

SUBPROCESS OPERATIONS
STORAGE ALLOCATION.

COMPASS - VER 2.

11/03/71 15.37.13.

PAGE

1

ADDRESS LENGTH

0- 171
171

BINARY CONTROL CARDS.

IDENT SUBPROC
END

BLOCKS	TYPE	ADDRESS	LENGTH
ABSOLUTE*	ABSOLUTE	0	167
PROGRAM*	LOCAL	0	171
ESUBP	COMMON	0	656

ENTRY POINTS.

E.MKSUB	-	0	E.FSON	-	262	I.JMRET	-	4	E.SRETN	-	405
L.MKSUB	-	133	I.FSON	-	20	F.SPRET	-	373	L.SRETN	-	53
E.DLSUR	-	133	F.DSPSP	-	302	I.SPRET	-	2	GETP	-	22
L.DLSUR	-	77	I.DSPSP	-	42	F.FRET	-	375	E.GETP	-	460
JUMPCAL	-	0	E.ERCAL	-	344	I.FRET	-	2	L.GETP	-	57
SUBPCAL	-	5	I.ERCAL	-	5	RETURN	-	51	FINDSUB	-	25
FINDS	-	14	E.SESM	-	351	F.RET	-	377	OMYGO	-	33
E.FINDS	-	232	I.SESM	-	10	I.RET	-	2	E.OMYGO	-	537
L.FINDS	-	15	F.SESMG	-	361	F.PRET	-	401	L.OMYGO	-	117
E.MODPC	-	247	I.SESMG	-	6	I.PRET	-	4	ENVIORN	-	36
L.MODPC	-	13	E.JMRET	-	367	SRETN	-	17			

EXTERNAL SYMBOLS.

E.ECS	MAPOUT	I.LOCK	S.ECSRA	NEGPT	PUTCAP	MACSZ	BTGPAP
DISASTR	MAPIN	TOSPROC	BDTBLK	ESTK	SYSFRET	COMP	NEGPAP
REFER	S.USB1	NEWTOS	ROOM	PUTRETP	NOFIND	NOFATH	S.CMFLI
ENDSYS	S.USERA	E.ERROR	NEGIX	FRRPROC	INSTK	SAMNA	
REALLOC	SYSRET	PARMBUF	BIGIX	CHKPTR	NLEAF	CLMOT	

```

IDENT      SUBPROC
*
* SYSTEM ACTIONS HEREIN -
*   CRESUBP  CREATE A SUBPROCESS
*   DELSUBP  DELETE A SUBPROCESS
*   JUMPCAL  SUBPROCESS JUMP CALL
*   SUBPCAL  NORMAL SUBPROCESS CALL
*   MODPC    MODIFY P-COUNTER OF SUBP IN STACK
*   FSON     FIND SON OF GIVEN SUBP
*   DSPSP    DISPLAY SUBPROCESS DESCRIPTOR
*   ERRCAL   USER INITIATED ERROR (RETURN WITH ERROR)
*   ESMSETN  SET ERROR SELECTION MASK, ANY SUBPROCESS
*   ESMSET   SET ERROR SELECTION MASK, CURRENT SUBPROCESS
*   JMRET    JUMP RETURN
*   SPRET    RETURN AND REDO OPERATION
*   FRET     USER INITIATED F-RETURN
*   PRET     RETURN WITH PARAMETERS
*   ARM      ARM INTERRUPT INHIBIT ON A SUBPROCESS
*   DSARM    DISARM ETC
*
* EXTERNAL SUBROUTINES CONTAINED HEREIN -
*   FINSUB   LOCATE SUBPROCESS GIVEN ITS CLASS CODE
*   ENVIORN  ESTABLISH ENVIRONMENT OF TOP OF STACK
*
* INTERNAL SUBROUTINES CONTAINED HEREIN -
*   RELOCAT  DIDDLE PTRS IN SUBP TABLE DURING CREATION/DELETION
*   FINDNTHS SEARCH STACK FOR NTH OCCURANCE OF GIVEN SUBP
*   OMYGOD   PROCESS METAMORPHOSIS

```

0
0
0

```

INTSYS      XTEXT
ERRNUMS     XTEXT
PROCSYM     XTEXT
EXT         E.ECS,DIS,STR
EXT         REFER,ENDSYS,REALLOC,MAPOUT,MAPIN
EXT         S,USRB1,S,USERA,SYSPRET
WECS        MACRO
+           WE
           RJ
WECS        ENDM
RECS        MACRO
+           RE
           RJ
RECS        ENDM
ECSMAC      XTEXT
CBLOCK      MICRO
EXT         1,*/ESUBP/*
           T.LOCK

```

0

*
* THIS SYSTEM ACTION CREATES A SUBPROCESS. PARAMETERS EXPLAINED AND
* MNEMONICIZED BELOW.
*

66		ORG	D.PARAM	
			PARAMETER	MNEUMONICS
66	SP.CLCD	BSS	2	SUBPROCESS NAME (CLASS CODE)
70	SP.FACC	BSS	2	NAME OF SUBPROCESS FATHER (CLASS CODE)
72	SP.MAP	BSS	1	NUMBER OF MAP ENTRIES
73	SP.COMB	BSS	1	SPACE FOR COMPILED MAP
74	SP.FL	BSS	1	SUBPROCESS FIELD LENGTH
75	SP.ENTRY	BSS	1	SUBPROCESS ENTRY POINT
76	SP.CLIST	BSS	2	SUBPROCESS LOCAL C-LIST
		USE	*	
63	LOCALC	EQU	D.TEMP6	. LENGTH OF LOCAL C-LIST
64	FATHPTR	EQU	D.TEMP7	. PTR TO FATHER IN OLD DESC
50	V1	EQU	D.SCR2+0	. INCREASE IN LOCAL C-LIST BUFFER
51	V2	EQU	D.SCR2+1	. INCREASE IN FULL C-TABLE (CELLS)
52	V3	EQU	D.SCR2+2	. INCREASE IN STACK SIZE (CELLS)
53	V4	EQU	D.SCR2+3	. INCREASE IN SUBP DESC TABLE
54	V5	EQU	D.SCR2+4	. INCREASE IN FULL PATH FL
55	ADR4	EQU	D.SCR2+5	. ADDRESS WHERE SUBP ADDED OR DELETED

0

ECSCODE MRSUB

*
* CHECK PARAMETERS: SEE IF PROCESS EXPANSION IS LEGAL, ETC.
* ALSO, CALCULATE CHANGES IN SIZE OF DIFFERENT PARTS OF PROCESS DESC.
*

L	51	5131000067	CRESUB0	SA3	B1+SP.CLCD+1	
		6170000053		SB7	CRESUB1	. TEST FOR DUPLICATE SUBPROCESS NAME
L	52	0400000025 +		EQ	FINDSUB	
L	53	0560000200	CRESUB1	NE	R6,B0,CRESUB26	. JUMP IF DUPLICATE NAME EXISTS
		5131000071		SA3	B1+SP.FACC+1	
L	54	6170000055		SB7	CRESUB2	. FIND FATHER
		0400000025 +		EQ	FINDSUB	
L	55	0460000201	CRESUB2	EQ	R6,B0,CRESUB27	. JUMP IF FATHER DOESNT EXIST
		75651		SX6	A5-B1	. SAVE PTR TO FATHER, REL B1
		75761		SX7	A6-B1	. SAVE PTR TO LOCATION FOR NEW SUBP
L	56	7266777776		SX6	X6-SD.CC	
		5161000064		SA6	R1+FATHPTR	
L	57	7277777776		SX7	X7-SD.CC	
		5171000055		SA7	B1+ADR4	
L	60	5015000006		SA1	A5-SD.CC+SD.RAFL	RA,FL WORD OF FATHER
		20130		LX1	6+18	
L	61	5141000074		SA4	R1+SP.FL	
		0334000173		NG	X4,CRESUB34	. ERROR IF PROPOSED FL NEGATIVE
L	62	73110		SX1	X1	. RA+FL OF FATHER
		36114		IX1	X1+X4	. PROPOSED RA+FL OF NEW SUBP
		5131000126		SA2	B1+P.ROHEAD	. FETCH CURRENT LENGTH OF PROCESS

SUBPROCESS OPERATIONS
SUBPROCESS CREATION

COMPASS - VER 2.
FSUBP

11/03/71 15.37.56.

PAGE

4

```

L 63 73630
      37616
      0324000065
L 64 13666
L 65 5161000054
      5111000075
L 66 7221777771
      0332000174
L 67 37114
      0321000176
L 70 5015000003
      5121000162
L 71 20130
      73110
      20230
      73220
L 72 37121
      20173
      7211000001
L 73 5121000161
      20252
      73220
L 74 37121
      13666
      0321000076
L 75 7160000002
L 76 5161000051
      5111000073
L 77 73110
      7221777774
L 100 0332000202
      5121000072
L 101 7242777776
      0334000175
L 102 73220
      73320
      20301
      36323
L 103 36113
      7261000071
L 104 5161000053
      5121000077
L 105 73020
      56010
L 106 0110000001
L 107 54300
      43647
      13223
      11262
L 110 0312000177
      15036
L 111 0110000001
L 112 54200

```

CRSUB3

CRSUB4

```

SX6      X3
IX6      X1-X6
PL      X6,CRSUB3      . SKIP IF INCREASE
BX6      X6-X6      . SUBSTITUTE 0 INCREASE
SA6      R1+V5      . SAVE INCREASE
SA1      R1+SP.ENTRY . CHECK OUT PROPOSED ENTRY POINT
SX2      X1-6      . ERROR IF ENTRY LESS THAN 6
NG      X2,CRSUB28
IX1      X1-X4
PL      X1,CRE28X . ERROR IF ENTRY GE FL
SA1      A5-SD,CC+SD.PTRS . TEST FOR INCREASE IN FULL C-TABLE
SA2      R1+P.STACK
LX1      6+18
SX1      X1      . MAX STACK PTR OF FATHER
LX2      6+18      . STACK END
SX2      X2
IX1      X2-X1      . 2=ROOT, 4=SON OF ROOT, ...
LX1      60-1
SX1      X1+1      . PATH LENGTH FROM ROOT TO NEW SUBP
SA2      R1+P.CTABLE
LX2      6+36      . CURRENT MAX PATH
SX2      X2
IX1      X2-X1
BX6      X6-X6      . PROVISIONAL 0 INCREASE
PL      X1,CRSUB4      . JUMP IF NO INCREASE
SX6      2      . INCREASED SPACE REQUIRED FOR C-TABLE
SA6      R1+V2
SA1      R1+SP.COMP . CHECK OUT COMPILED MAP SIZE
SX1      X1
SX2      X1-3      . MUST BE AT LEAST 3 TO HOLD EVEN 1 ENTRY
NG      X2,CRSUB31
SA2      R1+SP.MAP
SX4      X2-1      . MUST BE AT LEAST 1 MAP ENTRY
NG      X4,CRSUB35
SX2      X2
SX3      X2
LX3      1
IX3      X2+X3      . SPACE REQUIRED BY LOGICAL MAP
IX1      X1+X3      . SPACE FOR LOGICAL MAP + COMPILED MAP (ALMOST)
SX6      X1+1+SD:SIZE . ONE FOR 0 AT END OF LOG MAP +SPDESC
SA6      R1+V4      . SAVE INCREASE IN SUBP DESC TABLE
SA2      R1+SP.CLIST+1
SX0      X2      . COMPUTE SIZE OF LOCAL C-LIST
SA0      B1
RECS      1      . READ CLIST MOT
SA3      A0
MX6      39
BX5      X2-X3
BX2      X6*X2      . CHECK UN
NZ      X2,CRSUB33 . C-LIST DOESNT EXIST
BX0      -X6*X3      . ECS A(C-LIST)
RECS      1      . READ LENGTH
SA2      A0

```


10622
L 113 5131000161 5161000063
20330
73330
L 114 37232 13633
0322000116
L 115 14622 20601
5161000050
L 116 5161000050 5141000054
L 117 36446 5111000053
36441
L 120 5121000051 36442
L 121 7120000000 x 5130000000 x
L 122 37232 5131000126
73330
L 123 37223 37224
0332000203
L 124 7160000127 5111000053
L 125 5121000126 20252
73220
L 126 0400000000 x
L 127 5111000164 6170000132
L 130 6130000006
L 131 63211 56523
0400000000 x
L 132 0306000135 0330000135
L 133 56120 43601
20673
12616
L 134 54610
L 135 5132000004 43652
11663

BX6 X2 . SAVE LENGTH
SA6 R1+LOCALC
SA3 R1+P.CTABLE
LX3 6+18 . COMPARE WITH CURRENT MAX C-LIST SIZE
SX3 X3
IX2 X3-X2
BX6 X3-X3 . PROVISIONAL 0 INCREASE
PL X2,CRESUB7 . JUMP IF NO INCREASE
BX6 -X2 . CALCULATE INCREASE IN CELLS
LX6 1
SA6 R1+V1 . SAVE INCREASE IN LOCAL C-LIST BUFFER
SA4 R1+V5 . ADDITIONAL SPACE FOR SUBPROCESS CORES
IX4 X4+X6 . V1+V5
SA1 R1+V4
IX4 X4+X1 . V1+V4+V5
SA2 R1+V2
IX4 X4+X2 . V1+V2+V4+V5 (V3 IS 0)
SX2 ENDSYS . CALCULATE CORE AVAILABLE
SA3 =XS.CMFL
IX2 X3-X2 . AVAILABLE LESS CURRENTLY USED
SA3 R1+P.ROHEAD
SX3 X3 . SPACE CURRENTLY USED
IX2 X2-X3
IX2 X2-X4 . LESS NEEDED ADDITIONAL SPACE
NG X2,CRESUB32 . JUMP IF TOO BIG

*
* ALLOCATE SPACE FOR NEW VERSION OF PROCESS DESCRIPTOR
*

SX6 CRESUB8 . RETLINK
SA1 R1+V4 . ADDITIONAL ECS NEEDED
SA2 R1+P.ROHEAD . GET PROCESS MOT
LX2 6+36
SX2 X2
EQ =XREALLOC

*
* SWAP OUT ALL MAPS CURRENTLY IN
*

CRESUB8 SA1 R1+P.MAPSIN . HEAD OF MAPIN CHAIN
SB7 CRESUB95 . RETLINK
SB3 SD.MAP . NICE CONSTANT
CRESUB9 SB2 R1+X1 . ABS A(SUBP DESC)
SA5 R2+B3 . MAP STUFF
EQ =XMAPOUT . SWAP ONE OUT
CRESUB95 ZR X6,CRESUB10 . NO ANNOYING ERRORS
NG X6,CRESUB10 . WELL, IT WAS JUST HIS MAP TURNED OFF
SA1 R2 . DARN! HAVE TO FLAG HIM FOR A
MX6 1 . PENDING MAP ERROR
LX6 60-SD.FMPE
BX6 X1+X6
SA6 A1
SA3 R2+SD.PTRS . FOLLOW THE MAPIN CHAIN
MX6 42
BX6 X6*X3 . CLEAR OLD POINTER

L 136 54630
73130
0321000131

SA6 A3
SX1 X3
PL X1,CRESUB9

. NEG POINTER ENDS CHAIN

*
* CONSTRUCT NEW SUBP DESCRIPTOR AT END OF TABLE

* SD.FIPT VFD 36/0,18/C-LIST LENGTH,18/NEXT PTR

L 137 5111000055
63611
L 140 5121000063
5131000053
L 141 20222
12623
5166000000

SA1 R1+ADR4 . WHERE FIRST WORD OF NEW SUBP DESC GOES
SB6 X1+B1
SA2 R1+LOCALC . LENGTH OF CLIST
SA3 R1+V4 . LENGTH OF NEW SUBP DESC
LX2 18
BX6 X2+X3
SA6 R6+SD.FIPT

* SD.CC VFD 60/CLASS CODE

L 142 5111000067
10611
L 143 5166000001

SA1 R1+SP.CC+1
BX6 Y1
SA6 R6+SD.CC

* SD.ESM VFD 60/0(NO ERRORS ACCEPTED)

L 144 5166000002
13646

BX6 X6-X6
SA6 R6+SD.ESM

* SD.CLIST VFD 60/UNIQUE NAME, MOT OF LOCAL C-IST

L 145 10611
5111000077
5166000003

SA1 R1+SP.CLIST+1
BX6 Y1
SA6 R6+SD.CLIST

* SD.PTRS VFD 6/0,18/MAX STACK,18/FATHER,18/0(MAP NOT IN)

L 146 5111000064
63211
L 147 5122000004
20230
L 150 7262777775
20622
12616
L 151 20622
5166000004

SA1 R1+FATHERPTR . RETRIEVE PTR TO FATHER
SB2 X1+B1 . ABS A(FATHER)
SA2 R2+SD.PTRS . GET PTR WORD OF FATHER
LX2 6*18 . FATHERS MAX STACK
SX6 X2-2 . SONS MAX STACK
LX6 18
BX6 X1+X6 . STICK IN FATHER PTR
LX6 18
SA6 R6+SD.PTRS

* SD.Orig VFD 6/0,18/ENTRY POINT,18/MAP ORIGIN,18/C-LIST ORIGIN

L 152 5111000075
20122
L 153 5122000005
5132000000
L 154 20252
20352

SA1 R1+SP.ENTRY
LX1 18
SA2 R2+SD.Orig . GET FATHERS ORIGINS
SA3 R2+SD.MAP
LX2 6*36 . FATHERS MAP ORIGIN
LX3 6*36 . LENGTH OF FATHERS MAP

63320
73633
L 155 12661 5132000000
20352
L 156 20222 63330
73253
20622
L 157 12662 5166000005

*
*
*
*

L 160 5111000073 10611
20622
L 161 5121000072 12626
20622
L 162 7211000002 12641
L 163 5166000006

*
*
*

L 164 20130 5112000007
73610
L 165 63320 5121000074
73213
20222
12662
L 166 20622 5111000073
L 167 7211777775 12641
L 170 5166000007

*
*
*

L 171 13666 6170000000 x
5161000052
L 172 0400000033 +

S83 X2
SX6 X3+B3
BX6 X6+X1
SA3 R2+SD.FIPT . ADD FATHERS C-LIST LENGTH
LX3 6+36 . TO FATHER C-LIST ORIGIN TO GET
LX2 18
S83 X3
SX2 X2+B3
LX6 18 . SONS C-LIST ORIGIN
BX6 X6+X2
SA6 R6+SD.ORIG

SD.MAP VFD 6/0,18/COMP MAP SIZE,18/ELOG MAP ENTRIES
18/PTR TO LOG MAP, REL THIS WORD

SA1 R1+SP.COMP
BX6 X1
LX6 18
SA2 R1+SP.MAP
BX6 X2+X6
LX6 18
SX1 X1+2
BX6 X6+X1
SA6 R6+SD.MAP

SD.RAFL VFD 6/0,18/RA+FL,18/RA,18/SPACE LEFT IN COMPILED MAP

SA1 R2+SD.RAFL . GET FATHERS RA,FL WORD
LX1 6+18
SX6 X1 . FATHERS RA+FL IS SONS RA
SA2 R1+SP.FL
S83 X2
SX2 X1+B3
LX2 18
BX6 X6+X2
LX6 18
SA1 R1+SP.COMP
SX1 X1-2 . AVAILABLE SPACE IS SPECIFIED SIZE-2
BX6 X6+X1 . STICK IN COMPILED SPACE LEFT
SA6 R6+SD.RAFL

* NOW GET OMYGOD TO TRANSFORM THE OLD DESCRIPTOR INTO THE NEW ONE

*

S87 =XSYSRET
BX6 X6-X6
SA1 R1+V3 . STACK DIENT CHANGE SIZE
EQ OMYGON

*

ERRORS

SUBPROCESS OPERATIONS
SUBPROCESS CREATION

COMPASS - VER 2.
FSUBP

11/03/71 15.37.58.

PAGE 8

		*			
		*			ERRORS
173	6140000005	CRESUB34	ERROR	5,NEGP	AR
174	6140000006	CRESUB38	ERROR	6,NEGP	AR
175	6140000003	CRESUB35	ERROR	3,NEGP	AR
176	6140000006	CRE28X	ERROR	6,BIGP	AR
177	6140000007	CRESUB33	ERROR	7,CLMOT	
200	6140000001	CRESUB26	ERROR	1,SAMNA	
201	6140000002	CRESUB27	ERROR	2,NOFATH	
202	6140000004	CRESUB31	ERROR	4,COMP	
203	6140000000	CRESUB32	ERROR	0,MACSZ	
204		END ECS		MKSUB	

133
L 51 5131000067 6170000053
L 52 0400000025 +
L 53 04600000147 6025777776
L 54 5112000004 20152 63710
L 55 04700000143

L 56 63320 51210000163
20252
63750
67221

L 57 6141000004
L 60 56147 20152
63510
L 61 04520000143 56117 63717
L 62 6133777776 0703000060

L 63 51110000162 6271000001
L 64 20152 6231000001
L 65 56117 63510 04520000144
L 66 6177777775 0673000065

L 67 51110000164 6170000072

```
*
* SYSTEM ACTION TO DELETE A SUBPROCESS. THE SUBP MUST BE A LEAF ON
* THE SUBPROCESS TREE; THE ROOT CANNOT BE DELETED EVEN IF IT IS A LEAF.
*
*      PARAM = C: CLASS CODE (SUBPROCESS NAME)
*
*      ECSCODE  DELSUB
*
*      DELSUB  SA3      R1+SP,CLCD+1      . GET SUBPROCESS NAME
*              SB7      DELSUB1      . RETLINK
*              EQ        FINDSUB      . LOOK UP SUBPROCESS
*      DELSUB1 EQ        R6,B0,DELSUB29      . ERROR - NO SUCH SUBP
*              SB2      R5-SD,CC      . INDEX OF SUBP
*              SA1      R2+SD,PTRs      . CHECK FOR ROOT
*              LX1      60-18
*              SB7      X1
*              EQ        R7,B0,DELSUB31      . ERROR - CANT DELETE ROOT
*
*      MAKE SURE DELSUBP IS A LEAF
*
*              SA2      R1+P.SUBPDT
*              SB3      X2      . NUMBER OF SUBPS
*              LX2      60-18
*              SB7      X2      . INDEX OF FIRST SUBP
*              SB2      R2-B1      . INDEX OF DELSUBP
*      * B2 = INDEX OF DELSUBP, REL R1
*              SB4      R1+SD,PTRs
*      DELSUB2 SA1      R4+B7      . GET FATHER PTR
*              LX1      60-18
*              SB5      X1
*              EQ        R5,B2,DELSUB31      . ERROR - DELSUBP NOT LEAF
*              SA1      R1+B7      . FOLLOW PTR TO NEXT SUBP
*              SB7      X1+B7
*              SB2      R3-1      . TALLY SUBPS
*              GT        R3,B0,DELSUB2
*
*      CHECK FOR DELSUBP IN STACK
*
*              SA1      R1+P.STACK
*              SB7      X1+1      . TOP OF STACK PTR
*              LX1      60-18
*              SB3      X1+1      . BOTTOM OF STACK PTR
*      DELSUB3 SA1      R1+B7      . STACK ENTRY
*              SB5      X1
*              EQ        R5,B2,DELSUB27      . ERROR - DELSUBP IN STACK
*              SB7      R7-2      . MOVE TO NEXT STACK ENTRY
*              GE        R7,B3,DELSUB3      . FINISH STACK
*
*      SWAP OUT ALL MAPS
*
*              SA1      R1+P.MAPSIN      . HEAD OF MAP CHAIN
*              SB7      DELSUB4      . RETLINK FROM MAPOUT
```

SUBPROCESS OPERATIONS
DELETE SUBPROCESS

COMPASS - VER 2.
FSUBP

11/03/71 15.37.59.

PAGE 10

L 70 6130000006
L 71 63411 56543 0400000000 x
L 72 0306000075 0334000075
L 73 56140 43601 20673 12661
L 74 54610
L 75 5134000004 43652 11663
L 76 54630 73130 0321000071

DELSUB5 SB3 SD.MAP
SB4 R1+X1 . SUBP DESC. ABSOLUTE
SA5 R4+B3 . MAP STUFF
EQ =XMAPOUT . SWAP ONE SUBP OUT
DELSUB4 ZR X6,DELSUB6 . NO MAP ERRORS AT ALL
NG X6,DELSUB6 . WELL, ONLY A MAP OFF ERROR
SA1 R4 . DARN! HAVE TO FLAG HIM FOR A
MX6 1 . PENDING MAP ERROR
LX6 60-SD.FMPE
BX6 X6+X1
SA6 A1
DELSUB6 SA3 R4+SD.PTR . FOLLOW MAP CHAIN
MX6 42
BX6 X6+X3 . CLEAR MAP IN FLAG
SA6 A3
SX1 X3
PL X1,DELSUB6 . NEG PTR ENDS CHAIN

*
* DECREMENT REFERENCE COUNTS ON MAP OF DELSUBP
*

L 77 6132000006 56113 63313
L 100 7163777774 5161000056
L 101 43573 6140000145
L 102 6160000103
L 103 5111000056 7211000003
L 104 53211 0312000107
L 105 0332000111 7211000003
L 106 0400000104
L 107 10611 54610 7146377777
L 110 6170000146 0400000000 x

SB3 R2+SD.MAP
SA1 R1+B3 . GET LOG MAP PTR
SB4 X1+B3
SX6 R3-3 . PTR TO FIRST LOG MAP -3, REL B1
SA6 B1+P.TEMP1
MX5 59 . INCREMENT COUNTS BY -1
SB4 DELSUB28 . DISASTER IF BLOCK MISSING
SB6 DELSUB7 . NORMAL RETURN LINK
DELSUB7 SA1 R1+P.TEMP1 . RECOVER LAST LOG MAP ADDR
SX1 X1+3
DELSUB8 SA2 R1+X1 . CHECK FOR SWAPPING DIRECTIVE
NZ X2,DELSUB9 . LIVE ENTRY
NG X2,DELSUB10 . END OF MAP
SX1 X1+3 . GET NEXT ENTRY
EQ DELSUB8
DELSUB9 BX6 X1
SA6 A1 . SAVE NEW ENTRY
SX4 3777778 . HUGE FL
SB7 DELSUB37 . ERROR IF FILE GONE, ETC
EQ =XREFER . DECREMENT COUNTS

*
* CHECK FOR DECREASE IN PROCESS SIZE DUE TO
* 1) LOCAL C-LIST
* 2) FULL C-TABLE
* 3) SUBP ADDRESS SPACE
*

L 111 5111000163 73710 20152
L 112 63510 66300 66400 66600

DELSUB10 SA1 R1+P.SUBPDT
SX7 X1 . NUMBER OF SUBPS
LX1 60-18
SB5 X1 . ORIG OF SUBP TABLE
SB3 R0 . PROVISIONAL MAX C-LIST
SB4 R0 . PROVISIONAL MAX DEPTH OF TREE
SB6 R0 . PROVISIONAL MAX RA+FL

SUBPROCESS OPERATIONS
DELETE SUBPROCESS

COMPASS - VER 2.
FSUBP

11/03/71 15.37.59.

PAGE 11

L	113	56115	DELSUB11	SA1	R1+B5	
		0452000122		EQ	R5,B2,DELSUB14	. SKIP DELSUBP
L	114	5021000007		SA2	A1+SD,RAFL	. GET RA,FL WORD
		20230		LX2	60-36	
		63720		SB7	X2	. RA+FL
L	115	0667000116		LE	R7,B6,DELSUB12	. SKIP IF NOT NEW MAX
		66670		SB6	B7	. NEW MAX
L	116	5021000004	DELSUB12	SA2	A1+SD,PTRS	. GET WORD WITH DEPTH IN TREE
		20230		LX2	60-36	
		63720		SB7	X2	. MAX STACK PTR (INDICATES DEPTH)
L	117	0647000120		LE	R7,B4,DELSUB13	. SKIP IF NOT NEW MAX
		66470		SB4	R7	. NEW MAX
L	120	10211	DELSUB13	BX2	X1	. GET LENGTH LOCAL C-LIST
		20252		LX2	60-18	
		63720		SB7	X2	
L	121	0637000122		LE	R7,B3,DELSUB14	. SKIP IF NOT NEW MAX
		64370		SB3	B7	. NEW MAX
L	122	63515	DELSUB14	SB5	R5+X1	. ADVANCE TO NEXT SUBP
		7277777776		SX7	X7-1	. TALLY SUBP
L	123	0317000113		NZ	X7,DELSUB11	. ENOUGH IS ENOUGH
		54112		SA1	R1+B2	. GET LENGTH OF DELSUBP LOCAL C-LIST
		20152		LX1	60-18	
L	124	63710		SB7	X1	
		43600		MX6	0	. ASSUME NO CHANGE
		0637000126		LE	R7,B3,DELSUB15	. JUMP IF NO CHANGE
L	125	77637		SX6	R3-B7	. -(NUMBER OF SLOTS FREED)
		20601		LX6	1	. GIVES WORDS FREED
L	126	5161000050	DELSUB15	SA6	R1+V1	. SET CHANGE IN LOCAL C-LIST BUFFER
		5021000004		SA2	A1+SD,PTRS	. GET DEPTH OF DELSUBP IN TREE
L	127	20230		LX2	60-36	
		63720		SB7	X2	. MAX STACK PTR INDICATES DEPTH
		43600		MX6	0	. ASSUME NO CHANGE
L	130	0647000131		LE	R7,B4,DELSUB16	. JUMP IF NO CHANGE
		77647		SX6	R4-B7	. WORDS SAVED IN FULL C-TABLE
L	131	5161000051	DELSUB16	SA6	R1+V2	. SET CHANGE IN FULL C-TABLE
		43600		MX6	0	
L	132	5161000052		SA6	R1+V3	. NO CHANGE IN STACK
		20122		LX1	18	
		73610		SX6	X1	. SIZE OF THIS SUBP DESCRIPTOR
L	133	14666		BX6	-X6	. SIZE DECREASING OF CORSE
		5161000023		SA6	R1+V4	. SET CHANGE IN SUBP TABLE SIZE
L	134	5021000007		SA2	A1+SD,RAFL	
		20230		LX2	60-36	
		63720		SB7	X2	. RA+FL OF DELSUBP
L	135	43600		MX6	0	. ASSUME NO CHANGE
		0776000136		LT	R7,B6,DELSUB17	. JUMP IF NO CHANGE
		77667		SX6	R6-B7	
L	136	5161000054	DELSUB17	SA6	R1+V5	. SET CHANGE IN MAX FULL PATH CORE
		76620		SX6	B2	
L	137	5161000055		SA6	R1+ADR4	. SET INDEX WHERE SUBP DISAPPEARED

*
* REALLOCATE PROCESS DESCRIPTOR IN ECS
*

SUBPROCESS OPERATIONS
DELETE SUBPROCESS

COMPASS - VER 2.
FSUBP

11/03/71 15.38.00.

PAGE 12

			5111000053
L	140	7160000142	
			5121000126
L	141	20252	
		73220	
			0400000000 X
L	142	6170000000	X
			0400000033 +

SA1
SX6
SA2
LX2
SX2
EQ
SB7
EQ

```

R1+V4      . CHANGE IN ECS SIZE OF DESCRIPTOR
DELSUB18   . RETLINK
R1+P.ROHEAD . GET PROCESS MOT
60-18
X2
=XREALLOC
=XSYSRET   . METAMORPHOSE THE PROCESS AND RETURN TO THE
OMYGOD     . USER

```

* MISC ERRORS, DISASTERS

143	61400000001	
144	61400000001	
145	01000000000	X
146	01000000000	X
147	61400000001	
150		

DELSUB31	ERROR
DELSUB27	ERROR
DELSUB28	RJ
DELSUB27	RJ
DELSUB29	ERROR
	ENDECS

```

I: NLEAF      • DELSUBP IS ROOT OR IS NOT A LEAF
I: INSTK      • DELSUBP IS IN STACK
=XDISASTR     • BLOCK GONE FROM FILE IN MAP
=XDISASTR     • FILE GONE FROM MAP (NOT DISASTER WITH 2CPS)
I: NOFIND     • DELSUBP DOESNT EXIST
DELSUB

```



```

*
*
* THIS IS THE ECS ACTION OF SUBPROCESS
* JUMP CALL. ALL IS AS WITH NORMAL CALL
* EXCEPT THAT WE HAVE TO MAKE IT SEEM AS
* THOUGH THE CALLER WERE THE CURRENT SUBPROCESS
* IN THE PREVIOUS STACK ENTRY.
*
0 5131000125          6170000002 +
1 0200000025 +
2 0460000013 +
JUMPCAL ENTRY
SA3      JUMPCAL
SB7      R1+P.PARAMC . GET NAME OF SUBPROCESS
JP       JPCAL1 . RETLINK
EQ       FINDSUB . LOCATE SUBP DESCRIPTOR
EQ       R6,B0,SUBPC15 . JUMP IF NO SUCH SUBP
*
* WE CAN DECREMENT THE STACK
*
3 7160000002          5111000162
          37616
          54610
SA1      R1+P.STACK . THIS MAKES IT SEEM LIKE
SX6      2 . THE CALLER WAS THE GUY IN THE PREVIOUS
IX6      X1-X6 . STACK ENTRY
SA6      A1
*
* NOW WE MAKE A NEW TOP OF STACK AND EXIT TO THE TOP OF STACK
* PROCESSOR BY JUMPING INTO THE REGULAR CALL MECHANISM
*
4 0400000010 +
EQ       SUBPC2

```

```

*
*
* THIS IS THE ECS ACTION OF SUBPROCESS CALL
* THE ACTUAL PARAMETER LIST FOR THE SUBPROCESS CALL
* HAS BEEN FORMED
* THE ONLY INPUT IS R1 = PROCESS ORIGIN
*
ENTRY SUBPCAL
EXT TOSPROC, NEWTOS, E, ERROR
*
5 5131000125 SUBPCAL SA3 R1+P, PARAMC GET NAME OF SUBPROCESS
6 0200000025 + SB7 SUBPC1
7 0460000013 + JP FINDSUB
SUBPC1 EQ B6, B0, SUBPC15 ERROR, NO SUCH SUBPROCESS
*
* MAKE A NEW TOP OF STACK FOR THIS SUBP AND PROCESS IT
*
10 65251 SUBPC2 SB2 A5-B1 . PTR TO SUBP, REL B1
6122777776 SB2 B2-SD, CC
66300 SB3 B0 . ENTRY TYPE = CALL
11 6160000000 Y SB6 TOSPROC . ERROR RETLINK
6170000000 Y SB7 TOSPROC . RETLINK
12 0400000000 Y EQ NEWTOS
*
*
13 6140000000 SUBPC15 ERROR R, NO FIND

```

```
*
* FIND NTH OCCURANCE OF SPECIFIED SUBPROCESS ON STACK.
* AT ENTRY
*      B7 = RETLINK
*      AP1 = C:CLASS CODE SPECIFYING SUBPROCESS
*      AP2 = D:SKIP COUNT  -1= OCCURANCE NEAREST BOTTOM OF STACK
*                        0= OCCURANCE NEAREST TOP OF STACK
*                        1= NEXT TO TOP OCCURANCE, ETC
*
```

```
*AT EXIT
*      B4 = INDEX OF DESIRED STACK ENTRY; GENERATES ERROR IF
*          SPECIFIED ENTRY DOES NOT EXIST
*
```

```
* DOES NOT USE A0/X0, A4/X4, A5/X5, A6/X6, A7/X7, B7
*
```

14			FINDNTAS	BSS	0	
232				ECSSUB	0	FINDS, BUFA
450	5111000070			SA1	B1+P, PARAM+2	
		6221000001		SB2	X1+1	MODIFIED SKIP COUNT
451	0720000463			LT	R2, B0, FNS4	COUNT LESS THAN LEGAL
		66400		SB4	B0	PROVISIONAL STACK INDEX
452	5121000162			SA2	R1+P, STACK	
		63521		SB5	R1+X2	TOP OF STACK
		20252		LX2	60-18	
453	63321			SB3	R1+X2	BOTTOM OF STACK
		5111000047		SA1	R1+P, PARAM+1	DESIRED CLASS CODE
454	56350		FNS1	SA3	B5	FETCH STACK ENTRY
		63631		SB6	R1+X3	
		5134000001		SA3	R6+SD, CC	FETCH CLASS CODE
455	13331			BX3	X3-X1	
		0313000440		NZ	X3, FNS2	JUMP IF NOT SPECIFIED CLASS CODE
456	6122777776			SB2	B2-1	COUNT OCCURANCES
		67451		SB4	B5-B1	REMEMBER INDEX FOR -1 CASE
457	0420000462			EQ	R2, B0, FNS3	JUMP IF DESIRED OCCURANCE
460	6155777775		FNS2	SB5	B5-2	ADJUST STACK ENTRY
		0654000454		GE	R5, B3, FNS1	CONTINUE TO BOTTOM OF STACK
461	0620000464			GE	R2, B0, FNS5	SUBP NOT IN STACK
462	0270000000		FNS3	JP	B7	
463	6140000002		FNS4	ERROR	2, NEGBAR	
464	6140000002		FNS5	ERROR	2, BIGPAR	
465				ENDSUB	FINDS, BUFA	

*
* THIS SYSTEM ACTION SETS THE P-COUNTER OF A SPECIFIED OCCURANCE OF
* A SPECIFIED SUBPROCESS IN THE STACK. THE P-COUNTER QUALIFIER IS SET TO
* ABOUT TO EXECUTE.
*

* PARAMETERS

* AP1 C:CLASS CODE SPECIFYING SUBPROCESS
* AP2 D:OCCURANCE OF INTEREST IN STACK
* -1= EARLIEST (BOTTOM MOST) OCCURANCE
* 0= LATEST (TOP MOST) OCCURANCE
* 1= NEXT TOP TOP OCCURANCE
* ETC
* AP3 D:NEW VALUE FOR P-COUNTER
*

247			ECSCODE	MODPC
L 51	6170000052		SB7	*+1
		0400000014 *	EQ	FINDNTHS
L 52	56114		SA1	B1+B4
		5021000001	SA2	A1+1
		63521	SB5	X2+B1
L 53	5135000007		SA3	B5+SD,RAFL
		20352	LX3	60-18
		63230	SB2	X3
L 54	20352		LX3	60-18
		63330	SB1	X3
		67332	SB3	B3-B2
L 55	5141000071		SA4	B1+P.PARAM+3
		63240	SB2	X4
L 56	0720000062		LT	B2,B0,MODPC1
		0623000063	GE	B2,B3,MODPC2
L 57	43652		MX6	42
		11161	BX1	X6*X1
		43651	MX6	1
		20670	LX6	60-SF.PCQ1
L 60	12116		BX1	X1+X6
		20673	LX6	SF.PCQ1+60-SF.PCQ2
		15116	BX1	-X6*X1
		76620	SX1	B2
L 61	12661		BX6	X6+X1
		54610	SA6	A1
		0400000000 X	EQ	=XSYSRET

L 62	6140000003	MODPC1	ERROR	3,NEGP
L 63	6140000003	MODPC2	ERROR	3,BIGP
L 64			ENDECS	MODPC

*
* FIND NTH SON OF SPECIFIED SUBPROCESS
* F-RETURNS IF N TOO LARGE; ERROR IF N .IE. 0 OR FATHER DOESNT EXIST.
*

* PARAMETERS

* AP1 C:CLASS CODE OF FATHER
* AP2 D:N=1 FOR ELDEST SON;
* =2 FOR NEXT ELDEST, ETC
* AP3 D:LIST INDEX TO RETURN CAP FOR SON
*

262			ECSCODE	FSON	
L 51	5131000067		SA3	R1+P.PARAM+1	. LOCATE FATHER
	6170000053		SB7	FSON1	
L 52	0400000025 +		EQ	FINDSUB	
L 53	0460000067	FSON1	EQ	R6,B0,FSON10	. NO SUCH FATHER
	6025777776		SB2	A5-SD.CC	. FATHER PTR, ABS
L 54	67221		SB2	R2-B1	. FATHER, DEL B1
	5111000070		SA1	R1+P.PARAM+2	
	63310		SB3	X1	. N, AS IN NTH SON
L 55	0603000070		LE	R3,B0,FSON11	. NO, NO
	5111000163		SA1	R1+P.SUBPDT	
L 56	20152		LX1	60-18	
	63511		SB5	X1+B1	. PTR TO ROOT SUBP, ABS
L 57	56150	FSON2	SA1	R5	. FIRST WORD OF SUBP
	0301000000 X		ZR	X1,=XSYSFRET	. END OF TABLE, NO NTH SON
	63515		SB5	R5+X1	. ADVANCE SUBP PTR
L 60	5125000004		SA2	R5+SD.PTRs	
	20252		LX2	60-18	
	63620		SB6	X2	. FATHER PTR
L 61	0562000057		NE	R6,B2,FSON2	JUMP IF NOT A SON
	6133777776		SB3	R3-1	. COUNT SONS
L 62	0530000057		NE	R3,B0,FSON2	JUMP IF NOT N FOUND
	5011000001		SA1	A1+SD.CC	. SUBSTITUTE SON CLASS CODE FOR FATHER
L 63	10611		BX6	X1	
	5161000047		SA6	R1+P.PARAM+1	
L 64	5111000071		SA1	R1+P.PARAM+3	. FETCH INDEX FOR CAP FOR SON
	6160000000 X		SB6	=XSYSFRET	
L 65	6130000003		SB3	3	. ERROR MODIFIER
	6151000066		SB5	R1+P.PARAM	. A(CAP)
L 66	0400000000 X		EQ	=XPUTCAP	. XIT VIA PUTCAP
L 67	6140000001	FSON10	ERROR	1:NOFATH	. NO SUCH FATHER
L 70	6140000002	FSON11	ERROR	2:NEGPAT	. NO IMAGINARY SONS
L 71			ENDECS	FSON	

```
*
* DISPLAY SUBPROCESS DESCRIPTOR IN SPECIFIED BUFFER. AS MANY WORDS AS
* THE BUFFER WILL HOLD ARE DISPLAYED. BUFFER SIZE .LE.0 OR SUBPROCESS
* NON-EXISTANT CAUSE ERRORS.
*
* PARAMETERS
*      AP1      C:SUBPROCESS TO BE DISPLAYED
*      AP2      D:PTR TO BUFFER
*      AP3      D:BUFFER SIZE
*
*      ECSCODE  DSPSP
*      SA3      R1+P.PARAM+1      . LOCATE SUBP
*      SB7      DSPSP1
*      EQ       FINDSUB
*      DSPSP1   EQ      B6,B0,DSPSP12      . NO SUCH SUBP
*      SB5      B5-SD.CC      . PTR TO SUBP, ABS
*      SA1      R1+P.PARAM+2      . BUFFER ADR, REL RA
*      SA2      R1+P.PARAM+3      . SIZE OF BUFFER
*      SA3      R1+P.XPACK+2      . USER FL
*      LX3      60-36
*      SB3      X3      . FL
*      SB2      X1      . BUFFER
*      *B2=BUFFER, REL RA
*      SB3      B3-B2      . MAX POSSIBLE BUFFER SIZE
*      LE      R3,B0,DSPSP10      . BUFFER EXCEEDS FL
*      SB4      X2      . SIZE OF BUFFER
*      LE      B4,B0,DSPSP11      . NEG SIZE
*      GE      B3,B4,DSPSP2      . BUFFER SIZE OK
*      SB4      B3      . SUBSTITUTE AVAILABLE SPACE FOR SPECIFIED SPACE
*      *B4=NUMBER OF CELLS AVAILABLE
*      *B5=SUBP DESCRIPTOR, ABS
*
*      * WORD1  VFD      A/FLAGS:1A/INTERRUPT DATUM:1B/C-LIST LENGTH:1B/FL
*
*      DSPSP2   MX0      42
*      SA1      B5      . FETCH FIRST WORD
*      SA2      B5+SD.RAFL
*      LX2      40-18
*      BX3      -X0*X2      . SUBP RA
*      LX2      40-18
*      BX6      -X0*X2      . SUBP BA+FL
*      IX6      X6-X3      . SUBP FL
*      BX7      X0*X1      . ELIMINATE FUNNY PTR
*      BX6      X1-X6      . PUT FI IN LOW BITS
*      SA3      R1+P.XPACK+1
*      LX3      60-36      . USER RA TO LOW BITS
*      SA6      X3+B2      . STORE FIRST DISPLAYED WORD
*      SB4      B4-1
*      EQ      B4,B0,=XSYSRET      . QUIT MAYBE
*
*      * WORD2  VFD      A0/FATHERS CLASS CODE
*
*      SA1      B5+SD.PTRs
```

```
302
L 51 5131000067 6170000053
L 52 0400000025 +
L 53 0460000112 6055777776
L 54 5111000070 5127000071
L 55 5131000142 20340 63330
L 56 63210 67332 0404000110
L 57 63420 0604000111
L 60 0634000061 66430
L 61 43052 56150 5125000007
L 62 20252 15320 20252 15620
L 63 37663 11101 12616
L 64 5131000141 20330 53632
L 65 6144777776 0440000000 X
L 66 5115000004
```

		20152		LX1	40-18
		63311		SB3	R1+X1 . PTR TO FATHER, ABS
L	67	5113000001		SA1	R3+SD.CC . FETCH
		10611		Bx6	X1
L	70	5066000001		SA6	A6+1 . STORE WORD 2
		6144777776		SB4	R4-1
L	71	0440000000 X		EQ	R4,B0,SYSPRT . QUIT MAYBE
			* WORD 3	VFD	40/SURP CLASS CODE
		5115000001		SA1	R5+SD.CC
L	72	10611		Bx6	X1
		5066000001		SA6	A6+1
L	73	6144777776		SB4	R4-1
		0440000000 X		EQ	R4,B0,SYSPRT
			* WORD 4	VFD	40/0.18/ENTRY POINT,18/MAP ORIGIN,18/C-LIST ORIGIN
L	74	5115000005		SA1	R5+SD.Orig
		10611		Bx6	X1
L	75	5066000001		SA6	A6+1 . STORE WORD 4
		6144777776		SB4	R4-1
L	76	0440000000 X		EQ	R4,B0,SYSPRT
			* WORD 5	VFD	1/MAP OFF FLAG,5/0.18/COMP MAP SIZE, 18/NUMB OF LOG MAP ENTRIES,18/DEPTH IN SURP TREE
L	77	11101		SA1	R5+SD.MAP
		5125000004		Bx1	X0*X1 . ELIM LOG MAP PTR
		20230		SA2	R5+SD.PTRs
L	100	5131000162		LX2	60-36
		20352		SA3	R1+P.STACK
		15220		LX3	40-18 . SIZE OF STACK
L	101	15330		Bx2	-X0*X2
		37623		Bx3	-X0*X3
		21601		IX6	X2-X3 . HIGHEST ALLOWABLE STACK ENTRY
		12616		AX6	1 . GIVES DEPTH IN SUBP TREE
L	102	5066000001		Bx6	X1+X6 . COMBINE WITH MAP STUFF
		6144777776		SA6	A6+1 . STORE STH WORD
L	103	0440000000 X		SB4	R4-1
				EQ	R4,B0,SYSPRT
			* WORD 6	VFD	40/ERROR SELECTION MASK
L	104	10611		SA1	R5+SD.ESM
		5066000001		Bx6	X1
L	105	6144777776		SA6	A6+1 . STORE WORD 6
		0440000000 X		SB4	R4-1
				EQ	R4,B0,SYSPRT
			* WORD 7	VFD	40/SECOND WORD OF CAP FOR C-LIST
L	106	5115000003		SA1	R5+SD.CLSI

SUBPROCESS OPERATIONS
DISPLAY SUBPROCESS

COMPASS - VER 2.
FSUBP

11/03/71 15.38.03.

PAGE 20

L	107	5066000001	10671	BX6	x1
				SA6	A6+1
		0400000000 x		EQ	SYSRET

L	110	6140000002	DSPSP10	ERROR	2,BIGPAR
L	111	6140000003	DSPSP11	ERROR	3,NEGPAP
L	112	6140000001	DSPSP12	ERROR	1,NOFIND
L	113			END ECS	DSPSP

* THIS SYSTEM ACTION CAUSES ERROR PROCESSING FOR THE SPECIFIED ERROR
* TO BE INITIATED FOR THE SUBPROCESS NEXT TO THE CURRENT TOP OF STACK.

* PARAMETERS

* AP1 DIERROR CLASS
* AP2 DIERROR NUMBER
*

344			ECSCODE	ERCAL	
L 51	5111000066		SA1	B1+P.PARAM+0	
		10611	BX6	X1	
L 52	5161000071		SA6	B1+SUP.EC	
		5011000001	SA1	A1+1	
L 53	10611		BX6	X1	
		5161000072	SA6	B1+SUP.EN	
L 54	7160000005		SX6		
		5161000066	SA6	B1+SUP.KEY	KLUDGE THE PARAMETERS AROUND AND
L 55	0400000017		EQ	SUPRET	LET SUPRET DO THE WORK
L 56			ENDECS	ERCAL	

*
* SYSTEM ACTION TO SET ERROR SELECTION MASK OF SPECIFIED SUBP

* PARAMETERS

* AP1 D: POINTER TO WORD CONTAINING 60-BIT ESM
* AP2 C: CLASS CODE OF SUBPROCESS
*

351			ECSCODE	SESM	
L 51	5131000070		SA3	R1+P.PARAM+2	. CLASS CODE OF SUBP
		6170000053	SB7	ESMSETN1	. RETLINK
L 52	0400000025	+	EQ	FINDSUB	. LOCATE SUBPROCESS
L 53	0460000060		ESMSETN1	B6.B0.ESMSETN3	. NO SUCH SUBP
		5117000066	SA1	R1+P.PARAM	. CHECK LEGALITY OF POINTER
L 54	6160000001		SB6		
		6170000056	SB7	ESMSETN2	
L 55	0400000000	X	EQ	EXCHKPTR	
L 56	54100		ESMSETN2	A0	. FETCH NEW ESM
	10611		BX6	X1	
		5065000001	SA4	A5-SD.CC+SD.ESM	. STORE IN SUBP DESCRIPTOR
L 57	0400000000	X	EQ	EXSYSSET	
L 60	6140000002		ESMSETN3	ERROR	
L 61			ENDECS	SESM	

*
* SYSTEM ACTION TO SET ERROR SELECTION MASK OF CURRENT SUBPROCESS
*

* PARAMETERS

* API POINTER TO WORD CONTAINING 60-BIT ESM

361						
L	51	5111000066		ESMSET	SA1	SESMG
		6160000001			SB6	R1+P.PARAM . MAKE SURE POINTER IS OK
L	52	6170000053			SB7	T
		0400000000 X			EQ	ESMSETI . RETLINK
L	53	5111000162		ESMSETI	SA1	EXCHKPTR
		7211000001			SX1	R1+P.STACK . FETCH TOP OF STACK
L	54	53111			SA1	X1+1
		54300			SA3	R1+X1 . GET PTR TO CURRENT SUBP FROM TOS
		10643			BX6	A0 . FETCH NEW ESM
L	55	6261000002			SB6	X3
		56676			SA6	X1+SD.ESM
L	56	0400000000 X			EQ	R1+B6 . STORE NEW ESM
L	57			ENDECS	SESMG	

*
* SYSTEM ACTION TO RETURN TO THE AP2TH OCCURANCE OF SUBPROCESS AP1.
*

* PARAMETERS

* AP1 C:SUBPROCESS TO RETURN TO
* AP2 D:N=-1 FOR EARLIEST OCCURANCE ON STACK
* =0 FOR OCCURANCE NEAREST TOP OF STACK
* =1 FOR OCCURANCE NEXT TO TOP OF STACK, ETC
*

367
L 51 6170000052 0400000014 +
L 52 5111000162 43652 11161
L 53 76640 12616 54610
L 54 0400000000 X
L 55

ECSCODE JMRET
SB7 **1 . FIND DESIRED STACK ENTRY
EQ FINDNTHS
SA1 B1+P.STACK . FETCH STACK CONTROL WORD
MX6 42
BX1 X6*X1 . ELIMINATE OLD TOP OF STACK PTR
SX6 B4 . DESIRED STACK ENTRY
BX6 X1+X6
SA6 A1
EQ XTOSPROC . RETURN TO NEW TOP OF STACK
ENDECS JMRET

SUBPROCESS OPERATIONS
RETURN AND REDO OPER

COMPASS - VER 2.
FSUBP

11/03/71 15.38.04.

PAGE 25

373
L 51 7160000004
5161000066
L 52 0400000017 +
L 53

SPRET

ECSCODE
SX6
SA6
EQ
ENDECS

SPRET
4
R1+P.PARAM
SUPRET
SPRET

• KLUDGE KEY TO REDO AND
• EXIT VIA SUPER RETURN

SURPROCESS OPERATIONS
USER-INITIATED F-RETURN ACTION

COMPASS - VER 2.
FSUBP

11/03/71 15.38.04.

PAGE 26

375
L 51 7160000002 FRET
516T000066
L 52 0400000017 +
L 53

ECSCODE FRET
SX6 2
SA6 R1+P.PARAM
EQ SUPRET
ENDECS FRET

. KLUDGE KEY AND GO TO SUPER RETURN

SUBPROCESS OPERATIONS
USER-INITIATED RETURN ACTION

COMPASS - VER 2.

11/03/71 15.38.04.

PAGE 27

	377			ENTRY	RETURN	
I	51	7160000000	RET	ECSCODE	RET	
		5167000066		SX6	0	. KLUDGE KEY AND GO TO SUPER RETURN
I	52	0400000017 +		SA6	B1+P.PARAM	
L	53			EQ	SUPRET	
			51 RETURN	ENDECS	RET	
				EQU	RET	

* THIS IS THE ECS ACTION FOR SUBPROCESS RETURN WITH PARAMETERS.
* IT KLUDGES THE PARAMETER AREA AND JUMPS INTO THE SUPER RETURN
* ACTION.

* PARAMETERS

* AP1

* AP2

*

D: DATA CONTROL WORD - 30/WORD COUNT, 30/PIR TO FIRST WORD
D: CAP CONTROL WORD - 30/CAP COUNT, 30/IDX OF FIRST CAP

401
L 51 5111000066 5027000001
L 52 10611 10722 54620
L 53 5076000001 7160000001
L 54 54610 0400000017 +
L 55

PRET

ECSCODE

SA1

SA2

BX6

BX7

SA6

SA7

SX6

SA6

EQ

ENDECS

PRET

A1+P.PARAM

A1+1

X1

X2

A2

A6+1

1

A1

SUPRET

PRET

. MOVE CONTROL WORDS UP ONE
. AND STICK IN KEY FOR RETURN
. PARAMETERS ONLY

. EXIT VIA SUPER RETURN

* THIS ACTION WILL DO ANY COMBINATION OF THINGS WHICH OTHER RETURN
* ACTIONS DO, DEPENDING ON THE VALUE OF THE KEY.

* THERE ARE 6 DATA PARAMETERS, AS FOLLOWS:

66		ORG	D.PARAM	
66	SUP.KEY	BSS	1	. THIS DETERMINES THE ACTION
67	SUP.BD	BSS	1	. BLOCK DATA POINTER, IF APPLICABLE
70	SUP.BC	BSS	1	. BLOCK CAPABILITY PTR, IF APPLICABLE
71	SUP.EC	BSS	1	. ERROR CLASS, IF APPLICABLE
72	SUP.EN	BSS	1	. ERROR NUMBER, IF APPLICABLE
73	SUP.INT	BSS	1	. INTERRUPT DATUM, IF APPLICABLE
		USE	*	

* THE KEY DETERMINES HOW THE ACTION WILL PERFORM. STARTING WITH BIT 59.
* THE LOW ORDER BIT, IF THE BIT IS ONE WE HAVE -

59	RETURN WITH PARAMETERS
58	F-RETURN
57	REDO X1
56	COMPLETE OPERATION
55	RETURN WITH ERROR
54	RETURN WITH INTERRUPT

* THESE BITS ARE ESSENTIALLY PROCESSED IN THE ORDER IN WHICH THEY ARE LISTED.
* SO THAT IF THE COMPLETE OPERATION BIT IS SET, FOR EXAMPLE, IT WILL
* OBVIATE THE EFFECT OF THE F-RETURN OF REDO BITS.

17				
405		SUPRET	BSS	
650	5111000086		ECSSUB	
	15671		SA1	SRCTN, BUFB
651	5161000055		SA6	R1+SUP.KEY
	20173		LX1	. FETCH KEY
652	0321000656		PL	R1+P.SCR2+5
	6170000655		SB7	. SAVE FOR LATER
653	5111000067		SA1	60-1
	5121000070		SA2	X1, SUPRET2
654	0400000022		EQ	. SKIP IF NO PARAMETERS
655	5111000055	SUPRET1	SA1	SUPRET1
	20173		SA2	. RETLINK
656	5121000162		SA2	R1+SUP.BD
	7160000002		SA2	R1+SUP.BC
657	37626		IX6	. BLOCK PARAMETER INFO
	63760		SB7	GETRETP
	20242		LX2	. STASH BLOCK PARAMETERS IN PARMBUF
	63620		SB6	R1+P.SCR2+5
660	0776000722		LT	. RESTORE KEY
	54620		SA6	60-1
	53261		SA2	R1+P.STACK
661	20173		LX1	. DECREMENT THE STACK
	0321000643		PL	2
	20230		LX2	X2-X6
				X6
				60-18
				X2
				. STACK ORIGIN
				B7, B6, SUPRET90
				. STACK WOULD BE EMPTY
				A2
				. SET NEW STACK POINTERS
				R1+X6
				. FETCH NEW TOP OF STACK
				60-1
				X1, SUPRET3
				. SKIP IF NO F-RETURN
				60-36
				. INCREMENT F-RETURN COUNT

SUBPROCESS OPERATIONS
SUPER RETURN OPERATION

COMPASS - VER 2.
FSUBP

11/03/71 15.38.05.

PAGE 30

L	662	7160000001		SX6	7	
		36242		IX2	X6+X2	
		20244		LX2	36	
L	663	20173	SUPRET3	LX1	60-1	
		0321000645		PL	X1, SUPRET4	. SKIP IF NO REPEAT BIT
		43601		MX6	7	
L	664	20670		LX6	60-SF.PCQ1	
		12226		BX2	X2+X6	. SET ABOUT TO EXECUTE BIT
		20663		LX6	60-SF.PCQ2-SF.PCQ1	
		15226		BX2	-X6*X2	. CLEAR ALMOST FINISHED BIT
L	665	20173	SUPRET4	LX1	60-1	
		0321000647		PL	X1, SUPRET5	. SKIP IF NO COMPLETE BIT
		43601		MX6	7	
L	666	20670		LX6	60-SF.PCQ1	
		15226		BX2	-X6*X2	. CLEAR ABOUT TO EXECUTE BIT
		20643		LX6	60-SF.PCQ2-SF.PCQ1	
		12262		BX2	X6+X2	. SET ALMOST FINISHED BIT
L	667	10622	SUPRET5	BX6	X2	
		54620		SA6	A2	. SET NEW TOP OF STACK
		6160000671		SB6	SUPRET6	. ERROR RETURN
L	670	6170000672		SB7	SUPRET7	. RETLINK
		0400000036		EQ	ENVIORN	
L	671	6170000675	SUPRET6	SB7	SUPRET8	. SKIP RETURNING PARAMETERS IF ENVIORN ERROR
		0400000000		EQ	=XERRPROC	. GENERATE NEW TOS FOR ERROR
L	672	5111000055	SUPRET7	SA1	R1+P.SCR2+5	. RESTORE KEY
		20173		LX1	60-1	
L	673	0321000675		PL	X1, SUPRET8	. SKIP IF NO PARAMETERS
		6160000675		SB6	SUPRET8	. ERROR RETURN
L	674	6170000675		SB7	SUPRET8	. RETLINK
		0400000000		EQ	=XOUTRTP	. MOVE THE PARAMETERS TO TOS CORE
L	675	5111000055	SUPRET8	SA1	R1+P.SCR2+5	. RESTORE KEY
		20147		LX1	60-5	
L	676	0321000702		PL	X1, SUPRET9	. SKIP IF NO ERROR STUFF
		5121000071		SA2	R1+SUP.EC	. GIVE ERROR CLASS AND NUMBER TO
L	677	10622		BX6	X2	. THE ERROR PROCESSOR
		5121000072		SA2	R1+SUP.EN	
		10722		BX7	X2	
L	700	6170000701		SB7	SUPRET10	. RETLINK
		0400000000		EQ	=XERRPROC	
L	701	5111000055	SUPRET10	SA1	R1+P.SCR2+5	. RESTORE KEY
		20147		LX1	60-5	
L	702	20173	SUPRET9	LX1	60-1	
		0321000000		PL	X1, =XTOSPROC	. EXIT VIA TOP OF STACK PROCESSOR
						. IF NO INTERRUPT STUFF
L	703	5111000162		SA1	R1+P.STACK	
		7211000001		SX1	R1+1	
L	704	53111		SA1	R1+X1	. GET CURRENT SUBP FORM TOP OF STACK
		63210		SB7	X1	. PTR TO CURRENT SUBP, REL R1
L	705	56112	SUPRET11	SA1	R1+B2	. SD.FINT WORD OF SUBP
		20100		LX1	SD.FINT	
		0331000712		NG	X1, SUPRET13	. DONE - INTERRUPT PENDING HERE
L	706	20102		LX1	SD.FDIS=SD.FINT	
		0321000713		PL	X1, SUPRET12	. FOUND A SUBP TO TAKE THE INT

```

L 707 5011000004 SA1 A1+SD.PTRs-SD.FINT
      20152 LX1 60-18
      63210 SB2 X1 . FATHER PTR
L 710 0520000705 NE B2.B0.SUPRET11 . CONTINUE IF NOT AT ROOT
L 711 0000000000 PATCH2 PS 0 . BETTER CHANGE THE NEXT INST TOO
L 712 0000000000 x SUPRET13 PS XTOSPROC . DONE - INTERRUPT ALREADY PENDING OR
      * . NOBODY WANTS ONE
      *
      * FOUND A SUBP TO TAKE INTERRUPT
      *
L 713 43601 SUPRET12 MX6 T
      20172 LX1 60-SD.FDIS
      20674 LX6 60-SD.FINT
      12116 BX1 X1+X6 . SET PENDING INTERRUPT BIT
L 714 43622 MX6 T8
      20666 LX6 60-6
      15116 BX1 -X6*X1 . ERASE POSSIBLE OLD INT DATUM
L 715 5121000073 SA2 R1+SUP.INT
      20244 LX2 36
      11262 BX2 X6+X2 . DISPOSE OF XTRA BITS
L 716 12612 BX6 X1+X2 . STICK IN NEW INTERRUPT DATUM
      54610 SA6 A1 . STORE NEW SD.FINT WORD
      5111000165 SA1 R1+P.INTERR . INCREMENT NUMBER OF INTERRUPT
L 717 7160000001 SX6 T . SUBPS
      366A1 IX6 X6+X1
      54610 SAA A1
L 720 0000000000 PATCH3 PS
L 721 0400000000 x EQ XTOSPROC . EXIT VIA TOP OF STACK PROCESSOR
      *
L 722 6140000000 SUPRET40 ERROR 3-ESTK
L 723 ENDSUB SRETN.BUFR

```

COPY RETURN PARAMETERS TO ECS BUFFER

X1= DATA CONTROL WORD
X2= CAPABILITY CONTROL WORD
R7= RETURN LINK

ALL REGISTERS USED
ON EXIT: *P.PARAMC* = BUFFER CONTROL DATA
(24/0,18/CAP COUNT,18/WORD COUNT)

22		EXT	DARMBUF	
460		BSS	0	
		ECSSUB	GETP, RUF A	
450	5131000142	SA3	R1+P.XPACK+2	CHECK OUT DATA CONTROL WORD
	21344	AX3	36	SUBPROCESS FL
	73410	SX4	X1	STARTING ADDR FOR DATA WORDS
451	0334000517	NG	X4,ERR1	ERROR: NEGATIVE POINTER
	20146	LX1	60-30	
	63510	SB5	X1	DATA WORD COUNT
452	0750000520	LT	R5,B0,ERR2	ERROR: NEGATIVE WORD COUNT
	73645	SX6	X4+B5	
	37636	IX6	X3-X6	
453	0336000526	NG	X6,ERR10	ERROR: DATA BLOCK NOT WITHIN FL
	63220	SB2	X2	CHECK OUT CAPABILITY CONTROL WORD
	20236	LX2	60-30	STARTING CAPABILITY INDEX
454	0720000523	LT	R2,B0,ERR5	ERROR: NEGATIVE C-LIST INDEX
	63320	SB3	X2	COUNT OF CAPABILITIES
	66433	SB4	B3+B3	
455	0730000521	LT	R3,B0,ERR3	ERROR: NEGATIVE CAP COUNT
	6144777405	SB4	R4-P.PBUFL	
456	5121000141	SA2	R1+P.XPACK+1	
	66445	SB4	R4+B5	
	21244	AX2	36	SUBPROCESS RA
457	0704000525	GT	R4,B0,ERR7	ERROR: TOO MUCH STUFF FOR BUFFER
	36224	IX2	X2+X4	
	53020	SA0	X2	
460	7100000000 X	SX0	DARMBUF	
461	0125000000	WE	R5	MOVE DATA WDS TO ECS
	0100000000 X	RJ	R.ECS	
462	76650	SX6	R5	
	36006	IX0	X0+X6	INCR BUFFER POINTER
	76730	SX7	R3	
	20722	LX7	18	
463	12776	BX7	X7+X6	
	5171000125	SA7	R1+P.PARAMC	SAVE BUFFER CONTROL DATA
464	0430000516	EQ	R3,B0,GETRETP9	IF NO CAPABILITIES TO PASS

SUBPROCESS OPERATIONS
COPY RETURN PARAMETERS TO ECS BUFFER

COMPASS - VER 2. 11/03/71 15.38.07.
FSUBP

PAGE 34

STEP TO NEXT TRIP

L	506	0430000516 0524000470	GETRETP5	EQ NE	B3,B0,GETRETP9 B2,B4,GETRETP2	JP IF DONE (COUNT= 0) JP IF NOT TIME FOR NEXT C-LIST
L	507	5044000002 62440 66200		SA4 SB4 SB2	A4+2 X4 B0	LENGTH OF NEXT C-LIST
L	510	0304000522 0200000470		ZR JP	X4,ERR4 GETRETP2	ERROR: INDEX TOO LARGE

X-FER CAPABILITIES FROM LOCAL C-LIST

L	511	67542 0635000512 66530	GETRETP7	SB5 LE	B4-B2 B5,B3,GETRETP8	FIND MIN(LENGTH-INDEX,COUNT)
L	512	67335 7162000147 73662	GETRETP8	SB5 SB3 SX6 SX6	B3 B3-B5 B2+P,LOCALC X6+B2	DECR COUNT
L	513	53061 66635 66240		SA0 SB6 SB2	B1+X6 B5+B5 B4	ABS CM STARTING ADDR
L	514	0126000000 0100000000 X		WE RJ	B6 B, ECS	INCR INDEX WRITE CAPABILITIES
L	515	76660 36006 0200000506		SX6 IX0 JP	B6 X0+X6 GETRETP5	INCR BUFFER ADDRESS

L	516	0270000000	GETRETP9	JP	B7	DONE RETURN
---	-----	------------	----------	----	----	----------------

L	517	6140000001	ERR1	ERROR	I, NEGRT
L	520	6140000002	ERR2	ERROR	V, NEGRT
L	521	6140000001	ERR3	ERROR	I, NEGRT
L	522	6140000002	ERR4	ERROR	N, BIGIX
L	523	6140000002	ERR5	ERROR	N, NEGIX
L	524	6140000002	ERR6	ERROR	V, CLNOT
L	525	6140000000	ERR7	ERROR	O, ROOM
L	526	6140000000	ERR10	ERROR	O, BDTBLK
L	527			ENDSUB	GETP, BUFA

```

*
* THIS ROUTINE ALWAYS FINDS THE REQUIRED CLASS CODE, BECAUSE
* IT STORES IT AT THE END OF THE SUBPROCESS TABLE BEFORE IT STARTS.
* (SO IF THE ADDRESS DELIVERED IN A5 IS AT THE END OF THE TABLE (=A6),
* NO SUBPROCESS WITH THAT CLASS CODE EXISTS.
*
* AT ENTRY -
*   X3 = NAME OF SUBPROCESS (CLASS CODE)
*   B7 = RETLINK
*
* AT EXIT -
*   A6 = A(SUBP WITH GIVEN CLASS CODE)+SD*CC
*   A6 = A(END OF SUBPROCESS TABLE)+SD*CC
*   B6 = 0 IF NO SUCH SUBP
*
* REG USED -
*   A4/X4, A5/X5, A6/X6, B3, B6
*
25 5151000163      ENTRY FINDSUB
                        SA5  R1+P.SUBPDT . GET PTR TO SUBP DESCRIPTORS
26 20530          SB3  SD.CC . CONSTANT
                        LX5  6+18 . PTR BEYOND LAST EXTANT SUBP
                        SB6  X5+B3 . PLUS DISTANCE TO CLASS CODE
                        63653
*
* STORE DESIRED CLASS CODE AT END OF TABLE TO END THE LOOP
*
27 20522          BX6  X3
                        SA6  B1+B6 . STORE CLASS CODE
                        LX5  18 . ADDR OF FIRST SUBP, REL B7
                        SA4  B1+X5 . PICK UP FIRST WORD OF FIRST SP
30 54543          FINDSUB1 SA5  A4+B3 . GET CLASS CODE
                        BX5  X5-X3
                        SB6  X4
                        SA4  A4+B6 . MOVE TO NEXT SUBP
31 0315000030 +   NZ  X5-FINDSUB1 . CONTINUE IF NOT FOUND
                        SB6  A5 . GET EXIT SIGNAL
                        SB6  A6-B6
                        JP  A7 . EXIT
                        64650
                        65666
32 0270000000

```

```
*
* THIS ROUTINE REDOES ALL POINTERS, ETC., IN THE PROCESS DESCRIPTOR
* WHENEVER THE SIZE OF SOMETHING CHANGES. IT
*   1) CONSTRUCTS THE NEW DESCRIPTOR IN ECS
*   2) RECALCULATES USERB1
*   3) SWAPS EVERYTHING BACK IN, ALL READY TO RUN
*
* AT ENTRY
*   B1 IS THE OLD USERB1
*   THE OLD PROCESS DESCRIPTOR IS IN CORE AS USUAL AT B1
*   THE PROCESS DESCRIPTOR HAS BEEN REALLOCATED TO THE NEW SIZE
*   IN ECS
*   IF THERE IS A NEW SUBP, ITS DESCRIPTOR HAS BEEN CONSTRUCTED
*   (WITH OLD POINTERS) AT THE END OF THE SUBPROCESS
*   DESCRIPTOR TABLE. MAP STUFF NOT INCLUDED WITH NEW
*   DESCRIPTOR.
*   ALL SUBPS ARE SWAPPED OUT.
*   B7 = RETLINK
*   B1+P.SCR2+0 CONTAINS V1=THE CHANGE IN THE LENGTH OF THE LOCAL C-LIST BUF
*   (CELLS, NOT CAPS)
*   B1+P.SCR2+1 CONTAINS V2=THE CHANGE IN THE LENGTH OF THE FULL C-TABLE
*   B1+P.SCR2+2 CONTAINS V3=THE CHANGE IN THE LENGTH OF THE CALL STACK
*   B1+P.SCR2+3 CONTAINS V4=THE CHANGE IN THE LENGTH OF THE SUBPROCESS TABLE
*   B1+P.SCR2+4 CONTAINS V5=THE CHANGE IN THE MAX FULL-DEPTH FL
*   B1+P.SCR2+5 CONTAINS ADRA=THE ADDR (REL B1) AT WHICH A SUBPROCESS WAS
*   ADDED OR DELETED, OR 0 IF THERE WAS NO CHANGE LIKE THAT
```

33
537

```
OMYGOD      BSS      0
ECSSUB      0MYGO.BUFA
```

```
* GET ECS A(PROCESS DESC)
```

```
L 450 5111000126 20142 73010
L 451 56010 43747 5121000127
L 452 0110000001
L 453 54100 15017 20242 73220
L 454 36002
```

```
SA1      B1+P.ROHEAD
LX1      6+36
SX0      X1      . PROCESS NOT
SA0      B1      . SCRATCH AREA
MX7      39
SA2      B1+P.ROHEAD+1
RECS     1
SA1      A0
BX0      -X7*X1      . GET ECS A(PROC)
LX2      6+36      . FIND A(VAR LENGTH DESC)
SX2      X2
IX0      X0+X2
```

```
*X0=ECS A(VAR DESC)
```

```
* FIX P.CTABLE
```

```
VFD      6/0.18/MAX LEN LOCAL C-LIST,18/ENTRIES
          IN FULL C-TABLE,18/ORIG FULL C-LIST
```

```
L 455 5121000050 5111000141 5131000051
L 456 36432
```

```
SA1      B1+P.CTABLE
SA2      B1+V1
SA3      B1+V2
IX4      X3+X2
```


Address	Instruction	Comment
	*X4=V1+V2	
457	36662	ADJUST ORIG FULL C-TABLE BY V1+V2
460	20622	ADJUST LEN(C-LIST BUF) BY V1/2
	54610	ADJUST FULL PATH BY V2/2
		STORE NEW P-C-TABLE
	* FIX UP P-STACK	VFD 6/0:18/STACK LIMIT,18/STACK ORIGIN,18/TOP OF STACK
461	5121000052	SA1+P-STACK
462	20630	SA1+V3
	36614	SA1+V3
463	20622	SA1+V3
464	54610	SA1+V3
	36642	SA1+V3
		SAVE OLD A(TOP OF STACK), ABSOLUTE
		ADJUST TOP OF STACK BY V1+V2
		RESTORE FULL PATH INCREASE
		ADJUST STACK LIMIT BY V1+V2+V3
		SAVE OLD A(STACK ORIGIN), ABSOLUTE
		ADJUST STACK ORIGIN BY V1+V2
		STORE NEW P-STACK
	*X4=V1+V2+V3	
	* NOW MESSAGE THE STACK AND WRITE OUT THE NEW VERSION	
	WORD 1	VFD 24/0:18/PTR TO END OF PATH,18/PTR TO CURRENT
465	63420	SA1+ADR4
	5121000053	SA1+ADR4
466	67265	SA1+ADR4
467	5115000001	SA1+ADR4
	63310	SA1+ADR4
470	0643000471	SA1+ADR4
471	20652	SA1+ADR4
	63360	SA1+ADR4
472	0643000473	SA1+ADR4
473	20622	SA1+ADR4
	54610	SA1+ADR4
	6155000002	SA1+ADR4
		CM A(OLD STACK) FOR LATER WRITE
		ALMOST HOW MANY WORDS TO WRITE
		GET WORD 1 OF STACK
		ADJUST CURRENT BY V1+V2+V3
		AND BY V4 IF ITS ABOVE THE CRITICAL LIMIT
		ADJUST END OF PATH BY V1+V2+V3
		AND BY V4 IF ETC
		ADVANCE STACK ADDRESS AND CONTINUE TO

SUBPROCESS OPERATIONS
PROCESS METAMORPHOSIS

COMPASS - VER 2. 11/03/71 15.38.08.
FSUBP

PAGE 38

L 474 0665000467
L 475 0122000002

LE
WECS

R5,B6,OMY3 TOP OF STACK
R2+2 . WRITE OUT ACTIVE STACK ENTRIES

*
* FIX UP P.SUBPDT VFD 6/0*18/NEXT SUBP.18/SUBP TABLE ORIG.18/ESUBPS
*

L 476 5111000163
L 477 7130000001

SA1
ZR

R1+P.SUBPDT
X2,OMY4 . IF V4= 0, NO CHANGE IN ESUBPS

L 477 7130000001
L 500 0322000502

SX3
IX1
SB2

1
X1+X3 . INCREMENT ESUBPS
X1 . NUMBER OF SUBPS TO MESSAGE

L 500 0322000502
L 501 37113
L 502 20130

PL
IX1
SB2

X2,OMY4
X1-X3 . OOPS, DECREMENT ESUBPS
X1 . DELETED ONE MESSAGES TOO.

L 501 37113
L 502 20130
L 503 36664

IX1
LX1
IX6

X1-X3
6+18
X1+X4 . ADJUST NEXT SUBP BY V1+V2+V3+V4

L 503 36664
L 504 63611

IX6
LX6
IX6
LX6
SA6

X6+X2
18
X6+X4 . ADJUST ORIG OF TABLE BY V1+V2+V3
18
A1

*
* NOW MESSAGE POINTERS IN SUBP TABLE
*

SD.PTRS
SD.RAFL

VFD 6/0*18/MAX STACK.18/FATHER.18/MAPIN LIST
VFD 6/0*18/RA+FL.18/RA.18/SPACE LEFT IN COMP MAP

L 504 63611
L 505 5116000004

LX1
SB6
SA0

18
X1+B1 . SAVE OLD CM A(SUBP TABLE), ABSOLUTE
R6 . SET UP FOR LATER WRITE

L 505 5116000004
L 506 20622

IX7
SA1
LX1

X2+X4 . GIVES V1+V2+V3+V4
R6+SD.PTRS . HANDLE ROOT SPECIALLY TO AVOID
6+18 . MESSING UP O FATHER POINTER

L 506 20622
L 507 5116000004

IX6
LX6
EQ

X1+X4 . ADJUST MAX STACK BY V1+V2+V3
18
OMY5 . JUMP INTO LOOP

L 507 5116000004
L 510 20622

SA1
LX1
IX6

R6+SD.PTRS
6+18
X1+X4 . ADJUST MAX STACK BY V1+V2+V3

L 510 20622
L 511 0645000512

LX6
SB5
IX6

18
X6
X6+X4 . ADJUST FATHER BY V1+V2+V3

L 511 0645000512
L 512 20622

LE
IX6
LX6

R5,B4,OMY5
X6+X2 . AND BY V4 SOMETIMES
18
A1 . STORE NEW SD.PTRS WORD

L 512 20622
L 513 20130

SA6
SA1
LX1

R6+SD.RAFL
6+18
X1+X7 . ADJUST RA+FL BY V1+V2+V3+V4

L 513 20130
L 514 20622

IX6
LX6
IX6
LX6

18
X6+X7 . ADJUST RA BY V1+V2+V3+V4
18

```

      54610
L 515 5066000001 13646
      541A0
      63616
L 516 6122777776
      0702000507

      SA6
      BX6
      SA6
      SA1
      SB6
      SB2
      GT
      A1
      X6=X6
      A6+1
      B6
      X1+B6
      B2-1
      B2,B0,OMY6

      . STORE NEW SD.RAFL WORD
      . CLEAR MAP COUNTS TO FORCE RECOMPILATION
      . GET NEXT SUBP POINTER
      . DONT DO TOO MANY

*
* WRITE OUT THE SUBPROCESS TABLE
*
L 517 5111000163
L 520 20152
      20252
      73220
      73310
L 521 37232
      36002
      201A2
      63610
L 522 63230
      67662
      5121000053
L 523 0322000531
      652A1
      67342
L 524 0123000000
L 525 54003
      63220
      550A2
      76230
L 526 36002
      67363
      6133000002
L 527 0123000000
L 530 0400000536
L 531 6222777767
      67342
      340A1
      43674
L 532 0123000000
L 533 7116777776
      340A1
      43674
L 534 54600
L 535 0120000001

      SA1
      SA2
      LX1
      LX2
      SX2
      SX3
      IX2
      IX0
      LX1
      SB6
      SB2
      SB6
      SA2
      PL
      SB2
      SB3
      WECS
      SA1
      SB2
      SA0
      SX2
      IX0
      SB3
      SB3
      WECS
      EQ
      SB2
      SB3
      WECS
      SX1
      IX0
      MX6
      SA6
      WECS
      B1+P.SUBPDT
      B1+P.STACK
      A0-18
      A0-18
      X2
      X1
      X3-X2
      X0+X2
      A0-18
      X1
      X3
      B6-B2
      B1+V4
      X2,OMY7
      A0-B1
      B4-B2
      B3
      A0+B3
      X2
      A0-B2
      B3
      X0+X2
      B6-B3
      B3+SD.CC+1
      B3
      OMY8
      X2-SD.SIZE
      B6-B2
      B3
      B6-1
      X0+X1
      B0
      A0
      1

      . CALCULATE ECS A(SUBPROCESS TABLE)=
      . ECS A(VAR DESC)+ORIG SUBP TABLE-ORIG STACK
      . CM A(BEG NEW SUBP TABLE)
      . END OF NEW SUBP TABLE
      . SIZE OF NEW SUBP TABLE, LESS SD.CC WORDS
      . IF V4 .GE. 0, WRITE WHOLE THING AT ONCE
      . WRITE WORDS UP TO CRITICAL ADDRESS
      . ADVANCE CM A BY WORDS WRITTEN +V4
      . ADVANCE ECS A BY EWORDS WRITTEN SO FAR
      . TAKE CARE OF WORDS ALREADY WRITTEN
      . WRITE THRU PHONY CLASS CODE
      . NOT ALL OF THE NEW SUBP EXISTS
      . OLD SUBPS + SD.SIZE WORDS OF NEW ONE
      . HORROR KUDGE - WE MUST ADD THE -0
      . AT THE END OF THE NEW SUBP LOG MAP

*
* FIX P.ROHEAD, P.ROHEAD+1
*
L 536 7160000001
      5160000000 x
L 537 56110
      43747
      15017
L 540 5101000126

      OMY8
      SX6
      SA6
      SA1
      MX7
      BX0
      SA0
      1
      XI.LOCK
      B1
      29
      -X7*X1
      B1+P.ROHEAD
```

SUBPROCESS OPERATIONS
PROCESS METAMORPHOSIS

COMPASS - VER 2.
FSUBP

11/03/71 15.38.09.

PAGE 40

```

L 541 0110000003
L 542 54100
      5121000050
L 543 5131000051
      5141000052
L 544 36223
      36224
      5131000053
L 545 36223
      36443
      5131000054
L 546 36223
      36612
      63240
      54610
L 547 5010000001
      36614
      54610
L 550 0120000041

L 551 5110000000 X
      63110
      67112
L 552 76610
      5160000000 X
L 553 5101000126
      13646
L 554 5160000000 X
L 555 0110000041
L 556 5160000000 X
      5121000127
L 557 63220
      20252
      73220
      36002
L 560 5111000162
      20152
      53011
L 561 0112000000
L 562 5060777776
      7160777773
L 563 5161000164
      5111000146
L 564 43630
      15116
      7161000140
L 565 20644
      12616
      54610
L 566 6160000000 X
      0400000036 +

```

RECS
SA1
SA2
SA3
SA4
IX2
IX2
SA3
IX2
IX4
SA3
IX2
IX6
SB2
SA6
SA7
IX4
SA6
WECS

```

3
A0
R1+V1
R1+V2
R1+V3
X2+X3
X2+X4
R1+V4
X2+X3
X4+X3
R1+V5
X2+X3
X1+X2
X6
A1
A0+1
X1+X4
A1
P.PROCRO+P.PROCRW

```

• ACCUMULATE V1+V2+V3+V4+V5= TOTAL CHANGE
• IN CM SIZE OF PROCESS IN X2

• ACCUMULATE V3+V4=TOTAL CHANGE IN ECS SIZE
• OF PROCESS IN X4

• ADJUST PROCESS LENGTH BY V1+V2+V3+V4+V5
• SAVE NEW PROCESS LENGTH

ADJUST VAR DES LENGTH BY V3+V4

WRITE OUT FIXED-LENGTH DESC

* RECALCULATE USERB1. SWAP USER BACK IN. SET MA IN XPACK

SA1
SB1
SB1
SX6
SA6
SA0
BX6
SA6
RECS
SA6
SA2
SB2
LX2
SX2
IX0
SA1
LX1
SA0
RECS
SA6
SX6
SA6
SA1
MX6
BX1
SX6
LX6
BX6
SA6
SB6
EQ

```

=XS.CMFL
X1
R1-B2
R1
=XS.USERB1
R1+P.ROHEAD
X6-X6
=XS.USERA
P.PROCRO+P.PROCRW
I-LOCK
R1+P.ROHEAD+1
X2
60-18
X2
X0+X2
R1+P.STACK
60-18
R1+X1
B2
A0-1
=SD.PTRS
R1+P.MAPSIN
R1+P.XPACK+6
P4
=X6*X1
R1+P.XPACK
36
X1+X6
A1
=XE.ERROR
ENVIORN

```

• END OF SPACE

• B1 = END OF SPACE-SPACE REQUIRED

• CLEAR RA TO PREVENT CHIPCAL ERRORS

• READ FIXED-LENGTH DESC

• UNLOCK INTERRUPTS

• LENGTH OF VAR DESC

• ORG OF VAR DESC REL P.ROHEAD

• CM ORIG(VAR DESC)

• READ VAR DESC

• SET 0 AT END OF FULL C-TABLE

• CLEAR MAP CHAIN FOR ENVIORN

• SET NEW MA IN XPACK

• RETURN IN CASE OF ERROR

• NORMAL RETURN IS TO B7

SUBPROCESS OPERATIONS
PROCESS METAMORPHOSIS

COMPASS - VER 2.
FSUBP

11/03/71 15.38.10.

PAGE 41

L 567

ENDSUB OMVGO.BUFA

```

*
* ENVIORN ESTABLISHES THE PROPER MEMORY SPACE AND FULL C-LIST FOR
* THE TOP OF STACK. IT ALSO SETS RA AND FL IN THE EXCHANGE PACKAGE
* AREA OF THE PROCESS AND THE RA IN S.USFRA.
*
* AT ENTRY
*     B1 = PROCESS SCRATCH AREA
*     B6 = ERROR RETURN LINK
*     B7 = RETLINK
*
* AT EXIT TO B6:
*     X6,X7 ARE SET UP FOR A CALL OF E.ERROR
*
* CALLS MAPIN, MAPOUT
*
* USES P.TEMP1 THRU P.TEMP7, P.SCR2, P.SCR2+1
*
* REGISTER ALLOCATION
*     B1 = PROCESS SCRATCH AREA
*     B2 = A(SD.PTRS WORD) OF CURRENT SUBP IN TOP OF STACK
*     B3 = C-LIST TABLE POINTER, ABSOLUTE
*     B4 = A(SD.PTRS WORD) IN PREVIOUS SUBP IN MAPIN CHAIN
*     = PMOSP, THE PREVIOUS MAP CHAIN SUBPROCESS
*     B5 = A(SD.PTRS WORD) OF CURRENT SUBP IN FULL PATH
*     = FPSP, THE FULL PATH SUBPROCESS
*     B6 = A(SD.PTRS WORD) OF CURRENT SUBP IN MAP CHAIN
*     = MCSP, THE MAP CHAIN SUBPROCESS
*     B7 = B1+SD.RAFL-SD.PTRS, A NICE CONSTANT
*
*     A1,X1 = MXT 42
*     A2,X2 = CURRENT SUBP IN FULL PATH, DERIVES FROM B5
*     A3,X3 = CURRENT SUBP IN MAP CHAIN, DERIVES FROM B6
*
*     B2,B4,B5,B7 ARE RELATIVE TO B1
*
* SET UP REGISTERS

```

```

36 76670
    20622
    76740
    12667
37 5161000051
    13646
40 5161000050
    5151000162
41 63651
    5156000001
42 6225000004
    21522
43 6255000004
    5151000161

```

```

ENTRY
ENVIORN SX6
        LX6
        SX7
        BX6
        SA6
        BX6
        SA6
        SA5
        SB6
        SA5
        SB2
        AX5
        SB5
        SA5
        ENVIORN
        B7
        T8
        B6
        X6+X7
        B1+P.SCR2+1
        X6-X6
        B1+P.SCR2
        B1+P.STACK
        B1+X5
        B6+1
        X5+SD.PTRS
        T8
        X5+SD.PTRS
        B1+P.CTABLE
        . SAVE RETLINKS
        . INITIALIZE ERROR FLAG TO
        . NO ERROR
        . GET TOP OF STACK
        . SECOND WORD OF TOS
        . CURRENT SUBP
        . FPSP = END OF PATH
        . GET C-TABLE POINTER

```

44 63351
6171000003 43152
45 6140000164
56275 20230
46 73620
5161000044 20244

SB3 X5+B1
SB7 R1+SD.RAFL-SD.PTRS . NICE CONSTANT
MX1 A+36 . MASK
SB4 P.MAPSIN . PMCSP
SA2 B7+B5 . RA,FL OF EOP
LX2 A+18
SX6 X2
SA4 R1+P.TEMP7 . SAVE RA+FL TO CALC FL LATER
LX5 36 . RESTORE X2

*
* FIGURE OUT HOW TO MERGE EOP INTO MAP CHAIN
*
* TEST IF EOP ALREADY IN (THAT WOULD BE EASY)
*

47 56415
63640 0560000104 *

SA4 R1+B5 . GET MAP FLAG OF EOP SUBP
SB6 X4
NE R6,B0,E50 . EOP SUBP IS ALREADY IN, JUMP

*
* TEST IF MAPIN LIST IS NULL
*

50 56514
6265000004
51 0460000112 *

E10 SA5 R1+B4
SB6 R5+SD.PTRS . INITIALIZE MCSP
EQ R6,B0,E60 . JUMP IF END OF MAP CHAIN

*
* SEE IF EOP IS ABOVE MCSP
*

20222 15421
52 56376
20330 15531 37645
53 0326000074 *

LX2 A0-18
BX4 -X1*X2 . RA OF EOP
SA3 B7+B6 . GET RA,FL WORD OF MCSP
LX3 A0-36
BX5 -X1*X3 . RA+FL OF MCSP
IX6 X4-X5
PL X6,E40 . EOP IS ABOVE MCSP, JUMP

*
* SEE IF EOP IS BELOW MCSP
*

20222 15421
54 20322
15531 37644
55 0336000064 *

LX2 A0-18
BX4 -X1*X2 . RA+FL OF EOP
LX3 18
BX5 -X1*X3 . RA OF MCSP
IX6 X5-X4
NG X6,E35 . EOP COLLIDES WITH MCSP, JUMP

*
* MOVE ONE DOWN THE MAP CHAIN
*

66440 20244
56 0400000050 *

SB4 R6 . PMCSP REPLACED BY MCSP
LX2 36 . RESTORE X2 FOR LOOP
EQ E10 . PROCESS NEW MCSP

*
* AT THE TOP OF THIS LOOP, WE HAVE AT LEAST ONE SUBP OF THE FULL PATH
* IN.
*
* IS CURRENT FPSP IN?

```

57 56215      63620      0560000104 +
      E20      SA2      R1+B5      . GET FPSP POINTER WORD
      SB6      X2      . MAPIN FLAG
      NE      R6,B0,E50 . JUMP IF FPSP ALREADY IN

*
* SEE IF THERE ARE ANY SUBPS LEFT ON THE MAP CHAIN
*
60 56514      6265000004
61 0460000112 +
      E30      SA5      R1+B4      . GET POINTER WORD OF PMCSP
      SB6      X5+SD,PTRs . PTR TO MCSP
      EQ      R6,B0,E50 . JUMP IF END OF MAP CHAIN

*
* SEE IF FPSP IS ABOVE MCSP
*
      56275      20252
      SA2      R7+B5      . RA,FL WORD OF FPSP
      LX2      60-18
      BX4      -X1*X2      . RA OF FPSP
      SA3      R7+B6      . RA,FL WORD OF MCSP
      LX3      60-38
      BX5      -X1*X3      . RA+FL OF MCSP
      IX6      X4-X5
      PL      X6,E40      . JUMP IF FPSP ABOVE MCSP

*
* WE HAVE COLLIDED WITH A SUBP IN THE MAP CHAIN.
* DECHAIN IT AND SWAP IT OUT.
* BEND PTR IN PMCSP.
*
64 56316      56414      11613      54630
      E35      SA3      R1+B6      . MCSP'S CHAIN POINTER
      SA4      R1+B4      . PMCSP'S CHAIN POINTER
      BX6      X1*X3
      SA6      A3      . CLEARED PTR MEANS NOT IN CORE
      BX3      X3-X6      . POINTER
      BX6      X1*X4
      BX6      X6-X3
      SA6      A4      . UPDATE PTR IN PMCSP

*
* SWAP OUT MCSP
*
66 6170000070 +
      50530000002
67 0400000000 X
      SB7      E36      . RETLINK
      SA5      A3+SD,MAP-SD,PTRs . GET MAP INFO FOR MAPOUT
      EQ      =XMAPOUT . SWAP HIM OUT

*
* CHECK FOR UNFORTUNATE/ANNOYING ERRORS
*
70 6171000003      03040000060 +
71 03360000060 +
      E36      SB7      R1+SD,RAFL-SD,PTRs . RESTORE B7
      ZR      X6,E30      . WHEW, NO ERRORS
      NG      X6,E30      . IGNORE MAP OFF CONDITION

*
* FLAG SUBP FOR FILE-RIPPED-OFF MAP ERROR
*
72 43601      5055777771
      20673      12656
      SA5      A5-SD,MAP+SD,FIPT . GET FLAG WORD
      MX6      1
      LX6      60-SD,FMPE      . POSITION FLAG
      BX6      X5+X6

```



```

73 0400000060 + 54650 SA6 A5 . STORE FLAGGED SUBP HEADER
EQ E30 . CONTINUE

*
* SPLICE FPSP INTO MAP CHAIN AND SWAP FPSP IN
*
* PATCH INTO MAP CHAIN
*
74 7165777773 E40 SX6 R5-SD.PTRS . NEW POINTER FOR PMCSP
56574 SA5 R1+B4 . PMCSP POINTER WORD
11415 BX4 X1*X5 . ELIMINATE OLD PTR
75 12646 BX6 X4+X6
54650 SA6 A5 . PMCSP NOW POINTS TO FPSP
13654 BX6 X5-X4 . POINTER FROM PMCSP
56415 SA4 R1+B5 . PTR WORD OF FPSP
76 12646 BX6 X4+X6
54640 SA6 A4 . FPSP NOW POINTS WHERE PMCSP USED TO

*
* SWAP IN FPSP
*
77 6170000100 + 5052777776 SA5 A2+SD.MAP-SD.RAFL . MAP INFO FOR MAPIN
0400000000 X SB7 E45 E45 . RETLINK
EQ BXMAPIN
100 6171000003 E45 SB7 R1+SD.RAFL-SD.PTRS . RESTORE B7
0314000103 + NZ X6,E46 . JUMP IF MAP ERROR WHILE SWAPPIN IN
101 5053777771 SA5 A5-SD.MAP+SD.FIPT . SEE IF IT HAS ONE PENDING
20561 LX5 SD.FMPE . FROM LAST SWAPOUT
102 0325000104 + PL X5,E50 . JUMP IF NO ERROR PENDING
7160000001
103 5161000050 E46 SX6 I
SA6 R1+P.SCR2 . REMEMBER THE ERROR

*
* NOW FPSP IS IN AND BECOMES PMCSP
*
104 66450 E50 SB4 R5 . ADJUST PMCSP = FPSP

*
* MAKE ENTRY IN FULL C-LIST FOR FPSP
*
76451 SX4 R5+B1
5244777776 SA4 X4-SD.PTRS+SD.CLST . STORE CLIST UN.MOT
105 10644 BX6 X4
5163777776 SA6 B3-1
106 5044777774 SA4 A4-SD.CLST+SD.FIPT . STORE C-LIST LENGTH
20452 LX4 A0-18
15641 BX6 X1*X4
5066777776 SA6 A6-1
64340 SB3 A6 . ADJUST FULLC-TABLE ADDR

*
* ADVANCE TO NEXT SUBP IN FULL PATH, IF ONE EXISTS
*
110 0452000132 + EQ R5,B2,E71 . ALL ARE IN
56275 SA2 R1+B5 . PTR WORD OF FPSP
21222 AX2 T8
111 6252000004 SB5 X2+SD.PTRS . NEW FPSP
0400000057 + EQ E20 . BRING IN ANOTHER ONE

```

```

*
* THERE ARE NO SUBPS BELOW US IN CORE, SO WE CAN JUST SWAP IN #TIL
* WE HAVE THE REST OF THE FULL PATH IN.
*
* MAKE PMOSP POINT TO FPSP, FPSP BECOMES PMOSP
*
112 7165777773      6175000002      E60      SX6      B5-SD.PTRS      . PTR FOR PMOSP
113 56514           11515           SB7      B5-SD.PTRS+SD.MAP . USED SOON
114 66450           12656           SA5      R1+B4      . PMOSP PTR WORD
115 66450           54650           BX5      X1*X5      . WIPE OLD PTR
116 66450           54650           BX6      X5*X6
117 66450           54650           SA6      A5      . NEW PTR WORD
118 66450           54650           SB4      B5      . FPSP BECOMES PMOSP
*
* SWAP IN FPSP
*
119 56517           6170000116 +    SA5      R1+B7      . MAP INFO FOR MAPIN
120 0400000000 X    SB7      R65      . RETLIN
121 6171000003      0314000121 +    EQ      =XMAPIN
122 5055777771      20501           E65      SB7      R1+SD.RAFI-SD.PTRS . RESTORE B7
123 0325000122 +    7160000001      NZ      X6,E66      . JUMP IF MAP ERROR WHILE SWAPPIN IN
124 5161000050      7160000001      SA5      A5-SD.MAP+SD.FIPT . SEE IF IT HAS ONE PENDING
125 5161000050      7160000001      LX5      SD.FMDE      . FROM LAST SWAPOUT
126 5161000050      7160000001      PL      X5,E67      . JUMP IF NO ERROR PENDING
127 5161000050      7160000001      SX6      !
128 5161000050      7160000001      SA6      R1+P.SCR2      . REMEMBER THE ERROR
*
* MAKE ENTRY IN FULL C-LIST FOR FPSP
*
129 76451           5244777776      E67      SX4      B5+B1
130 5163777776      5244777776      SA4      X4-SD.PTRS+SD.CLIST . STORE CLIST UN.MOT
131 20452           15641           BX6      X4
132 6036000000      04520000130 +    SA6      B3-1
133 6036000000      04520000130 +    SA4      A4-SD.CLIST+SD.FIPT . STORE C-LIST LENGTH
134 6036000000      04520000130 +    LX4      B0-18
135 6036000000      04520000130 +    BX6      =X1*X4
136 6036000000      04520000130 +    SA6      A6-1
137 6036000000      04520000130 +    SB3      A6+0      . ADJUST FULLC-TABLE ADDR
138 6036000000      04520000130 +    EQ      B5,B2,E70 . ALL DONE SWAPPING
*
* ADVANCE TO NEXT SUBP IN FULL PATH
*
139 56215           21222           SA2      R1+B5      . PTR WORD
140 6252000004      21222           AX2      T8
141 0400000112 +    6252000004      SB5      X2+SD.PTRS
142 0400000112 +    6252000004      EQ      E60
*
* FULL PATH CODE IS IN: END THE MAP CHAIN, GET LOCAL C-LIST,
* SET UP DAE MAP ENTRY, SET RA,FL, ETC.
*
* END MAP CHAIN
*
143 7160777773      E70      SX6      -SD.PTRS      . -SD.PTRS IS LAST FLAG

```

131	12656	54650	15641	56515	BX6 SA5 BX6 SA6	-X1*X6 R1+B5 X5+X6 A5	. PTR WORD OF CURRENT SUBP . STICK IN FLAG . SET END OF MAPIN CHAIN
* SET RA, FL, MA IN XPACK AND SET S-USERA							
132	56275	20252	63750	73621	E71 SA2 LX2 SB7 SX6 SA6 SA3	R7+B5 60-18 X2 X2+B1 =XS-USERA	. RA, FL OF CURRENT SUBP . ABSOLUTE USER RA
133	5160000000	X	5131000141		SA3 LX3 BX3 BX6 LX6 SA6	R1+P.XPACK+1 6+18 X1*X3 X3+X6 36 A3	. RA WORD FROM XPACK . WIPE OLD RA . STICK IN NEW RA . NEW RA IN XPACK
134	20330	11313	12656	20644	SA3 SB7 SX6 SA3 LX3 BX3 BX6 LX6 SA6	R1+P.TEMP7 -B7 X3+B7 R1+P.XPACK+2 6+18 X1*X3 X3+X6 36 A3	. RETRIEVE RA+FL OF END OF PATH . CALCULATE FL OF FULL PATH . FL WORD . NEW FL IN XPACK
135	54630	5131000044	67707		* FINISH P-CLIST		
136	73637	5131000142	20330		* DO DIRECT ECS ACCESS MAP ENTRY (IF IT EXISTS)		
137	11313	12636	20644	54630	SX6 SA6	B3-B1 R1+P-CLIST	. SET PTR TO LOCAL C-LIST
140	77631	5161000140			SA1 SA2 MX7 BX1 BX2 SB5 SA3 LX3 PL	R1+P.XPACK+4 R1+P.XPACK+5 34 -X7*X1 -X7*X2 R2+SD-MAP-SD.PTRS+MP-CHAP R1+B5 1 X3+E75 23 39	. CLEAR OLD ECSRA.ECSFL . GET A (FIRST MAP ENT) . FIRST LOGICAL MAP ENTRY . INSPECT DAE BIT . JUMP IF NO DAE ENTRY . GET ECS ADDR
141	5111000144	5121000145			MX7 BX6 SA4 IX6 LX6 BX1 LX3 BX6	-X7*X3 =XS-ECSRA X4+X6 36 X1+X6 36 -X7*X3	. ISOLATE ADDR . GET SYSTEM ECS RA . USER ECSRA = LOCAL ADDR+ECS RA FOR SYSTEM . INSERT IN XPACK WORD . GET ECSFI = WORD COUNT . ISOLATE WORD COUNT
142	43730	15117	15257				
143	6152000005	54315	20301				
144	0323000150	20327	43747				
145	15637	5140000000	X	36646			
146	20644	12116	20344	15637			

SUBPROCESS OPERATIONS
ENVIORN - ESTABLISH ENVIORNMENT OF TOP OF STACK

COMPASS - VER 2. 11/03/71 15.38.13.

PAGE 48

147	20644		LX6	36	. POSITION IT
	12226		BX2	X2+X6	. STICK IT IN XPACK WORD
150	10611	E75	BX6	X1	
	10722		BX7	X2	
	54670		SA6	A1	. SET ECSRA, ECSFL IN XPACK
	54720		SA7	A2	
* RESTORE RETURN LINKS					
151	5111000051		SA1	B1+P.SCR2+1	
	63670		SB6	X1	
	21122		AX1	18	
152	63710		SB7	X1	
* READ IN LOCAL C-LIST					
	56430		SA4	B3	. LENGTH OF LOCAL C-LIST
	20401		LX4	1	
	63540		SB5	X4	. LENGTH IN WORDS
153	5044000001		SA4	A4+1	. UNIQUE NAME NOT
	73040		SX0	X4	
	56010		SA0	B1	
154	0110000001		RECS	1	. READ NOT
155	5131000000		SA4	B1+0	
	13444		BX4	X3-X4	
	43647		MX6	39	
156	11464		BX4	X6+X4	
	0314000145 +		NZ	X4-E80	
	15036		BX0	BX6+X3	
157	43173		MX1	39	
	5101000147		SA0	B1+P.LOCALC	
	37001		IX0	X0-X1	. SKIP LENGTH WORD
160	0115000000		RECS	B5	. READ THE CLIST
* EXIT IF NO ERROR					
161	5111000050		SA1	B1+P.SCR2	. INSPECT ERROR FLAG
	0311000163 +		NZ	X1-E76	. ON DEAR
162	0270000000		JP	37	. RETURN TO CALLER
163	7160000005	E76	SX6	F.PROC	. SIGNAL MAP HASSLE
	7170000005		SX7	F.MAPER	
164	0260000000		JP	B6	. TAKE ERROR EXIT
* THE LOCAL C-LIST IS GONE					
165	7160000005	E80	SX6	F.PROC	
	7170000004		SX7	F.OGONE	
166	5111000050		SA1	B1+P.SCR2	. SEE IF A MAP WAS BAD TOO
	0301000170 +		ZR	X1-E81	. NO
167	7170000006		SX7	F.BOTH	. GIVE THE CALLER THE BAD NEWS
170	0260000000	E81	JP	B6	. TAKE ERROR EXIT
171			END		

SUBPROCESS OPERATIONS
ENVIORN - ESTABLISH ENVIORNMENT OF TOP OF STACK

COMPASS - VER 2.

11/03/71 15.38.13.

PAGE 49

36362 STORAGE USED
6600 ASSEMBLY

2503 STATEMENTS
13.160 SECONDS

477 SYMBOLS
808 REFERENCES

SUBPROCESS OPERATIONS
SYMBOLIC REFERENCE TABLE.

COMPASS - VER 2.

11/03/71 15.38.13.

PAGE 50

ADR4	55		3/22 D	3/46 S	6/09	11/50 S	37/35		
BDTBLK	0	EXTERNAL*	34/45						
BIGIX	0	EXTERNAL*	34/41						
BIGPAR	0	EXTERNAL*	8/07	15/45	16/49	20/10			
BUFA	450		15/18	15/46	32/16	34/46	36/30	41/02	
			15/46	15/46	34/46	34/46	41/02	41/02	
BUF8	650		29/31	31/36	31/36	31/36			
CHKPTR	0	EXTERNAL*	22/16	23/11					
CLMOT	0	EXTERNAL*	8/08	34/43					
CMBUFF	50		3/29	16/16	18/12	22/09	24/11	26/02	28/11
			8/13	16/50	20/13	22/23	24/21	26/06	28/21
			9/09	17/12	21/10	23/08	25/02	27/03	
			12/18	17/49	21/20	23/21	25/06	27/07	
COMP	0	EXTERNAL*	8/11						
CRESUBP	51		3/33 L						
CRESUB1	53		3/34	3/36 L					
CRESUB10	135		5/44	5/45	5/51 L				
CRESUB2	55		3/38	3/40 L					
CRESUB26	200		3/36	8/08 L					
CRESUB27	201		3/40	8/09 L					
CRESUB28	174		4/28	8/04 L					
CRESUB3	65		4/22	4/05 L					
CRESUB31	202		4/21	8/10 L					
CRESUB32	203		5/25	8/11 L					
CRESUB33	177		4/50	8/07 L					
CRESUB34	173		3/50	8/03 L					
CRESUB35	175		4/24	8/05 L					
CRESUB4	76		4/25	4/27 L					
CRESUB7	116		5/08	5/11 L					
CRESUB8	127		5/29	5/38 L					
CRESUB9	131		5/41 L	6/03					
CRESUB95	132		5/29	5/44 L					
CRE28X	176		4/10	8/06 L					
DELSUB	51		9/10 L						
DELSUB1	53		9/11	9/13 L					
DELSUB10	111		10/33	10/47 L					
DELSUB11	113		11/01 L	11/20					
DELSUB12	116		11/06	11/08 L					
DELSUB13	120		11/11	11/13 L					
DELSUB14	122		11/02	11/16	11/18 L				
DELSUB15	126		11/25	11/28 L					
DELSUB16	131		11/33	11/35 L					
DELSUB17	136		11/46	11/48 L					
DELSUB18	142		12/02	12/07 L					
DELSUB2	60		9/29 L	9/36					
DELSUB27	144		9/46	12/13 L					
DELSUB28	145		10/27	12/14 L					
DELSUB29	147		9/13	12/16 L					
DELSUB3	65		9/44 L	9/48					
DELSUB31	143		9/18	9/32	12/12 L				
DELSUB37	146		10/39	12/15 L					
DELSUB4	72		9/53	10/05 L					
DELSUB5	71		10/02 L	10/17					

SUBPROCESS OPERATIONS
SYMBOLIC REFERENCE TABLE.

COMPASS - VER 2.

11/03/71 15.38.14.

PAGE 51

DELSUB6	75		10/05	10/06	10/12 L
DELSUB7	103		10/28	10/29 L	
DELSUB8	104		10/31 L	10/35	
DELSUB9	107		10/32	10/36 L	
DISASTR	0	EXTERNAL*	12/14	12/15	
DSPSP1	53		18/13	18/15 L	
DSPSP10	110		18/25	20/09 L	
DSPSP11	111		18/27	20/10 L	
DSPSP12	112		18/15	20/11 L	
DSPSP2	61		18/28	18/35 L	
ENDSYS	0	EXTERNAL*	5/18		
ENVIORN	36	PROGRAM*	30/22	40/53	42/39 E 42/40 L
ERRPROC	0	EXTERNAL*	30/24	30/39	
ERR1	517		32/20	34/37 L	
ERR10	526		32/26	34/44 L	
ERR2	520		32/22	34/38 L	
ERR3	521		32/24	34/39 L	
ERR4	522		33/13	34/09	34/40 L
ERR5	523		32/31	34/41 L	
ERR6	524		33/36	34/42 L	
ERR7	525		32/39	34/43 L	
ESMSET	51		23/28 L		
ESMSETN	51		22/09 L		
ESMSETN1	53		22/10	22/12 L	
ESMSETN2	56		22/15	22/17 L	
ESMSETN3	60		22/12	22/21 L	
ESMSET1	53		23/10	23/12 L	
ESTK	0	EXTERNAL*	31/35		
E10	50	PROGRAM*	43/21 L	43/48	
E20	57	PROGRAM*	44/02 L	45/53	
E30	60	PROGRAM*	44/08 L	44/45	44/46 45/02
E35	64	PROGRAM*	43/42	44/27 L	
E36	70	PROGRAM*	44/28	44/44 L	
E40	74	PROGRAM*	43/33	44/21	45/08 L
E45	100	PROGRAM*	45/21	45/23 L	
E46	103	PROGRAM*	45/24	45/29 L	
E50	104	PROGRAM*	43/17	44/04	45/27 45/33 L
E60	112	PROGRAM*	43/23	44/10	46/07 L 46/46
E65	116	PROGRAM*	46/18	46/20 L	
E66	121	PROGRAM*	46/21	46/26 L	
E67	122	PROGRAM*	46/24	46/30 L	
E70	130	PROGRAM*	46/39	46/53 L	
E71	132	PROGRAM*	45/49	47/08 L	
E75	150	PROGRAM*	47/44	48/03 L	
E76	163	PROGRAM*	48/39	48/41 L	
E80	165	PROGRAM*	48/28	48/47 L	
E81	170	PROGRAM*	48/50	48/52 L	
E.BOTH	6		48/51		
E.CGONE	4		48/48		
E.DLSUB	133	ERUBP	9/08 L		
E.DSPSP	302	ERUBP	18/11 L		

SUBPROCESS OPERATIONS
SYMBOLIC REFERENCE TABLE.

COMPASS - VER 2.

11/03/71 15.38.14.

PAGE 52

E.ECS	0	EXTERNAL*	4/46	31/36	33/45	34/46	39/28	39/45	40/32	48/24
			4/53	32/45	33/50	36/40	39/36	40/02	40/42	48/34
			15/46	33/32	34/25	38/03	39/40	40/20	41/02	
E.ERCAL	344	EAUBP	21/39 L							
E.ERROR	0	EXTERNAL*	40/52							
E.FINDS	232	EAUBP	15/17 L							
E.FRET	375	EAUBP	26/01 L							
E.FSON	262	EAUBP	17/11 L							
E.GETP	460	EAUBP	32/15 L							
E.JMRET	367	EAUBP	24/10 L							
E.MAPER	5		48/42							
E.MKSUB	0	EAUBP	3/28 L							
E.MODPC	247	EAUBP	16/15 L							
E.OMYGO	537	EAUBP	36/29 L							
E.PRET	401	EAUBP	28/10 L							
E.PROC	5		48/41	48/47						
E.RET	377	EAUBP	27/02 L							
E.SESM	351	EAUBP	22/08 L							
E.SESMG	361	EAUBP	23/07 L							
E.SPRET	373	EAUBP	25/01 L							
E.SRETN	405	EAUBP	29/20 L							
FATHPTR	64		3/16 D	3/44 S	6/36					
FINDNTHS	14	PROGRAM*	15/16 L	16/17	24/12					
FINDS	14	PROGRAM*	15/46 L							
FINDSUB	25	PROGRAM*	3/35	9/12	14/12	18/11	35/19 E			
			3/39	13/11	17/14	22/11	35/20 L			
FINOSUB1	30	PROGRAM*	35/31 L	35/35						
FNS1	454		15/27 L	15/36						
FNS2	460		15/31	15/35 L						
FNS3	462		15/34	15/38 L						
FNS4	463		15/20	15/43 L						
FNS5	464		15/37	15/44 L						
FRET	51		26/02 L							
FSON1	53		17/13	17/15 L						
FSON10	67		17/15	17/46 L						
FSON11	70		17/20	17/47 L						
FSON2	57		17/24 L	17/30	17/32					
GETP	22	PROGRAM*	34/46 L							
GETRETP	22	PROGRAM*	29/29	32/14 L						
GETRETP1	466		33/10 L	33/14						
GETRETP2	470		33/08	33/17 L	34/04	34/16				
GETRETP3	471		33/18	33/20 L						
GETRETP4	473		33/21	33/25 L						
GETRETP5	506		34/03 L	34/28						
GETRETP7	511		34/15 L							
GETRETP8	512		34/16	34/18 L						
GETRETP9	516		32/53	34/03	34/31 L					
INSTK	0	EXTERNAL*	12/14							
I.LOCK	0	EXTERNAL*	39/49 S	40/32 S						
JPCAL1	2	PROGRAM*	13/10	13/12 L						
JUMPCAL	0	PROGRAM*	13/28 E	13/09 L						
LOCALC	63		3/15 D	5/02 S	6/11					
L.DLSUB	77		12/18 D							

SUBPROCESS OPERATIONS
SYMBOLIC REFERENCE TABLE.

COMPASS - VER 2.

11/03/71 15.38.14.

PAGE 53

L.DSPSP	42		20/13	D				
L.ERCAL	5		21/20	D				
L.FINDS	15		15/46	D	15/46			
L.FRET	2		26/06	D				
L.FSON	20		17/49	D				
L.GETP	57		34/46	D	34/46			
L.JMRET	4		24/21	D				
L.MKSUB	133		8/13	D				
L.MODPC	13		16/50	D				
L.OMYGO	117		41/02	D	41/02			
L.PRET	4		28/21	D				
L.RET	2		27/07	D				
L.SESM	10		22/23	D				
L.SESMG	6		23/21	D				
L.SPRET	2		25/06	D				
L.SRETN	53		31/36	D	31/36			
MACSZ	0	EXTERNAL*	8/12					
MAPIN	0	EXTERNAL*	45/22		46/19			
MAPOUT	0	EXTERNAL*	5/43		10/04	44/40		
MODPC1	62		16/29		16/47	L		
MODPC2	63		16/30		16/48	L		
MP.CHAP	3		47/41					
NEGIX	0	EXTERNAL*	34/42					
NEGPARG	0	EXTERNAL*	8/04		8/06	16/48	20/11	34/40
			8/05		15/44	17/48	34/39	
NEGPT	0	EXTERNAL*	34/38					
NEWTO	0	EXTERNAL*	14/22					
NLEAF	0	EXTERNAL*	12/13					
NOFATH	0	EXTERNAL*	8/10		17/47			
NOFIND	0	EXTERNAL*	12/17		14/26	20/12	22/22	
OMYGO	33	PROGRAM*	41/02	L				
OMYGOD	33	PROGRAM*	7/47		12/08	36/28	L	
OMY1	471		37/44		37/46	L		
OMY2	473		37/49		37/51	L		
OMY3	467		37/41	L	38/01			
OMY4	502		38/07		38/11	38/15	L	
OMY5	512		38/37		38/44	38/46	L	
OMY6	507		38/38	L	39/07			
OMY7	531		39/24		39/37	L		
OMY8	536		39/36		39/48	L		
PARMBUF	0	EXTERNAL*	32/42					
PATCH2	711		31/05	L				
PATCH3	720		31/27	L				
PRET	51		28/11	L				
PUTCAP	0	EXTERNAL*	17/40					
PUTRETP	0	EXTERNAL*	30/30					
P.CLIST	160		33/05		47/32	S		
P.CTABLE	161		4/20		5/03	36/50	42/53	
P.INTERR	165		31/23					
P.LOCALC	167		34/19		48/31			
P.MAPSIN	164		5/38		9/52	40/44	S	43/04

SUBPROCESS OPERATIONS
SYMBOLIC REFERENCE TABLE.

COMPASS - VER 2.

11/03/71 15.38.15.

PAGE 54

P.PARAM	66		3/55	16/27	17/35 S	18/12	21/10	23/0A	27/04 S
			15/18	17/12	17/36	18/17	22/09	25/03 S	28/11
			15/26	17/18	17/39	18/18	22/13	26/03 S	29/07
P.PARAMC	125		13/09	14/10	32/52 S				
P.PBUFL	372		32/35						
P.PROCRO	12		40/19	40/31					
P.PROCRW	27		40/19	40/31					
P.ROHEAD	126		3/53	5/31	36/33	39/53	40/33		
			5/31	12/03	36/38	40/28			
P.SCR	0		33/39						
P.SCR1	50		33/30						
P.SCR2	50		3/17	3/20	29/33 S	30/37	42/46 S	48/10	
			3/18	3/21	29/40	30/40	45/29 S	48/38	
			3/19	3/22	30/25	42/44 S	46/26 S	48/40	
P.STACK	162		4/12	13/16	19/32	24/13	30/45	39/12	42/47
			9/40	15/22	23/17	29/42	37/14	40/38	
			9/22	10/47	17/21	35/20	38/06	39/11	
P.SUBPDT	163		10/25 S	10/29					
P.TEMP1	56								
P.TEMP6	62		3/15						
P.TEMP7	64		3/16	43/08 S	47/19				
P.XPACK	140		18/19	32/17	40/45	47/13	47/36		
			18/25	32/36	40/48	47/22	47/37		
REALLOC	0	EXTERNAL*	5/34	12/06					
REFER	0	EXTERNAL*	10/40						
RET	51		27/03 L	27/07					
RETURN	51		27/01 E	27/07 D					
ROOM	0	EXTERNAL*	34/44						
SAMNA	0	EXTERNAL*	8/19						
SD.CC	1		3/23	3/47	6/21 S	14/13	17/16	18/16	19/11
			3/45	4/11	9/14	15/20	17/33	19/03	22/19 S
			6/22 S	19/53	45/38	45/41	46/31	46/34	35/21
SD.CLST	2		6/26 S	19/45	22/19 S	23/17			39/34
SD.ESM	2								
SD.FDIS	2		20/52	21/12					
SD.FINT	0		20/50	20/52	31/01	31/13			
SD.FIPT	0		6/15 S	7/04	44/50	45/23	45/41	46/22	46/34
SD.FMPE	1		5/48	10/09	44/52	45/24	46/23		
SD.MAP	6		5/40	7/24 S	10/21	44/30	45/20	46/08	47/41
			6/51	10/01	19/28	44/50	45/25	46/22	
SD.ORIG	5		6/50	7/11 S	19/19				
SD.PTRS	4		4/11	9/28	18/30	20/43	44/09	45/38	46/31
			5/51	10/12	19/30	42/50	44/39	45/53	46/45
			6/38	11/08	31/01	42/52	44/44	46/07	46/53
			6/44 S	11/29	38/33	43/02	45/08	46/08	47/41
			9/15	17/27	38/38	43/23	45/23	46/20	
SD.RAFL	7		3/47	7/40 S	11/42	18/37	43/02	45/20	46/20
			7/28	11/03	16/21	38/48	44/44	45/23	
SD.SIZE	10		4/40	39/37					
SF.PC01	4		16/34	16/36	30/07	30/09	30/14	30/16	
SF.PC02	5		16/36	30/09	30/16				
SPRET	51		25/02 L						
SP.CLCD	66		3/07 L	3/33	6/19	9/10			
SP.CLIST	76		3/13 L	4/42	6/30				
SP.COMP	73		3/10 L	4/28	7/16	7/37			

SUBPROCESS OPERATIONS
SYMBOLIC REFERENCE TABLE.

COMPASS - VER 2.

11/03/71 15.38.15.

PAGE 55

SP.ENTRY	75		3/12 L	4/06	6/48				
SP.FACC	70		3/08 L	3/37					
SP.FL	74		3/11 L	3/49	7/31				
SP.MAP	72		3/09 L	4/32	7/19				
SRETN	17	PROGRAM*	31/26 L						
SUBPCAL	5	PROGRAM*	14/07 E	14/10 L					
SUBPC1	7	PROGRAM*	14/11	14/13 L					
SUBPC15	13	PROGRAM*	13/12	14/13	14/25 L				
SUBPC2	10	PROGRAM*	13/24	14/17 L					
SUPRET	17	PROGRAM*	21/18	25/04	26/04	27/05	28/19	29/29 L	
SUPRET1	655		29/26	29/40 L					
SUPRET10	701		30/38	30/40 L					
SUPRET11	705		30/49 L	31/04					
SUPRET12	713		30/53	31/11 L					
SUPRET13	712		30/51	31/06 L					
SUPRET2	656		29/35	29/42 L					
SUPRET3	663		29/52	30/04 L					
SUPRET4	665		30/05	30/11 L					
SUPRET5	667		30/12	30/18 L					
SUPRET6	671		30/20	30/23 L					
SUPRET7	672		30/21	30/25 L					
SUPRET8	675		30/23	30/27	30/28	30/29	30/31 L		
SUPRET9	702		30/33	30/42 L					
SUPRET90	722		29/48	31/34 L					
SUB.BC	70		29/10 L	29/38					
SUB.BD	67		29/19 L	29/37					
SUB.BC	71		21/12	29/11 L	30/34				
SUB.BN	72		21/15	29/12 L	30/36				
SUB.INT	73		29/13 L	31/18					
SUB.KEY	66		21/17 S	29/08 L	29/31				
SYSFRET	0	EXTERNAL*	17/25						
SYSRET	0	EXTERNAL*	7/44	16/41	18/49	19/15	19/41	20/03	23/19
			12/07	17/37	19/07	19/23	19/49	22/20	
S.CHFL	0	EXTERNAL*	5/19	40/23					
S.ECSRA	0	EXTERNAL*	47/48						
S.USFRA	0	EXTERNAL*	40/30 S	47/12 S					
S.USRBI	0	EXTERNAL*	40/27 S						
TOSPROC	0	EXTERNAL*	14/20	14/21	24/19	30/43	31/06	31/28	
V1	50		3/17 D	5/11 S	11/28 S	36/51	40/03		
V2	51		3/18 D	4/27 S	5/16	11/35 S	36/52	40/04	
V3	52		3/19 D	7/46 S	11/37 S	37/15	40/05		
V4	53		3/20 D	5/14	6/12	12/01	39/23		
			4/41 S	5/30	11/41 S	37/33	40/08		
V5	54		3/21 D	4/05 S	5/12	11/43 S	40/11		