	0654	ADD L	
0411	6F	0655	MOV L,A	
0412	13	0656	INX D	
0413	C3 FA 03	0657	JMP SHE1	
0416		0658	HCOV1 EQU \$	CONVERT INPUT TO HEX
0416	D6 30	0659	SUI 030H	
0418	FE 0A	0660	CPI 00AH	
041A	D8	0661	RC	
041B	D6 07	0662	SUI 007H	
041D	FE 10	0663	CPI 010H	
041F	C9	0664	RET .	

*mispatch  
to  
down  
to*

0420			0665	DUMP EQU \$	FORMATTING MEMORY DUMP ROUTINES
0420	CD	F1 03	0666	CALL SCONV	
0423	E5		0667	PUSH H	
0424	1A		0668	LDAX D	
0425	E6	7F	0669	ANI 7FH	
0427	26	00	0670	MVI H,0	
0429	6C		0671	MOV L,H	
042A	FE	2B	0672	CPI '-'	
042C	C2	3F 04	0673	JNZ D01	
042F	2B		0674	DCX H	
0430	CD	C3 03	0675	CALL PSCAN	
0433	FE	01	0676	CPI 1	
0435	CA	42 04	0677	JZ D02	
0438	D1		0678	POP D	
0439	D5		0679	PUSH D	
043A	19		0680	DAD D	
043B	2B		0681	DCX H	
043C	C3	42 04	0682	JMP D02	
043F	CD	C3 03	0683	D01 CALL PSCAN	
0442	D1		0684	D02 POP D	
0443	EB		0685	XCHG .	
0444			0686	DLOOP EQU \$	
0444	CD	AC 03	0687	CALL CRLF	
0447	CD	BA 04	0688	CALL ABOUT	
044A	CD	EE 04	0689	CALL O2BL	
044D	06	04	0690	MVI B,4	
044F	0E	10	0691	MVI C,010H	
0451	E5		0692	PUSH H	
0452	D5		0693	PUSH D	
0453			0694	DLP1 EQU \$	
0453	7E		0695	MOV A,M	
0454	C5		0696	PUSH B	
0455	CD	BF 04	0697	CALL HBOUT	
0458	7D		0698	MOV A,L	
0459	93		0699	SUB E	
045A	7C		0700	MOV A,H	
045B	9A		0701	SBB D	
045C	C1		0702	POP B	
045D	D2	71 04	0703	JNC ASCII	
0460	23		0704	INX H	
0461	05		0705	DCR B	
0462	C2	6A 04	0706	JNZ CONU	
0465	CD	F1 04	0707	CALL BOUT	
0468	06	04	0708	MVI B,4	
046A			0709	CONU EQU \$	
046A	0D		0710	DCR C	
046B	C2	53 04	0711	JNZ DLP1	
046E	C3	74 04	0712	JMP A1	
0471	CD	F6 04	0713	ASCII CALL POS	
0474	D1		0714	A1 POP D	
0475	E1		0715	POP H	
0476	C5		0716	PUSH B	
0477	CD	EE 04	0717	CALL O2BL	
047A	06	2A	0718	MVI B,'*'	
047C	CD	19 00	0719	CALL SOUT	
047F	C1		0720	POP B	

0480 0E 10  
0482 7E  
0483 05  
0484 FE 20  
0486 D2 13 04  
0487 3E 2E  
0488 FE 5F  
0489 C2 92 04  
0490 3E 2E  
0492 FE 7F  
0494 DA 99 04  
0497 3E 2E  
0499 47  
049A CD 19 00  
049B 7D  
049E 93  
049F 7C  
04A0 9A  
04A1 DA AC 04  
04A4 06 2A  
04A6 CD 19 00  
04A9 03 5F 02  
04AC D1  
04AD 23  
04AE 0D  
04AF C2 02 04  
04B2 06 2A  
04B4 CD 19 00  
04B7 C3 44 04  
04BA  
04BA 7C  
04BB CD 09 05  
04BE 7D  
04BF CD 09 05  
04C2 CD 1F 00  
04C5 C8  
04C6 E6 7F  
04C8 FE 00  
04CA CA 5F 02  
04CD CD E2 04  
04D0 FE 20  
04D2 C0  
04D3  
04D3 CD 1F 00  
04D6 CA D3 04  
04D9 CD E2 04  
04DC FE 00  
04DE C0  
04DF C3 5F 02  
04E2 FE 30  
04E4 D8  
04E5 FE 3A  
04E7 D0  
04E8 E6 0F  
04EA 32 14 0D  
04ED C9  
04EE CD F1 04  
04F1  
04F1 06 20  
04F3 C3 19 00

0721 DLOP MVI C,16  
0722 DLP2 MOV A,M  
0723 PUSH B  
0724 CPI 22H  
0725 JNC DOWN  
0726 MVI A,'.'  
0727 DOWN CPI 5FH  
0728 JNZ DN0  
0729 MVI A,'.'  
0730 DN0 CPI 7FH  
0731 JC DN1  
0732 MVI A,'.'  
0733 DN1 MOV B,A  
0734 CALL SOUT  
0735 MOV A,L  
0736 SUB E  
0737 MOV A,H  
0738 SBB D  
0739 JC MORE  
0740 MVI B,'\*'  
0741 CALL SOUT  
0742 JMP COMND  
0743 MORE POP B  
0744 INX H  
0745 DCR C  
0746 JNZ DLP2  
0747 MVI B,'\*'  
0748 CALL SOUT  
0749 JMP DLOOP  
0750 ABOUT EQU \$  
0751 MOV A,H  
0752 CALL HEOUT  
0753 MOV A,L }  
0754 HBOUT EQU \$  
0755 CALL HEOUT  
0756 CALL SINP  
0757 RZ  
0758 ANI 7FH  
0759 CPI 0  
0760 JZ COMND  
0761 CALL SESPD  
0762 CPI 020H  
0763 RNZ  
0764 WTLP1 EQU \$  
0765 CALL SINP  
0766 JZ WTLP1  
0767 CALL SESPD  
0768 CPI 0  
0769 RNZ  
0770 JMP COMND  
0771 SESPD CPI '0'  
0772 RC  
0773 CPI '9'+1  
0774 RNC  
0775 ANI 0FH  
0776 STA SPEED  
0777 RET  
0778 02BL CALL BOUT PRINT 2 BLANKS  
0779 BOUT EQU \$ PRINT 1 BLANK  
0780 MVI B,020H  
0781 JMP SOUT

04F1	06 20	0780
04F3	C3 19 A0	0781
04F6	05	0782
04F7	C2 FF A4	0783
04FA	CD F1 A4	0784
04FD	06 04	0785
04FF	0D	0786
0500	C8	0787
0501	C5	0788
0502	CD EE A4	0789
0505	C1	0790
0506	C3 F6 A4	0791
0509	4F	0792
050A	0F	0793
050B	0F	0794
050C	0F	0795
050D	0F	0796
050E	CD 12 A5	0797
050F	79	0798
0510	E6 0F	0799
0514	C6 30	0800
0516	FE 3A	0801
0518	DA 1D A5	0802
051B	C6 07	0803

Bout	MVI B, 020H
	JMP Sout
(D)	DCR B
	JNZ (A)
	CALL (B) Bout
(A)	MVI B, 04H
	DCR C
	RZ
	PUSH B
	CALL (C) 02BL
	POP B
	JMP (D)
	MOV C, A
	RRC
	RRC
	RRC
	RRC
	CALL (E)
	MOV A, C
(E)	ANI 0FH
	ADI 030H
	CPI 03AH
	JC (F)
	ADI 007H

0

051	D	47
051	E	C3 19 A4
	H	CD <del>0200B</del> <sup>F1 A3</sup>
052	4	E5
052	5	AF
052	6	32 0F AD
052	9	CD AC A3
052	C	06 3A
052	E	CD AB A2
05	<del>31</del>	CD BB A1
05	34	0E 01
053	6	CD A5 A1
053	9	EB
053	A	0E 03
053	C	CD E5 A3
053	F	CA 29 A5
054	<del>23</del>	FE 2F
44	<del>00</del>	CA 51 A2
47	<del>00</del>	CD F7 A3
4A	<del>00</del>	FE 3A
4C	<del>00</del>	CA 57 A5
4	<del>00</del>	7D
50	<del>00</del>	E1
51	<del>00</del>	77

0804
0805
0806
0807
0809
0810
0811
0812
0813
0814
0815
0816
0817
0818
0819
0820
0821
0822
0823
0824
0825
0826
0827
0828

(F)

MOV	B, A
JMP	Sout
CALL	SCONV
PUSH	H
XRA	A
STA	<del>0</del> OPORT
CALL	<del>0</del> CRLF
MVI	<sup>B</sup> 03AH
CALL	<del>0</del> CONT
CALL	<del>0</del> CREM
MVI	C, 0FH
CALL	<del>0</del> <del>0</del> <del>0</del> <del>0</del> <sup>VDADR</sup>
XCHG	
MVI	C, 03H
CALL	<del>0</del> SCHR1
JZ	(M)
<del>0</del>	CPI 02FH
JZ	<del>0</del> Comn1
CALL	<del>0</del> SNEY
CPI	03AH
JZ	<del>0</del> ENL03
MOV	A, L
POP	H
MOV	M, A

(M)



0552 ~~4D~~

23

0829

INX H

{ 53 ~~4D~~

ES

0830

PUSH H

{ 54 ~~4D~~

C3 3A A5

0831

JMP (Q)

↓ 57

0832

0833

0834

0835

0836

0837

0557  
0557 E3  
0558 13  
0559 C3 3A 05  
055C  
055C CD F1 03  
055F E5  
0560 2A 93 00  
0563 C9  
0564  
0564 21 1A 0D  
0567  
0567 CD CE 03  
056A 06 06  
056C  
056C 1A  
056D E6 7F  
056F FE 20  
0571 CA 8A 05  
0574 FE 2F  
0576 CA 8A 05  
0579 77  
057A 13  
057B 23  
057C 05  
057D C2 6C 05  
0580  
0580 EB  
0581  
0581 3A 9A 00  
0584 F6 3F  
0586 77  
0587 C3 5F 02  
058A  
058A 36 00  
058C 23  
058D 05  
058E C2 8A 05  
0591 FE 2F  
0593 3E 01  
0595 C2 9E 05  
0598 13  
0599 CD E3 03  
059C D6 30  
059E  
059E E6 01  
05A0 3E 80  
05A2 C2 A6 05  
05A5 1F  
05A6  
05A6 32 3A 0D  
05A9 C9  
05AA  
05AA CD B5 05  
05AD C3 51 02  
05B0  
05B0 3A A5 0F  
05B3 B7  
05B4 C9

0837 ENLO3 EQU \$  
0838 XTHL .  
0839 INX D  
0840 JMP ENLO1  
0841 EXEC EQU \$  
0842 CALL SCONV  
0843 EXEC1 EQU \$ PROCESS EXECUTE COMMANDS  
0844 PUSH H  
0845 LHLD WHERE  
0846 RET .  
0847 NAMES EQU \$ PROCESS NAMES WHEN REQUIRED BY C#  
0848 LXI H,THEAD  
0849 NAME EQU \$  
0850 CALL SBLK  
0851 MVI B,006H  
0852 NAME1 EQU \$  
0853 LDAX D  
0854 ANI 7FH  
0855 CPI 020H  
0856 JZ NFIL  
0857 CPI 02FH  
0858 JZ NFIL  
0859 MOV M,A  
0860 INX D  
0861 INX H  
0862 DCR B  
0863 JNZ NAME1  
0864 ERR1 EQU \$  
0865 XCHG .  
0866 ERR2 EQU \$ A ? MARK TO SCREEN IF WE DONT UNDE#  
0867 LDA OMASK  
0868 ORI '?'  
0869 MOV M,A  
0870 JMP COMND  
0871 NFIL EQU \$  
0872 MVI M,000H  
0873 INX H  
0874 DCR B  
0875 JNZ NFIL  
0876 CPI 02FH  
0877 MVI A,001H  
0878 JNZ DEFLT  
0879 INX D  
0880 CALL SCHR  
0881 SUI 030H  
0882 DEFLT EQU \$  
0883 ANI 001H  
0884 MVI A,080H  
0885 JNZ STUNT  
0886 RAR .  
0887 STUNT EQU \$  
0888 STA SNUMF  
0889 RET .  
0890 PER1 EQU \$  
0891 CALL NAOUT  
0892 JMP COMN1  
0893 ALOAD EQU \$  
0894 LDA FNUMF  
0895 ORA A  
0896 RET .

05B5  
05B5 16 08  
05B7 21 19 0D  
05BA CD 05 05  
05BD CD F1 04  
05C0 2A 23 0D  
05C3 CD BA 04  
05C6 CD F1 04  
05C9 2A 21 0D  
05CC CD BA 04  
05CF CD F1 04  
05D2 C3 AC 03  
05D5  
05D5 7E  
05D6 B7  
05D7 C2 DC 05  
05DA 3E 20  
05DC  
05DC CD 1D 05  
05DF 23  
05E0 15  
05E1 C2 D5 05  
05E4 C9  
05E5  
05E5 CD CE 03  
05E8 CA 80 05  
05EB D5  
05EC CD F1 03  
05EF E3  
05F0 11 65 03  
05F3 CD F7 02  
05F6 C3 E1 02  
05F9  
05F9 78  
05FA  
05FA 32 14 0D  
05FD C9  
05FE  
05FE 32 0E 0D  
0601 C9  
0602  
0602 32 0F 0D  
0605 C9  
0606  
0606 22 08 0D  
0609 C9  
060A  
060A 22 0A 0D  
060D C9  
060E  
060E 32 20 0D  
0611 C9  
0612  
0612 32 18 0D  
0615 C9

0897 NAOUT EQU \$ OUTPUT NAMES WHEN NEEDED  
0898 MVI D,008H  
0899 LXI H,THEAD-1  
0900 CALL NLOOP  
0901 CALL BOUT  
0902 LHLD LOADR  
0903 CALL ADOUT  
0904 CALL BOUT  
0905 LHLD BLOCK  
0906 CALL ADOUT  
0907 CALL BOUT  
0908 JMP CRLF  
0909 NLOOP EQU \$  
0910 MOV A,M  
0911 ORA A  
0912 JNZ CHRLI  
0913 MVI A,020H  
0914 CHRLI EQU \$  
0915 CALL OUTH  
0916 INX H  
0917 DCR D  
0918 JNZ NLOOP  
0919 RET .  
0920 SET EQU \$ PROCESS SET COMMANDS  
0921 CALL SBLK  
0922 JZ ERR1  
0923 PUSH D  
0924 CALL SCONV  
0925 XTHL .  
0926 LXI D,SETAB  
0927 CALL FDCOM  
0928 JMP DISP0  
0929 STSPD EQU \$  
0930 MOV A,B  
0931 DISPD EQU \$ SET DISPLAY SPEED  
0932 STA SPEED  
0933 RET .  
0934 SETIN EQU \$ SET INPUT PORT  
0935 STA IPORT  
0936 RET .  
0937 SETOT EQU \$ SET OUTPUT PORT  
0938 STA OPORT  
0939 RET .  
0940 SETCI EQU \$ SET CUSTOM INPUT ADDRESS  
0941 SHLD SYSRAM  
0942 RET .  
0943 SETCO EQU \$ SET CUSTOM OUTPUT ADDRESS  
0944 SHLD UOPRT  
0945 RET .  
0946 SETTY EQU \$ SET FILE TYPE  
0947 STA HTYPE  
0948 RET .  
0949 SETNU EQU \$ SET PRINT CRLF DELAY  
0950 STA NUCNT  
0951 RET .

0616  
0616 CD 64 05  
0619 21 5F 02  
061C CD C3 03  
061F E5  
0620 21 1A 0D  
0621 7E  
0624 FE 3F  
0626 CA 42 06  
0629 CD F4 02  
062C CA 32 06  
062F 1B  
0630 36 00  
0632  
0632 7E  
0633 12  
0634 13  
0635 23  
0636 7E  
0637 12  
0638 13  
0639 E1  
063A EB  
063B 73  
063C 23  
063D 72  
063E 23  
063F 36 00  
0641 C9  
0642 CD AC 03  
0645 21 EE 0C  
0648 7E  
0649 B7  
064A CA AC 03  
064D 16 02  
064F CD D5 05  
0652 CD F1 04  
0655 5E  
0656 23  
0657 56  
0658 23  
0659 EB  
065A CD BA 04  
065D CD EE 04  
0660 EB  
0661 C3 48 06  
0664  
0664 E1  
0665  
0665 D1  
0666  
0666 AF  
0667 37  
0668 C9

0952 CUSET EQU \$ ADD AND DELETE CUSTOM COMMANDS  
0953 CALL NAMES  
0954 LXI H,COMND  
0955 CALL PSCAN  
0956 PUSH H  
0957 LXI H,THEAD  
0958 MOV A,M  
0959 CPI '?'  
0960 JZ LISTIT  
0961 CALL FDCOU  
0962 JZ CUSE2  
0963 DCX D  
0964 MVI M,000H  
0965 CUSE2 EQU \$  
0966 MOV A,M  
0967 STAX D  
0968 INX D  
0969 INX H  
0970 MOV A,M  
0971 STAX D  
0972 INX D  
0973 POP H  
0974 XCHG .  
0975 MOV M,E  
0976 INX H  
0977 MOV M,D  
0978 INX H  
0979 MVI M,0  
0980 RET .  
0981 LISTIT CALL CRLF DISPLAY CUSTOM COMMANDS  
0982 LXI H,CUTAB  
0983 LOOPIT MOV A,M  
0984 ORA A  
0985 JZ CRLF  
0986 MVI D,2  
0987 CALL NLOOP  
0988 CALL BOUT  
0989 MOV E,M  
0990 INX H  
0991 MOV D,M  
0992 INX H  
0993 XCHG  
0994 CALL ADOUT  
0995 CALL O2BL  
0996 XCHG  
0997 JMP LOOPIT  
0998 TERE2 EQU \$  
0999 POP H  
1000 TERE1 EQU \$  
1001 POP D  
1002 TERE0 EQU \$  
1003 XRA A  
1004 STC .  
1005 RET .

*5F02*

*02 05 00 00 00 00*  
*00 00 00 00*

0669	CD	84	06	1004	PHEAD EQU \$
066C	C5			1007	CALL PLOAD
066D	26	00		1008	PUSH B
066F	2E	06		1009	MVI H,0
0671	09			1010	MVI L,6
0672	06	01		1011	DAD B
0674	0E	00		1012	MVI B,1
0676	CD	7B	06	1013	MVI C,0
0679	E1			1014	CALL PSTOR
067A	C9			1015	POP H
067B				1016	RET .
067B	23			1017	PSTOR EQU \$
067C	71			1018	INX H
067D	23			1019	MOV M,C
067E	70			1020	INX H
067F	23			1021	MOV M,B
0680	73			1022	INX H
0681	23			1023	MOV M,E
0682	72			1024	INX H
0683	C9			1025	MOV M,D
0684				1026	RET .
0684	23			1027	PLOAD EQU \$
0685	4E			1028	INX H
0686	23			1029	MOV C,M
0687	46			1030	INX H
0688	23			1031	MOV B,M
0689	5E			1032	INX H
068A	23			1033	MOV E,M
068B	56			1034	INX H
068C	C9			1035	MOV D,M
068D				1036	RET .
068D	06	05		1037	DHCOMP EQU \$
				1038	MVI B,005H
068F	1A			1039	DHLOP EQU \$
0690	BE			1040	LDAX D
0691	C0			1041	CMP M
0692	05			1042	RNZ
0693	C8			1043	DCR B
0694	23			1044	RZ
0695	13			1045	INX H
0696	C3	8F	06	1046	INX D
				1047	JMP DHLOP



06A3		0016	*****
06A3		0017	*
06A3		0018	* BLOCK WRITE PROCESSING
06A3		0019	*
06A3		0020	*****
06A3		0021	WTAPE EQU \$
06A6	CD 99 06	0022	CALL UNIT GO PROCESS UNIT
06A6	CD 7E 08	0023	CALL INIT INIT FCB
06A9	CD 9A 08	0024	CALL WHDR GO WRITE HEADER
06AC	D4 4A 09	0025	CNC WDATA NOW THE DATA
06AF	D8	0026	RC RET IF WRITE ERROR
06B0	E5	0027	PUSH H
06B1	2A 90 0F	0028	LHLD FCBBG GET START ADDRESS
06B4	22 92 0F	0029	SHLD FCBBG USE AS END ADDRESS
06B7	21 8E 0F	0030	LXI H,FCBID POINT AT FCB
06B8		0031	INR M INCR BLOCK ID
06BB	21 8B 0F	0032	LXI H,FCB POINT AT FCB
06BE	CD 91 09	0033	CALL SAVE WRITE EOF
06C1	E1	0034	POP H
06C2	C9	0035	RET
06C3		0036	*****
06C3		0037	*
06C3		0038	* READ BLOCK PROCESSING
06C3		0039	*
06C3		0040	*****
06C3		0041	RTAPE EQU \$
06C3	CD 99 06	0042	CALL UNIT PROCESS UNIT
06C6	C5	0043	PUSH B SAVE
06C7	D5	0044	PUSH D MOST
06C8	E5	0045	PUSH H REGS
06C9	CD 7E 08	0046	CALL INIT INIT FCB
06CC	CD DD 06	0047	CALL RHEAD READ THE HEADER
06CD	D4 20 07	0048	CNC RDATA IF NO ERR READ DATA
06D2	16 00	0049	MVI D,0
06D4	1E 00	0050	MVI E,0
06D6	D4 20 07	0051	CNC RDATA AND EOF
06D9	E1	0052	POP H RECOVER
06DA	D1	0053	POP D SAVED
06DB	C1	0054	POP B REGS
06DC	C9	0055	RET
06DD		0056	*****
06DD		0057	*
06DD		0058	* READ HEADERS AND CHECK AGAINST REQUEST
06DD		0059	*
06DD		0060	*****
06DD		0061	RHEAD EQU \$
06DD	D5	0062	PUSH D SAVE DE
06DE		0063	RTAP1 EQU \$
06DE	E5	0064	PUSH H AND HL
06DF	CD 08 07	0065	CALL RDHDR GO READ FIRST HEADER FOUND
06E2	DA 64 06	0066	JC TEREZ
06E5	E1	0067	POP H RECOVER PTR TO REQUESTED HDR
06E6	E5	0068	PUSH H AND RESAVE
06E7	11 1A 0D	0069	LXI D,THEAD POINT AT HDR JUST READ
06EA	CD 8D 06	0070	CALL DHCMP GO SEE IF IT MATCHES REQ
06ED	E1	0071	POP H RECOVER HL AGAIN
06EE	C2 DE 06	0072	JNZ RTAP1 NO MATCH READ NEXT HDR
06F1	C5	0073	PUSH PSW SAVE FLAGS
06F2	CD 1F 00	0074	CALL SINP CHECK FOR USR INTERVENTION
06F5	CA 05 07	0075	JZ RTAP2 JUMP IF SO

06F8 E6 7F  
06FA FE 00  
06FC C2 05 07  
06FF 32 8B 0F  
0702 C3 64 06  
0705 C1  
0706  
0706 D1  
0707 C9  
0708  
0708  
0708  
0708  
0708  
0708 21 1A 0D  
070B 22 90 0F  
070E 26 48  
0710 2E 00  
0712 22 8E 0F  
0715 3E 45  
0717 32 8C 0F  
071A 21 8B 0F  
071D C3 72 09  
0720  
0720  
0720  
0720  
0720  
0720  
0720 E5  
0721 7A  
0722 B3  
0723 2A 21 0D  
0724 E8  
0727 C2 2D 07  
072A 2A 23 0D  
072D  
072D D5  
072E 22 90 0F  
0731 26 44  
0733 2E 00  
0735 22 8E 0F  
0738 3A ED 0C  
073B 32 8C 0F  
073E 21 8B 0F  
0741 CD 72 09  
0744 DA 64 06  
0747 D1  
0748 E1  
0749 AF  
074A C9

U

0076 ANI 7FH  
0077 CPI 0  
0078 JNZ RTAP2  
0079 STA FCBC  
0080 JMP TERE2  
0081 RTAP2 POP PSW RECOVER FLAGS  
0082 TPER EQU \$  
0083 POP D RECOVER DE  
0084 RET  
0085 \*\*\*\*\*  
0086 \*  
0087 \* BUILD FILE CONTROL BLOCK  
0088 \*  
0089 \*\*\*\*\*  
0090 RDHDR EQU \$  
0091 LXI H,THEAD WHERE HEADER GOES  
0092 SHLD FCBBG PUT IN FCB  
0093 MVI H,'H' HEADER TYPE RECORD  
0094 MVI L,0 BLOCK ID  
0095 SHLD FCBBG PUT IN FCB  
0096 MVI A,'E' READ TYPE OPERATION  
0097 STA FCBC PUT IN FCB  
0098 LXI H,FCB POINT AT FCB  
0099 JMP GET GO GET HDR  
0100 \*\*\*\*\*  
0101 \*  
0102 \* READ DATA BLOCKS  
0103 \*  
0104 \*\*\*\*\*  
0105 RDATA EQU \$  
0106 PUSH H SAVE HL  
0107 MOV A,D SEE IF USER WANTS  
0108 ORA E LOAD OFFSET  
0109 LHLD BLOCK GET BLOCK SIZE FROM HEADER  
0110 XCHG INTO DE  
0111 JNZ NOFF JUMP IF LOAD OFFSET REQUESTED  
0112 LHLD LOADR ELSE USE LOAD ADDR IN HDR  
0113 NOFF EQU \$  
0114 PUSH D SAVE BLOCK SIZE  
0115 SHLD FCBBG PUT LOAD ADDRESS IN FCB  
0116 MVI H,'D' SPECIFY DATA TYPE BLOCK  
0117 MVI L,0 READ ANY DATA BLOCK  
0118 SHLD FCBBG PUT IN FCB  
0119 LDA RDVE SPECIFY A READ TYPE OPERATION  
0120 STA FCBC PUT IN FCB  
0121 LXI H,FCB POINT AT FCB  
0122 CALL GET GO READ THE BLOCK  
0123 JC TERE2 JMP IF READ ERROR  
0124 POP D ELSE RECOVER BLOCK SIZE  
0125 POP H RECOVER HL  
0126 XRA A CLEAR THE FLAGS  
0127 RET



074B		0128	*****
074B		0129	*
074B		0130	* OPEN A FILE ROUTINE
074B		0131	*
074C		0132	*****
074B		0133	BOPEN EQU \$
074B	E5	0134	PUSH H SAVE HL
074C	CD 68 07	0135	CALL LFCB GO CHECK STATUS OF FILE
074F	C2 65 06	0136	JNZ TERE1 JUMP IF ALL IS NOT WELL
0752	32 87 0F	0137	STA HFLG RESET HDR WRITTEN FLAG
0755	36 01	0138	MVI M,1 INDICATE FILE OPEN
0757	23	0139	INX H NEXT PART OF FILE CTL BLOCK
0758	77	0140	MOV M,A ZERO THAT
0759	23	0141	INX H
075A	77	0142	MOV M,A AND THAT
075B	11 49 0D	0143	LXI D,FBUF1 POINT AT FIRST BUFFER
075E	3A A5 0F	0144	LDA FNUMF GET WHICH FILE
0761	82	0145	ADD D USE AS INDEX TO CORR BUFFER
0762	57	0146	MOV D,A SAVE WHICH FILE
0763		0147	UBUF EQU \$
0763	C1	0148	POP B RECOVER PTR TO HDR
0764	B7	0149	ORA A SEE IF FILE 1 OR 2
0765	C3 7B 06	0150	JMP PSTOR SAVE STUFF IN FILE CTL BLOCK
0768		0151	*****
0768		0152	*
0768		0153	* CHECK STATUS OF FILE CTL BLOCK
0768		0154	*
0768		0155	*****
0768		0156	LFCB EQU \$
0768	21 3B 0D	0157	LXI H,FCBA1 POINT AT FIRST CTL BLOCK
076B	1F	0158	RAR
076C	E6 01	0159	ANI 1
0771	CA A5 0F	0160	STA FNUMF SAVE FILE IN WORK AREA
0771	CA 77 07	0161	JZ LFCB1 JUMP IF FILE 1
0774	21 42 0D	0162	LXI H,FCBA2 ELSE POINT AT 2ND CTL BLOCK
0777		0163	LFCB1 EQU \$
0777	7E	0164	MOV A,M GET OPEN CLOSE BYTE
0778	B7	0165	ORA A USE TO SET FLAGS
0779	37	0166	STC SET CARRY IN CASE OF ERROR
077A	C9	0167	RET
077B		0168	*****
077B		0169	*
077B		0170	* FILE CLOSE ROUTINE
077B		0171	*
077B		0172	*****
077B		0173	PCLOS EQU \$
077B	CD 68 07	0174	CALL LFCB GO CHECK STATUS OF CTL BLK
077E	C8	0175	RZ RET IF NOT OPEN CARRY SET
077F	B7	0176	ORA A CLEAR CARRY
0780	3C	0177	INR A SEE IF READ
0781	36 00	0178	MVI M,0 CLOSE THE FILE
0783	C8	0179	RZ RET IF READ
0784	21	0180	INX H INDEX INTO
0785	23	0181	INX H CTL BLOCK
0786	7E	0182	MOV A,M GET INDEX INTO BUFFER

0707 CD 84 06  
0708A C5  
0708B 21 07 00  
0708E 09  
0708F B7  
0709 CA B6 07  
0709C 3E  
0709D 77  
0709E 23  
0709F 36 00  
070A 23  
070A 72  
070B 60  
070B 69  
070C 3A 87 0F  
070D B7  
070E C2 AF 07  
070F 3C  
0710 32 87 0F  
0711 CD 9A 08  
0712 DA 65 06  
0713 CD 4A 09  
0714 DA 65 06  
0715 E1  
0716 AF  
0717 77  
0718 23  
0719 77  
071A E1  
071B 3A 87 0F  
071C B7  
071D C2 C9 07  
071E AF  
071F 32 87 0F  
0720 CD 9A 08  
0721 D4 4A 09  
0722 C9  
0723  
0724  
0725  
0726  
0727  
0728 CD 68 07  
0729 C8  
072A 3C  
072B FA 66 06  
072C 36 FF  
072D 23  
072E 7E  
072F E5  
0730 23  
0731 CD 84 06  
0732 E1

0183 CALL PLOAD PLUS STUFF FROM CTL BLK  
0184 PUSH B  
0185 LXI H, BLKOF OFFSET TO BLOCK SIZE  
0186 DAD B  
0187 ORA A TEST COUNT  
0188 JZ EOFW BUFF EMPTY JUST WRITE EOF  
0189 PUSH H SAVE BLOCK SIZE  
0190 MOV M, A BYTE COUNT  
0191 INX H  
0192 MVI M, 0  
0193 INX H  
0194 MOV M, E  
0195 INX H  
0196 MOV M, D  
0197 MOV H, B  
0198 MOV L, C  
0199 LDA HFLG GET HDR WRITTEN FLAG  
0200 ORA A SEE IF WE'VE WRITTEN HDR  
0201 JHFIN JMP IF WE HAVE  
0202 INR A  
0203 STA HFLG WE HAVE NOW  
0204 CALL WHDR GO MAKE PREVIOUS STATEMENT CORRECT  
0205 JC TERE1 JMP IF ERR WRITING HDR  
0206 HFIN CALL WDATA GO WRITE LAST DATA BLK  
0207 JC TERE1 JMP IF ERR WRITING DATA  
0208 POP H RECOVER BLK SIZE PTR  
0209 EOFW XRA A  
0210 MOV M, A SET BLK SIZE OF ZERO  
0211 INX H IN BOTH BYTES  
0212 MOV M, A OF BLK SIZE  
0213 POP H  
0214 LDA HFLG SEE IF HDR IS WRITTEN  
0215 ORA A  
0216 JNZ HFIN2 JMP IF SO  
0217 XRA A  
0218 STA HFLG RESET HDR WRITTEN FLAG  
0219 CALL WHDR AND GO WR HDR  
0220 HFIN2 CNC WDATA AND EOF  
0221 RET  
0222 \*\*\*\*\*  
0223 \*  
0224 \* BYTE READ PROCESSING  
0225 \*  
0226 \*\*\*\*\*  
0227 RTBYT EQU \$  
0228 CALL LFCB CHECK STATUS OF CTL BLK  
0229 RZ RET W/CARRY IF NOT OPEN  
0230 INR A CHECK IF WRITE  
0231 JM TERE0 JMP IF SO  
0232 MVI M, -1 SET READ IN CASE FIRST READ  
0233 INX H  
0234 MOV A, M GET READ COUNT  
0235 PUSH H  
0236 INX H  
0237 CALL PLOAD GO GET STUF FROM CTL BLK  
0238 POP H

07DF B7  
 07E0 C2 24 08  
 07E3  
 07E3 D5  
 07E4 E5  
 07E5 2C  
 07E6 CD 69 06  
 07E9 3A 87 0F  
 07EC B7  
 07ED C2 FA 07  
 07F0 3C  
 07F1 32 87 0F  
 07F4 CD DD 06  
 07F7 DA 64 06  
 07FA  
 07FA CD 20 07  
 07FD DA 64 06  
 0800 21 92 0F  
 0803 3A 90 0F  
 0806 BE  
 0807 C2 0F 08  
 080A 23  
 080B 3A 91 0F  
 080E BE  
 080F E1  
 0810 CA 79 08  
 0811 22  
 0812 CA 98 0F  
 0817 EB  
 0818 2A 92 0F  
 081B 7D  
 081C 93  
 081D E1  
 081E 77  
 081F 23  
 0820 36 00  
 0822 2B  
 0823 D1  
 0824  
 0824 3D  
 0825 77  
 0826 23  
 0827 7E  
 0828 34  
 0829 83  
 082A 5F  
 082B D2 2F 08  
 082E 14  
 082F  
 082F 1A  
 0830 B7  
 0831 C9

0239 ORA A SEE IF ANY THING IN BUFFER  
 0240 JNZ GTBYT IF SO GO GET A BYTE  
 0241 RDNBK EQU \$  
 0242 PUSH D SAVE DE  
 0243 PUSH H AND HL  
 0244 INX H  
 0245 CALL PHEAD GO PREPARE HDR FOR A READ  
 0246 LDA HFLG SEE IF WEVE READ HDR  
 0247 ORA A  
 0248 JNZ ISRD JMP IF SO  
 0249 INR A  
 0250 STA HFLG IF NO SET WE ARE NOW  
 0251 CALL RHEAD AND GO DO IT  
 0252 JC TERE2 JMP ON READ ERR  
 0253 ISRD EQU \$  
 0254 CALL RDATA GO READ A DATA BLOCK  
 0255 JC TERE2 JMP ON READ ERR  
 0256 LXI H,FCBEN GET END OF DATA PTR  
 0257 LDA FCBBG AND BYTE 1 OF BEG OF DATA  
 0258 CMP M SEE IF THEY AGREE  
 0259 JNZ NOTEF IF NOT ITS NOT EOF  
 0260 INX H POINT AT BYTE 2 OF FCBEN  
 0261 LDA FCBBG+1 GET BYTE 2 OF FCBBG  
 0262 CMP M CHECK BYTE 2  
 0263 NOTEF POP H  
 0264 JZ EOFER JMP IF EOF  
 0265 RDSR E  
 0266 LHLD FCBBG GET BEG OF DATA  
 0267 XCHG IN DE  
 0268 LHLD FCBEN GET END OF DATA IN HL  
 0269 MOV A,L AND  
 0270 SUB E CALC  
 0271 POP H  
 0272 MOV M,A THE  
 0273 INX H BLK SIZE  
 0274 MVI M,0  
 0275 DCX H  
 0276 POP D  
 0277 GTBYT EQU \$  
 0278 DCR A DCR BYTE CNT  
 0279 MOV M,A UPD BYTE CNT  
 0280 INX H  
 0281 MOV A,M GET BUFFER PTR  
 0282 INR M INR PTR  
 0283 ADD E  
 0284 MOV E,A  
 0285 JNC RT1  
 0286 INR D  
 0287 RT1 EQU \$  
 0288 LDAX D GET THE DATA BYTE  
 0289 ORA A RESET CARRY  
 0290 RET

0832		0291	*****
0832		0292	*
0832		0293	* BUILD FCB ID SUBROUTINE
0832		0294	*
0832		0295	*****
0832		0296	BLID EQU \$
0832	26 44	0297	MVI H,'D' INDICATE DATA TYPE BLK
0834	11 89 0F	0298	LXI D,FIN1 POINT AT ID CTL BLK
0837	3A A5 0F	0299	LDA FNUMF GET UNIT CTL BLK
083A	B7	0300	ORA A
083B	CA 3F 08	0301	JZ DAT2 JMP IF UNIT ZERO
083E	13	0302	INX D POINT AT ID 2 CTL BLK
083F	1A	0303	DAT2 LDAX D GET ID
0840	6F	0304	MOV L,A PUT IN L
0841	3C	0305	INR A UPD TO NEXT ID
0842	12	0306	STAX D SAVE IN CTL BLK
0843	22 8E 0F	0307	SHLD FCBID PUT IN FCB
0846	C9	0308	RET
0847		0309	*****
0847		0310	*
0847		0311	* BYTE WRITE SUBROUTINE
0847		0312	*
0847		0313	*****
0847		0314	WTBYT EQU \$
0847	CD 68 07	0315	CALL LFCB GO CHK CTL BLK
084A	C8	0316	RZ RET IF NOT OPEN
084B	3C	0317	INR A CHK IF READ
084C	C8	0318	RZ RET IF SO
084D	34 FE	0319	MVI M,0FEH SET TO WR IN CASE FIRST WR
084F	23	0320	INX H ON INTO CTL BLK
0850	23	0321	INX H TWO BYTES
0851	78	0322	MOV A,B GET THE BYTE IN A
0852	C5	0323	PUSH PSW SAVE IT
0853	E5	0324	PUSH H AND PTR INTO CTL BLK
0854	CD 84 06	0325	CALL PLOAD GO GET VARIOUS PTRS
0857	E1	0326	POP H RECOVER CTL BLK PTR
0858	7E	0327	MOV A,M GET BYTE CNT
0859	83	0328	ADD E
085A	5F	0329	MOV E,A
085B	D2 5F 08	0330	JNC WT1
085E	14	0331	INR D
085F		0332	WT1 EQU \$
085F	C1	0333	POP PSW GET THE BYTE TO BE SAVED
0860	12	0334	STAX D PUT IN BUFFER
0861	B7	0335	ORA A RESET FLAGS
0862	34	0336	INR M INR BYTE CNT
0863	C0	0337	RNZ RET IF BUFF NOT FULL
0864	CD 69 06	0338	CALL PHEAD BUILD HDR
0867	3A 87 0F	0339	LDA HFLG SEE IF WEVE WRITTEN HDR
086A	B7	0340	ORA A
086B	C2 76 08	0341	JNZ HWTN JMP IF WE HAVE
086E	3C	0342	INR A ELSE INDICATE WE ARE
086F	32 87 0F	0343	STA HFLG
0872	CB 9A 08	0344	CALL WHDR AND GO DO IT
0875	D8	0345	RC BAIL OUT IF ERROR
0876		0346	HWTN EQU \$
0876	C3 4A 09	0347	JMP WDATA GO WRITE THE BUFFER

0879  
0879  
0879  
0879  
0879  
0879 AF  
087A 3D  
087B 37  
087C D1  
087D C9  
087E  
087E  
087E  
087E  
087E  
087E E5  
087F 26 FF  
0881 6C  
0882 22 90 0F  
0885 22 94 0F  
0888 22 96 0F  
088B 23  
088C 22 8B 0F  
088F 22 8D 0F  
0892 22 92 0F  
0895 22 8E 0F  
0898 E1  
0899 C9  
089A  
089A  
089A  
089A  
089A D5  
089B E5  
089C 16 00  
089E 1E 0F  
08A0 22 90 0F  
08A3 19  
08A4 21 00 0E  
08A7 26 48  
08A9 2E 00  
08AB E5  
08AC 5D  
08AD 21 89 0F  
08B0 3A A5 0F  
08B3 B7  
08B4 CA B8 0B  
08B7 23  
08B8 1C  
08B9 73  
08BA E1  
08BB 22 8E 0F  
08BE E1  
08BF E5

0348 \*\*\*\*\*  
0349 \*  
0350 \* END OF FILE EXIT  
0351 \*  
0352 \*\*\*\*\*  
0353 EOFER EQU \$  
0354 XRA A CLEAR FLAGS  
0355 DCR A SET MINUS  
0356 STC AND CARRY  
0357 POP D RECOVER DE  
0358 RET  
0359 \*\*\*\*\*  
0360 \*  
0361 \* FCB INITIALIZATION SUBROUTINE  
0362 \*  
0363 \*\*\*\*\*  
0364 INIT PUSH H  
0365 MVI H,0FFH SET H AND  
0366 MOV L,H L TO FOXES  
0367 SHLD FCBBG PUT IN BG  
0368 SHLD FCBAS AS  
0369 SHLD FCBER AND ER  
0370 INX H HL NOW ZEROS  
0371 SHLD FCBC PUT IN RC  
0372 SHLD FCBF  
0373 SHLD FCBE EN  
0374 SHLD FCBEID AND ID  
0375 POP H  
0376 RET  
0377 \*\*\*\*\*  
0378 \*  
0379 \* SET UP FCB TO WRITE HDR FILE  
0380 \*  
0381 \*\*\*\*\*  
0382 WHDR EQU \$  
0383 PUSH D SAVE DE  
0384 PUSH H AND HL  
0385 MVI D,0 SET DE TO  
0386 MVI E,0FH LEN OF HDR  
0387 SHLD FCBBG PUT ADDR OF HDR IN FCB  
0388 DAD D CALC END ADDR OF HDR  
0389 SHLD FCBE PUT THAT IN FCB  
0390 MVI H,'H' SPECIFY HDR TYPE RCD  
0391 MVI L,0 AND HDR ID OF 0  
0392 PUSH H SAVE IT  
0393 MOV E,L COPY OF ID  
0394 LXI H,FIN1 POINT AT ID CTL BLK  
0395 LDA FNUMF GET UNIT  
0396 ORA A IS IT UNIT 1  
0397 JZ ITWO JUMP IF IT IS  
0398 INX H POINT AT UNIT 2 CTL BLK  
0399 ITWO INR E SET NEXT ID  
0400 MOV M,E SAVE ID IN CTL BLK  
0401 POP H RECOVER HDR TY & ID  
0402 SHLD FCBEID PUT IN FCB  
0403 POP H RECOVER PTR TO HDR  
0404 PUSH H AND RESAVE

```

0800 1E 04
0802 19
0803 7E
0804 B7
0805 CA 14 09
0808 FE 31
080A DA 14 09
080D FE 3A
080F D2 14 09
08D2 E6 0F
08D4
08D4 06 01
08D6 3D
08D7 CA 16 09
08DA C5
08DB C5
08DC AF
08DD 32 9A 0F
08E0 32 98 0F
08E3
08E3 CD 23 09
08E6
08E6 DA 09 09
08E9 CA 09 09
08EC 2A 9F 0F
08EF 3A 9D 0F
08F2 BD
08F3 C2 E3 08
08F6 3A 9E 0F
08F9 BC
08FA C2 E3 08
08FD
08FD C1
08FE C1
08FF B8
0900 CA 14 09
0903 04
0904 C5
0905 C5
0906 C3 E3 08
0909 C1
090A C1
090B 3E 06
090D 32 8B 0F
0910 37
0911 C3 20 09
0914 3E FF
0916 3C
0917 32 8D 0F
091A 21 8B 0F
091D CD 91 09
0920 E1
0921 D1
0922 C9

```

```

0405 MVI E,4 POINT AT 5TH CHAR OF NAME
0406 DAD D INR HL TO POINT THERE
0407 MOV A,M GET 5TH CHAR
0408 ORA A SEE IF ZERO
0409 JZ NOIT2 JMP IF IT IS
0410 CPI '1' SEE IF ITS BETWEEN 1
0411 JC NOIT2 JMP IF LESS THAN 1
0412 CPI '9'+1 AND 9
0413 JNC NOIT2 JMP IF GREATER THAN 9
0414 ANI 0FH REMOVE ASCII BIAS
0415 UNOIT EQU $
0416 MVI B,1 SET 1 FOR FIRST FILE
0417 DCR A REDUCE REQ FILE POS BY 1
0418 JZ NOIT3 JMP IF REQ WAS POS 1
0419 PUSH B SAVE OUR 1
U 0420 PUSH PSW SAVE USERS FILE POS REG-1
0421 XRA A
0422 STA CBFP ZERO CB FILE POS
0423 STA CBRC AND CB RET CODE
0424 FIND EQU $
0425 CALL CAT1 USE CAT UTIL TO STEP
0426 * THRU THE RECORDS
0427 JC CATER JMP IF CAT HAD AN ERR
0428 JZ CATER JMP IF EOT ENCOUNTERED
0429 LHLD CBEN CHK FOR EOF
0430 LDA CBBG RECORDS TO
0431 CMP L STEP THROUGH
0432 JNZ FIND EACH FILE
0433 LDA CBBG+1 UNTIL
0434 CMP H REQ FILE POS -1
0435 JNZ FIND IS FOUND
0436 FIND2 EQU $
U 0437 POP PSW RECOVER USRS FILE POS-1
0438 POP B AND CURR FILE NR
0439 CMP B SEE IF WERE THERE
0440 JZ NOIT2 JMP WHEN THEY AGREE
0441 INR B ELSE BUMP POS BY 1
0442 PUSH B SAVE POS
U 0443 PUSH PSW USERS REQUEST
0444 JMP FIND AND TRY AGAIN
0445 CATER POP B CLEAR
0446 POP B THE STACK
0447 MVI A,6
0448 STA FCBRC
0449 STC
0450 JMP OUT AND GET OUT
0451 NOIT2 MVI A,-1
0452 NOIT3 INR A
0453 STA FCBFP SET UP FILE POSITION
0454 LXI H,FCB POINT AT FCB
0455 CALL SAVE AND GO SAVE HEADER
0456 OUT POP H RECOVER HL
0457 POP D AND DE
0458 RET

```

0923		0459	*****
0923		0460	*
0923		0461	* THIS IS THE CALL TO THE CAT FUNCTION
0923		0462	* WE DO SOME TRICKY STACK SWAPPING SO
0923		0463	* HANG ON.
0923		0464	*
0923		0465	*****
0923		0466	CAT1 EQU \$
0923	22 85 0F	0467	SHLD HLSAV TEMP SAVE OF HL
0926	26 00	0468	MVI H,0 ZERO HL
0928	2E 00	0469	MVI L,0
092A	07	0470	DAD SP AND RECOVER USR STK PTR
092B	0F E2 0F	0471	LXI SP,STAK2 NOW SET OUR OWN STACK
092E	C5	0472	PUSH B SAVE BC
092F	D5	0473	PUSH D DE
0930	E5	0474	PUSH H AND USER SP
0931	Z1 42 09	0475	LXI H,CTBK GET OUR RET POINT
0934	E5	0476	PUSH H PUT ON STACK
0935	Z1 98 0F	0477	LXI H,CB POINT AT CB
0938	3A A5 0F	0478	LDA FNUMF GET UNIT
093B	B7	0479	ORA A CHECK IF UNIT 1
093C	CA 64 00	0480	JZ ESFC1 JMP IF UNIT 1
093F	C3 8C 00	0481	JMP ESFC2 ELSE ITS UNIT 2
0942	E1	0482	CTBK POP H CAT FUNCTION RETURNS HERE
0943	D1	0483	POP D
0944	C1	0484	POP B
0945	F9	0485	SPHL RESTORE USER SP
0946	2A 85 0F	0486	LHLD HLSAV RECOVER USR HL
0949	C9	0487	RET
094A		0488	*****
094A		0489	*
094A		0490	* WRITE DATA SUBROUTINE
094A		0491	*
094A		0492	*****
094A		0493	WDATA EQU \$
094A	E5	0494	PUSH H SAVE HL
094B	D5	0495	PUSH D DE
094C	C5	0496	PUSH B BC
094D	16 00	0497	MVI D,0
094F	1E 07	0498	MVI E,7 OFFSET TO BLK SIZE
0951	19	0499	DAD D MOVE HL TO THERE
0952	4E	0500	MOV C,M GET BLK SIZE
0953	23	0501	INX H
0954	46	0502	MOV B,M IN BC
0955	23	0503	INX H
0956	5E	0504	MOV E,M GET LOADR
0957	23	0505	INX H
0958	56	0506	MOV D,M IN DE
0959	EB	0507	XCHG HL HAS LOADR
095A	22 90 0F	0508	SHLD FCBBG PUT IN FCB
095D	09	0509	DAD B CALC END ADDR
095E	22 92 0F	0510	SHLD FCBBG PUT IN FCB
0961	CB 32 08	0511	CALL BLID GO BUILD ID
0964	AF	0512	XRA A
0965	32 8D 0F	0513	STA FCBBG SET FILE POS OF 0
0968	Z1 8B 0F	0514	LXI H,FCB POINT AT FCB
096B	CD 91 09	0515	CALL SAVE GO WRITE DATA BLK
096E	C1	0516	POP B RECOVER BC
0970	F D1	0517	POP D DE
0970	E1	0518	POP H HL
0971	C9	0519	RET

0972			0520 *****		
0972			0521 *		
0972			0522 * THIS IS THE CALL TO THE GET FUNCTION		
0972			0523 * WE DO SOME TRICKY STACK SWAPPING SO		
0972			0524 * HANG ON.		
0972			0525 *		
0972			0526 *****		
0972	22	85	0F		0527 GET EQU \$
0975	26	00			0528 SHLD HLSAV
0977	2E	00			0529 MVI H,0
0979	09			U	0530 MVI L,0
097A	01	E2	0F	U	0531 DAD SP
097D	C5				0532 LXI SP,STAK2
097E	D5				0533 PUSH B
097F	E5				0534 PUSH D
0980	21	42	09		0535 PUSH H
0983	E5				0536 LXI H,CTBK
0984	2A	85	0F		0537 PUSH H
0987	3A	A5	0F		0538 LHLD HLSAV
098A	B7				0539 LDA FNUMF
098B	CA	5E	00		0540 ORA A
098E	C3	86	00		0541 JZ ESFG1
0991					0542 JMP ESFG2
0991					0543 *****
0991					0544 *
0991					0545 * THIS IS THE CALL TO THE SAVE FUNCTION
0991					0546 * WE DO SOME TRICKY STACK SWAPPING SO
0991					0547 * HANG ON.
0991					0548 *
0991					0549 *****
0991					0550 SAVE EQU \$
0991	22	85	0F		0551 SHLD HLSAV
0994	26	00			0552 MVI H,0
0996	2E	00			0553 MVI L,0
0998	09			U	0554 DAD SP
0999	01	E2	0F	U	0555 LXI SP,STAK2
099C	C5				0556 PUSH B
099D	D5				0557 PUSH D
099E	E5				0558 PUSH H
099F	21	42	09		0559 LXI H,CTBK
09A2	E5				0560 PUSH H
09A3	2A	85	0F		0561 LHLD HLSAV
09A6	3A	A5	0F		0562 LDA FNUMF
09A9	B7				0563 ORA A
09AA	CA	61	00		0564 JZ ESFS1
09AD	C3	89	00		0565 JMP ESFS2
09B0					0566 *****
09B0					0567 *
09B0					0568 * THIS IS THE CALL TO THE CERTIFY ROUTINE
09B0					0569 * WE DO SOME TRICKY STACK SWAPPING SO HANG ON
09B0					0570 *
09B0					0571 *****
09B0					0572 ESFCE EQU \$
09B0	22	85	0F		0573 SHLD HLSAV
09B3	26	00			0574 MVI H,0
09B5	2E	00			0575 MVI L,0
09B7	09			U	0576 DAD SP
09B8	01	E2	0F	U	0577 LXI SP,STAK2



098B	C5		0578	PUSH B
098C	D5		0579	PUSH D
098D	E5		0580	PUSH H
098E	21	CF 09	0581	LXI H,CEBK
09C1	E5		0582	PUSH H
09C2	2A	85 0F	0583	LHLD HLSAV
09C3	3A	A5 0F	0584	LDA FNUMF
09C8	B7		0585	ORA A
09C9	CA	67 00	0586	JZ ESCE1
09CC	C3	8F 00	0587	JMP ESCE2
09CF	22	85 0F	0588	CEBK SHLD HLSAV
09D2	C3	42 09	0589	JMP CTBK
09D5			0590	*****
09D5			0591	*
09D5			0592	* THIS IS THE LOAD AND RUN COMMAND PROCESSING
09D5			0593	* ALSO VERIFY
09D5			0594	*
09D5			0595	*****
09D5	3E	56	0596	VERIF MVI A,'V' SPECIFY VERIFY
09D7	32	ED 0C	0597	STA RDVE
09DA	C3	DE 09	0598	JMP LOAD1
09DD	3E		0599	LOADX DB 3EH ENTER HERE TO AUTO EXEC
09DE	AF		0600	LOAD1 XRA A ENTER HERE FOR LOAD ONLY
09DF	C5		0601	PUSH PSW
09E0	21	2A 0D	0602	LXI H,DHEAD POINT AT ALT HEADER
09E3	CD	67 05	0603	CALL NAME GET A NAME IF SUPPLIED
09E6	CD	99 06	0604	CALL UNIT
09E9	26	00	0605	MVI H,0
09EB	2E	00	0606	MVI L,0
09ED	CD	C3 03	0607	CALL PSCAN CHECK FOR OPTIONAL LOAD ADDR
09F0	EB		0608	TLOA2 XCHG
09F1	21	2A 0D	0609	LXI H,DHEAD POINT AT ALT HDR AREA
09F4	7E		0610	MOV A,M GET FIRST BYTE OF NAME
09F5	B7		0611	ORA A AND SEE IF ONE SUPPLIED
09F6	C2	FC 09	0612	JNZ TLOA2 JMP IF NO NAME DESIRED
09F9	21	2A 0D	0613	LXI H,THEAD POINT AT HDR INPUT AREA
09FC	E5		0614	TLOA3 PUSH H SAVE PTR
09FD	3A	A5 0F	0615	LDA FNUMF GET UNIT
0A00	E1		0616	POP H RECOVER PTR TO HDR
0A01	CD	13 00	0617	CALL RDBLK GO READ THE BLOCK
0A04	DA	57 0B	0618	JC ERRS0 EXIT IF ANY READ ERRORS
0A07	CD	B5 05	0619	CALL NADUT OUTPUT HDR JUST READ
0A0A	C1		0620	POP PSW RECOVER FLAGS
0A0B	B7		0621	ORA A TEST THEM
0A0C	C8		0622	RZ RET IF JUST LOAD
0A0D	3A	20 0D	0623	LDA HTYPE CHECK IF EXECUTABLE FILE
0A10	B7		0624	ORA A
0A11	FA	2D 0A	0625	JM PERX JUMP IF NOT
0A14	3A	1F 0D	0626	LDA THEAD+5 THIS MUST BE 0 TO AUTO EX
0A17	B7		0627	ORA A
0A18	C2	2D 0A	0628	JNZ PERX BAIL OUT IF IT ISNT
0A1B	2A	25 0D	0629	LHLD XEQAD GET AUTO EXEC ADDR FM HDR
0A1E	7C		0630	MOV A,H
0A1F	FE	FF	0631	CPI 0FFH
0A21	C2	2A 0A	0632	JNZ EX5
0A24	7D		0633	MOV A,L
0A25	FE	FF	0634	CPI 0FFH
0A27	CA	2D 0A	0635	JZ PERX
0A2A	C3	5F 05	0636	EX5 JMP EXEC1 AND GO DO IT
0A2D	3E	07	0637	PERX MVI A,7
0A2F	32	8B 0F	0638	STA FCBRC
0A32	C3	63 0B	0639	JMP ERRS

0A35  
0A35  
0A35  
0A35  
0A35  
0A35 CD 64 05  
0A38 C3 99 06  
0A3B CD 35 0A  
0A3E CD AC 03  
0A41 Z1 87 0C  
0A44 CD B6 0B  
0A47 CD AC 03  
0A4A Z6 FF  
0A4C ZE FF  
0A4E E5  
0A4F CD B0 09  
0A52 DA 66 0A  
0A55 C2 86 0A  
0A58 E3  
0A59 Z1 97 0C  
0A5C CD B6 0B  
0A5F E1  
0A60 CD BA 04  
0A63 C3 92 0A  
0A66  
0A66 FE 05  
0A68 CA 82 0A  
0A6B FE 10  
0A6D CA 51 0B  
0A70 FE 02  
0A72 CA 86 0A  
0A75 Z1 77 0C  
0A78 CD B6 0B  
0A7B CD AC 03  
0A7E E1  
0A7F C3 92 0A  
0A82 E1  
0A83 C3 8B 0B  
0A86 E5  
0A87 Z1 AC 0C  
0A8A CD B6 0B  
0A8D E1  
0A8E CD BA 04  
0A91 E1  
0A92 CD 1F 00  
0A95 CA 9D 0A  
0A98 E6 7F  
0A9A FE 00  
0A9C C8  
0A9D E5  
0A9E CD AC 03  
0AA1 E1  
0AA2 C3 4E 0A

0640 \*\*\*\*\*  
0641 \*  
0642 \* PROCESS CERTIFY COMMAND  
0643 \*  
0644 \*\*\*\*\*  
0645 SETUP CALL NAMES  
0646 JMP UNIT  
0647 CERT1 EQU \$  
0648 CALL SETUP GET UNIT  
0649 CALL CRLF A NEW LINE PLEASE  
0650 LXI H,SIMES SIZING TAPE MESSAGE  
0651 CALL MESS OUTPUT THE MESSAGE  
0652 CALL CRLF NOW A NEW LINE  
0653 MVI H,0FFH MAX CAPACITY  
0654 MVI L,0FFH  
0655 C2 PUSH H CAPACITY ON STACK  
0656 CALL ESFCE GO CERTIFY  
0657 JC C3 JMP IF ANY ERRORS  
0658 JNZ C4 JMP OF NOT EOT  
0659 XTHL UPDATED CAPACITY TO STACK  
0660 LXI H,SYSMC APPROX CAP MSG  
0661 CALL MESS PUT OUT THE LINE  
0662 POP H CAPACITY TO HL  
0663 CALL ADOUT PRINT IT  
0664 JMP C5  
0665 C3 EQU \$  
0666 CPI 5 WRITE PROTECTED  
0667 JZ C3A YES  
0668 CPI 10H  
0669 JZ ERRSX  
0670 CPI 2 DUMMY FILE FOUND  
0671 JZ C4 YES  
0672 LXI H,UTIM2 AN ERROR MESSAGE  
0673 CALL MESS PUT IT OUT  
0674 CALL CRLF A NEW LINE  
0675 POP H CAPACITY TO STACK  
0676 JMP C5  
0677 C3A POP H CLEAN UP STACK  
0678 JMP IT95 AND GET OUT WITH MSG  
0679 C4 PUSH H CAPACITY TO STACK  
0680 LXI H,SYSME  
0681 CALL MESS PUT IT OUT  
0682 POP H CAPACITY TO HL  
0683 CALL ADOUT CAPACITY OUT  
0684 POP H  
0685 C5 CALL SINP ANY CHAR WAITING  
0686 JZ C5A NOPE  
0687 ANI 7FH STRIP PARITY  
0688 CPI 00H  
0689 RZ TIME TO QUIT  
0690 C5A PUSH H SAVE CAPACITY  
0691 CALL CRLF A NEW LINE  
0692 POP H RECOVER CAPACITY  
0693 JMP C2

Get to be the  
Aform

0AA5				0694	*****
0AA5				0695	*
0AA5				0696	* PROCESS STORE COMMAND
0AA5				0697	*
0AA5				0698	*****
0AA5				0699	STOR1 EQU \$
0A05	CD	35	0A	0700	CALL SETUP GET NAME TO BE SAVED BY
0A08	CD	F1	03	0701	CALL SCONV GET BEGGING ADDR
0AAB	E5			0702	PUSH H SAVE IT
0AAC	CD	F1	03	0703	CALL SCONV GET END ADDR
0AAF	E3			0704	XTHL SWAP WITH TOP OF STACK
0AB0	E5			0705	PUSH H
0AB1	26	FF		0706	MVI H,0FFH
0AB3	6C			0707	MOV L,H
0AB4	CD	03	03	0708	CALL PSCAN
0AB7	22	25	0D	0709	SHLD XEQAD
0ABA	E1			0710	POP H
0ABB	22	23	0D	0711	SHLD LOADR BEGINNING TO HEADER
0ABE	31			0712	POP D RECOVER END
0ABF	7B			0713	MOV A,E AND
0AC0	95			0714	SUB L CALC
0AC1	6F			0715	MOV L,A THE
0AC2	7A			0716	MOV A,D BLOCK
0AC3	9C			0717	SBB H SIZE
0AC4	67			0718	MOV H,A
0AC5	22	21	0D	0719	SHLD BLOCK INTO HEADER
0AC8	21	1A	0D	0720	LXI H,THEAD POINT AT HDR
0ACB	3A	3A	0D	0721	LDA SNUMF
0ACE	CD	16	00	0722	CALL WRBLK AND GO WRITE THE RECORD
0AD1	DA	63	0B	0723	JC ERRS JMP ON ERROR
0AD4	C9			0724	RET
0AD5				0725	*****
0AD5				0726	*
0AD5				0727	* PROCESS CTLG COMMAND
0AD5				0728	*
0AD5				0729	*****
0AD5				0730	CTLG1 EQU \$
0AD5	CD	35	0A	0731	CALL SETUP GET UNIT
0AD8	CD	AC	03	0732	CALL CRLF AND A NEW LINE
0ADB	21	3C	0C	0733	LXI H,CTHDR GET HDR MSG
0ADE	CD	B6	0B	0734	CALL MESS OUTPUT THE MESSAGE
0AE1	CD	AC	03	0735	CALL CRLF AND A NEW LINE
0AE4	21	F2	0A	0736	LXI H,CALOP GET OUR RETURN
0AE7	E5			0737	PUSH H PUT ON STACK
0AE8	3A	A5	0F	0738	LDA FNUMF GET SOL UNIT
0AEB	B7			0739	ORA A SEE IF UNIT 1
0AEC				0740	* GO TO START OF TAPE WITH THIS JUMP
0AEC	CA	5B	00	0741	JZ EWTP1 JMP IF SO
0AEF	C3	83	00	0742	JMP EWTP2 ELSE ITS UNIT 2
0AF2				0743	CALOP EQU \$ WE'LL RET HERE
0AF2	21	1A	0D	0744	LXI H,THEAD POINT AT HDR GOES
0AF5	11	1A	0D	0745	LXI D,THEAD AND WHERE WE WANT IT READ
0AF8	CD	DD	06	0746	CALL RHEAD GO GET THE HEADER
0AFB	DA	51	0B	0747	JC ERRSX CARRY EXITS THE FUNCTION
0AFE	CD	B5	05	0748	CALL NAOUT GO PRINT HDR
0B01	C3	F2	0A	0749	JMP CALOP GO DO IT AGAIN

0B04		0750	*****
0B04		0751	*
0B04		0752	* COME HERE TO BUILD NEW SYSTEM
0B04		0753	*
0B04		0754	*****
0B04	CD 35 0A	0755	BILD1 CALL SETUP GET UNIT
0B07	CD AC 03	0756	CALL CRLF
0B0A	21 50 0C	0757	LXI H,BUMSG GET BUILD MSG
0B0D	CD B6 0B	0758	CALL MESS PUT OUT MSG
0B10	CD AC 03	0759	CALL CRLF
0B13	CD D3 04	0760	CALL WTLPI WAIT FOR ANSWER
0B16	CD 7E 03	0761	CALL INIT
0B19	21 8D 0F	0762	LXI H,FCBFP
0B1C	34	0763	INR M
0B1D	23	0764	INX H
0B1E	34	0765	INR M
0B1F	23	0766	INX H
0B20	36 41	0767	MVI M,'A'
0B22	21 00 00	0768	LXI H,START
0B25	22 90 0F	0769	SHLD FCBBG
0B28	36 F3	0770	MVI M,0F3H
0B2A	22 94 0F	0771	SHLD FCBAS
0B2D	21 08 0D	0772	LXI H,SYSRAM
0B30	22 92 0F	0773	SHLD FCBEN
0B33	21 0B 0F	0774	LXI H,FCB
0B36	CD 91 09	0775	CALL SAVE WRITE NEW RECORD
0B39	32 8B 0F	0776	STA FCBRC
0B3C	DA 63 0B	0777	JC ERRS
0B3F	3E 56	0778	MVI A,'V'
0B41	32 8C 0F	0779	STA FCBOP
0B44	21 8B 0F	0780	LXI H,FCB
0B47	CD 72 09	0781	CALL GET
0B4A	32 8B 0F	0782	STA FCBRC
0B4D	DA 63 0B	0783	JC ERRS
0B50	C9	0784	RET
0B51		0785	*****
0B51		0786	*
0B51		0787	* COME HERE TO PRINT ALL ERR MESSAGES
0B51		0788	*
0B51		0789	*****
0B51	CD 63 0B	0790	ERRSX CALL ERRS
0B54	C3 51 02	0791	JMP COMN1
0B57	C1	0792	ERRS0 POP PSW
0B58	E5	0793	PUSH H
0B59	CD 63 0B	0794	CALL ERRS
0B5C	E1	0795	POP H
0B5D	CD B5 05	0796	CALL NAOUT
0B60	C3 51 02	0797	JMP COMN1

U



0834 \*\*\*\*\*  
0835 \*  
0836 \* ALL SYSTEM MESSAGES FOLLOW  
0837 \*  
0838 \*\*\*\*\*  
0839 DB 49H  
0840 DW SYSRAM-UPE  
0841 UPE DB VE-UPE-1  
0842 ASC ' UNCORRECTABLE P.E. '  
  
0843 VE DB TIF-VE-1  
0844 ASC ' VERIFY ERROR '  
  
0845 TIF DB WP-TIF-1  
0846 ASC ' TAPE IS FULL '  
  
0847 WP DB FPE-WP-1  
0848 ASC ' WRITE PROTECTED '  
  
0849 FPE DB CAS-FPE-1  
0850 ASC ' FILE POSITION ERROR '  
  
0851 CAS DB UNK-CAS-1  
0852 ASC ' CAN'T AUTO START ''  
  
0853 UNK DB CTHDR-UNK-1  
0854 ASC ' UNKNOWN ERROR '  
  
0855 CTHDR DB BUMSG-CTHDR-1  
0856 ASC ' NAME TY LOAD SIZE '

08BB 49  
08BC 4A 01  
08BE 14  
08BF 20 55 4E 43  
4F 52 52 45  
43 54 41 42  
4C 45 20 50  
2E 45 2E 20  
  
08D3 0E  
08D4 20 56 45 52  
49 46 59 20  
45 52 52 4F  
52 20  
  
08E2 0E  
08E3 20 54 41 50  
45 20 49 53  
20 46 55 4C  
4C 20  
  
08F1 11  
08F2 20 57 52 49  
54 45 20 50  
52 4F 54 45  
43 54 45 44  
20  
  
0903 15  
0904 20 46 49 4C  
45 20 50 4F  
53 49 54 49  
4F 4E 20 45  
52 52 4F 52  
20  
  
0919 12  
091A 20 43 41 4E  
27 54 20 41  
55 54 4F 20  
53 54 41 52  
54 20  
  
092C 0F  
092D 20 55 4E 4B  
4E 4F 57 4E  
20 45 52 52  
4F 52 20  
  
093C 13  
093D 20 4E 41 4D  
45 20 54 59  
20 4C 4F 41  
44 20 53 49  
5A 45 20

*main system error  
severity*

*also in monitor*

*also in monitor*

*also in monitor*

*also in monitor*

*also in monitor*

0C50 26  
0C51 4D 4F 55 4E  
54 20 53 43  
52 20 54 41  
50 45 20 41  
4E 44 20 50  
52 45 53 53  
20 53 50 41  
43 45 20 4F  
52 20 43 4E  
43 4C

0C77 0F  
0C78 20 46 49 4C  
45 20 4E 4F  
54 20 46 4F  
55 4E 44

0C87 0F  
0C88 43 45 52 54  
49 46 59 49  
4E 47 20 54  
41 50 45

0C97 14  
0C98 41 50 50 52  
4F 58 2E 20  
43 41 50 41  
43 49 54 59  
20 2E 2E 2E

0CAC 0F  
0CAD 45 52 52 4F  
52 20 43 4F  
55 4E 54 20  
2E 2E 2E

0C 14  
0CDB 20 55 4E 49  
54 20 44 47  
4E 50 20 4E  
4F 54 20 45  
58 49 53 54

0CD1 1B  
0CD2 45 53 46 54  
4F 53 20 56  
45 52 53 49  
4F 4E 20 33  
2E 30 20 30  
32 2F 31 39  
2F 37 39

0CED  
0CED 45  
0CEE  
0CEE 00 00  
0CF0 00 00  
0CF2 00 00  
0CF4 00 00  
0CF6 00 00  
0CF8 00 00  
0CFA 00 00  
0CFC 00 00  
0CFE 00 00  
0D00 00 00  
0D02 00 00  
0D04 00 00  
0D06 00 00

0857 BUMSG DB UTIM2-BUMSG-1 HIT 1504  
0858 ASC 'MOUNT SCR TAPE AND PRESS SPACE OR CNCL'

0859 UTIM2 DB SIMES-UTIM2-1  
0860 ASC 'FILE NOT FOUND' - also in monitor

0861 SIMES DB SYSMC-SIMES-1  
0862 ASC 'CERTIFYING TAPE'

0863 SYSMC DB SYSME-SYSMC-1  
0864 ASC 'APPROX. CAPACITY ...' - also in monitor (Sasha)

0865 SYSME DB NOU-SYSME-1  
0866 ASC 'ERROR COUNT ...' - also in monitor (Sasha)

0867 NOU DB LEVEL-NOU-1  
0868 ASC 'UNIT DOES NOT EXIST'

0869 LEVEL DB ENDEM-LEVEL-1  
0870 ASC 'ESPTOS VERSION 3.0 02/19/79'  
UNICOS Version 3.1 08/22/81

0871 ENDEM EQU \$  
0872 RDVE DB 'E'  
0873 CUTAB EQU \$  
0874 DW 0  
0875 DW 0  
0876 DW 0  
0877 DW 0  
0878 DW 0  
0879 DW 0  
0880 DW 0  
0881 DW 0  
0882 DW 0  
0883 DW 0  
0884 DW 0  
0885 DW 0  
0886 DW 0

0D08	0888 SYSRAM EQU \$
0D08	0889 UIPRT DS 2
0D0A	0890 UOPRT DS 2
0D0C	0891 DFLTS DS 2
0D0E	0892 IPORT DS 1
0D0F	0893 QPORT DS 1
0D10	0894 SVCHR DS 1
0D11	0895 NCHAR DS 1
0D12	0896 LINE DS 1
0D13	0897 BOT DS 1
0D14	0898 SPEED DS 1
0D15	0899 ESCFL DS 1
0D16	0900 DS 1
0D17	0901 INPTR DS 1
0D18	0902 NUCNT DS 1
0D19	0903 DS 1
0D1A	0904 THEAD DS 5
0D1F	0905 DS 1
0D20	0906 HTYPE DS 1
0D21	0907 BLOCK DS 2
0D23	0908 LOADR DS 2
0D25	0909 XEQAD DS 2
0D27	0910 HSPR DS 3
0D2A	0911 DHEAD DS 16
0D3A	0912 SNUMF DS 1
0D3B	0913 FCBA1 DS 7
0D42	0914 FCBA2 DS 7
0D49	0915 FBUF1 DS 512
0F49	0916 DS 60
0F85	0917 SYSTP EQU \$
0F85	0918 HLSAV DS 2
0F87	0919 HFLG DS 1
0F88	0920 EOT1 DS 1
0F89	0921 FIN1 DS 2
0F8B	0922 FCB EQU \$
0F8B	0923 FCBRC DS 1
0F8C	0924 FCBOP DS 1
0F8D	0925 FCBFP DS 1
0F8E	0926 FCBID DS 1
0F8F	0927 FCBTY DS 1
0F90	0928 FCBBG DS 2
0F92	0929 FCBEN DS 2
0F94	0930 FCBAS DS 2
0F96	0931 FCBER DS 2
0F98	0932 CB EQU \$
0F98	0933 CBRC DS 1
0F99	0934 CBOP DS 1
0F9A	0935 CBFP DS 1
0F9B	0936 CBID DS 1
0F9C	0937 CBTY DS 1
0F9D	0938 CBBG DS 2
0F9F	0939 CBEN DS 2
0FA1	0940 CBAS DS 2
0FA3	0941 CBER DS 2
0FA5	0942 FNUMF DS 1
0FA6	0943 DS 60
0FE2	0944 STAK2 EQU \$

Added by Betty  
Feb 1970



SI	0474	0712							
ACOUT	04BA	0688	0903	0906	0994				
AINP	0022								
ALDAB	05B0								
ADUT	001C								
ARET1	0226	0368							
ARET2	022B	0351							
AI	0471	0703							
BLKOF	0007								
BOUT	04F1	0707	0778	0784	0901	0904	0907	0988	
CHAR	00FB	0166							
CHRLI	05DC	0912							
CLERA	023D	0377							
CLIN1	0179	0209	0261						
CLINE	0173	0310	0418						
CMN1	0251	0415	0582	0827	0892				
CMND	025F	0034	0404	0742	0760	0770	0870	0954	
COMTA	0315	0437	0496						
CONT	02AB	0422	0817						

CONU	046A	0706	
CONU2	04FF	0783	
COPRC	02B1	0403	
CREM	01BB	0171	0323 0428 0818-
CRLF	03AC	0586	0687 0815 0908 0981 0985
CUR	0149	0316	
CURCH	009B	0145	0227
CURSET	022F	0349	
CURSC	0147	0265	0345
CUSE2	0632	0962	
CUSET	0616	0517	
D1C1	B008	0088	
D1CE1	B011	0089	
D1GET	B00B	0086	
D1SAV	B00E	0087	
D1WT	B014	0085	
DEFLT	059E	0878	
DHCMP	068D		
DHLOP	068F	1047	
DISP0	02E1	0928	
DISP1	02E4	0441	
DISPD	05FA	0547	
DISPT	02E6	0124	0173
DLOOP	0444	0749	
DLOP	0480		
DLP1	0453	0711	
DLP2	0482	0746	
DN0	0492	0728	
DN1	0499	0731	
DO1	043F	0673	
DO2	0442	0677	0682
DOWN	048B	0725	
DUMP	0420	0501	
EDIT	0699		
ENLQ1	053A	0836	0840
ENLQ3	0557	0830	
ENLOP	0529	0825	
ENTER	0521	0503	
ERAS1	0157	0238	
ERAS3	016F	0210	
ERR1	0580	0435	0635 0653 0922
ERR2	0581	0443	0454
ERRIT	0392	0570	
ERR01	039D	0575	
ERR0T	0399	0565	
ESCE1	0067		
ESCE2	008F		
ESCS	01E8	0134	
ESCSP	01F1	0324	
ESFC1	0064		
ESFC2	008C		
ESFG1	005E		
ESFG2	0086		
ESFS1	0061		
ESFS2	0089		
EWTP1	005B	0084	
EWTP2	0083		
EXEC	055C	0505	
EXEC1	055F		
EXND	0083	0084	
EXINT	006A	0384	

BILD1	0B04				
BLID	0832	0511			
BLOCK	0D21	0109	0719		
BOPEN	074E				
BOT	0D13				
BUMSG	0C50	0757	0855	0857	
C	0A4E	0693			
CS	0A66	0657			
C3A	0A82	0667			
C4	0A86	0658	0671		
C5	0A92	0664	0676		
C5A	0A9D	0686			
CALOP	0AF2	0736	0749		
CAS	0C19	0820	0849	0851	
CAT1	0923	0425			
CATER	0909	0427	0428		
CB	0F98	0477			
CBAS	0FA1				
CBBC	0F9D	0430	0433		
CBEN	0F9F	0429			
CBER	0FA3				
CBFP	0F9A	0422			
CBID	0F9B				
CBOP	0F99				
CBRC	0F98	0423			
CBTY	0F9C				
CEBK	09CF	0581			
CERT1	0A3B				
CTBK	0942	0475	0536	0559	0589
CTHDR	0C3C	0733	0853	0855	
CTLG1	0AD5				
CUTAB	0CEE				
E	083F	0301			
ELETS	0D0C				
DHEAD	0D2A	0602	0609		
DRZ	069F	0011			
ENDEM	0CED	0869			
EOFER	0879	0264			
EOFW	07B6	0188			
EOT1	0F88				
ERRS	0B63	0639	0723	0777	0783 0790 0794
ERRS0	0B57	0618			
ERRSX	0B51	0669	0747		
ESCFL	0D15				
ESFOE	09B0	0656			
EX5	0A2A	0632			
FBUF1	0D49	0143			
FCB	0F8B	0032	0098	0121	0454 0514 0774 0780
FCBA1	0D3B	0157			
FCBA2	0D42	0162			



PCLOS	000A				
PDCOM	02F7	0440	0494	0927	
PDCOU	02F4	0442	0961		
POPEN	0007				
PCLIN	0282	0400	0407	0426	
POBAC	00CD	0325			
POBK	00E3	0138			
POUT	04BF	0697			
PDOV1	0416	0652			
PDOU1	0512	0798			
PDOU2	0509	0752	0755		
PINITA	0001				
PINPR	00A3	0044			
PINTBL	038A	0118			
PIOPRC	00A7	0127			
PKBDR	0001	0063			
PKBIN	003A	0567			
PKDATA	00FC	0065			
PLISTI	0642	0960			
PLOOPI	0648	0997			
PLORE	04AC	0739			
PLNAME	0567				
PLNAME1	056C	0863			
PLNAMES	0564	0953			
PLNAOUT	05B5	0891			
PLNCOM	030B	0480	0485		
PLNEXT	00E7	0169			
PLNFIL	058A	0856	0858	0875	
PLNLOOP	05D5	0900	0918	0987	
PLNOTLO	0296	0410	0412		
PLNOXST	009C	0094	0095	0096	0097
PLNULOT	03BA	0601			
PLORL	04EE	0689	0717	0789	0995
PLORAR	00FF	0338			
PLOROK	0139	0185	0188		
PLOMASK	009A	0180	0232	0255	0305
PLOUTBL	0382	0126			
PLOUTH	051D	0600	0804	0915	
PLOUTPR	00AF	0042			
PLPATCH	006B				
PLPBACK	01C3	0541			
PLPCR	01D0	0537			
PLPCUR	0194	0276	0311	0342	
PLPDATA	00FD	0050	0058		
PLPDOWN	0143	0529			
PLPER1	05AA				
PLPERSE	014D	0379	0525		
PLPESC	01E2	0543			
PLPHEAD	0669				
PLPHOME	0166	0535			
PLPLEFT	0190	0303	0531		
PLPLF	01D6	0539			
PLPLOAD	0684	1007			
PLPOS	04F6	0713	0791		
PLPFLDR	0002	0048			
PLPPLIN	0025	0569			
PLPLOU	002E	0056	0564		
PLPRIT	019A	0533			
PLPROMP	03A4	0399			
PLPSTAN	03C3	0675	0683	0955	
PLPSTOR	067B	1014			
PLPUP	0189	0527			

