L 12

```
   SSSSSSSSSSSS  YYY         YYY     SSSSSSSSSSSS
   SSSSSSSSSSSS  YYY         YYY     SSSSSSSSSSSS
   SSSSSSSSSSSS  YYY         YYY     SSSSSSSSSSSS
SSS              YYY         YYY  SSS
SSS              YYY         YYY  SSS
SSS              YYY         YYY  SSS
SSS                 YYY   YYY     SSS
SSS                 YYY   YYY     SSS
SSS                 YYY   YYY     SSS
   SSSSSSSSS           YYY           SSSSSSSSS
   SSSSSSSSS           YYY           SSSSSSSSS
   SSSSSSSSS           YYY           SSSSSSSSS
            SSS        YYY                    SSS
            SSS        YYY                    SSS
            SSS        YYY                    SSS
            SSS        YYY                    SSS
            SSS        YYY                    SSS
            SSS        YYY                    SSS
SSSSSSSSSSSS           YYY        SSSSSSSSSSSS
SSSSSSSSSSSS           YYY        SSSSSSSSSSSS
SSSSSSSSSSSS           YYY        SSSSSSSSSSSS
```

```
**FILE**ID**SYSMAILBX                                          I




  SSSSSSSS  YY      YY    SSSSSSSS  MM      MM    AAAAAA      IIIIII    LL          BBBBBBBB    XX      XX
  SSSSSSSS  YY      YY    SSSSSSSS  MM      MM    AAAAAA      IIIIII    LL          BBBBBBBB    XX      XX
SS          YY      YY  SS          MMMM  MMMM  AA      AA      II      LL          BB      BB  XX      XX
SS          YY      YY  SS          MMMM  MMMM  AA      AA      II      LL          BB      BB  XX      XX
SS            YY  YY    SS          MM  MM  MM  AA      AA      II      LL          BB      BB    XX  XX
SS            YY  YY    SS          MM  MM  MM  AA      AA      II      LL          BB      BB    XX  XX
  SSSSSS        YY        SSSSSS    MM      MM  AA      AA      II      LL          BBBBBBBB        XX
  SSSSSS        YY        SSSSSS    MM      MM  AA      AA      II      LL          BBBBBBBB        XX
        SS      YY              SS  MM      MM  AAAAAAAAAA      II      LL          BB      BB    XX  XX
        SS      YY              SS  MM      MM  AAAAAAAAAA      II      LL          BB      BB    XX  XX
        SS      YY              SS  MM      MM  AA      AA      II      LL          BB      BB  XX      XX      ....
        SS      YY              SS  MM      MM  AA      AA      II      LL          BB      BB  XX      XX      ....
SSSSSSSS        YY      SSSSSSSS    MM      MM  AA      AA    IIIIII    LLLLLLLLLL  BBBBBBBB    XX      XX      ....
SSSSSSSS        YY      SSSSSSSS    MM      MM  AA      AA    IIIIII    LLLLLLLLLL  BBBBBBBB    XX      XX      ....

LL            IIIIII      SSSSSSSS
LL            IIIIII      SSSSSSSS
LL              II      SS
LL              II      SS
LL              II      SS
LL              II      SS
LL              II        SSSSSS
LL              II        SSSSSS
LL              II              SS
LL              II              SS
LL              II              SS
LL              II              SS
LLLLLLLLLL    IIIIII    SSSSSSSS
LLLLLLLLLL    IIIIII    SSSSSSSS
```

Table of contents

```
0000            .TITLE  SYSMAILBX - SYSTEM SERVICES TO CREATE AND DELETE MAILBOXES
0000            .IDENT  'V04-000'
0000
0000    ;
0000    ;****************************************************************************
0000    ;*                                                                          *
0000    ;*  COPYRIGHT (c) 1978, 1980, 1982, 1984 BY                                 *
0000    ;*  DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASSACHUSETTS.                  *
0000    ;*  ALL RIGHTS RESERVED.                                                    *
0000    ;*                                                                          *
0000    ;*  THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND COPIED   *
0000    ;*  ONLY IN  ACCORDANCE WITH  THE  TERMS  OF  SUCH  LICENSE  AND WITH THE   *
0000    ;*  INCLUSION OF THE ABOVE COPYRIGHT NOTICE. THIS SOFTWARE OR  ANY  OTHER   *
0000    ;*  COPIES THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY   *
0000    ;*  OTHER PERSON.  NO TITLE TO AND OWNERSHIP OF  THE  SOFTWARE IS  HEREBY   *
0000    ;*  TRANSFERRED.                                                            *
0000    ;*                                                                          *
0000    ;*  THE INFORMATION IN THIS SOFTWARE IS  SUBJECT TO CHANGE WITHOUT NOTICE   *
0000    ;*  AND  SHOULD  NOT  BE  CONSTRUED AS  A COMMITMENT BY DIGITAL EQUIPMENT   *
0000    ;*  CORPORATION.                                                            *
0000    ;*                                                                          *
0000    ;*  DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE  OR  RELIABILITY OF ITS   *
0000    ;*  SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL.                 *
0000    ;*                                                                          *
0000    ;*                                                                          *
0000    ;****************************************************************************
0000    ;
0000    ; D. N. CUTLER 1-OCT-76
0000    ;
0000    ; MODIFIED BY:
0000    ;
0000    ;       V03-018 RAS0327         Ron Schaefer            24-Jul-1984
0000    ;               Change $TRNLNM item list to use the retlen parameter,
0000    ;               and only check a word-sized length value.
0000    ;
0000    ;       V03-017 LMP0271         L. Mark Pilant,         29-Jun-1984  13:06
0000    ;               Disallow ACLs on shared memory mailboxes.
0000    ;
0000    ;       V03-016 LJK0285         Lawrence J. Kenah       29-May-1984
0000    ;               Check status returned by invocation of LOCK macro. Exit
0000    ;               with error if attempt to gain interlock fails.
0000    ;
0000    ;       V03-015 RNH0001         Richard N. Holstein     08-May-1984
0000    ;               Fix typo in CREATE_SHMUCB which omitted a register.
0000    ;
0000    ;       V03-014 TMK0001         Todd M. Katz            28-Apr-1984
0000    ;               Modify EXE$CREMBX to increase the size of the logical name
0000    ;               associated with a mailbox so that it can be full sized
0000    ;               ( i.e. - 0 < LOGNAM size =< LNM$C_NAMLENGTH ). This is
0000    ;               accomplished through use of a KRP. When a logical name is
0000    ;               specified, it is translated by the internal executive routine
0000    ;               MMG$MBXTRNLOG. The KRP is utilized as the buffer to contain the
0000    ;               translated logical name, or the specified logical name if
0000    ;               MMG$MBXTRNLOG was unable to translate it.
0000    ;
0000    ;       V03-013 MSH0029         Michael S. Harvey       9-Apr-1984
0000    ;               Eliminate security holes in temporary mailbox creation where
```

```
0000  ;                one without appropriate privilege could create system or
0000  ;                group wide logical names. This undoes RAS0185 below.
0000  ;
0000  ;        V03-012 LMP0221         L. Mark Pilant,         30-Mar-1984  16:07
0000  ;                Change UCB$L_OWNUIC to ORB$L_OWNER and UCB$W_VPROT to
0000  ;                ORB$W_PROT.
0000  ;
0000  ;        V03-011 ROW0240         Ralph O. Weber          11-OCT-1983
0000  ;                Correct calling sequence to IOC$CVT_DEVNAM in CREATE_LOGNAM to
0000  ;                supply a R4, name-type, argument.  Use R4=-1 which produces
0000  ;                the same format device name as the previous R4 contents, a PCB
0000  ;                address, produced.  Also increase MBXNAMSIZ to accomodate
0000  ;                possible cluster format device names.
0000  ;
0000  ;        V03-010 RAS0185         Ron Schaefer             5-Sep-1983
0000  ;                For compatibility reasons, mailbox creation implies
0000  ;                logical name table privileges:  TMPMBX -> GRPNAM
0000  ;                and PRMMBX -> SYSNAM for creating the logical names.
0000  ;                So we do a $SETPRV as necessary to set/clear xNAM.
0000  ;                Add page-locking code to CREATE_LOGNAM subroutine.
0000  ;
0000  ;        V03-009 RAS0175         Ron Schaefer             1-Aug-1983
0000  ;                Turn the "_" in the mailbox name into the proper
0000  ;                LNM$M_TERMINAL attribute.
0000  ;
0000  ;        V03-008 DMW4058         DMWalp                  22-Jun-1983
0000  ;                Change $xxLNM value parameters to be by reference
0000  ;
0000  ;        V03-007 DMW4052         DMWalp                  20-Jun-1983
0000  ;                Added LNM$_LNMB_ADDR in place of ^XFFFF
0000  ;                Replaced HALT with BUGCHECK
0000  ;                Replaced ENQ call with mutex
0000  ;
0000  ;        V03-006 DMW4031         DMWalp                  26-May-1983
0000  ;                1st pass at intergating new logical name structures.
0000  ;
0000  ;        V03-005 ROW0180         Ralph O. Weber          13-APR-1983
0000  ;                Test for possibly zero UCB pointer in DDB in routine
0000  ;                GET_SHMTEMPLATE.
0000  ;
0000  ;        V03-004 ROW0168         Ralph O. Weber           3-MAR-1983
0000  ;                Change resource locking routine calls to G^ references.
0000  ;
0000  ;        V03-003 SRB0067         Steve Beckhardt         21-Feb-1983
0000  ;                Fixed branch displacement to eliminate truncation error.
0000  ;
0000  ;        V03-002 DMW4011         DMWalp                  19-Nov-1982
0000  ;                Recode Logical Name Block to UCB ( or MTL ) pointer
0000  ;
0000  ;        V03-001 ROW0127         Ralph O. Weber          19-SEP-1982
0000  ;                Make changes required to use new UCB creation routines in
0000  ;                UCBCREDEL.  Modify CHECK_QUOTA to use IOC$_CHKMBXQUOTA.
0000  ;                Modify CREATE_SHMUCB and CREATE_LOCALUCB to use appropriate
0000  ;
0000  ;                UCB creation routines.  Fix CREATE_SHMUCB to use elevated IPL
0000  ;                one page locking technique.  Modify COMP_CHARGES to use
0000  ;                IOC$DEBIT_UCB.  Add R4 to inputs lists for CREATE_SHMUCB
```

```
                            0000   115 ;              and CREATE_LOCALUCB.
                            0000   116 ;
                            0000   117 ; SYSTEM SERVICES TO CREATE AND DELETE MAILBOXES
                            0000   118 ;-
```

```
                 0000            .SBTTL  DATA DECLARATIONS
                 0000
                 0000      ;
                 0000      ; MACRO LIBRARY CALLS
                 0000      ;
                 0000
                 0000            $ADPDEF                         ;DEFINE ADAPTER CONTROL BLOCK OFFSETS
                 0000            $CCBDEF                         ;DEFINE CCB OFFSETS
                 0000            $CRBDEF                         ;DEFINE CRB OFFSETS
                 0000            $DDBDEF                         ;DEFINE DDB OFFSETS
                 0000            $DEVDEF                         ;DEFINE DEVICE CHARACTERISTIC BITS
                 0000            $DYNDEF                         ;DEFINE DATA STRUCTURE TYPE CODES
                 0000            $IDBDEF                         ;DEFINE INTERRUPT DISPATCH BLOCK
                 0000            $IPLDEF                         ;DEFINE INTERRUPT PRIORITY LEVELS
                 0000            $JIBDEF                         ;DEFINE JIB OFFSETS
                 0000            $LCKDEF                         ;DEFINE LOCK MANAGER OFFSETS
                 0000            $LNMDEF                         ;DEFINE LNM OFFSETS
                 0000            $LNMSTRDEF                      ;DEFINE LNM STRUCTURE OFFSETS
                 0000            $MBXDEF                         ;DEFINE SHARED MEMORY MAILBOX
                 0000            $ORBDEF                         ;DEFINE OBJECT'S RIGHTS BLOCK OFFSETS
                 0000            $PCBDEF                         ;DEFINE PCB OFFSETS
                 0000            $PRDEF                          ;DEFINE PROCESSOR REGISTERS
                 0000            $PRVDEF                         ;DEFINE PRIVILEGE BITS
                 0000            $PSLDEF                         ;DEFINE PSL OFFSETS
                 0000            $SHBDEF                         ;DEFINE SHARED MEMORY CONTROL BLOCK
                 0000            $SHDDEF                         ;DEFINE SHARED MEMORY DATAPAGE
                 0000            $SSDEF                          ;DEFINE SYSTEM STATUS CODES
                 0000            $UCBDEF                         ;DEFINE UCB OFFSETS
                 0000            $VECDEF                         ;DEFINE INTERRUPT VECTOR OFFSETS
                 0000
                 0000      ;
                 0000      ; LOCAL SYMBOLS
                 0000      ;
        00000011 0000      MBXNAMSIZ = 17                        ;MAXIMUM MAILBOX DEVICE NAME SIZE
                 0000                                            ;  _NODENM$MBxnnnnn:
                 0000
                 0000      ;
                 0000      ; ARGUMENT LIST OFFSET DEFINITIONS FOR CREATE MAILBOX
                 0000      ;
                 0000
                 0000            $OFFSET 4,POSITIVE,<-
                 0000                    PRMFLG,-                ;MAILBOX CREATION TYPE INDICATOR
                 0000                    CRCHAN,-                ;ADDRESS TO STORE ASSIGNED CHANNEL
                 0000                    MAXMSG,-                ;MAXIMUM MAILBOX MESSAGE SIZE
                 0000                    BUFQUO,-                ;MAXIMUM BYTES FOR MESSAGE BUFFERING
                 0000                    PROMSK,-                ;PROTECTION MASK FOR CREATED MAILBOX
                 0000                    ACMODE,-                ;ACCESS MODE FOR CHANNEL ASSIGNMENT
                 0000                    LOGNAM,-                ;ADDRESS OF LOGICAL NAME STRING DESCRIPTOR
                 0000                    >
                 0004      PRMFLG:
                 0008      CRCHAN:
                 000C      MAXMSG:
                 0010      BUFQUO:
                 0014      PROMSK:
                 0018      ACMODE:
                 001C      LOGNAM:
                 0000
```

```
0000   170 ;
0000   171 ; ARGUMENT LIST OFFSET DEFINITIONS FOR DELETE MAILBOX
0000   172 ;
0000   173        $OFFSET 4,POSITIVE,<-
0000   174               DLCHAN,-                  ;CHANNELT NUMBER OF SPECIFIED MAILBOX
0000   175               >
0004       DLCHAN:
0000   176 ;
0000   177 ; STACK LOCAL STORAGE SYMBOLS
0000   178 ;
0000   179        $OFFSET 0,NEGATIVE,<-
0000   180               LOCKID,-                  ;LOCK IDENT
0000   181               LOCKSTATUS,-              ;LOCK BLOCK STATUS
0000   182               CHANADDR,-                ;ADDRESS TO STORE CHANNEL NUMBER
0000   183               CCBADDR,-                 ;CCB ADDRESS FOR THAT CHANNEL
0000   184               CHANNUM,-                 ;ASSIGNED CHANNEL NUMBER
0000   185               SHBADDR,-                 ;SHARED MEMORY CONTROL BLOCK ADDR
0000   186               UCB_ADDRESS,-             ;UCB ADDRESS, ACCESSED VIA AN ITEMLIST
0000   187               CRE[NM_END_LIST,-         ;END OF ITEM LIST
0000   188               I6_RET_LEN,-              ;ITEM 6 RETURN LENGTH ADDRESS   {NORMAL
0000   189               I6_BUF_ADDR,-             ;ITEM 6 BUFFER ADDRESS          {TRANSLATION
0000   190               I6_CODE_LEN,-             ;ITEM 6 ITEM CODE & BUFFER LEN  {STRING
0000   191               I5_RET_[EN,-              ;ITEM 5 RETURN LENGTH ADDRESS   {TRANSLATION
0000   192               I5_BUF_ADDR,-             ;ITEM 5 BUFFER ADDRESS          {TERMINAL
0000   193               I5_CODE_LEN,-             ;ITEM 5 ITEM CODE & BUFFER LEN  {ATTRIBUTE
0000   194               I4_RET_[EN,-              ;ITEM 4 RETURN LENGTH ADDRESS   {TRANSLATION
0000   195               I4_BUF_ADDR,-             ;ITEM 4 BUFFER ADDRESS          {INDEX
0000   196               I4_CODE_LEN,-             ;ITEM 4 ITEM CODE & BUFFER LEN  { #0
0000   197               I3_RET_[EN,-              ;ITEM 3 RETURN LENGTH ADDRESS   {BACKPTR
0000   198               I3_BUF_ADDR,-             ;ITEM 3 BUFFER ADDRESS          {TO UCB
0000   199               I3_CODE_LEN,-             ;ITEM 3 ITEM CODE & BUFFER LEN  {
0000   200               I2_RET_[EN,-              ;ITEM 2 RETURN LENGTH ADDRESS   {BACKPTR
0000   201               I2_BUF_ADDR,-             ;ITEM 2 BUFFER ADDRESS          {TO UCB
0000   202               I2_CODE_LEN,-             ;ITEM 2 ITEM CODE & BUFFER LEN  {INDEX
0000   203               I1_RET_[EN,-              ;ITEM 1 RETURN LENGTH ADDRESS   {UCB TO
0000   204               I1_BUF_ADDR,-             ;ITEM 1 BUFFER ADDRESS          {LNMB
0000   205               I1_CODE_LEN,-             ;ITEM 1 ITEM CODE & BUFFER LEN  {PTR
0000   206               KRP,-                     ;ADDRESS OF ALLOCATED KRP
0000   207               LOCALSIZ,-                ; SIZE OF LOCAL STORAGE
0000   208               >
FFFC       LOCKID:
FFF8       LOCKSTATUS:
FFF4       CHANADDR:
FFF0       CCBADDR:
FFEC       CHANNUM:
FFE8       SHBADDR:
FFE4       UCB_ADDRESS:
FFE0       CRE[NM_END_LIST:
FFDC       I6_RET_LEN:
FFD8       I6_BUF_ADDR:
FFD4       I6_CODE_LEN:
FFD0       I5_RET_[EN:
FFCC       I5_BUF_ADDR:
FFC8       I5_CODE_LEN:
FFC4       I4_RET_[EN:
FFC0       I4_BUF_ADDR:
FFBC       I4_CODE_LEN:
```

```
                                      FFB8   I3_RET_LEN:
                                      FFB4   I3_BUF_ADDR:
                                      FFB0   I3_CODE_LEN:
                                      FFAC   I2_RET_LEN:
                                      FFA8   I2_BUF_ADDR:
                                      FFA4   I2_CODE_LEN:
                                      FFA0   I1_RET_LEN:
                                      FF9C   I1_BUF_ADDR:
                                      FF98   I1_CODE_LEN:
                                      FF94   KRP:
                                      FF90   LOCALSIZ:
                                      0000
                                      0000   ;
                                      0000   ; REDEFINE LOCAL STORAGE ITEM LIST FOR $TRNLNM AND $CRELNM
                                      0000   ;
                            FFFFFFF0  0000   ZERO_INDEX      = CCBADDR
                            FFFFFFB0  0000   TRNLNM_END_LIST = I3_CODE_LEN
                            FFFFFFB4  0000   UCB_ADDR_SIZE   = I3_BUF_ADDR
                                      0000
                                      0000   ;
                                      0000   ; LOCAL DATA CONSTANTS
                                      0000   ;
                                      0000
                                  00000000           .PSECT  Y$EXEPAGED
                                      0000
                                      0000   ;
                                      0000   ; BACK POINTER INDEX VALUE
                                      0000   ;
                                      0000   BACKPTR_INDEX:
                            FFFFFF81  0000           .LONG    LNMX$C_BACKPTR
                                      0004
                                      0004   ;
                                      0004   ; TERMINAL LOGICAL NAME ATTRIBUTE
                                      0004   ;
                                      0004   LNM_TERM_ATTR:
                            00000200  0004           .LONG   LNM$M_TERMINAL
                                      0008
                                      0008   ;
                                      0008   ; Pointers to mailbox table names descriptors
                                      0008   ;
                                      0008   MAILBOX_TABLES:
                            00000010' 0008           .ADDRESS 10$
                            0000002D' 000C           .ADDRESS 20$
45 54 24 4D 4E 4C 00000018'010E0000'  0010   10$:    .ASCID   "LNM$TEMPORARY_MAILBOX"
 4C 49 41 4D 5F 59 52 41 52 4F 50 4D  001E
                            58 4F 42  002A
45 50 24 4D 4E 4C 00000035'010E0000'  002D   20$:    .ASCID   "LNM$PERMANENT_MAILBOX"
 4C 49 41 4D 5F 54 4E 45 4E 41 4D 52  003B
                            58 4F 42  0047
```

```
004A            .SBTTL  EXE$CREMBX - CREATE MAILBOX
004A     ;+
004A     ; EXE$CREMBX - CREATE MAILBOX
004A     ;
004A     ; THIS SERVICE PROVIDES THE CAPABILITY TO CREATE A VIRTUAL MAILBOX DEVICE
004A     ; AND ASSIGN AN I/O CHANNEL TO IT.
004A     ;
004A     ; INPUTS:
004A     ;
004A     ;       PRMFLG(AP) = MAILBOX CREATION TYPE INDICATOR.
004A     ;               LOW BIT CLEAR INDICATES TEMPORARY MAILBOX.
004A     ;               LOW BIT SET INDICATES PERMANENT MAILBOX.
004A     ;       CRCHAN(AP) = ADDRESS TO STORE ASSIGNED CHANNEL NUMBER.
004A     ;       MAXMSG(AP) = MAXIMUM MAILBOX MESSAGE SIZE.
004A     ;       BUFQUO(AP) = MAXIMUM BYTES FOR MESSAGE BUFFERING.
004A     ;       PROMSK(AP) = PROTECTION MASK FOR CREATED MAILBOX.
004A     ;       ACMODE(AP) = ACCESS MODE CHANNEL IS TO BE ASSIGNED TO.
004A     ;       LOGNAM(AP) = ADDRESS OF LOGICAL NAME STRING DESCRIPTOR.
004A     ;
004A     ;       R4 = CURRENT PROCESS PCB ADDRESS.
004A     ;
004A     ; OUTPUTS:
004A     ;
004A     ;       R0 LOW BIT CLEAR INDICATES FAILURE TO CREATE MAILBOX.
004A     ;
004A     ;               R0 = SS$_ACCVIO - LOGICAL NAME STRING OR LOGICAL NAME STRING
004A     ;                   DESCRIPTOR CANNOT BE READ BY CALLING ACCESS MODE,
004A     ;                   OR CHANNEL NUMBER CANNOT BE WRITTEN BY CALLING
004A     ;                   ACCESS MODE.
004A     ;
004A     ;               R0 = SS$_BADPARAM - BUFQUO exceeds 65535 minus the size of a
004A     ;                   mailbox UCB (approximately 65355).
004A     ;
004A     ;               R0 = SS$_EXBYTLM - Process has insufficient buffer I/O byte
004A     ;                   count quota to allocate mailbox UCB or satisfy buffer
004A     ;                   requirements.
004A     ;
004A     ;               R0 = SS$_INSFMEM - SUFFICIENT SYSTEM DYNAMIC MEMORY DOES NOT
004A     ;                   EXIST TO ALLOCATE MAILBOX UCB.
004A     ;
004A     ;               R0 = SS$_IVLOGNAM - GREATER THAN MAXIMUM LENGTH LOGICAL
004A     ;                   NAME STRING SPECIFIED.
004A     ;
004A     ;               R0 = SS$_NOIOCHAN - NO I/O CHANNEL IS AVAILABLE FOR ASSIGNMENT.
004A     ;
004A     ;               R0 = SS$_NOPRIV - PROCESS DOES NOT HAVE PRIVILEGE TO CREATE
004A     ;                   SPECIFIED TYPE OF MAILBOX.
004A     ;
004A     ;               R0 = SS$_INTERLOCK - UNABLE TO OBTAIN MAILBOX TABLE INTERLOCK.
004A     ;
004A     ;               R0 = SS$_EXPORTQUOTA - PORT'S MAILBOX CREATION QUOTA EXCEEDED.
004A     ;
004A     ;               R0 = SS$_NOSHMBLOCK - NO SHARED MEMORY MAILBOX TABLE BLOCKS
004A     ;                   ARE AVAILABLE.
004A     ;
004A     ;               R0 = SS$_OPINCOMPL - Duplicate unit number encountered while
004A     ;                   linking shared memory mailbox UCB.  (This should never
                        004A  ;                       happen.  Submit an SPR if it does.)
                        004A  ;
                        004A  ;       R0 LOW BIT SET INDICATES SUCCESSFUL COMPLETION.
                        004A  ;
                        004A  ;               R0 = SS$_NORMAL - NORMAL COMPLETION., IF LOGICAL NAME SPECIFIED,
                        004A  ;                       NEW ENTRY DID NOT SUPERSEDE PREVIOUS ENTRY.
                        004A  ;
                        004A  ;               R0 = SS$_SUPERSEDE - NORMAL COMPLETION, NEW ENTRY SUPERSEDED
                        004A  ;                       PREVIOUS ENTRY IN SPECIFIED LOGIC
                        004A  ;-
                        004A
                        004A  BAD_LOGNAM:
    50   0154 8F    3C  004A          MOVZWL  #SS$_IVLOGNAM,R0        ;INVALID LOGICAL NAME
    57     94 AD    D0  004F  EXIT:   MOVL    KRP(FP),R7              ;RETRIEVE ADDRESS OF ALLOCATED KRP
              0B    13  0053          BEQL    10$                     ;IMMEDIATELY RETURN IF NO KRP ALLOCATED
56   00000000'GF    9E  0055          MOVAB   G^CTL$GL_KRPFL,R6       ;ELSE RETRIEVE ADDRESS OF KRP LISTHEAD
     04 B6    67    0E  005C          INSQUE  (R7),@4(R6)             ;AND DEALLOCATE KRP TO LOOKASIDE LIST
                    04  0060  10$:    RET                             ;EXIT SYSTEM SERVICE
                        0061
                  0FFC  0061          .ENTRY  EXE$CREMBX,^M<R2,R3,R4,R5,R6,R7,R8,R9,R10,R11>
    5E     90 AE    9E  0063          MOVAB   LOCALSIZ(SP),SP         ;ALLOCATE LOCAL STORAGE
           94 AD    D4  0067          CLRL    KRP(FP)                 ;NO KRP ALLOCATED
       50     0C    3C  006A          MOVZWL  S^#SS$_ACCVIO,R0        ;ASSUME ACCESS VIOLATION
    56     08 AC    D0  006D          MOVL    CRCHAN(AP),R6           ;GET ADDRESS TO STORE CHANNEL NUMBER
                        0071          IFNOWRT #2,(R6),EXIT            ;CAN CHANNEL NUMBER BE WRITTEN?
    F4 AD     56    D0  0077          MOVL    R6,CHANADDR(FP)         ;SAVE ADDRESS TO STORE CHANNEL NUMBER
              66    B4  007B          CLRW    (R6)                    ;CLEAR CHANNEL NUMBER IN CASE OF ERROR
                        007D  ;
                        007D  ; GET AND TRANSLATE SPECIFIED LOGICAL NAME (IF ANY)
                        007D  ;
    5B     1C AC    D0  007D          MOVL    LOGNAM(AP),R11          ;GET ADDRESS OF LOGICAL NAME STRING DESCRIPT
              56    13  0081          BEQL    20$                     ;IF EQL NO LOGICAL NAME SPECIFIED
                        0083          IFNORD  #8,(R11),EXIT           ;CAN LOGICAL NAME STRING DESCRIPTOR BE READ?
       7E     6B    7D  0089          MOVQ    (R11),-(SP)             ;MOVE DESCRIPTOR TO KERNEL STACK
       59     5E    D0  008C          MOVL    SP,R9                   ;GET ADDRESS OF DESCRIPTOR
    5A     04 A9    D0  008F          MOVL    4(R9),R10               ;GET ADDRESS OF LOGICAL NAME STRING
       5B     69    3C  0093          MOVZWL  (R9),R11                ;GET LENGTH OF STRING IN BYTES
              B2    13  0096          BEQL    BAD_LOGNAM              ;IF EQL STRING HAS ZERO LENGTH
                        0098          ASSUME  LNM$C_NAMLENGTH LE 512
  00FF 8F     5B    B1  0098          CMPW    R11,#<NM$C_NAMLENGTH    ;LOGICAL NAME TOO LARGE?
              AB    1A  009D          BGTRU   BAD_LOGNAM              ;IF GTRU YES
                        009F          IFNORD  R11,(R10),EXIT          ;CAN ENTIRE LOGICAL NAME STRING BE READ?
                        00A5
    5E     F0 AE    9E  00A5          MOVAB   -SHD$C_NAMLENGTH(SP),SP ;CREATE MEMORY NAME BUFFER
              5E    DD  00A9          PUSHL   SP                      ;CREATE BUFFER DESCRIPTOR
              0F    DD  00AB          PUSHL   #SHD$C_NAMLENGTH-1      ;...
       5A     5E    D0  00AD          MOVL    SP,R10                  ;GET ADDRESS OF DESCRIPTOR
                        00B0
56   00000000'GF    9E  00B0          MOVAB   G^CTL$GL_KRPFL,R6       ;RETRIEVE ADDRESS OF KRP QUEUE LISTHEAD
    57     04 B6    0F  00B7          REMQUE  @4(R6),R7               ;RETRIEVE KRP FROM LOOKASIDE LIST
              04    1C  00BB          BVC     10$                     ;CONTINUE IF GOT ONE
                        00BD          BUG_CHECK       KRPEMPTY,FATAL  ;OTHERWISE BUGCHECK
                        00C1
    94 AD     57    D0  00C1  10$:    MOVL    R7,KRP(FP)              ;SAVE ADDRESS OF KRP IN LOCAL STORAGE
              57    DD  00C5          PUSHL   R7                      ;CREATE BUFFER DESCRIPTOR
     000000FF 8F    DD  00C7          PUSHL   #LNM$C_NAMLENGTH        ;...
       5B     5E    D0  00CD          MOVL    SP,R11                  ;GET ADDRESS OF DESCRIPTOR
```

```
            FF2D'   30  00D0             BSBW    MMG$MBXTRNLOG           ;TRANSLATE THE LOGICAL NAME
            03 50   E8  00D3             BLBS    R0,20$                  ;IF LBC FAILURE
             FF76   31  00D6             BRW     EXIT
                        00D9
                        00D9      ;
                        00D9      ; GET A FREE CHANNEL AND CHECK PRIVILEGES
                        00D9      ;
                        00D9
                        00D9      20$:   SETIPL  #IPL$_ASTDEL            ;BLOCK AST DELIVERY
            FF21'   30  00DC             BSBW    IOC$FFCHAN              ;FIND A FREE CHANNEL
            28 50   E9  00DF             BLBC    R0,TO_EXIT_LOWER_IPL    ;IF LBC ERROR
                        00E2             ASSUME  CHANNUM+4 EQ CCBADDR
       EC AD   51   7D  00E2             MOVQ    R1,CHANNUM(FP)          ;SAVE CHANNEL NUMBER AND CCBADDR
       59   04 AC   D0  00E6             MOVL    PRMFLG(AP),R9           ;GET FLAGS ARGUMENT
     50   017C 8F   3C  00EA             MOVZWL  #SS$_IVSTSFLG,R0        ;ASSUME BAD FLAGS SET
59   FFFFFFFE 8F    D3  00EF             BITL    #^C<T>,R9               ;ANY BITS BUT DEFINED BITS SET?
               12   12  00F6             BNEQ    TO_EXIT_LOWER_IPL       ;IF NEQ YES
         50    24   3C  00F8             MOVZWL  #SS$_NOPRIV,R0          ;ASSUME PROCESS DOES NOT HAVE PRIVILEGE
                        00FB             IFPRIV  PRMMBX,DISPATCH         ;CAN PROCESS CREATE PERMANENT MAILBOX?
            06 59   E8  0101             BLBS    R9,TO_EXIT_LOWER_IPL    ;IF LBS NEEDS PRMMBX PRIVILEGE
                        0104             IFPRIV  TMPMBX,DISPATCH         ;CAN PROCESS CREATE TEMPORARY MAILBOX?
                        010A      TO_EXIT_LOWER_IPL:
             009F   31  010A             BRW     EXIT_LOWER_IPL
                        010D
                        010D      ;
                        010D      ; DISPATCH TO APPROPRIATE TYPE MAILBOX CREATION ROUTINE
                        010D      ;
                        010D      DISPATCH:
       00000000'GF  16  010D             JSB     G^SCH$IOLOCKW           ;LOCK I/O DATABASE FOR WRITE
               5B   D5  0113             TSTL    R11                     ;LOGICAL NAME SPECIFIED?
               2C   13  0115             BEQL    NONAME                  ;IF EQL NO - UNNAMED LOCAL MAILBOX
               6A   B5  0117             TSTW    (R10)                   ;MEMORY NAME SPECIFIED?
               46   12  0119             BNEQ    SHAREDMEM               ;IF NEQ YES - SHARED MEMORY MAILBOX
```

```
                          011B  ;
                          011B  ; CREATE NAMED LOCAL MEMORY MAILBOX
                          011B  ;
                          011B  NAMEDLOCAL:
55   00000000'EF     9E   011B          MOVAB   MB$UCB0,R5              ;GET ADDRESS OF TEMPLATE UCB
            0160     30   0122          BSBW    SEARCH_MBXUCB           ;CHECK IF MAILBOX EXISTS
              56     D5   0125          TSTL    R6                      ;MAILBOX FOUND?
              30     12   0127          BNEQ    ASSIGN_IT               ;IF NEQ YES
            00CF     30   0129          BSBW    CHECK_QUOTA             ;CHECK BYTE COUNT QUOTA
           6B 50     E9   012C          BLBC    R0,EXIT_UNLOCKIODB      ;IF LBC INSUFFICIENT QUOTA
            00E0     30   012F          BSBW    CREATE_LOCALUCB         ;CREATE LOCAL MEMORY UCB
           65 50     E9   0132          BLBC    R0,EXIT_UNLOCKIODB      ;IF LBC FAILURE
            0127     30   0135          BSBW    ASSIGN                  ;ASSIGN CHANNEL
            01B7     30   0138          BSBW    CREATE_LOGNAM           ;CREATE LOGICAL NAME
           73 50     E9   013B          BLBC    R0,EXIT_DEASSIGN        ;IF LBC DEASSIGN CHANNEL AND UCB
        50    01     3C   013E          MOVZWL  #SS$_NORMAL,R0          ;SET FINAL STATUS
              57     11   0141          BRB     EXIT_UNLOCKIODB         ;RELEASE CREATE MAILBOX LOCK AND EXIT
```

```
                        0143  ;
                        0143  ; CREATE UNNAMED LOCAL MEMORY MAILBOX
                        0143  ;
                        0143  NONAME:                                        ;CREATE UNNAMED MAILBOX
 55   00000000'EF  9E   0143          MOVAB   MB$UCB0,R5                     ;GET ADDRESS OF TEMPLATE UCB
             00AE  30   014A          BSBW    CHECK_QUOTA                    ;CHECK BYTE COUNT QUOTA
            4A 50  E9   014D          BLBC    R0,EXIT_UNLOCKIODB             ;IF LBC INSUFFICIENT QUOTA
             00BF  30   0150          BSBW    CREATE_LOCALUCB                ;CREATE A LOCAL MAILBOX UCB
            44 50  E9   0153          BLBC    R0,EXIT_UNLOCKIODB             ;IF LBC FAILURE
            74 A6  D4   0156          CLRL    UCB$L_LOGADR(R6)               ;SET NO NAME FOR MAILBOX
                        0159  ASSIGN_IT:                                     ;
             0103  30   0159          BSBW    ASSIGN                         ;ASSIGN CHANNEL AND EXIT
        50     01  3C   015C          MOVZWL  #SS$_NORMAL,R0                 ;SET FINAL STATUS
               39  11   015F          BRB     EXIT_UNLOCKIODB                ;RELEASE CREATE MAILBOX LOCK AND EXIT
```

```
                       0161     426 ;
                       0161     427 ; CREATE SHARED MEMORY MAILBOX
                       0161     428 ;
                       0161     429 SHAREDMEM:
 50     24       3C    0161     430          MOVZWL  S^#SS$_NOPRIV,R0           ;ASSUME NO PRIVILEGE
                       0164     431          IFNPRIV SHMEM,EXIT_UNLOCKIODB      ;PROCESS CREATE SHARED MEMORY OBJECTS?
      0247       30    016A     432          BSBW    GET_SHMTEMPLATE            ;GET ADDRESS OF TEMPLATE UCB
     2A 50       E9    016D     433          BLBC    R0,EXIT_UNLOCKIODB         ;IF LBC FAILURE
      0112       30    0170     434          BSBW    SEARCH_MBXUCB              ;CHECK IF MAILBOX EXISTS
        56       D5    0173     435          TSTL    R6                         ;MAILBOX FOUND?
        E2       12    0175     436          BNEQ    ASSIGN_IT                  ;IF NEQ YES
      0081       30    0177     437          BSBW    CHECK_QUOTA                ;CHECK BYTE COUNT QUOTA
     1D 50       E9    017A     438          BLBC    R0,EXIT_UNLOCKIODB         ;IF LBC FAILURE
      028B       30    017D     439          BSBW    SEARCH_SHMMBX              ;SEARCH SHARED MEMORY FOR MAILBOX
                       0180     440                                             ; AND CREATE IT IF NOT FOUND
     17 50       E9    0180     441          BLBC    R0,EXIT_UNLOCKIODB         ;IF LBC FAILURE
      039A       30    0183     442          BSBW    CREATE_SHMUCB              ;CREATE SHARED MEMORY UCB
     0E 50       E9    0186     443          BLBC    R0,EXIT_CLEAR_FLAGS        ;IF LBC FAILURE
      00D3       30    0189     444          BSBW    ASSIGN                     ;ASSIGN CHANNEL
      0163       30    018C     445          BSBW    CREATE_LOGNAM              ;CREATE LOGICAL NAME
     1F 50       E9    018F     446          BLBC    R0,EXIT_DEASSIGN           ;IF LBC DEASSIGN CHANNEL AND UCB
 50     01       3C    0192     447          MOVZWL  #SS$_NORMAL,R0             ;SET FINAL STATUS
        03       11    0195     448          BRB     EXIT_UNLOCKIODB            ;RELEASE CREATE MAILBOX LOCK AND EXIT
```

```
                            0197  EXIT_CLEAR_FLAGS:
                 08 A7   94 0197          CLRB    MBX$B_FLAGS(R7)         ;CATASTROPHIC ERROR - CLEAR ALL FLAGS
                            019A  EXIT_UNLOCKIODB:                        ;EXIT, UNLOCKING I/O DATABASE
                    50   DD 019A          PUSHL   R0                      ;SAVE STATUS
        54 00000000'GF   D0 019C          MOVL    G^CTL$GL_PCB,R4         ;LOAD PCB INTO R4 FOR CALL
           00000000'GF   16 01A3          JSB     G^SCH$IOUNLOCK          ;UNLOCK IO DATA BASE
                 50 8ED0    01A9          POPL    R0                      ;RESTORE STATUS
                            01AC  EXIT_LOWER_IPL:                         ;EXIT, UNLOCKING WITH STATUS IN R0
                            01AC          SETIPL  #0                      ;RESTORE IPL IF NEEDED
                    34   11 01AF          BRB     RETURN                  ;GO EXIT SYSTEM SERVICE
                            01B1
                            01B1  EXIT_DEASSIGN:
                    50   DD 01B1          PUSHL   R0                      ;SAVE SYSTEM SERVICE STATUS
        54 00000000'GF   D0 01B3          MOVL    G^CTL$GL_PCB,R4         ;LOAD PCB INTO R4 FOR CALL
           00000000'GF   16 01BA          JSB     G^SCH$IOUNLOCK          ;UNLOCK IO DATA BASE
                            01C0          SETIPL  #0                      ;RESTORE IPL IF NEEDED
                 0E 59   E9 01C3          BLBC    R9, 10$                 ;TEMP MAILBOX DO NOT DELETE
                            01C6          $DELMBX_S -                     ;MARK PERMANENT MAILBOX FOR DELETE
                            01C6                  CHAN  = CHANNUM(FP)     ;
                 23 50   E9 01D1          BLBC    R0,BUGCHECK             ;SHOULD ALWAYS BE ABLE TO DELETE
                            01D4  10$:    $DASSGN_S -                     ;DELETE ASSIGNED CHANNEL, MAILBOX
                            01D4                  CHAN  = CHANNUM(FP)     ; AND UCB
                 15 50   E9 01DF          BLBC    R0,BUGCHECK             ;SHOULD ALWAYS BE ABLE TO DEASSIGN
                 50 8ED0    01E2          POPL    R0                      ;RESTORE STATUS
                            01E5
              57    94 AD D0 01E5  RETURN: MOVL   KRP(FP),R7              ;RETRIEVE ADDRESS OF ALLOCATED KRP
                    0B   13 01E9          BEQL    10$                     ;IMMEDIATELY RETURN IF NO KRP ALLOCATED
        56 00000000'GF   9E 01EB          MOVAB   G^CTL$GL_KRPFL,R6       ;ELSE RETRIEVE ADDRESS OF KRP LISTHEAD
             04 B6    67  0E 01F2          INSQUE  (R7),@4(R6)             ;AND DEALLOCATE KRP TO LOOKASIDE LIST
                         04 01F6  10$:    RET                             ;EXIT SYSTEM SERVICE
                            01F7
                            01F7  BUGCHECK:
                            01F7          BUG_CHECK SSRVEXCEPT,FATAL
```

```
                          01FB            .SBTTL  CHECK_QUOTA - CHECK BUFFER I/O BYTE COUNT QUOTA
                          01FB      ;++
                          01FB      ; THIS ROUTINE IS CALLED TO CHECK IF PROCESS HAS ENOUGH BUFFER I/O
                          01FB      ; BYTE COUNT QUOTA FOR TEMPORARY MAILBOX UCB AND MESSAGES.
                          01FB      ;
                          01FB      ; INPUTS:
                          01FB      ;
                          01FB      ;       BUFQUO(AP) = SPECIFIED MAILBOX BUFFER QUOTA FOR MESSAGES.
                          01FB      ;       R4 = CURRENT PROCESS PCB ADDRESS
                          01FB      ;       R9 = 0 IF TEMPORARY MAILBOX.
                          01FB      ;          = 1 IF PERMANENT MAILBOX.
                          01FB      ;
                          01FB      ; OUTPUTS:
                          01FB      ;
                          01FB      ;       R0 = SUCCESS OR FAILURE.
                          01FB      ;       R8 = BYTE COUNT QUOTA FOR MESSAGES.
                          01FB      ;--
                          01FB      CHECK_QUOTA:                                    ;CHECK BUFFER QUOTA
             50   01  3C  01FB              MOVZWL  S^#SS$_NORMAL,R0                ;ASSUME SUCCESS
          58   10 AC  3C  01FE              MOVZWL  BUFQUO(AP),R8                   ;GET SPECIFIED BUFFER QUOTA
                  07  12  0202              BNEQ    10$                             ;IF NEQ BUFER QUOTA SPECIFIED
  58   00000000'EF    3C  0204              MOVZWL  IOC$GW_MBXBFQUO,R8              ;GET DEFAULT BUFFER QUOTA
               03 59  E8  020B      10$:    BLBS    R9, 30$                         ; Branch if permanent mailbox.
               FDEF'  30  020E              BSBW    IOC$CHKMBXQUOTA                 ; Check for enough BYTCNT quota.
                      05  0211      30$:    RSB                                     ; Return to caller.
```

```
                         0212           .SBTTL  CREATE_LOCALUCB - CREATE LOCAL MEMORY MAILBOX UCB
                         0212   ;++
                         0212   ; THIS ROUTINE IS CALLED TO CREATE A UCB FOR A LOCAL MEMORY MAILBOX.
                         0212   ;
                         0212   ; INPUTS:
                         0212   ;
                         0212   ;       R4 = CURRENT PROCESS PCB ADDRESS
                         0212   ;       R5 = ADDRESS OF TEMPLATE MAILBOX UCB.
                         0212   ;       R8 = BYTE COUNT QUOTA FOR MESSAGES.
                         0212   ;
                         0212   ; OUTPUTS:
                         0212   ;
                         0212   ;       R0 = SUCCESS OR FAILURE.
                         0212   ;       R6 = ADDRESS OF CREATED UCB.
                         0212   ;--
                         0212           .ENABL  LSB
                         0212   CREATE_LOCALUCB:                        ;CREATE LOCAL MEMORY MAILBOX UCB
              FDEB'  30  0212           BSBW    IOC$CLONE_UCB           ;CLONE MAILBOX UCB
              46 50  E9  0215           BLBC    R0,20$                  ;IF LBC ALLOCATION FAILURE
           56    52  D0  0218           MOVL    R2,R6                   ;SAVE ADDRESS OF ALLOCATED UCB
         18 A6   58  B0  021B           MOVW    R8,UCB$W_BUFQUO(R6)     ;SET BUFFER QUOTA
              2C A6  D4  021F           CLRL    UCB$L_PID(R6)           ;CLEAR OWNER PID
         50   1C A6  D0  0222           MOVL    UCB$L_ORB(R6),R0        ;GET ORB ADDRESS
         0B A0   01  88  0226           BISB2   #ORB$M_PROT_16,ORB$B_FLAGS(R0)  ;USE SOGW PROTECTION WORD
      18 A0   14 AC  B0  022A           MOVW    PROMSK(AP),ORB$W_PROT(R0) ;SET PROTECTION MASK
       60   00BC C4  D0  022F           MOVL    PCB$L_UIC(R4), ORB$L_OWNER(R0) ;Establish owner UIC.
      42 A6   0C AC  B0  0234           MOVW    MAXMSG(AP),UCB$W_DEVBUFSIZ(R6) ;SET MAXIMUM MESSAGE SIZE
                 08  12  0239           BNEQ    COMP_CHARGES            ;IF NEQ MESSAGE SIZE SPECIFIED
42 A6   00000000'EF  B0  023B           MOVW    IOC$GW_MBXMXMSG,UCB$W_DEVBUFSIZ(R6) ;SET DEFAULT MAXIMUM SIZE
                         0243   ;
                         0243   ; COMPUTE BYTE COUNT CHARGES
                         0243   ;
                         0243   COMP_CHARGES:
              44 A6  D4  0243           CLRL    UCB$L_DEVDEPEND(R6)     ;INIT USER VISABLE CURRENT MESSAGE COUNT
 56 A6   08 A6   58  A1  0246           ADDW3   R8, UCB$W_SIZE(R6), -   ;Compute byte count charge for
                         024C                   UCB$W_CHARGE(R6)        ;this UCB.
              05 59  E9  024C           BLBC    R9, 10$                 ;Branch if temporary mailbox.
      07 68 A6   00  E3  024F           BBCS    #UCB$V_PRMMBX, -        ;Mark mailbox permanent and
                         0254                   UCB$W_DEVSTS(R6), 19$   ;skip temporary mailbox stuff.
         68 A6   02  A8  0254   10$:    BISW    #UCB$M_DELMBX, -        ;Mark mailbox temporary (deletable).
                         0258                   UCB$W_DEVSTS(R6)
              FDA5'  30  0258           BSBW    IOC$DEBIT_UCB           ;Charge byte count quota for UCB, etc.
            50   01  3C  025B   19$:    MOVZWL  #SS$_NORMAL, R0         ;Set operation completed successfully.
                     05  025E   20$:    RSB                             ;Return to caller.
                         025F           .DSABL  LSB
```

```
                                025F   559          .SBTTL  ASSIGN - ASSIGN CHANNEL TO MAILBOX
                                025F   560 ;++
                                025F   561 ; THIS ROUTINE IS CALLED TO ASSIGN A CHANNEL TO THE MAILBOX
                                025F   562 ;
                                025F   563 ; INPUTS:
                                025F   564 ;
                                025F   565 ;        CHANADDR(FP) = ADDRESS TO STORE CHANNEL NUMBER.
                                025F   566 ;        CHANNUM(FP) = CHANNEL NUMBER.
                                025F   567 ;        CCBADDR(FP) = ADDRESS OF CCB FOR CHANNEL NUMBER.
                                025F   568 ;        R6 = ADDRESS OF UCB
                                025F   569 ;
                                025F   570 ; OUTPUTS:
                                025F   571 ;
                                025F   572 ;        CHANNEL NUMBER STORED AT CALLER'S ADDRESS.
                                025F   573 ;        I/O DATABASE UNLOCKED AND IPL SET TO 0
                                025F   574 ;--
                                025F   575 ASSIGN:                                       ;ASSIGN CHANNEL
                                025F   576          ASSUME  CHANNUM+4 EQ CCBADDR
                                025F   577          ASSUME  CCBADDR+4 EQ CHANADDR
                          5B    DD  025F   578          PUSHL   R11                     ;SAVE ADDRESS OF LOGNAME DESCRIPTOR
                 5A    F0 AD    7D  0261   579          MOVQ    CCBADDR(FP),R10         ;RETRIEVE CCB AND CHANNEL ADDRESSES
                 6B    EC AD    B0  0265   580          MOVW    CHANNUM(FP),(R11)       ;STORE CHANNEL NUMBER
                     6A   56    D0  0269   581          MOVL    R6,CCB$L_UCB(R10)       ;STORE UCB ADDRESS IN CCB
    50    18 AC    02     00    EF  026C   582          EXTZV   #0,#2,ACMODE(AP),R0     ;GET SPECIFIED ACCESS MODE
                      FD8B'     30  0272   583          BSBW    EXE$MAXACMODE           ;MAXIMIZE ACCESS MODE
          09 AA    50     01    81  0275   584          ADDB3   #1,R0,CCB$B_AMOD(R10)   ;STORE ASSIGN ACCESS MODE
            54   00000000'GF    D0  027A   585          MOVL    G^CTL$GL_PCB,R4         ;LOAD PCB INTO R4 FOR CALL
                          5B  8ED0  0281   586          POPL    R11                     ;RESTORE ADDRESS OF LOGNAME DESCRIPTOR
                                05  0284   587          RSB
```

```
                                   0285          .SBTTL  SEARCH_MBXUCB - SEARCH FOR SPECIFIED MAILBOX UCB
                                   0285    ;++
                                   0285    ; THIS ROUTINE IS CALLED TO SEARCH FOR THE SPECIFIED MAILBOX UCB.
                                   0285    ; IF IT IS FOUND, THE MAILBOX ALREADY EXISTS AND THE UCB ADDRESS IS
                                   0285    ; RETURNED.  IF IT IS NOT FOUND, THE MAILBOX MUST BE CREATED.
                                   0285    ;
                                   0285    ; INPUTS:
                                   0285    ;
                                   0285          R9 = 0 IF MAILBOX IS TEMPORARY
                                   0285               1 IF MAILBOX IS PERMANENT
                                   0285          R11 = ADDRESS OF MAILBOX NAME DESCRIPTOR
                                   0285    ;
                                   0285    ; OUTPUTS:
                                   0285    ;
                                   0285          IF MAILBOX UCB FOUND.
                                   0285                  R6 = ADDRESS OF MAILBOX UCB.
                                   0285    ;
                                   0285          IF MAILBOX UCB NOT FOUND.
                                   0285                  R6 = 0
                                   0285    ;
                                   0285    ;--
                                   0285    SEARCH_MBXUCB:                          ;SEARCH FOR MAILBOX UCB
                                   0285
                                   0285          ; SET UP THE TRANSLATION INDEX TO BE THE BACK POINTER
                                   0285          ;
    98 AD   00010004 8F     D0     0285          MOVL    #LNM$_INDEX@16!4,I1_CODE_LEN(FP)
       9C AD    FD6F CF     DE     028D          MOVAL   BACKPTR_INDEX,I1_BUF_ADDR(FP)
                  A0 AD     D4     0293          CLRL    I1_RET_LEN(FP)
                                   0296
                                   0296          ; GET THE TRANSLATION STRING AND SIZE ( OR HOPEFULLY THE UCB MAILBOX )
                                   0296          ;
    A4 AD   00020004 8F     D0     0296          MOVL    #LNM$_STRING@16!4,I2_CODE_LEN(FP)
          A8 AD    E4 AD    DE     029E          MOVAL   UCB_ADDRESS(FP),I2_BUF_ADDR(FP)
          AC AD    B4 AD    DE     02A3          MOVAL   UCB_ADDR_SIZE(FP),I2_RET_LEN(FP)
                                   02A8
                  B0 AD     D4     02A8          CLRL    TRNLNM_END_LIST(FP)     ;END ITEM LIST
                                   02AB
   50    18 AC    02   00   EF     02AB          EXTZV   #0,#2,ACMODE(AP),R0     ;EXTRACT SPECIFIED MODE
                   FD4C'    30     02B1          BSBW    EXE$MAXACMODE           ;MAXIMIZE ACCESS MODE
                      50    DD     02B4          PUSHL   R0                      ;SAVE IT FOR REFERENCE
                 50   5E    D0     02B6          MOVL    SP,R0                   ;SAVE POINTER TO ACCESS MODE
        51   FD4A CF49      D0     02B9          MOVL    MAILBOX_TABLES[R9],R1   ;CALCULATE TABLE
                                   02BF
                                   02BF          $TRNLNM_S -                             ;SEE IF THE MAILBOX EXIST
                                   02BF                  TABNAM = (R1),-                 ;USE THE CORRECT TABLE
                                   02BF                  LOGNAM = (R11),-                ;POINT TO DESCRIPTOR
                                   02BF                  ACMODE = (R0),-                 ;PREVIOUS ACCESS MODES
                                   02BF                  ITMLST = I1_CODE_LEN(FP)        ;ITEM LIST
                                   02D1
                 5E   04    C0     02D1          ADDL2   #4,SP                   ;POP ACCESS MODE
                   18 50    E9     02D4          BLBC    R0,20$                  ;IF ERROR NO TRANSLATION
             B4 AD    04    B1     02D7          CMPW    S^#4,UCB_ADDR_SIZE(FP)  ;TEST IF IT IS THE CORRECT SIZE
                      12    12     02DB          BNEQ    20$                     ;CAN NOT BE POINTER, SHOULD NOT HAPPEN
             56    E4 AD    D0     02DD          MOVL    UCB_ADDRESS(FP),R6      ;LOAD THE ?UCB ADDRESS INTO R6
             56    8C A6    DE     02E1          MOVAL   -UCB$L_LOGADR(R6),R6    ;POINT TO THE BEGINNING OF ?UCB
             0A A6    10    91     02E5          CMPB    #DYN$C_UCB,UCB$B_TYPE(R6);CHECK TO SEE WHERE THE POINTER POINTS
                      04    12     02E9          BNEQ    20$                     ;CHECK FAILED ( MUST BE MTL )
                      5C A6   B6  02EB    647         INCW    UCB$W_REFC(R6)          ;INCREMENT REFERENCE COUNT
                              05  02EE    648         RSB                             ;RETURN
                         56   D4  02EF    649 20$:    CLRL    R6                      ;NO UCB FOUND
                              05  02F1    650         RSB                             ;RETURN
```

```
                               02F2          .SBTTL  CREATE_LOGNAM - CREATE LOGICAL NAME
                               02F2     ;++
                               02F2     ; THIS ROUTINE IS CALLED TO FILL-IN THE ALLOCATED LOGICAL NAME BLOCK
                               02F2     ; AND INSERT IT INTO THE APPROPRIATE LOGICAL NAME TABLE.
                               02F2     ;
                               02F2     ; INPUTS:
                               02F2     ;
                               02F2     ;       R6 = ADDRESS OF MAILBOX UCB.
                               02F2     ;       R9 = 0 IF TMPMBX, 1 IF PRMMBX.
                               02F2     ;       R11 = ADDRESS OF LOGICAL NAME DESCRIPTOR.
                               02F2     ;
                               02F2     ; OUTPUTS:
                               02F2     ;
                               02F2     ;       R0 = INSERTION STATUS.
                               02F2     ;
                               02F2     ;       R1, R2, R4, and R5 are corrupted.
                               02F2     ;
                               02F2     ;--
                               02F2    CREATE_LOGNAM:                                   ;CREATE LOGICAL NAME
                               02F2
                               02F2             ; STORE INFORMATION SO IT IS ACCESSABLE FROM ITEM LIST
                               02F2             ;
     E4 AD    74 A6        DE  02F2             MOVAL   UCB$L_LOGADR(R6),UCB_ADDRESS(FP)
                               02F7                                                     ;MOVE ADDR OF UCB TO LOCAL STORAGE
              F0 AD        D4  02F7             CLRL    ZERO_INDEX(FP)                  ;MOVE ZERO INDEX TO LOCAL STORAGE
        5E    EF AE        DE  02FA             MOVAL   -MBXNAMSIZ(SP),SP               ;STORAGE FOR EQUIVALENCE NAME
     D8 AD       6E        DE  02FE             MOVAL   (SP),I6_BUF_ADDR(FP)            ;SAVE THE POINTER TO THE BUFFER
        55       56        D0  0302             MOVL    R6,R5                           ;MOVE UCB ADDRESS TO R5 FOR SUB-CALL
                               0305             DSBINT  100$                            ;LOCK THIS CODE FOR HIGH IPL EXECUTION.
        51    D8 AD        D0  030F             MOVL    I6_BUF_ADDR(FP),R1              ;LOCATION OF OUTPUT BUFFER
        50       11        D0  0313             MOVL    #MBXNAMSIZ,R0                   ;SIZE OF OUTPUT BUFFER
        54       01        CE  0316             MNEGL   #1,R4                           ;SET FOR DISPLAY FORMAT DEVICE NAME
              FCE4'        30  0319             BSBW    IOC$CVT_DEVNAM                  ;CONVERT UCB TO DEVICE NAME STRING
     D4 AD       51        9A  031C             MOVZBL  R1,I6_CODE_LEN(FP)              ;SAVE SIZE OF OUTPUT BUFFER
                               0320             ENBINT                                  ;RETURN OLD IPL
                               0323
                               0323             ; STUFF THE LOGICAL NAME BLOCK ADDRESS IN THE UCB
                               0323             ;
98 AD    00090004 8F       D0  0323             MOVL    #LNM$_LNMB_ADDR@16!4,I1_CODE_LEN(FP)
      9C AD    74 A6       DE  032B             MOVAL   UCB$L_LOGADR(R6),I1_BUF_ADDR(FP)
               A0 AD       D4  0330             CLRL    I1_RET_LEN(FP)
                               0333
                               0333             ; SET UP THE BACK POINTER TO THE UCB FROM THE LOGICAL NAME BLOCK
                               0333             ;
A4 AD    00010004 8F       D0  0333             MOVL    #LNM$_INDEX@16!4,I2_CODE_LEN(FP)
    A8 AD    FCC1 CF       DE  033B             MOVAL   BACKPTR_INDEX,I2_BUF_ADDR(FP)
               AC AD       D4  0341             CLRL    I2_RET_LEN(FP)
B0 AD    00020004 8F       D0  0344             MOVL    #LNM$_STRING@16!4,I3_CODE_LEN(FP)
      B4 AD    E4 AD       DE  034C             MOVAL   UCB_ADDRESS(FP),I3_BUF_ADDR(FP)
               B8 AD       D4  0351             CLRL    I3_RET_LEN(FP)
                               0354
                               0354             ; SET UP NORMAL EQUIVALENCE STRING
                               0354             ;
BC AD    00010004 8F       D0  0354             MOVL    #LNM$_INDEX@16!4,I4_CODE_LEN(FP)
      C0 AD    F0 AD       DE  035C             MOVAL   ZERO_INDEX(FP),I4_BUF_ADDR(FP)
               C4 AD       D4  0361             CLRL    I4_RET_LEN(FP)
C8 AD    00030004 8F       D0  0364             MOVL    #LNM$_ATTRIBUTES@16!4,I5_CODE_LEN(FP)
```

```
        CC AD   FC94 CF    DE  036C            MOVAL   LNM_TERM_ATTR,I5_BUF_ADDR(FP)
                  D0 AD    D4  0372            CLRL    I5_RET_LEN(FP)
                  D4 AD    B7  0375            DECW    I6_CODE_LEN(FP)         ;REMOVE LEADING "_" FROM COUNT
           D6 AD     02    B0  0378            MOVW    #LNM$_STRING,I6_CODE_LEN+2(FP)
                  D8 AD    D6  037C            INCL    I6_BUF_ADDR(FP)         ;REMOVE LEADING "_" FROM STRING
                  DC AD    D4  037F            CLRL    I6_RET_LEN(FP)
                               0382
                  E0 AD    D4  0382            CLRL    CRELNM_END_LIST(FP)     ;END ITEM LIST
                               0385
  50    18 AC    02   00   EF  0385            EXTZV   #0,#2,ACMODE(AP),R0     ;EXTRACT SPECIFIED MODE
                    FC72'  30  038B            BSBW    EXE$MAXACMODE           ;MAXIMIZE ACCESS MODE
                      50   DD  038E            PUSHL   R0                      ;SAVE IT FOR REFERENCE
                               0390
              50     6E    DE  0390            MOVAL   (SP),R0                 ;POINTER TO ACCESS MODE
         51    FC70 CF49   D0  0393            MOVL    MAILBOX_TABLES[R9],R1   ;CALCULATE TABLE
                               0399
                               0399            $CRELNM_S -
                               0399                    TABNAM  = (R1),-
                               0399                    LOGNAM  = (R11),-
                               0399                    ACMODE  = (R0),-
                               0399                    ITMLST  = I1_CODE_LEN(FP)
                               03AB
           5E    15 AE     DE  03AB            MOVAL   MBXNAMSIZ+4(SP),SP      ;RETURN STORAGE OF EQUIVALENCE NAME
                               03AF
                           05  03AF            RSB
                               03B0  100$:                                     ;IPL TO LOCK PREVIOUS CODE IN MEMORY
                     00000008  03B0            .LONG   IPL$_SYNCH
                               03B4            ASSUME  <.-CREATE_LOGNAM> LE 512 ;CAN ONLY LOCK 1 PAGE
```

```
                              03B4             .SBTTL  GET_SHMTEMPLATE - GET SHARED MEMORY MAILBOX TEMPLATE UCB
                              03B4    ;++
                              03B4    ;
                              03B4    ; THIS ROUTINE IS CALLED TO GET THE ADDRESS OF THE "TEMPLATE" MAILBOX
                              03B4    ; UCB FOR A SPECIFIED SHARED MEMORY.
                              03B4    ;
                              03B4    ; INPUTS:
                              03B4    ;
                              03B4    ;        R10 = ADDRESS OF SHARED MEMORY NAME DESCRIPTOR.
                              03B4    ;
                              03B4    ; OUTPUTS:
                              03B4    ;
                              03B4    ;        R0 = SUCCESS OR FAILURE.
                              03B4    ;        R5 = ADDRESS OF TEMPLATE MAILBOX UCB.
                              03B4    ;        SHBADDR(FP) = ADDRESS OF SHB.
                              03B4    ;--
                              03B4   GET_SHMTEMPLATE:                                ;GET ADDRESS OF TEMPLATE UCB
 56    00000000'EF       DE   03B4             MOVAL    EXE$GL_SHBLIST,R6          ;GET ADDRESS OF SHARED MEMORY LISTHEAD
                              03BB   10$:
                              03BB             ASSUME   SHB$L_LINK EQ 0
           56    66      D0   03BB             MOVL     SHB$L_LINK(R6),R6          ;GET ADDRESS OF NEXT SHB
                 45      13   03BE             BEQL     40$                        ;IF EQL NO SUCH MEMORY
       55    04  A6      D0   03C0             MOVL     SHB$L_DATAPAGE(R6),R5      ;GET ADDRESS OF DATAPAGE
 00    04 BA     6A      2D   03C4             CMPC5    (R10),@4(R10),#0,-         ;NAMES MATCH?
       21 A5     0F           03C9                      #SHD$C_NAMLENGTH-1,SHD$T_NAME+1(R5)
                 ED      12   03CC             BNEQ     10$                        ;IF NEQ NO
    32 0B A6     00      E1   03CE             BBC      #SHB$V_CONNECT,SHB$B_FLAGS(R6),40$ ;BR IF NOT CONNECTED
 50    00000000'EF       DE   03D3             MOVAL    IOC$GL_DEVLIST,R0          ;GET ADDRESS OF DDB LISTHEAD
                              03DA   20$:
                              03DA             ASSUME   DDB$L_LINK EQ 0
           50    60      D0   03DA             MOVL     DDB$L_LINK(R0),R0          ;GET ADDRESS OF NEXT DDB
                 26      13   03DD             BEQL     40$                        ;IF EQL NOT FOUND
       55    04  A0      D0   03DF             MOVL     DDB$L_UCB(R0),R5           ;GET ADDRESS OF UCB
                 F5      13   03E3             BEQL     20$                        ;BRANCH IF NO UCB ADDRESS HERE
    F0 38 A5     14      E1   03E5             BBC      #DEV$V_MBX,UCB$L_DEVCHAR(R5),20$ ;IF CLR NOT MAILBOX
    EB 68 A5     03      E1   03EA             BBC      #UCB$V_SHMMBX,UCB$W_DEVSTS(R5),20$ ;IF CLR NOT SHARED MEM MAILBOX
       51    24  A5      D0   03EF             MOVL     UCB$L_CRB(R5),R1           ;GET CRB ADDRESS
       51    38  A1      D0   03F3             MOVL     CRB$L_INTD+VEC$L_ADP(R1),R1 ;GET ADAPTER CONTROL BLOCK ADDRESS
       56    30  A1      D1   03F7             CMPL     ADP$L_SHB(R1),R6           ;SHARED MEMORY BLOCK ADDRESS MATCH?
                 DD      12   03FB             BNEQ     20$                        ;IF NEQ NO - WRONG MEMORY
       E8 AD     56      D0   03FD             MOVL     R6,SHBADDR(FP)             ;SAVE SHB ADDRESS
           50    01      D0   0401             MOVL     #1,R0                      ;SET SUCCESS
                         05   0404             RSB                                 ;
                              0405   40$:
   50    037C 8F         3C   0405             MOVZWL   #SS$_SHMNOTCNCT,R0         ;SET MEMORY NOT CONNECTED
                         05   040A             RSB                                 ;
```

```
                                040B   788              .SBTTL  SEARCH_SHMMBX - SEARCH FOR SHARED MEMORY MAILBOX
                                040B   789 ;++
                                040B   790 ; THIS ROUTINE IS CALLED TO SEARCH THE SPECIFIED SHARED MEMORY MAILBOX
                                040B   791 ; TABLE TO SEE IF THE SPECIFIED MAILBOX ALREADY EXISTS.
                                040B   792 ;
                                040B   793 ; INPUTS:
                                040B   794 ;
                                040B   795 ;        R11 = ADDRESS OF MAILBOX NAME DESCRIPTOR.
                                040B   796 ;        SHBADDR(FP) = ADDRESS OF SHARED MEMORY CONTROL BLOCK.
                                040B   797 ;
                                040B   798 ; OUTPUTS:
                                040B   799 ;
                                040B   800 ;        R0 = SUCCESS IF FOUND
                                040B   801 ;
                                040B   802 ;                R7 = ADDRESS OF MAILBOX CONTROL BLOCK.
                                040B   803 ;
                                040B   804 ;                EXITS TO UNLOCK_SHMMBX.
                                040B   805 ;
                                040B   806 ;        ELSE, NOT FOUND AND
                                040B   807 ;
                                040B   808 ;                R7 = ADDRESS OF FIRST UNALLOCATED MAILBOX BLOCK.
                                040B   809 ,
                                040B   810 ;                EXITS TO CREATE_SHMMBX
                                040B   811 ;
                                040B   812 ;        R6 = ADDRESS OF SHARED MEMORY CONTROL BLOCK.
                                040B   813 ;
                                040B   814 ;        MAILBOX TABLE MUTEX OWNED, MAILBOX TABLE LOCK OWNED, IPL = 31.
                                040B   815 ;--
                                040B   816 SEARCH_SHMMBX:                               ;SEARCH SHARED MEMORY MAILBOXES
    50    00000000'EF     DE    040B   817              MOVAL   EXE$GL_SHMMBMTX,R0          ;GET ADDRESS OF MAILBOX MUTEX
          00000000'GF     16    0412   818              JSB     G^SCH$LOCKW                 ;LOCK SHARED MEMORY MAILBOX MUTEX
                                0418   819              SETIPL  SHMMBX_CODE                 ;LOCK CODE IN MEMORY
       56     E8 AD       D0    041F   820              MOVL    SHBADDR(FP),R6              ;GET ADDRESS OF SHB
       51     04 A6       D0    0423   821              MOVL    SHB$L_DATAPAGE(R6),R1       ;GET ADDRESS OF MEMORY DATAPAGE
                                0427   822              LOCK    #SHD$V_MBXLCK,SHD$B_FLAGS(R1) ;LOCK MAILBOX TABLE
            08 50         E8    0445   823              BLBS    R0,10$                      ;CONTINUE IF LOCK ATTEMPT SUCCEEDED
    50    038C 8F         3C    0448   824              MOVZWL  #SS$_INTERLOCK,R0           ;STORE FINAL STATUS
             00B6         31    044D   825              BRW     UNLOCK_SHMMBX_ERROR         ; AND EXIT THROUGH COMMON CODE
                                0450   826
      57   51   61        C1    0450   827 10$:         ADDL3   SHD$L_MBXPTR(R1),R1,R7      ;GET ADDRESS OF FIRST MAILBOX
        7E    1A A1       3C    0454   828              MOVZWL  SHD$W_MBXMAX(R1),-(SP)      ;GET NUMBER OF MAILBOXES
                 7E       D4    0458   829              CLRL    -(SP)                       ;SET NO UNALLOCATED ENTRY YET
                                045A   830 20$:
     20 08 A7    01       E1    045A   831              BBC     #MBX$V_VALID,MBX$B_FLAGS(R7),30$ ;IF CLR MAILBOX NOT VALID
        50    20 A7       9A    045F   832              MOVZBL  MBX$T_NAME(R7),R0           ;GET MAILBOX NAME SIZE
  00     04 BB   6B       2D    0463   833              CMPC5   (R11),@4(R11),#0,-          ;MAILBOX NAME MATCH ONE SPECIFIED?
         21 A7   50             0468   834                      R0,MBX$T_NAME+1(R7)         ;
                 19       12    046B   835              BNEQ    40$                         ;IF NEQ NO - TRY NEXT ONE
            5E   08       C0    046D   836              ADDL    #8,SP                       ;REMOVE MAILBOX COUNT + POINTER
        50    15 A6       9A    0470   837              MOVZBL  SHB$B_PORT(R6),R0           ;GET PORT NUMBER
     00 0C A7    50       E6    0474   838              BBSSI   R0,MBX$W_REF(R7),35$        ;SET PORT'S REFERENCE FLAG
                                0479   839 35$:                                         ;
            50   01       D0    0479   840              MOVL    S^#SS$_NORMAL,R0            ;SET SUCCESS
               007A       31    047C   841              BRW     UNLOCK_SHMMBX               ;
                                047F   842 30$:
                 6E       D5    047F   843              TSTL    (SP)                        ;FIND AN UNALLOCATED ENTRY YET?
                 03       12    0481   844              BNEQ    40$                         ;IF NEQ YES
  6E   57    D0  0483    845        MOVL    R7,(SP)                ;SAVE ADDRESS OF UNALLOCATED ENTRY
                 0486    846 40$:
  57   30    C0  0486    847        ADDL    #MBX$C_LENGTH,R7       ;INCREMENT MAILBOX POINTER
CD 04  AE    F5  0489    848        SOBGTR  4(SP),20$              ;DECREMENT MAILBOX COUNTER AND LOOP
       57  8ED0  048D    849        POPL    R7                     ;GET ADDRESS OF UNALLOCATED ENTRY
  5E   04    C0  0490    850        ADDL    #4,SP                  ;REMOVE MAILBOX COUNT FROM STACK
                 0493    851                                       ;CONTINUE IN CREATE_SHMMBX
```

```
                                0493                .SBTTL  CREATE_SHMMBX - CREATE SHARED MEMORY MAILBOX BLOCK
                                0493      ;++
                                0493      ; THIS ROUTINE IS CALLED TO CREATE A SHARED MEMORY MAILBOX CONTROL BLOCK.
                                0493      ;
                                0493      ; INPUTS:
                                0493      ;
                                0493      ;         R6 = ADDRESS OF SHARED MEMORY CONTROL BLOCK.
                                0493      ;         R7 = ADDRESS OF FIRST UNALLOCATED ENTRY IN MAILBOX TABLE.
                                0493      ;         R8 = BYTE COUNT QUOTA OF MESSAGES.
                                0493      ;         00(SP) = SAVED PROCESSOR IPL.
                                0493      ;
                                0493      ;         MAILBOX TABLE MUTEX OWNED, MAILBOX TABLE LOCK OWNED, IPL = 31.
                                0493      ;
                                0493      ; OUTPUTS:
                                0493      ;
                                0493      ;         R0 = SUCCESS OR FAILURE.
                                0493      ;         R7 = ADDRESS OF CREATED MAILBOX CONTROL BLOCK.
                                0493      ;--
                                0493      CREATE_SHMMBX:                                  ;CREATE SHARED MEMORY MAILBOX
        50    03B4 8F     3C    0493              MOVZWL  #SS$_NOSHMBLOCK,R0              ;ASSUME NO SLOT AVAILABLE
                     57   D5    0498              TSTL    R7                              ;WAS A SLOT AVAILABLE?
                     5D   13    049A              BEQL    UNLOCK_SHMMBX                   ;IF EQL NO SLOT FOUND
        50    03AC 8F     3C    049C              MOVZWL  #SS$_EXPORTQUOTA,R0             ;ASSUME PORT'S MAILBOX LIMIT EXCEEDED
         51     04 A6     D0    04A1              MOVL    SHB$L_DATAPAGE(R6),R1           ;GET DATAPAGE ADDRESS
         52     15 A6     9A    04A5              MOVZBL  SHB$B_PORT(R6),R2               ;GET PORT NUMBER
              5C A142     B5    04A9              TSTW    SHD$W_MBXQUOTA(R1)[R2]          ;DOES PORT HAVE ANY MAILBOX QUOTA LEFT?
                     4A   15    04AD              BLEQ    UNLOCK_SHMMBX                   ;IF LEQ QUOTA EXHAUSTED
              5C A142     B7    04AF              DECW    SHD$W_MBXQUOTA(R1)[R2]          ;DECREMENT PORT'S QUOTA
          09 A7      52   90    04B3              MOVB    R2,MBX$B_CREATPORT(R7)          ;SET CREATOR PORT NUMBER
                     67   7C    04B7              CLRQ    MBX$Q_MSG(R7)                   ;INITIALIZE MESSAGE QUEUE LISTHEAD
                                04B9              ASSUME MBX$W_REF+2 EQ MBX$W_READER
                                04B9              ASSUME MBX$W_READER+2 EQ MBX$W_READAST
                                04B9              ASSUME MBX$W_READAST+2 EQ MBX$Q_WRITAST
                  0C A7   7C    04B9              CLRQ    MBX$Q_REF(R7)                   ;INIT REF FLAGS, READER, READAST, AND WRITAS
      0C A7   01     52   78    04BC              ASHL    R2,#1,MBX$W_REF(R7)             ;SET THIS PORT'S REFERENCE FLAG
                  16 A7   B4    04C1              CLRW    MBX$W_MSGCNT(R7)                ;CLEAR MESSAGE COUNT
      14 A7       0C AC   B0    04C4              MOVW    MAXMSG(AP),MBX$W_MAXMSG(R7) ;SET MAXIMUM MESSAGE SIZE
                     08   12    04C9              BNEQ    10$                             ;IF NEQ MESSAGE SIZE SPECIFIED
  14 A7     00000000'EF   B0    04CB              MOVW    IOC$GW_MBXMXMSG,MBX$W_MAXMSG(R7) ;SET DEFAULT MAXIMUM SIZE
          18 A7      58   B0    04D3      10$:    MOVW    R8,MBX$W_BUFFQUO(R7)            ;SET BUFFER QUOTA
        1A A7     14 AC   B0    04D7              MOVW    PROMSK(AP),MBX$W_PROT(R7) ;SET PROTECTION MASK
     1C A7      00BC C4   D0    04DC              MOVL    PCB$L_UIC(R4),MBX$L_OWNUIC(R7) ;SET OWNER UIC
                     30   BB    04E2              PUSHR   #^M<R4,R5>                      ;SAVE MOVC REGISTERS
     00      04 BB   6B   2C    04E4              MOVC5   (R11),@4(R11),#0,-              ;SET MAILBOX NAME
             21 A7   0F         04E9                      #15,MBX$T_NAME+1(R7)            ;
                     30   BA    04EC              POPR    #^M<R4,R5>                      ;RESIORE MOVC REGISTERS
             20 A7   6B   90    04EE              MOVB    (R11),MBX$T_NAME(R7)            ;SET MAILJOX NAME SIZE
             08 A7   02   88    04F2              BISB    #MBX$M_VALID,MBX$B_FLAGS(R7) ;MARK MAILBOX VALID
                50   01   D0    04F6              MOVL    S^#SS$_NORMAL,R0                ;SET SUCCESS
                                04F9      UNLOCK_SHMMBX:                                  ;UNLOCK SHARED MEMORY MAILBOXES
           51     04 A6   D0    04F9              MOVL    SHB$L_DATAPAGE(R6),R1           ;GET ADDRESS OF MEMORY DATAPAGE
                                04FD              UNLOCK  #SHD$V_MBXLCK,SHD$B_FLAGS(R1) ;UNLOCK MAILBOX TABLE
                                0506      UNLOCK_SHMMBX_ERROR:                            ;ERROR EXIT PATH THAT RELEASES MUTEX
                     50   DD    0506              PUSHL   R0                              ;SAVE EXIT STATUS
     50     00000000'EF   DE    0508              MOVAL   EXE$GL_SHMMBMTX,R0              ;GET ADDRESS OF MAILBOX MUTEX
            00000000'GF   16    050F              JSB     G^SCH$UNLOCK                    ;UNLOCK SHARED MEMORY MAILBOX MUTEX
                   50 8ED0      0515              POPL    R0                              ;RESTORE EXIT STATUS
             0518        SETIPL  #IPL$_ASTDEL            ;RESTORE IPL
         05  051B        RSB                             ;
             051C SHMMBX_CODE:                           ;IPL TO LOCK PREVIOUS CODE IN MEMORY
   00000008  051C        .LONG   IPL$_SYNCH
             0520        ASSUME  <.-SEARCH_SHMMBX> LE 512 ;CAN ONLY LOCK 1 PAGE
```

```
                        0520          .SBTTL  CREATE_SHMUCB - CREATE SHARED MEMORY MAILBOX UCB
                        0520  ;++
                        0520  ; THIS ROUTINE IS CALLED TO CREATE A UCB FOR A SHARED MEMORY MAILBOX.
                        0520  ;
                        0520  ; INPUTS:
                        0520  ;
                        0520  ;       R4 = CURRENT PROCESS PCB ADDRESS
                        0520  ;       R5 = ADDRESS OF TEMPLATE MAILBOX UCB.
                        0520  ;       R6 = ADDRESS OF SHARED MEMORY CONTROL BLOCK.
                        0520  ;       R7 = ADDRESS OF SHARED MEMORY MAILBOX CONTROL BLOCK.
                        0520  ;       R8 = BYTE COUNT QUOTA FOR MESSAGES.
                        0520  ;
                        0520  ; OUTPUTS:
                        0520  ;
                        0520  ;       R0 = SUCCESS OR FAILURE.
                        0520  ;       R6 = ADDRESS OF CREATED UCB.
                        0520  ;
                        0520  ;       EXECUTES AT DEVICE IPL, IPL$_MAILBOX.
                        0520  ;       I/O DATABASE LOCKED FOR WRITE.
                        0520  ;--
                        0520  CREATE_SHMUCB:                          ;CREATE SHARED MEMORY MAILBOX UCB
                        0520          DSBINT  70$                     ;Lock out PRQ's until UCB complete and
                        052A                                          ;lock this code for high IPL execution.
             FAD3'  30  052A          BSBW    IOC$COPY_UCB            ;Make a copy of template UCB.
             68 50  E9  052D          BLBC    R0, 20$                 ;Branch if copy operation failed.
     54 A2   0A A7  B0  0530          MOVW    MBX$W_UNIT(R7), -       ;Establish the unit number of the
                        0535                  UCB$W_UNIT(R2)          ;new UCB.
             FAC8'  30  0535          BSBW    IOC$LINK_UCB            ;Link the new UCB.
             5D 50  E9  0538          BLBC    R0, 20$                 ;Branch if link operation failed.
      009C C2   56  D0  053B          MOVL    R6,UCB$L_MB_SHB(R2)     ;SAVE ADDRESS OF SHB
             0C A6  D6  0540          INCL    SHB$L_REFCNT(R6)        ;INCREMENT MEMORY REFEPENCE COUNT
   00A8 C2   15 A6  9A  0543          MOVZBL  SHB$B_PORT(R6),UCB$L_MB_PORT(R2) ;SET PORT NUMBER
           56   52  D0  0549          MOVL    R2,R6                   ;SAVE ADDRESS OF UCB
      0098 C6   57  D0  054C          MOVL    R7,UCB$L_MB_MBX(R6)     ;SAVE ADDRESS OF MAILBOX
  00A0 C6  00A0 C6  DE  0551          MOVAL   UCB$L_MB_WIOQFL(R6),UCB$L_MB_WIOQFL(R6) ;INIT WRITE I/O REQUEST QUEU
  00A4 C6  00A0 C6  DE  0558          MOVAL   UCB$L_MB_WIOQFL(R6),UCB$L_MB_WIOQBL(R6) ;
     18 A6   18 A7  B0  055F          MOVW    MBX$W_BUFFQUO(R7),UCB$W_DUFQUO(R6) ;SET BUFFER QUOTA
     42 A6   14 A7  B0  0564          MOVW    MBX$W_MAXMSG(R7),UCB$W_DEVBUFSIZ(R6) ;SET MAXIMUM MESSAGE SIZE
        50   1C A6  D0  0569          MOVL    UCB$L_ORB(R6),R0        ;GET ORB ADDRESS
        0B A0   09  88  056D          BISB2   #ORB$M_PROT_16 -        ;USE SOGW PROTECTION WORD
                        0571                  !ORB$M_NOACL,ORB$B_FLAGS(R0)            ;NO ACLS PERMITTED
        0B A0   02  8A  0571          BICB2   #ORB$M_ACL_QUEUE,ORB$B_FLAGS(R0)        ;THEREFORE, NO QUEUE
             04 A0  D4  0575          CLRL    ORB$L_ACL_MUTEX(R0)                     ; AND NO MUTEX
                        0578          ASSUME  ORB$L_ACL_DESC EQ ORB$L_ACL_COUNT+4
             28 A0  7C  0578          CLRQ    ORB$L_ACL_COUNT(R0)                     ; AND NO ACL PRESENT
     18 A0   1A A7  B0  057B          MOVW    MBX$W_PROT(R7),ORB$W_PROT(R0) ;SET PROTECTION MASK
        60   1C A7  D0  0580          MOVL    MBX$L_OWNUIC(R7),ORB$L_OWNER(R0) ;Establish owner UIC.
              FCBC  30  0584          BSBW    COMP_CHARGES            ;COMPUTE QUOTA CHARGES
        52   54 A6  3C  0587          MOVZWL  UCB$Q_UNIT(R6),R2       ;GET UNIT NUMBER
        51   24 A6  D0  058B          MOVL    UCB$L_CRB(R6),R1        ;GET CRB ADDRESS
        51   2C A1  D0  058F          MOVL    CRB$L_INTD+VEC$L_IDB(R1),R1 ;GET IDB ADDRESS
      18 A142   56  D0  0593          MOVL    R6,IDB$L_UCBLST(R1)[R2] ;SET UCB ADDRESS IN IDB LIST
                        0598  20$:
                        0598          ENBINT                          ;ALLOW PRQ PROCESSING NOW
                    05  059B          RSB                             ;
                        059C
              0000000B  059C  70$:    .LONG   IPL$_MAIIBOX                    ;Construct used to temporarily
```

```
05A0   975         ASSUME  <. - CREATE_SHMUCB> LE 512       ;lock less than one page at
05A0   976                                                  ;elevated IPL.
```

```
                             05A0               .SBTTL  EXE$DELMBX - DELETE MAILBOX
                             05A0      ;+
                             05A0      ; EXE$DELMBX - DELETE MAILBOX
                             05A0      ;
                             05A0      ; THIS SERVICE PROVIDES THE CAPABILITY TO MARK A PERMANENT MAILBOX FOR
                             05A0      ; DELETION. ACTUAL DELETION OF THE MAILBOX OCCURS WHEN ITS REFERENCE COUNT
                             05A0      ; REACHES ZERO.
                             05A0      ;
                             05A0      ; INPUTS:
                             05A0      ;
                             05A0      ;       DLCHAN(AP) = CHANNEL NUMBER THAT THE MAILBOX TO BE DELETED IS ASSIGNED TO.
                             05A0      ;
                             05A0      ;       R4 = CURRENT PROCESS PCB ADDRESS.
                             05A0      ;
                             05A0      ; OUTPUTS:
                             05A0      ;
                             05A0      ;       R0 LOW BIT CLEAR INDICATES FAILURE TO MARK MAILBOX FOR DELETE.
                             05A0      ;
                             05A0      ;               R0 = SS$_DEVNOTMBX - SPECIFIED CHANNEL IS NOT ASSIGNED TO A
                             05A0      ;                       MAILBOX.
                             05A0      ;
                             05A0      ;               R0 = SS$_IVCHAN - INVALID CHANNEL NUMBER SPECIFIED.
                             05A0      ;
                             05A0      ;               R0 = SS$_NOPRIV - PROCESS DOES NOT HAVE PRIVILEGE TO DELETE A
                             05A0      ;                       PERMANENT MAILBOX, SPECIFIED CHANNEL IS NOT ASSIGNED
                             05A0      ;                       TO A DEVICE, OR THE CALLER DOES NOT HAVE SUFFICIENT
                             05A0      ;                       PRIVILEGE TO ACCESS THE CHANNEL.
                             05A0      ;
                             05A0      ;       R0 LOW BIT SET INDICATES SUCCESSFUL COMPLETION.
                             05A0      ;
                             05A0      ;               R0 = SS$_NORMAL - NORMAL COMPLETION.
                             05A0      ;-
                             05A0
                       003C  05A0               .ENTRY  EXE$DELMBX,^M<R2,R3,R4,R5>
     50    04 AC         3C  05A2               MOVZWL  DLCHAN(AP),R0                 ;GET CHANNEL NUMBER OF MAILBOX
             FA57'       30  05A6               BSBW    IOC$VERIFYCHAN                ;VERIFY CHANNEL NUMBER
            22 50        E9  05A9               BLBC    R0,20$                        ;IF LBC INVALID CHANNEL
        55     61        D0  05AC               MOVL    CCB$L_UCB(R1),R5              ;GET ASSIGNED DEVICE UCB ADDRESS
   50     0074 8F        3C  05AF               MOVZWL  #SS$_DEVNOTMBX,R0             ;SET DEVICE NOT MAILBOX STATUS
 15 38 A5      14        E1  05B4               BBC     #DEV$V_MBX,UCB$L_DEVCHAR(R5),20$ ;IF CLR, DEVICE NOT MAILBOX
       50      24        3C  05B9               MOVZWL  #SS$_NOPRIV,R0                ;ASSUME CALLER DOES NOT HAVE PRIVILEGE
 0A 68 A5      00        E1  05BC               BBC     #UCB$V_PRMMBX,UCB$W_DEVSTS(R5),10$ ;IF CLR, NOT PERMANENT MAILBOX
                             05C1               IFNPRIV PRMMBX,20$                    ;CALLER HAVE PRIVILEGE TO DELETE?
    68 A5      02        A8  05C7               BISW    #UCB$M_DELMBX,UCB$W_DEVSTS(R5) ;MARK MAILBOX FOR DELETION
       50      01        3C  05CB      10$:     MOVZWL  #SS$_NORMAL,R0                ;SET NORMAL COMPLETION
                         04  05CE      20$:     RET                                   ;
                             05CF
                             05CF               .END
```

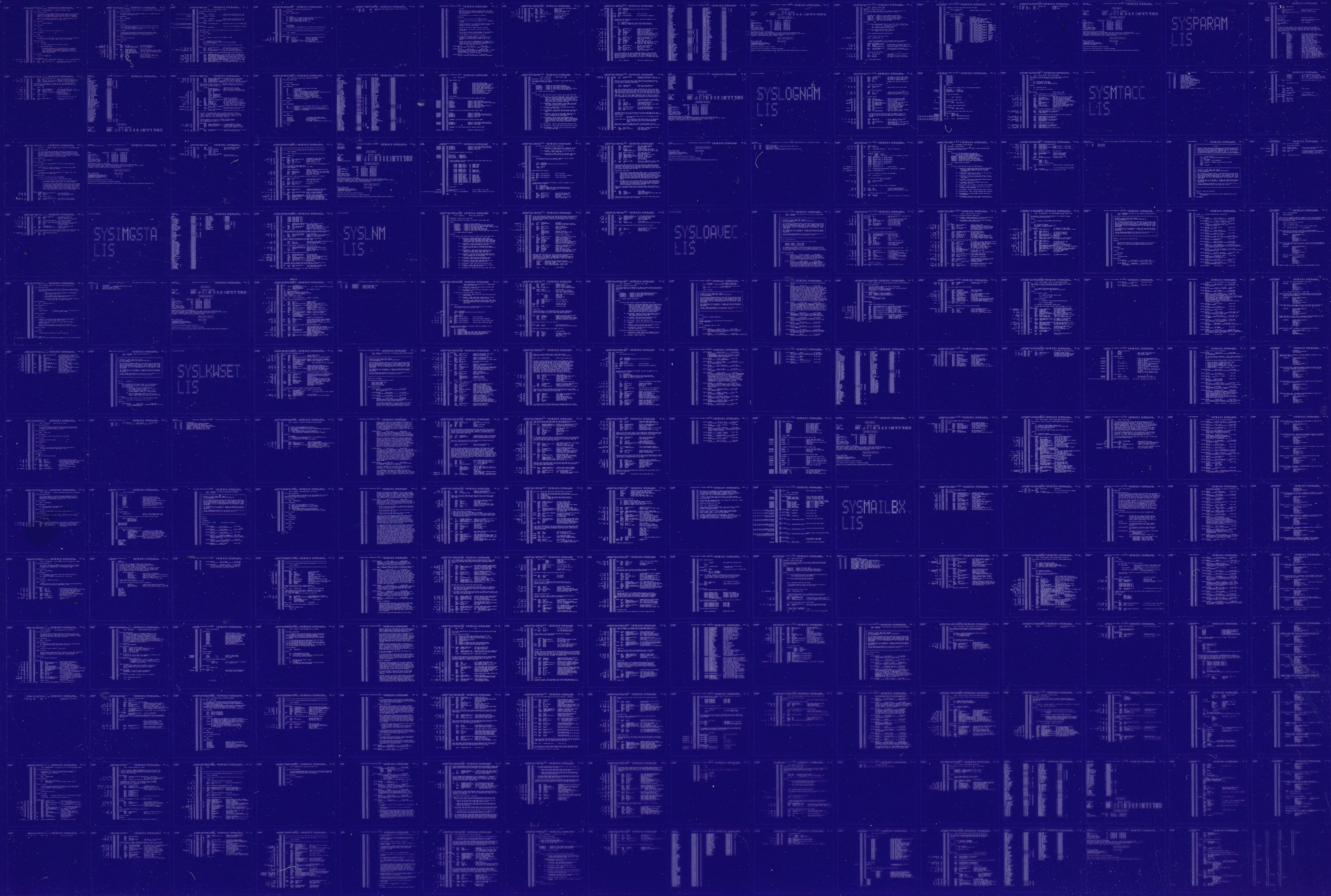
```
ACMODE                00000018              I5_BUF_ADDR           FFFFFFCC
ADP$L_SHB           = 00000030              I5_CODE_LEN           FFFFFFC8
ASSIGN                0000025F R      02    I5_RET_LEN            FFFFFFD0
ASSIGN_IT             00000159 R      02    I6_BUF_ADDR           FFFFFFD8
BACKPTR_INDEX         00000000 R      02    I6_CODE_LEN           FFFFFFD4
BAD_LOGNAM            0000004A R      02    I6_RET_LEN            FFFFFFDC
BUFQUO                00000010              IDB$L_UCBLST        = 00000018
BUG$_KRPEMPTY         *********  X    02    IOC$CHKMBXQUOTA       *********  X    02
BUG$_SSRVEXCEPT       *********  X    02    IOC$CLONE_UCB         *********  X    02
BUGCHECK              000001F7 R      02    IOC$COPY_UCB          *********  X    02
CCB$B_AMOD          = 00000009              IOC$CVT_DEVNAM        *********  X    02
CCB$L_UCB           = 00000000              IOC$DEBIT_UCB         *********  X    02
CCBADDR               FFFFFFF0              IOC$FFCHAN            *********  X    02
CHANADDR              FFFFFFF4              IOC$GL_DEVLIST        *********  X    02
CHANNUM               FFFFFFEC              IOC$GW_MBXBFQUO       *********  X    02
CHECK_QUOTA           000001FB R      02    IOC$GW_MBXMXMSG       *********  X    02
COMP_CHARGES          00000243 R      02    IOC$LINK_UCB          *********  X    02
CRB$L_INTD          = 00000024              IOC$VERIFYCHAN        *********  X    02
CRCHAN                00000008              IPL$_ASTDEL         = 00000002
CREATE_LOCALUCB       00000212 R      02    IPL$_MAILBOX        = 0000000B
CREATE_LOGNAM         000002F2 R      02    IPL$_SYNCH          = 00000008
CREATE_SHMMBX         00000493 R      02    KRP                   FFFFFF94
CREATE_SHMUCB         00000520 R      02    LNM$C_NAMLENGTH     = 000000FF
CRELNM_END_LIST       FFFFFFE0              LNM$M_TERMINAL      = 00000200
CTL$GL_KRPFL          *********  X    02    LNM$_ATTRIBUTES     = 00000003
CTL$GL_PCB            *********  X    02    LNM$_INDEX          = 00000001
DDB$L_LINK          = 00000000              LNM$_LNMB_ADDR      = 00000009
DDB$L_UCB           = 00000004              LNM$_STRING         = 00000002
DEV$V_MBX           = 00000014              LNMX$C_BACKPTR      = FFFFFF81
DIR...              = FFFFFFFF              LNM_TERM_ATTR         00000004 R      02
DISPATCH              0000010D R      02    LOCALSIZ              FFFFFF90
DLCHAN                00000004              LOCKID                FFFFFFFC
DYN$C_UCB           = 00000010              LOCKSTATUS            FFFFFFF8
EXE$CREMBX            00000061 RG     02    LOGNAM                0000001C
EXE$DELMBX            000005A0 RG     02    MAILBOX_TABLES        00000008 R      02
EXE$GL_LOCKRTRY       *********  X    02    MAXMSG                0000000C
EXE$GL_SHBLIST        *********  X    02    MB$UCB0               *********  X    02
EXE$GL_SHMMBMTX       *********  X    02    MBX$B_CREATPORT     = 00000009
EXE$MAXACMODE         *********  X    02    MBX$B_FLAGS         = 00000008
EXIT                  0000004F R      02    MBX$C_LENGTH        = 00000030
EXIT_CLEAR_FLAGS      00000197 R      02    MBX$L_OWNUIC        = 0000001C
EXIT_DEASSIGN         000001B1 R      02    MBX$M_VALID         = 00000002
EXIT_LOWER_IPL        000001AC R      02    MBX$Q_MSG           = 00000000
EXIT_UNLOCKIODB       0000019A R      02    MBX$T_NAME          = 00000020
GET_SHMTEMPLATE       000003B4 R      02    MBX$V_VALID         = 00000001
I1_BUF_ADDR           FFFFFF9C              MBX$W_BUFFQUO       = 00000018
I1_CODE_LEN           FFFFFF98              MBX$W_MAXMSG        = 00000014
I1_RET_LEN            FFFFFFA0              MBX$W_MSGCNT        = 00000016
I2_BUF_ADDR           FFFFFFA8              MBX$W_PROT          = 0000001A
I2_CODE_LEN           FFFFFFA4              MBX$W_READAST       = 00000010
I2_RET_LEN            FFFFFFAC              MBX$W_READER        = 0000000E
I3_BUF_ADDR           FFFFFFB4              MBX$W_REF           = 0000000C
I3_CODE_LEN           FFFFFFB0              MBX$W_UNIT          = 0000000A
I3_RET_LEN            FFFFFFB8              MBX$W_WRITAST       = 00000012
I4_BUF_ADDR           FFFFFFC0              MBXNAMSIZ           = 00000011
I4_CODE_LEN           FFFFFFBC              MMG$MBXTRNLOG         *********  X    02
I4_RET_LEN            FFFFFFC4              NAMEDLOCAL            0000011B R      02
```

```
NONAME                    00000143 R      02    TRNLNM_END_LIST         = FFFFFFB0
ORB$B_FLAGS             = 0000000B              UCB$B_TYPE              = 0000000A
ORB$L_ACL_COUNT         = 00000028              UCB$L_CRB               = 00000024
ORB$L_ACL_DESC          = 0000002C              UCB$L_DEVCHAR           = 00000038
ORB$L_ACL_MUTEX         = 00000004              UCB$L_DEVDEPEND         = 00000044
ORB$L_OWNER             = 00000000              UCB$L_LOGADR            = 00000074
ORB$M_ACL_QUEUE         = 00000002              UCB$L_MB_MBX            = 00000098
ORB$M_NOACL             = 00000008              UCB$L_MB_PORT           = 000000A8
ORB$M_PROT_16           = 00000001              UCB$L_MB_SHB            = 0000009C
ORB$W_PROT              = 00000018              UCB$L_MB_WIOQBL         = 000000A4
PCB$L_UIC               = 000000BC              UCB$L_MB_WIOQFL         = 000000A0
PCB$Q_PRIV              = 00000084              UCB$L_ORB               = 0000001C
PR$_IPL                 = 00000012              UCB$L_PID               = 0000002C
PRMFLG                    00000004              UCB$M_DELMBX            = 00000002
PROMSK                    00000014              UCB$V_PRMMBX            = 00000000
PRV$V_PRMMBX            = 0000000B              UCB$V_SHMMBX            = 00000003
PRV$V_SHMEM             = 0000001B              UCB$W_BUFQUO            = 00000018
PRV$V_TMPMBX            = 0000000F              UCB$W_CHARGE            = 00000056
RETURN                    000001E5 R      02    UCB$W_DEVBUFSIZ         = 00000042
SAVABS...               = FFFFFF90              UCB$W_DEVSTS            = 00000068
SCH$IOLOCKW               ********    X   02    UCB$W_REFC              = 0000005C
SCH$IOUNLOCK              ********    X   02    UCB$W_SIZE              = 00000008
SCH$LOCKW                 ********    X   02    UCB$W_UNIT              = 00000054
SCH$UNLOCK                ********    X   02    UCB_ADDRESS               FFFFFFE4
SEARCH_MBXUCB             00000285 R      02    UCB_ADDR_SIZE           = FFFFFFB4
SEARCH_SHMMBX             0000040B R      02    UNLOCK_SHMMBX             000004F9 R      02
SHAREDMEM                 00000161 R      02    UNLOCK_SHMMBX_ERROR       00000506 R      02
SHB$B_FLAGS             = 0000000B              VEC$L_ADP               = 00000014
SHB$B_PORT              = 00000015              VEC$L_IDB               = 00000008
SHB$L_DATAPAGE          = 00000004              ZERO_INDEX              = FFFFFFF0
SHB$L_LINK              = 00000000
SHB$L_REFCNT            = 0000000C
SHB$V_CONNECT           = 00000000
SHBADDR                   FFFFFFE8
SHD$B_FLAGS             = 0000009F
SHD$C_NAMLENGTH         = 00000010
SHD$L_MBXPTR            = 00000000
SHD$T_NAME              = 00000020
SHD$V_MBXLCK            = 00000003
SHD$W_MBXMAX            = 0000001A
SHD$W_MBXQUOTA          = 0000005C
SHMMBX_CODE               0000051C R      02
SS$_ACCVIO              = 0000000C
SS$_DEVNOTMBX           = 00000074
SS$_EXPORTQUOTA         = 000003AC
SS$_INTERLOCK           = 0000038C
SS$_IVLOGNAM            = 00000154
SS$_IVSTSFLG            = 0000017C
SS$_NOPRIV              = 00000024
SS$_NORMAL              = 00000001
SS$_NOSHMBLOCK          = 000003B4
SS$_SHMNOTCNCT          = 0000037C
SYS$CRELNM                ********   GX   02
SYS$DASSGN                ********   GX   02
SYS$DELMBX                ********   GX   02
SYS$TRNLNM                ********   GX   02
TO_EXIT_LOWER_IPL         0000010A R      02
```

```
                                                    +----------------+
                                                    ! Psect synopsis !
                                                    +----------------+

PSECT name                      Allocation          PSECT No.  Attributes
----------                      ----------          ---------  ----------
.  ABS  .                       00000000  (     0.)  00 (   0.)  NOPIC    USR    CON    ABS    LCL NOSHR NOEXE NORD   NOWRT NOVEC BYTE
$ABS$                           FFFFFFFC  (     0.)  01 (   1.)  NOPIC    USR    CON    ABS    LCL NOSHR   EXE   RD     WRT NOVEC BYTE
Y$EXEPAGED                      000005CF  (  1487.)  02 (   2.)  NOPIC    USR    CON    REL    LCL NOSHR   EXE   RD     WRT NOVEC BYTE

                                                +------------------------+
                                                ! Performance indicators !
                                                +------------------------+

Phase                        Page faults    CPU Time       Elapsed Time
-----                        -----------    --------       ------------
Initialization                        37    00:00:00.08    00:00:01.31
Command processing                   130    00:00:00.56    00:00:03.01
Pass 1                               494    00:00:20.62    00:00:53.71
Symbol table sort                      0    00:00:02.96    00:00:06.31
Pass 2                               199    00:00:04.21    00:00:09.07
Symbol table output                   25    00:00:00.19    00:00:00.41
Psect synopsis output                  2    00:00:00.01    00:00:00.02
Cross-reference output                 0    00:00:00.00    00:00:00.00
Assembler run totals                 889    00:00:28.64    00:01:13.87

The working set limit was 1950 pages.
116811 bytes (229 pages) of virtual memory were used to buffer the intermediate code.
There were 100 pages of symbol table space allocated to hold 1914 non-local and 31 local symbols.
1026 source lines were read in Pass 1, producing 23 object records in Pass 2.
49 pages of virtual memory were used to define 47 macros.

                                                +--------------------------+
                                                ! Macro library statistics !
                                                +--------------------------+

Macro library name                               Macros defined
------------------                               --------------
_$255$DUA28:[SYS.OBJ]LIB.MLB;1                             26
_$255$DUA28:[SYSLIB]STARLET.MLB;2                          18
TOTALS (all libraries)                                     44

2124 GETS were required to define 44 macros.

There were no errors, warnings or information messages.

MACRO/LIS=LIS$:SYSMAILBX/OBJ=OBJ$:SYSMAILBX MSRC$:SYSMAILBX/UPDATE=(ENH$:SYSMAILBX)+EXECML$/LIB
```

0386 AH-BT13A-SE
VAX/VMS V4.0

DIGITAL EQUIPMENT CORPORATION
CONFIDENTIAL AND PROPRIETARY



SYSPARAM
LIS

SYSLOGNAM
LIS

SYSMTACC
LIS

SYSIMGSTA
LIS

SYSLNM
LIS

SYSLOAVEC
LIS

SYSLKWSET
LIS

SYSMAILBX
LIS