

ADDRESS LENGTH
 0 573
 573

BINARY CONTROL CARDS.

IDENT SYSENR
END

BLOCKS	TYPE	ADDRESS	LENGTH
ABSOLUTE*	ABSOLUTE	0	170
PROGRAM*	LOCAL	0	573
ESENT	COMMON	0	404

ENTRY POINTS.

TOSPROC	-	3	NEWTOS	-	516	L.FRRR	-	26	E.STOPP	-	270
SYSRET	-	7	NEWTP	-	516	KILLP	-	526	L.STOPP	-	16
S.RETU	-	44	E.NEWT	-	51	E.KILLP	-	244	PUTRETP	-	541
SYSFRET	-	0	L.NEWT	-	145	L.KILLP	-	12	PPAR	-	541
SFRET	-	27	E.ERROR	-	521	DOSWAP	-	531	E.PPAR	-	306
E.SFRET	-	0	ERRPROC	-	522	E.DSWAP	-	256	L.PPAR	-	76
L.SFRET	-	51	ERRR	-	523	L.DSWAP	-	12			
ACTIONL	-	125	E.FRRR	-	216	STOPP	-	534			

EXTERNAL SYMBOLS.

GETCAP	CREABLK	UCAP	GETEVF	CHMPRO	CAPTY	I.LOCK	IPD
SUBPCAL	WRITFIL	FCAP	CLRDAE	DSPSCIX	PSANY	DISASTR	E.FCS
SUBPJUM	READFIL	ACAP	SETDAE	JUMPCAL	BDBLK	S.USERA	PRINT
E.MUT	EVNT	OROPT	MEVHANG	PARMBUF	NOXJ	ENVIORN	
S.QUANT	EVPHANG	CHKBLK	MEVHNGF	RIGCNT	CLMOT	Z.ASYS	
S.CHARG	SAVREG	DELBLK	DSPCLOX	NEGPAR	NEGIX	Z.ESYS	
S.SYSTN	RESTORE	MARZERO	RDAT	NOTCL	BCBLK	SWABOUT	
S.USRTM	RETURN	MPCHGRD	BCAP	BIGIX	NEGPT	Z.LSYS	
S.SWPTM	UDAT	MPCHGRW	TRDB	BIGPCT	DESCHED	BIGPT	
S.OLDTM	FDAT	MOVBLK	CHMPRW	NEGPT	CLKWAIT	NOOP	

IDENT SYSENTR

```

*
* THE ROUTINES CONTAINED HEREIN ARE THE INTERFACE BETWEEN THE SYSTEM
* AND THE USER. THE SYSTEM RETURNS TO THE USER AT S.RETU AND THE
* USER CALL THE SYSTEM AT S.RETU+1.
*
* THERE ARE NO SYSTEM ACTIONS HEREIN, ALTHOUGH THE JUMP TABLE
* WHICH TRANSFERS TO THE VARIOUS ACTIONS IS HERE.
*
* EXTERNAL SUBROUTINES HEREIN
*   TOSPROC, SYSRET, S.RETU, SYSFRET THESE ARE SEPARATE ENTRIES
*   ON THE ONLY CODE WHICH RETURNS TO THE USER.
*   TOSPROC IS THE GENERAL PROCESSOR OF THE TOP OF
*   STACK AND HAS SEARCHES FOR INTERRUPTS AND F-RETURN
*   HANDLING IN IT. THE OTHER TWO ENTRIES ARE
*   CONCESSIONS TO EFFICIENCY AND ARE USED FOR
*   NORMAL RETURNS FROM SYSTEM ACTIONS AND FOR NORMAL
*   RETURNS FROM THE SWAPPER RESPECTIVELY.
*   ERRPROC, E.ERROR THE ERROR HANDLING CODE
*   NEWTOS CREATES A NEW TOP OF STACK (ONLY CODE WHICH DOES)
*   PUTRETP MOVE RETURN PARAMETERS TO TARGET SUBPROCESS CORE
*
* INTERNAL SUBROUTINES HEREIN
*   OPINTER FETCH PARAMETERS FOR ONE ORDER OF AN OPERATION
*   INTO AN AREA AND PARMBUFF
*

```

0
0
0
0
0

```

RECS      MACRO      A
*          RE         A
          RJ         =XF.ECS
RECS      ENDM
WECS      MACRO      A
*          WE         A
          RJ         =XF.ECS
WECS      ENDM
ECSMAC    XTEXT
CBLOCK    MICRO      I:*/ESENT/*
TYPES     XTEXT
INTSYS    XTEXT
PROCSYM   XTEXT
ERRNUMC   XTEXT

```

```
*
* THIS CODE PROCESSES THE TOP OF STACK. IT IS THE ONLY METHOD
* WHEREBY THE SYSTEM RELINQUISHES CONTROL TO A PROCESS. IT IS THUS
* USED AS THE LAST STEP IN
* A) RETURN FROM A SYSTEM ACTION, INCLUDING USER CALLED
* RETURN ACTIONS
* B) SUBPROCESS CALLS, INCLUDING ERROR AND INTERRUPT CALLS
* C) TO RESTART A PROCESS AFTER SWAPIN
```

```
* IT ASSUMES THAT THE ENVIRONMENT OF THE TOP OF STACK HAS BEEN
* ESTABLISHED IN THE WAY THE ROUTINE ENVIORN WOULD DO IT.
* THE INFORMATION IN THE TOP OF STACK IS USED TO EITHER RETURN
* CONTROL TO THE CURRENT SUBPROCESS, INITIATE ERROR PROCESSING FOR
* ANY OF A NUMBER OF INFRACTIONS, FIRE UP ANY SUBPROCESS WITH A
* PENDING INTERRUPT WHICH HAS PRIORITY OVER THE CURRENT SUBPROCESS,
* HANDLE ANOTHER ORDER OR F-RETURN IF F-RETURN PROCESSING IS GOING
* ON, ETC.
```

```
* AT ENTRY
* B1 = PROCESS SCRATCH AREA
* P-STACK POINTS TO TOP OF STACK
* ENVIORN OF TOP OF STACK IS ESTABLISHED
```

```
* THE FUNCTIONS OF THE SEPARATE ENTRIES ARE
* TOSPROC GENERAL RETURN - LOOK FOR PRIORITY INTERRUPTS,
* BUT DO NOT RETURN PARAMETERS
* SYSRET GARDEN-VARIETY RETURN FROM SYSTEM ACTION -
* DO NOT SEARCH FOR PRIORITY INTERRUPT, DO NOT
* RETURN PARAMETERS. ALSO USED FOR SYSTEM F-RETURN
* AFTER INCREMENTING F-RETURN COUNT IN TOP OF STACK
* S.RETU USED BY THE SWAPPER WHEN IT KNOWS NOTHING
* SIGNIFICANT HAS HAPPENED TO THE PROCESS SINCE
* IT WAS LAST RUNNING
* SYSFRET THIS ENTRY DOES AN F-RETURN FROM A SYSTEM ACTION
```

```
*
* ENTRY TOSPROC, SYSRET, S.RETU, SYSFRET
0 5111000164 SYSFRET SA1 B1+P-STACK . GET STACK POINTER
      53111 SA7 B1+X1 . GET TOP OF STACK
1 7160000001 SX6 I
      20624 LX6 36 . POSITION I AT F-RETURN COUNT
      36661 IX6 X6+X1 . INCREMENT F-RETURN COUNT
2 54610 SA6 A1
      0400000007 + EQ SYSRET . JUMP TO NORMAL SYSRET
3 6150000001 TOSPROC SB5 I . DONT RETURN PARAMETERS
      6170000005 + SB7 TOSPR3 . RETURN LINK
4 0400000000 + EQ =XPRINT . LOOK FOR PRIORITY INTERRUPT PENDING
5 0420000010 + TOSPR3 EQ B2:B0, TOSPR2 . NONE WERE FOUND
*
* GO SET UP AND PROCESS A NEW TOP OF STACK FOR THE INTERRUPT SUBP
*
6 6170000003 SB3 P-INTOFF . TYPE OF CALL = INTERRUPT
      6130000002 SB7 TOSPROC . RETLIN
      04000000516 + EQ =XNEWTOS . B2 HAS RELEVANT SUBP DESCRIPTOR
```

```

7 6150000001 SYSRET SB5 I . DONT RETURN PARAMETERS
*
* GET TOS AND BRANCH ON PC0
*
10 5111000164 53111 TOSPR2 SA1 B1+P.STACK . STACK POINTER
    SA1 R1+X1 . WORD n OF TOS
*
* A1 REMAINS UNCHANGED UNTIL VERY LATE IN THE GAME
* X1 REMAINS THE SAME, ALBEIT USUALLY SHIFTED
*
11 0931000032 + 20104 LX1 SF.PC01
    NG X1.TOSPR1 . JUMP IF ABOUT TO EXECUTE
12 0331000020 + 20151 LX1 SF.PC02-SF.PC01
    NG X1.TOSPR7 . JUMP IF ALMOST FINISHED
*
* WE ARE IN THE MIDDLE OF AN XJ
* FIRST WE RETURN PARAMETERS IF ANY ARE TO BE RETURNED
*
13 6160000015 + 0550000016 + NE B5,B0,TOSPR4 . NONE TO RETURN
    SB6 TOSPR5 . RETLINK
    SB7 TOSPROC . ERROR RETURN
14 0400000541 + EQ OUTRETS . GO PASS THE PARAMS
15 5111000164 53111 TOSPR5 SA1 B1+P.STACK . RETRIEVE STACK WORD
    SA1 R1+X1
    LX1 SF.PC02
*
* SEE IF F-RETURN LOGIC MUST BE INVOKED
*
16 20123 63210 TOSPR4 LX1 B+6-SF.PC02 . GET F-RET COUNT
    SB2 X1 . USED AGAIN MUCH LATER IF F-RET
    NE B2,B0,TOSPR6 . JUMP IF NON-ZERO F-RETURN COUNT
17 20151 LX1 B6+SF.PC02 . REPOSITION X1
*
* THE GUY AT THE TOS IS ALMOST FINISHED DOING AN XJ. IT REMAINS
* TO CALCULATE HIS NEW P-COUNTER FROM THE P-COUNTER IN THE STACK
* AND THE OFFSET IN THE XJ AND RETURN TO HIM (IF THE NEW P-COUNTER
* IS OK, THAT IS).
*
20 20167 5121000143 21244 TOSPR7 LX1 B0-SF.PC02
    SA2 R1+P.XPACK+1 . USER RA
    AX2 36
21 63610 53226 21263 SB6 X1 . P-COUNTER
    SA2 B6+X2 . FETCH XJ
    SB5 X2 . SAVE OFFSET
    AX2 51 . CHECK OUT ALLEGED XJ
22 7222777764 SX2 X2-013B
23 5141000144 0312000560 + 21444 21263 TOSPR8 NZ X2,ERR28 . XJ IS NOT AN XJ
    SA4 R1+P.XPACK+2 . USER FL
    AX4 36
    SX6 B6+B5
24 7266000001 37444 SX6 X6+1 . SET PROPOSED P-COUNTER
    IX4 X6-X4
25 0324000552 + PL X4,ERR15X . EXCEEDS FL

```

SYSTEM ENTRY / EXIT
TOP OF STACK PROCESSOR

COMPASS - VER 2.

11/15/71 22.19.07.

PAGE 5

0334000551 +
26 040000033 +

NG
EQ

X6,ERR15 . NEGATIVE
TOSPR2.5

```

27      TOSPR6   BSS      0
0      ECSSUB   SFRET.BUFA

*
* WERE INTO THE F-RETURN HASSE
* GET THE OPERATION AND SEE IF IT HAS ENIF ORDERS
*
L 450 20122      LX1      18      . GET A(IP LIST) REL USER RA
      5121000143 SA2      R1+P.XPACK+1  USER RA
      21244      AX2      36
L 451 63320      SB3      X2      . USER DA
      53113      SA1      X1+B3  . GET IPO
*
* A1 = PTR TO LAST INPUT PARAMETER PROCESSED FOR OPINTER
*
L 452 6160000453 SB3      B0      . ERROR MODIFIER FOR GETCAP
      64300      SB6      FRET3   . RETURN LINK
      9400000000 X EQ      =XGETCAP . GET CAP FOR OPERATION
L 453 54200      FRET3   SA2      A0      . FIRST WORD OF IPO CAPABILITY
      7222776100 SX2      X2-T.OPER
L 454 0312000516 NZ      X2.FRET17 . IPO NOT OPER
      50300000001 SA3      A0+1   . SECOND WORD OF CAP
L 455 73030      SX0      X3      . READ AND CHECK MOT
      5101000050 SA0      B1+P.SCR2
L 456 0110000001 RECS    I      . READ MOT
L 457 54200      SA2      A0
      43647      MX6      39
      13302      BX3      X3-X2
      11363      BX3      X6-X3
L 460 0313000517 NZ      X3.FRET18 . OPERATION GONE
      15026      BX0      =X6-X2  . ECS A(OPFR)
L 461 0110000002 RECS    I+PS.MASKI . READ HDR WORD AND TYPE MASK
L 462 54200      SA2      A0      . HEADER WORD
      20230      LX2      6+18
      63620      SB6      X2      . LENGTH OF FIRST ORDER
L 463 6170000001 SB7      I      . CONSTANT (USED BY OPINTER TOO)
      21252      AX2      6+36  . NUMBER OF ORDERS
      76620      SX6      B2      . F-RETURN COUNT
L 464 37362      IX3      X6-X2
      0323000513 PL      X3.FRET11 . JUMP IF NO MORE ORDERS
L 465 5161000057 SA6      B1+P.TEMP2  . SAVE F-RETURN COUNT
      13646      BX6      X6-X6
L 466 5161000060 SA6      B1+P.TEMP3  . INITIAL COUNT OF ORDERS PROCESSED
      7160000000 X SX6      =XPARMBUF . INITIALIZE POINTER TO ECS BUFFER
L 467 5161000063 SA6      B1+P.TEMP6  . FOR BLOCK PARAMETERS FOR OPINTER
      5151000143 SA5      B1+P.XPACK+1
L 470 21544      AX5      36
      73750      SX7      X5      . USER RA
      54557      SA5      A5+B7
L 471 36757      AX5      36
      5171000021 IX7      X5+X7  . USER FL, ABSOLUTE
      10600      SA7      B1+P.TEMP4  . SAVE CALLER ADDR SPACE LIMIT, ABS
L 472 5161000056 BX6      X0
      SA6      B1+P.TEMP1  . SAVE ECS A(OPFR)

```

```

      7130000002          SX3      I+PS.MASKL      . READ FIRST ORDER
L 473 36003              IX0      X0+X3
      56010              SA0      B1
      6151000170          SB5      B1+P.LOCALC      . SET UP FOR OPINTER
L 474 5131000162          SA3      B1+P.CLIST
      5141000065          SA6      B1+P.PARAM-1      . JUST INITIALIZING XA
L 475 53331              SA3      B1+X3
      63430              SB4      X3      . LENGTH OF LOCAL C-LIST

*
* THIS LOOP PROCESSES ORDERS UNTIL IT RUNS OUT OR GETS AN ERROR
*
L 476 0116000000          FRET5    RECS      B6      . READ NEXT ORDER
L 477 54200              SA2      A0
      0332000505          NG      X2,FRET7      . JUMP IF NO PARAMETERS
L 500 5131000061          SA3      B1+P.TEMP4      . CALLER ADDR SPACE LIMIT
      63220              SB2      X3
      54327              SA3      A2+B7      . GET NUMBER OF VISIBLE PARAMS
L 501 64610              SB6      A1      . PLUS CURRENT IP LIST ADDR
      63636              SB6      B6+X3
      0465000520          GE      B6,B2,FRET19      . IP LIST TOO BIG
L 502 20230              LX2      A+18
      63220              SB2      X2      . PTR TO FIRST PS DATA
L 503 54202              SB6      FRET7      . RETLINK
      5030000003          SA2      A0+B2      . PREFETCH PS DATA
L 504 0400000365          SA3      A0+3      . PREFETCH PS BYTES
L 505 5151000060          EQ      OPINTER      . PROCESS ORDER
      73727              SA5      B1+P.TEMP3      . ORDERS PROCESSED -1
L 506 5141000057          SX7      X5+B7      . ORDERS PROCESSED
      37527              SA4      B1+P.TEMP3      . F-RETURN COUNT
      54750              IX5      X4=X7
L 507 0335000111          SA7      AS      . INCREMENTED ORDER COUNT
      5141000002          NG      X5,UCALL3      . JUMP IF NECESSARY ORDERS NONE
L 510 20430              SA4      B1+2      . GET INFO ABOUT NEXT ORDER
      63640              LX4      A+18
      5151000056          SB6      X4      . LENGTH OF NEXT ORDER
L 511 20422              SA5      B1+P.TEMP1      . ECS A(OPERATION)
      73040              LX4      18
      34005              SX0      X4      . ORIGIN OF NEXT ORDER
      56010              IX0      X0+X5      . ECS A(THIS ORDER)
L 512 0400000476          SA0      B1
      EQ      FRET5      . PROCESS ANOTHER ORDER

*
* INSUFFICIENT ORDERS FOR F-RETURN COUNT
*
L 513 5111000164          FRET11  SA1      B1+P.STACK      . RESTORE A1,X1
      51111              SA1      B1+X1
L 514 6150000000          SB5      0      . OFFSET IS 0 FOR F-RETURN
      63610              SB6      X1      . KLUDGE F-COUNTER INTO B6
L 515 0400000023          EQ      TOSPRB      . JUMP INTO ALMOST FINISHED CASE

*
* HANDLE MISC ERRORS
*
L 516 6140000000          FRET17  ERROR      0,IP0

```

SYSTEM ENTRY / EXIT
TOP OF STACK PROCESSOR - F-RETURN HASSE

COMPASS - VER 2.
FSENT

11/15/71 22.19.07.

PAGE 8

L	517	6140000000	FRET18	ERROR	0,NOOP
L	520	06460	FRET19	ERROR	R6,BIGPT
L	521			ENDSUB	SFRET,BUFA

```

*
* THE CURRENT SUBP IS ABOUT TO EXECUTE THE INST INDICATED BY THE
* P-COUNTER IN THE STACK. SET THE P-COUNTER IN THE XPACK, UPDATE THE
* CLOCKS: SWAPOUT IF QUANTUM UP ELSE XJ TO USER.
*
32 21104          TOSPR1  AX1      SF.PCQI
      73610        SX6      X1
33 5121000142    TOSPR2.5 SA2      B1+P.XPACK . GET P-COUNTER FROM TOS
      42720        MX7      60-36 . ELIMINATE OLD P-COUNTER
      15227        BX6      -X7*X2 . FROM XPACK
34 20644          LX6      36
      12626        BX6      X2+X6
      54620        SA6      A2 . PUT IN NEW P-COUNTER
35 6170000036    SB7      *+1
      0400000000  EQ      =X2.LSYS . SIGNAL LEAVING SYSTEM, GET X1=S.CHARG
36 5120000000    SA2      S.OLDTIM
      5131000140  SA3      B1+P.SYSTIM . USER SYSTEM CLOCK
37 10611         BX6      X1 . UPDATE LAST CHARGE TIME
      54620        SA6      A2
      37512        IX5      X1-X2 . ELAPSED TIME
      36753        IX7      X5+XJ . ADVANCE USER SYSTEM CLOCK
40 5140000000    SA4      S.SYSTIM . SYSTEMSYSTEM CLOCK
      54720        SA7      A3
      36654        IX6      X5+X4
41 54640         SA6      A4 . UPDATE SYSTEM SYSTEM TIME
      7170000000  SX7      0 .**** DISPLAY ****
42 5170000002    SA7      S.FNFLO .**** USER ****
      5140000000  SAA     S.QUANT
43 0324000000    PL      X4,BXSWAPOUT . SWAPOUT IF QUANTUM GONE
44 0131000142    XJ      B1+P.XPACK . RETURN TO USER
S.RETU

```

45 7170000002

5170000002

```
*
* THE USER COMES BACK HERE (NATURALLY)
*
* USERCAL SX7      5 ***** DISPLAY *****
*           SA7      5 ***** SYSTEM *****
```

ENTRY FROM USER INITIATED CEJ IS TO THIS ROUTINE

BI = ORIGIN OF THE PROCESS SCRATCH AREA

THE CEJ IS DECODED TO FIND AN INPUT PARAMETER LIST

THE FIRST ELEMENT OF THE INPUT PARAMETER LIST (IPO) IS INTERPRETED AS A CAPABILITY INDEX FOR AN OPERATION

THE OPERATION (IPO) IS THEN INTERPRETED TO FORM AN ACTUAL PARAMETER LIST FOR THE SYSTEM CALL

PARAMETERS ARE DRAWN FROM EITHER THE OPERATION (IPO) OR FROM THE INPUT PARAMETER LIST AS DIRECTED BY THE OPERATION

CAPABILITY PARAMETERS SUPPLIED BY THE USER AS INDICES IN HIS FULL C-LIST (IN THE INPUT PARAMETER LIST)

ARE CHECKED FOR THE CORRECT TYPE AND REQUIRED OPTIONS. FINALLY, A TRANSFER IS MADE TO THE CORRECT SYSTEM ROUTINE TO PROCESS THE USER REQUEST

SUBROUTINES CALLED: .OBINTER, GETCAP

```
EXT GETCAP, SUBCAL, SUBJUM, E.MOT
EXT S.QUANT, S.CHARG, S.SYSTEM, S.USRTM, S.SWPTM, S.OLDTM
```

UPDATE CLOCKS

46 6170000047

0400000000 X

```
SB7 *+I
EQ =XZ, RAYS . SIGNAL ENTERING SYSTEM, GET XI=S.CHARG
```

47 5120000000 X

5131000137

```
SA2 S.OLDTM
SA3 BI+P.USRTM USER TOTAL USER TIME
```

50 10611

54620

37512

36753

```
BX6 XI
SA6 AN UPDATE S.OLDTM
IX5 XI-X2
IX7 X3+X3
```

51 5140000000 X

54720

36654

```
SA4 S.USRTM SYSTEM TOTAL USER TIME
SA7 A3
```

52 5064000000

```
IX6 X3+X4
SA6 A4+0
```

53 6170000001

54217

21144

```
SA1 BI+P.XPACK FETCH USER P-COUNTER
SB7 I
SA2 AI+B7 FETCH USER RA
```

```
AX1 36
```

54	6221777776		SB2	X1-1	. A(CEJ)	
		21244	AX2	36		
		53422	SA4	X2+B2	FETCH THE WORD CONTAINING THE CEJ	
55	21436		AX4	30		
		63440	SB4	X4	LOWER 18 BITS OF THE CEJ INSTR	
					NOTE..THE USER CALL CEJ MUST BE	
					IN THE HIGH ORDER PART OF THE WORD	
		0640000060 *	GE	B4,B0,UCALL1	JP IF IP LIST ADDR W/ CEJ	
56	7160000017		SX6	17B	LOW ORDER CEJ SPECIFIES A REGISTER	
		15444	BX4	-X4*X6	WHICH CONTAINS THE IP LIST ADDR	
		63340	SB3	X4		
57	54413		SA4	A1+B3	FETCH THE REGISTER FOR XPACK	
		63440	SB4	X4	18 BITS OF REGISTER	
					* SET UP FIRST WORD OF TOP OF STACK	
60	76640		UCALL1	SX6	B4	. PROPOSED A(IPLIST)
		43322	MX3	18		. GET RID OF XTRA SIGN BITS
		15643	BX6	-X3*X6		
		20622	LX6	18		
61	76320		SX3	92	. A(CEJ) FOR P-COUNTER	
		43704	MX7	4	. MASK TO PRESERVE EXISTING FLAGS	
		12646	BX6	X3+X6		
		54327	SA3	A2+B7	. FETCH USER FL	
62	21344		AX3	36		
		63230	SB2	X3	. USER FL	
		5131000164	SA3	B1+P,STACK	. TOP OF STACK POINTER	
63	53331		SA3	B1+X3	. TOP OF STACK	
		11373	BX3	X7*X3	. PRESERVE STACK FLAGS	
		12643	BX6	X6+X3		
		54630	SA6	A3	SET NEW TOP OF STACK WORD	
64	0740000553 *		LT	B4,B0,ERRN1	. ERROR - NEG IPLIST	
		0449000554 *	LE	B2,B4,ERRN1X	. ERROR - IPLIST EXCEEDS FL	
65	0160000067 *		SB6	UCALL2		
		53124	SA1	X2+B4	USE RA + IP LIST ADDR	
		66300	SB3	B0	.. ERROR MODIFIER FOR GETCAP	
66	0200000000 X		CP	GETCAP	GET CAPABILITY	
67	54200		UCALL3	SA2	A0	LOAD THE CAPABILITY INTO REGISTERS
		7222776160	SX2	X2-T,OVER	MAKE SURE IPO IS CAP FOR AN OPERATION	
		54327	SA3	A2+B7		
70	0312000545 *		NZ	X2,ERRN3	ERROR- IPO NOT A CAP FOR AN OPERATION	
		73030	SX0	X3	NOT INDEX FOR OPERATION	
71	5101000050		SA0	B1+P,SCR2		
72	0110000001		RECS	1	READ MOT ENTRY FROM ECS	
73	43647		MX6	39	TEST MOT ENTRY TO BE SURE IT IS	
		5020000000	SA2	A0+0	STILL THE CORRECT ENTRY FOR THE OPER	
		13332	BX3	X3=X2		
74	11336		BX3	X3*X6		
		0313000547 *	NZ	X3,ERR40	ERROR - OPERATION NOT IN MOT	
		15026	BX0	-X6*X2		
75	0110000002		RECS	1+PS,MASKL	READ HEADER AND MASK BITS	

115	10622		*	UCALL14	BX6	X2	CLASS CODE AND INDEX (PRE-FETCHED)
	20473				LX4	59	
		5161000125			SA6	B1+P.PARAMC	STORE AT END OF PARAM AREA
116	43030				MX0	6+18	EXTRACT AND STORE NUM OF PARAM AND
	15640				BX6	X0*X4	NUM OF CAPABILITY PARAM
		5151000001			SA5	B1+1	
117	21522				AX5	18	
	73550				SX5	X5	
		20514			LX5	36	
		12665			BX6	X6+X5	ADD ACTUAL PARAM SIZE
120	55667				SA6	A6-B7	
			*				
			*	IFNE		PS.MASKL*1.1	DOO ERROR IF PS.MASKL .NE. 1
		5121000001			SA2	B1+P.SCR2+1	FETCH PARAM TYPE BIT MASK
		10622			BX6	X2	
121	55667				SA6	A6-B7	PASS PARAM TYPE BIT MASK
			*				
		5151000002			SA5	B1+2	GET ACTION CODE
122	76660				SB6	X5	
		63650			SX6	B6	GO SET ACTION TYPE
123	6170000124		*		SB7	*+1	
		0400000000			EQ	XZ.ASYS	
124	0460000125		*		JP	B6+ACTIONL	GO DO ACTION

EACT	CPROC	.. CREATE A PROCESS	ECSACT	1
EACT	CREFIL	.. CREATE A FILE	ECSACT	1
CACT	CREBLK		ECSACT	1
CACT	WRFILE	.. WRITE FILE	ECSACT	1
CACT	REFILE	.. READ FILE	ECSACT	1
EACT	MKEC	.. MAKE EVENT CHANNEL	ECSACT	1
CACT	EVENT	.. SEND AN EVENT	ECSACT	1
CACT	HANG	.. GET AN EVENT OR HANG	ECSACT	1
EACT	CRECC	.. CREATE A CLASS CODE	ECSACT	1
EACT	NWIMP	.. CREATE NEW TEMP PART OF CLASS CODE	ECSACT	1
CACT	SREG	.. SAVE REGISTERS	ECSACT	1
CACT	RREG	.. RESTORE REGISTERS	ECSACT	1
EACT	MKSUBP	.. MAKE A SUBPROCESS	ECSACT	1
EACT	DSCAP	.. DISPLAY A CAPABILITY	ECSACT	1
EACT	MOVEC	.. MOVE CAP WITHIN FULL CLIST; MASK OPTION	ECSACT	1
EACT	CAPIN	.. MOVE A CAPABILITY INTO FULL CLIST	ECSACT	1
EACT	CAPOU	.. MOVE A CAPABILITY OUT OF FULL CLIST	ECSACT	1
CACT	CALSUB	.. CALL A SUBPROCESS	ECSACT	1
CACT	JUMSUB	.. JUMP CALL	ECSACT	1
CACT	RETURN	.. RETURN	ECSACT	1
EACT	FRETURN	.. RETURN	ECSACT	1
EACT	ESMGEN	.. NEW ERROR SELECTION MASK (ANY SUBP)	ECSACT	1
EACT	ESMLOC	.. NEW ERROR SELECTION MASK (LOCAL SUBP)	ECSACT	1
EACT	MKOPER	.. MAKE AN OPERATION	ECSACT	1
PARAMETER SPECIFICATION CHANGES TO OPERATIONS			ECSACT	1
CACT	USRDAT	.. PS TO USER DATUM	ECSACT	1
CACT	FIXDAT	.. PS TO FIXED DATUM	ECSACT	1
CACT	USRCAP	.. PS TO USER CAPABILITY	ECSACT	1
CACT	FIXCAP	.. PS TO FIXED CAPABILITY	ECSACT	1
CACT	ANYCAP	.. PS TO ANY CAPABILITY	ECSACT	1
CACT	BLKDAT	.. PS TO BLOCK DATA	ECSACT	1
CACT	BLKCAP	.. PS TO BLOCK CAPABILITY	ECSACT	1
CACT	ADDOPT	.. ADD OPTION BITS	ECSACT	1
EACT	COPYOP	.. COPY AN OPERATION	ECSACT	1
CACT	CHKBLK		ECSACT	1
CACT	DELBLK		ECSACT	1
EACT	DELFIL	.. DELETE A FILE	ECSACT	1
EACT	REDSHP	.. READ SHAPE NOS.	ECSACT	1

* CACT	MAPZRO	.. ZERO A MAP ENTRY	ECSACT
CACT	MPCHRO	.. CHANGE MAP ENTRY (R-O)	ECSACT
CACT	MPCHRW	.. CHANGE MAP ENTRY (R/W)	ECSACT
CACT	MOVBLK	.. MOVE A FILE BLOCK	ECSACT
* EACT	DISMAP	.. DISPLAY MAP	ECSACT
EACT	JPRET	.. JUMP RETURN	ECSACT
EACT	PRINT		ECSACT
EACT	NEWUN	.. CHANGE UNIQUE NAMES	ECSACT
EACT	DISPST	.. DISPLAY STACK	ECSACT
EACT	DISSEN	.. DISPLAY STACK ENTRY	ECSACT
EACT	DSPMAP	.. DISPLAY FULL MAP ENTRY	ECSACT
EACT	DELC	.. DELETE A CLIST	ECSACT
EACT	ADDORD	.. ADD AN ORDER TO AN OPERATION	ECSACT
EACT	CCCLOA		ECSACT
EACT	DONATE	.. TRANSFER BETWEEN ALLOCATION BLOCKS	ECSACT
EACT	CRALBK	.. CREATE ALLOCATION BLOCK	ECSACT
EACT	MODPC	.. MODIFY COUNTER OF STACK ENTRY	ECSACT
EACT	DLPRO	.. DESTROY A PROCESS	ECSACT
* CACT	GETEVT	.. GET AN EVENT OR F-RETURN	ECSACT
* EACT	DELAB	.. DESTROY AN ALLOCATION BLOCK	ECSACT
EACT	DELSUB	.. DELETE A SUBPROCESS	ECSACT
EACT	DPROD	.. DISPLAY A PROCESS	ECSACT
* CACT	CLDAD	.. CLEAR DIRECT ACCESS TO ECS	ECSACT
CACT	SETDAD	.. SET DIRECT ACCESS TO ECS	ECSACT
EACT	SETIIB	.. SET INTERRUPT INHIBIT BIT	ECSACT
EACT	CLRIB	.. CLEAR INTERRUPT INHIBIT BIT	ECSACT
CACT	MGETH	.. GET EVENT FROM LIST OFCHNNLS/HANG	ECSACT
CACT	MGETF	.. GET EVENT FROM LIST OFCHNNLS/F-RETURN	ECSACT
* EACT	DELEC	.. DESTROY EVENT CHANNEL	ECSACT
* CACT	DCLOX	.. DISPLAY CLOCK TIMES IN USER CORE	ECSACT
EACT	QSPAB	.. DISPLAY ALLOCATION BLOCK	ECSACT
EACT	TIMDT	.. OPERATOR TIME AND DATE	ECSACT
EACT	USRER	.. USER INITIATED ERROR	ECSACT
* EACT	RETPAR	.. RETURN WITH PARAMETERS	ECSACT
EACT	DISPOP	.. DISPLAY AN OPERATION	ECSACT
EACT	CAGEN	.. MAKE CAP CREATING AUTHORIZATION	ECSACT
EACT	CGEN	.. CREATE A CAPABILITY	ECSACT
EACT	DSPSP	.. DISPLAY SUBPROCESS DESCRIPTOR	ECSACT
* CACT	TRDB	.. TEST A RESET DIRTY BIT	ECSACT
EACT	INCHR	.. INCREMENT CHARGE RATE	ECSACT
EACT	DSPOB	.. DISPLAY OBJECT	ECSACT

|||||

SYSTEM ENTRY / EXIT
SYSTEM ACTION JUMP TABLE

COMPASS - VER 2.

11/15/71 52.19.12.

PAGE 18

273	0200000000	x
276	0200000000	x
202	0200000000	x
203	0200000000	y
317	0200000000	x
326	0200000000	x
327	0200000000	y
330	0200000000	y
163	0200000000	x

PUTACT	J.MGETF*MEVHNGF
PUTACT	J.DCLOX*DSPCLOX
PUTACT	J.BLKDAT*BDAT
PUTACT	J.BLKCAP*RCAP
PUTACT	J.TRDB*TRDB
PUTACT	J.CHMPRW*CHMPRW
PUTACT	J.CHMPRO*CHMPRO
PUTACT	J.DSCLX*DSPSCLX
PUTACT	J.JUMSUB*JUMPCAL

		*		PS ACTION 1 ... GO TO NEXT PS BYTE WORD
372	54337	*	OPINTER4 SA3	A3+87
	0200000366 +	*	JP	FETCH NEXT WORD FULL OF PS BYTES
373		*	BSS	I
		*		PS ACTION 2 ... #NONE# PARAM SPECIFICATION
374	0200000544 +	*	OPINTER5 JP	ERR26
375		*	BSS	I
		*		PS ACTION 3 ... USER SUPPLIED DATUM
376	54117	*	OPINTER6 SA1	A1+87
	10611	*	BX6	FETCH DATUM FROM IP LIST
	5044000001	*	SA6	A6+1
377	0200000366 +	*	JP	STORE IN ACTUAL PARAM AREA
		*		OPINTER1
		*		PS ACTION 4 ... FIXED DATUM
400	10622	*	OPINTER7 BX6	X2
	5022000001	*	SA2	DATUM WAS PREFETCHED
	54667	*	SA6	A2+1
401	0200000366 +	*	JP	PREFETCH NEXT PS DATA WORD
		*		A6+87
		*		STORE DATUM IN ACTUAL PARAM AREA
		*		OPINTER1
		*		PS ACTION 5 ... FIXED CAPABILITY
402	10622	*	OPINTER8 BX6	X2
	54227	*	SA2	PREFETCHED PS DATUM
	54667	*	SA6	A2+87
	10622	*	BX6	A6+87
403	54227	*	SA2	X2
	54667	*	SA6	A2+87
	0200000366 +	*	JP	A6+87
		*		STORE AND WD OF CAPABILITY
		*		OPINTER1
		*		PS ACTION 6 ... BLOCK PARAMETER
404	54117	*	OPINTR13 SA1	A1+87
	0322000443 +	*	PL	BLOCK CONTROL FROM IP LIST
405	0200000457 +	*	JP	X2, OPINTR16
		*		OPINTR17
		*		JP ON DATA BLOCK
		*		PROCESS BLOCK CAPABILITIES
		*		PS ACTION 7 ... USER SUPPLIED CAPABILITY
406	54117	*	OPINTER9 SA1	A1+87
	63610	*	SB6	X1
	0760000557 +	*	LT	B6, B0, ERR22X
407	0664000414 +	*	LE	B4, B6, OPINTR11
	66626	*	SB6	B6+86
	56456	*	SA4	B5+86
410	0331000430 +	*	NG	X1, OPINTR15
		*		FETCH 1ST WD OF CAPABILITY
		*		JP IF INDIRECT REFERENCE
411	15524	*	OPINTR10 BX5	-X4*X2
	10644	*	BX6	X4
		*		TEST CAP FOR TYPE AND OPTIONS

412	0315000546 +	54667 54447	SA6 SA4 NZ	A6+B7 A4+B7 X5,ERR25	STORE 1ST WD OF CAPABILITY FETCH 2ND WD OF CAPABILITY JP IF TYPE OR OPTION ERROR
413	54227	15644 54667	BX6 SA6 SA2 JP	X4 A6+B7 A2+B7 OPINTR1	STORE 2ND WD OF CAPABILITY PRE-FETCH NEXT PS DATUM
	0200000346 +				
414	5141000162	7244000002	OPINTR11 SA4 SX4	B1+P,C-LIST X4+2	ORIG OF C-LIST TABLE LENGTH OF 2ND C-LIST
415	53441	67664	SA4 SB6	X4+B1 B6-B4	
416	63340	0304000555 +	ZR SB3	X4,ERR22 X4	ERROR-CAP INDEX TOO LARGE
	0763000421 +		LT	B6,B3,OPINTR14	JP IF INDEX IN 2ND C-LIST
417	5044000002	67663	OPINTR12 SA4 SB6	A4+2 B6-B3	LOOP TILL PROPER C-LIST
420	0304000555 +	63340	SB3 ZR	X4 X4,ERR22	ERROR-CAP INDEX TOO LARGE
	0663000417 +		GE	B6,B3,OPINTR12	JP IF NOT FOUND C-LIST YET
421	5044000001	73040 54067	OPINTR14 SA4 SX0	A4+1 X4	GET UNIQUE NAME AND NOT INDEX OF THE C-LIST
422	0110000001		SA0 RECS	A6+B7 I	READ NOT ENTRY
423	54500	13445	SA5 BX4	A0 X4-X5	TEST NOT ENTRY
	47647		MX6 BX4	39 X6*X5	
424	0314000550 +	11446	NZ	X4,ERR41	ERROR- C-LIST GONE
	15056		BX0	X6*X5	ECS ADDR OF C-LIST
	76460		SX4	B6	C-LIST INDEX
425	20401	7244000001	LX4 SX4	I X4+1	
	36040		IX6 RECS	X4+X0 2	READ CAPABILITY
427	54400	0321000411 +	SA4 PL	A0 X1,OPINTR10	FETCH 1ST WD OF CAPABILITY JP IF NOT INDIRECT REFERENCE
430	7264776400	0314000556 +	OPINTR15 SX6 NZ	X4-T,C-LIST X6,ERR27	TEST FOR CAPABILITY FOR C-LIST JP IF NOT C-LIST CAPABILITY
431	54447	73040	SA4 SX0	A4+B7 X4	
	21126		AX1	30	SHIFT TO INDIRECT INDEX
	54067		SA0	A6+B7	

432 0110000001
 433 73110
 0331000557 +
 43047
 434 54500
 13445
 17440
 15050
 435 0110000001
 436 0314000550 +
 5050000000
 437 37515
 0325000555 +
 20101
 440 7211000001
 34001
 441 0110000002
 442 54400
 0200000471 +

RECS
 SX1
 NG
 MX0
 SA5
 BX4
 BX4
 BX0
 RECS
 NZ
 SA5
 IX5
 PL
 LX1
 SX1
 IX0
 RECS
 SA4
 JP

READ MOT ENTRY
 X1
 X1,ERR22X
 29
 A0
 X4-X5
 X4*X0
 X0*X5
 X4,ERR41
 A0+0
 X1-X5
 X5,ERR22
 1
 X1+1
 X0+X1
 2
 A0
 OPINTR10

ERROR-INDIRECT INDEX NEGATIVE
 TEST MOT ENTRY
 READ LENGTH OF C-LIST
 ERROR- C-LIST GONE FROM MOT
 ERROR-INDEX TOO LARGE
 READ CAPABILITY
 FETCH 1ST WD OF CAPABILITY

*
 *
 *
 *
 *
 *

***** BLOCK DATA PARAMTER *****

443 73410
 20136
 0334000561 +
 444 63210
 63320
 0720000562 +
 445 5151000063
 0730000563 +
 446 10055
 73752
 54700
 20522
 447 43673
 54667
 76600
 12665

OPINTR16
 SX4
 LX1
 NG
 SB2
 SB3
 LT
 SA5
 GT
 BX0
 SX7
 SA7
 LX5
 MX6
 SA6
 SX6
 BX6

X1
 00-30
 X4,ERR30
 X1
 X2
 B2,B0,ERR32
 B1+P,TEMP6
 B2,B3,ERR33
 X5
 X5+B2
 B5
 B8
 B9
 A6+B7
 B2
 X6+X5

X4 * PTR TO DATA BLOCK
 ERROR: NEGATIVE POINTER
 B2 * COUNT
 MAX COUNT FOR OPERATION
 ERROR: NEGATIVE COUNT
 ERROR: USER COUNT TOO LARGE
 BUFFER ADDRESS
 INCR BUFFER ADDRESS
 1ST WD OF ACTUAL PARAM = -1
 CONSTRUCT ACT PARAM CONTROL

*
 *
 *
 *
 *

THE ABC

 * 0 * ECS BUFR * ACT. NUM * MAX NUM *
 * ADDR * PASSED * WDS *

450 20622
 76730
 12667

LX6
 SX7
 BX6

B8
 B3
 X6+X7

MAX COUNT

```

54667
451 5151000144
      21544
      73742
452 37557
      0335000544 +
453 5151000143
      21544
      36554
454 53050
455 0122000000
456 54227
      43771
      0200000366 +

```

```

SA6      A6+B7
SA5      B1+P.XPACK+2
AX5      36
SX7      X4+B2
IX5      X5-X7
NG       X5,ERR34
SA5      B1+P.XPACK+1
AX5      36
IX5      X5+X4
SA0      X5
WECS     B2
SA2      A2+B7
MX7      57
JP       OPINTER1

```

```

FL
ERROR: BLOCK EXTENDS PAST FL
DA
WRITE BLOCK DATA TO BUFFER
NEXT DS DATUM
RESTORE MASK
PROCEED TO NEXT PARAM

```

***** BLOCK CAPABILITY PARAMETER *****

```

457 63210
      20136
      5151000063
460 73110
      0720000557 +
      10055
461 0331000562 +
      43673
      54667
462 20522

```

```

OPINTR17 SB2      X1
          LX1      40-30
          SA5     B1+P.TEMP
          SX7     X1
          LT      B2,B0,ERR31
          BX0     X5
          NG      X1,ERR2
          MX6     B0
          SA6     A6+B7
          LX5     B0

```

```

B2 + STARTING INDEX
BUFFER ADDRESS
CAPABILITY COUNT
ERROR: NEGATIVE CAP INDEX
ERROR: NEGATIVE COUNT
1ST WD ACTUAL PARAM + -1
FORM APC ==> 2ND WD OF ACTUAL PARAM

```

THE ACTUAL PARAMETER CONTROL FOR BLOCK CAPABILITY

```

*****
* 1 * ECS BUFR * ACT. NUM * MAX. NUM *
*   ADD   * CAPS   * CAPS   *
*****

```

```

12651
      20652
      12662
463 54667
      36701
      34771
      54750
464 73220
      37721
      0337000563 +

```

```

BX6      X5+X1
LX6      B0
BX6      X5+X2
SA6      A6+B7
IX7      X0+X1
IX7      X7+X1
SA7      A5
SX2      X2
IX7      X2-X1
NG       X7,ERR33

```

```

ADD IN MAX COUNT AND CAP FLAG
INCR BUFFER ADDRESS
MAX CAP COUNT FROM OPER
USER WANTS TO PASS TOO MUCH

```

465	10200		BX2	X0	Y2 P BUFFER ADDR
		0624000474 *	GE	R2,B4,OPINTR19	JP IF NOT IN LOCAL C-LIST
					XFER CAPABILITIES FROM LOCAL C-LIST
		67342	SB3	B4-B2	FIND MIN(LENGTH-INDEX, COUNT)
466	76730		SX7	B3	
		37771	IX7	X7-X1	
		0337000470 *	NG	X7,OPINTR18	
467	63310		SB3	X1	
470	56052		SA0	B5+B2	B3 = COUNT TO XFER
		54002	SA0	A0+B2	ABS STARTING CM ADDR
		76730	SX7	B3	
		66333	SB3	B3+B3	
471	0123000000		WECS	B3	DECR COUNT
472	37117		IX1	X1-X7	INDEX + LENGTH LOC C-LIST
		66240	SB2	B4	JP IF DONE
		0307000515 *	ZR	X1,OPINTR23	INCR BUFFER ADDR
473	73203		SX2	X0+B3	
					B2 = C-LIST INDEX
					X1 = COUNT TO GO
					X2 = BUFFER ADDR
					FIND STARTING C-LIST
474	5141000162		OPINTR19	SA4	B1+P,C-LIST
		53441	SA4	B1+X4	
		67224	SB2	B2-B4	
475	5044000002		OPINTR20	SA4	A4+2
		63340	SB3	X4	LEN OF NEXT C-LIST
476	0723000500 *		LT	B2,B3,OPINTR21	JP IF GOT A C-LIST
		67223	SB2	B2-B3	
477	0314000475 *		NZ	X4,OPINTR20	
		0200000565 *	JP	ERR95	ERROR: BLOCK NOT WITHIN FULL C-LIST
					B2 = INDEX
					B3 = LENGTH OF C-LIST
					X1 = COUNT
					X2 = BUFFER ADDR
					A4 = FULL C-TABLE POINTER
					X5 = ECS ADDR OF C-LIST
					SET UP NEXT C-LIST
500	5054000001		OPINTR21	SA5	A4+1
		5107000052	IFNE	PS, MASKL, 7, 1	
			SA5	B1+P, SCR2+PS, MASKL+1	
501	73050		SX0	X5	
		43747	MX7	39	
		10655	BX6	X5	
502	0110000001		RECS	1	READ MOT
503	54500		SA5	A0	
		13665	BX6	X6-X5	
		11676	BX6	X7-X6	
		15557	BX5	-X7-X5	ECS ADDR OF C-LIST
504	0316000566 *		NZ	X6, ERD36	ERROR: C-LIST GONE
		43773	MX7	59	

```

37557 IX5 X5-X7
*
* MOVE CAPABILITIES
*
505 76720 OPINTR2 SX7 B2 INDEX
      20701 LX7 1
      34057 IX0 X5+X7
506 011000002 RECS 2 READ ONE CAPABILITY
507 10022 BX0 X2 BUFFER ADDR
      66227 SB2 B2+B7 INCR INDEX
      721177776 SX1 X1-1 DECR COUNT
510 012000002 WECS 2 WRITE ONE CAPABILITY TO BUFFER
511 722000002 SX2 X0+2 INCR BUFFER ADDR
*
*
512 0532000505 + ZR X1,OPINTR3 JP IF DONE (COUNT EXHAUSTED)
      0501000515 + NE B3,B2,OPINTR2 LOOP TO END OF C-LIST
*
* STEP TO NEXT C-LIST
*
513 63340 SA4 A4+2
      66200 SB3 X4 LENGTH NEXT C-LIST
      0430000565 + SB2 B0 INDEX IN NEXT C-LIST
514 0200000500 + EQ B3,B0,ERR35 ERROR:BLOCK NOT WITHIN FULL C-LIST
      JP OPINTR2
*
* ALL DONE
*
515 54227 OPINTR3 SA2 A2+B7 FETCH NEXT PS DATUM
      43771 MX7 B7 RESTORE MASK
      0200000366 + JP OPINTR1
*
*

```

```

*
* THIS IS THE ONLY ROUTINE WHICH GENERATES A NEW TOP OF STACK. IT
* IS USED FOR NORMAL SUBPROCESS CALL, ERROR CALL AND INTERRUPT CALL.
* THE FOLLOWING ARE DONE
*
* 1 - TEST FOR STACK TOO FULL FOR SUBP BEING CALLED (EXCEPT
*     ON ERROR CALL - I.E. ERROR HAS ALREADY CHECKED)
*
* 2 - DETERMINE FULL PATH
*
* 3 - ADJUST THE STACK PTRS AND MAKE THE NEW STACK ENTRY
*
* 4 - ESTABLISH THE ENVIORN FOR THE NEW TOS
*
* 5 - SET THE LOW CORE OF THE CALLED SUBP
*     A = 0 IF CALLED SUBP IS NOT ANCESTOR (PROPER)
*     OF OLD CURRENT SUBP
*     B = RA, MAP ORIGIN, C-LIST ORIGIN OF OLD CURRENT OTHERWISE
*
* 6 - PLACE PARAMETERS APPROPRIATE TO TYPE OF CALL IN LOW CORE

```

* AT ENTRY -

```

* B1 = A (PROCESS SCRATCH AREA)
* B2 = A (DESCRIPOR OF CALLED SUBP), REL B1
* B3 = ENTRY OFFSET
*     0 FOR NORMAL SUBPROCESS CALL
*     1 FOR ERROR CALL
*     2 FOR INTERRUPT CALL
*     3 FOR INITIAL CALL OF PROCESS (DAMN NUISANCE)
*
* B7 = RETLINK

```

* IN CASE B3 = 0 OR 1, THE PARAMETERS ARE EXPECTED IN THE ACTUAL PARAMETER AREA.

```

* CALLS ENVIORN, ERRPROC
* USES P.TEMP1 THRU P.TEMP7 DIRECTLY OR INDIRECTLY
* USER P.SCR2+0 THRU +5

```

```

* NEWTOS MAY DISCOVER ALL SORTS OF ERRORS, BUT IT MERRILY
* CHURNS ALONG MAKING NEW STACK ENTRIES AND WILL SUCCEED
* IN GETTING BACK TO B7 AFTER MAKING 1 OR MORE
* STACK ENTRIES UNLESS IT HAS TO DESCHEDULE THE PROCESS
* FOR ERROR ERROR OR KILL IT FOR STACK FULL.

```

* ENTRY NEWTOS

* SEE IF CALLED SUBP IS LEGAL WITH CURRENT STACK

516									
51									
650	66412								
		51140000h4							
			76670						
L	651	5161000054							
			20130						
L	652	5121000164							
			63710						
				63620					
L	653	0767000656							

```

NEWTOS
BSS
ECSSUB NEWTP.BUFR
SB4 B1+B2 . SUBP DESC. ABS
SA1 B4+SD.PTR . PTR WORD
SX6 B7 . SAVE RETLINK
SA6 B1+P.SCR2+4
LX1 6+16
SA2 B1+P.STACK . GET STACK POINTERS
SB7 X1 . MAX ALLOWED STACK PTR
SB6 X2 . CURRENT STACK PTR
GT B7,B6,NEWT1 . JUMP IF OK TO CALL

```

654 0460000656 6762777776
655 7170000006 7160000004
656 7160000002 34622 54620
657 5114000005 27744 67503
660 73115
661 20704 43701 20770 12117
662 5017777776 63670 21122
663 63510 49000 74790 66450
664 0452000671
665 0556000666 74050
666 66515 5115000004 20130
667 63510 0775000671

SB6 B3-P,ERROFF . OK IF ERROR CALL
EQ B6,B0,NEW1
SX6 F.SUBP . GENERATE STACK FULL ERROR
SX7 F.FULSTK
EQ NEWTERR
*
* SET UP NEW STACK ENTRY
*
NEWT1 SX6 2 . INCREMENT STACK PTR
IX6 X6+X2
SA6 A2
*
* CALCULATE P-COUNTER FROM ENTRY POINT AND OFFSET
*
SA1 B4+SD.ORIG . ENTRY POINT WORD
AX1 36
SB5 -B3 . OFFSET SUBTRACTS
SX1 X1+B5 . P-COUNTER
*
* SET INTERRUPT INHIBITED BIT IN STACK AND SET PCQ = ABOUT TO EXECUTE
*
MX7 1
LX7 60-SF,PCQ1 . ABOUT TO EXECUTE BIT
BX1 X1+X7
LX7 SF,PCQ1-SF.II . INTERRUPT INHIBIT BIT
BX7 X1+X7
SA7 B1+X6 . SET FIRST WORD OF NEW STACK
*
* DETERMINE NEW END OF PATH SUBPROCESS - SEARCH ANCESTORS OF
* OLD END OF PATH UP TO THE LEVEL OF THE CALLEE LOOKING FOR THE
* CALLEE.
*
SA1 A7-1 . GET OLD END OF PATH
SB6 X1 . OLD CURRENT
AX1 18
SB5 X1 . OLD END OF PATH
MX0 0 . LOW CORE INDICATOR ASSUMED 0
SX7 B2 . END OF PATH ASSUMED = CALLEE
SB4 B5 . SAVE OLD EOP JUST IN CASE
*
* B2 = CALLEE
* B5 = ANCESTOR OF OLD EOP NOW BEING INVESTIGATED
* B6 = OLD CURRENT
* B7 = LEVEL INDICATOR (STACK LIMIT) OF CALLEE
*
EQ B5,B2,NEW2 . TRIVIAL IF CALLEE = OLD EOP
NE B5,B6,NEW3.I . SKIP IF NOT OLD CURRENT
SX0 B5 . REMEMBER PASSING OLD CURRENT
NEWT3.SJ SB5 B1+B5 . ABS SUBP PTR
SA1 B5+SD.PTRs
LX1 6+18
SB5 X1 . GET LEVEL INDICATOR
GT B5,B7,NEW2 . WONT FIND HIM IF ALREADY ABOVE HIM

L	670	63510	20122	LX1 SB5 NE SX7	18 X1 B5, B2, NEWT3 B4	FATHER JUMP IF NOT CALLEE OOPS, OLD EOP IS NEW EOP
		0552000645	76740			
				* SET WORD 1 OF STACK		
L	671	20722	76620	NEWT2 LX7 SX6	18 B2	POSITION END OF PATH CALLEE
		12727				
L	672	5077000001	20044	SA7 LX0	A7+1 B6	SET SECOND WORD OF STACK POSITION CORE INDICATOR
		20622				
L	673	76730	12767	LX6 SX7	18 B3	POSITION CALLEE SAVE OFFSET
		12767				
L	674	5171000055	12767	BX7 BX7 SA7	X6+X7 X0+X7 B1+P, SCR2+5	SAVE WHILE CALLING ENVIORN
				* ESTABLISH ENVIORN FOR NEW TOP OF STACK		
				*		
L	675	5161000052	43600	MX6 SA6	0 B1+P, SCR2+2	ERROR FLAG TO 0
		6160000677				
L	676	6170000700	0400000000 X	SB6 SB7	NEWT4.M1 NEWT4	ERROR RETLIN RETLINK
		5171000053				
L	677	5161000052	13555	EQ SA6	ENVIORN B1+P, SCR2+2	SAVE CONFOUNDED ERROR FLAG
		5171000053				
L	700	6170000001	13666	SA7 SB7	B1+P, SCR2+3 1	
		13666				
L	701	13777	5110000000 X	BX5 BX6 BX7	X5+X5 X6+X6 X7+X7	DEFAULT VALUES FOR CELLS 5,4,3
		53610				
L	702	54667	54667	SA1 SA6	X5, USERA X1	CLEAR 0,1,2
		5117000055				
L	703	63310	21122	SA6 SA6	B1+P, SCR2+5 X1	RECOVER REGISTERS OFFSET
		63270				
L	704	0301000713	21122	AX1 SB2 AX1	X1 18 X1	CALLEE LOW CORE INDICATOR JUMP IF BATH DIDNT EXPAND
		63471				
		66521		* PREPARE CELL 3 WORD - ORIG OF CALLER IN ADDRESS SPACE OF CALLEE		
				*		
L	705	5114000007	5125000007	SB4 SB5 SA1	X1+B1 B2+B1 B4+SD, RAFL	ABS ADDR(CALLER) ABS ADDR(CALLEE)
		21122				
L	706	21122	21222	SA2 AX1 AX2	B5+SD, RAFL 18 18	CALLEE RA CALLERE RA
		63620				
L	707	77756	63510	SB6 SB5 SX7	X2 X1 B5-B6	DIFFERENCE FOR CELL 3

```

*
* PREPARE CELL 4 WORD - ORIG OF CALLER C-LIST IN FULL C-LIST OF CALLEE
*
L 710 502277775 5114000005 63620 63510
      SA1      B4+SD.ORIG
      SA2      A2-SD.RAFL+SD.ORIG
      SB6      X2      . DIFFERENCE OF C-LIST ORIGINS
      SB5      X1
      SX6      B5-B6
*
* PREPARE CELL 5 WORD - ORIG OF CALLEE MAP IN FULL MAP OF CALLEE
*
L 712 63510 21122 21222 63620 77556
      AX1      I8
      AX2      I8
      SB6      X2      . DIFFERENCE OF MAP ORIGINS
      SB5      X1
      SX5      B5-B6
*
* SET CELLS 3,4,5
*
L 713 54767 54677 10655 54667
      NEWT5    SA7      A6+B7      . SET CELL 3
              SA6      A7+B7      . SET CELL 4
              SX5      X5
              SA6      A6+B7      SET CELL 5
*
* PASS CALL PARAMETERS ACCORDIN TO OFFSET
*
L 714 0430000745 EQ      B3,B0,SUBPC7      . JUMP IF NORMAL CALL
L 715 0536000723 SB6      B.EFFOFF
              NE      B3,B6,NEWT7      . JUMP IF NOT ERROR CALL
*
* ERROR CALL
*
L 716 10622 5121000066 54667 54357 10633
      SA2      B1+P,PARAM
      SX6      X2
      SA6      A6+B7
      SA3      A2+B7
      SX6      X3
      SA6      A6+B7
      SA1      B1+P,SCR2+2
      NZ      X1,NEWT8.4      . EXIT IF NO ENVIRON ERROR
      SX4      X2-E.PROC      . IF NOT PROCESS ERROR CLASS, NORMAL
      NZ      X4,NEWT8.5      PROCESSING
      SX4      X3-E.MAPER      . ONLY A PURE MAP SCREW-UP
      NZ      X4,NEWT8.5      INTERESTS US HERE
      EQ      NEWT8.4      . IF A MAP ERROR REPEATS, CALL HIM AND
                          . HOPE FOR THE BEST
*
* INTERRUPT CALL
*
L 723 6160000002 0536000735
      NEWT7    SB6      B.INTOFF
              NE      B3,B6,NEWT9      . JUMP IF NOT INTERRUPT CALL
*
* INTERRUPT CALL
*
L 724 5121000164 SA2      B1+P,STACK      . SET INTERRUPTED BIT IN

```

```

L 725 53221 722777775
      43701
      28773 12772
L 726 54720
      54720
      56172 43701
L 727 20774
      15717
      54710 21144
L 730 73610
      5066777774
L 731 5111000167
      7277777776
L 732 54710
L 733 0000000000
L 734 0400000736
L 735 6160000003
      0536000744
L 736 5111000052
      0311000741
L 737 5111000054
      63710
L 740 0270000000
L 741 10611
      5111000053
      10711
L 742 5111000054
      63710
L 743 0400000522 +
L 744 0100000000 X
    
```

```

SX2 X2-2 . PREVIOUS STACK ENTRY
SA2 B1+X2
MX7 1
LX7 60-SF.FINT
BX7 X7+X2
SA7 A2
SA7 A2
SA1 B1+82 . A(SUBP), ABSOLUTE
MX7 1 . CLEAR PENDING INTERRUPT BIT ON SUBP
LX7 60-SD.FINT
BX7 X7*X1
SA7 A1
AX1 36 . HAPPENS TO CONTAIN INTERRUPT DATUM
SX6 X1
SA6 A6-3 . MOVE TO SUBP CORE
SA1 B1+P.INTERR . DECREMENT PENDING INTERRUPT COUNT
SX7 X1-1
SA7 A1
PS EQ
EQ NEWT8
*
* NEWT9 SB6 B.ITLOFF
* NE B3,B6,NEWT10 . NOT INITIAL CALL EITHERSS
*
* NEW TOS ALL SET - RETURN TO CALLER IF NO ENVIORN ERROR
*
* SEE IF HE HAS AN ENVIORN ERROR
*
NEWT8 SA1 B1+P.SCR2+2
NZ X1,NEWT8.5 . SIGH, AN EROR
NEWT8.4 SA1 B1+P.SCR2+4 . RECOVER EXIT AND SPLIT
SB7 X1
JP B7
*
NEWT8.5 BX6 X1 . RECOVER ERROR NUMBER AND
SA1 B1+P.SCR2+3 . ERROR CLASS AND RETURN
*
NEWTERR BX7 X1 . VIA ERROR PROCESSING
SA1 B1+P.SCR2+4 . RECOVER EXIT
SB7 X1
EQ ERRPROC
RJ X6ISASTR
    
```

```

*
* NORMAL CALL
*
L 745 66521 5115000007 20152
L 746 63310 20152 624777772
L 747 67443 5111000124 63310
L 750 0634001010 21122 66600
L 751 5141000162 53441 63440
L 752 63310 0634001011 21122
L 753 63310 6151000170
L 754 5121000065 5111000123

```

```

SUBPC7 SB5 B2+B1 . ABS A(SUBP)
SA1 B5+SD.RAFI . GET SUBP FL
LX1 60-18 . SUBP RA
SB3 X1 60-18 . SUBP RA+FL
LX1 60-18
SB4 X1=5
SB4 B4-B3 . SUBP FL-2
SA1 B1+P.PARAMC-1
SB3 X1
GE B3,B4.SUBPC7
AX1 18
SB6 B0
SA4 B1+P.CLIST
SA4 B1+X4
SB4 X4
SB3 X1
LE B4,B3.SUBPC8
AX1 18
SB3 X1
SB5 B1+P.LOCALC
SA2 B1+P.PARAM-1
SA1 B1+P.PARAMC-2

```

```

PARAMETER TOTALS
FL NEEDED IN CALLEE FOR PARAMS
ERROR: NOT ROOM IN SUBP FL
CAPABILITY COUNTER
LEN LOC C-LIST (CALLEE)
NUM CAPS COMMING
ERROR: TOO MANY CAP PARAMS
ACTUAL SIZE OF ACTUAL PARAM
LOCAL C-LIST ORIG (ABS)
ACTUAL PARAM AREA POINTER
PARAMETER TYPE BIT MASK

```

```

REGISTERS FOR XFER OF PARAMETERS
X1 = PARAMETER TYPE BIT MASK
A2/X2 = ACTUAL PARAM PTR / DATA
A6 = PTR TO USER PARAM AREA (ABS)
B2 = INDEX OF CALLED SUBP
B3 = COUNTER FOR ACTUAL PARAM AREA
B5 = LOCAL C-LIST ORIGIN (ABS)
B6 = CAPABILITY INDEX COUNTER
B7 = CONST 1

```

```

TRANSFER PARAMETERS
B3,B0.SUBPC12 JP IF DONE
X1.SUBPC9 JP ON BLOCK PARAM OR CAPAB PARAM

```

```

DATUM PARAMETER
A2+B7 FETCH DATUM
X2
A6+B7 STORE DATUM
B3-B7 DECR ACTUAL COUNTER
LX1 SHIFT TYPE MASK
LP SUBPC8

```

```

L 755 0430001000 0337000760
L 756 54227 10622 54667 67337
L 757 20101 0200000755

```

```

SUBPC8 B0
B6
SA2 A2+B7
DX6 X2
SA6 A6+B7
SB3 B3-B7
LX1 LX1
LP LP

```

CAPABILITY PARAMETER

I	760	54227		SUBPC9	SA2	A2+B7	1ST WD OF APPARENT CAPABILITY
		20101			LX1	I	SHIFT TYPE MASK
			673377775		SB3	B3-2	DECR ACTUAL COUNTER
I	761	7202000001		SX0	X2+1		NO CAP CAN HAVE -1 IN LOW 18 BITS
			0300000765		ZR	X0, SUBPC10	JP IF REALLY BLOCK PARAMETER
				*			
				*			
L	762	76660		SX6	B6		INDEX OF CAP
		54667		SA6	A6+B7		STORE CAP INDEX
			20661	LX6	I		
			10722	BX7	X2		1ST WD OF CAPABILITY
L	763	53765		SA7	B5+X6		
		54227		SA3	A2+B7		END WD OF CAPABILITY
			10752	BX7	X2		
			66667	SB6	B6+B7		INCR CAPABILITY COUNTER
L	764	54777		SA7	A7+B7		
			0200000765	LP	SUBPC8		
				*			
				*			
				*			
				*			
L	765	54227		SUBPC15	SA2	A2+B7	
			0332000772	NO	X2, SUBPC11		JP IF CAPABILITY BLOCK
				*			
				*			
			63420				
L	766	21222		SB4	X2		MOVE DATA BLOCK
		73620		AX2	I8		MAX NUM WORDS
			56667	SX6	X2		ACTUAL NUM WORDS
			54667	SA6	A6+B7		PASS SIZE OF BLOCK TO USER
L	767	54664		SA0	A6+B7		ABS ADDR FOR BLOCK
			63420	SA6	A6+B4		INCR USER PARAM POINTER
			21252	SB4	X2		ACTUAL NUM WORDS
			73620	AX2	I8		
L	770	0114000000		SX0	X2		BUFFER ADDR
				RECS	B4		GET DATA BLOCK
				*			
L	771	0200000765		LP	SUBPC8		
				*			
				*			
				*			
				*			
L	772	76660		SUBPC17	SX6	B6	CAPABILITY INDEX
		63626			SB6	B6+X2	INCR CAPABILITY COUNTER
			20252	LX2	60-18		
			73720	SX7	X2		
L	773	63470		SB4	X7		ACTUAL NUM CAPS
		20722		LX7	I8		
			12667	BX6	X6+X7		PASS ACTUAL NUM AND STARTING INDEX
			54667	SA6	A6+B7		
L	774	20252		LX2	60-18		
		73020		SX0	X2		BUFFER ADDR
			73660	SX6	X6		STARTING INDEX IN LOCAL C-LIST
			20601	LX6	I		

L	775	53065		SA0	B5+X6	
		66444		SB4	B4+B4	
L	776	0114000000		RECS	B4	READ CAPABILITIES
L	777	0200000755	*	JP	SUBPC8	
			*			
			*			
			*			
			*			
			*			
L	1000	5141000162	SUBPC12	SA4	B1+P.CLIST	FINISHED WITH PARAMETERS
		72447		SX4	B4+B7	WRITE MODIFIED PORTION OF LOCAL
		53441		SA4	B1+X4	C-LIST TO THE ECS COPY OF SAME.
L	1001	73040		SX0	X4	GET ECS ADDR OF LOC C-LIST
		5101000051		SA0	B1+P.SCRB+I	
L	1002	0110000001		RECS	1	NOT AND UNIQUE NAME
L	1003	54500		SA5	A0	READ NOT ENTRY FOR LOCAL C-LIST
		13445		BX4	X4-X5	
		43747		MX7	X0	
		11447		BX4	X4+X7	
L	1004	0314001013		NZ	X4+SUBPC19	ERROR- OBJECT GONE FORM NOT
		15057		BX0	X7+X5	
		66666		SB6	B6+B6	CAPABILITY COUNT
L	1005	76470		SX4	B7	SKIP PAST LENGTH WORD OF C-LIST
		36004		IX0	X0+X4	
		54690		SA0	B5	
L	1006	0126000000		NECS	B6	ABS CM ADDR ORIG LOCAL C-LIST
L	1007	0400000736		EQ	NEWS	WRITE CHANGES TO ECS COPY OF LOCAL C-L
						. GO EXIT
			*			
			*			
			*			
			*			
			*			
L	1010	7170000007	SUBPC17	SX7	F ROOM	
		0400001012		EQ	C14	
L	1011	7170000010	SUBPC18	SX7	F NCAP	
L	1012	7160000004	CL4	SX6	F SUBP	
		0400000742		EQ	NEWERR	
L	1013	7170000000	SUBPC16	SX7	F CLNOT	
		7160000010		SX6	F MISCE	
L	1014	0400000742		EQ	NEWERR	
L	1015		ENDSUB		NEWTP BUTE	

```
*
* THIS ROUTINE HANDLES ERROR PROCESSING BY
* 1 - LOCATING A SUBP TO HANDLE THE ERROR
* 2 - MAKING SURE IT CAN BE CALLED WITH THE CURRENT STACK
*   ELSE CHANGING TO A STACK FULL ERROR
* 3 - CHANGING TO A NO SUBP TO TAKE ERROR IF NONE ARE
*   WILLING TO FIELD THE INDICATED ERROR
* 4 - DESCHEDULING THE PROCESS IF NOBODY WILL TAKE THAT ERROR
* 5 - USUALLY USING NEWTOS TO CALL A NICE
*   SUBP WILLING TO HANDLE THE ERROR
*
```

* AT ENTRY

```
* X6 = ERROR CLASS
* X7 = ERROR NUMBER
* B7 = RETLINK
*
```

* NOTE: IF THE ERROR CLASS IS .LT. 0 OR .GE.60, AN ERROR OUT OF RANGE
* ERROR IS SUBSTITUTED.

* CALLS NEWTOS, ENVIORN

* USES P.TEMP1 THRU P.TEMP7 ONE WAY OR ANOTHER

```
*
* ENTRY E.ERROR*ERRPROC
521 6170000003 * E.ERROR SB7 TOSPROC . RETLINK FOR MOST SYSTEM ACTION ERRORS
522 ERRPROC BSS 0
522 0000000000 PATCH2 PS
216 ECSSUB ERR, BUFA
450 5171000067 SA7 B1+P.PARAN+1 . SAVE ERROR NUMBER
51110000164 SA1 B1+P.STACK . STACK POINTER
451 63410 SB4 X1 . TOP OF STACK PTR
* B4=TOP, USED LATER
20130 LX1 60-35
64510 SB5 X1 . END OF STACK
452 0645000526 * GE B4,B5,KILLPROC . SORRY BABY, STACK FULL
0334000466 NG X6,ERROR0 . JUMP IF CLASS OUT OF RANGE
453 63360 SB3 X6 . SHIFT = ERROR CLASS
7120000074 SX2 60
37262 LX2 X6-X2
454 0322000466 PL X2,ERROR0 . JUMP IF CLASS OUT OF RANGE
6154000001 SB5 B4+1
56115 SA1 B1+B5 . WORD I OF OLD TOP OF STACK
456 63211 SB2 X1+B1 . A(SUBP), ABSOLUTE
5132000012 SA3 B2+SD,ESM . ERROR SELECTION MASK
22493 LX4 B3,X3 . BIT FOR THIS CLASS
5112000004 SA1 B2+SD,PTB . WORD WITH FATHER PTR
460 20152 0334000471 ERROR1.5 NG X4,ERROR2 . JUMP OUT IF THIS SUBP WILL TAKE IT
63210 LX1 60-18
SB2 X1 . FATHER PTR
0520000456 NE B2,B0,ERROR1 . LOOP IF NOT AT ROOT
*
```

* NO SUBPROCESS WANTS THIS ERROR -

* IF THIS IS THE SECOND TIME AROUND FOR THIS ERROR,
* WE DESCHEDULE THE PROCESS (MAY RESTART LATER IF AN
* APPROPRIATE INTERRUPT ARRIVES)
*

L 461 7256777765
 0305000534 +
L 462 5151000067
 5160000537 +
L 463 10755
 5170000540 +
 43030
L 464 15750
 20644
 11606
 12767

SX5
ZR
SA5
SA6
BX7
SA7
MX0
BX7
LX6
BX6
BX7

X6-E.NOERR
X5.STOPPROC
B1+P.PARAM+1
STOPPROC2
X5
STOPPROC3
24
-X0*X5
36
X0*X6
X6+X7

: JUMP IF NOERR TWICE
: SAVE ORIGINAL ERROR STUF

.. BEGIN TO PACK OLD ERROR INTO X7

*
* X7 := VRD 24/OLD ERR CLASS, 18/OLD MODIFIER, 18/OLD NUMBER
*

L 465 7160000012
 0400000521 +
L 466 5160000537 +
 5170000540 +
L 467 7160000004
 7170000016
L 470 0400000521 +

SX6
EQ
SA6
SA7
SX6
SX7
EQ

E.NOERR
E.ERROR
STOPPROC2
STOPPROC3
E.SUBP
E.BIGER
E.ERROR

. SUBSTITUTE NO SUBP TO TAKE ERROR
. SAVE ERROR
. SUBSTITUTE ERROR TOO BIG ERROR

* FOUND A SUBP TO TAKE EROR

L 471 20130
 63510
 20144
L 472 0754000460
 67221

ERROR2 LX1
SB5
LX1
LT
SB2

60-36
X1
36
B5-B4.ERRORI.5
B2-B1

. STACK LIMIT THIS SUBP
. SORRY, STACK TOO FULL FOR THIS SUBP
. A(CALLEE), REL B1

* RESET ERROR CLASS BIT IN THIS SUBP

L 473 6140000074
 67343
 15440
L 474 22734
 54730

MX0
SB4
SB3
BX4
LX7
SA7

1
60
B4-B3
-X0*X4
B3,X4
A3

* GET NEWTOS TO MAKE A NEW STACK ENTRY AND EXIT FOR US

* B2 = CALLEE
* B3 = OFFSET = ERROR CALL
* B7 = RETLINK

L 475 6130000001
 5161000066
 0400000516 +
L 476

SA6
SB3
EQ
ENDSUB

B1+P.PARAM
P.ERROFF
NEWTOS
ERRR,BUFA

Address	Offset	Instruction	Comments
526 244		KILLPRC BSS 0 ECSSUB KILLP.BUFA	
		*	
		*	MUST DESTROY PROCESS
		*	
450	7160000001 5160000000 x	SX6 1 SA6 =XI.LOCK	LOCK OUT PPU INTERRUPTS
		*	
451	5111000126 201E2 73010	SA1 B1+P.ROHEAD LX1 6+36 SX0 XI SA0 B1	GET MAT OFR PROCESS
452	56010	SA0 B1	
453	0110000001	RECS 1	
		*	
454	43647 54100 1E016	MX6 39 SA1 A0 BX0 X6*XI	GET P.ROHEAD FORM PROCESS IN ECS
455	0110000001	RECS 1	
		*	
456	54100 7160004200 20660	SA1 A0 SX6 PF.KILL LX6 36+1E BX6 XI+X6	SET DESTROY BIT AND PENDING ACTION FLAGS
457	12616 54610	SA6 A1	
460	0120000001	RECS 1	WRITE BACK TO ECS ... PROCESS WILL BE DESTROYED NEXT TIME IT SWAPS IN
		*	
461	13666 5160000000 x	BX6 X6-X6 SA6 =XI.LOCK	CLEAR PPU LOCKOUT
462		ENDSUB KILLP.BUFA	
531		BSS 0	
256		ECSSUB DSWAP.BUFA	
		*	
		*	
450	43601 5160000000 x	MX6 1 SA6 S.QUANT	MUST STOP QUANTUM CLOCK
451	6160000000 x	SB6 =XCLKWAIT	
452	6166777776 0560000452	SB6 B6-1 NE B6+B0.DOSWAP	LOOP TO GIVE PPU TIME TO INTERFERE
453	54160 20101 0331000531 +	SA1 A6 LX1 1 NG XI.DOSWAP	TEST QUANTUM CLOCK
		*	
		*	
454	5110000000 x 5120000000 x	SA1 S.CHARG SA2 S.OLDTM	UPDATE SYSTEM CLOCKS
455	5131000140 10611 54620	SA3 B1+P.SYSTIM BX6 XI SA6 A2	
456	37512 36735 5140000000 x	IX5 X1-X2 IX7 X3+X5 SA4 S.SYSTIM SA7 A3	
457	54730 36645	IX6 X4+X5	

SYSTEM ENTRY / EXIT
ERROR PROCESSOR - KILL PROCESS IF STACK FULL

COMPASS - VER 2.
FSENT

11/15/71 22.19.20.

PAGE 37

I	460	0000000000	54640	PATCH3	SA6	A4
				*	PS	
				*		
L	461	0200000000	x	JP	=XSWAPOUT	CALI SWAPPER
L	462			ENDSUB	DSWAP.BUFA	

```

534          STOPROC  BSS          0
270          ECSSUB    STOPP.BUFA

*
*
*          STOP PROCESS...MAY BE RESTARTED BY INTERRUPT
*
L 450 7160000001          SX6          1          STOP CPU INTERRUPTS
          5160000000 x    SA6          =XI.LOCK
L 451 5111000126          SA1          BI+P.ROHEAD
          20152          LX1          6+36
          73110          SX1          XI

*
*          RESET RUNNING FLAG
*
L 452 73010          SX0          XI
          56010          SA0          BI
L 453 0110000001          RECS          I          READ NOT
L 454 54200          SA2          A0
          43647          MX6          C0
          15056          BX0          =X6*X2          ECS ADDR OF P.ROHEAD
L 455 0110000001          RECS          I
L 456 54200          SA2          A0
          43601          MX6          I
          20672          LX6          60-PT.R
          15226          BX2          =X6*X2          CLEAR RUNNING FLAG1
L 457 20665          LX6          PT.R-PT.R
          12626          BX6          X2+X6          .GET DESCH FOR ERROR ERROR FLAG
          54650          SA6          A2
L 458 0120000001          RECS          I
*
L 461 6170000462          SBT          STOPROC1
          0200000000 x    JP          =XDESCHED          RESCHEDULE PROCESS
*
L 462 13666          STOPROC1  BX6          X6-X6          RELEASE CPU INTERRUPT LOCKOUT
          5160000000 x    SA6          =XI.LOCK
*
*
L 463 5140000537          SA4          STOPROC2          SAVE ORIG ERROR IN X6 AND X7 OF USER
          5150000540 +    SA5          STOPROC3
L 464 10644          BX6          X4
          10755          BX7          X5
          5161000160          SA6          BI+P.XPACK+14
L 465 5171000161          SA7          BI+P.XPACK+15
*
*
          0200000531 +    JP          DOSWAP          FORCE A SWAPOUT
*
*
L 466          ENDSUB          STOPP.BUFA
537          STOPROC2  BSSZ          1          SAVE ERROR CLASS IN CASE OF
540          STOPROC3  BSSZ          1
  
```

```

*
* THIS SUBROUTINE COPIES PARAMETERS FROM PARMBUF IN ECS
* INTO THE CORE OF THE TOP OF STACK.
*
* AT ENTRY
*
* B1 = PROCESS SCRATCH ARE
* B6 = RETURN LINK
* B7 = ERROR RETURN LINK
*
* P.PARMC = BUFFER CONTROL WORD
*          FORMAT: VED 42/NUMBER OF CAPS, 18/NUMBER OF DATA
*

```

	ENTRY	PUTRETD	PUTRETD
541		BSS	0
306		ECSSUB	PPAR.BUFA
450	76660	SX6	B6
	5161000556	SA6	B1+B7, MEMO1
451	5111000164	SA1	B1+B7, STACK
	52111	SA1	B1+X1
452	5120000000	SA2	BXS, USERA
	63250	SB2	X2
	53412	SA6	X1+B2
453	67221	SB2	B2-B1
	20104	LX1	ST, PCQ1
	0331000535	NG	X1, PUTRETD
454	20101	LX1	ST, PCQ1
	0331000535	NG	X1, PUTRETD
	10146	BX1	X4
455	21163	AX1	B1
	7211777764	SX1	X1-0168
456	0311000544	NZ	X1, ERRO
	20451	LX4	GO-19
457	0324000535	PL	X4, PUTRETD
	5014000001	SA1	BA+1
460	63410	SB4	B4
	5121000144	SA2	B1+B, XPACK+2
	21244	AX2	B6
461	6232777775	SB3	X2-B
	0740000537	LT	B4, B6, ERRI
462	0734000540	GT	B4, B6, ERR2
			ERROR: NEGATIVE POINTER
			ERROR: POINTER TOO LARGE
			* GET CONTROL WORDS, GET B RETURNED, CHECK VALIDITY OF DATA RETURN
	66442	SB4	B4+B2
	56114	SA1	B1+B4
463	5021000001	SA2	A1+1
	5131000125	SA3	B1+B, PARMC
464	43417	MX4	15
	15114	BX1	-X4*X1
	15224	BX2	-X4*X2
	73410	SX4	X1
465	20136	LX1	30
	0334000537	NG	X4, ERRI
	63410	SB4	X1

```

* PTR REL B1
* DATA CONTROL WORD
* CAPABILITY CONTROL WORD
* BUFFER CONTROL WORD
* ELIMINATE POSSIBLE PREVIOUS
* NUMBER RETURNED FIELDS
*
* START OF DATA BUFFER, REL USER RA
* MAX COUNT TO LOW X1
* ERROR: NEG DATA POINTER
*
* MAX ALLOWED WORD COUNT

```

I 466 63544
0735000545 63530
L 467 73442
0745000470 66450
I 470 76640
20655 20126 12616
L 471 54610

PUTRETP1

SB5
GT
SB5
SX4
LT
SB4
SX6
LX6
LX1
BX6
SA6

X4+B4 . END OF DATA BUFFER REL USER RA
B5, B3, ERR10
X3
X4+B2
B4, B5, PUTRETP1
B5
B4
B5
B0
X1+X6
A1

ERROR: NOT ROOM IN SUPP FL
NUM DATA WORDS IN BUFFER
REL CM ADDR FOR DATA WDS
FIND MIN (RETURN CONTROL, NUM
WDS IN BUFFER)
PUT THE NUMBER RETURNED IN
THE AUTHORIZATION FOR THE
HAPPY RECEIVER

B4 = NUM DATA WDS TO MOVE
B5 = NUM DATA WDS IN BUFFER
X4 = CM ADDR FOR DATA WDS, REL B1
X2 = CAPABILITY RETURN CONTROL
X3 = BUFFER DATA

CHECK OUT CAPABILITY RETURN

21322
L 472 0720000542 63320
20226 63320
L 473 63630
0736000474 66360
L 474 76630
20655 20226 12626
L 475 54620

PUTRETP2

AX3
SB2
LT
LX2
SB3
SB6
LT
SB3
SX6
LX6
LX2
BX6
SA6

B
X2
B2, B0, ERR5
B0
X2
X3
B3, B6, PUTRETP2
B5
B3
B5
B0
X2+X6
A2

STARTING CAPABILITY INDEX
ERROR: NEGATIVE C-LIST INDEX

NUM CAPS IN BUFFER
FIND MIN (CAPS IN BUFFER, RETURN
CONTROL)
PUT THE NUMBER RETURNED IN
THE AUTHORIZATION SO THE
HAPPY RECEIVER CAN TELL
WHAT'S GOING ON

B2 = STARTING CAP INDEX
B3 = NUM CAPS TO MOVE
B4 = NUM DATA WDS TO MOVE
B5 = NUM DATA WDS IN BUFFER
X4 = REL CM ADDR FOR DATA

CHECK FOR CAPS IN RANGE

5121000122 76150
L 476 66523
53521 63620
L 477 0665000502
L 500 5055000002 67556 63650
L 501 0305000541
0654000500

PUTRETP3

SA2
SX1
SB5
SA5
SB6
GE
SA5
SB5
SB6
ZR
GE

B1+P.CLIST
B5
B2+B3
B1+X2 . SIZE OF LOCAL C-LIST
X5
B6, B5, PUTRETP4
A5+2 . LENGTH OF NEXT C-LIST
B5-B6
X5
X5, ERR4
B5, B6, PUTRETP3

X1 = NUM DATA WDS IN BUFFER
MAX CAP INDEX
JUMP IF NOT IN LOCAL C-LIST
ERROR: TOOM MANY CAPS

```

* WE ARE READY TO MOVE
*
*
* CAPABILITIES OK
* **** MOVE DATA WORDS ****
L 502 53041 7100000000 X PUTRETP4 SA0 B1+X4 ABS CM ADDR OF DATA BUFFER
L 503 0114000000 SX0 PARMBUF
L 504 36101 RECS B4
IX1 X0+X1 INCR BUFFER ADDRESS
*
* **** MOVE CAPABILITIES ****
*
* EQ B3,B0,PUTRET13 SKIP IF NO CAPS TO MOVE
*
* FIND THE C-LIST WHERE THE CAPS START
*
L 505 63440 53421 SA4 B1+X2 LEN LOCAL C-LIST
L 506 5044000002 0724000530 SB4 X4
GT B4,B2,PUTRET11 IF IN LOCAL C-LIST
L 507 0624000506 67254 SA4 A4+2 LEN NEXT C-LIST
SB2 B2-B4 LOOK FOR PROPER C-LIST
SB4 X4
GE B2,B4,PUTRETP5
*
* B2 = INDEX WHERE STUFF IS LANDING IN THE C-LIST WERE CURRENTLY
* WORKING ON
*
*
* MOVE NEXT SECTION OF CAPABILITIES
*
L 510 67542 0635000571 PUTRETP6 SB5 B4-B2 * SPACE LEFT IN THIS C-LIST
LE B5,B4,PUTRETP7 * (P.SCR/2)
L 511 6160000024 66530 PUTRETP7 SB5 B3 * WE HAVE LESS THAN THAT TO MOVE
LE B5,SCR/2 *
L 512 66560 0665000513 PUTRETP8 SB5 B6 * BUT NOT ENOUGH SCRATCH SPACE
* B5 = NUM OF CAPS TO MOVE
* B4 = LENGTH OF CURRENT C-LIST
L 513 66655 5101000000 PUTRETP8 SB6 B5+B6
SA0 B1+P.SCR
L 514 0116000000 10011 SX0 X1
RECS B6 HEAD CAPABILITIES
L 515 73106 74600 SX1 X0+B6 INCR BUFFER ADDR
SX6 A0 * SAVE FOR LATER
SB3 B3-B5 * DECR COUNT OF CAPS LEFT TO MOVE
*
*
* MOVE CAPABILITIES TO C-LIST
*
* X1 = BUFFER ADDRESS
* X6 = CM ADDR
* B6 = COUNT*2
* B2 = INDEX
* MOT AND U.N. OF C-LIST
L 516 5054000001 5101000050 PUTRETP9 SA5 A4+1
L 517 7205000000 SA0 B1+P.SCR2
SX0 X5+0

```

```

520 0110000001 42727
521 54300 13553 11575 15037
522 0915000543 42773 37007
523 76220 20201 36002 53060
524 0126000000
*
*
*
525 0430000535 62255
526 0524000510
*
*
*
527 63440 5047000002 66200 0200000510
*
*
*
530 67542 PUTRETT1 0635000521 66530
531 67335 PUTRETT2 7162000170 73662
532 73661 66655 10071 53060
533 0116000000
534 73106 0200000576
*
*
*
535 5111000056 PUTRETT3 63610
536 0260000000
*
*

```

```

MX7 39
RECS 1
SA3 A0
BX5 X5-X3
BX5 X7-X5
BX0 X7-X3
NZ X5,ERR6
MX7 59
IX0 X0-X7
SX2 52
LX2 1
IX0 X0+X2
SA0 X6
RECS X6

```

READ NOT

ERROR: C-LIST GONE

A(C-LIST PROPER)

INDEX REL THIS C-LIST

WRITE CAPABILITIES

STEP TO NEXT SECTION

JP IF DONE (COUNT = 0)

INCR INDEX

JP IF NOT DONE W/ C-LIST

MOVE TO NEXT C-LIST

LENGTH

INDEX = 0

XFER TO LOCAL C-LIST

FIND MIN(LENGTH-INDEX,COUNT)

DECR COUNT

ABS CM ADDR

READ CAPABILITIES

INCR BUFFER ADDR

GO COPY TO ECS COPY OF C-LIST

ALL DONE

RECOVER RETLINK

SYSTEM ENTRY / EXIT
RETURN PARAMETERS

COMPASS - VER 2.
FSENT

11/15/71 22.19.22.

PAGE 43

*** ERRORS ***

1
1
1
1
1
1
1
1
1
1

537 66400
540 66400
541 66400
542 66400
543 66400
544 66400
545 66400
546

*
*

ERR1 ERROR
ERR2 ERROR
ERR4 ERROR
ERR5 ERROR
ERR6 ERROR
ERR9 ERROR
ERR10 ERROR
ENDSUB

80, NEGPT
80, BIGPT
80, BCPBLK
80, NEGIX
80, CLMOT
80, NOXJ
80, BDTBLK
PPAR, BUFA

NEGATIVE POINTER
FL EXCEEDED
CAP BLK NOT ALL IN FU C-LIST
NEGATIVE C-LIST INDEX
C-LIST GONE FROM MOT
XJ CLOBBERED
DATA BLK EXCEEDS FL

```

*
*
544 6170000000 X 0400000567 + ERR26 SB7 EXXSABY
      GETMOD
545 66400 ERR23 ERROR B0,IPO
546 6170000000 X 0400000567 + ERR25 SB7 EXCAPTY
      EQ GETMOD
547 66400 ERR40 ERROR B0,NOOP
550 66400 ERR41 ERROR B0,CLMOT
551 66400 ERR15 ERROR B0,NEBDCY
552 66400 ERR15X ERROR B0,BIGBCT
553 66400 ERR21 ERROR B0,NEBCT
554 66400 ERR21X ERROR B0,BIGBT
      554 + ERR24 EQU
555 6170000000 X 0400000567 + ERR22 SB7 EXBIGIX
      EQ GETMOD
556 6170000000 X 0400000567 + ERR27 SB7 EXNOTCL
      EQ GETMOD
557 6170000000 X ERR22X SB7 EXNEGIX
560 66400 ERR28 ERROR B0,NOXJ
561 6170000000 X 0400000567 + ERR30 SB7 EXNEGCT
      EQ GETMOD
      557 + ERR31 EQU
      ERR32 SB7 EXNEGQAK
      EQ GETMOD
562 6170000000 X 0400000567 + ERR33 ERROR B0,BIGCNT
563 66400 ERR34 SB7 EXBDBLK
      EQ GETMOD
564 6170000000 X 0400000567 + ERR35 SB7 EXBDBLK
      EQ GETMOD
565 6170000000 X 0400000567 + ERR36 SB7 EXCLMOT
566 6170000000 X ERR36 SB7
567 74510 GETMOD SX5
      5121000144 SA2
      53221 SA2
570 21222 AX2
      73220 X2
      37552 IX5
571 5121000142 SA2
      21244 AX2
      37552 IX5
572 0270000000 JP
573 END B7

```

CALCULATE MODIFIER FROM CURRENT ADDRESS IN IPLIST

EXIT TO ERROR PROCESSOR

36662 STORAGE USED
6600 ASSEMBLY

2834 STATEMENTS
14.714 SECONDS

590 SYMBOLS
1182 REFERENCES

SYSTEM ENTRY / EXIT
SYMBOLIC REFERENCE TABLE.

ACAP	0	EXTERNAL*	17/42							
ACTIONL	125	PROGRAM*	12/49	17/32	17/36	17/41	17/45	17/50	18/01	18/04
			13/25	17/32	17/37	17/41	17/46	17/50	18/02	18/05
			14/27	17/33	17/37	17/42	17/46	17/51	18/02	18/07
			17/28	17/33	17/38	17/42	17/47	17/51	18/03	18/07
			17/29	17/34	17/38	17/43	17/47	17/52	18/03	18/08
			17/30	17/34	17/39	17/43	17/48	17/52	18/04	18/08
			17/30	17/35	17/39	17/44	17/48	17/52	18/04	18/09
			17/31	17/35	17/40	17/44	17/49	17/53	18/05	18/09
			17/31	17/36	17/40	17/45	17/49	18/01	18/05	18/10
ACTLIM	10		14/28	D						
BCAP	0	EXTERNAL*	18/04							
BCPBLK	0	EXTERNAL*	43/06	44/29						
BDAT	0	EXTERNAL*	18/03							
BDBLK	0	EXTERNAL*	43/10	44/27						
BIGNY	0	EXTERNAL*	44/27							
BIGIX	0	EXTERNAL*	44/15							
BIGPCT	0	EXTERNAL*	44/12							
BIGPT	0	EXTERNAL*	8/03	43/05	44/14					
BIFA	450		6/03	8/04	36/01	36/30	36/32	37/07	38/52	43/11
			8/04	34/30	36/01	36/30	37/07	38/03	38/52	43/11
			8/04	36/01	36/03	36/30	37/07	38/52	39/15	43/11
BIFB	650		26/45	33/48	33/48	33/48				
CAPTY	0	EXTERNAL*	44/06							
CHKBLK	0	EXTERNAL*	17/44							
CHMPO	0	EXTERNAL*	18/07							
CHMPRW	0	EXTERNAL*	18/06							
CLKWAIT	0	EXTERNAL*	36/36							
CLNOT	0	EXTERNAL*	43/08	44/10	44/31					
CLDPAE	0	EXTERNAL*	17/31							
CL	1012		33/40	33/42	L					
CREABLK	0	EXTERNAL*	17/29							
DELBLK	0	EXTERNAL*	17/45							
DESCHED	0	EXTERNAL*	38/32							
DISASTR	0	EXTERNAL*	30/46							
DOSWAP	531	PROGRAM*	36/30	L	36/41	38/47				
DOSWAP1	452		36/37	L	36/38					
DSPCLOX	0	EXTERNAL*	18/02							
DSPGCLX	0	EXTERNAL*	18/08							
DSEAP	531	PROGRAM*	37/07	L						
ENVIRON	0	EXTERNAL*	28/25							
ERROR0	466		34/37	34/41	35/21	L				
ERROR1	456		34/44	L	34/51					
ERROR1.5	460		34/49	L	35/32					
ERROR2	471		34/48	L	35/29	L				
ERRORDC	522	PROGRAM*	30/45	34/24	E	34/27	L			
ERRR	523	PROGRAM*	36/01	L						
ERR1	537		39/38	39/52	43/03	L				
ERR10	545		40/02	43/09	L					
ERR15	551	PROGRAM*	5/01	44/10	L					
ERR15X	552	PROGRAM*	4/53	44/11	L					
ERR2	540		39/39	43/04	L					
ERR21	553	PROGRAM*	11/33	44/12	L					

SYSTEM ENTRY / EXIT
SYMBOLIC REFERENCE TABLE.

ERR21X	554	PROGRAM*	11/34	44/13 L	44/14				
ERR22	555	PROGRAM*	21/15	21/22	22/13	44/15 L			
ERR22X	557	PROGRAM*	20/46	22/03	44/19 L	44/23			
ERR23	545	PROGRAM*	11/43	44/05 L					
ERR24	554	PROGRAM*	12/10	44/14 D					
ERR25	546	PROGRAM*	21/03	44/06 L					
ERR26	544	PROGRAM*	20/09	44/03 L					
ERR27	556	PROGRAM*	21/49	44/17 L					
ERR28	560	PROGRAM*	4/47	44/20 L					
ERR30	561	PROGRAM*	22/31	44/21 L					
ERR31	557	PROGRAM*	23/31	44/23 D					
ERR32	562	PROGRAM*	22/34	23/33	44/24 L				
ERR33	563	PROGRAM*	22/36	23/53	44/26 L				
ERR34	564	PROGRAM*	23/37	44/27 L					
ERR35	565	PROGRAM*	24/33	25/23	44/29 L				
ERR36	566	PROGRAM*	24/32	44/31 L					
ERR4	541		40/51	43/05 L					
ERR40	547	PROGRAM*	11/51	44/08 L					
ERR41	550	PROGRAM*	21/35	22/10	44/09 L				
ERR5	542		40/23	43/06 L					
ERR6	543		42/37	43/07 L					
ERR9	544		39/30	43/08 L					
EVCHANG	0	EXTERNAL*	17/33						
EVNT	0	EXTERNAL*	17/32						
E.BIGER	16		35/24						
E.CLMOT	0		33/44						
E.DSWAP	256	EVENT	36/31 L						
E.FCS	0	EXTERNAL*	5/25	12/01	22/10	25/09	33/30	36/24	36/30
			6/32	12/05	22/16	25/13	33/48	36/30	36/52
			7/13	21/31	23/14	32/36	36/01	37/07	41/08
			8/24	21/42	24/16	33/04	36/14	38/18	41/42
			11/47	22/02	24/48	33/19	36/19	38/22	42/03
E.ERROR	521	PROGRAM*	34/24 E	34/26 L	35/20	35/25			
E.FRR	216	EVENT	34/29 L						
E.FULSTK	6		27/24						
E.KILLER	244	EVENT	36/32 L						
E.MAPER	5		29/43						
E.MISCE	10		33/45						
E.MOT	0	EXTERNAL*							
E.NCAD	10		33/41						
E.NEFTP	51	EVENT	26/44 L						
E.NOERR	12		35/35	35/19					
E.PPAR	306	EVENT	39/14 L						
E.PROC	5		29/41						
E.ROOM	7		33/39						
E.SFRET	0	EVENT	6/32 L						
E.STOPP	270	EVENT	38/32 L						
E.SUBP	4		27/33	33/42	35/23				
FCAP	0	EXTERNAL*	17/41						
FDAT	0	EXTERNAL*	17/39						
FRET11	513		6/39	7/45 L					
FRET17	516		6/20	7/53 L					
FRET18	517		6/29	8/01 L					

SYSTEM ENTRY / EXIT
SYMBOLIC REFERENCE TABLE.

COMPASS - VER 2.

11/15/71 22.19.24.

PAGE 47

FRET19	520		7/20	8/02	L				
FRET3	453		6/16	6/18	L				
FRET5	476		7/12	7/41	L				
FRET7	505		7/14	7/23			7/27	L	
GETCAP	0	EXTERNAL*	6/17	11/38					
GETEVF	0	EXTERNAL*	17/30						
GETMOD	567	PROGRAM*	44/24	44/16		44/22	44/29	44/32	L
			44/27	44/18		44/25	44/30		
IPO	0	EXTERNAL*	8/21	44/06					
I.LOCK	0	EXTERNAL*	36/27	36/28	S	38/28	38/35	S	
JUMPCAL	0	EXTERNAL*	18/29						
J.ADDOPT	57		15/45	17/43	D				
J.ADDORD	114		16/15		D				
J.ANYCAP	34		15/41	17/42	D				
J.ARMIT	234		17/19		D				
J.BLKCAP	56		15/43	18/04	D				
J.BLKDAT	35		15/42	18/03	D				
J.CAGEN	164		16/46		D				
J.CALSUB	35		15/26	17/36	D				
J.CAPIN	31		15/23		D				
J.CAPOU	33		15/24		D				
J.CCLOA	116		16/16		D				
J.CGEN	166		16/47		D				
J.CHKBLK	62		15/49	17/44	D				
J.CHMPRO	202		17/24	18/07	D				
J.CHMPRW	201		17/23	18/06	D				
J.CLRDAE	137		16/28	17/51	D				
J.CLRHIB	143		16/31		D				
J.CLRMG	226		17/16		D				
J.COPYOP	60		15/47		D				
J.CROUT	232		17/18		D				
J.CPROC	2		15/21		D				
J.CRUIN	230		17/17		D				
J.CRZRO	210		17/29		D				
J.CRALBK	122		16/18		D				
J.CREBLK	6		15/24	17/29	D				
J.CRECC	15		15/19		D				
J.CRECL	0		14/53		D				
J.CREFIL	4		15/22		D				
J.DCLOX	151		16/37	18/02	D				
J.DELAB	131		16/24		D				
J.DELBLK	63		15/20	17/45	D				
J.DELCL	112		16/14		D				
J.DELEC	147		16/35		D				
J.DELFIL	64		15/22		D				
J.DELSUB	133		15/25		D				
J.DISIT	236		17/20		D				
J.DISMAP	74		16/27		D				
J.DISpOp	162		16/45		D				
J.DISPST	104		16/11		D				
J.DISSEN	106		16/12		D				
J.DLOpp	220		17/13		D				
J.DLPRO	126		16/20		D				

SYSTEM ENTRY / EXIT
SYMBOLIC REFERENCE TABLE.

J.DONATE	120	16/17	D	
J.DPROD	135	16/26	D	
J.DSCAP	25	15/20	D	
J.DSCLX	203	17/25	D	18/08
J.DSFMAP	110	16/13	D	
J.DSPAB	152	16/39	D	
J.DSPALC	177	17/01	D	
J.DSPOB	175	16/53	D	
J.DSPSP	170	16/48	D	
J.EFMGEN	42	15/20	D	
J.EFMLOC	44	15/21	D	
J.EVENT	13	15/10	D	17/32
J.FIXCAP	53	15/40	D	17/41
J.FIXDAT	51	15/38	D	17/39
J.FRETUR	40	15/29	D	
J.FSON	204	17/27	D	
J.GETEVF	130	16/22	D	17/50
J.GRAB	222	17/14	D	
J.HANG	14	15/11	D	17/33
J.INCLR	173	16/52	D	
J.INMTR	212	17/10	D	
J.JPREY	75	16/38	D	
J.JUMSUB	36	15/27	D	18/09
J.KAPZRO	70	16/22	D	17/46
J.MGETF	145	16/33	D	18/01
J.MGETH	145	16/22	D	17/52
J.MKEC	11	15/38	D	
J.MKOPER	46	15/23	D	
J.MKSUBP	23	15/19	D	
J.MODBC	124	16/19	D	
J.MOVBLK	73	16/55	D	17/49
J.MOVCP	214	17/11	D	
J.MOVEC	27	15/21	D	
J.MOVMT	215	17/12	D	
J.MPCKRO	71	16/53	D	17/47
J.MPCHRW	72	16/54	D	17/48
J.NEWUN	102	16/10	D	
J.NWTMP	17	15/14	D	
J.PINT	100	16/29	D	
J.REDAHP	66	15/22	D	
J.REFILE	10	15/26	D	17/31
J.RETPAR	160	16/42	D	
J.RETURN	37	15/28	D	17/37
J.RREG	22	15/17	D	17/35
J.SETDAE	140	16/29	D	17/52
J.SETIIB	141	16/20	D	
J.SPRT	206	17/28	D	
J.SREG	21	15/16	D	17/34
J.STMSG	224	17/15	D	
J.TIMDT	154	16/40	D	
J.TROB	172	16/50	D	18/05
J.USRCAP	52	15/39	D	17/40
J.USRDAT	50	15/37	D	17/38

SYSTEM ENTRY / EXIT
SYMBOLIC REFERENCE TABLE.

OPINTR14	421	PROGRAM*	21/17	21/27	L				
OPINTR15	430	PROGRAM*	20/50	21/48	L				
OPINTR16	443	PROGRAM*	20/39	22/29	L				
OPINTR17	457	PROGRAM*	20/40	23/27	L				
OPINTR18	470	PROGRAM*	24/19	24/11	L				
OPINTR19	474	PROGRAM*	24/19	24/24	L				
OPINTR20	475	PROGRAM*	24/28	24/32	L				
OPINTR21	500	PROGRAM*	24/30	24/41	L	25/24			
OPINTR22	505	PROGRAM*	25/15	25/16	L				
OPINTR23	515	PROGRAM*	24/18	25/15	L	25/28			
OROPT	0	EXTERNAL*	17/43						
PARMBUF	0	EXTERNAL*	6/43	12/16		41/06			
PATCH1	733		30/19	L					
PATCH2	522	PROGRAM*	34/28	L					
PATCH3	460		37/22	L					
P.F.FE	11		38/36						
P.F.KILL	4200		36/31						
P.F.R	2		38/24	38/26					
P.PAR	541	PROGRAM*	43/11	L					
P.PRINT	0	EXTERNAL*	3/46						
P.SANY	0	EXTERNAL*	44/29						
P.S.MASKL	1		6/31	7/01		11/50	12/01	24/43	
PUTRETP	541	PROGRAM*	4/22	39/12	F	39/12			
PUTRETP1	470		40/05	40/07	L				
PUTRETP2	474		40/27	40/29	L				
PUTRETP3	500		40/48	40/52	L				
PUTRETP4	502		40/47	41/05	L				
PUTRETP5	505		41/19	41/22	L				
PUTRETP6	510		41/30	42/20	L	42/25			
PUTRETP7	511		41/41	41/33	L				
PUTRETP8	513		41/34	41/38	L				
PUTRETP9	516		41/31	42/41	L				
PUTRETI1	530		41/18	42/29	L				
PUTRETI2	531		42/20	42/32	L				
PUTRETI3	535		29/24	29/26	L	29/22	41/17	42/18	42/44
P.CLIST	162		7/35	12/24		21/14	24/21	31/16	33/12
P.ERROFF	1		27/21	29/28		35/21			40/42
P.INTERR	167		30/16						
P.INTOFF	2		3/51	29/48					
P.ITLOFF	3		30/22						
P.LOCALC	170		7/14	12/23		31/23	42/23		
P.PARAK	66		7/26	12/18	S	29/33	31/24	34/20	35/07
P.PARAKC	125		12/14	31/11	S	31/25	39/43		35/50
P.PETS	62		19/36	19/40	S	19/50			
P.ROHEAD	126		36/19	38/09					
P.SOR	0		41/29						
P.SORL	50		12/24	41/33					
P.SOR2	50		6/23	24/43		28/22	28/32	30/32	33/17
			11/45	28/48	S	28/26	29/39	30/41	41/52
			13/16	28/17	S	28/27	30/20	30/43	
P.STACK	164		3/37	4/23		11/28	29/52	39/17	
			4/05	7/45		26/50	34/31	44/33	
P.SYSTIM	140		9/17	36/46					

SYSTEM ENTRY / EXIT
SYMBOLIC REFERENCE TABLE.

P.TEMP1	56		6/53 S	7/36	39/16 S	42/44			
P.TEMP2	57		6/40 S	7/29					
P.TEMP3	50		6/42 S	7/27					
P.TEMP4	51		6/51 S	7/15					
P.TEMP5	62		19/36						
P.TEMP6	63		6/44 S	12/17 S	22/35	23/29			
P.USRTIM	137		10/39						
P.XPACK	142		4/40	6/08	9/08	10/50	23/08	38/43 S	44/38
			4/48	6/45	9/30	23/03	38/42 S	39/35	
DEADFIL	0	EXTERNAL*	17/21						
RESTORE	0	EXTERNAL*	17/25						
RETURN	0	EXTERNAL*	17/27						
SAVEG	0	EXTERNAL*	17/24						
SD.FSM	2		34/25						
SD.FINT	0		30/10						
SD.ORIG	5		27/15	29/04	29/25				
SD.PTRS	4		26/46	27/50	34/47				
SD.RAFL	7		28/47	28/48	29/05				
SETDAE	0	EXTERNAL*	17/22			31/05			
SETRET	27	PROGRAM*	8/24 L						
SF.FINT	1		30/04						
SF.II	0		27/25						
SF.PCR1	4		4/11	4/13	9/06	27/23	27/25	39/23	39/25
SF.PCR2	5		4/19	4/25	6/29	4/32	4/39	39/25	
STOPP	534	PROGRAM*	38/32 L						
STOPROC	534	PROGRAM*	35/26						
STOPROCI	462		38/31	38/01 L					
STOPROCI2	537	PROGRAM*	35/28	38/34 L					
STOPROCI3	540	PROGRAM*	35/10 S	35/21 S	38/28	38/51 L			
SUBPCAL	0	EXTERNAL*	35/10 S	35/22 S	38/30	38/53 L			
SUBPC10	765		17/26						
SUBPC11	772		32/25	32/22 L					
SUBPC12	1000		32/23	32/42 L					
SUBPC17	1010		31/39	33/14 L					
SUBPC17	1010		31/13	33/30 L					
SUBPC18	1011		31/20	33/41 L					
SUBPC19	1012		32/23	33/44 L					
SUBPC7	745		29/27	31/04 L					
SUBPC8	755		31/39 L	31/49	32/17	32/37	33/05		
SUBPC9	760		31/40	32/01 L					
SUBPJUM	0	EXTERNAL*							
SWAPOUT	0	EXTERNAL*							
SYSTEMET	0	PROGRAM*	9/29	37/05					
SYSTEMET	7	PROGRAM*	3/36 E	3/37 L					
SYCHARG	0	EXTERNAL*	3/36 E	3/43	4/01 L				
S.FNPLG	2		36/44						
S.OLDTM	0	EXTERNAL*	9/27 S	10/05 S					
S.QUANT	0	EXTERNAL*	9/16	10/33	36/45				
S.RETU	44	PROGRAM*	9/28	36/35 S					
S.SWPTM	0	EXTERNAL*	3/36 E	9/30 L					
S.SYSTEM	0	EXTERNAL*							
S.USERA	0	EXTERNAL*	9/22	36/51					
S.USRTM	0	EXTERNAL*	28/22	39/19					
	0	EXTERNAL*	10/44						

SYSTEM ENTRY / EXIT
SYMBOLIC REFERENCE TABLE.

TABLE 240

14/27	D	15/19	D	15/39		16/04		16/19	D	16/39		17/04		17/19	D
14/33		15/19		15/39	D	16/05		16/19		16/39	D	17/05		17/19	
14/33	D	15/20		15/39		16/05	D	16/20		16/39		17/05	D	17/20	D
14/33		15/20	D	15/40		16/05		16/20	D	16/40		17/05		17/20	D
15/01		15/20		15/40		16/07		16/20		16/40	D	17/07		17/20	
15/01	D	15/21		15/40		16/07	D	16/22		16/40		17/07	D	17/22	
15/01		15/21	D	15/41		16/07		16/22	D	16/41		17/07		17/22	
15/02		15/21		15/41	D	16/09		16/22		16/41	D	17/08		17/31	
15/02	D	15/23		15/41		16/09	D	16/24		16/41		17/08	D	17/32	
15/02		15/23	D	15/42		16/09		16/24	D	16/42		17/08		17/33	
15/04		15/23		15/42		16/09		16/24		16/42	D	17/09		17/34	
15/04	D	15/24		15/42		16/09	D	16/25		16/42		17/09	D	17/35	
15/04		15/24	D	15/43		16/09		16/25	D	16/43		17/09		17/36	
15/05		15/24		15/43	D	16/10		16/25		16/43	D	17/10		17/37	
15/05	D	15/26		15/43		16/10	D	16/26		16/43		17/10	D	17/38	
15/05		15/26	D	15/43		16/10		16/26	D	16/44		17/10		17/39	
15/06		15/26		15/45		16/11		16/26		16/44	D	17/11		17/40	
15/06	D	15/27		15/45	D	16/11	D	16/28		16/44		17/11	D	17/41	
15/06		15/27	D	15/47		16/11		16/28	D	16/47		17/11		17/42	
15/08		15/27		15/47		16/11		16/28		16/47	D	17/12		17/43	
15/08	D	15/28		15/47		16/12		16/29		16/47		17/12	D	17/44	
15/08		15/28	D	15/49		16/12	D	16/29	D	16/48		17/12		17/45	
15/10		15/28		15/49		16/13		16/29		16/48	D	17/12		17/46	
15/10	D	15/29		15/49		16/13	D	16/30		16/49		17/13	D	17/47	
15/10		15/29	D	15/50		16/13		16/30	D	16/50		17/13		17/48	
15/11		15/29		15/50		16/14		16/30		16/50	D	17/14		17/49	
15/11	D	15/30		15/50		16/14	D	16/31		16/50		17/14	D	17/50	
15/11		15/30	D	15/50		16/14		16/31	D	16/50		17/14		17/51	
15/13		15/30		15/50		16/15		16/31		16/50	D	17/15		17/52	
15/13	D	15/31		15/50		16/15	D	16/32		16/50		17/15	D	17/53	
15/14		15/31		15/50		16/15		16/32	D	16/50		17/15		18/01	
15/14	D	15/33		15/50		16/15	D	16/33		16/50	D	17/16		18/02	
15/14		15/33	D	15/50		16/15		16/33	D	16/50		17/16	D	18/03	
15/16		15/33		15/50		16/15		16/33		16/50	D	17/16		18/04	
15/16	D	15/37		15/50		16/17		16/35		16/50		17/17	D	18/05	
15/16		15/37	D	15/50		16/17	D	16/35	D	16/50		17/17		18/06	
15/17		15/37		15/50		16/17		16/35		16/50	D	17/17		18/07	
15/17	D	15/38		15/50		16/18		16/35	D	16/50		17/18	D	18/08	
15/17		15/38	D	15/50		16/18	D	16/37		16/50		17/18		18/09	
15/19		15/38		15/50		16/18		16/37	D	16/50		17/18		18/10	
15/19	D	15/38		15/50		16/19		16/37		16/50	D	17/19			
3/36	F	3/44	L	3/50		4/27		4/26							
4/12		9/06	L												
3/47		4/05	L												
5/02		9/08	L												
3/25		3/47	L												
4/19		4/29	L												
4/20		4/23	L												
4/21		6/01	L												
4/14		4/39	L												
4/48	L	7/49													
18/05															
21/48															

TOSPROC	3	P PROGRAM*
TOSPR1	32	P PROGRAM*
TOSPR2	10	P PROGRAM*
TOSPR2.5	33	P PROGRAM*
TOSPR3	5	P PROGRAM*
TOSPR4	16	P PROGRAM*
TOSPR5	15	P PROGRAM*
TOSPR6	27	P PROGRAM*
TOSPR7	20	P PROGRAM*
TOSPR8	23	P PROGRAM*
TRDB	0	EXTERNAL*
T.CLIST	1377	

SYSTEM ENTRY / EXIT
 SYMBOLIC REFERENCE TABLE.

T.OPER	1677		6/79	11/41	
UCALL1	60	PROGRAM*	11/08	11/18	L
UCALL16	115	PROGRAM*	12/43	13/02	L
UCALL2	67	PROGRAM*	11/35	11/40	L
UCALL25	110	PROGRAM*	12/12	12/31	L
UCALL3	111	PROGRAM*	7/32	12/28	12/41 L
UCAP	0	EXTERNAL*	17/40		
UDAT	0	EXTERNAL*	17/38		
USERCAL	45	PROGRAM*	10/04		L
WRITFIL	0	EXTERNAL*	17/30		
Z.ASYS	0	EXTERNAL*	12/48	13/24	
Z.ESYS	0	EXTERNAL*	10/37		
Z.LSYS	0	EXTERNAL*	9/15		