

H01 11:24 SEP 08, 1975

1  
2  
3 01 0000U  
4  
5  
6  
7  
8  
9  
10  
11  
12  
13  
14  
15  
16  
17  
18  
19  
20  
1\* 00000001  
2\*  
22

\*M\* TELLUSR PRINTS REASON FOR JOBSTEP ABORT FOR BATCH & GHOST USERS<sup>1</sup>  
DEF TELLUSR: XDELTA LABEL FOR TELLUSR MODULE.  
TELLUSR: EGU \*  
\*P\*  
\*P\* NAME: TELLUSR  
\*P\*  
\*P\* PURPOSE: TO PUT OUT ERROR MESSAGES TO USERS WHOSE JOB STEPS  
\*P\* ARE ABNORMALLY TERMINATED FOR ANY REASON.  
\*P\* DESCRIPTION: TELLUSR WRITES A MESSAGE THRU MIXX (ASSIGNING IT  
\*P\* TO THE 'DB' DEVICE) GIVING THE REASON FOR A  
\*P\* JOBSTEP ABORT:  
\*P\* ERRORED/ABORTED BY OPERATOR  
\*P\* ERRORED/ABORTED BY THE PROGRAM ITSELF  
\*P\* LIMIT EXCEEDED (WHICH=LIMIT MSG FROM ERRMSG FILE)  
\*P\* I/O ERROR (WHICH=ERROR MSG FROM ERRMSG FILE)  
\*P\* (ALSO TELLS ON WHICH DCB)  
\*P\* OTHER ABORT (ABORTCODE MSG FROM ERRMSG FILE)  
\*P\* IT ALSO PRINTS THE PROGRAM LOCATION WHERE THE  
\*P\* ABORT OCCURRED.  
\*P\*  
BITS SET 1 GET DEFINITIONS OF BITS & MASKS.  
\*S\* SYSTEM UTS  
PCC 0

25		DEF	CLSXX	ENTRY TO CLOSE MIXX DCB
27		DEF	TELLU8R	ENTRY TO PRINT THE ABORT-REASON MESSAGE
28	*			
29		REF	CLSSEG	# OF 'CLOSE' MONITOR OVERLAY
30		REF	ER0	BITS 24-31 INPUT = ERROR SUBCODE
31	*,*			BITS 15-31 OUTPUT WHICH MAX. EXCEEDED
1*		REF	HEX	INPUT CHARACTERS '0' = 'F'
2*		REF	TSTACK	
32		REF	J:ABC	BITS 0-7 INPUT = ERROR CODE
33	*,*			OUTPUT = 0 IF MSG PRINTED
34		REF	J:ASSIGN	BIT 1 OUTPUT = NO BUFCHK ON M:WRITE
35	*,*			BITS 23-31 INPUT = WHICH MAX EXCEEDED
36		REF	J:DCBLINK	BITS 15-31 INPUT => USER DCB TABLE
37		REF	J:JIT	BASE ADDRESS OF JIT
41		REF	M:XX	OUTPUT DCB USED TO GET ERRMSG & PRINT
42		REF	MSRRDWT	ROUTINE READS OR WRITES THRU DCB
44		REF	0PNSEG	# OF 'OPEN' MONITOR OVERLAY
45		REF	PRINTV	ROUTINE PRINTS LINE GIVEN BUFFER, #CHAR

H01 11:24 SEP 08, 1975

		*	REGISTER	EQUATES
2*		R0	EQU	0
3*	00000000	R1	EQU	1
4*	00000001	R2	EQU	2
5*	00000002	R3	EQU	3
6*	00000003	R4	EQU	4
7*	00000004	R5	EQU	5
8*	00000005	R6	EQU	6
9*	00000006	R7	EQU	7
10*	00000007	R8	EQU	8
11*	00000008	R9	EQU	10
12*	0000000A	R10	EQU	10
13*	0000000A	R11	EQU	11
14*	0000000B	R12	EQU	12
15*	0000000C	R13	EQU	13
16*	0000000D	R14	EQU	14
17*	0000000E	R15	EQU	15
18*	0000000F			

```

20*
21*
22*
23*
24*
25*
26*
27*
28*
29*
30*
31*
32*
33*
34*
35*
36*
37*
38*
39*
40*
41*
42*
43* 01 00000 02200000 A
    01 00001 80000000 N
44* 01 00002 02200000 A
    01 00003 80E00000 N
45* 01 00004 6A000164
46* 01 00005 B2100000 X
47* 01 00006 222000F1 A
48* 01 00007 F5200001 A
49* 01 00008 22200001 A
50* 01 00009 224FFFF8 A
51* 01 0000A F2480000 X
52* 01 0000B 22300008 A
53* 01 0000C 21400001 A
54* 01 0000D 69460017
    
```

```

*D* NAME: TELLUSR
*D*
*D* REGISTERS: R14 PRESERVED ALL OTHERS ZAPPED.
*D* INTERFACE: MSRRDWT,(0PNSEG,0),(CLSSEG,0)
*D* ENVIRONMENT: MASTER MAPPED.
*D* INPUT: R0 = RETURN ADDRESS.
*D* R14 = ADDRESS OF BUFFER PAGE FOR TEMP USAGE.
*D* *TSTACK = SAVED JRNST AT USER EXIT TIME.
*D* (0-7)=EXIT TYPE; 80=TRAP, 40=I/OERR, 20=LIMIT,
*D* 10=LINEHANGUP, 08=0PABORT, 04=0PERROR,
*D* 02=MIXXX, 01=MIERR,
*D* (10-14)=WHO RUNNING; 002=PROCESSOR, 001=USER,
*D* 0008=LOADER, 0000=MONITOR.
*D* JIASSIGN(23-31)=WHICH LIMIT EXCEEDED: 100=PDISK NET,
*D* 80=TIME, 40=SCRATCHTAPE, 20=TDISK,
*D* 10=PDISK, 08=D0, 04=U0, 02=L0, 01=P0.
*D* JIABC(0-7)=ERRCODE (TRAP OR I/OERR)
*D* FR0(24-31)=SUBCODE (TRAP OR I/OERR)
*D* * THROUGHOUT THIS ROUTINE:
*D* * R1 = ADDRESS OF SCRATCH BUFFER PAGE.
*D* * R2 = CURRENT BYTE INDEX INTO BUFFER.
*D*
TELLUSR EGU *
      PUSH R0          SAVE RETURN ADDRESS.
      PUSH R14        SAVE BUFFER ADDRESS.
      BAL,R0 CLSXX      CLOSE MIXX IF OPEN.
      LW,R1 *TSTACK    R1 =>BUFFER.
      LI,R2 11'
      STB,R2 *R1
      LI,R2 1          INITIAL VFC = TOP-OF-PAGE.
                        R2 = INITIAL BUFFINDEX (1 IN).
      LI,R4 *8
      LB,R4 *TSTACK,R4 R4 = RNST BYTE 0.
      LI,R3 8          R3 = COUNTER FOR SCANNING RNST.
SCAN CI,R4 1          IS IT THIS BIT...
      BANZ CVEC,R3    ---> YES, GO TO ITS ROUTINE.
    
```

11:24 SEP 08, 1975

55\* 01 0000E 2540007F A  
 56\* 01 0000F 6430000C  
 57\*  
 58\* 01 00010  
 59\* 01 00010 22100000 A  
 60\* 01 00011 75100000 X  
 61\* 01 00012  
 62\* 01 00012  
 63\* 01 00012 6A000164  
 64\* 01 00013 02200000 A  
 01 00014 8AE00000 N  
 65\* 01 00015 02200000 A  
 01 00016 8A000000 N  
 66\* 01 00017 E8000000 A  
 67\*  
 68\*  
 69\* 01 00017  
 76 01 00018 68000020  
 77 01 00019 68000020  
 78 01 0001A 68000020  
 79 01 0001B 68000020  
 80 01 0001C 68000012  
 81 01 0001D 680000F8  
 82 01 0001E 6800008D  
 83 01 0001F 68000102

SLS,R4 =1  
 BDR,R3 SCAN

\* EXIT.  
 RETURN EQU \*

LI,R1 0  
 STB,R1 JIABC

RETURN1 EQU \*  
 CHKPT EQU \*

BAL,R0 CLSXX  
 PULL R14

PULL R0

B \*R0

\*  
 \*  
 CVEC EQU \*-1

B MSGOUT  
 B MSGOUT  
 B MSGOUT  
 B MSGOUT  
 B CHKPT  
 B MAXMSG  
 B IBERR  
 B ILLEGALTRAP

NO.  
 TRY NEXT BIT.

RESET JIABC TO SHOW MSG PRINTED.

CLOSE MIXX IF OPENED.  
 RESTORE BUFFER ADDRESS TO R14.

RESTORE RETURN ADDRESS.

--> EXIT TELLUSR.

MIERR CAL  
 MIXX CAL  
 ERRORED BY OPERATOR  
 ABORTED BY OPERATOR

I/O ERROR OF SOME KIND

MO1 11:24 SEP 08, '75

85			*			
86			*			
87			*			
88			*	COMMON USAGE OF REGISTERS IN THIS ROUTINE:		
89			*	R5 = INDEX TO A MESSAGE (FROM GETWHO)		
90			*	R1 = BUFFER ADDRESS OF BLOCKING BUFFER		
91			*	R2 = INDEX INTO THE BUFFER		
92			*	R6 = MESSAGE ADDRESS		
93		01 0002U	*			
94	01 00020	3266003C	MSGOUT	EGU	\$	
95	01 00021	6A000034		LW,6	MSG5,3	GET MESSAGE ADDRESS
96			*	BAL,0	FORM	FORM THE MESSAGE
97			*			
98	01 00022	21300002 A		C1,3	2	IS IT BY THE OPERATOR
99	01 00023	69200028		BG	0P	YES, CONTINUE
100	01 00024	6A00002B		BAL,0	GETWHO	GET THE GUILTY PARTY
101	01 00025	326A0041		LW,6	MSG5,5	TELL WHO DID IT
102	01 00026	68300028		BEZ	*+2	IF FLAG WAS MEANINGLESS
103	01 00027	6A000034		BAL,0	FORM	FORM THE MESSAGE
1*		01 00028	0P	EGU	\$	
109	01 00028	6A000074		BAL,0	AT	TELL HIM WHERE IT HAPPENED
110	01 00029	6AC0018C		BAL,12	WRTXX	WRITE THE MESSAGE
111	01 0002A	68000010		B	RETURN	DONE

```

113
114
115
116
117      01 0002B
1* 01 0002B 32300000 X
2* 01 0002C 3237FFFE A
119 01 0002D 2530006F A
120 01 0002E 22500005 A
121 01 0002F 21300001 A
122 01 00030 E9400000 A
123 01 00031 2530007F A
124 01 00032 6450002F A
125 01 00033 E8000000 A
126
127
128
129
130
131
132
133      01 00034
134 01 00034 F2800006 A
135 01 00035 22500001 A
136 01 00036 F24A0006 A
137 01 00037 F5440001 A
138 01 00038 20200001 A
139 01 00039 20500001 A
140 01 0003A 31500008 A
141 01 0003B 68200036 A
142 01 0003C E8000000 A
143
144
145
146
147
148      01 0003C
    
```

```

*
* RETURNS AN INDEX TO MSGS1 IN R5
* TO GIVE THE BY XXXX MESSAGE
*
GETWH8 EQU $
      LW,R3 TSTACK
      LW,R3 =2,R3      R3 = SAVED JIRNST WORD.
      SLS,R3 =17
      LI,R5 5          5 RUN FLAGS
TRY    CI,R3 1        PICK OFF THE GUILTY ONE
      BANZ =0        GOT HIM, RETURN
      SL6,R3 =1      GET THE NEXT BIT
      BDR,R5 TRY     TRY AGAIN
      B =0          GOT NONE
*
* 8 = COUNT IN MESSAGE
* 2 = PTR INTO BUFFER
* 1 = BUFFER ADDRESS
* 6 = MSG ADDRESS
*
FORM   DESTROYS 4,5,8
      EQU $
      LB,R8 =6      GET THE COUNT
      LI,R5 1
STORE  LB,R4 =6,5   TRANSFER THE MESSAGE
      STB,R4 =1,2
      AI,R2 1      BUMP THE POINTER
      AI,R5 1      BUMP THE MSG POINTER
      CW,R5 8      FINISHED THE MESSAGE
      BLE STORE   NOT YET, CONTINUE
      B =0
*
*
*
*
MSGS  EQU $=1
    
```

HO1 11124 SEP 08, 1975

149	01	0003D	00000047		DATA	JERR	
150	01	0003E	0000004C		DATA	JAB	
151	01	0003F	00000051		DATA	JBERR	
152	01	00040	00000059		DATA	JAERR	
153				*			
154				*			
155	01	00041	00000064	MSG51	DATA	BYM0N	
156	01	00042	0000006D		DATA	BYPR0C	
157	01	00043	0000006A		DATA	BYUSR	
158	01	00044	00000067		DATA	BYL0AD	
159	01	00045	00000000 A		DATA	0	
160	01	00046	00000000 A		DATA	0	
161				*			
162				*			
163	01	00047	10404040 A	JERR	TEXTC	!	J0B FRR0RED !
	01	00048	40D1D6C2 A				
	01	00049	40C5D9D9 A				
	01	0004A	D6D9C5C4 A				
	01	0004B	40404040 A				
164	01	0004C	10404040 A	JAB	TEXTC	!	J0B AB0RTED !
	01	0004D	40D1D6C2 A				
	01	0004E	40C1C2D6 A				
	01	0004F	D9E3C5C4 A				
	01	00050	40404040 A				
165	01	00051	10404040 A	JBERR	TEXTC	!	J0B ERR0RED BY 0PERATOR !
	01	00052	40D1D6C2 A				
	01	00053	40C5D9D9 A				
	01	00054	D6D9C5C4 A				
	01	00055	40C2E840 A				
	01	00056	D6D7C5D9 A				
	01	00057	C1E3D6D9 A				
	01	00058	40404040 A				
1*	01	00059	20404040 A	JAERR	TEXTC	!	J0B AB0RTED BY 0PERATOR 0R CANCELED !
	01	0005A	40D1D6C2 A				
	01	0005B	40C1C2D6 A				
	01	0005C	D9E3C5C4 A				
	01	0005D	40C2E840 A				



H01 11:24 SEP 08, 1975

	01	0005E	D6D7C5D9	A			
	01	0005F	C1E3D6D9	A			
	01	00060	40D6D940	A			
	01	00061	C3C1D8C3	A			
	01	00062	C5D3C5C4	A			
	01	00063	40404040	A			
167	01	00064	08C2E840	A	BYMON	TEXTC	IBY MONITOR !
	01	00065	D4D6D5C9	A			
	01	00066	E3D6D940	A			
168	01	00067	0AC2E840	A	BYLOAD	TEXTC	IBY LOADER !
	01	00068	D3D6C1C4	A			
	01	00069	C5D94040	A			
169	01	0006A	08C2E840	A	BYUSR	TEXTC	IBY USER !
	01	0006B	E4E2C5D9	A			
	01	0006C	40404040	A			
170	01	0006D	08C2E840	A	BYPRBC	TEXTC	IBY PROCESSOR !
	01	0006E	D7D9D6C3	A			
	01	0006F	C5E2E2D6	A			
	01	00070	D9404040	A			
171					*		
172					*		
173	01	00071	08C1E340	A	TXTAT	TEXTC	IAT !
174					*		
175	01	00072	04404040	A	SPACES	TEXTC	! !
	01	00073	40404040	A			

198				*			
199				*			
200				*			
201				*			
202				*			
203				*			
204		01 00074		*	DESTROYS	5,6,7,11	
205	01	00074	22B1FFFF	A	AT	EGU	*
206	01	00075	4AB00002	N		LI,11	X11FFFF1
207		01 00076				LS,11	TSTACK+2
208	01	00076	09000000	X	AT11	EGU	*
209	01	00077	22600071			PSW,0	TSTACK
210	01	00078	6A000034			LI,6	TXTAT
211	01	00079	22700000	A		BAL,0	FORM
212	01	0007A	3250000B	A		LI,7	0
213	01	0007B	6A00007E			LW,5	11
214	01	0007C	08000000	X		BAL,0	TRANS
215	01	0007D	E8000000	A		PLW,0	TSTACK
216						B	*0
217					*		
218					*		
219					*		
220					*		
221					*		
222					*		
223		01 0007E			*		
224	01	0007E	22800008	A	TRANS	EGU	*
225	01	0007F	22400000	A	TL00P	LI,8	8
226	01	00080	25400304	A		LI,4	0
227	01	00081	33000004	A		SCD,4	4
228	01	00082	69300085			MTW,0	4
229	01	00083	680E0084			BNEZ	TL0AD
230	01	00084	64800080			B	\$+1,7
231	01	00085	22700001	A	TL0AD	BDR,8	TL00P+1
1*	01	00086	72680000	X		LI,7	1
233	01	00087	F5640001	A		LB,R6	HEX,R4
234	01	00088	20200001	A		STB,6	*1,2
						AI,2	1

ROUTINE TO PUT OUT THE AT MESSAGE,  
TAKING THE LOCATION FROM THE PSD IN  
TSTACK

DESTROYS 5,6,7,11  
ADDRESS MASK  
GET THE ADDRESS  
ENTRY FOR MSG = AT 11  
SAVE RETURN  
PUT IN AT  
SKIP ZEROS  
IN THE ADDRESS  
RESTORE RETURN

R5 = WORD IN HEX TO BE TRANSLATED TO EBCDIC  
R7 = 0 => SUPPRESS LEADING ZEROS, = 1 => PUT THEM IN  
ASSUMES R1 = BUF, 2 = PTR INTO IT  
DESTROYS 4,5,6,7,8

COUNTER FOR HEX CONVERSION  
GET 4 BITS  
IS IT ZERO  
NO, PACK IT IN  
SUPPRESS ZEROS  
TURN OFF THE FLAG  
PICK UP THE EBCDIC  
AND PUT IT AWAY  
BUMP THE COUNTER

H01 11:24 SEP 08, '75  
235 01 00089 6480007F  
236 01 0008A E#000000 A

BDR,8 TL00P  
B \*0

GET THE FULL WORD  
YES, QUIT

```

240 *
241 *
242 * ROUTINE TO HANDLE THE I/O ERRORS
243 *
244 01 0008B 0000007F A X7F DATA X17F1
245 01 0008C 000E0000 A XMASK DATA X1E00001
246 *
247 *
248 01 0008D EQU $
249 01 0008D 72C00000 X LB,12 J,ABC GET THE ABORT CODE
250 01 0008E 21C00080 A CI,12 X1801 IS IT > 80
251 01 0008F 69200012 BG RETURN1 YES, DON'T HANDLE THIS MSG
252 01 00090 6A000118 BAL,0 WRITERR WRITE OUT THE ERROR
253 01 00091 22600072 LI,6 SPACES PRECEED THIS AT WITH 4 SPACES
254 01 00092 6A000034 BAL,0 FORM
255 01 00093 6A000074 BAL,0 AT TELL HIM WHERE
1* 01 00094 6A000173 BAL,R0 0PNXX OPEN MIXX TO DO DEVICE
257 01 00095 6AC0018C BAL,12 WRTXX WRITE THE RECORD
258 01 00096 6A000198 BAL,0 SETBUF SET VFC CHAR IN BUFFER
259 01 00097 2231FFFF A LI,3 X11FFFF1
260 01 00098 4A300002 N LS,3 TSTACK+2 GET THE PSD ADDRESS
261 *
262 * IN: R3 = ADDRESS FROM THE PSD IN THE TSTACK
263 * OUT: R4 = CAL INSTRUCTION IF FOUND, OTHERWISE EXIT TO NOCALEP
264 *
265 01 00099 GETCAL EQU $
266 01 00099 3130000F A CW,3 X1F1 IN A REGISTER
267 01 0009A 6920009C BG $+2 NO
268 01 0009B 20300005 N AI,3 TSTACK+5 YES, GET THE RIGHT LOCATION
269 01 0009C 22600002 A LI,6 2 DO IT TWICE
270 01 0009D 203FFFFF A AI,3 =1
271 01 0009E 82400003 A GETCAL1 LW,4 *3
272 01 0009F 72500004 A LB,5 4 GET THE 0P CODE
273 01 000A0 4B50008B AND,5 X7F WITHOUT THE INDIRECT BIT
274 01 000A1 21500004 A CI,5 X141
275 01 000A2 683000B3 BE 00TCAL YES, GOT IT
276 01 000A3 21500067 A CI,5 X1671 IS IT AN EXU

```

H01 11:24 SEP 08, 1975

277 01 000A4 683000A8  
 278 01 000A5 20300001 A  
 279 01 000A6 6460009E  
 280 01 000A7 68000010  
 281  
 282  
 283 01 000A8  
 284 01 000A8 72500004 A  
 285 01 000A9 21500080 A  
 286 01 000AA 684000B0  
 287 01 000AB 2251FFFF A  
 288 01 000AC CA400004 A  
 289 01 000AD 2140000F A  
 290 01 000AE 692000B0  
 291 01 000AF 20400005 N  
 292 01 000B0 2231FFFF A  
 293 01 000B1 4A300004 A  
 294 01 000B2 68000099  
 295  
 296 01 000B3  
 297 01 000B3 2231FFFF A  
 298 01 000B4 4A300004 A  
 299 01 000B5 31400000 F  
 \*\*\*\* UNDEF SYM  
 300 01 000B6 684000BD  
 301 01 000B7 2130000F A  
 302 01 000B8 692000BA  
 303 01 000B9 32360005 N  
 304 01 000BA 2130000F A  
 305 01 000BB 692000BD  
 306 01 000BC 20300005 N  
 307 01 000BD  
 308 01 000BD 3270008C  
 309 01 000BE 31400007 A  
 310 01 000BF 684000C4  
 311 01 000C0 4A600004 A  
 312 01 000C1 2560006F A

BE GETEXU  
 AI,3 1  
 BDR,6 GETCAL  
 B RETURN  
 \*  
 \*  
 GETEXU EQU \*  
 LB,5 4  
 CI,5 X1801  
 BAZ GETEXU1 NO  
 LI,5 X11FFFF1 ADDRESS MASK  
 LS,4 \*4 GET THE ADDRESS  
 CI,4 XIF1  
 BG \*+2 NO  
 AI,4 TSTACK+5 YES, CREATE PROPER DISPLACEMENT  
 LI,3 X11FFFF1 ADDRESS MASK  
 LS,3 4 GET THE ADDRESS INTO 3  
 B GETCAL AND TRY AGAIN  
 \*  
 GETCAL EQU \*  
 IBC LI,3 X11FFFF1  
 LS,3 4  
 CW,4 Y8  
 BAZ FPT  
 CI,3 XIF1 YES, WAS IT TO A REGISTER  
 BG INDR NO, CONTINUE  
 LW,3 TSTACK+5,3 GET THE ADDRESS FROM REGISTER  
 CI,3 XIF1 IS IT IN A REGISTER  
 BG \*+2 NO, CONTINUE  
 AI,3 TSTACK+5 YES, GET IT  
 FPT EQU \*  
 LW,7 XMASK SEE IF IT WAS INDEXED  
 CW,4 7  
 BAZ NOINDEX  
 LS,6 4  
 SLS,6 =17 GET THE REGISTER

YES, TRACE IT OUT  
 BUT ONLY TRY ONCE  
 THEN GIVE UP

GET THE OP CODE AGAIN

NO  
 ADDRESS MASK  
 GET THE ADDRESS

NO  
 YES, CREATE PROPER DISPLACEMENT  
 ADDRESS MASK  
 GET THE ADDRESS INTO 3  
 AND TRY AGAIN

GET THE FPT ADDRESS  
 WAS IT INDIRECT

NO, CONTINUE  
 YES, WAS IT TO A REGISTER  
 NO, CONTINUE  
 GET THE ADDRESS FROM REGISTER  
 IS IT IN A REGISTER  
 NO, CONTINUE  
 YES, GET IT

SEE IF IT WAS INDEXED

GET THE REGISTER

HO1 11124 SEP 08, 175

313	01	000C2	326C0005	N	LW,6	TSTACK+5,6	
314	01	000C3	30300006	A	AW,3	6	INDEX
315		01 000C4			NOINDEX	8	
316	01	000C4	2130000F	A	CI,3	XIF'	IN A REGISTER
317	01	000C5	692000C7		BG	*+2	NO
318	01	000C6	20300005	N	AI,3	TSTACK+5	YES, POINT TO THE RIGHT PLACE
319	01	000C7	82300003	A	LW,3	*3	
320	01	000C8	31300000	F	CW,3	Y8	IS IT INDIRECT
**** UNDEF	SYM						
321	01	000C9	684000D0		BAZ	NOINDR	NO, CONTINUE
322	01	000CA	2251FFFF	A	LI,5	XI'1FFFF'	
323	01	000CB	4A500003	A	LS,5	3	GET THE ADDRESS
324	01	000CC	2150000F	A	CI,5	XIF'	A REGISTER
325	01	000CD	692000CF		BG	*+2	NO, CONTINUE
326	01	000CE	20500005	N	AI,5	TSTACK+5	
327	01	000CF	82300005	A	LW,3	*5	
**** UNDEF	SYM						
1*	01	000D0	48300000	F	NOINDR	AND,3	GET IT AS AN ADDRESS ONLY.
329	01	000D1	22400000	N	LI,4	J,DCBLINK	
330	01	000D2	F2500004	A	DCBL00P	LB,5	*4
331	01	000D3	683000DC		BEZ	NXTCHN	NO COUNT, GET LINK
332	01	000D4	20500004	A	AI,5	4	SKIP THE NAME
333	01	000D5	2550007E	A	SLS,5	*2	
334	01	000D6	30500004	A	AW,5	4	
335	01	000D7	81300005	A	CW,3	*5	IS IT OUR DCB
336	01	000D8	683000E0		BE	FOUND	YES, SUCCESS
337	01	000D9	20500001	A	AI,5	1	NEXT NAME
338	01	000DA	32400005	A	LW,4	5	
339	01	000DB	680000D2		B	DCBL00P	
340					*		
341					*		
342	01	000DC	82400004	A	NXTCHN	LW,4	*4
343	01	000DD	683000E7		BEZ	NODCB	IS IT A LINK
344	01	000DE	20400001	A	AI,4	1	NO, CONTINUE
345	01	000DF	680000D2		B	DCBL00P	YES, BUMP PAST LINK
346					*		
347					*		

```

348
349
350
351      01 000E0
352 01 000E0 32700004 A
353 01 000E1 226000F5
354 01 000E2 6A000034
355 01 000E3 32600007 A
356 01 000E4 6A000034
357 01 000E5 6AC0018C
358 01 000E6 68000010
359
360
361 01 000E7 226000ED
362 01 000E8 6A000034
363 01 000E9 32B00003 A
364 01 000EA 6A000076
365 01 000EB 6AC0018C
366 01 000EC 68000010
367
368
369 01 000ED 10404040 A
      01 000EE 40D5D6D5 A
      01 000EF 60C5F7C9 A
      01 000F0 E2E3C5D5 A
      01 000F1 E340C4C3 A
      01 000F2 C240C1C4 A
      01 000F3 C4D9C8E2 A
      01 000F4 E2404040 A
370 01 000F5 0B404040 A
      01 000F6 40D6D540 A
      01 000F7 C4C3C240 A
    
```

```

*      3 = ADDRESS OF DCB
*      4 = ADDRESS OF ITS NAME
*
FOUND  EQU      *
      LW,7      4      SAVE THE DCB NAME LOCATION
      LI,6      0ND C BMSG
      BAL,0     FORM   TELL HIM WHICH DCB DID IT
      LW,6      7      PRINT THE DCB NAME
      BAL,0     FORM
      BAL,12    WRTXX  WRITE THE RECORD
      B         RETURN
*
*
NDCB   LI,6      NDCBMSG  PUT OUT THE MSG
      BAL,0     FORM
      LW,11     3      AND TELL HIM THE DCB ADDR
      BAL,0     AT11
      BAL,12    WRTXX  WRITE THE RECORD
      B         RETURN  AND GET OUT
*
*
NDCBMSG TEXTC ' NON-EXISTENT DCB ADDRESS '
*
*
0ND C BMSG TEXTC ' ON DCB '
    
```

524				*			
525				*			
526				*			
527				*			
528		01	000F8	MAXMSG	EQU	*	
529	01	000F8	22400020 A		LI,4	3P	DETERMINE WHICH ONE
530	01	000F9	65300000 X		INT,3	J:ASSIGN	
531	01	000FA	45300000 F		AND,3	M9	
**** UNDEF	SYM						
532	01	000FB	25300001 A		SLS,3	1	
533	01	000FC	698000FE		BCS,8	*+2	
534	01	000FD	644000FB		BDR,4	*-2	
535	01	000FE	2251FFFF A		LI,5	X'1FFFF'	
536	01	000FF	47400000 F		STB,4	J:JIT+ER0	
537	01	00100	224000B3 A		LI,4	X'B3'	
538	01	00101	75400000 X		STB,4	J:ABC	
539		01	00102	ILLEGALTRAP	EQU	*	
540	01	00102	6A00002B		BAL,0	GETWHO	
541	01	00103	09500000 X		PBW,5	TSTACK	
545	01	00104	6A000118		BAL,0	WRITERR	GIVE THE MSG FROM ERRMSG
546	01	00105	22600072		LI,6	SPACES	
547	01	00106	6A000034		BAL,0	FORM	
548	01	00107	08500000 X		PLW,5	TSTACK	
549	01	00108	326A0041		LW,6	MSG\$1,5	PICK UP THE MSG
550	01	00109	6530010B		BEZ	TRAP2	UNLESS IT WAS MEANINGLESS
551	01	0010A	6A000034		BAL,0	FORM	
552	01	0010B	6A000074	TRAP2	BAL,0	AT	
553	01	0010C	22600113		LI,6	CONTMSG	TELL WHY
554	01	0010D	6A000034		BAL,0	FORM	FORM IT UP
555	01	0010E	82500002 N		LW,5	*TSTACK+2	WHERE WE TRAPPED
556	01	0010F	22700001 A		LI,7	1	DON'T SUPPRESS ZEROS
557	01	00110	6A00007E		BAL,0	TRANS	
558	01	00111	6AC0018C		BAL,12	WRTXX	
559	01	00112	65000010		B	RETURN	DONE
560				*			
561				*			
562	01	00113	1040E6CB A	CONTMSG	TEXTC	' WHICH CONTAINS '	



H01 11:24 SEP 08, '75

01	00114	C9C3C840	A
01	00115	C3D6D5E3	A
01	00116	C1C9D5E2	A
01	00117	4U404040	A

```

2*
3*
4*
5*
6*
7*
8*           01 00118
9* 01 00118   72C00000 X
10* 01 00119  E8300000 A
11* 01 0011A  20C30000 A
12* 01 0011B  25C00008 A
13* 01 0011C  22D000FF A
14* 01 0011D  4AC00000 F
15* 01 0011E  35C20000 A
16* 01 0011F  02200000 A
        01 00120  84200000 A
**** ILLEGAL AF
17* 01 00121  022000C0 A
18* 01 00122  2A200158
19* 01 00123  24220001 A
20* 01 00124  22600000 N
21* 01 00125  32700001 A
22* 01 00126  20700002 A
23* 01 00127  22800014 A
24* 01 00128  00000000 X
**** UNDEF C0M
**** ILLEGAL CF
25* 01 00129  82100000 X
26* 01 0012A  52500000 X
27* 01 0012B  21500020 A
28* 01 0012C  68400142
29* 01 0012D  32500000 F
**** UNDEF SYM
30* 01 0012E  47500000 X
31* 01 0012F  22600000 N
32* 01 00130  32700000 X
33* 01 00131  22800010 A
    
```

```

*
*
*
*
*
*
*
*
WRITERR EQU
    
```

```

ROUTINE TO READ THE ERROR MESSAGE FILE AND PRINT THE
ERROR MESSAGE OR THE ERROR NUMBER IF NO SUCH MESSAGE.
IN: R0=LINK, R1=WA(BUFFER)
OUT: R1 PRESERVED.
    
```

```

LB,R12      J,ABC
BEZ         *R0
AI,R12     X10300001
SLS,R12    +8
LI,R13     X'FF'
LS,R12     J:JIT+ER0
STW,R12    0,R1
PUSH       2,R0
    
```

```

IS THERE REALLY AN ERROR...
---> NO. DON'T PRINT ANYTHING.
    
```

R12= 03 / 00 / J:ABC / 00

```

R12= 03 / 00 / J:ABC / J:ER0
SAVE KEY IN WORD0 OF BUFFER.
SAVE RETURN & BUFFER ADDRESS.
    
```

```

LCI        12
LM,R2     0PEN
STM,R2    1,R1
LI,R6     M:XX
LW,R7     R1
AI,R7     2
LI,R8     X'14'
OVERLAY   0PNSEG,0
    
```

```

GET 0PEN FPT AND
COPY IT TO BUFFER.
R6 = DCB ADDRESS.
    
```

```

R7 =>FPT + 1.
R8 = FPT CODE (M:0PEN).
OPEN MIXX TO ERRMSG FILE.
    
```

```

LW,R1     *TSTACK
LH,R5     MIXX
CI,R5     X'20'
BAZ       KEYCONV
LW,R5     Y4
    
```

```

REFRESH BUFFER ADDRESS IN R1.
IF MIXX DIDN'T GET 0PEN,
WE CAN'T READ IT.
SET !DONT-CHECK-BUFFER-ACCESS!
    
```

```

STB,R5    J:ASSIGN
LI,R6     M:XX
LW,R7     TSTACK
LI,R8     X'10'
    
```

```

FOR THE READ.
R6 = DCB ADDRESS.
R7 =>FPT.
R8 = FPT CODE (M:READ).
    
```

34\* 01 00132 32A0019C  
 35\* 01 00133 22A00142  
 36\* 01 00134 32B00001 A  
 37\* 01 00135 20B00001 A  
 38\* 01 00136 22C0008C A  
 39\* 01 00137 32D00001 A  
 40\* 01 00138 02200000 A  
 01 00139 8450000A A

LW,R9 L(X'78000000')  
 LI,R10 KEYCONV  
 LW,R11 R1  
 AI,R11 1  
 LI,R12 140  
 LW,R13 R1  
 PUSH 5,R9

SET UP FPT;  
 \*ABN  
 \*BUF = (BUFFER) + 1  
 \*SIZE = 140  
 \*KEY IS IN BUFFER WORD 0.  
 PUT FPT INTO STACK.

\*\*\*\* ILLEGAL AF

41\* 01 0013A 6AB00000 X  
 42\* 01 0013B 22BFFFFB A  
 43\* 01 0013C 13B00000 X  
 44\* 01 0013D 21A00000 A  
 45\* 01 0013E 69300142  
 46\* 01 0013F 3240000D N  
 47\* 01 00140 20400003 A  
 48\* 01 00141 6800014D  
 49\* 01 00142  
 50\* 01 00142 82100000 X  
 51\* 01 00143 32320000 A  
 52\* 01 00144 25300008 A  
 53\* 01 00145 22400003 A  
 54\* 01 00146 22200000 A  
 55\* 01 00147 25200104 A  
 56\* 01 00148 72240000 X  
 57\* 01 00149 F5280001 A  
 58\* 01 0014A 20400001 A  
 59\* 01 0014B 21400009 A  
 60\* 01 0014C 69100146  
 61\* 01 0014D  
 62\* 01 0014D 02200000 A  
 01 0014E 8A100000 N  
 63\* 01 0014F 3220019D  
 64\* 01 00150 35220000 A  
 65\* 01 00151 32200004 A  
 66\* 01 00152 6A000173  
 67\* 01 00153 6AC0018C

BAL,R11 MSRRDWT  
 LI,R11 =5  
 MSP,R11 TSTACK  
 CI,R10 0  
 BNE KEYCONV  
 LW,R4 MIXX+13  
 AI,R4 4-1  
 B WTERRBUT  
 EQU \*  
 LW,R1 \*TSTACK  
 LW,R3 0,R1  
 SLS,R3 +8  
 LI,R4 3  
 GO LI,R2 0  
 SLD,R2 +4  
 LB,R2 HEX,R2  
 STB,R2 \*R1,R4  
 AI,R4 +1  
 CI,R4 3+6  
 BL GO  
 WTERRBUT EQU \*  
 PULL R1  
 LW,R2 L(11 )  
 STW,R2 0,R1  
 LW,R2 R4  
 BAL,R0 @PNXX  
 BAL,R1 @WRTXX

READ MESSAGE FROM ERRMSG FILE.  
 REMOVE FPT.  
 WAS THE READ SUCCESSFUL...  
 ==> NO, PRINT KEY ONLY.  
 YES, GET SIZE OF MESSAGE  
 R4=MSGSIZE (+4 1WORD, =1 CR)  
 ==> GO PRINT MESSAGE NOW.  
 REFRESH BUFFER ADDRESS IN R1.  
 GET KEY AGAIN.  
 STRIP OFF BYTE COUNT.  
 START 3 BYTES INTO BUFFER.  
 GET A HALFBYTE OF KEY.  
 CONVERT TO EBCDIC.  
 STORE IN BUFFER.  
 REPEAT FOR 6 HALFBYTES.  
 GET ERRMSG OR DECODED # IN BUF.  
 REMOVE BUFFER ADDRESS FROM STACK.  
 PUT PAGE=EJECT INTO  
 BEGINNING OF BUFFER.  
 R2 = RECORD LENGTH,  
 NOW OPEN MIXX TO DO DEVICE.  
 WRITE ERROR MESSAGE.

11:24 SEP 08, '75  
 68\* 01 00154 6A000198  
 69\* 01 00155 02200000 A  
     01 00156 8A000000 N  
 70\* 01 00157 E8000000 A  
 71\*  
 72\* 01 00158 0U000000 A  
 73\* 01 00159 C1020001 A  
 74\* 01 0015A 0U000142  
 75\* 01 0015B 0U000142  
 76\* 01 0015C 0U000001 A  
 77\* 01 0015D 0U000000 A  
 78\* 01 0015E 01000202 A  
 79\* 01 0015F 06C5D9D9 A  
     01 00160 D4E2C740 A  
 80\* 01 00161 02010202 A  
 81\* 01 00162 7AE2E8E2 A  
     01 00163 4U404040 A

\*  
 OPEN

BAL,RO    SETBUF  
 PULL      RO  
  
 B          \*RO  
  
 DATA     0  
 DATA     X'1020001'  
 PZE       KEYCONV  
 PZE       KEYCONV  
 DATA     1  
 DATA     0  
 DATA     X'01000202'  
 TEXTC     'ERRMSG'  
  
 DATA     X'02010202'  
 TEXT      ';SYS     '

SET UP BUFFER AGAIN.  
 (RESTORE RETURN ADDRESS)  
  
 RETURN TO CALLER.  
  
 \*ERR  
 \*ABN  
 \* IN  
 \*BTD = 0  
 \*FILENAME:  
  
 \*ACCOUNT#:

\* LIO P:NONE.

CLSXX EQU \*  
 LI,R2 2  
 LH,R1 MIXX  
 CI,R1 X'0020'  
 BAZ \*R0  
 LW,R1 YB

CLOSE MIXX IF OPEN.  
 (SAVE).

IF CLOSED ALREADY,  
 ---> FINISHED.  
 R1/R2 = MICLOSE FPT.

SAVE RETURN & FPT IN STACK.

R6 = MIXX DCB ADDRESS.

R7 =>FPT.  
 R8 = M;CLOSE FPT CODE.  
 CLOSE MIXX.

RESTORE RETURN & LEVEL STACK.

---> RETURN.

\* LIO P:1,2.

BPNXE EQU \*  
 LCI 5  
 LM,R3 XXT0D8  
 PUSH 8,R0  
 LH,R1 MIXX  
 CI,R1 X'0020'  
 BAZ BPNXE10  
 BAL,R0 CLSXX  
 BPNXE10 LI,R6 MIXX

OPEN MIXX TO 'D8'.

LOAD FPT FOR MIBPEN TO 'D8'.  
 SAVEREGS & FPT TO STACK.

IS MIXX OPEN...  
 ---> NO.  
 YES. CLOSE IT FIRST.  
 R6 =>DCB (MIXX).

83\*  
 84\* 01 00164  
 85\* 01 00164 22200002 A  
 86\* 01 00165 52100000 X  
 87\* 01 00166 21100020 A  
 88\* 01 00167 E8400000 A  
 89\* 01 00168 32100000 F

\*\*\*\* UNDEF SYM

90\* 01 00169 02200000 A  
 01 0016A 8B300000 A

\*\*\*\* ILLEGAL AF

91\* 01 0016B 22600000 N  
 92\* 01 0016C 32700000 X  
 93\* 01 0016D 207FFFFFFF A  
 94\* 01 0016E 22800015 A  
 95\* 01 0016F 00000000 X

\*\*\*\* UNDEF COM

\*\*\*\* ILLEGAL CF

96\* 01 00170 02200000 A  
 01 00171 8A300000 A

\*\*\*\* ILLEGAL AF

97\* 01 00172 E8000000 A

99\*  
 100\* 01 00173  
 101\* 01 00173 02200050 A  
 102\* 01 00174 2A300187  
 103\* 01 00175 02200000 A  
 01 00176 8B800000 A

\*\*\*\* ILLEGAL AF

104\* 01 00177 52100000 X  
 105\* 01 00178 21100020 A  
 106\* 01 00179 6840017B  
 107\* 01 0017A 6A000164  
 108\* 01 0017B 22600000 N

```

H01 11:24 SEP 08, 175
109* 01 0017C 32700000 X
110* 01 0017D 207FFFFD A
111* 01 0017E 22800014 A
112* 01 0017F 00000000 X
**** UNDEF COM
**** ILLEGAL CF
113* 01 00180 22200100 A
114* 01 00181 22304130 A
115* 01 00182 47200000 X
116* 01 00183 75200014 N
117* 01 00184 02200000 A
      01 00185 8A800000 A
**** ILLEGAL AF
118* 01 00186 E8000000 A
119*
120* 01 00187 00000000 A
121* 01 00188 C0040000 A
122* 01 00189 00000189
123* 01 0018A 0000018A
124* 01 0018B 0000C4D6 A

126*
127*      01 0018C
128* 01 0018C 52E00000 X
129* 01 0018D 21E00020 A
130* 01 0018E 69400194
131* 01 0018F 02200000 A
      01 00190 8AC00000 N
132* 01 00191 6A000173
133* 01 00192 02200000 A
      01 00193 8AC00000 N
627 01 00194 32E00001 A
628 01 00195 32F00002 A
629 01 00196 22600000 N
630 01 00197 68000000 X

```

```

LW,R7 TSTACK
AI,R7 #3
LI,R8 X'14'
OVERLAY 0PNSEG,0

LI,R2 X'100'
LI,R3 X'4130'
STS,R2 M;XX
STB,R2 M;XX+20
PULL 8,R0

B *R0

*
*XT0DB DATA 0
DATA X'C0040000'
PZE *
PZE *
DATA 'DB'

* I11,2 L:12 P11
WRTXX EQU *
LW,R14 MIXX
CI,R14 X'0020'
BANZ WRTXX10
PUSH R12

BAL,R0 0PNXX
PULL R12

WRTXX10 LW,14 1
LW,15 2
LI,6 M;XX
B PRINTV

```

```

R7 =>FPT #1.
R8 = FPT CODE (MISOPEN).
OPEN MIXX.

SET VFC,
CLEAR FBCD, SET BTD=0
IN MIXX,
NO PAGE HEADERS.
RESTORE REGS & LEVEL STACK.

---> RETURN.

MISOPEN MIXX, (DEVICE, 'DB')

*ERR (NONZER0)
*ABN (NONZER0)
*DEVICE, 'DB'

WRITE TO MIXX BUF,R1 SIZE,R2.

IS MIXX OPEN...
---> YES.
NO.

OPEN MIXX TO 'DB' DEVICE.

SET BUFFER ADDRESS
SET BUFFER SIZE
SET DCB ADDRESS

```

H01 11:24 SEP 08, 1975

23

```
631 *
635 01 00198 22200040 A SETBUF LI,2 ' ' VFC CHAR
636 01 00199 F5200001 A STB,2 *1 TO BUFFER
637 01 0019A 22200001 A LI,2 1 BUFFER POINTER
638 01 0019B E8000000 A B *0
639 *
640 END
01 0019C 78000000 A
01 0019D F1404040 A
```

CONTROL SECTION SUMMARY: 01 0019E PT 0 02 00000 PT 0 03 00000 PT 1 04 00000 PT 0

\* SYMBOL VALUES

AT/01 00074	AT11/01 00076	BITS/00000001	BYL8AD/01 00067
BYM8N/01 00064	BYPR8C/01 0006D	BYUSR/01 0006A	CHKPT/01 00012
C8NTMSG/01 00113	CVEC/01 00017	DCBL88P/01 000D2	F8RM/01 00034
F8UND/01 000E0	FPT/01 0008D	GETCAL/01 00099	GETCAL1/01 0009E
GETEXU/01 000A8	GETEXU1/01 000B0	GETWH8/01 0002B	88/01 00146
88TCAL/01 000B3	ILLEGALTRAP/01 00102		INDR/01 000BA
I8C/01 000B3	I8ERR/01 0008D	JITCB/EXT	JAB/01 0004C
JAERR/01 00059	JERR/01 00047	JBERR/01 00051	KEYCONV/01 00142
MASK/FUNC	MAXMSG/01 000F8	MSG8UT/01 00020	MSG8/01 0003C
MSG81/01 00041	NMASK/FUNC	N8DCB/01 000E7	N8DCBMSG/01 000ED
N8INDEX/01 000C4	N8INDR/01 000D0	NXTCHN/01 000DC	8NDCBMSG/01 000F5
8P/01 00028	8PEN/01 00158	8PNXX/01 00173	8PNXX10/01 0017B
RETURN/01 00010	RETURN1/01 00012	R0/00000000	R1/00000001
R10/0000000A	R11/0000000B	R12/0000000C	R13/0000000D
R14/0000000E	R15/0000000F	R2/00000002	R3/00000003
R4/00000004	R5/00000005	R6/00000006	R7/00000007
R8/00000008	R9/0000000A	SCAN/01 0000C	SETBUF/01 00198
SPACES/01 00072	STORE/01 00036	TL8AD/01 00088	TL88P/01 0007F
TRANS/01 0007E	TRAP2/01 0010B	TRY/01 0002F	TXAT/01 00071
WRITERR/01 00118	WRTXX/01 0018C	WRTXX10/01 00194	WTERR8UT/01 0014D
XMASK/01 0008C	XT8D8/01 00187	X7F/01 0008B	

\* EXTERNAL DEFINITIONS

CLSXX/01 00164	TELLUSR/01 00000	TELLUSR1/01 00000
----------------	------------------	-------------------

\* PRIMARY REFERENCES

CLSSFG	ER8	HEX	J1ABC	J1ASSIGN	J1DCBLINK	J1JIT
M1XX	M8RRDWT	8PNSEG	PRINTV	TSTACK		

\* N8 SECONDARY REFERENCES

\* UNDEFINED SYMBOLS

M17	M9	Y4	Y8
-----	----	----	----

\* ERROR SEVERITY LEVEL 3

\* ERROR LINES

299	320	327.1	531	562.16	562.24	562.29
562.40	562.89	562.90	562.95	562.96	562.103	562.112
562.117						