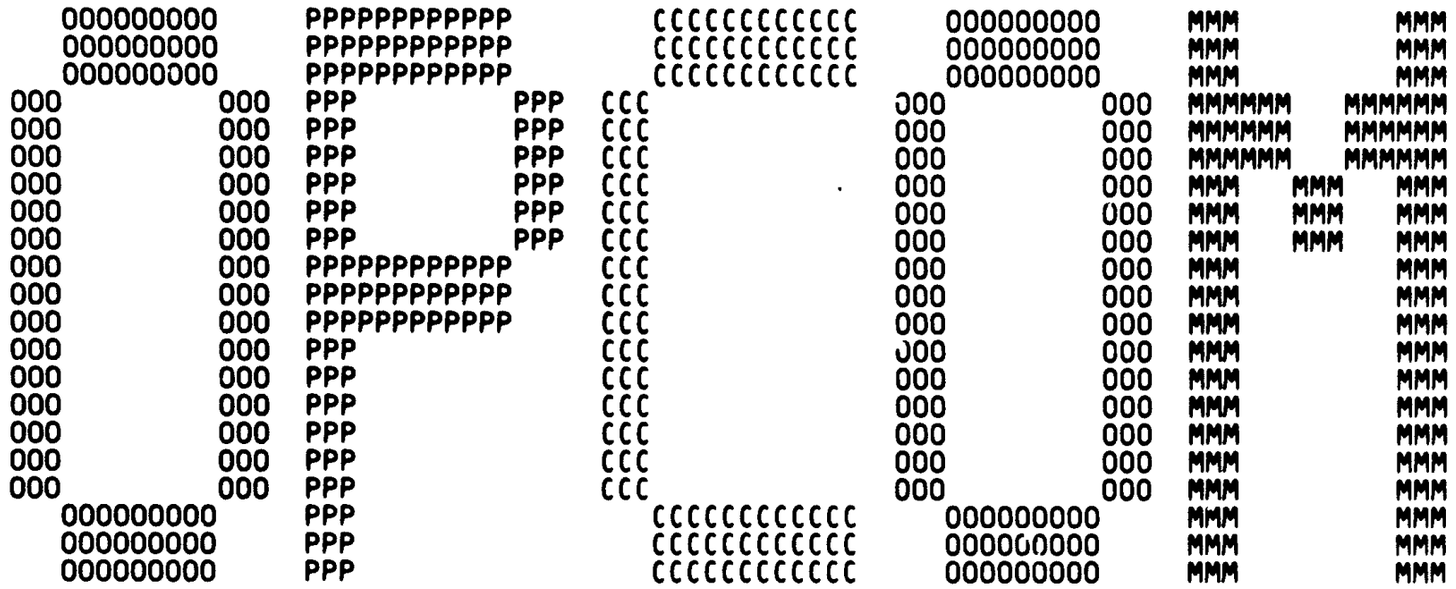

_$2

Sym

ALL

ASC

BOD
BOD
BOD
BOD
BOD
BOD
BOD
BOD
BUG
BYP
CAN
CAN
CAN
CHE
CHE

CLU
CLU
CLU
CLU
CLU
CLU
CLU

CLU
CLU
CLU
CLU
CLU
CLU
CLU
CLU

CLU
CLU

```
  000000    PPPPPPPP      CCCCCCCC    000000    MM      MM  MM      MM    AAAAAA      IIIIII    NN      NN
  000000    PPPPPPPP      CCCCCCCC    000000    MM      MM  MM      MM    AAAAAA      IIIIII    NN      NN
00      00  PP      PP  CC          00      00  MMMM  MMMM  MMMM  MMMM  AA      AA      II      NN      NN
00      00  PP      PP  CC          00      00  MMMM  MMMM  MMMM  MMMM  AA      AA      II      NN      NN
00      00  PP      PP  CC          00      00  MM  MM  MM  MM  MM  MM  AA      AA      II      NNNN    NN
00      00  PP      PP  CC          00      00  MM  MM  MM  MM  MM  MM  AA      AA      II      NNNN    NN
00      00  PPPPPPPP    CC          00      00  MM      MM  MM      MM  AA      AA      II      NN  NN  NN
00      00  PPPPPPPP    CC          00      00  MM      MM  MM      MM  AA      AA      II      NN  NN  NN
00      00  PP          CC          00      00  MM      MM  MM      MM  AAAAAAAAAA      II      NN    NNNN
00      00  PP          CC          00      00  MM      MM  MM      MM  AAAAAAAAAA      II      NN    NNNN
00      00  PP          CC          00      00  MM      MM  MM      MM  AA      AA      II      NN      NN      ....
00      00  PP          CC          00      00  MM      MM  MM      MM  AA      AA      II      NN      NN      ....
  000000    PP            CCCCCCCC    000000    MM      MM  MM      MM  AA      AA    IIIIII    NN      NN      ....
  000000    PP            CCCCCCCC    000000    MM      MM  MM      MM  AA      AA    IIIIII    NN      NN      ....

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
```

```
;     1          0001  0 MODULE  OPC$OPCOMMAIN   (
;     2          0002  0                         LANGUAGE (BLISS32),
;     3          0003  0                         IDENT = 'V04-000',
;     4          0004  0                         MAIN = OPCOM_MAIN
;     5          0005  0                        ) =
;     6          0006  0
;     7          0007  0 !*****************************************************************************
;     8          0008  0 !*                                                                           *
;     9          0009  0 !*  COPYRIGHT (c) 1978, 1980, 1982, 1984 BY                                  *
;    10          0010  0 !*  DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASSACHUSETTS.                    *
;    11          0011  0 !*  ALL RIGHTS RESERVED.                                                     *
;    12          0012  0 !*                                                                           *
;    13          0013  0 !*  THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND COPIED   *
;    14          0014  0 !*  ONLY IN  ACCORDANCE WITH  THE  TERMS  OF  SUCH  LICENSE  AND WITH THE  *
;    15          0015  0 !*  INCLUSION OF THE ABOVE COPYRIGHT NOTICE. THIS SOFTWARE OR  ANY  OTHER   *
;    16          0016  0 !*  COPIES THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY   *
;    17          0017  0 !*  OTHER PERSON.  NO TITLE TO AND OWNERSHIP OF  THE  SOFTWARE IS  HEREBY   *
;    18          0018  0 !*  TRANSFERRED.                                                             *
;    19          0019  0 !*                                                                           *
;    20          0020  0 !*  THE INFORMATION IN THIS SOFTWARE IS  SUBJECT TO CHANGE WITHOUT NOTICE   *
;    21          0021  0 !*  AND  SHOULD  NOT  BE  CONSTRUED AS  A COMMITMENT BY DIGITAL EQUIPMENT   *
;    22          0022  0 !*  CORPORATION.                                                             *
;    23          0023  0 !*                                                                           *
;    24          0024  0 !*  DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE  OR  RELIABILITY OF ITS   *
;    25          0025  0 !*  SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL.                  *
;    26          0026  0 !*                                                                           *
;    27          0027  0 !*                                                                           *
;    28          0028  0 !*****************************************************************************
;    29          0029  0
;    30          0030  0 !++
;    31          0031  0 ! FACILITY:
;    32          0032  0 !
;    33          0033  0 !       OPCOM
;    34          0034  0 !
;    35          0035  0 ! ABSTRACT:
;    36          0036  0 !
;    37          0037  0 !       This module contains the top leve logic for OPCOM, the
;    38          0038  0 !       OPerator COmunication Manager.  OPCOM will provide the
;    39          0039  0 !       interface between a user and an operator on the system.
;    40          0040  0 !       Specifically, this module contains the routines responsible
;    41          0041  0 !       for starting OPCOM in an orderly manner, receiving requests
;    42          0042  0 !       and dispatching them to the proper handler, and for the
;    43          0043  0 !       orderly shutdown of OPCOM.
;    44          0044  0 !
;    45          0045  0 ! Environment:
;    46          0046  0 !
;    47          0047  0 !       VAX/VMS operating system.
;    48          0048  0 !
;    49          0049  0 ! Author:
;    50          0050  0 !
;    51          0051  0 !       Steven T. Jeffreys
;    52          0052  0 !
;    53          0053  0 ! Creation date:
;    54          0054  0 !
;    55          0055  0 !       March 10, 1981
;    56          0056  0 !
;    57          0057  0 ! Revision history:
```

```
;    58          0058  0 !
;    59          0059  0 !       V03-006 CWH3006         CW Hobbs                        25-Jul-1984
;    60          0060  0 !               Remove DEBUG routines to save disk space on kit.
;    61          0061  0 !
;    62          0062  0 !       V03-005 CWH3169         CW Hobbs                        5-May-1984
;    63          0063  0 !               Second pass for cluster-wide OPCOM:
;    64          0064  0 !                - Add RC25, RDRX, TU81 and MAYA microcode messages
;    65          0065  0 !                - Add conditional assembly code for OPC$INCREASE_STRESS and
;    66          0066  0 !                  OPC$CHECK_POOL functions
;    67          0067  0 !
;    68          0068  0 !       V03-004 CWH3004         CW Hobbs                        22-Sep-1983
;    69          0069  0 !               Make dump conditional on a logical name OPC$DUMP_MAILBOX.
;    70          0070  0 !
;    71          0071  0 !       V03-003 CWH3003         CW Hobbs                        16-Sep-1983
;    72          0072  0 !               Add CLUMBX (connection manager) message dispatch.
;    73          0073  0 !
;    74          0074  0 !       V03-002 CWH3002         CW Hobbs                        30-Jul-1983
;    75          0075  0 !               Various and sundry things to make OPCOM distributed
;    76          0076  0 !               across the cluster.
;    77          0077  0 !
;    78          0078  0 !       V03-001 RLRV3A1         Robert L. Rappaport     5-Apr-1982
;    79          0079  0 !               Added MSG$_UDA50MVER and MSG$_DUPUNITNO.
;    80          0080  0 !
;    81          0081  0 !       V02-003 STJ0156         Steven T. Jeffreys      05-Feb-1982
;    82          0082  0 !               Added two new mount verification messages to the
;    83          0083  0 !               dispatcher table; MVCOMPLETE and MVABORTED.
;    84          0084  0 !
;    85          0085  0 !       V02-002 STJ0066         Steven T. Jeffreys      14-Jul-1981
;    86          0086  0 !               Changed function dispatcher to recognize the mount
;    87          0087  0 !               verification messages.
;    88          0088  0 !
;    89          0089  0 !--
;    90          0090  0
;    91          0091  1 BEGIN                                                   ! Start of OPCOMMAIN
;    92          0092  1
;    93          0093  1 LIBRARY 'SYS$LIBRARY:LIB.L32';
;    94          0094  1 LIBRARY 'LIB$:OPCOMLIB';
;    95          0095  1
;    96          0096  1
```

```
;    98          0097  1 GLOBAL ROUTINE OPCOM_MAIN : NOVALUE =
;    99          0098  1
;   100          0099  1 !++
;   101          0100  1 ! Functional description:
;   102          0101  1 !
;   103          0102  1 !       This is the main routine for OPCOM.  When OPCOM is
;   104          0103  1 !       started, control is transfered here.  The main routine
;   105          0104  1 !       will call a special initialization routine to set things
;   106          0105  1 !       up, and then enter its main loop, servicing requests as
;   107          0106  1 !       they arrive.
;   108          0107  1 !
;   109          0108  1 ! Input:
;   110          0109  1 !
;   111          0110  1 !       None.
;   112          0111  1 !
;   113          0112  1 ! Implicit Input:
;   114          0113  1 !
;   115          0114  1 !       None.
;   116          0115  1 !
;   117          0116  1 ! Output:
;   118          0117  1 !
;   119          0118  1 !       None.
;   120          0119  1 !
;   121          0120  1 ! Implict output:
;   122          0121  1 !
;   123          0122  1 !       None.
;   124          0123  1 !
;   125          0124  1 ! Side effects:
;   126          0125  1 !
;   127          0126  1 !       None.
;   128          0127  1 !
;   129          0128  1 ! Routine value:
;   130          0129  1 !
;   131          0130  1 !       None.
;   132          0131  1 !--
;   133          0132  1
;   134          0133  2 BEGIN                                           ! Start of OPCOM_MAIN
;   135          0134  2
;   136          0135  2 MACRO
;   137          0136  2
;   138          0137  2         MSGTYPE  =  0,0,16,0%,                  ! Message type code
;   139          0138  2         RQSTCODE = 38,0,8,0%,                   ! Request type code
;   140          0139  2         CLM_CODE = 39,0,8,0%;                   ! Request type code for cluster message
;   141          0140  2
;   142          0141  2 GLOBAL
;   143          0142  2
;   144          0143  2         FINISHED        : LONG INITIAL (FALSE); ! Boolean used for loop control
;   145          0144  2
;   146          0145  2 EXTERNAL ROUTINE
;   147          0146  2         DUMP_LOG_FILE,                          ! Make a formatted dump in the log file
;   148          0147  2         SHARE_FAO_BUFFER,                       ! Run something through $FAO
;   149          0148  2         OPCOM_INIT              : NOVALUE,      ! OPCOM initialization
;   150          0149  2         TIME_STAMP              : NOVALUE,      ! Does periodic timestamp
;   151          0150  2         WRITE_LOG_FILE,                         ! Write random string to the log file
;   152          0151  2         !
;   153          0152  2         ! Various message handlers for old format messages.
;   154          0153  2         !
```

```
;  155          0154  2          UNKNOWN_HANDLER          : NOVALUE,      ! Unknown message type handler
;  156          0155  2          CLUMBX_HANDLER           : NOVALUE,      ! Cluster mailbox messages (special format)
;  157          0156  2          DEVICE_HANDLER           : NOVALUE,      ! Device online/offline messages (special format)
;  158          0157  2          TERME_HANDLER            : NOVALUE,      ! Enable operator message handler
;  159          0158  2          LOGI_HANDLER             : NOVALUE,      ! Init logfile message handler
;  160          0159  2          RQST_HANDLER             : NOVALUE,      ! Request handler
;  161          0160  2          RPLY_HANDLER             : NOVALUE,      ! Reply handler
;  162          0161  2          CNCL_HANDLER             : NOVALUE,      ! Cancel handler
;  163          0162  2          STS_HANDLER              : NOVALUE,      ! Status handler
;  164          0163  2          SECO_HANDLER             : NOVALUE,      ! Security message handler
;  165          0164  2          !
;  166          0165  2          ! The following are message handlers for the new format messages.
;  167          0166  2          !
;  168          0167  2          OPRENABLE_HANDLER        : NOVALUE,      ! Operator enable handler
;  169          0168  2          LOGFILE_HANDLER          : NOVALUE,      ! Logfile control handler
;  170          0169  2          REQUEST_HANDLER          : NOVALUE,      ! Request handler
;  171          0170  2          REPLY_HANDLER            : NOVALUE,      ! Reply handler
;  172          0171  2          CANCEL_HANDLER           : NOVALUE,      ! Cancel handler
;  173          0172  2          STATUS_HANDLER           : NOVALUE,      ! Status handler
;  174          0173  2          SHUTDOWN_HANDLER         : NOVALUE,      ! Shutdown handler
;  175          0174  2          SECURITY_HANDLER         : NOVALUE,      ! Security handler
;  176          0175  2 !        DEBUG_HANDLER            : NOVALUE,      ! Debug handler, comment out when not needed
;  177          0176  2          CLUSMSG_HANDLER          : NOVALUE;      ! Cluster message handler
;  178          0177  2
;  179          0178  2 EXTERNAL
;  180          0179  2
;  181          0180  2          GLOBAL_STATUS    : BITVECTOR,            ! Global status bits
;  182          0181  2          OPER_MBX_CHAN    : WORD;                 ! Operator mailbox channel
;  183          0182  2
;  184          0183  2 LOCAL
;  185          0184  2
;  186          0185  2          RDB              : $ref_bblock,          ! RDB control structure
;  187          0186  2          IOSB             : $bblock [8],          ! I/O status block
;  188          0187  2          REQUEST_BUFFER   : $bblock [OPC$K_MAXREAD], ! Request receive buffer
;  189          0188  2          REQUEST_DESC     : $desc_block,          ! Request buffer descriptor
;  190          0189  2          STATUS           : LONG;
;  191          0190  2
;  192          0191  2 !
;  193          0192  2 ! Perform the necessary initialization.
;  194          0193  2 !
;  195          0194  2 OPCOM_INIT ();
;  196          0195  2
;  197          0196  2 !
;  198          0197  2 ! Initialize the request buffer descriptor.
;  199          0198  2 !
;  200          0199  2 REQUEST_DESC [DSC$B_DTYPE] = 0;
;  201          0200  2 REQUEST_DESC [DSC$B_CLASS] = 0;
;  202          0201  2 REQUEST_DESC [DSC$A_POINTER] = REQUEST_BUFFER;
;  203          0202  2
;  204          0203  2 !
;  205          0204  2 ! Enter the main loop.
;  206          0205  2 !
;  207          0206  2 WHILE NOT .FINISHED DO
;  208          0207  3     BEGIN
;  209          0208  3     !
;  210          0209  3     ! If a timestamp is pending and OPCOM is not busy with
;  211          0210  3     ! a request, then do the timestamp.
```

```
;  212          0211  3      !
;  213          0212  3      GLOBAL_STATUS [GBLSTS_K_BUSY] = FALSE;                  ! OPCOM is not busy
;  214          0213  3      IF .GLOBAL_STATUS [GBLSTS_K_TIMESTAMP_PENDING]
;  215          0214  3      THEN
;  216          0215  3          TIME_STAMP ();
;  217          0216  3
;  218          0217  3      !
;  219          0218  3      ! Issue a read request to the operator mailbox.
;  220          0219  3      !
;  221        P 0220  4      IF NOT (STATUS = $QIOW(FUNC = IO$_READVBLK,
;  222        P 0221  4                             CHAN = .OPER_MBX_CHAN,
;  223        P 0222  4                             IOSB = IOSB,
;  224        P 0223  4                             EFN  = EFN_K_MAILBOX,
;  225        P 0224  4                             P1 = REQUEST_BUFFER,
;  226        P 0225  4                             P2 = OPC$K_MAXREAD
;  227          0226  4                            ))
;  228          0227  3      THEN
;  229          0228  3          $signal_stop (.STATUS);
;  230          0229  3
;  231          0230  3      !
;  232          0231  3      ! Check the status of the read.
;  233          0232  3      !
;  234          0233  4      IF NOT (STATUS = .IOSB [0,0,16,0])
;  235          0234  3      THEN
;  236          0235  3          IF .STATUS NEQ SS$_ENDOFFILE            ! A COPY to _MBA2: will produce this
;  237          0236  3          THEN
;  238          0237  3              $signal_stop (.STATUS);
;  239          0238  3
;  240          0239  3      !
;  241          0240  3      ! Since OPCOM now has a request to service, set the GBLSTS_K_BUSY bit.
;  242          0241  3      ! This serves as an interlock to prevent the asynchronous
;  243          0242  3      ! timestamp function from going off at an inappropriate time.
;  244          0243  3      !
;  245          0244  3      GLOBAL_STATUS [GBLSTS_K_BUSY] = TRUE;
;  246          0245  3
;  247          0246  3      !
;  248          0247  3      ! Set the request buffer length in the descriptor.
;  249          0248  3      !
;  250          0249  3      REQUEST_DESC [DSC$W_LENGTH] = .IOSB [2,0,16,0];
;  251          0250  3
;  252          0251  3      !
;  253          0252  3      ! For debugging, write the message into the log file, and flush
;  254          0253  3      !
;  255          0254  4      BEGIN
;  256          0255  4      LOCAL
;  257          0256  4          lcl_buf : $bvector [16],
;  258          0257  4          out_dsc : VECTOR [2, LONG];
;  259          0258  4      out_dsc [0] = 16;
;  260          0259  4      out_dsc [1] = lcl_buf;
;  261          0260  4      IF $trnlog (lognam=%ASCID 'OPC$DUMP_MAILBOX', rsllen=out_dsc, rslbuf=out_dsc) EQL ss$_normal
;  262          0261  4      THEN
;  263          0262  4          DUMP_LOG_FILE (REQUEST_DESC, %ASCID 'Record received in mailbox');
;  264          0263  3      END;
;  265          0264  3
;  266          0265  3      !
;  267          0266  3      ! For debugging, check the cluster configuration at every mailbox read.  To put additional
;  268          0267  3      ! stress on the system, request acks from every node if the configuration has changed.  Make
```

```
;  269          0268  3     ! all this noise switchable by defining the logical name OPC$INCREASE_STRESS (presence of
;  270          0269  3     ! the name is significant, not its value).
;  271          0270  3     !
;  272        L 0271  3     %IF %VARIANT GTR 0                              ! Only include if /VARIANT specified
;  273        U 0272  3     %THEN
;  274        U 0273  3     BEGIN
;  275        U 0274  3     LOCAL               lcl_buf : $bvector [16], out_dsc : VECTOR [2, LONG];
;  276        U 0275  3     EXTERNAL            lcl_csid, nod_head : vector [2, long];
;  277        U 0276  3     EXTERNAL ROUTINE    clusutil_configure, clusmsg_ack_please;
;  278        U 0277  3     out_dsc [0] = 16; out_dsc [1] = lcl_buf;
;  279        U 0278  3     IF $trnlog (lognam=%ASCID 'OPC$INCREASE_STRESS', rsllen=out_dsc, rslbuf=out_dsc) EQL ss$_normal
;  280        U 0279  3     THEN
;  281        U 0280  3         BEGIN
;  282        U 0281  3         IF clusutil_configure ()                    ! If true, then a node has come or gone
;  283        U 0282  3         THEN
;  284        U 0283  3             BEGIN
;  285        U 0284  3             LOCAL       nod : $ref_bblock;
;  286        U 0285  3             nod = .nod_head [0];
;  287        U 0286  3             WHILE .nod NEQ nod_head [0]
;  288        U 0287  3             DO
;  289        U 0288  3                 BEGIN
;  290        U 0289  3                 IF .nod [nod_l_node_csid] NEQ .lcl_csid
;  291        U 0290  3                 THEN
;  292        U 0291  3                     BEGIN
;  293        U 0292  3                     nod [nod_v_ack_pend] = false;
;  294        U 0293  3                     CLUSMSG_ACK_PLEASE (.nod);
;  295        U 0294  3                     END;
;  296        U 0295  3                 nod = .nod [nod_l_flink];
;  297        U 0296  3                 END;
;  298        U 0297  3             END;
;  299        U 0298  3         END;
;  300        U 0299  3     END;
;  301          0300  3     %FI                                             ! End of conditionally compiled message codes.
;  302          0301  3
;  303          0302  3     !
;  304          0303  3     ! For debugging, check pool for corruption.  N.B. this puts a severe load on the system at IPL 11!
;  305          0304  3     !
;  306        L 0305  3     %IF %VARIANT GTR 0                              ! Only include if /VARIANT specified
;  307        U 0306  3     %THEN
;  308        U 0307  3     BEGIN
;  309        U 0308  3     LOCAL               lcl_buf : $bvector [16], out_dsc : VECTOR [2, LONG];
;  310        U 0309  3     out_dsc [0] = 16; out_dsc [1] = lcl_buf;
;  311        U 0310  3     IF $trnlog (lognam=%ASCID 'OPC$CHECK_POOL', rsllen=out_dsc, rslbuf=out_dsc) EQL ss$_normal
;  312        U 0311  3     THEN
;  313        U 0312  3         BEGIN
;  314        U 0313  3         LOCAL
;  315        U 0314  3             arglist : VECTOR [2, LONG];
;  316        U 0315  3         EXTERNAL ROUTINE
;  317        U 0316  3             monitor_pool;
;  318        U 0317  3         arglist [0] = 1;
;  319        U 0318  3         arglist [1] = %X'1111';
;  320        U 0319  3         $CMKRNL (routin=monitor_pool, arglst=arglist);
;  321        U 0320  3         END;
;  322        U 0321  3     END;
;  323          0322  3     %FI                                             ! End of conditionally compiled debugging code.
;  324          0323  3
;  325          0324  3     !
```

```
;  326     0325  3     ! Dispatch the request to the proper handler.  Some messages do not come through $SNDOPR,
;  327     0326  3     ! and require special treatment.  For example, device on/offline messages are sent via a
;  328     0327  3     ! call to EXE$SNDEVMSG, and are in a different format from most of the other known message types.
;  329     0328  3     !
;  330     0329  3     SELECTONEU .REQUEST_BUFFER [MSGTYPE] OF
;  331     0330  3         SET
;  332     0331  3         [MSG$_CLUMBX] :         CLUMBX_HANDLER (REQUEST_DESC);
;  333     0332  3         [MSG$_DEVOFFLIN] :      DEVICE_HANDLER (REQUEST_DESC);
;  334     0333  3         [MSG$_DEVONLIN] :       DEVICE_HANDLER (REQUEST_DESC);
;  335     0334  3         [MSG$_DEVOFFLINX] :     DEVICE_HANDLER (REQUEST_DESC);
;  336     0335  3         [MSG$_WRONGVOL] :       DEVICE_HANDLER (REQUEST_DESC);
;  337     0336  3         [MSG$_DEVWRTLCK] :      DEVICE_HANDLER (REQUEST_DESC);
;  338     0337  3         [MSG$_MVCOMPLETE] :     DEVICE_HANDLER (REQUEST_DESC);
;  339     0338  3         [MSG$_MVABORTED] :      DEVICE_HANDLER (REQUEST_DESC);
;  340     0339  3         [MSG$_UDA50MVER] :      DEVICE_HANDLER (REQUEST_DESC);
;  341     0340  3         [MSG$_DUPUNITNO] :      DEVICE_HANDLER (REQUEST_DESC);
;  342     0341  3         [MSG$_TM78MVER] :       DEVICE_HANDLER (REQUEST_DESC);
;  343     0342  3         [MSG$_RC25MVER] :       DEVICE_HANDLER (REQUEST_DESC);
;  344     0343  3         [MSG$_RDRXMVER] :       DEVICE_HANDLER (REQUEST_DESC);
;  345     0344  3         [MSG$_TU81MVER] :       DEVICE_HANDLER (REQUEST_DESC);
;  346     0345  3         [MSG$_MAYAMVER] :       DEVICE_HANDLER (REQUEST_DESC);
;  347     0346  3
;  348     0347  4         [MSG$_OPRQST] : BEGIN
;  349     0348  4                         !
;  350     0349  4                         ! Dispatch the request to the proper handler.
;  351     0350  4                         !
;  352     0351  4                         CASE .REQUEST_BUFFER [RQSTCODE]
;  353     0352  4                             FROM  0  TO  OPC$_X_REQUEST_END_MARK-1  OF
;  354     0353  4                             SET
;  355     0354  4                             !
;  356     0355  4                             ! The following request types are in the old format.
;  357     0356  4                             !
;  358     0357  4                             [OPC$_RQ_TERME] :       TERME_HANDLER (REQUEST_DESC);
;  359     0358  4                             [OPC$_RQ_LOGI] :        LOGI_HANDLER (REQUEST_DESC);
;  360     0359  4                             [OPC$_RQ_RQST] :        RQST_HANDLER (REQUEST_DESC);
;  361     0360  4                             [OPC$_RQ_REPLY] :       RPLY_HANDLER (REQUEST_DESC);
;  362     0361  4                             [OPC$_RQ_CANCEL] :      CNCL_HANDLER (REQUEST_DESC);
;  363     0362  4                             [OPC$_RQ_STATUS] :      STS_HANDLER (REQUEST_DESC);
;  364     0363  4                             [OPC$_RQ_SECURITY] :    SEC0_HANDLER (REQUEST_DESC);
;  365     0364  4                             !
;  366     0365  4                             ! The following request types are in the new format.
;  367     0366  4                             !
;  368     0367  4                             [OPC$_X_OPRENABLE] :    OPRENABLE_HANDLER (REQUEST_DESC);
;  369     0368  4                             [OPC$_X_LOGFILE] :      LOGFILE_HANDLER (REQUEST_DESC);
;  370     0369  4                             [OPC$_X_REQUEST] :      REQUEST_HANDLER (REQUEST_DESC);
;  371     0370  4                             [OPC$_X_REPLY] :        REPLY_HANDLER (REQUEST_DESC);
;  372     0371  4                             [OPC$_X_CANCEL] :       CANCEL_HANDLER (REQUEST_DESC);
;  373     0372  4                             [OPC$_X_STATUS] :       STATUS_HANDLER (REQUEST_DESC);
;  374     0373  4                             [OPC$_X_SHUTDOWN] :     SHUTDOWN_HANDLER (REQUEST_DESC);
;  375     0374  4                             [OPC$_X_SECURITY] :     SECURITY_HANDLER (REQUEST_DESC);
;  376     0375  4                             !
;  377     0376  4                             ! Debug handler, comment out when not necessary
;  378     0377  4                             !
;  379     0378  4                         !   [OPC$_X_DEBUG] :        DEBUG_HANDLER (REQUEST_DESC);
;  380     0379  4                             !
;  381     0380  4                             ! Requests for cluster-related messages.
;  382     0381  4                             !
```

```
;  383          0382  4                        [OPC$_X_CLUSMSG] :      CLUSMSG_HANDLER (REQUEST_DESC);
;  384          0383  4                                                !
;  385          0384  4                                                ! Let the unknown message handler figure out what to do with it.
;  386          0385  4                                                !
;  387          0386  4                        [INRANGE,OUTRANGE] :    UNKNOWN_HANDLER (REQUEST_DESC);
;  388          0387  4                        TES;
;  389          0388  3                    END;
;  390          0389  3
;  391          0390  4            [OTHERWISE] :   BEGIN
;  392          0391  4                                                !
;  393          0392  4                                                ! This is an unknown message type.  Let the unknown message handler log it.
;  394          0393  4                                                !
;  395          0394  4                                                UNKNOWN_HANDLER (REQUEST_DESC);
;  396          0395  3                                                END;
;  397          0396  3            TES;
;  398          0397  2    END;
;  399          0398  1 END;                                                    ! End of OPCOM_MAIN


                                                                        .TITLE  OPC$OPCOMMAIN
                                                                        .IDENT  \V04-000\

                                                                        .PSECT  $PLIT$,NOWRT,NOEXE,2

4F  42  4C  49  41  4D  5F  50  4D  55  44  24  43  50  4F  00000 P.AAB:   .ASCII  \OPC$DUMP_MAILBOX\                                    ;
                                                    58  0000F
                                                  010E0010  00010 P.AAA:   .LONG   17694736
                                                  00000000' 00014          .ADDRESS P.AAB
64  65  76  69  65  63  65  72  20  64  72  6F  63  65  52  00018 P.AAD:   .ASCII  \Record received in mailbox\<0><0>                    ;
      00  00  78  6F  62  6C  69  61  6D  20  6E  69  20  00027
                                                  010E001A  00034 P.AAC:   .LONG   17694746
                                                  00000000' 00038          .ADDRESS P.AAD                                           ;

                                                                        .PSECT  $GLOBAL$,NOEXE,2

                                                  00000000  00000 FINISHED::
                                                                        .LONG   0                                                  ;

                                                                        .EXTRN  DUMP_LOG_FILE, SHARE_FAO_BUFFER                    ;
                                                                        .EXTRN  OPCOM_INIT, TIME_STAMP
                                                                        .EXTRN  WRITE_LOG_FILE, UNKNOWN_HANDLER
                                                                        .EXTRN  CLUMBX_HANDLER, DEVICE_HANDLER
                                                                        .EXTRN  TERME_HANDLER, LOGI_HANDLER
                                                                        .EXTRN  RQST_HANDLER, RPLY_HANDLER
                                                                        .EXTRN  CNCL_HANDLER, STS_HANDLER
                                                                        .EXTRN  SECU_HANDLER, OPRENABLE_HANDLER
                                                                        .EXTRN  LOGFILE_HANDLER
                                                                        .EXTRN  REQUEST_HANDLER
                                                                        .EXTRN  REPLY_HANDLER, CANCEL_HANDLER
                                                                        .EXTRN  STATUS_HANDLER, SHUTDOWN_HANDLER
                                                                        .EXTRN  SECURITY_HANDLER
                                                                        .EXTRN  CLUSMSG_HANDLER
                                                                        .EXTRN  GLOBAL_STATUS, OPER_MBX_CHAN
                                                                        .EXTRN  SYS$QIOW, LIB$STOP
                                                                        .EXTRN  SYS$TRNLOG

                                                                        .PSECT  $CODE$,NOWRT,2
```

```
                                001C 00000          .ENTRY  OPCOM_MAIN, Save R2,R3,R4                       ; 0097
                          54    0000G CF   9E 00002  MOVAB   GLOBAL_STATUS, R4
                          5E    F5D8  CE   9E 00007  MOVAB   -2600(SP), SP
                    0000G CF          00   FB 0000C  CALLS   #0, OPCOM_INIT                                 ; 0194
                                1A    AE   B4 00011  CLRW    REQUEST_DESC+2                                  ; 0199
                             1C AE    20   AE   9E 00014  MOVAB   REQUEST_BUFFER, REQUEST_DESC+4           ; 0201
                                01    0000' CF   E9 00019  1$:    BLBC    FINISHED, 2$                       ; 0206
                                                04 0001E  RET
                                64      40   8F   8A 0001F  2$:    BICB2   #64, GLOBAL_STATUS                 ; 0212
                       05                64   05   E1 00023  BBC     #5, GLOBAL_STATUS, 3$                  ; 0213
                    0000G CF            00   FB 00027  CALLS   #0, TIME_STAMP                              ; 0215
                                                7E   7C 0002C  3$:    CLRQ    -(SP)                          ; 0226
                                                7E   7C 0002E  CLRQ    -(SP)
                                7E      0A00  8F   3C 00030  MOVZWL  #2560, -(SP)
                                          34  AE   9F 00035  PUSHAB  REQUEST_BUFFER
                                                7E   7C 00038  CLRQ    -(SP)
                                          F8  AD   9F 0003A  PUSHAB  IOSB
                                          31  DD   9F 0003D  PUSHL   #49
                                7E      0000G CF   3C 0003F  MOVZWL  OPER_MBX_CHAN, -(SP)
                                          03  DD   00044  PUSHL   #3
                          00000000G       00   0C   FB 00046  CALLS   #12, SYS$QIOW
                                          53   50   D0 0004D  MOVL    R0, STATUS
                                          10   53   E9 00050  BLBC    STATUS, 4$
                                          53     F8 AD   3C 00053  MOVZWL  IOSB, STATUS                     ; 0233
                                          13   53   E8 00057  BLBS    STATUS, 5$
                          00000870        8F   53   D1 0005A  CMPL    STATUS, #2160                       ; 0235
                                               0A   13 00061  BEQL    5$
                                               53   DD 00063  4$:    PUSHL   STATUS                      ; 0237
                          00000000G       00   01   FB 00065  CALLS   #1, LIB$STOP
                                               04 0006C  RET
                                64      40   8F   88 0006D  5$:    BISB2   #64, GLOBAL_STATUS                 ; 0244
                                18    AE     FA AD   B0 00071  MOVW    IOSB+2, REQUEST_DESC                ; 0249
                                      6E           10   D0 00076  MOVL    #16, OUT_DSC                        ; 0258
                                04    AE     08   AE   9E 00079  MOVAB   LCL_BUF, OUT_DSC+4                 ; 0259
                                               7E   7C 0007E  CLRQ    -(SP)                               ; 0260
                                               7E   D4 00080  CLRL    -(SP)
                                          0C   AE   9F 00082  PUSHAB  OUT_DSC
                                          10   AE   9F 00085  PUSHAB  OUT_DSC
                                      0000' CF   9F 00088  PUSHAB  P.AAA
                          00000000G       00   06   FB 0008C  CALLS   #6, SYS$TRNLOG
                                          01   50   D1 00093  CMPL    R0, #1
                                               0C   12 00096  BNEQ    6$
                                      0000' CF   9F 00098  PUSHAB  P.AAC                                     ; 0262
                                          1C   AE   9F 0009C  PUSHAB  REQUEST_DESC
                                0000G CF          02   FB 0009F  CALLS   #2, DUMP_LOG_FILE
                                      52    20   AE   3C 000A4  6$:    MOVZWL  REQUEST_BUFFER, R2          ; 0329
                                0059  8F          52   B1 000A8  CMPW    R2, #89                             ; 0331
                                               0A   12 000AD  BNEQ    7$
                                          18   AE   9F 000AF  PUSHAB  REQUEST_DESC
                                0000G CF          01   FB 000B2  CALLS   #1, CLUMBX_HANDLER
                                               66   11 000B7  BRB     9$
                                      05          52   B1 000B9  7$:    CMPW    R2, #5                      ; 0332
                                               59   13 000BC  BEQL    8$
                                      07          52   B1 000BE  CMPW    R2, #7                      ; 0333
                                               54   13 000C1  BEQL    8$
                                0050  8F          52   B1 000C3  CMPW    R2, #80                     ; 0334
```

```
                                        4D  13 000C8          BEQL    8$
                                0051    52  B1 000CA          CMPW    R2, #81                                         ;0335
                                        46  13 000CF          BEQL    8$
                                0052    52  B1 000D1          CMPW    R2, #82                                         ;0336
                                        3F  13 000D6          BEQL    8$
                                0054    52  B1 000D8          CMPW    R2, #84                                         ;0337
                                        38  13 000DD          BEQL    8$
                                0055    52  B1 000DF          CMPW    R2, #85                                         ;0338
                                        31  13 000E4          BEQL    8$
                                0057    52  B1 000E6          CMPW    R2, #87                                         ;0339
                                        2A  13 000EB          BEQL    8$
                                0058    52  B1 000ED          CMPW    R2, #88                                         ;0340
                                        23  13 000F2          BEQL    8$
                                005A    52  B1 000F4          CMPW    R2, #90                                         ;0341
                                        1C  13 000F9          BEQL    8$
                                005D    52  B1 000FB          CMPW    R2, #93                                         ;0342
                                        15  13 00100          BEQL    8$
                                005E    52  B1 00102          CMPW    R2, #94                                         ;0343
                                        0E  13 00107          BEQL    8$
                                005F    52  B1 00109          CMPW    R2, #95                                         ;0344
                                        07  13 0010E          BEQL    8$
                                0060    52  B1 00110          CMPW    R2, #96                                         ;0345
                                        0A  12 00115          BNEQ    10$
                            18      AE  9F 00117 8$:          PUSHAB  REQUEST_DESC
                        0000G   CF      01  FB 0011A          CALLS   #1, DEVICE_HANDLER
                                        7D  11 0011F 9$:      BRB     20$
                                08      52  B1 00121 10$:     CMPW    R2, #8                                          ;0347
                                        31  12 00124          BNEQ    12$
                         15             00  46 AE  8F 00126  CASEB   REQUEST_BUFFER+38, #0, #21                       ;0351
                0043    0039            002F    00CF    0012B 11$:  .WORD   34$-11$,-
                006B    0061            0057    004D    00133               13$-11$,-
                007F    0075            00CF    00CF    0013B               14$-11$,-
                00A7    009D            0093    0089    00143               15$-11$,-
                00C5    00BB            00CF    00B1    0014B               16$-11$,-
                                        00CF    00CF    00153               17$-11$,-
                                                                            18$-11$,-
                                                                            19$-11$,-
                                                                            34$-11$,-
                                                                            34$-11$,-
                                                                            21$-11$,-
                                                                            22$-11$,-
                                                                            23$-11$,-
                                                                            24$-11$,-
                                                                            25$-11$,-
                                                                            26$-11$,-
                                                                            28$-11$,-
                                                                            34$-11$,-
                                                                            30$-11$,-
                                                                            32$-11$,-
                                                                            34$-11$,-
                                                                            34$-11$
                                    00A0    31 00157 12$:     BRW     34$                                             ;0386
                            18      AE  9F 0015A 13$:         PUSHAB  REQUEST_DESC                                    ;0357
                0000G   CF              01  FB 0015D          CALLS   #1, TERME_HANDLER
                                        76  11 00162          BRB     27$
                            18      AE  9F 00164 14$:         PUSHAB  REQUEST_DESC                                    ;0358
                0000G   CF              01  FB 00167          CALLS   #1, LOGI_HANDLER
```

```
                        76  11 0016C          BRB     29$                                                     ;
                     18 AE  9F 0016E 15$:     PUSHAB  REQUEST_DESC                                            ; 0359
            0000G    CF     01 FB 00171       CALLS   #1, RQST_HANDLER                                        ;
                        76  11 00176          BRB     31$                                                     ;
                     18 AE  9F 00178 16$:     PUSHAB  REQUEST_DESC                                            ; 0360
            0000G    CF     01 FB 0017B       CALLS   #1, RPLY_HANDLER                                        ;
                        76  11 00180          BRB     33$                                                     ;
                     18 AE  9F 00182 17$:     PUSHAB  REQUEST_DESC                                            ; 0361
            0000G    CF     01 FB 00185       CALLS   #1, CNCL_HANDLER                                        ;
                        76  11 0018A          BRB     35$                                                     ;
                     18 AE  9F 0018C 18$:     PUSHAB  REQUEST_DESC                                            ; 0362
            0000G    CF     01 FB 0018F       CALLS   #1, STS_HANDLER                                         ;
                        6C  11 00194          BRB     35$                                                     ;
                     18 AE  9F 00196 19$:     PUSHAB  REQUEST_DESC                                            ; 0363
            0000G    CF     01 FB 00199       CALLS   #1, SECO_HANDLER                                        ;
                        62  11 0019E 20$:     BRB     35$                                                     ;
                     18 AE  9F 001A0 21$:     PUSHAB  REQUEST_DESC                                            ; 0367
            0000G    CF     01 FB 001A3       CALLS   #1, OPRENABLE_HANDLER                                   ;
                        58  11 001A8          BRB     35$                                                     ;
                     18 AE  9F 001AA 22$:     PUSHAB  REQUEST_DESC                                            ; 0368
            0000G    CF     01 FB 001AD       CALLS   #1, LOGFILE_HANDLER                                     ;
                        4E  11 001B2          BRB     35$                                                     ;
                     18 AE  9F 001B4 23$:     PUSHAB  REQUEST_DESC                                            ; 0369
            0000G    CF     01 FB 001B7       CALLS   #1, REQUEST_HANDLER                                     ;
                        44  11 001BC          BRB     35$                                                     ;
                     18 AE  9F 001BE 24$:     PUSHAB  REQUEST_DESC                                            ; 0370
            0000G    CF     01 FB 001C1       CALLS   #1, REPLY_HANDLER                                       ;
                        3A  11 001C6          BRB     35$                                                     ;
                     18 AE  9F 001C8 25$:     PUSHAB  REQUEST_DESC                                            ; 0371
            0000G    CF     01 FB 001CB       CALLS   #1, CANCEL_HANDLER                                      ;
                        30  11 001D0          BRB     35$                                                     ;
                     18 AE  9F 001D2 26$:     PUSHAB  REQUEST_DESC                                            ; 0372
            0000G    CF     01 FB 001D5       CALLS   #1, STATUS_HANDLER                                      ;
                        26  11 001DA 27$:     BRB     35$                                                     ;
                     18 AE  9F 001DC 28$:     PUSHAB  REQUEST_DESC                                            ; 0373
            0000G    CF     01 FB 001DF       CALLS   #1, SHUTDOWN_HANDLER                                    ;
                        1C  11 001E4 29$:     BRB     35$                                                     ;
                     18 AE  9F 001E6 30$:     PUSHAB  REQUEST_DESC                                            ; 0374
            0000G    CF     01 FB 001E9       CALLS   #1, SECURITY_HANDLER                                    ;
                        12  11 001EE 31$:     BRB     35$                                                     ;
                     18 AE  9F 001F0 32$:     PUSHAB  REQUEST_DESC                                            ; 0382
            0000G    CF     01 FB 001F3       CALLS   #1, CLUSMSG_HANDLER                                     ;
                        08  11 001F8 33$:     BRB     35$                                                     ; 0329
                     18 AE  9F 001FA 34$:     PUSHAB  REQUEST_DESC                                            ; 0394
            0000G    CF     01 FB 001FD       CALLS   #1, UNKNOWN_HANDLER                                     ;
                     FE14   31  00202 35$:    BRW     1$                                                      ; 0206
                            04  00205          RET                                                             ; 0398

; Routine Size:  518 bytes,    Routine Base.  $CODE$ + 0000

;  400          0399  1
;  401          0400  1 END
;  402          0401  0 ELUDOM                                  ! End of OPCOMMAIN
```

```
;                                   PSECT SUMMARY
;
;       Name                    Bytes                           Attributes
;
;  $GLOBAL$                         4  NOVEC,  WRT,  RD ,NOEXE,NOSHR,  LCL,  REL,  CON,NOPIC,ALIGN(2)
;  $PLIT$                          60  NOVEC,NOWRT,  RD ,NOEXE,NOSHR,  LCL,  REL,  CON,NOPIC,ALIGN(2)
;  $CODE$                         518  NOVEC,NOWRT,  RD ,  EXE,NOSHR,  LCL,  REL,  CON,NOPIC,ALIGN(2)
;
;                                 Library Statistics
;
;                                        -------- Symbols --------      Pages       Processing
;       File                             Total    Loaded   Percent      Mapped      Time
;
;  _$255$DUA28:[SYSLIB]LIB.L32;1          18619        36         0        1000        00:01.8
;  _$255$DUA28:[OPCOM.OBJ]OPCOMLIB.L32;1    633        20         3          43        00:00.9
;
;                                   COMMAND QUALIFIERS
;
;       BLISS/CHECK=(FIELD,INITIAL,OPTIMIZE)/LIS=LIS$:OPCOMMAIN/OBJ=OBJ$:OPCOMMAIN MSRC$:OPCOMMAIN/UPDATE=(ENH$:OPCOMMAIN)
;
; Size:          518 code + 64 data bytes
; Run Time:          00:12.6
; Elapsed Time:      00:35.0
; Lines/CPU Min:      1906
; Lexemes/CPU-Min:  11576
; Memory Used:  180 pages
; Compilation Complete
```

LOGFILE
LIS

OPCOMDEF
LIS

OPCOMDATA
LIS

OPCOMINI
LIS

OPCOMUTIL
LIS

OPCOMLIB
LIS

OPCDEFTMP
LIS

OPCOMMAIN
LIS

OPCOMOLD
LIS

OPCOMRPLY
LIS

OPCCRASH
LIS

OPERUTIL
LIS

OPCOMROST
LIS