ADDRESS	LENGTH
0	56
56	

BINARY CONTROL CARDS.

```
IDENT    PROCESS
END
```

BLOCKS	TYPE	ADDRESS	LENGTH
ABSOLUTE*	ABSOLUTE	0	171
PROGRAM*	LOCAL	0	56
EPROC	COMMON	0	533

ENTRY POINTS.

E.CPROC	-	0	E.STMSG	-	372	ARMIT	-	51	L.PTINT	-	1
L.CPROC	-	227	L.STMSG	-	22	F.ARMIT	-	424	CHKSCHD	-	0
CPUIN	-	51	E.DLPRO	-	414	L.ARMIT	-	10	PRIRINT	-	11
E.CPUIN	-	227	L.DLPRO	-	1	DISIT	-	51	PRINT	-	11
L.CPUIN	-	53	CLIIB	-	51	F.DISIT	-	434	E.PRINT	-	521
CPOUT	-	51	E.CLIIR	-	415	L.DISIT	-	21	L.PRINT	-	12
E.CPOUT	-	302	L.CLIIR	-	3	E.DSPST	-	455	S.CHIP	-	14
L.CPOUT	-	53	STIIB	-	51	L.DSPST	-	26	S.ARITH	-	34
CLRMG	-	51	E.STIIB	-	420	DSSTE	-	51	CHIPCAL	-	54
E.CLRMG	-	355	L.STIIB	-	3	F.DSSTE	-	503	ARITHER	-	55
L.CLRMG	-	15	E.DPROD	-	423	L.DSSTE	-	15			
STMSG	-	51	L.DPROD	-	1	E.PTINT	-	520			

EXTERNAL SYMBOLS.

E.ERROR	TOSPROC	NOFIND	NOROOM	NEGIX	DISASTR	MAKEOBJ	E.ECS
I.LOCK	FINDSUB	FATSON	BLMISS	BIGPT	SYSRET	REFEB	
SCHED	CHKPTR	NOCP	CLMOT	BIGPAR	PUTCAP	S.CMEL	
SYSFRET	NEGPT	MISSOR	BIGIX	NEGPAR	CAPAB	ENDSYS	

```
                IDENT   PROCESS
*
* MOST ROUTINES CONCERNED WITH GOLBAL MANAGEMENT OF THE PROCESS DESCRIPT
* (AS OPPOSED TO BEING CONCERNED WITH A PARTICULAR SUBPROCESS OR MAP)
* ARE COLLECTED HERE.  EXCEPTIONS ARE MANY, INCLUDING -
*         STACK MANIPULATIONS IN SYSENT
*         ESTABLISHMENT OF PROCESS CORE IN SUBPROC
*
* SYSTEM ACTIONS HEREIN -
*         MAKPROC   CREATE A PROCESS
*         DELPROC   DELETE A PROCESS
*         DISPROC   DISPLAY A PROCESS
*         SETMESS   SET MESSAGE MECHANISM
*         CLRMESS   CLEAR MESSAGE MECHANISM
*         TIMIN     MOVE TIME INTO PROCESS
*         TIMOUT    MOVE TIME OUT OF PROCESS
*         SETIIB    SET INTERRUPT INHIBIT BIT
*         CLRIIB    CLEAR INTERRUPT INHIBIT BIT
*         ARMINT    ARM INTERRUPTS FOR A GIVEN SUBP
*         DISARM    DISARM INTERRUPTS FOR A GIVEN SUBP
*         DSPST     DISPLAY STACK
*         DSSTE     DISPLAY STACK ENTRY
*         PTINT     SEND INTERRUPT
*
* EXTERNAL SUBROUTINES HEREIN -
*         PRIRINT   SEARCH FOR INTERRUPT SUBPROCESS WITH PRIORITY
*                   OVER CURRENT SUBP
*
* INTERNAL SUBROUTINES HEREIN -
*         ARYTH     HANDLE USER ARITHMETIC ERRORS
*         CHIP      HANDLE USER CHIP CALL ERRORS
*


0       ERRNUMS   XTEXT
0       INTSYS    XTEXT
        RECS      MACRO   K
        +         RE      K
                  RJ      =XE.ECS
        RECS      ENDM
        WECS      MACRO   K
        +         WE      K
                  RJ      =XE.ECS
        WECS      ENDM
0       TYPES     XTEXT
0       ALOCSYM   XTEXT
0       ECSMAC    XTEXT
        CBLOCK    MICRO   1,,*/EPROC/*
                  EXT     E.ERROR,I.LOCK.SCHED,SYSFRET
                  EXT     TOSPROC,FINDSUB
```

```
                              LIST      X
     0               PROCSYM  XTEXT
                     *                                                                PROCSYM     1
                     *                  PROCESS SYMBOLS                               PROCSYM     1
                     *                                                                PROCSYM     1
                 40  P.PARAML  EQU      40B      . SIZE OF PARAM BUF + SUBP CALL WORDS PROCSYM    1
                 35  P.APLL    EQU      P.PARAML-3       ACTUAL SPACE FOR PARAMETERS  PROCSYM     1
                  4  P.SCHERL  EQU      4        . LENGTH OF SCHEDULER AREA           PROCSYM     1
                 50  P.SCRL    EQU      50B      . LENGTH OF PRINCIPAL SCRATCH AREA   PROCSYM     1
                  6  P.SCR2L   EQU      6        . LENGTH OF ANOTHER SCRATCH AREA     PROCSYM     1
                372  P.PBUFL   EQU      250      . BLOCK PARAMETER BUFFER LENGHT      PROCSYM     1
                     *                                                                PROCSYM     1
     0                         ORG      0                                             PROCSYM     1
     0               P.SCR     BSS      P.SCRL   . PRIMARY SCRATCH AREA               PROCSYM     1
    50               P.SCR2    BSS      P.SCR2L  . ANOTHER SCRATCH AREA               PROCSYM     1
    56               P.TEMP1   BSS      1        . . MORE SCRATCH STORAGE             PROCSYM     1
    57               P.TEMP2    BSS     1                                             PROCSYM     1
    60               P.TEMP3   BSS      1                                             PROCSYM     1
    61               P.TEMP4   BSS      1                                             PROCSYM     1
    62               P.TEMP5   BSS      1                                             PROCSYM     1
    63               P.TEMP6   BSS      1                                             PROCSYM     1
    64               P.TEMP7   BSS      1                                             PROCSYM     1
                     *                                                                PROCSYM     1
    65                         BSS      1        . AN XTRA STORE IN P.PARAM LANDS HERE PROCSYM    1
    66               P.PARAM   BSS      P.PARAML . ACTUAL PARAMETER AREA              PROCSYM     1
                125  P.PARAMC  EQU      *-1      . END OF ACTUAL PARM AREA (CLASS CODE, PROCSYM   1
                     *                           . + 2 OTHER WORDS GO HERE ON SUBP CALL) PROCSYM  1
                     *                                                                PROCSYM     1
                     *                                                                PROCSYM     1
                     *                   PROCESS READ ONLY DESCRIPTOR                 PROCSYM     1
                     *                                                                PROCSYM     1
   126               P.ROHEAD  BSS      2                                             PROCSYM     1
                     *                                                                PROCSYM     1
                     * NAMES OF PROCESS STATE FLAGS,                                  PROCSYM     1
                     * FOR CROSS-REFERENCE PURPOSES ONLY.                             PROCSYM     1
                     *                                                                PROCSYM     1
                  0  PF.P      EQU      0        . SOMETHING PENDING AT SWAPIN TIME   PROCSYM     1
                  1  PF.W      EQU      1        . UNCHAIN FROM EVENT CHANNEL         PROCSYM     1
                  2  PF.R      EQU      2        . SCHEDULED                          PROCSYM     1
                  3  PF.I      EQU      3        . PENDING INTERRUPT                  PROCSYM     1
                  4  PF.D      EQU      4        . PENDING DESTRUCTION OF PROCESS     PROCSYM     1
                  5  PF.E      EQU      5        . PSEUDO-PROCESS FLAG                PROCSYM     1
                  6  PF.C      EQU      6        . PROCESS IS IN CORE                 PROCSYM     1
                  7  PF.V      EQU      7        . RESCHEDULED BY ARRIVAL OF EVENT    PROCSYM     1
                 10  PF.S      EQU      8        . DESCHEDULED FOR TIMER OUT          PROCSYM     1
                 11  PF.EE     EQU      9        . DESCHEDULED FOR ERROR ERROR        PROCSYM     1
                 12  PF.H      EQU      10       . DESCHEDULED CAUSE HUNG ON EVENT CHANNEL PROCSYM 1
               4010  PF.CREA   EQU      4010B    . STATE BITS FOR NEW PROCESS         PROCSYM     1
               4200  PF.KILL   EQU      4200B    . FLAGS SET TO KILL PROCESS          PROCSYM     1
               6636  PF.SWPN   EQU      6636B    . FLAGS ALWAYS TURNED OFF AT SWAPIN  PROCSYM     1
                 16  PF.DSCHD  EQU      0016B    . ANY OF THESE PREVENTS THE PROCESS  PROCSYM     1
                     *                           . FROM RUNNING                       PROCSYM     1
                     *                                                                PROCSYM     1
```

```
                    * BACK TO NAMING CELLS IN PROCESS DESCRIPTOR                                    PROCSYM  1
                    *                                                                              PROCSYM  1
130                 P.SCHED   BSS       P.SCHEDL                                                   PROCSYM  1
                    *                                                                              PROCSYM  1
134                 P.MSGEVC  BSS       1           . MESSAGE MECH - EVENT CHANNEL                 PROCSYM  1
135                 P.MSGDAT  BSS       1           . MESSAGE MECH - EVENT TO SEND                 PROCSYM  1
136                 P.TIMER   BSS       1           . PROCESS TIMER                                PROCSYM  1
 11     P.PROCRO  EQU       *-P.ROHEAD  . LENGTH OF PROC READ/ONLY DESCRIPTOR                      PROCSYM  1
                    *                                                                              PROCSYM  1
                    *                                                                              PROCSYM  1
                    *                   PROCESS READ / WRITE DESCRIPTOR                            PROCSYM  1
                    *                                                                              PROCSYM  1
 137    P.RWHEAD  EQU       *                                                                      PROCSYM  1
137                 P.USRTIM  BSS       1           . USER TIME                                    PROCSYM  1
140                 P.SYSTIM  BSS       1           . SYSTEM TIME                                  PROCSYM  1
141                 P.SWPTIM  BSS       1           . SWAP TIME                                    PROCSYM  1
142                 P.XPACK   BSS       16                                                         PROCSYM  1
162                 P.CLIST   BSS       1                                                          PROCSYM  1
163                 P.CTABLE  BSS       1                                                          PROCSYM  1
164                 P.STACK   BSS       1                                                          PROCSYM  1
165                 P.SUBPNT  BSS       1                                                          PROCSYM  1
166                 P.MAPSIN  BSS       1                                                          PROCSYM  1
167                 P.INTERR  BSS       1                                                          PROCSYM  1
170                           BSS       1           . CELL FOR SIZE OF LOCAL CLIST                 PROCSYM  1
  32    P.PROCRW  EQU       *-P.RWHEAD                                                             PROCSYM  1
171     P.LOCALC  EQU       *                                                                      PROCSYM  1
                    *                                                                              PROCSYM  1
                              USE       *                                                          PROCSYM  1
                    *                                                                              PROCSYM  1
                    *                                                                              PROCSYM  1
   0    PS.TEMP   SET       P.PARAML/60                                                            PROCSYM  1
                    IFNE      P.PARAML-PS.TEMP*60,1                                                PROCSYM  1
   1    PS.TEMP   SET       PS.TEMP+1           . ROUND UP                                         PROCSYM  1
   1    PS.MASKL  EQU       PS.TEMP             . LENGTH OF PARAMETER BIT MASK AREA                PROCSYM  1
                    *                                                                              PROCSYM  1
                    * SYMBOLIC NAMES OF THINGS IN THE SUBPROCESS DESCRIPTOR.                       PROCSYM  1
                    *                                                                              PROCSYM  1
                    * WORDS                                                                        PROCSYM  1
                    *                                                                              PROCSYM  1
  10    SD.SIZE   EQU       8           . SIZE OF SUBP DESC. EXCLUDING MAPS                        PROCSYM  1
   0    SD.FIPT   EQU       0           . FLAGS, INT DATUM, PTR TO NEXT SUBP                       PROCSYM  1
                    IFNE      SD.FIPT-0,1                                                          PROCSYM  1
   1    SD.CC     EQU       1           . CLASS CODE WORD                                          PROCSYM  1
   2    SD.ESM    EQU       2           . ERROR SELECTION MASK WORD                                PROCSYM  1
   3    SD.CLST   EQU       3           . UNIQUE NAME, MOT OF LOCAL CLIST                          PROCSYM  1
   4    SD.PTRS   EQU       4           . MAX STACK, FATHER, MAPIN CHAIN PTRS                      PROCSYM  1
   5    SD.ORIG   EQU       5           . ENTRY POINT, MAP AND C-LIST ORIGINS                      PROCSYM  1
   6    SD.MAP    EQU       6           . COMPILED AND LOGICAL MAP SIZE,                           PROCSYM  1
                    *                   . LOGICAL MAP POINTER                                      PROCSYM  1
                    * SD.MAP BEGINS THE MAP AREA AND MUST COME AT THE END                         PROCSYM  1
   7    SD.RAFL   EQU       SD.MAP+1    . RA+FL,RA,SPACE LEFT IN COMP MAP                          PROCSYM  1
```

```
          *                                                                            PROCSYM      1
          * FLAGS                                                                      PROCSYM      1
          *                                                                            PROCSYM      1
     0    SD.FINT    EQU      0            . INTERRUPT PENDING                          PROCSYM      1
     1    SD.FMPF    EQU      1            . MAP ERROR PENDING                          PROCSYM      1
     2    SD.FDIS    EQU      2            . INTERRUPTS DISARMED                        PROCSYM      1


          *                                                                            PROCSYM      1
          * ENTRY POINT OFFSETS FOR DIFFERENT SUBPROCESS CALLS                         PROCSYM      1
          *                                                                            PROCSYM      1
     0    P.CLLOFF   EQU      0            . NORMAL SUBPROCESS CALL                     PROCSYM      1
     1    P.ERROFF   EQU      1            . OFFSET FOR ERROR CALLS                     PROCSYM      1
     2    P.INTOFF   EQU      2            . INTERRUPT CALL                             PROCSYM      1
     3    P.ITLOFF   EQU      3            . INITIAL CALL OF PROCESS                    PROCSYM      1
          *                                                                            PROCSYM      1
          * MAP NAMES - ALL ARE FOR CROSS-REFERNECE PURPOSES ONLY                      PROCSYM      1
          *                                                                            PROCSYM      1
     1    MP.RAFL    EQU      1            . RA.FL WORD REL TO SD.MAP WORD              PROCSYM      1
     2    MP.CNT     EQU      2            . MAP COUNT WORD, REL SD.MAP WORD            PROCSYM      1
     3    MP.CMAP    EQU      3            . FIRST COMPILED MAP WORD, REL SD.MAP        PROCSYM      1


     3    MP.SIZE    EQU      3            . SIZE OF ONE LOGICAL MAP ENTRY              PROCSYM      1
     0    MP.FILE    EQU      0            . FILE ID, REL THIS MAP ENTRY                PROCSYM      1
     1    MP.FADD    EQU      1            . . FILE ADDR, REL THIS MAP ENRY             PROCSYM      1
     2    MP.CADD    EQU      2            . CM ADDR, REL THIS MAP ENTRY                PROCSYM      1


          *                                                                            PROCSYM      1
          * THIS IS WHERE THE CLOCK FLAG IS SET                                        PROCSYM      1
          *                                                                            PROCSYM      1
     2    S.FNFLG    EQU      2            . THATS RIGHT, CELL 2 ABSOLUTE               PROCSYM      1
          *                                                                            PROCSYM      1
          *                                                                            PROCSYM      1
          * STACK FLAG NAMES.                                                          PROCSYM      1
          * SF.PCQ1 AND SF.PCQ2 ARE FOR CROSS-REFERENCE ONLY                           PROCSYM      1
          *                                                                            PROCSYM      1
     0    SF.II      EQU      0            . INTERRUPT INHIBITED                        PROCSYM      1
     1    SF.FINT    EQU      1            . INTERRUPTED                                PROCSYM      1
     4    SF.PCQ1    EQU      4            . ABOUT TO EXECUTE BIT                       PROCSYM      1
     5    SF.PCQ2    EQU      5            . ALMOST FINISHED BIT                        PROCSYM      1
                     LIST     -X
```

```
                                *
                                * ACTION TO CREATE A PROCESS.  THE PARAMETERS ARE EXPLAINED BELOW. START
                                * WITH API.  THE PROCESS IS CREATED COMPLETE WITH A ROOT SUBPROCESS WITH
                                * A R/W AND A R/O MAP ENTRY.  IT IS CREATED DESCHEDULED FOR TIMER OUT.
                                * MINOR KLUDGE - IT HAS THE PENDING ACTION FLAG SST TO GET
                                * THE P-COUNTER SET IN THE XPACK WHEN FIRST SWAPPED IN.
                                *
  0                                     ORG     0
                                *                       PARAMETER MNEMONICS
                                *
                                *
  0                             MP.ALLOC  BSS     2       CAPABILITY FOR ALLOCATION BLOCK
  2                             MP.CAPX   BSS     1       INDEX IN FULL C-LIST TO RETURN CAP
  3                             MP.CHWD   BSS     1       NUMBER OF CHAINING WORDS
  4                             MP.STACK  BSS     1       NUMBER OF ENTRIES IN THE STACK
  5                             MP.CLSCD  BSS     2       CLASS CODE FOR SUBPROCESS
  7                             MP.MAP    BSS     1       NUMBER OF ENTRIES IN THE SUBP MAP
 10                             MP.COMD   BSS     1       NUMBER OF WORDS RESERVED FOR COMPILED
 11                             MP.FL     BSS     1       FIELD LENGTH OF SUBPROCESS
 12                             MP.ENTRY  BSS     1       ENTRY POINT OF SUBPROCESS
 13                             MP.CLIST  BSS     2       C-LIST FOR SUBPROCESS
 15                             MP.FILE1  BSS     2       FILE FOR 1ST SWAPPING DIRECTIVE
 17                             MP.FLAD1  BSS     1       FILE ADDRESS IF 1ST SWAPPING DIRECTIVE
 20                             MP.CMAD1  BSS     1       CORE ADDRESS FOR 1ST SWAPPING DIR
 21                             MP.WDCT1  BSS     1       WORD COUNT FOR 1ST SWAPPIND DIRECTIVE
 22                             MP.FILE2  BSS     2       FILE FOR 2ND SWAPPING DIRECTIVE
 24                             MP.FLAD2  BSS     1       FILE ADDR FOR 2ND SWAPPING DIRECTIVE
 25                             MP.CMAD2  BSS     1       CORE ADDR FOR 2ND SWAPPING DIRECTIVE
 26                             MP.WDCT2  BSS     1       WORD COUNT FOR 2ND SWAPPING DIRECTIVE
                                *
                                        USE     *
                                *
                                *                       TEMPORARIES
                                *
                          56    CLEN      EQU     P.TEMP1         LENGTH OF C-LIST
                          57    STACKO    EQU     P.TEMP2         PTR TO STACK ORIG (REL B1)
                          60    SUBPT     EQU     P.TEMP3         PTR TO SUBPROCESS TABLE (REL B1)
                          61    DESCSIZ   EQU     P.TEMP4       . TOTAL SIZE OF PROCESS DESC IN CM
                          62    CORELEN   EQU     P.TEMP5         LENGTH OF PROCESS IN CORE
                          63    ECSAD     EQU     P.TEMP6
                          64    COMPSPA   EQU     P.TEMP7       . SPACE LEFT IN COMPILED MAP
                                *
                                *
  0                                     ECSCODE CPROC
                                *
                                * GET LENGTH OF C-LIST
                                *
  1  51  5111000102             MAKPROC   SA1     B1+P.PARAM+MP.CLIST+1
                 73010                  SX0     X1
                     56010              SA0     B1
  2  52  0110000001                     RECS    1               READ MOT ENTRY
  3  53  54200                          SA2     A0
                 43547                  MX5     39
```

```
                    1311 2                  BX1     X1-X2           CHECK UNIQUE NAME
                         11151              BX1     X5*X1
L     54  0311000275                        NZ      X1,MKPROC44     ERROR..C-LIST GONE
                    1600 5                  BX0     -X5*X2          ECS ADDR OF C-LIST
L     55  0110000001                        RECS    1
L     56  54100                             SA1     A0              LENGTH OF C-LIST
              10611                         BX6     X1
                   5161000056               SA6     B1+CLEN         SAVE LENGTH OF C-LIST
                                *
                                * BEGIN CHECKING PARAMETERS AND ACCUMULATING
                                * CORE AND ECS LENGTH OF THE PARCESS
                                *
L     57  20101                             LX1     1               SPACE FOR LOCAL C-LIST BUFFER
                                *                                     ADD SCRATCH LENGTH, DEAD CELL BEFORE
                                *                                     ACTUAL PARAM AREA, ACTUAL PARAM BUFFER
                                *                                     FIXED LENGTH PROCESS DESCRIPTOR, AND
                                *                                     FULL C-LIST BUFFER
                   7261000174               SX6     X1+P.ROHEAD-P.SCR+P.PROCRO+P.PROCRW+3
                                *
                                * X6 = CM LENGTH OF NEW PROCESS, BEING ACCUMULATED
                                *
L     60  5161000067                        SA6     B1+STACKO       SAVE POINTER TO STACK
                                *
                                * CHAINING WORDS
                                *
                    5111000071              SA1     B1+P.PARAM+MP.CHWD
L     61  7221777776                        SX2     X1-1
                         03300002 3         NG      X2,MKPROC3      ERROR..LESS THAN 1 CHAINING WORD
L     62  7221777721                        SX2     X1-P.SCRL-5-1
                         03200000267        PL      X2,MK30X        ERROR - TOO MANY CHAINING WORDS
                                *
                                *                                     SPACE FOR CHAINING WORDS, ZERO WORD
                                *                                     AFTER CHAINING WORDS, CONSTANT LENGTH
                                *                                     PROCESS DESCRIPTOR
L     63  6221000044                        SB2     X1+1+P.PROCRO+P.PROCRW
                                *
                                * B2 = ECS LENGTH OF NEW PROCESS, BEING ACCUMULATED
                                *
                                *
                                * STACK LENGTH
                                *
                    5131000072              SA3     B1+P.PARAM+MP.STACK
L     64  7223777776                        SX2     X3-1
                         03300002 4         NG      X2,MKPROC3      ERROR..LESS THAN 1 STACK ENTRY
L     65  20301                             LX3     1
              63232                         SB2     B2+X3           . ECS LENGTH + LENGTH OF STACK
                         36643              IX6     X6+X3           . SIMILARLY FOR CM LENGTH
L     66  5161000060                        SA6     B1+SUBPT        SAVE SUPROCESS TABLE ORIGH
                                *
                                * LOGICALMAP SIZE+ SUBP DESC SIZE + DEAD WORDS AT END OF SUBP TABLE
                                *
                    5111000075              SA1     B1+P.PARAM+MP.MAP
L     67  10211                             BX2     X1
```

```
                20201                       LX2     1
                  36251                     IX2     X2+X1       * 3*(NUM OF LGICAL MAP ENTRIES)
L     70  7232000013                        SX3     X2+I+SD.SIZE+SD.CC+1 * SPACE FOR MAP WITH =0 AT THE E
                                                                  * AND SUBP TABLE WITH XTRA CLASS
                                                                  * AT THE END
                  0332000255                NG      X2,MKPROC35     ERROR..LESS THAN 2 MAP ENTRIES
L     71  63232                             SB2     B2+X3
                  36663                     IX6     X6+X3
                                *
                                * COMPILED MAP BUFFER
                                *
                  5121000076                SA2       B1+P.PARAM+MP.COMP    ADD COMPILED MAP BUFFER
L     72  0332000256                        NG        X2,MKPROC43     ERROR..NEG SPACE FOR COMPILED MAP
                  36642                     IX6       X6+X2
                      63222                 SB2       B2+X2
L     73  5161000041                        SA6     B1+DESCSIZ          * SAVE TOTAL CM SIZE OF PROC RES
                                *
                                * CHECK SUBP FL
                                *
                  5111000077                SA1     B1+P.PARAM+MP.FL
L     74  0331000257                        NG      X1,MKPROC35     ERROR..NEGATIVE FIELD LENGTH DECLARED
                  36641                     IX6     X6+X1
L     75  7120000000 X                      SX2     =XENDSYS
                  14222                     BX2     -X2
L     76  5130000000 X                      SA3     =XS.CMFL    * CM AVAILABLE
                  36232                     IX2     X3+X2       * LESS SYSTEM RESIDENT
                    37262                   IX2     X6-X2
L     77  0322000270                        PL      X2,MK3EX        ERROR. FL TOO LARGE
                  5161000062                SA6       B1+CORELEN      SAVE CORE SIZE
                                *
                                * ENTRY POINT
                                *
L     100 5121000100                        SA2     B1+P.PARAM+MP.ENTRY
                  7333777766                SX3     X2-9
L     101 0333000260                        NG      X3,MKPROC36     ERROR..ENTRY POINT LESS THAN 8
                  37221                     IX2     X2-X1
                      73220                 SX2     X2
L     102 0322000271                        PL      X2,MK36X        ERROR - ENTRY .GT. FL
                                *
                                * PUT TOGETHER THE LOGICAL MAP ENTRIES IN P.SCR2 (LATER WRITTEN TO ECS I
                                * ALL GOES WELL).
                                *
                                * MAKE DUMMY RUN ON MAP ENTRIES TO TEST FOR
                                * FILE  BLOCKS PRESENT
                                *                 NOTE....*REFER* DOES NOT DESTROY  B2 WHERE WE HAVE
                                *                 THE TOTAL ECS SPACE NEEDED FOR THE PROCESS
                                *
                                * READ/WRITE ENTRY
                                *
                  5111000104                SA1     B1+P.PARAM+MP.FILE1+1     MAKE UP LOGICAL MAP ENTRY
L     103 5121000105                        SA2     B1+P.PARAM+MP.FLAD1       FOR 1ST SWAPPING DIRECTIVE
                  0335000261                NG      X2,MKPROC36     ERROR..FILE ADDRESS NEGATIVE
      104 10611                             BX6     X1
```

```
          10722                             BX7      X2
               5161000050                   SA6      B1+P.SCR2
 105  5076000001                            SA7      A6+1
               5111000106                   SA1      B1+P.PARAM+MP.CMAD1
 106  73110                                 SX1      X1
      0331000262                            NG       X1,MKPROC39     ERROR. CORE ADDRESS NEGATIVE
                    20122                   LX1      18
 107  5121000107                            SA2      B1+P.PARAM+MP.WDCT1
               73220                        SX2      X2
 110  0332000263                            NG       X2,MKPROC40     ERROR. COUNT OF WORDS NEGATIVE
               12712                        BX7      X1+X2
 111  5077000001                            SA7      A7+1
               5141000077                   SA4      B1+P.PARAM+MP.FL     CALL *REFER* FOR DUMMY RUN
 112  7110000050                            SX1      P.SCR2          PTR TO LOGICAL MAP ENTRY ( REL B1 )
               13555                        BX5      X5-X5           BUMP REF COUNT = 0
 113  6130000002                            SB3      2               . COUNT OF COMPILED SPACE USED
               6140000276                   SB4      MKPROC42        RETURN IF BLOCK MISSING
 114  6160000116                            SB6      MKPROC99        NORMAL RETURN
               6170000272                   SB7      MKPROC41        RETURN IF WORD COUNT PLUS CM OR FILE A
                                   *                                 ADDR IS TOO LARGE
 115  0200000000 X                          JP       =XREFER
                                   *
                                   *
                                   * READ ONLY ENTRY
                                   *
 116  5111000111                   MKPROC99 SA1      B1+P.PARAM+MP.FILE2+1    MAKE LOGICAL MAP ENTRY
               5121000112                   SA2      B1+P.PARAM+MP.FLAD2   FOR 2ND SWAPPING DIRECTIVE
 117  0332000264                            NG       X2,MK39X        ERROR - FILE ADD NEGATIVE
               10611                        BX6      X1
                    10722                   BX7      X2
 120  5161000053                            SA6      B1+P.SCR2+3
               5076000001                   SA7      A6+1
 121  5111000113                            SA1      B1+P.PARAM+MP.CMAD2
               73110                        SX1      X1
 122  0331000265                            NG       X1,MK39X        ERROR - CORE ADDRESS NEGATIVE
               20122                        LX1      18
 123  5121000114                            SA2      B1+P.PARAM+MP.WDCT2
               73220                        SX2      X2
 124  0332000266                            NG       X2,MK40X        ERROR - WD COUNT NEGATIVE
               12712                        BX7      X1+X2
                    43101                   MX1      1               . SET READ/ONLY BIT
 125  12771                                 BX7      X7+X1
               5077000001                   SA7      A7+1
 126  5141000077                            SA4      B1+P.PARAM+MP.FL     CALL *REFER* FOR DUMMY RUN
               7110000053                   SX1      P.SCR2+3        PTR TO LOGICAL MAP ENTRY ( REL B1 )
 127  6140000276                            SB4      MKPROC42        RETURN IF MISSING BLOCK
               6160000131                   SB6      MKPROC98        RETURN IF SUCESSFUL
 130  6170000272                            SB7      MKPROC41        RETURN IF WORD COUNT PLUS CM ADDR
                                   *                                 OR FILADDR IS ILLEGAL
               0200000000 X                 JP       =XREFER
                                   *
 131  5111000076                   MKPROC98 SA1      B1+P.PARAM+MP.COMP    CHECK COMPILED SPACE USED
               63410                        SB4      X1
```

```
L     132  0743000277                          LT      B4,B3,MK43X       ERROR - NOT ENOUGH SPACE FOR MAP
                                    *
                                    *                   TEST CAPABILITY INDEX
                                    *
                       5111000070               SA1     B1+P.PARAM+MP.CAPI
L     133  0331000273                          NG        X1,MKPROC37           FROR..CAP INDEX NEGATIVE
                       5121000162               SA2     B1+P.CLIST
L     134  53221                               SA2     B1+X2
                 37112                         IX1     X1-X2
                       0331000137               NG      X1,MKPROC2        JP IF INDEX IN LOCAL C-LIST
L     135  5022000002              MKPROC1     SA2     A2+2
                       0302000274               ZR      X2,MK37X          ERROR - CAPAB INDEX TOO LARGE
L     136  37112                               IX1     X1-X2
                       0321000135               PL      X1,MKPROC1
```

```
                                *
                                * GET ECS SPACE, NOT, UNIQUE NAME
                                * NOTE...MAY FAIL DUE TO LACK OF FUNDS
                                *
L     137  515100006 7          MKPROC2   SA5     B1+P.PARAM+MP.ALLOC+1
                                * B2 = ECS SIZE OF PROC FOR MAKEOBJ
                   614000141              SB4     MKPROC3         RETURN LINK
L     140  7170000001           SX7       AT.PROC          TYPE OF OBJECT TO ALLOCATE
                   020000000 X            JP      =XMAKEOBJ        CALL ALLOCATOR
                                *
                                *                 STORE AWAY THE CAPABILITY
L     141  5111000070           MKPROC3   SA1     B1+P.PARAM+MP.CAPI      CAPABILITY INDEX
                    10655                 BX6     X5                SAVE ECS ADDRESS
L     142  5161000063                     SA6     B1+ECSAD
                   615000000 X            SB5     =XCAPAR     . WHERE MAKEOBJ LEFT THE CAP
L     143  6160000144              SB4     MKPROC4         RETURN LINK
                   020000000 X            JP      =XPUTCAP         CALL *PUTCAP*
                                *
                                * INCREMENT REFERENCE COUNTS FOR REAL ON SWAPPING
                                * DIRECTIVES   ( AT P.SCR2 AND P.SCR2+3 )
                                *
L     144  5141000077           MKPROC4   SA4     B1+P.PARAM+MP.FL   CALL *REFER* FOR BUMP COUNTS
                   711000050              SX1       P.SCR2
L     145  7150000001           SX5       1
                   613000002              SB3       2
L     146  6140000252           SB4       MKPROC20        DISASTER RETURN
                   616000150              SB6       MKPROC5
L     147  6170000252           SB7       MKPROC20        DISASTER RETURN
                   020000000 X            JP      =XREFER
L     150  5141000077           MKPROC5   SA4     B1+P.PARAM+MP.FL   CALL *REFER* FOR 2ND  SWAPPING DIR
                   711000053              SX1       P.SCR2+3
L     151  6160000153           SB6       MKPROC6
                   617000252              SB7       MKPROC20
L     152  020000000 X           JP        =XREFER
L     153  5111000076           MKPROC6   SA1     B1+P.PARAM+MP.COMP
                   766430         SX6     B2                * COMPUTE AND SAVE SPACE LEFT IN
                        37616     IX6     X1-X6             *  COMPILED MAP BUFFER
L     154  5161000064             SA6     B1+COMPSPA
                                *
                                *
                                *                 BEGIN TO CONSTRUCT THE PROCESS
                                *
                                *                 PROCESS FIXED LENGTH DESCRIPTOR
                                *
                                * P.ROHEAD        VFD   12/4010B,12/NUM CH WD+1,18/MOT INDEX,
                                *                       18/PROC LEN IN CORE
                                *
                   712000040 10           SX2     PE.CREA     . FLAGS FOR STATE OF NEW PROCESS
L     155  20214                 LX2       12
                        5111000071        SA1     B1+P.PARAM+MP.CHWD   NUM CHAINING WORDS
L     156  7261000001           SX6       X1+1
                    12662         BX6     X6+X2
                         20622    LX6     18
```

```
L    157  5120000001 x                    SA2     CAPAB+1         GET MOT INDEX FROM CAPABILITY
                 73250                    SX2     X2
                     12662                BX6     X6+X2
L    160  20622                           LX6     18
                51210000042               SA2     B1+CORELEN      CORE LENGTH OF PROCESS
                     12662                BX6     X6+X2
L    161  56610                           SA6     B1
                                  *
                                  * P.ROHEAD+1      VFD   6/0,18/0,
                                  *                       18/FULL C-LIST TAB ORIG (REL P.ROHEAD),
                                  *                       18/LEN OF PROC VAR DESC = SD.SIZE+SD.CC
                                  *                             +3*(NUM LOG MAP ENTRIES)+1
                                  *                             +2*(NUM OF STACK ENTRIES)
                                  *                               +COMP MAP SPACE
                                  *
               722100000044               SX2     X1+P.PROCR0+P.PROCRW+1   COMPUT C-TABLE ORIG
                     20222                LX2     18
                                  *                       COMPUTE LENGTH OF PROC VARIABLE DESC
L    162  5131000057                      SA3     B1+STACK0       . STACK ORIGIN
                     5141000061           SA4     B1+DESCSIZ      . TOTAL CM SIZE OF DESC
L    163  37443                           IX4     X4-X3           . LENGTH OF VAR DESC
                     12624                BX6     X2+X4
                          5064000001      SA6     A6+1
                                  *
                                  * ZERO SCHEDULER DATA AND EXCHANGE PACKAGE
                                  *
L    164  136666                          BX6     X6-X6
                          6120000024      SB2     P.SCHEDL+14
L    165  5066000001           MKPROC7    SA6     A6+1
                               6125777776 SB2     B2-1
L    166  0502000165                      NE      B0,B2,MKPROC7
                                  *
                                  * SET EXIT MODE = 7
                                  *
                          43703           MX7     3
                               20763      LX7     36+3+12
L    167  5076777763                      SA7     A6-12
                                  *
                                  * SET MA = B1+P.XPACK
                                  *
                          54310           SA3     B1
                                          SA4     =XS.CMFL
L    170  5140000000 x                    SX3     X3
                     73340                IX3     X4-X3           . B1 FOR NEW PROCESS
                          37343           SX7     X3+P.XPACK
L    171  7273000142                      LX7     36
                               20744      SA7     A6-9            . STORE MA FOR NEW PROCESS
L    172  5076777766
                                  *
                                  *                 P.CLIST       IS SET BY ENVIORN
                                  *
                                  *
                                  * P.CTABLE         VFD   6/0,18/LEN(LOCAL C-LIST)/2,18/LEN(FULL C-TABLE)
                                  *                       18/PTR TO ORIG OF FULL C-TABLE (REL B1)
```

```
                                         *
              5141000056                     SA4    B1+CLEN       . RETRIEVE LEN(LOCAL C-LIST)
    173 20422                                LX4    18
              7160000001                     SX4    1             . C-TABLE SIZE = 1
                         12664               BX4    X6+X4
    174 20622                                LX6    18
              5131000057                     SA3    B1+STACKO
    175 7273777776                           SX7    X3-1              STACK ORIGIN - 1
                    12667                    BX6    X6+X7
    176 5066000002                           SA6    A6+2
                                         *
                                         * P.STACK                VFD    6/0,18/LAST ENTRY PTR,18/FIRST ENTRY PTR,
                                         *                                    18/TOP OF STACK POINTER ALL(REL B1)
                                         *
              5141000060                     SA4    B1+SUBPT      .   ORIG OF SUBPROCESS TABLE  .
    177 5121000057                           SA2    B1+STACKO     . ORIG OF STACK
              7264777775                     SX6    X4-2          . END OF STACK
    200 20622                                LX6    18
              12662                          BX6    X6+X2         . TOP OF STACK = ORIGIN OF STACK
                    20622                    LX6    18
                         12662               BX6    X6+X2
    201 5066000001                           SA6    A6+1
                                         *
                                         * P.SUBPDT               VFD     6/0,18/PTR TO NEXT SUBP TO CREATE
                                         *                                  18/PTR TO ORIG OF SUBP TABLE BOTH REL B1
                                         *                                  18/NUMBER OF SUBPS
                                         *
              5121000061                     SA2    B1+DESCSIZ
    202 7262777775                           SX6    X2-1-SD.CC
                    20622                    LX6    18
                         12664               BX6    X6+X4
    203 20622                                LX6    18
              43773                          MX7    59
                    37667                    IX6    X6-X7
    204 5066000001                           SA6    A6+1
                                         *
                                         *
                                         * WRITE OUT PROCESS READ ONLY AND READ WRITE DESCRIPTOR
                                         *
              5111000063                     SA1    B1+ECSAD          RECOVER ECS ADDRESS
    205 10011                                BX0    X1
              56010                          SA0    B1
    206 0120000040                           WECS    P.MAPSIN-P.ROHEAD
                                         *
                                         *                  LEAVE ALL TEMPORIES AND CHAINING WORDS ZERO
                                         *                  BY INCREMENTING ECS ADDR  (ALLOCATOR ZEROS ECS )
                                         *                   (SKIP DOWN TO VAR LEN DESC IN ECS)
                                         *
    207 5121000071                           SA2    B1+P.PARAM+MP.CHWD   . NUMB OF CHAINING WORDS
                    7222000044               SX2    X2+1+P.PROCRO+P.PROCRW          . +0 WORD +FIXED LEN D
    210 36002                                IX0    X0+X2
                                         *
                                         * STACK WITH ONE ENTRY
```

```
                                *
                43601                   MX6     1           . SET INTERRUPT INHIBIT FOR ROOT
                    10746               BX7     X6
                        20704           LX7     SF.PCQ-SF.II
L     211  12767                        BX7     X6+X7       . SET ABOUT TO EXECUTE IN STACK
                    20770               LX7     60-SF.PCQ
                        5111000100      SA1     B1+P.PARAM+MP.ENTRY
L     212  7211777774                   SX1     X1-P.ITLOFF
                    12671               BX6     X7+X1       . PUT P-COUNTER IN STACK
                        56610           SA6     A1          . SET FIRST WORD OF STACK
L     213  5111000060                   SA1     B1+SUBPT        ORIG OF SUBP TABLE
                    10671               BX6     X1
                        20622           LX6     18
L     214  12661                        BX6     X6+X1           TOP OF PATH = CURRENT SUBPROCESS
                    5066000001          SA6     A6+1        . SET SECOND WORD OF STACK
                                *
                                * WRITE OUT STACK ENTRY
                                *
L     215  0120000002                   WECS    2
L     216  5151000072                   SA5     B1+P.PARAM+MP.STACK   INCREMENT ECS ADDR TO
                    20501               LX5     1               SUBPROCESS TABLE (SKIP OVER REST
                        73550           SX5     X5                OF STACK)
L     217  36005                        IX0     X0+X5
                                *
                                *               SUBPROCESS TABLE ENTRY
                                *
                                * WORD SD.FIPT      VFD       6/FLAGS = 0
                                *                   18/INTERRUPT DATUM = 0
                                *                   18/C-LIST LENGTH
                                *                   18/NEXT PTR = SD.SIZE+COMP MAP+3*LOG MAP+1
                                *
                    5121000076          SA2     B1+CLEN     . RETRIEVE LENGTH OF CLIST
                        10622           BX6     X2
I     220  20622                        LX6     18
                    5121000075          SA2     B1+P.PARAM+MP.MAP
                        73120           SX1     X2
L     221  20101                        LX1     1
                    36221               IX2     X2+X1
                        5111000076      SA1     B1+P.PARAM+MP.COMP
I     222  36221                        IX2     X2+X1
                    7222000011          SX2     X2+SD.SIZE+1
                        12662           BX6     X6+X2
L     223  5161000000                   SA6     B1+SD.FIPT
                                *
                                * WORD SD.CC       NAME OF SUBPROCESS (CLASS CODE)
                                *
                    5111000074          SA1     B1+P.PARAM+MP.CLSCD+1
I     224  10611                        BX6     X1
                    5161000001          SA6     B1+SD.CC
                                *
                                * WORD SD.ESM       ERROR SELECTION MASK
                                *
                        436001          MX6     1           . SET MASK FOR ERROR ERROR ONLY
```

```
L    225  20662                        LX6     60-E.NOERR
              5161000002               SA6     B1+SD.ESM
                                  *
                                  * WORD SD.CLST     UNIQUE NAME, MOT OF LOCAL C-LIST
                                  *
L    226  5111000102                  SA1     B1+P.PARAM+MP.CLIST+1
                  10611               BX6     X1
L    227  5161000003                  SA6     B1+SD.CLST
                                  *
                                  * WORD SD.PTRS     VFD       6/0
                                  *                           18/MAX STACK PTR (REL B1)
                                  *                           18/FATHER PTR (0 FOR ROOT)
                                  *                           18/MAPIN LIST PTR (REL B1) (0 FOR NOW)
                                  *
                   5111000060     SA1     B1+SUBPT  . START OF SUBP TABLE IS ALMOST END OF STA
L    230  7261777773                SX6     X1-4      . NEXT TO LAST STACK ENTRY IS THE LIMIT
                    20644          LX6     36
L    231  5161000004                SA6     B1+SD.PTRS
                                  *
                                  * WORD SD.ORIG     VFD       6/0
                                  *                           18/ENTRY POINT
                                  *                           18/MAP ORIG (0 FOR ROOT)
                                  *                           18/ C-LIST ORIG ( 0 FOR ROOT)
                                  *
                   5111000100     SA1     B1+P.PARAM+MP.ENTRY
L    232  10611                    BX6     X1
               20644               LX6     36
                   5161000005     SA6     B1+SD.ORIG
                                  *
                                  * WORD SD.MAP      VFD       1/MAP ON-OFF FLAG (0=ON)
                                  *                           5/0
                                  *                           18/COMPILER MAP SIZE
                                  *                           18/NUM OF LOGICAL MAP ENTRIES
                                  *                           18/PTR TO LOG MAP, REL THIS WORD
                                  *
L    233  5111000076                SA1     B1+P.PARAM+MP.COMP
                  10611               BX6     X1
                    20622             LX6     18
L    234  5121000075                SA2     B1+P.PARAM+MP.MAP
                  12642               BX6     X6+X2
                    20622             LX6     18
L    235  7211000002                SX1     X1+2
                  12641               BX6     X6+X1
L    236  5161000006                SA6     B1+SD.MAP
                                  *
                                  * WORD SD.RAFL     VFD       6/0
                                  *                           18/RA+FL (REL B1)
                                  *                           18/RA (REL B1)
                                  *                           18/SPACE LEFT IN COMPILED MAP
                                  *
                   5111000061     SA1     B1+DESCSIZ                . =RA
L    237  5121000077                SA2     B1+P.PARAM+MP.FL
                    36612          IX6     X1+X2     . RA+FL
```

```
                      20622              LX6     18
    240  12661                           BX6     X6+X1     . RA
              20622                      LX6     18
                   5111000064            SA1     B1+COMPSPA          . RETRIEVE SPACE LEFT
    241  12661                           BX6     X6+X1
              5161000007                 SA6     B1+SD.RAFL
                                  *                        WRITE OUT SUBP TABLE
    242  0120000010                      WECS    SD.SIZE
                                  *
                                  * THE COUNTS ON THE COMPILED MAP ARE 0 TO FORCE COMPILATION:
                                  * SKIP OVER THE COMPILE MAP TO WRITE THE LOGICAL MAP
                                  *
    243  5111000076                      SA1     B1+P.PARAM+MP.COMP
                   7211000010            SX1     X1+SD.SIZE          SKIP COMP MAP AND SUBP DESC
    244  36001                           IX0     X0+X1
                                  *
                                  *
                                  * WRITE OUT SWAPPING DIRECTIVES
                                  *
              5101000050                 SA0     B1+P.SCR2 . ADDR WHERE MAPS WERE SET UP
    245  0120000006                      WECS    MP.SIZE+MP.SIZE
                                  *                        INCREMENT ECS ADDR TO END OF LOG MAP
    246  5111000075                      SA1     B1+P.PARAM+MP.MAP
                   10211                 BX2     X1
                      20201              LX2     1
    247  36221                           IX2     X2+X1
              36002                      IX0     X0+X2
                                  *
                   43674                 MX6     60                  -0 FOR END OF MAP
                      54600              SA6     A0
    250  0120000001                      WECS    1
                                  *
    251  0200000000 X                    JP      =XSYSRET




                                  *
                                  *                DISASTER/MISC ERRORS
                                  *
    252  0100000000 X             MKPROC20 RJ    =XDISASTR
    253  6140000003               MKPROC30 ERROR 3,NEGPAR
    254  6140000004               MKPROC31 ERROR 4,NEGPAR
    255  6140000006               MKPROC33 ERROR 6,NEGPAR
    256  6140000007               MKPROC43 ERROR 7,NEGPAR
    257  6140000010               MKPROC35 ERROR 8,NEGPAR
    260  6140000011               MKPROC36 ERROR 9,NEGPAR
    261  6140000014               MKPROC38 ERROR 12,NEGPAR
    262  6140000015               MKPROC39 ERROR 13,NEGPAR
    263  6140000016               MKPROC40 ERROR 14,NEGPAR
    264  6140000020               MK38X    ERROR 16,NEGPAR
    265  6140000021               MK39X    ERROR 17,NEGPAR
```

```
    266  6140000022          MK40X     ERROR    18,NEGPAR
    267  6140000003          MK30X     ERROR    3,BIGPAR
    270  6140000010          MK35X     ERROR    8,BIGPAR
    271  6140000011          MK36X     ERROR    9,BIGPAR
    272  6140000000          MKPROC41  ERROR    0,BIGPT
    273  6140000002          MKPROC37  ERROR    2,NEGIX
    274  6140000002          MK37X     ERROR    2,BIGIX
    275  6140000012          MKPROC44  ERROR    10,CLMOI
    276  6140000016          MKPROC42  ERROR    14,BLMISS
    277  6140000000          MK43X     ERROR    0,NOROOM
    300                                ENDECS   CPROC
```

```
*  MACROS
*
*          CHKMOT    WHERE,ERROR          ;  CHECKS MOT ENTRY.  FINDS
*                                             UNIQUE NAME/MOT AT ↑WHERE↑ AND EXITS
*                                             TO ↑ERROR↑ IF UNIQUE NAMES DONT MATCH
*
*          LEAVES
*                    X0:=38 BIT MASK
*                    X1:=CONTENTS OF ↑WHERE↑
*                    X2:=PTR PART OF MOT ENTRY
*                    A0:=B1
*
CHKMOT     MACRO     WHERE,ERROR
           SA1       B1+WHERE
           SX0       X1
           SA0       B1
           RECS      1
           SA2       A0
           MX0       39
           BX3       X1-X2
           BX3       X3*X0
           NZ        X3,ERROR
           BX2       -X0*X2
           ENDM
*
*  THIS DUMMY MACRO MARKS WHERE AB STUFF SHOULD BE LOCKED IN THE
*  TWO-CP CASE
*
RESERVE    MACRO
RESERVE    ENDM
*
*  THIS ONE MARKS THE UNLOCK SPOTS
*
RELEASE    MACRO
RELEASE    ENDM
*
*          SET       WHERE,REG            ;  SETS LOCKS (LIKE I.LOCK)
*                                             TO CONTENTS OF REG
*
SETLOCK    MACRO     WHERE,REG
           SX6       REG
           SA6       WHERE
           ENDM
*
*          FAMILY    ABLK,OBJ,ERROR       ;  CHECKS IF THE MOT PTR FOR
*                                             ABLK IS THE SAME AS THE ONE IN
*                                             THE ALLOCATOR HEADER WORD FOR OBJ.
*          ABLK:=PTR REL. B1 FOR WD. 1 OF CAPABILITY FOR ALLOCATION
*                                             BLOCK
*          OBJ:=PTR REL B1 FOR WD 1 OF CAPABILITY FOR OBJ
*          ERROR:=WHERE TO JUMP IN CASE OBJ DOESN↑T BELONG TO ABLK
*
*          INPUT
```

```
*           A0:=ABS CM ADR OF 1 WD BUFFER
*
*           OUTPUT
*           X0:= TEST RESULT
*           X1:= ALLOCATOR HEADER WORD FOR OBJ. MOT PTR RIGHT JUSTIFIED
*                                (BY A LEFT CIRCULAR SHIFT)
*           X2:=WD 1 OF AB CAPABILITY
*
FAMILY      MACRO     ABLK,OBJ,ERROR
            SA1       B1+OBJ          .. FETCH OBJ CAPABILITY
            SX0       X1               ISOLATE MOT PTR
            RECS      1
            SA1       A0              .. FETCH MOT FOR *OBJ*
            MX0       39
            BX0       -X0*X1
            SX1       IP.AWDS-1
            IX0       X0-X1           .. ADDR OF ALLOCATOR FIELD
            SA2       B1+ABLK
            RECS      1               .. READ ALLOCATOR HEADER WORD
            SA1       A0
            LX1       6+18            .. RIGHT JUSTIFY AB POINTER
            BX0       X1-X2
            SX0       X0              .. ISOLATE RESULT
            NZ        X0,ERROR
            ENDM
```

```
                                *
                                * CPUIN
                                *       MOVE  CPU TIME INTO A PROCESS. CLEAR PF.S IF TIMER GOES POSITIVE.
                                *       CHECK IF THE PROCESS RECEIVING THE TIME SHOULD BE
                                *       RESCHEDULED.
                                *       USER X6 IS SET TO THE NEW VALUE OF P.TIMER
                                *
                                *       ACTUAL PARAMETER AREA: MNEUMONICS
                                *
                                        ORG     P.PARAM
66                              APT.AB  BSS     2       .. CI  AB
66                              APT.P   BSS     2       .. CI  PROCESS
70                              APT.D   BSS     1       .. DI  AMOUNT OF TIME TO MOVE
72                                      USE     *
                                *
                                *  REGISTER ALLOCATION
                                *
                                *       A0:=B1
                                *       B3:=1:=CONSTANT
                                *       X5:=MOT ENTRY OF IP2 (PROCESS).
                                *       X7:=IP3:=TIME MOVED / X7:=P.TIMER*IP3
                                *
                                *       TEMPORARIES USED ! P.TEMP1
                                *
                                        ENTRY   CPUIN
227                                     ECSCODE CPUIN
1   51  5151000072             CPUIN   SA5     B1+APT.D        .. FETCH TIME TO MOVE FOR ERROR CHECKING
                    0335000117          NG      X5,CPUIN7       .. JP IF ERROR (TIME<0)
1   52  10755                         BX7     X5              .. X7:=IP3:=AMOUNT OF TIME TO MOVE
                                *
                                * TEST ALLOCATION BLOCK : JUMP TO CPUIN9 IF AN ERROR
                                *
                                        CHKMOT  APT.AB+1,CPUIN9
                        5111000047      BX0     X2              .. CHKMOT LEAVES X2:=PTR IN AB*S MOT
                            10022       SB3     1               . NICE CONSTANT
1   57  6130000001                      SA0     B1+B3
                            5401        RESERVE ALLOCBLK
1   60  0110000013                      RECS    AB.SIZE         . READ AB
1   61  5010000010                      SA1     A0+AB.CPAVL     . TIME AVAILABLE
                        5020000011      SA2     A0+AB.CPUSD             . TIME CONSUMED
1   62  37412                           IX4     X1-X2
                        37447           IX4     X4-X7           . TIME NOT COMMITTED LES PROPOSED DONATION
                            0334000122  NG      X4,CPUIN11              . SORRY, NOT THAT MUCH TO GIVE
1   63  366627                          IX6     X2+X7           . NEW AR.CPUSD FIELD
                        54620           SA6     A2
                            10440       BX4     X0              . PRESERVE ECS A(AR)
                                *
                                * DETERMINE IF PROCESS OK.  JP TO CPUIN10 IF NOT
                                *
1   64  5111000071                      CHKMOT  APT.P+1,CPUIN10
                                10522   BX5     X2              .. X5:=PTR IN MOT ENTRY OF IP2 LEFT BY CHK
1   70  5111000071                      FAMILY  (APT.AB+1),(APT.P+1),CPUIN12    .. SEE IF IP2 IS A
                                *                       DESCENDENT OF IP1
```

```
                                            *
                                            * SET I.LOCK AND BEGIN TO MOVE TIME
                                            *
                       76630                         SETLOCK   I.LOCK.B3
                       7110000010                    SX1       P.TIMER-P.ROHEAD    .. P.TIMER OFFSET
 L    101   36051                                    IX5       X5+X1      .. X5=ECS ADDR OF PROCESS
 L    102   0110000001                               RECS      1
 L    103   54300                                    SA3       A0
                  36737                              IX7       X3+X7      .. X3:=OLD P.TIMER  X7:=NEW P.TIMER
                       5070000000                    SA7       A0+0
 L    104   0120000001                               WECS      1
 L    105   5171000161                               SA7       B1+P.XPACK+15         . SEND RESPONSE TO USER
                                            *
                                            * UPDATE ALLOCATION BLOCK. PERFORM RESCHEDULING LOGIC IF APPROPRIATE
                                            *
                       54013                         SA0       B1+B3      . CM A(AB)
                           10044                     BX0       X4
 L    106   0120000013                               WECS      AB.SIZE    .. UPDATE AB.CPUSD FIELD IN APT.AB
 L    107                                            RELEASE   ALLOCBLK
 L    107   0333000112                               NG        X3.CPUIN6 .. JP IF OLD P.TIMER .LE. 0
                       0303000112                    ZR        X3.CPUIN6
                                            *
                                            * CLEAR I.LOCK AND RETURN
                                            *
 L    110   76600                           CPUIN5   SETLOCK   I.LOCK.B0 .. I.LOCK:=B0
 L    111   0200000000 X                             JP        SYSRET     .. RETURN
                                            *
                                            * P.TIMER WAS NEG: IF NOW GT 0. UNSET TIMER OUT FLAG AND
                                            * SEE ABOUT RESCHEDULING THIS PROCESS
                                            *
 L    112   0337000110                      CPUIN6   NG        X7.CPUIN5 .. DON'T BOTHER RESCHEDULING IF NEW P.TIME
                       0307000110                    ZR        X7.CPUIN5    .LE. 0
 L    113   10055                                    BX0       X5         . ADDRESS OF P.ROHEAD
                  43301                              MX3       1
                       20344                         LX3       60-PF.S    .. POSITION MASK TO PF.S
 L    114   0110000001                               RECS      1          .. FETCH P.ROHEAD FROM ECS
 L    115   54100                                    SA1       A0
                  15613                              BX6       -X3*X1     .. CLEAR PF.S
                                            *
                                            *        SEE FI THE PROCESS GETS RESCHERULES
                                            *
                       6170000110                    SB7       CPUIN5     . RETLINK
 L    116   0400000000 +                             EQ        CHKSCHD
                                            *
                                            * ERRORS
                                            *
 L    117   6140000001                      CPUIN7   ERROR     1.NEGPAR   .. NEGATIVE TIME
 L    120   6140000001                      CPUIN9   ERROR     1.MISSOB   .. AB GONE
 L    121   6140000002                      CPUIN10  ERROR     2.MISSOB   .. PROCESS GONE
 L    122   6140000003                      CPUIN11  ERROR     3.NOCP     .. AB CAN'T MOVE THAT MUCH TIME
 L    123   6140000002                      CPUIN12  ERROR     2.FATSON   .. IP2 NOT DESCENDED FROM IP1
                                            *
                                            *
```

```
L    124                    ENDECS    CPUIN
```

```
                                *
                                * CPOUT
                                *       MOVE CPU TIME FROM A PROCESS TO AN AB
                                *
                                *       CHECK TO SEE IF P.TIMER GOES NEGATIVE.  IF SO, ONLY MOVE
                                *       ENOUGH TIME TO MAKE P.TIMER EQUAL TO ZERO
                                *       USER X6 IS SET TO NEW VALUE OF P.TIMER IN IP2
                                *
                                *       ACTUAL PARAMETER AREA (REL B1+P.PARAM)
                                *
                                *       APT.AB          C: AB (WD 0)
                                *       +1                 (WD 1)
                                *       APT.P           C: PROCESS (WD 0)
                                *       +1
                                *       APT.D           D: DATUM : AMOUNT OF TIME TO MOVE
                                *
                                *       REGISTER ALLOCATION
                                *
                                *       X4:= IP1  := MOT OF AB
                                *       X5:= IP2  := MOT OF PROCESS
                                *       X7:= IP3  :=AMT OF TIME TO MOVE/ X7:=NEW VALUE OF P.TIMER
                                *       B2:= 1    := CONSTANT
                                *
      302
      51  5141000072           CPOUT   ENTRY   CPOUT
      52  0000000000           PATCHI  ECSCODE CPOUT
      53  0334000120                   SA4     B1+APT.D
                  6121000001            PS
      54  10744                        NG      X4,CPOUT7   .. ERROR: TIME<0
                                       SB2     1           .. REGISTER ALLOCATION
                                       BX7     X4          .. X7:= IP3 := AMT TO MOVE, REGISTER ALLOC
                                *
                                *       CHECK MOT OF THE AB.  BRANCH TO CPOUT8 IF THERE IS AN ERROR
                                *
                                *
                                *
                                *
                5111000047              CHKMOT  APT.AB+1,CPOUT8
                                *
                                *       X1:= WD 1 OF AB CAPABILITY
                                *       X2:= MOT OF AB
                                *       A0:=B1
                                *       X0:= 30 BIT MASK
                                *
                          10422         BX4     X2          .. REGISTER ALLOCATION. X4:=MOT ENTRY OF A
                                *
      61  5111000071                   CHKMOT  P.PARAM+3,CPUING    .. CHECK MOT OF PROCESS.  JP TO
                                *                                    IF THERE IS AN ERROR
                                *
                          10522         BX5     X2          .. REGISTER ALLOCATION  ( X5 :=PTR IN MOT
                                                                ENTRY FOR PROCESS
      65  5111000071                   FAMILY  (APT.AB+1),(APT.P+1),CPOUT10  ..DOES THE PROCESS
                                *                                    BELONG TO THE AB?  JP CPOUT10 IF NOT.
                                *
                                *       FIND OUT IF IP2 = CURRENT PROCESS IN THIS CPU
```

```
                                *
                                *       IDPROC  APT.P+1     .. ADDR OF WD 1 OF A CAPABILITY FOR A PROC
                                *
                    744030      *       SETLOCK T.LOCK.B2   .. SET T.LOCK(I=B2:=1)
                                *
                    0301000106  *       ZR      X1,CPOUT4           JP IF SAME PROCESS
                                *
                                * IF PROCESS DOESNT REALLY HAVE X4 AMOUNT OF TIME TO RELEASE, X4 WILL BE
                                * TO CURRENT VALUE OF P.TIMER=ALL THE TIME IN THE PROCESS
                                *
  76  7110000010                *       SX1     P.TIMER-P.ROHEAD    .. OFFSET FROM P.ROHEAD
                    340561      *       IX0     XS+X1       .. ABS ECS ADDR. OF P.TIMER
  77  0110000001                *       RECS    1
 100  54100                     *       SA1     A0
                    376177      *       IX6     X1-X7       .. X6:=NEW P.TIMER
                    0334000104  *       NG      X6,CPOUT3   .. WOOPSS DIDNT HAVE THAT MUCH TIME AFTER
 101  54600                     *       SA6     A0
 102  0120000001    CPOUT2      WECS    1                   .. RESET ECS VERSION OF P.TIMER
 103  0400000112                *       JP      CPOUT6      .. GO FIX UP AB
                                *
                                * P.TIMER WENT NEGATIVE.  RESET X4 TO CURRENT VALUE OF P.TIMER
                                *
 104  76600         CPOUT3      SX6     B0                  .. P.TIMER MUST = ZERO
                    10711               BX7     X1                  ... AB RECEIVED AMT. OF TIME IN OLD P.TIME
                          5066000000    SA6     A0+0
 105  0200000102                *       JP      CPOUT2
                                *
                                * IP2=CURRENT PROCESS.  MUST RESET P.TIMER IN CORE
                                *
 106  5111000136    CPOUT4      SA1     B1+P.TIMER
                    376177              IX6     X1-X7               .. X6:=NEW VALUE OF P.TIMER
 107  0336000104                *       NG      X6,CPOUT3   .. WOOPSS TOO MUCH
                    546170      *       SA6     A1                  .. CORE VERSION OF P.TIMER
                          54600 *       SA6     A0                  .. PREPARE TO MODIFY ECS VERSION
 110  100055                    *       BX0     X5                  .. X5:=NOT ENTRY OF PROCESS
                    7110000010  *       SX1     P.TIMER-P.ROHEAD
                          36001 *       IX0     X0+X1               .. ECS ADDRESS OF P.TIMER
 111  0200000102                *       JP      CPOUT2      .. GO RESET ECS VERSION OF P.TIMER
                                *
                                * RESET THE AB
                                *
 112  7110000010    CPOUT6      SX1     AB.CPAVL
                    340174              IX0     X1+X4               .. ECS ADD OF TIME AVAILABLE
 113  0110000001                *       RECS    1
 114  54100                     *       SA1     A0
                    367174      *       IX7     X1+X4               .. INCREMENT CPU TIME AVAILABLE
                    547600      *       SA7     A0
 115  0120000001                *       WECS    1
 116  76600                     *       SETLOCK     I.LOCK,B0 .. CLEAR I.LOCK (=B0)
 117  5160100162                *       SA4     B1+P.XPACK+16       .. USER X6 := NEW VALUE OF P.TIM
                                                                   ON IP2
                    0200000000 X        JP      SYSRET
 120  6140000003    CPOUT7      ERROR   3,NEGPAR           .. TIME TO MOVE < 0
```

```
    121  6140000001              CPOUT8    ERROR    1,MISSOB  .. AB GONE
    122  6140000002              CPOUT9    ERROR    2,MISSOB  .. PROCESS GONE
    123  6140000002              CPOUT10   ERROR    2,FATSON  .. PROCESS NOT SON OF AB
    124                                    ENDECS   CPOUT
                                *
```

```
                                *
                                * CLRMG
                                *       CLEAR MESSAGE MECHANISM IN A PROCESS
                                *
                                *       ↑IF↑ MISSOR(IP1) ↑THEN↑ ERROR RETURN ↑FI↑
                                *       SETLOCK(I.LOCK)
                                *       ZEROMSGMECH(IP1)
                                *       CLEARLOCK(I.LOCK)
                                *       RETURN
                                *       ↑END↑
                                *
                                        ENTRY   CLRMG
     355                                ECSCODE CLRMG
                                *
                                *
  1   51  5111000047            CLRMG   CHKMOT  P.PARAM*1,CLRMG3    .. CHECK MOT OF PROCESS. JUMP
                                *                                    TO CLRMG3 IF THERE IS AN ERROR
                      13666             BX6     X6=X6               .. PREPARE TO CLEAR MESSAGE MECHANISM
                                *       IDPROC  P.PARAM*1           .. WD 1 OF A CAPABILITY FOR IP1 (PROCESS)
                                *                                   CHECK IF IP1 = CURRENT PROCESS IN THIS CPU
                                *                                   X0:=X2
                                *                                   X1:= RESULT OF COMPARISON (0==>IDENDICAL PROCESS)
                                *                                   X2:= P.ROHEAD OF ECS VERSION OF PROCESS
                                *
  1   55  7160000001                    SETLOCK I.LOCK,1
  1   56  5161000135                    SA6     B1+P.MSGDAT
                                *
  1   57  54600                 CLRMG2  SA6     A0              ZERO THE ECS VERSION
              5060000001                SA6     A0+1            THIS CODE ASSUMES THAT P.EVCDAT DIRECTLY
                                *                               FOLLOWS P.FVCDAT IN ECS.
  1   60  7110000006                    SX1     P.MSGFVC-P.ROHEAD
                    346001              IX0     X0+X1           .. READ ECS ADDRESS
  1   61  0120000002                    WECS    2
  1   62  5160000000 X                  SA6     I.LOCK          CLEAR I.LOCK
  1   63  0000000000            PATCH2  PS
  1   64  0200000000 Y                  JP      SYSRET
  1   65  6140000001            CLRMG3  ERROR   I.MISSOB
  1   66                                ENDECS  CLRMG
```

```
                                *
                                * STMSG
                                *       SET THE MESSAGE MECHANISM IN A PROCESS.
                                *       THE MESSAGE MECHANISM IS THE SECOND WORD OF A CAPABILITY FOR A
                                *       CHANNEL AND A 60 BIT DATUM
                                *
                                *       ↑IF↑ MISSOB(IP2)  ↑THEN↑  ERROR RETURN  ↑FI↑
                                *       ↑IF↑  MISSOB(IP1)  ↑THEN↑ CLEARLOCK(I.LOCK), ERROR RETURN ↑FI↑
                                *       SETMSGMECH(IP1)
                                *       CLEARLOCK(I.LOCK)
                                *       RETURN
                                *
                                *       ACTUAL PARAMETER AREA RELATIVE BI+P.PARAM
                                *
                                *       +0        C:  PROCESS (WD 0)
                                *       +1                (WD 1)
                                *       +2        C:  EVENT CHANNEL (WD 0)
                                *       +3                          (WD 1)
                                *       +4        D:  DATUM
                                *
                                *       X6:= MOT OF IP2 (EVCH)
                                *       X5:= IP3:= DATUM
                                *
                                *

                                        ENTRY     STMSG
                                        ECSCODE   STMSG
372
L       51  5111000071                 STMSG     CHKMOT  P.PARAM+3,STMSG3    .. CHECK UNIQUE NAME OF EVENT
                                    *                   CHANNEL.  JP TO STMSG3 IF AN ERROR
                                                BX6     X2              .. SAVE MOT ENTRY OF EVCH
                                    *
L       55  5111000067                         CHKMOT  P.PARAM+1,STMSG4    .. CHECK UNIQUE NAME OF PROCESS.
                                    *                   JP TO STMSG4 IF AN ERROR
L       61  5151000072                         SA5     B1+P.PARAM+4        .. X5:=IP3:=DATUM

                                    *                   IP2=CURRENT PROCESS IN CPU)
                                          546000            SA6     A0              .. XMIT EVCH
                                              10655         BX6     X5
L       62  5060000061                        SA6     A0+1            .. XMIT DATUM
                                          7164000001        SETLOCK I.LOCK.1
                                    *
                                          7120000006        SX2     P.MSGEVC-P.ROHEAD
        64  36002                             IX0     X0+X2           ABS ECS ADDR
        65  0120000002                        WECS    2
        66  76600                             SETLOCK I.LOCK.B0 .. CLEAR I.LOCK
        67  0000000000              PATCH3    PS
        70  0200000000  X                     JP      SYSRET
        71  6140000001              STMSG3    ERROR   1,MISSOB    .. PROCESS GONE FROM MOT
        72  6140000002              STMSG4    ERROR   2,MISSOB    .. EVNT CHANNEL GONE FROM MOT
        73                                    ENDECS  STMSG
```

```
      414                       ECSCODE   DLPRO
L      51  0000000000           PS
L      52                       ENDECS    DLPRO
```

CLIIB--CLEAR INTERRUPT INHIBIT BIT ON TOP OF STACK (

```
                                *
                                * CLIIB
                                *       CLEAR INTERRUPT INHIBIT BIT ON TOP OF STACK (TOS)
                                *
                                *       PARAAMETERLESS ACTION
                                *
                                *       ZEROBIT(SF.II,TOS)
                                *       ↑GOTO↑ TOSPROC
                                *
                                *  NOTE: TOSPROC CHECKS IF THERE IS A PENDING INTERRUPT.
                                *       IF SO, THE INTERRUPT IMMEDIATELY STRIKES.  SEE TOSPROC FOR
                                *       FURTHER DETAILS
                                *
                                        ENTRY   CLIIB
                                        ECSCODE CLIIB
                                *
L     51  43001                  CLIIB   MX0     1               .. MASK
              5111006144                SA1     B1+P.STACK      .. FETCH PTR TO TOS
                    53211               SA2     X1+B1           .. FETCH WD 0, TOS
L     52  20074                         LX0     60-SF.II        .. POSITION MASK TO SF.II
              15620                     BX6     -X0*X2          .. CLEAR SF.II ON TOS
                    5066000000          SA6     A2+0            .. UPDATE TOS ENTRY
L     53  0200000000 X                  JP      TOSPROC         .. GO CHECK FOR PENDING INTERRUPTS
                                                                SEE NOTE ABOVE AND COMMENTS IN TOSPROC
                                *
                                *
L     54                        ENDECS  CLIIB
```

```
                                    *
                                    * STIIB
                                    *       SET INTERRUPT INHIBIT BIT ON TOP OF STACK
                                    *
                                    *       PARAMETERLESS ACTION
                                    *
                                            ENTRY     STIIB
     420                                    ECSCODE   STIIB
                                    *
L    51  43001                      STIIB   MX0       1
             20074                          LX0       60-SF.II  .. POSITION MASK TO SF.II
                                    *
                   5111000164               SA1       B1+P.STACK          .. FETCH PTR TO CURRENT TOP OF
                                    *                                   STACK
L    52  53211                              SA2       B1+X1     .. FETCH W0.0 OF TOS ENTRY
             12620                          BX6       X2+X0     .. SET SF.TT
                   54630                    SA6       A2        .. STORE ALTERED STACK ENTRY
L    53  0200000000 X                       JP        SYSRET
L    54                                     ENDECS    STIIB
```

```
    423                                 ECSCODE   DPROD
L    51  0000000000                     PS
L    52                                  ENDECS    DPROD
```

```
                                *
                                * ARMIT
                                *       ARM INTERRUPTS FOR A GIVEN SUBPROCESS
                                *
                                *       IF NOFIND(IPI) THEN ERROR RETURN
                                *       ELSE CLEARBIT(SD.FDIS,IPI) FI
                                *       RETURN
                                *
                                        ENTRY   ARMIT
     424                                ECSCODE ARMIT
                                *
L      51  617000005 3          ARMIT   SB7     ARMIT2          .. RETURN FOR FINDSUB
                 513000067              SA3     B1+P.PARAM+1            .. WD 1 OF CC (FOR FINDSUB)
L      52  0200000000 X                 JP      FINDSUB
                                *
                                *       DO ERROR RETURN IF NECESSARY  (NO SUCH SUBPROCESS)
                                *
L      53  64250                 ARMIT2  SB2     A5              .. ADDRESS OF DESCRIPTOR(IPI)+SD.CC
                     64360              SB3     A6              .. SET BY FINDSUB
                         0423000060     EQ      B2,B3,ARMIT3            .. JP IF NOSUCH SUBPROCESS
                                *
L      54  43001                       MX0     1
                     20072              LX0     60-SD.FDIS              .. POSITION MASK
                         5015777776     SA1     A5-SD.CC        .. FETCH SD.FIPT IN SUBPROCESS DESCRIPTOR
L      55  15610                       BX6     -X0*X1          .. TURN OFF SD.FDIS (==>TURN OFF INTERR
L      56  0000000000          PATCH4  PS                      ARE DISARMED BIT
                                *
L      57  0200000000 X               JP      SYSRET          .. RETURN
                                *
L      60  614000001           ARMIT3  ERROR   I.NOFIND        .. NO SUCH SUBPROCESS AS IPI
L      61                             ENDECS  ARMIT
```

```
                                      *
                                      * DISIT
                                      *          DISARM INTERRUPTS FOR A GIVEN SUBPROCESS
                                      *          ↑PROCEDURE↑ DISIT(CC);
                                      *          ↑BEGIN↑
                                      *          ↑IF↑ NOSUBP(CC)  ↑THEN↑ ERROR(NOFIND)  ↑FI↑
                                      *          ↑IF↑ PENDING INTERRUPT(CC)
                                      *          ↑THEN↑ ↑BEGIN↑
                                      *                    X:=PRINT(B1,RETLINK)
                                      *                    ↑IF X = ↑CC↑
                                      *                    ↑THEN↑ ↑BEGIN↑
                                      *                    PCQ(TOS):= ↑ABOUT TO EXECUTE↑
                                      *                    RETLINK:= SYSRET
                                      *                    JP TOSPROC
                                      *                    ↑END↑
                                      *                    ↑ELSE↑ F=RETURN  ↑FI↑
                                      *                    ↑END↑
                                      *          ↑ELSE↑ ↑BEGIN↑
                                      *                    SETBIT(SD.FDIS,↑CC↑)
                                      *                    RETURN
                                      *                    ↑END↑
                                      *          ↑FI↑
                                      *          ↑END↑
                                      *

                                                 ENTRY     DISIT
    434                                          ECSCODE   DISIT
                               66     CLASCDE    EQU       APT.A8

 L     51 6170000053                  DISIT      SB7       DISIT2     .. DETERMINE IF IP1 (=↑CC↑=↑CLASCDE↑)
                    5131000067                   SA3       B1+CLASCDE+1        REALLY EXISTS
 L     52 0200000000 X                           JP        FINDSUB
                                      *
                                      * FINDSUB LEAVES A5=ADDR. REL.B1 OF SUBPROCESS DESCRIPTOR (+SD.CC)
                                      *            A6=DUMMY VALUE. IF A5=A6, THERE IS AN ERROR
                                      *
 L     53 64250                       DISIT2     SB2       A5
                64360                            SB3       A6
                    0423000070                   EQ        B2,B3,DISIT5          .. ERROR NOSUCH SUBPROCESS
                                      *
                                      * SAVE ADDR OF SP. DESCRIPTOR IN B4.  CORRECT THE SD.CC OFFSET.
                                      * DETERMINE IF ↑CC↑ HAS A PENDING INTERRUPT.
                                      *
 L     54 6142777776                             SB4       B2-SD.CC
                    56414                        SA4       B1+B4
                         10044                   BX0       X4
 L     55 20074                                  LX0       60-SD.FINT
                0330000041                       NG        X0,DISIT3
                                      *
                                      *  SIGH.
                                      *          THAT WAS EASY.  SET SD.FDIS AND RETURN
```

```
                                       *
                  43001                          MX0       1
L     56  20072                                  LX0       60-SD.FDIS         .. POSITION MASK
               12604                             BX6       X0+X4        .. AND IT TO SD.FIPT (WD 0 OF SD DESCRIPTO
                    54640                        SA6       A4           .. UPDATE DESCRIPTOR
L     57  0000000000                   PATCH5    PS
L     60  0400000000 X                           JP        SYSRET
                                       *
                                       *  TOO BAD.
                                       *         AN INTERRUPT IS PENDING.  IF *CC* HAS PRIORITY. WE WILL LET
                                       *         IT RUN (AFTER SETTING THE PCQ IN CURRENT TOS).  OTHERWISE USER
                                       *         WILL GET AN F-RETURN
                                       *
L     61  6170000062                   DISIT3    SB7       DISIT4
                    0200000011 +                 JP        PRINT        .. SUBROUTINE RETURNS B2=0 IF NO PRIORITY
                                       *         INTERRUPT, B2=ADDR. OF SP. DESCRIPTOR IF THERE IS A PRIORITY
                                       *         INTERRUPT.
                                       *
L     62  0524000071                   DISIT4    NE        B2,B4,DISIT6        .. JP TO AN F-RETURN IF PRIORITY
                                       *                             INTERRUPT  .NE. *CC*
                                       *
                                       *  RESET PCQ TO *ABOUT TO EXECUTE* FOR CURRENT TOS
                                       *
                    5111000066                   SA1       B1+P.PARAM
L     63  53111                                  SA1       B1+X1
               43001                             MX0       1
                    20070                        LX0       60-SF.PCQ1            .. MAKE SURE SF.PCQ1 AND SF.PCQ2
                         15610                   BX6       -X0*X1                ARE BOTH ZERO.THEN SET SF.PCQ1
L     64  43001                                  MX0       1
               20067                             LX0       60-SF.PCQ2
                    15660                        BX6       -X0*X6
                         43001                   MX0       1
L     65  20070                                  LX0       60-SF.PCQ1
               12606                             BX6       X0+X6        .. AND SF.PCQ1=1
                    54610                        SA6       A1           .. UPDATE TOS
                                       *
                                       *  PCQ(TOS) NOW INDICATES *ABOUT TO EXECUTE*.  LET THE INTERRUPT RUN.
                                       *         WHEN THE INTERRUPT FINISHES, (AND IF THE STACK HASNT
                                       *         BEEN DIDDLED) CONTROL WILL PASS AGAINTO THIS  ACTION WHICH WIL
                                       *         THEN DISARM INTERRUPTS FOR *CC*
                                       *
L     66  0000000000                   PATCH5.5  PS
L     67  0200000000 X                           JP        TOSPROC
                                       *
                                       *  ERRORS:FRETURN
                                       *
L     70  6140000001                   DISIT5    ERROR     1:NOFIND
L     71  0200000000 X                 DISIT6    JP        SYSFRET
L     72                                         ENDECS    DISIT
```

```
    455                                                ECSCODE    DSPST
                                         *             DISPST AND  DISSEN ARE ECS ACTIONS TO DISPLAY THE STACK
                                         *
                                         *             DISPST
                                         *             PARAMETERS :                    D: USER CM ADDRESS
                                         *                                             D: WORD COUNT, MUST BE .GE. 4
                                         *             THIS ACTION REFORMATS THE STACJ ENTRIES AND TRANSFERS
                                         *             THEM TO AP1. IF THE WORD COUNT IS TOO SMALL , THE TOP N
                                         *             ARE TRANSFERRED,WHERE N=((AP2-1)/3) , THE TOTAL NUMBER
                                         *             OF ENTRIES IN THE STACK IS GIVEN AS THE FIRST WORD
L     51  6120000001                     DISPST        SB2        1
                    5111000143                         SA1        B1+P.XPACK+1
L     52  54212                                        SA2        A1+B2
               21144                                   AX1        36               .RA
                    5131000066                         SA3        B1+P.PARAM
L     53  0333000073                                   NG         X3.NEG1A
                    21244                              AX2        36               .FL
                         54432                         SA4        A3+B2            .WORD COUNT
L     54  63320                                        SB3        X2
               63530                                   SB5        X3
                    0653000074                         GE         B5,B3,BIG1A
L     55  63445                                        SB4        X4+B5
               67442                                   SB4        B4-B2            .END OF WRITE AREA
                    0734000075                         GT         B4,B3,BIG2A
L     56  67442                                        SB4        B4-B2
               43030                                   MX0        24
                    6274777773                         SB7        X4-4
L     57  5121000164                                   SA2        B1+P.STACK
                    0770000076                         NG         B7,NEG2P
L     60  73322                                        SX3        X2+B2            .END OF STACK
               21222                                   AX2        18               .STACK ORIGIN
                    73220                              SX2        X2
                         37232                         IX2        X3-X2
L     61  73222                                        SX2        X2+B2            .STACK LENGTH
               23622                                   AX6        X2,B2            .NUMBER OF ENTRIES
                    63365                              SB3        X6+B5
                         63323                         SB3        B3+X2
L     62  0743000063                                   LT         B4,B3,DSS2       .SENSE NO ROOM FOR ENTIRE STACK
                    67432                              SB4        B3-B2
L     63  63314                          DSS2          SB3        B4+X1            .END OF WRITE AREA-1
               63731                                   SB7        B1+X3
                    53615                              SA6        B5+X1
                         56172                         SA1        B7+B2
L     64  46000                                        NO         0
               6160000001                              SB6        SD.CC        .. POSITION OF CLASS CODE IN SP DESCRIPTOR

L     65  55212                          DSS3          SA2        A1-B2        .. X2=WD0 OF STACK ENTRY
               55122                                   SA1        A2-B2        .. X1=WD1 OF STACK ENTRY
                    63416                              SB4        X1+B6
                         56314                         SA3        B1+B4        .. X3=CC OF CURRENT SP
L     66  21122                                        AX1        18
               63416                                   SB4        X1+B6
```

```
                    54414                   SA4     R1+B4     .. X4=CC OF EOP SP
                                          *
                                          *STACK ENTRY  VFD  6/FLAGS, 18/F-RET COUNT, 18/IPLIST ADDR, 18/B-COUNTE
                                          *             VFD  24/0, 18/END OF PATH SP, 18/CURRENT SP
                                          *
                          10633           .       BX6     X3
L       67  54662                         .       SA6     A6+B2     .. XMIT CURRENT SP
                  10644                           BX6     X4
                       54462                      SA6     A6+B2     .. XMIT EOP SP
                            10611                 BX6     X1
L       70  54662                               SA6     A6+B2     .. XMIT WD 0 OF STACK ENTRY
                  64462                         SB4     A6+B2
                       0743000065               LT      R4,B3,DSS3          .. LOOP IF NOT THRU
        71  0000000000              PATCH6      PS
        72  0400000000 V                       EQ      SYSRET
        73  6140000001              NEG1A       ERROR   1,NEGPT
        74  6140000001              BIG1A       ERROR   1,BIGPT
        75  6140000002              BIG2A       ERROR    2,BIGPT
        76  6140000002              NEG2P       ERROR   2,NEGPAR
        77                                      ENDECS  DSPST
```

```
                                *
                                * DSSTE
                                *       DISPLAY SPECIFIC STACK ENTRY
                                *
                                *       ↑IF↑ NEGPARM(IP1) ↑THEN↑ ERROR ↑FI↑
                                *       ↑IF↑ FLERROR(IP1+3) ↑THEN↑ ERROR  ↑FI↑
                                *       ↑IF↑ NEGPARM(IP2) ↑THEN↑ ERROR ↑FI↑
                                *       ↑IF↑ BIGPARM(IP2) ↑THEN↑ ERROR ↑FI↑
                                *       WRITECORE( REFORMAT( STACKENTRY( IP2)),REFORMATSIZE)
                                *       RETURN
                                *
                                *       ACTUAL PARAMETER AREA RELATIVE P.PARAM
                                *
 66                                     ORG     P.PARAM
 66                     STK.CMAD BSS     1              .. DI CM ADDRESS
 67                     STK.WDCT BSS     1              .. DI WORD COUNT (FOR DSPST)
                    67  STK.INDX EQU     STK.WDCT       .. DI STACK INDEX (FOR DSSTE)
                                USE     *
                                *
                                *       THE SAME REFORMATING IS USED HERE AS IS USED IN DSPST
                                *       (DISPLAY  STACK)
                                *
                                *       REGISTER ALLOCATION
                                *       B2:= 1 (CONSTANT)
                                *       B7:= -1 (CONSTANT)
                                *
                                        ENTRY   DSSTE
                                        ECSCODE DSSTE
                                *
L                   DSSTE       SB2     1              .. B2 := 1 (CONSTTANANT)
L    51  6120000001 5111000066  SA1     B1+STK.CMAD    . CHECK OUT THE PROFFERED
L    52  6160000003             SB6     3              . BUFFER
L    53  6170000054             SB7     DSSTE1
L    53  0400000000 X           EQ      =XCHKPTR
                                *
                                *       DO ERROR CHECKING ON IP2.  FETCH STTCK ENTRY (IP2)
                                *
L    54  5111000164             DSSTE1  SA1     B1+P.STACK     .. X1 := C(P.STACK)=
                                *                       VFD   6/0,18/STACKEND,18/STACKORIGIN,18/ST
                                *                       (RELATIVE B1)
                                *
                                *
                 5121000067             SA2     B1+STK.INDX    . X2 := STK.INDX := IP2 = STACK
                                *                       INDEX RELATIVE TOS (TOP OF STACK)
                                *
L    55  36222                          IX2     X2+X2          .. DOUBLE INDEX (STACK ENTRIES ARE 2 WDS)
         0332000044                     NG      X2,DSSTE3      .. JP IF NEG STACK INDEX (ERROR)
             73410                      SX4     X1
L    56  37242                          IX2     X4-X2          .. X2 := STACK INDEX REL. STACK ORIGIN
         63421                          SB4     X2+B1          .. PTR TO STACK ENTRY
             2111p2                     AX1     18             .. RIGHT JUSTIFY STACK ORIGIN
                 63511                  SB5     X1+B1          .. PTR TO STACK ORIGIN
L    57  0745000065                     LT      B4,B5,DSSTE4           .. ERROR: INDEX TOO LARGE.
                                *                       REMEMBER THAT INDEX TOO LARGE (REL. TOS)
```

```
                                    *                                   ==> PTR LESS THAN STACK ORIGIN
                                    *
                    54140                   SA1     B4          .. X1 := STACK ENTRY (WD 0) =
                        10611               BX6     X1          .. VFD  6/FLAGS,18/FRET.CNT,18/TP.ADR,18/P.CT
L       60  54600                           SA6     A0
                54242                       SA2     B4+B2       .. X2 := STACK ENTRY (WD 1) =
                                    *                           VFD  24/0,18/ENDPATH.SP,18/CURRENT.SP
                                    *
                    63421                   SB4     X2+B1       .. ABS ADDR OF SUBPROCESS DESCRIPTOR
L       61  5144000001                       SA4     B4+SD.CC    .. FETCH CLASS CODE (CC)
                    10644               BX6     X4
                        54662           SA6     A6+B2       .. XMIT WD 1 (REFORMATTED ) := PC.CURRENT
L       62  21222                       AX2     18          .. RIGHT JUSTIFY END OF  PATH
                63421                   SB4     X2+B1
                    5144000001          SA4     B4+SD.CC    .. FETCH CC (END OF PATH)
L       63  10644                       BX6     X4
                54662                   SA6     A6+B2       .. XMIT CC OF END OF  PATH
                    0200000000 X        JP      SYSRET
L       64  6140000002        DSSTE3      ERROR   2,NEGPAR
L       65  6140000002        DSSTE4      ERROR   2,BIGPAR
L       66                              ENDECS  DSSTE
```

```
      520
       51  0000000000      ECSCODE   PTINT
                           PS
       52                  ENDECS    PTINT
```

```
                                *
                                * THIS ROUTINE INSPECTS THE FLAGS PRESENTED IN X6 AND RESCHEDULES THE
                                * PROCESS IF THE OMENS ARE CORRECT.  IT WRITES THE UPDATED P.ROHEAD
                                * INTO ECS
                                *
                                * AT ENTRY
                                *         X6 = P.ROHEAD WORD FOR THE PROCESS
                                *         X0 = ECS A(PROCESS)
                                *         A0 = A(SUITABLE SCRATCH CELL)
                                *         B7 = RETLINK
                                *         I.LOCK MUST BE SET
                                *
                                * USES EVERYTHING BUT B1,B2,B3,B4,B6
                                *
                                          ENTRY     CHKSCHD
0  10566                        CHKSCHD   BX5       X6
        20502                             LX5       PF.R
              033E000006 +                NG        X5,CHKSCHD1          . FINISHED IF ALREADY RUNNING
1  715000016                              SX5       PF.DSCHD
              20540                       LX5       60-12
                   11556                  BX5       X5*X6      . SEE IF ANY CRITICAL BITS ARE ON
2  031E000006 +                           NZ        X5,CHKSCHD1          . STILL NOT SCHEDULABLE
              435001                      MX5       1
                   20574                  LX5       60-PF.P    . SET PENDING ACTION FLAG FOR SWAPPER
3  12656                                  BX6       X5+X6
        20572                             LX5       PF.P+60-PF.R         . SET RUNNING FLAG
              12656                       BX6       X5+X6
                   54600                  SA6       A0
4  0120000001                             WECS      1          . WRITE UPDATE P.ROHEAD
5  20652                                  LX6       60-18
        730160                            SX1       X6         . PROCESS NOT FOR SCHED
              040000000000 X              EQ        =XSCHED    . SCHEDULE PROCESS AND EXIT
6  54600                        CHKSCHD1  SA6       A0
7  0120000001                             WECS      1          . WRITE NEW P.ROHEAD
10 0270000000                             JP        B7         . EXIT
```

```
                                *
                                * THIS ROUTINE SEARCHES FOR A SUBPROCESS WITH INTERRUPT PENDING WHICH
                                * HAS PRIORITY OVER THE CURRENT SUBP FROM THE TOP OF THE STACK
                                *
                                * A PENDING INTERRUPT FOR THE CURRENT SUBPROCESS WILL NOT BE
                                * REPORTED IF THE INTERRUPT INHIBIT BIT IS SET IN THE STACK.
                                *
                                *
                                * AT ENTRY
                                *         B7 = RETLINK
                                *         B1 = PROCESS SCRATCH AREA
                                *
                                * AT EXIT
                                *         B2 = 0, IF NONE
                                *         B2 = A(SUBP DESCRIPTOR), REL BT, OF NEAREST ANCESTOR WITH
                                *              INTERRUPT PENDING
                                *
                                *         PRESERVES B5
                                *
                                PRIRINT   ENTRY     PRIRINT
                                          ECSSUB    PRINT.BUFA
                                          SA2       B1+P.INTERR          .. CHECK IF WE NEED BOTHER
                                          SB2       0                    .. INITIALIZE B2
                                          ZR        X2,PRIRINT2          .. RETURN IF NO PENDING INTERRUP
                                *
                                * SET B2= CURRENT SUBPROCESS AND BEGIN SEARCH
                                *
                                          SA1       B1+P.STACK           .. FETCH POINTER TO CURRENT TOS
                                          SB2       X1+1      .. PTR TO WD 1 OF STAKC ENTRY
                                          SB3       SD.FINT   .. SHIFT VALUE OF INTERRUPT PENDING BIT
                                *
                                * WD1     VFD       24/0,18/END OF PATH SP, 18/CURRENT SP
                                *
                                *
                                * THE FIRST WORD OF THE SUBPROCESS DESCRIPTOR CONTAINS THE
                                * INTERRUPT BIT AND THE FATHER POINTER
                                *
                                * SD.FIPT VFD       4/FLAGS, 18/INTERRUPT DATUM, 18/CLIST LENGTH, 18/FAT
                                *
                                          SA2       B1+B2      . GET PTR TO CURRENT SUBP
                                          SA1       B1+X2      . WORD 1 OF CURRNET SUBP
                                          LX2       B3,X1
                                          PL        X2,PRIRINT3          . NO INT FOR CURRENT
                                          SA2       A2-1       . GET WORD 0 OF STACK
                                          LX2       SF.II      . SEE IF INTERRUPTS ARE INHIBITED
                                          PL        X2,PRIRINT2          . NO
PRIRINT3                  SA1       A1+SD.PTRS           . GET FATHER POINTER
                                          LX1       60-18
                                          SB2       X1
                                          EQ        B2,B0,PRIRINT2       . WERE AT THE ROOT
                                          SA1       B1+B2      . WORD 0 OF FATHER
```

```
                         22231                LX2      B3,X1
    460  0322000456                           PL       X2,PRIRINT3        . NOTHING PENDING, LOOP
    461  0270000000               PRIRINT2    JP       B7          .. RETURN WITH B2 SET APPROPRIATELY
    462                                       ENDSUB   PRINT,BUFA
                          11 +    PRIRINT     EQU      PRINT
```

```
                                ENTRY   S.CHIP,S.ARITH,CHIPCAL,ARITHER
                              *
                              * XPACK FOR RA+1 NOT 0
                              *
14                            S.CHIP    BSSZ    16
                              *
                              * XPACK FOR ARITH ERROR (P-COUNTER = 0)
                              *
34                            S.ARITH   BSSZ    16
54  0000000000                CHIPCAL   PS
55  0000000000                ARITHER   PS
56                                      END

                    36354     STORAGE USED          2305 STATEMENTS     475 SYMBOLS
                              6600 ASSEMBLY        11.841 SECONDS       715 REFERENCES
```

SYMBOLIC REFERENCE TABLE.

AB.CPAVL	10		20/39	24/42					
AB.CPUSD	11		20/40						
AB.SIZE	13		20/38	21/18					
APT.AB	66		20/11 L	20/33	20/53	23/34	23/51	33/29	
APT.D	72		20/13 L	20/27	23/26				
APT.P	70		20/12 L	20/50	20/52	23/50			
ARITHER	55	PROGRAM*	43/01 E	43/11 L					
ARMIT	51		32/09 E	32/12 L					
ARMIT2	53		32/12	32/18 L					
ARMIT3	60		32/20	32/30 L					
AT.PROC	1		11/08						
BIGIX	0	EXTERNAL*	17/08						
BIGPAR	0	EXTERNAL*	17/03	17/04	17/05	38/21			
BIGPT	0	EXTERNAL*	17/06	36/18	36/19				
BIG1A	74		35/21	36/17 L					
BIG2A	75		35/24	36/18 L					
BLMTSS	0	EXTERNAL*	17/10						
BUFA	450		41/22	42/05	42/05	42/05			
CAPAB	0	EXTERNAL*	11/15	12/01					
CHIPCAL	54	PROGRAM*	43/01 E	43/10 L					
CHKPTR	0	EXTERNAL*	37/34						
CHKSCHD	0	PROGRAM*	21/43	40/15 E	40/16 L				
CHKSCHD1	6	PROGRAM*	40/18	40/22	40/33 L				
CLASCDE	66		33/29 D	33/33					
CLEN	56		6/35 D	7/08 S	13/02	14/32			
CLIIB	51		29/14 E	29/17 L					
CLMOT	0	EXTERNAL*	17/09						
CLRMG	51		26/12 E	26/16 L					
CLRMG2	57		26/28 L						
CLRMG3	65		26/17	26/37 L					
CMRUFF	50		6/45	23/26	27/27	29/16	31/02	33/29	37/29
			17/12	25/05	27/51	29/27	31/04	34/50	38/22
			20/27	26/14	28/02	30/09	32/11	35/02	39/02
			22/02	26/39	28/04	30/20	32/32	36/21	39/04
COMPSPA	64		6/41 D	11/38 S	16/04				
CORELEN	62		6/39 D	8/29 S	12/05				
CPOUT	51		23/24 E	23/26 L					
CPOUT10	123		23/51	25/03 L					
CPOUT2	102		24/18 L	24/26	24/38				
CPOUT3	104		24/16	24/23 L	24/32				
CPOUT4	106		24/06	24/30 L					
CPOUT6	112		24/19	24/42 L					
CPOUT7	120		23/28	24/53 L					
CPOUT8	121		23/37	25/01 L					
CPOUT9	122		25/02 L						
CPUIN	51		20/35 E	20/27 L					
CPUIN10	121		20/51	21/49 L					
CPUIN11	122		20/43	21/50 L					
CPUIN12	123		20/53	21/51 L					
CPUIN5	110		21/25 L	21/31	21/32	21/42			
CPUIN6	112		21/20	21/21	21/31 L				
CPUIN7	117		20/38	21/47 L					
CPUIN9	120		20/34	21/48 L	23/46				

DESCSIZ	61		6/38	D	8/16	S	12/20	13/28	15/51				
DISASTR	0	EXTERNAL*	16/42										
DISIT	51		33/27	E	33/32	L							
DISIT2	53		33/32		33/39	L							
DISIT3	61		33/50		34/14	L							
DISIT4	62		34/14		34/19	L							
DISIT5	70		33/41		34/47	L							
DISIT6	71		34/19		34/48	L							
DISPST	51		35/11	L									
DSSTE	51		37/27	E	37/30	L							
DSSTE1	54		37/33		37/38	L							
DSSTE3	64		37/46		38/19	L							
DSSTE4	65		37/52		38/20	L							
DSS2	63		35/38		35/40	L							
DSS3	65		35/48	L	36/13								
ECSAD	63		6/40	D	11/14	S	13/40						
ENDSYS	0	EXTERNAL*	8/23										
E.ARMIT	424	EPROC	32/10	L									
E.CLIIB	415	EPROC	29/15	L									
E.CLRMG	355	EPROC	26/13	L									
E.CPOUT	302	EPROC	23/25	L									
E.CPROC	0	EPROC	6/44	L									
E.CPUIN	227	EPROC	20/26	L									
E.DISIT	434	EPROC	33/28	L									
E.DLPRO	414	EPROC	28/01	L									
E.DPROD	423	EPROC	31/01	L									
E.DSPST	455	EPROC	35/01	L									
E.DSSTE	503	EPROC	37/38	L									
E.ECS	0	EXTERNAL*	6/52		16/09		20/39	21/08	23/37	24/14	26/17	27/45	
			7/06		16/22		20/51	21/12	23/46	24/19	26/34	40/30	
			13/44		16/32		20/53	21/19	23/51	24/45	27/29	40/35	
			14/20		20/34		20/53	21/37	23/51	24/49	27/33	42/05	
E.ERROR	0	EXTERNAL*											
E.NOERR	12		15/01										
E.PRINT	521	EPROC	41/21	L									
E.PTINT	520	EPROC	39/01	L									
E.STIIB	420	EPROC	30/08	L									
E.STMSG	372	EPROC	27/36	L									
FATSON	0	EXTERNAL*	21/52		25/04								
FINDSUB	0	EXTERNAL*	32/14		33/34								
IP.AWDS	2		20/53		23/51								
I.LOCK	0	EXTERNAL*	21/05	S	21/26	S	24/05	S 24/50	S 26/26	S 26/34	S 27/41	S 27/46	S
L.ARMIT	10		32/32	D									
L.CLIIB	3		29/27	D									
L.CLRMG	15		26/39	D									
L.CPOUT	53		25/05	D									
L.CPROC	227		17/12	D									
L.CPUIN	53		22/02	D									
L.DISIT	21		34/50	D									
L.DLPRO	1		28/04	D									
L.DPROD	1		31/04	D									
L.DSPST	26		36/21	D									
L.DSSTE	15		38/22	D									

SYMBOLIC REFERENCE TABLE.

L.PRINT	12		42/05 D	42/05						
L.PTINT	1		39/04 D							
L.STI8	3		30/20 D							
L.STMSG	22		27/51 D							
MAKEOBJ	0	EXTERNAL*	11/09							
MAKPROC	51		6/48 L							
MISSOB	0	EXTERNAL*	21/49	21/50	25/02	25/03	26/38	27/49	27/50	
MKPROC1	135		10/11 L	10/14						
MKPROC2	137		10/10	11/05 L						
MKPROC20	252		11/26	11/28	11/33	16/45 L				
MKPROC3	141		11/07	11/12 L						
MKPROC30	253		7/28	16/43 L						
MKPROC31	254		7/44	16/44 L						
MKPROC33	255		8/06	16/45 L						
MKPROC35	257		8/21	16/47 L						
MKPROC36	260		8/35	16/48 L						
MKPROC37	273		10/06	17/06 L						
MKPROC38	261		8/52	16/49 L						
MKPROC39	262		9/06	16/50 L						
MKPROC4	144		11/16	11/22 L						
MKPROC40	263		9/10	16/51 L						
MKPROC41	272		9/19	9/48	17/05 L					
MKPROC42	276		9/17	9/46	17/09 L					
MKPROC43	256		8/13	16/46 L						
MKPROC44	275		7/03	17/08 L						
MKPROC5	150		11/27	11/30 L						
MKPROC6	153		11/32	11/35 L						
MKPROC7	165		12/29 L	12/31						
MKPROC98	131		9/47	9/52 L						
MKPROC99	116		9/18	9/26 L						
MK30X	267		7/30	17/02 L						
MK35X	270		8/28	17/03 L						
MK36X	271		8/38	17/04 L						
MK37X	274		10/12	17/07 L						
MK38X	264		9/28	16/52 L						
MK39X	265		9/35	16/53 L						
MK40X	266		9/39	17/01 L						
MK43X	277		10/01	17/10 L						
MP.ALLOC	0		6/12 L	11/05						
MP.CADR	2		5/29 D							
MP.CAPI	2		6/13 L	10/05	11/12					
MP.CHWD	3		6/14 L	7/26	11/50	13/49				
MP.CLIST	13		6/21 L	6/48	15/06					
MP.CLSCD	5		6/16 L	14/47						
MP.CMAD1	20		6/24 L	9/04						
MP.CMAD2	26		6/28 L	9/33						
MP.CMAP	3		5/32 D							
MP.CNT	2		5/21 D							
MP.COMP	10		6/18 L	8/12	9/52	11/35	14/39	15/36	16/13	
MP.ENTRY	12		6/20 L	8/33	14/07	15/25				
MP.FADR	1		5/28 D							
MP.FILE	0		5/27 D							
MP.FILE1	15		6/22 L	8/50						

```
MP.FILE2      22                  6/26 L    9/26
MP.FL         11                  6/19 L    8/20      9/13      9/44      11/22     11/30     15/52
MP.FLAD1      17                  6/23 L    8/51
MP.FLAD2      24                  6/27 L    9/27
MP.MAP         7                  6/17 L    7/52      14/35     15/39     16/23
MP.RAFL        1                  5/20 D
MP.SIZE        3                  5/26 D   16/21
MP.STACK       4                  6/15 L    7/42      14/20
MP.WDCT1      21                  6/25 L    9/08
MP.WDCT2      26                  6/29 L    9/37
NEGIX          0      EXTERNAL*   17/07
NEGPAR         0      EXTERNAL*   16/44    16/46      16/48     16/50     16/52     17/01     21/48     36/20
                                  16/45    16/47      16/49     16/51     16/53     17/02     25/01     38/20
NEGPT          0      EXTERNAL*   36/17
NEG1A         73                  35/16    36/16 L
NEG2P         76                  35/29    36/19 L
NOCP           0      EXTERNAL*   21/51
NOFIND         0      EXTERNAL*   32/31    34/48
NOROOM         0      EXTERNAL*   17/11
PATCH1        52                  23/27 L
PATCH2        63                  26/35 L
PATCH3        67                  27/46 L
PATCH4        56                  32/26 L
PATCH5        57                  34/06 L
PATCH5.5      66                  34/42 L
PATCH6        71                  36/14 L
PF.C           6                   3/43 D
PF.CREA     4010                  3/48 D   11/48
PF.D           4                   3/41 D
PF.DSCHD      16                   3/51 D   40/19
PF.E           5                   3/42 D
PF.EE         11                   3/46 D
PF.H          12                   3/47 D
PF.I           3                   3/40 D
PF.KILL     4200                   3/49 D
PF.P           0                   3/37 D   40/24     40/26
PF.R           2                   3/39 D   40/17     40/26
PF.S          10                   3/45 D   21/35
PF.SWPN     6636                   3/50 D
PF.V           7                   3/44 D
PF.W           1                   3/38 D
PRINT         11      PROGRAM*    34/15    42/05 L   42/05
PRIRINT       11      PROGRAM*    41/20 E  42/05 D
PRIRINT2     461                  41/24    41/48     41/52     42/03 L
PRIRINT3     456                  41/45    41/49 L   42/02
PS.MASKL       1                   4/34 D
PS.TEMP        1                   4/31 D    4/33 D    4/33      4/34
PUTCAP         0      EXTERNAL*   11/17
P.APLL        35                   3/07 D
P.CLIST      162                   4/18 L   10/07
P.CLLOFF       0                   5/13 D
P.CTABLE     163                   4/19 L
P.ERROFF       1                   5/14 D
```

P.INTERR	167		
P.INTOFF	2		
P.ITLOFF	3		
P.LOCALC	171		
P.MAPSIN	166		
P.MSGDAT	135		
P.MSGEVC	134		
P.PARAM	66		
P.PARAMC	125		
P.PARAML	40		
P.PBUFL	372		
P.PROCRO	11		
P.PROCRW	32		
P.ROHEAD	126		
P.RWHEAD	137		
P.SCHED	130		
P.SCHEDL	4		
P.SCR	0		
P.SCRL	50		
P.SCR2	50		
P.SCR2L	6		
P.STACK	164		
P.SUBPDT	165		
P.SWPTIM	141		
P.SYSTIM	140		
P.TEMP1	56		
P.TEMP2	57		
P.TEMP3	60		
P.TEMP4	61		
P.TEMP5	62		
P.TEMP6	63		
P.TEMP7	64		
P.TIMER	136		
P.USRTIM	137		
P.XPACK	142		
REFER	0	EXTERNAL*	9/21 9/50 11/29 11/34
SCHED	0	EXTERNAL*	40/32
SD.CC	1		4/43 D 13/29 32/24 35/45 38/15 8/03 14/49 S 33/46 38/10
SD.CLST	3		4/45 D 15/08 S
SD.ESM	2		4/44 D 15/02 S
SD.FDIS	2		5/06 D 32/23 34/03
SD.FINT	0		5/04 D 33/49 41/30
SD.FIPT	0		4/41 D 14/43 S
SD.FMPE	1		5/05 D
SD.MAP	6		4/48 D 4/51 15/44 S

4/23 L	41/22						
5/15 D							
5/16 D	14/08						
4/36 D							
4/22 L	13/43						
4/06 L	26/26 S						
4/05 L	26/31	27/42					
3/25 L	8/33	9/27	11/12	14/20	15/39	27/28	
6/48	8/50	9/33	11/22	14/35	15/52	27/32	
7/26	8/51	9/37	11/36	14/39	16/13	27/34	
7/42	9/04	9/44	11/35	14/47	16/23	32/13	
7/52	9/08	9/52	11/50	15/06	20/10	34/24	
8/12	9/13	10/05	13/48	15/25	23/45	35/15	
8/20	9/26	11/05	14/07	15/36	26/14	37/14	
3/26 D							
3/06 D	3/07	3/25	4/37				
3/11 D							
4/08 D	7/18	7/35	12/14	13/50			
4/25 D	7/18	7/35	12/16	13/50			
3/32 L	7/18	21/05	24/36	27/42			
4/08	13/43	24/11	26/31				
4/13 D	4/25						
4/23 L							
3/08 D	4/03	12/28					
3/14 L	7/18						
3/09 D	3/14	7/29					
3/15 L	9/02 S	9/14	9/31 S	9/45	11/29	11/31	16/26
3/10 D	3/15						
4/20 L	29/18	30/13	35/28	37/38	41/28		
4/21 L							
4/16 L							
4/15 L							
3/16 L	6/35						
3/17 L	6/36						
3/18 L	6/37						
3/19 L	6/38						
3/20 L	6/39						
3/21 L	6/40						
3/22 L	6/41						
4/07 L	21/05	24/11	24/36	24/36			
4/14 L							
4/17 L	12/45	21/12 S	24/50 S	35/12			

SD.ORIG	5		4/47	D	15/28	S			
SD.PTRS	4		4/46	D	15/18	S	41/49		
SD.RAFL	7		4/51	D	16/06	S			
SD.SIZE	10		4/40	D	8/03		14/41	16/08	16/14
SF.FINT	1		5/43	D					
SF.II	0		5/42	D	14/04		29/20	30/11	41/47
SF.PCQ1	4		5/44	D	14/04		14/06	34/27	34/33
SF.PCQ2	5		5/45	D	34/30				
STACKO	57		6/36	D	7/22	S	12/19	13/07	13/16
STIIB	51		30/07	E	30/10	L			
STK.CMAD	66		37/15	L	37/31				
STK.INDX	67		37/17	D	37/42				
STK.WDCT	67		37/16	L	37/17				
STMSG	51		27/25	E	27/28	L			
STMSG3	71		27/29		27/48	L			
STMSG4	72		27/23		27/49	L			
SURPT	60		6/37	D	7/48	S	13/15	14/11	15/15
SYSFRET	0	EXTERNAL*	34/48						
SYSRET	0	EXTERNAL*	16/33		24/52		27/47	32/28	36/15
			21/26		26/36		30/18	34/07	38/18
S.ARITH	34	PROGRAM*	43/01	E	43/09	L			
S.CHIP	14	PROGRAM*	43/01	E	43/05	L			
S.CMFL	0	EXTERNAL*	8/25		12/42				
S.FNFLG	2		5/36	D					
TOSPROC	0	EXTERNAL*	29/23		34/43				

```
13.17.44. 11/15/71 SCOP320 OF 08/27/71
13.17.53.$:  CM=18432/044000B AT CP=    0 SEC
13.17.58.NOMPASS,I=PROCESS,S=0
13.19.12. ASSEMBLY COMPLETE.
13.19.21.END
13.19.23.FIN
13.19.24.$:  USER CPU   -   12.579 SEC
13.19.24.$:  SCOPE CPU  -    5.152 SEC
13.19.25.$:  SCOPE ECS  -    5.769 SEC
13.19.25.$:  SCOPE SWAP-    14.482 SEC
13.19.25.$:  DISK CPU   -    4.591 SEC
13.19.25.$:  DISK ECS   -    8.222 SEC
13.19.26.$:  DISK SWAP  -     .949 SEC
13.19.26.$:  SYSTEXT    -    2070 LINES


13.22.21.       2083 LINES PRINTED BY PRINTER DRIVER ON LP 2.
        SAVE FOR VANCE
        SAVE FOR VANCE
        SAVE FOR VANCE
        SAVE FOR VANCE
        SAVE FOR VANCE
        SAVE FOR VANCE
        SAVE FOR VANCE
        SAVE FOR VANCE
        SAVE FOR VANCE
        SAVE FOR VANCE
        SAVE FOR VANCE
        SAVE FOR VANCE
```