```
RRRRRRRRRRRR     TTTTTTTTTTTTTTT  PPPPPPPPPPPP        AAAAAAAAA     DDDDDDDDDDD
RRRRRRRRRRRR     TTTTTTTTTTTTTTT  PPPPPPPPPPPP        AAAAAAAAA     DDDDDDDDDDD
RRRRRRRRRRRR     TTTTTTTTTTTTTTT  PPPPPPPPPPPP        AAAAAAAAA     DDDDDDDDDDD
RRR          RRR        TTT        PPP          PPP  AAA         AAA  DDD         DDD
RRR          RRR        TTT        PPP          PPP  AAA         AAA  DDD         DDD
RRR          RRR        TTT        PPP          PPP  AAA         AAA  DDD         DDD
RRR          RRR        TTT        PPP          PPP  AAA         AAA  DDD         DDD
RRR          RRR        TTT        PPP          PPP  AAA         AAA  DDD         DDD
RRR          RRR        TTT        PPP          PPP  AAA         AAA  DDD         DDD
RRRRRRRRRRRR            TTT        PPPPPPPPPPPP      AAA         AAA  DDD         DDD
RRRRRRRRRRRR            TTT        PPPPPPPPPPPP      AAA         AAA  DDD         DDD
RRRRRRRRRRRR            TTT        PPPPPPPPPPPP      AAA         AAA  DDD         DDD
RRR    RRR              TTT        PPP               AAAAAAAAAAAAAAA  DDD         DDD
RRR    RRR              TTT        PPP               AAAAAAAAAAAAAAA  DDD         DDD
RRR    RRR              TTT        PPP               AAAAAAAAAAAAAAA  DDD         DDD
RRR        RRR          TTT        PPP               AAA         AAA  DDD         DDD
RRR        RRR          TTT        PPP               AAA         AAA  DDD         DDD
RRR        RRR          TTT        PPP               AAA         AAA  DDD         DDD
RRR          RRR        TTT        PPP               AAA         AAA  DDDDDDDDDDDD
RRR          RRR        TTT        PPP               AAA         AAA  DDDDDDDDDDDD
RRR          RRR        TTT        PPP               AAA         AAA  DDDDDDDDDDDD
```

```
VV      VV  MM      MM    SSSSSSSS  RRRRRRRR    TTTTTTTTTT
VV      VV  MM      MM    SSSSSSSS  RRRRRRRR    TTTTTTTTTT
VV      VV  MMMM  MMMM  SS          RR      RR      TT
VV      VV  MMMM  MMMM  SS          RR      RR      TT
VV      VV  MM  MM  MM  SS          RR      RR      TT
VV      VV  MM  MM  MM  SS          RR      RR      TT
VV      VV  MM      MM    SSSSSS    RRRRRRRR        TT
VV      VV  MM      MM    SSSSSS    RRRRRRRR        TT
VV      VV  MM      MM          SS  RR   RR         TT
VV      VV  MM      MM          SS  RR   RR         TT
  VV  VV    MM      MM          SS  RR    RR        TT        ....
  VV  VV    MM      MM          SS  RR    RR        TT        ....
    VV      MM      MM  SSSSSSSS    RR      RR      TT        ....
    VV      MM      MM  SSSSSSSS    RR      RR      TT        ....

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     1          .TITLE  VMSRT - VMS PROTOCOL WITH CTERM HOOKS
0000     2          .IDENT  'V04-000'
0000     3 ;
0000     4 ;*****************************************************************************
0000     5 ;*                                                                           *
0000     6 ;*  COPYRIGHT (c) 1978, 1980, 1982, 1984 BY                                  *
0000     7 ;*  DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASSACHUSETTS.                   *
0000     8 ;*  ALL RIGHTS RESERVED.                                                     *
0000     9 ;*                                                                           *
0000    10 ;*  THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND COPIED    *
0000    11 ;*  ONLY IN  ACCORDANCE WITH  THE  TERMS  OF  SUCH  LICENSE  AND WITH THE   *
0000    12 ;*  INCLUSION OF THE ABOVE COPYRIGHT NOTICE. THIS SOFTWARE OR  ANY  OTHER    *
0000    13 ;*  COPIES THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY    *
0000    14 ;*  OTHER PERSON.  NO TITLE TO AND OWNERSHIP OF  THE  SOFTWARE IS  HEREBY    *
0000    15 ;*  TRANSFERRED.                                                             *
0000    16 ;*                                                                           *
0000    17 ;*  THE INFORMATION IN THIS SOFTWARE IS  SUBJECT TO CHANGE WITHOUT NOTICE    *
0000    18 ;*  AND  SHOULD  NOT  BE  CONSTRUED AS  A COMMITMENT BY DIGITAL EQUIPMENT   *
0000    19 ;*  CORPORATION.                                                             *
0000    20 ;*                                                                           *
0000    21 ;*  DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE  OR  RELIABILITY OF ITS    *
0000    22 ;*  SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL.                  *
0000    23 ;*                                                                           *
0000    24 ;*                                                                           *
0000    25 ;*****************************************************************************
0000    26 ;
0000    27
0000    28 ;++
0000    29 ;
0000    30 ; FACILITY:
0000    31 ;
0000    32 ;       SET HOST, aka RTPAD
0000    33 ;
0000    34 ; ABSTRACT:
0000    35 ;
0000    36 ;       This module handles all of the VMS specific remote terminal
0000    37 ;       protocol. It also contains hooks for the CTERM protocol.
0000    38 ;
0000    39 ; ENVIRONMENT:  VMS user mode
0000    40 ;
0000    41 ;--
0000    42 ;
0000    43 ; AUTHOR: Jake VanNoy, CREATION DATE: 14-Jan-1982
0000    44 ; ORIGINAL AUTHOR: W M Cardoza
0000    45 ;
0000    46 ; MODIFIED BY:
0000    47 ;
0000    48 ;       V03-008 JLV0364         Jake VanNoy             11-JUL-1984
0000    49 ;               Add code to skip $SETIMR in ^Y routine if SETIMR
0000    50 ;               already pending.
0000    51 ;               Fix broken branch due to change in QUIT.
0000    52 ;
0000    53 ;       V03-007 MHB0135         Mark Bramhall           10-Apr-1984
0000    54 ;               Fix register conventions when calling VMS_INDREAD.
0000    55 ;
0000    56 ;       V03-006 JLV0331         Jake VanNoy             28-FEB-1984
0000    57 ;               Integrate trace facility into permanent code.
```

```
                     0000     58 ;                Add secondary status to REM-F-NETMBX failure to
                     0000     59 ;                show why link was blown away.
                     0000     60 ;
                     0000     61 ;        V03-005 JLV0325         Jake VanNoy             10-JAN-1984
                     0000     62 ;                Add terminator in fake IOSB generated by VMS_INDREAD.
                     0000     63 ;
                     0000     64 ;        V03-004 JLV0296         Jake VanNoy             28-JUL-1983
                     0000     65 ;                add OUTBANDABO channel.
                     0000     66 ;
                     0000     67 ;        V03-003 JLV0266         Jake VanNoy             26-MAY-1983
                     0000     68 ;                Restore code to prevent purge typeahead on "Username:"
                     0000     69 ;
                     0000     70 ;        V03-002 MHB0092         Mark Bramhall            3-Mar-1983
                     0000     71 ;                Moved MAXMSG to $RTPADDEF.
                     0000     72 ;
                     0000     73 ;        V03-001 JLV0            Jake VanNoy             14-Jan-1983
                     0000     74 ;                Module created from VMS protocol code previously
                     0000     75 ;                held in RTPAD module. Hook for CTERM were included.
                     0000     76 ;
                     0000     77 ;**
                     0000     78
                     0000     79          .SBTTL  DECLARATIONS
                     0000     80 ;
                     0000     81 ; INCLUDE FILES:
                     0000     82 ;
                     0000     83
                     0000     84          $DVIDEF
                     0000     85          $RDPDEF
                     0000     86          $RTPADDEF
                     0000     87          $TSADEF
                     0000     88
                     0000     89 ;
                     0000     90 ; EQUATED SYMBOLS:
                     0000     91 ;
00000001             0000     92 ASSEM_TRACE = 1                     ; INCLUDE TRACE CODE
                     0000     93
00000026             0000     94 AST$T_BUF = CTP$B_PRO_MSGTYPE ; ***
                     0000     95
                     0000     96 ;
                     0000     97 ;        OP CODES
                     0000     98 ;
00000001             0000     99          OP_READ = 1
00000002             0000    100          OP_WRITE = 2
00000003             0000    101          OP_SETMODE = 3
00000004             0000    102          OP_SENSEMODE = 4
00000005             0000    103          OP_CANCEL = 5
00000006             0000    104          OP_BRDCST = 6
00000100             0000    105          OP_PRMPT = ^X100
                     0000    106
```

```
00000000     108 .PSECT  RTPAD,NOWRT
     0000    109
     0000    110         .SBTTL VMSRT - START UP VMS TO VMS PROTOCOL
     0000    111 ;++
     0000    112 ; FUNCTIONAL DESCRIPTION:
     0000    113 ;
     0000    114 ;       PERFORMS INITIALIZATION FUNCTIONS FOR VMS TO VMS PROTOCOL
     0000    115 ;
     0000    116 ; CALLING SEQUENCE:
     0000    117 ;
     0000    118 ;       CALLS   #0,VMSRT
     0000    119 ;
     0000    120 ; INPUT PARAMETERS:
     0000    121 ;
     0000    122 ;       NONE
     0000    123 ;
     0000    124 ; IMPLICIT INPUTS:
     0000    125 ;
     0000    126 ;       CHANNEL NUMBERS, ETC.
     0000    127 ;
     0000    128 ; OUTPUT PARAMETERS:
     0000    129 ;
     0000    130 ;       NONE
     0000    131 ;
     0000    132 ; IMPLICIT OUTPUTS:
     0000    133 ;
     0000    134 ;       NONE
     0000    135 ;
     0000    136 ; COMPLETION CODES:
     0000    137 ;
     0000    138 ;
     0000    139 ; SIDE EFFECTS:
     0000    140 ;
     0000    141 ;       SETS UP MAILBOX READS FOR LINK, UNSOLICITED TERMINAL INPUT
     0000    142 ;       ENABLES ^C, ^Y  AST'S
     0000    143 ;
     0000    144 ;--
     0000    145 ;
     0000    146
               000C  0000    147 .ENTRY  VMSRT,  ^M<R2,R3>
                     0002    148
 00000000'EF   B5    0002    149         TSTW    FINALACS                ;
          13   13    0008    150         BEQL    10$                     ; Branch if no access control string
                     000A    151         $PUTMSG_S MSGVEC = ACSIGNORE    ; Warn him we are ignoring it
                     001D    152 10$:
                     001D    153         ;
                     001D    154         ; Assume CTERM BIND ACCEPT message
                     001D    155         ;
50    00000000'EF    9E    001D    156         MOVAB   CT_BIND_ACC_MSG,R0      ; Address of buffer (BIND ACCEPT)
   51    0000'8F    3C    0024    157         MOVZWL  #CT_BIND_MSGLEN,R1      ; Length of message
                     0029    158
  50 00000000'EF    E8    0029    159         BLBS    CTERM_FLAG,25$          ; Branch if cterm
                     0030    160 ;
                     0030    161 ; *** WHAT USES THIS TERMUNIT GARBAGE??? NOTE CODE IS BUGGY...
                     0030    162 ;
50    00000000'EF    9E    0030    163         MOVAB   DEVNAM,R0               ;ADDRESS OF COUNTED STRING
00000002'EF    01 A0    B0    0037    164         MOVW    1(R0),TERMUNIT+2        ;USE PART OF DEV NAME AS HIGH UNIT
```

```
                50   00000000'EF   C0  003F   165          ADDL2   DEVNAMLEN,R0                ;GET TO LAST CHAR (CONTROLLER)
                 51    60    41  8F  83  0046   166          SUBB3   #^A/A/,(R0),R1              ;GET CONTROLLER NUMBER
                       51    51  9A  004B   167          MOVZBL  R1,R1
                    51   51    04  78  004E   168          ASHL    #4,R1,R1
                00000000'EF   51  A0  0052   169          ADDW2   R1,TERMUNIT                 ;ADD CONTROLLER NUMBER TO UNIT
                                      0059   170          ;
                                      0059   171          ; VAX config (BIND) message
                                      0059   172          ;
                    50   0000'CF   9E  0059   173          MOVAB   W^CHAR_BLOCK,R0             ; Characteristics
                    51   00B2'CF   9E  005E   174          MOVAB   W^CONFIG_CHAR,R1            ; CONFIG characteristics
                         81    80   7D  0063   175          MOVQ    (R0)+,(R1)+                 ;  first 8 bytes
                         81    80   D0  0066   176          MOVL    (R0)+,(R1)+                 ;  last four bytes
                                      0069   177
                50   00000084'EF   9E  0069   178          MOVAB   CONFIG_MSG,R0               ; Address of buffer
                    51   0014'8F   3C  0070   179          MOVZWL  #CONFIG_MSGLEN,R1           ; Length of message
                     00000000'EF   95  0075   180          TSTB    PROTO_ECO                   ; Eco is 0 for 8 bytes of char
                              03   12  007B   181          BNEQ    25$                         ; Long form of charateristics?
                         51   04   C2  007D   182          SUBL    #4,R1                       ; No, old form had 8 bytes of char
                                      0080   183          ;
                                      0080   184          ; Send BIND ACCEPT back to HOST
                                      0080   185          ;
                                      0080   186 25$:
                      52   26 A0   9E  0080   187          MOVAB   AST$T_BUF(R0),R2            ; Address of message
                           093D   30  0084   188          BSBW    WRITE_TO_NET_SYNC           ; Write message to NET
                                      0087   189          ;
                                      0087   190          ; Read from NET mailbox
                                      0087   191          ;
                    50   0036'CF   9E  0087   192          MOVAB   W^LINKMAIL,R0               ; Address of data area
                                      008C   193          $QIO_S  CHAN = MAILCHAN -            ; Link mailbox read
                                      008C   194                  FUNC = #IO$_READVBLK -
                                      008C   195                  IOSB = AST$Q_IOSB(R0) -
                                      008C   196                  ASTADR = ASTHANDLER -
                                      008C   197                  ASTPRM = R0 -
                                      008C   198                  P1 = AST$T_BUF(R0) -
                                      008C   199                  P2 = #40
                                      00B5   200          ONERROR QUIT                        ; Die if error
                                      00DD   201          ;
                                      00DD   202          ; Read from associated terminal mailbox
                                      00DD   203          ;
                07 00000000'EF   E8  00DD   204          BLBS    CTERM_FLAG,30$              ; Branch if CTERM
                  0D4E'CF    00   FB  00E4   205          CALLS   #0,W^UNSMSGDONE             ; Call read routine
                             05   11  00E9   206          BRB     40$                         ; Branch
                                      00EB   207 30$:
                  0000'CF    00   FB  00EB   208          CALLS   #0,W^CTERM_UNSMSGDONE       ; Call read routine
                                      00F0   209 40$:
                                      00F0   210
                                      00F0   211          ;
                                      00F0   212          ; Set up ^Y and assign channels needed for out of band processing
                                      00F0   213          ;
                                      00F0   214          $QIO_S  CHAN = CNTRLCHAN -                  ;WE WILL ALWAYS HANDLE ^Y
                                      00F0   215                  FUNC = #IO$_SETMODE!IO$M_CTRLYAST -
                                      00F0   216                  P1 = CNTRLC_AST -
                                      00F0   217                  P2 = #IO$M_CTRLYAST
                                      0119   218
                                      0119   219          $ASSIGN_S -                         ;CHANNEL FOR INCLUDE OUT OF BAND AST'S
                                      0119   220                  DEVNAM = TTYDESC, -
                                      0119   221                  CHAN = OUTBANDINC
```

```
                                012E
                                012E            $ASSIGN_S -                     ;CHANNEL FOR EXCLUDE OUT OF BAND AST'S
                                012E                    DEVNAM = TTYDESC, -
                                012E                    CHAN = OUTBANDEXC
                                0143
                                0143            $ASSIGN_S -                     ;CHANNEL FOR ABORT OUT OF BAND AST'S
                                0143                    DEVNAM = TTYDESC, -
                                0143                    CHAN = OUTBANDABO
                                0158
         3E 00000000'EF   E9    0158            BLBC    CTERM_FLAG,99$          ; Branch if not CTERM
                                015F            $ASSIGN_S -
                                015F                    DEVNAM = TTYDESC, -
                                015F                    CHAN = CTRLO_CHAN       ; ^O channel
                                0174
                                0174            $QIOW_S CHAN=CTRLO_CHAN,-
                                0174                    FUNC=#IO$_SETMODE!IO$M_OUTBAND!IO$M_INCLUDE,-
                                0174                    P1=  CTERM_CTRLO_AST,-
                                0174                    P2=  #CTRLO_MASK
                                019D
                                019D    99$:
               0E71       30    019D            BSBW    GETBUF                  ; Get a buffer for reading link
      60   000006DA'EF    9E    01A0            MOVAB   LINKRECV,(R0)           ; Insert state
                                01A7            $QIO_S  CHAN = LINKCHAN -
                                01A7                    FUNC = #IO$_READVBLK -
                                01A7                    IOSB = AST$Q_IOSB(R0) -
                                01A7                    ASTADR = ASTHANDLER -
                                01A7                    ASTPRM = R0 -
                                01A7                    P1 = AST$T_BUF(R0) -
                                01A7                    P2 = #MAXMSG
                                01D4            ONERROR QUIT                    ;
                                01FC    ;
                                01FC    ; The following pice of code was formerly done before reads to net were done
                                01FC    ; This caused disconects sent as a result of driver not found to be lost
                                01FC    ;
 000000F0'EF   00000000'EF   B0 01FC            MOVW    TERMUNIT,INIT_MSG+4+RDP$W_UNIT+AST$T_BUF
                                0207            ;
                                0207            ; Check for CTERM protocol. Note that the VAX protocol now
                                0207            ; sends what is interpretted as an unsolicted data message
                                0207            ; which in turn starts up a process on the other end.
                                0207            ; The CTERM protocol module first expects to read and respond
                                0207            ; with INIT messages before proceeding to the unsolicted
                                0207            ; data message.
                                0207
         15 00000000'EF   E8    0207            BLBS    CTERM_FLAG,100$         ; branch if VAX
      50   000000BE'EF    9E    020E            MOVAB   INIT_MSG,R0             ; Assume VAX protocol
      51   0000000A'8F    D0    0215            MOVL    #INIT_MSGLEN,R1         ; Length of message
           52     26 A0   9E    021C            MOVAB   AST$T_BUF(R0),R2        ; Address of buffer
               07A1       30    0220            BSBW    WRITE_TO_NET_SYNC       ; Write message to NET
                                0223    100$:
                          04    0223            RET                             ;
```

```
                     0224   273          .SBTTL ASTHANDLER - DISPATCH AST'S
                     0224   274 ;++
                     0224   275 ; FUNCTIONAL DESCRIPTION:
                     0224   276 ;
                     0224   277 ;       CALLS THE APPROPRIATE AST HANDLING ROUTINE
                     0224   278 ;
                     0224   279 ; CALLING SEQUENCE:
                     0224   280 ;
                     0224   281 ;       AST
                     0224   282 ;
                     0224   283 ; INPUT PARAMETERS:
                     0224   284 ;
                     0224   285 ;       THE AST PARAMETER IS A POINTER TO THE ADDRESS OF THE AST HANDLING
                     0224   286 ;       ROUTINE
                     0224   287 ;
                     0224   288 ; IMPLICIT INPUTS:
                     0224   289 ;
                     0224   290 ;       NONE
                     0224   291 ;
                     0224   292 ; OUTPUT PARAMETERS:
                     0224   293 ;
                     0224   294 ;       NONE
                     0224   295 ;
                     0224   296 ; IMPLICIT OUTPUTS:
                     0224   297 ;
                     0224   298 ;       NONE
                     0224   299 ;
                     0224   300 ; COMPLETION CODES:
                     0224   301 ;
                     0224   302 ;
                     0224   303 ; SIDE EFFECTS:
                     0224   304 ;
                     0224   305 ;       NONE
                     0224   306 ;
                     0224   307 ;--
                     0224   308 ;
                     0224   309 ASTHANDLER::
                0000 0224   310          .WORD   0
50    04 AC    D0   0226   311          MOVL    4(AP),R0                ; GET AST PARAMETER
00 B0    00    FB   022A   312          CALLS   #0,@(R0)                ; CALL SERVICE ROUTINE
                04   022E   313          RET
```

```
                                        .SBTTL PROCMSG - PROCESS A LINK MESSAGE
                                ;++
                                ; FUNCTIONAL DESCRIPTION:
                                ;
                                ;       ACT ON A QIO REQUEST PREVIOUSLY RECEIVED ON THE LINK
                                ;
                                ; CALLING SEQUENCE:
                                ;
                                ;       CALLS   #0,PROCMSG
                                ;
                                ; INPUT PARAMETERS:
                                ;
                                ;       R0 IS A POINTER TO AN AST CONTROL BLOCK
                                ;
                                ; IMPLICIT INPUTS:
                                ;
                                ;       READQIO
                                ;       WRITEQIO
                                ;
                                ; OUTPUT PARAMETERS:
                                ;
                                ;       NONE
                                ;
                                ; IMPLICIT OUTPUTS:
                                ;
                                ;       READQIO
                                ;       WRITEQIO
                                ;       RETSTATUS
                                ;
                                ; COMPLETION CODES:
                                ;
                                ; SIDE EFFECTS:
                                ;
                                ;       A QIO TO THE TERMINAL MAY BE PERFORMED.
                                ;
                                ;--
                                ;
                                PROCMSG:
                        00FC    .WORD   ^M<R2,R3,R4,R5,R6,R7>
                                ;
                                ; CTERM module calls CTERM_PROCMSG directly
                                ;
     57   0C A0   3C           MOVZWL  AST$W_OPCODE(R0),R7     ; Fetch internal opcode
                                ;
          28 A0   A8           BISW2   AST$T_BUF+RDP$W_MOD(R0),-
          26 A0                        AST$T_BUF+RDP$W_OPCODE(R0)      ;ADD OP MOD
        56   50   D0           MOVL    R0,R6
                                ;
                                ; Case instruction uses a byte, note that prompt flag
                                ; is in second byte.
                                ;
04   01   57   8F               CASEB   R7,#OP_READ,#OP_SENSEMODE
                        0009'   4$:     .WORD   READMSG-4$
                        00F4'           .WORD   WRITEMSG-4$
                        0162'           .WORD   SETMSG-4$
                        0387'           .WORD   SENSEMSG-4$
```

```
                     04   0249    372          RET
                          024A    373 ;
                          024A    374 ; A READ QIO REQUEST
                          024A    375 READMSG:
 60  0000095B'EF    9E   024A    376          MOVAB   QIODONE,AST$L_STATE(R0) ;NEW STATE
      52   26 A0    3C   0251    377          MOVZWL  AST$T_BUF+RDP$W_OPCODE(R0),R2
                          0255    378
                          0255    379 ; *** Note that the following two instructions are obsolete in a roundabout
                          0255    380 ; sort of way. LOGINOUT is now smart enough (in V4) to not do a purge typeahead
                          0255    381 ; to a remote terminal. Connections to a V3.x system will then have the
                          0255    382 ; type ahead purged.
                          0255    383
 52  000005D7'EF    AA   0255    384          BICW    FIRST_READ, R2          ; Don't purge type ahead on first read
     000005D7'EF    B4   025C    385          CLRW    FIRST_READ              ;  but only on the first read...
                          0262    386
      56   30 A0    3C   0262    387          MOVZWL  AST$T_BUF+RDP$L_TT_BCNT(R0),R6
                50    D4   0266    388          CLRL    R4
                55    D4   0268    389          CLRL    R5
          57  0100 8F    B3   026A    390          BITW    #OP_PRMPT,R7
                0E    13   026F    391          BEQL    10$
        54   39 A0    9E   0271    392          MOVAB   AST$T_BUF+RDP$T_TT_TERM+1(R0),R4         ;TERM MSG SIZE ADDR+1
        55   FF A4    9A   0275    393          MOVZBL  -1(R4),R5       ;TERM MSG SIZE
           54   55    C0   0279    394          ADDL2   R5,R4   ;PROMPT SIZE ADDRESS
           55   84    3C   027C    395          MOVZWL  (R4)+,R5
                          027F    396 10$:
0000001E'EF    38 A0    9A   027F    397          MOVZBL  AST$T_BUF+RDP$T_TT_TERM(RC),RTERMDES    ;TERMINATION CHARS
                04    12   0287    398          BNEQ    30$
                          0289    399 ; USE DEFAULT
                51    D4   0289    400          CLRL    R1
                07    11   028B    401          BRB     40$
                          028D    402 30$:
   51   0000001E'EF    9E   028D    403          MOVAB   RTERMDES,R1
                          0294    404 40$:
00000022'EF    39 A0    9E   0294    405          MOVAB   AST$T_BUF+RDP$T_TT_TERM+1(R0),RTERMDES+4
000005A7'EF    2A A0    D0   029C    406          MOVL    AST$T_BUF+RDP$L_REFID(R0),READQIO
     00000000'EF    95   02A4    407          TSTB    INDFLAG
                77    12   02AA    408          BNEQ    100$                    ; Branch if reading from a file
                          02AC    409 45$:
           53   50    D0   02AC    410          MOVL    R0,R3                   ; Save in case error later
                          02AF    411 ;
                          02AF    412 ; Test for read verify
                          02AF    413 ;
     00000000'8F    E1   02AF    414          BBC     #IO$V_EXTEND,-
        0C 26 A0         02B5    415                  AST$T_BUF+RDP$W_OPCODE(R0),46$  ; Branch if not read verify
                          02B8    416
     04 A4    54   C0   02B8    417          ADDL    R4,4(R4)                ; relocate address
     0C A4    54   C0   02BC    418          ADDL    R4,12(R4)               ; relocate address
     14 A4    54   C0   02C0    419          ADDL    R4,20(R4)               ; relocate address
                          02C4    420 46$:
                          02C4    421          $QIO_S  CHAN = READCHAN -
                          02C4    422                  FUNC = R2 -
                          02C4    423                  IOSB = AST$Q_IOSB(R0) -
                          02C4    424                  ASTADR = ASTHANDLER -
                          02C4    425                  ASTPRM = R0 -
                          02C4    426                  P1 = AST$T_BUF+RDP$T_TT_RDATA+2(R0) -
                          02C4    427                  P2 = R6 -
                          02C4    428                  P3 = AST$T_BUF+RDP$L_TT_TIMOUT(R0) -
```

```
                                 02C4                  P4 = R1 -
                                 02C4                  P5 = R4 -
                                 02C4                  P6 = R5
                                 02EE          IF_NO_QUOTA     QUIT
                 03 50     E8    031A          BLBS    R0,47$                  ; branch if ok
                  0293     31    031D          BRW     QIO_ERR                 ; Handle error
                                 0320  47$:
                  02FE     31    0320          BRW     PROCMSG_EXIT
                                 0323
                                 0323          ;
                                 0323          ; Read from a file, not the terminal
                                 0323          ;
                                 0323
                                 0323  100$:
                    51     DD    0323          PUSHL   R1                      ; Save
         51     3A A0      9E    0325          MOVAB   AST$T_BUF+RDP$T_TT_RDATA+2(R0),R1
                                 0329                                          ; Address for input
            53      56     D0    0329          MOVL    R6,R3                   ; Size for input
                  0C0B     30    032C          BSBW    VMS_INDREAD             ; Try the indirect file
                    51   8ED0    032F          POPL    R1                      ; Save
                  FF77     31    0332          BRW     45$                     ; If routine returns here,
                                 0335                                          ; RMS got EOF, must read from TTY.
                                 0335
                                 0335  ;
                                 0335  ; A WRITE QIO REQUEST
                                 0335  WRITEMSG:
   60     0000095B'EF      9E    0335          MOVAB   QIODONE,AST$L_STATE(R0) ;NEW STATE
            52   26 A0     3C    033C          MOVZWL  AST$T_BUF+RDP$W_OPCODE(R0),R2
            53   30 A0     3C    0340          MOVZWL  AST$T_BUF+RDP$L_TT_BCNT(R0),R3
  000005AB'EF    2A A0     D0    0344          MOVL    AST$T_BUF+RDP$L_REFID(R0),WRITEQIO
                                 034C          $QIO_S  CHAN = WRITECHAN -
                                 034C                  FUNC = R2 -
                                 034C                  IOSB = AST$Q_IOSB(R0) -
                                 034C                  ASTADR = ASTHANDLER -
                                 034C                  ASTPRM = R0 -
                                 034C                  P1 = AST$T_BUF+RDP$T_TT_WDATA(R0) -
                                 034C                  P2 = R3 -
                                 034C                  P4 = AST$T_BUF+RDP$L_TT_CARCON(R0)
                                 0374          IF_NO_QUOTA     QUIT
                  027E     31    03A0          BRW     PROCMSG_EXIT
                                 03A3  ;
                                 03A3  ; A SET MODE QIO REQUEST
                                 03A3  SETMSG:
   26 A0        0000'8F    B3    03A3          BITW    #IO$M_OUTBAND,AST$T_BUF+RDP$W_OPCODE(R0)
                    03     12    03A9          BNEQ    49$
                  009B     31    03AB          BRW     60$
                                 03AE  ; HANDLE AN OUT OF BAND AST REQUEST
            57      50     D0    03AE  49$:    MOVL    R0,R7                   ;SAVE R0
         53     30 A0      9E    03B1          MOVAB   AST$T_BUF+RDP$B_TT_OUTBAND(R0),R3       ;START OF DATA (MASKS)
            04      63     91    03B5          CMPB    (R3),#4                 ;FOR NOW, IT MUST BE A SINGLE LONGWORD
                    03     13    03B8          BEQL    51$
                  0265     31    03BA          BRW     OUTBAND_ERR
  000005CB'EF    01 A3     D1    03BD  51$:    CMPL    1(R3),INCMASK+4
                    31     13    03C5          BEQL    53$                     ;DON'T BOTHER WITH THE QIO
  000005CB'EF    01 A3     D0    03C7          MOVL    1(R3),INCMASK+4         ;GET THE INCLUDE MASK
                                 03CF          $QIOW_S CHAN = OUTBANDINC -
                                 03CF                  FUNC = #IO$_SETMODE!IO$M_OUTBAND!IO$M_INCLUDE -
```

```
                        03CF   486                     P1 = OUTBANDAST -
                        03CF   487                     P2 = #INCMASK
          04   05 A3    91   03F8   488 53$:     CMPB    5(R3),#4                ;AGAIN, A SINGLE LONGWORD
                 03     13   03FC   489             BEQL    55$
               0221     31   03FE   490             BRW     OUTBAND_ERR
000005D3'EF    06 A3    D1   0401   491 55$:     CMPL    6(R3),EXCMASK+4
                 31     13   0409   492             BEQL    58$                     ;DON'T BOTHER WITH THE QIO
000005D3'EF    06 A3    D0   040B   493             MOVL    6(R3),EXCMASK+4         ;EXCLUDE MASK
                        0413   494             $QIOW_S CHAN = OUTBANDEXC -
                        0413   495                     FUNC = #IO$_SETMODE!IO$M_OUTBAND -
                        0413   496                     P1 = OUTBANDAST -
                        0413   497                     P2 = #EXCMASK
          50    57     D0   043C   498 58$:     MOVL    R7,R0                   ;RESTORE R0
0000095B'EF    00     FB   043F   499             CALLS   #0,QIODONE
               01D8     31   0446   500             BRW     PROCMSG_EXIT
 26 A0    0000'8F     B3   0449   501 60$:     BITW    #IO$M_HANGUP,AST$T_BUF+RDP$W_OPCODE(R0)
                 29     13   044F   502             BEQL    70$
                        0451   503 ; HANGUP IS TREATED A PROGRAM EXIT
                        0451   504             QUIT    #SS$_NORMAL
                        047A   505 70$:
 60    0000095B'EF     9E   047A   506             MOVAB   QIODONE,AST$L_STATE(R0)         ;NEW STATE
          52    26 A0    3C   0481   507             MOVZWL  AST$T_BUF+RDP$W_OPCODE(R0),R2
000005A7'EF    2A A0    D0   0485   508             MOVL    AST$T_BUF+RDP$L_REFID(R0),READQIO
 26 A0    0000'8F     B3   048D   509             BITW    #IO$M_CTRLCAST,AST$T_BUF+RDP$W_OPCODE(R0)
                 03     12   0493   510             BNEQ    72$
               00A2     31   0495   511             BRW     80$
                        0498   512 ; CONTROL-C ENABLE OR DISABLE
               30 A0    D5   0498   513 72$:     TSTL    AST$T_BUF+RDP$L_TT_ASTPRM(R0)
                 6E     13   049B   514             BEQL    75$                     ;DISABLE
          000005A6'EF    95   049D   515             TSTB    CNTRCFLAG               ;IS THERE ALREADY ONE ENABLED?
                 03     13   04A3   516             BEQL    74$
               009A     31   04A5   517             BRW     90$                     ;YES
          000005A6'EF    96   04A8   518 74$:     INCB    CNTRCFLAG
                        04AE   519             $QIO_S  CHAN = CNTRLCHAN -      ;ENABLE
                        04AE   520                     FUNC = #IO$_SETMODE!IO$M_CTRLCAST -
                        04AE   521                     IOSB = AST$Q_IOSB(R0) -
                        04AE   522                     ASTADR = ASTHANDLER -
                        04AE   523                     ASTPRM = R0 -
                        04AE   524                     P1 = CNTRLC_AST -
                        04AE   525                     P2 = #IO$M_CTRLCAST
                        04DC   526             IF_NO_QUOTA     QUIT
               0116     31   0508   527             BRW     PROCMSG_EXIT
                        050B   528 75$:     $QIO_S  CHAN = CNTRLCHAN -      ;DISABLE
                        050B   529                     FUNC = #IO$_SETMODE!IO$M_CTRLCAST -
                        050B   530                     IOSB = AST$Q_IOSB(R0) -
                        050B   531                     ASTADR = ASTHANDLER -
                        050B   532                     ASTPRM = R0
          000005A6'EF    94   0531   533             CLRB    CNTRCFLAG               ;NO ^C'S ENABLED
                 00E7     31   0537   534             BRW     PROCMSG_EXIT
       26 A0    0000'8F     B3   053A   535 80$:     BITW    #IO$M_CTRLYAST,AST$T_BUF+RDP$W_OPCODE(R0)
                 1E     13   0540   536             BEQL    100$
                        0542   537 ; CONTROL-Y ENABLE OR DISABLE
04 A0    00000000 00000000'8F    7D   0542   538 90$:     MOVQ    #SS$_NORMAL,AST$Q_IOSB(R0)              ;PRETEND WE DID THE QIO
                        054E   539             $DCLAST_S ASTADR = ASTHANDLER -
                        054E   540                       ASTPRM = R0
               00C1     31   055D   541             BRW     PROCMSG_EXIT
                        0560   542 100$:
```

```
                 53    50    D0  0560    543          MOVL    R0,R3                       ;WE NEED THE BUFFER ADDRESS LATER
                 54    08    D0  0563    544          MOVL    #8,R4                       ;ASSUME SHORT FORM
00000000'EF    30    A0    7D  0566    545          MOVQ    AST$T_BUF+RDP$Q_TT_CHAR(R0),CHARBUF
          00000000'EF    95    056E    546          TSTB    PROTO_ECO                   ;IS IT LEVEL 0
                   0B    13    0574    547          BEQL    105$
                 54    04    C0  0576    548          ADDL    #4,R4                       ;LONG FORM
00000008'EF    44    A0    D0  0579    549          MOVL    AST$T_BUF+RDP$L_TT_CHAR2(R0),CHARBUF+8
                             0581    550 105$:    $QIO_S  CHAN = READCHAN -
                             0581    551                  FUNC = R2 -
                             0581    552                  IOSB = AST$Q_IOSB(R0) -
                             0581    553                  ASTADR = ASTHANDLER -
                             0581    554                  ASTPRM = R0 -
                             0581    555                  P1 = CHARBUF -
                             0581    556                  P2 = R4 -
                             0581    557                  P3 = AST$T_BUF+RDP$L_TT_SPEED(R0) -
                             0581    558                  P4 = AST$T_BUF+RDP$L_TT_FILL(R0) -
                             0581    559                  P5 = AST$T_BUF+RDP$L_TT_PARITY(R0)
                             05B0    560 ; TAKE CARE OF NOT ALWAYS GETTING AST ON ERROR
                6E    50    E8  05B0    561          BLBS    R0,PROCMSG_EXIT             ;NO ERROR
                             05B3    562
                             05B3    563 QIO_ERR:
              04    A3    50    B0  05B3    564          MOVW    R0,AST$Q_IOSB(R3)           ;MAKE SURE STATUS IS IN IOSB
                             05B7    565          $DCLAST_S ASTADR = ASTHANDLER -
                             05B7    566                    ASTPRM = R3
                   59    11  05C6    567          BRB     PROCMSG_EXIT
                             05C8    568 ;
                             05C8    569 ; A SENSE MODE QIO REQUEST
                             05C8    570 SENSEMSG:
                 53    50    D0  05C8    571          MOVL    R0,R3                       ;WE NEED THE BUFFER ADDRESS LATER
        60    0000095B'EF    9E  05CB    572          MOVAB   QIODONE,AST$L_STATE(R0)     ;NEW STATE
               52    26    A0  3C  05D2    573          MOVZWL  AST$T_BUF+RDP$W_OPCODE(R0),R2
000005A7'EF    2A    A0    D0  05D6    574          MOVL    AST$T_BUF+RDP$L_REFID(R0),READQIO
                 38    A0    7C  05DE    575          CLRQ    AST$T_BUF+RDP$Q_TT_SCHAR(R0)
                 40    A0    D4  05E1    576          CLRL    AST$T_BUF+RDP$Q_TT_SCHAR+8(R0)
                             05E4    577          $QIO_S  CHAN = READCHAN -
                             05E4    578                  FUNC = R2 -
                             05E4    579                  IOSB = AST$Q_IOSB(R0) -
                             05E4    580                  ASTADR = ASTHANDLER -
                             05E4    581                  ASTPRM = R0 -
                             05E4    582                  P1 = AST$T_BUF+RDP$Q_TT_SCHAR(R0) -
                             05E4    583                  P2 = #12
                             0609    584 ; TAKE CARE OF NOT ALWAYS GETTING AST ON ERROR
                15    50    E8  0609    585          BLBS    R0,PROCMSG_EXIT             ;NO ERROR
              04    A3    50    B0  060C    586          MOVW    R0,AST$Q_IOSB(R3)           ;MAKE SURE STATUS IS IN IOSB
                             0610    587          $DCLAST_S ASTADR = ASTHANDLER -
                             0610    588                    ASTPRM = R3
                   00    11  061F    589          BRB     PROCMSG_EXIT
                             0621    590
                             0621    591 PROCMSG_EXIT:
                   04        0621    592          RET
                             0622    593
                             0622    594 OUTBAND_ERR:
                             0622    595          $PUTMSG_S MSGVEC = BADOUTBAND
                             0635    596          QUIT
                             065A    597
```

```
                              065A         .SBTTL LINKRECV - PROCESS A RECEIVED MESSAGE
                              065A   ;++
                              065A   ; FUNCTIONAL DESCRIPTION:
                              065A   ;
                              065A   ;       PROCESS THE AST INDICATING THAT A MESSAGE WAS RECEIVED ON THE LINK
                              065A   ;
                              065A   ; CALLING SEQUENCE:
                              065A   ;
                              065A   ;       CALLS   #0,LINKRECV
                              065A   ;
                              065A   ; INPUT PARAMETERS:
                              065A   ;
                              065A   ;       R0 POINTS TO AN AST CONTROL BLOCK
                              065A   ;
                              065A   ; IMPLICIT INPUTS:
                              065A   ;
                              065A   ;       WRITEQIO
                              065A   ;       READQIO
                              065A   ;
                              065A   ; OUTPUT PARAMETERS:
                              065A   ;
                              065A   ;       NONE
                              065A   ;
                              065A   ; IMPLICIT OUTPUTS:
                              065A   ;
                              065A   ;       AN ENTRY MAY BE ADDED TO THE QUEUE OF PENDING READS OR WRITES.
                              065A   ;       RETSTATUS
                              065A   ;
                              065A   ; COMPLETION CODES:
                              065A   ;
                              065A   ;
                              065A   ; SIDE EFFECTS:
                              065A   ;
                              065A   ;       A NEW READ OF THE LINK IS INITIATED.  IF THERE IS A ERROR ON THIS QIO,
                              065A   ;       A $WAKE IS ISSUED TO CAUSE A PROGRAM EXIT.
                              065A   ;
                              065A   ;       A PREVIOUS I/O MAY BE CANCELED
                              065A   ;
                              065A   ;--
                              065A   ;
                              065A   .IF DF ASSEM_TRACE
                              065A   TRACE_RECV:
                    3F   BB   065A          PUSHR   #^M<R0,R1,R2,R3,R4,R5>
          51    06 A0    3C   065C          MOVZWL  AST$Q_IOSB+2(R0),R1
          52    26 A0    9E   0660          MOVAB   AST$T_BUF(R0),R2
   53     00000000'EF    9E   0664          MOVAB   DBG$LINKRECV,R3             ; Message received from net
                 F992'   30   066B          BSBW    DBG$TRACE_IO                ; ... log input
                    3F   BA   066E          POPR    #^M<R0,R1,R2,R3,R4,R5>
                    77   11   0670          BRB     TRACE_CONTINUE
                              0672   .ENDC
                              0672
                              0672   LINKRECV_ERR:
          52    04 A0    3C   0672          MOVZWL  AST$Q_IOSB(R0),R2                   ;SAVE ERROR STATUS
       0000'8F     52    B1   0676          CMPW    R2,#SS$_ABORT
                    1F   12   067B          BNEQ    20$
   00000597'EF     52    D0   067D          MOVL    R2,LINKERR                  ;SAVE ERROR
                              0684          $SETIMR_S DAYTIM = THREESEC -       ;JUST IN CASE MAILBOX DOESN'T GET REASON
```

```
                        0684   656                          ASTADR = LINKGONE
                  04    069B   657 10$:     RET                                ;LINK BROKE - LINK MBX WILL GET REASON
          0000'CF 95    069C   658 20$:     TSTB    W^WAKEFLAG                 ; has QUIT already happened
                  F9    12     06A0   659     BNEQ    10$                        ; if so, ignore error
                        06A2   660     $PUTMSG_S MSGVEC = DECNETERR       ;LINK ERROR
                        06B5   661     QUIT    R2
                        06DA   662
                        06DA   663 LINKRECV::
                  008C  06DA   664     .WORD   ^M<R2,R3,R7>
              92 04 A0  E9    06DC   665     BLBC    AST$Q_IOSB(R0),LINKRECV_ERR     ;ERROR ON LINK READ ??
                        06E0   666
                        06E0   667 .IF DF ASSEM_TRACE
                  01    E1     06E0   668     BBC     #RTLOG$V_TRACE,-
                0000'CF         06E2   669             W^RTLOG_FLAGS,-
                  03            06E5   670             TRACE_CONTINUE             ; branch if not tracing
                FF71    31      06E6   671     BRW     TRACE_RECV                 ; branch if tracing
                        06E9   672 TRACE_CONTINUE:
                        06E9   673 .ENDC
          06 00000000'EF  E9    06E9   674     BLBC    CTERM_FLAG,20$
                    F90D' 30    06F0   675     BSBW    CTERM_LINKRECV
                    0072  31     06F3   676     BRW     100$
                        06F6   677 20$:
              51    26 A0   3C    06F6   678     MOVZWL  AST$T_BUF+RDP$W_OPCODE(R0),R1
                    57      D4    06FA   679     CLRL    R7
          52    00001067'EF   9E    06FC   680     MOVAB   TERMOPS,R2
                53    82      3C    0703   681     MOVZWL  (R2)+,R3                   ;COUNT
                62    51      B1    0706   682 30$:     CMPW    R1,(R2)
                      06      12    0709   683     BNEQ    40$
              57    02 A2   3C    070B   684     MOVZWL  2(R2),R7                   ;INTERNAL OP CODE
                      06      11    070F   685     BRB     50$
                52    04      C0    0711   686 40$:     ADDL2   #4,R2                      ;SKIP
                  EF  53      F5    0714   687     SOBGTR  R3,30$
                        0717   688 50$:
              0C A0     57    B0    0717   689     MOVW    R7,AST$W_OPCODE(R0)        ; Save for later
                 06     57    91    071B   690     CMPB    R7,#OP_BRDCST
                        08    12    071E   691     BNEQ    60$
          000008BC'EF   16    0720   692     JSB     BROADCAST                  ;GO BROADCAST IT
                        40    11    0726   693     BRB     100$
                05      57    91    0728   694 60$:     CMPB    R7,#OP_CANCEL
                        08    12    072B   695     BNEQ    70$
          000007BD'EF   16    072D   696     JSB     CANCELIO                   ;GO CANCEL IT
                        33    11    0733   697     BRB     100$
                02      57    91    0735   698 70$:     CMPB    R7,#OP_WRITE
                        18    12    0738   699     BNEQ    90$
                        073A   700 ; A WRITE QIO
          000005AB'EF   D5    073A   701     TSTL    WRITEQIO
                        07    12    0740   702     BNEQ    80$
                        0742   703 ; NO WRITE IN PROGRESS
            FAE8 CF     00    FB    0742   704     CALLS   #0,PROCMSG
                        1F    11    0747   705     BRB     100$
                        0749   706 80$:
      000005BB'FF       60    0E    0749   707     INSQUE  (R0),@WRITEQ+4     ;QUEUE IT
                        16    11    0750   708     BRB     100$
                        0752   709 90$:
                        0752   710 ; A READ OR SETMODE OR SENSEMODE
          000005A7'EF   D5    0752   711     TSTL    READQIO
                        07    12    0758   712     BNEQ    95$
```

```
                        075A   713 ; NO READ IN PROGRESS
      FAD0 CF   00   FB 075A   714         CALLS   #0,PROCMSG
                07   11 075F   715         BRB     100$
                         0761   716 95$:
 000005B3'FF    60   0E 0761   717         INSQUE  (R0),@READQ+4   ;QUEUE IT
                         0768   718 100$:
           08A6   30   0768   719         BSBW    GETBUF          ;GET A BUFFER FOR READING LINK
   60   FF6B CF   9E   076B   720         MOVAB   LINKRECV,(R0)
                         0770   721         $QIO_S  CHAN = LINKCHAN -       ;READ LINK AGAIN
                         0770   722                 FUNC = #IO$_READVBLK -
                         0770   723                 IOSB = AST$Q_IOSB(R0) -
                         0770   724                 ASTADR = ASTHANDLER -
                         0770   725                 ASTPRM = R0 -
                         0770   726                 P1 = AST$T_BUF(R0) -
                         0770   727                 P2 = #MAXMSG
         1E 50   E8 079B   728         BLBS    R0,110$                 ;WAS THERE A LINK ERROR
 00000597'EF    50   D0 079E   729         MOVL    R0,LINKERR              ;SAVE ERROR
                         07A5   730         $SETIMR_S DAYTIM = THREESEC -   ;JUST IN CASE MAILBOX DOESN'T GET REASON
                         07A5   731                   ASTADR = LINKGONE
                   04   07BC   732 110$:   RET
                         07BD   733 ;
                         07BD   734 ;
                         07BD   735 ;
                         07BD   736 ; CANCEL AN I/O
                         07BD   737 ;
                         07BD   738 CANCELIO:
           52   50   D0 07BD   739         MOVL    R0,R2
                         07C0   740         $CANCEL_S CHAN = CNTRLCHAN       ;DISABLE ^C ON A CANCEL
      000005A6'EF   94   07CE   741         CLRB    CNTRCFLAG               ;NO CONTROL-C'S ENABLED
        50   52   D0 07D4   742         MOVL    R2,R0
 000005A7'EF    2A A0   D1 07D7   743         CMPL    AST$T_BUF+RDP$L_REFID(R0),READQIO
                15   12 07DF   744         BNEQ    20$
                         07E1   745 ; CANCEL THE READ
         0000105F'EF    16   07E1   746         JSB     BUFFREE
                         07E7   747         $CANCEL_S  CHAN = READCHAN
                   05   07F5   748         RSB
 000005AB'EF    2A A0   D1 07F6   749 20$:    CMPL    AST$T_BUF+RDP$L_REFID(R0),WRITEQIO
                15   12 07FE   750         BNEQ    30$
                         0800   751 ; CANCEL THE WRITE
         0000105F'EF    16   0800   752         JSB     BUFFREE
                         0806   753         $CANCEL_S  CHAN = WRITECHAN
                   05   0814   754         RSB
                         0815   755 30$:
   51    000005AF'EF   9E   0815   756         MOVAB   READQ,R1        ;GET QUEUE OF PENDING READS
           51   61   9E 081C   757 31$:    MOVAB   (R1),R1
   000005AF'8F   51   D1 081F   758         CMPL    R1,#READQ
                09   13 0826   759         BEQL    32$             ;END OF QUEUE
             2A A1   D1 0828   760         CMPL    AST$T_BUF+RDP$L_REFID(R1),-
             2A A0      082B   761                 AST$T_BUF+RDP$L_REFID(R0)
                ED   12 082D   762         BNEQ    31$             ;TRY THE NEXT ONE
                23   11 082F   763         BRB     39$             ;GO CANCEL IT
   51    000005B7'EF   9E   0831   764 32$:    MOVAB   WRITEQ,R1       ;GET QUEUE OF PENDING WRITES
           51   61   D0 0838   765 33$:    MOVL    (R1),R1
   000005B7'8F   51   D1 083B   766         CMPL    R1,#WRITEQ
                09   13 0842   767         BEQL    34$             ;END OF QUEUE
             2A A1   D1 0844   768         CMPL    AST$T_BUF+RDP$L_REFID(R1),-
             2A A0      0847   769                 AST$T_BUF+RDP$L_REFID(R0)
```

```
                     ED   12  0849   770          BNEQ    33$                ;TRY THE NEXT ONE
                     07   11  084B   771          BRB     39$                ;SAVE THE CANCEL - WRITE NOT DONE YET
          0000105F'EF     16  084D   772 34$:     JSB     BUFFREE            ;QIO ALREADY DONE - THROW OUT THE CANCEL
                          05  0853   773          RSB
                              0854   774 39$:
               51   61     0F  0854   775          REMQUE  (R1),R1 ;REMOVE THE ENTRY
    00000571'EF   2A A1     D0  0857   776          MOVL    AST$T_BUF+RDP$L_REFID(R1),CANMSG+RDP$L_REFID
          0000105F'EF     16  085F   777          JSB     BUFFREE ;WE DON'T NEED THE CANCEL ANYMORE
               50   51     D0  0865   778          MOVL    R1,R0
          0000105F'EF     16  0868   779          JSB     BUFFREE            ;WE DON'T NEED THE QIO EITHER
                              086E   780          $QIO_S  CHAN = LINKCHAN -       ;SEND THE CANCEL COMPLETE MSG
                              086E   781                  FUNC = #IO$_WRITEVBLK -
                              086E   782                  P1 = CANMSG -
                              086E   783                  P2 = #RDP$K_HEADERLEN+8
                              0893   784          ONERROR QUIT
                          05  08BB   785          RSB
                              08BC   786 ;
                              08BC   787 ;
                              08BC   788 ;
                              08BC   789 ; BROADCAST TO THE TERMINAL
                              08BC   790 ;
                              08BC   791 BROADCAST:
    0000057F'EF   30 A0   D0  08BC   792          MOVL    AST$T_BUF+RDP$L_TT_BCNT(R0),BRDDESC     ;COUNT
    00000583'EF   38 A0   9E  08C4   793          MOVAB   AST$T_BUF+RDP$T_TT_WDATA(R0),BRDDESC+4  ;ADDRESS
0A0D 8F   00000583'FF     B1  08CC   794          CMPW    @BRDDESC+4,#^XA0D                       ;CHECK FOR CR-LF
                     06   12  08D5   795          BNEQ    10$
          00000583'FF     B4  08D7   796          CLRW    @BRDDESC+4                              ;REMOVE IT - BRDCST ADDS ANOTHER ONE
               52   50     D0  08DD   797 10$:     MOVL    R0,R2
                              08E0   798          $BRDCST_S MSGBUF = BRDDESC -
                              08E0   799                    DEVNAM = TTYDESC
         30 A2    50     7D  08F7   800          MOVQ    R0,AST$T_BUF+RDP$Q_STATUS(R2)   ;RETURN AN IOSB
   62    00000A9F'EF     9E  08FB   801          MOVAB   LNKWRTDONE,AST$L_STATE(R2)              ;NEW STATE
    26 A2    FFFE 8F     B0  0902   802          MOVW    #RDP$C_END,AST$T_BUF+RDP$W_OPCODE(R2)   ;A STATUS MESSAGE
              28 A2     B4  0908   803          CLRW    AST$T_BUF+RDP$W_MOD(R2)
                              090B   804          $QIO_S  CHAN = LINKCHAN -       ;WRITE MESSAGE ON LINK
                              090B   805                  FUNC = #IO$_WRITEVBLK -
                              090B   806                  IOSB = AST$Q_IOSB(R2) -
                              090B   807                  ASTADR = ASTHANDLER -
                              090B   808                  ASTPRM = R2 -
                              090B   809                  P1 = AST$T_BUF(R2) -
                              090B   810                  P2 = #RDP$R_HEADERLEN+8
                              0932   811          ONERROR QUIT
                          05  095A   812          RSB
```

```
                                        095B   814          .SBTTL QIODONE - PROCESS A COMPLETED TERMINAL QIO
                                        095B   815 ;++
                                        095B   816 ; FUNCTIONAL DESCRIPTION:
                                        095B   817 ;
                                        095B   818 ;       HANDLE THE AST INDICATING THAT A TERMINAL QIO HAS COMPLETED
                                        095B   819 ;
                                        095B   820 ; CALLING SEQUENCE:
                                        095B   821 ;
                                        095B   822 ;       CALLS   #0,QIODONE
                                        095B   823 ;
                                        095B   824 ; INPUT PARAMETERS:
                                        095B   825 ;
                                        095B   826 ;       R0 POINTS TO AN AST CONTROL BLOCK
                                        095B   827 ;
                                        095B   828 ; IMPLICIT INPUTS:
                                        095B   829 ;
                                        095B   830 ;       NONE
                                        095B   831 ;
                                        095B   832 ; OUTPUT PARAMETERS:
                                        095B   833 ;
                                        095B   834 ;       NONE
                                        095B   835 ;
                                        095B   836 ; IMPLICIT OUTPUTS:
                                        095B   837 ;
                                        095B   838 ;       RETSTATUS
                                        095B   839 ;
                                        095B   840 ; COMPLETION CODES:
                                        095B   841 ;
                                        095B   842 ;
                                        095B   843 ; SIDE EFFECTS:
                                        095B   844 ;
                                        095B   845 ;       A STATUS MESSAGE IS WRITTEN TO THE LINK.  IF THERE IS AN ERROR ON THIS
                                        095B   846 ;       QIO, A $WAKE IS ISSUED TO CAUSE THE PROGRAM TO EXIT
                                        095B   847 ;
                                        095B   848 ;--
                                        095B   849 ;
                                        095B   850 QIODONE::
                                 00BC   095B   851          .WORD   ^M<R2,R3,R4,R5,R7>
                                        095D   852
                  0A 00000000'EF   E9   095D   853          BLBC    CTERM_FLAG,10$
                             F699'   30   0964   854          BSBW    CTERM_QIODONE
                                50   D5   0967   855          TSTL    R0                          ; Was a message returned?
                                56   12   0969   856          BNEQ    60$                         ; Branch if yes
                              0055   31   096B   857          BRW     70$                         ; Exit if it not
                                        096E   858 10$:
                             2A A0   D5   096E   859          TSTL    AST$T_BUF+RDP$L_REFID(R0)                ;CHECK FOR ZERO REF ID
                                0C   12   0971   860          BNEQ    20$
                          04 A0      01   9A   0973   861          MOVZBL  #1,AST$Q_IOSB(R0)        ;NO ERRORS
                00000A9F'EF          00   FB   0977   862          CALLS   #0,LNKWRTDONE    ;PRETEND WE SENT IT
                                        04   097E   863          RET
                                        097F   864 20$:
                      57   0C A0      3C   097F   865          MOVZWL  AST$W_OPCODE(R0),R7      ; Fetch internal opcode
                                01   57   91   0983   866          CMPB    R7,#OP_READ
                                11   12   0986   867          BNEQ    30$
                                        0988   868 ; IT WAS A READ SO WE NEED THE COUNT
                      51   06 A0      3C   0988   869          MOVZWL  AST$Q_IOSB+2(R0),R1      ;CHARACTERS BEFORE THE TERMINATOR
                      51   0A A0      A0   098C   870          ADDW2   AST$Q_IOSB+6(R0),R1      ;TOTAL CHARACTERS
```

```
                38 A0   51   B0  0990   871          MOVW    R1,AST$T_BUF+RDP$T_TT_RDATA(R0) ;SAVE IN LINK MESSAGE
                   51   02   A0  0994   872          ADDW2   #2,R1                    ;SIZE OF READ DATA PLUS COUNT
                        13   11  0997   873          BRB     50$
                             09  0999   874 30$:
                   02   57   91  0999   875          CMPB    R7,#OP_WRITE
                        04   12  099C   876          BNEQ    40$
                        51   D4  099E   877          CLRL    R1                       ;NO READ DATA
                        0A   11  09A0   878          BRB     50$
                             09A2   879 40$:
                             09A2   880 ; SETMODE OR SENSEMODE
                        51   D4  09A2   881          CLRL    R1                       ;ASSUME NO DATA
                   03   57   91  09A4   882          CMPB    R7,#OP_SETMODE
                        03   13  09A7   883          BEQL    50$
                   51   0C   D0  09A9   884          MOVL    #12,R1                   ;12 BYTES OF DATA
            26 A0  FFFE 8F   B0  09AC   885 50$:     MOVW    #RDP$C_END,AST$T_BUF+RDP$W_OPCODE(R0)    ;A STATUS MESSAGE
                28 A0   B4  09B2   886          CLRW    AST$T_BUF+RDP$W_MOD(R0)
              30 A0   04 A0   7D  09B5   887          MOVQ    AST$Q_IOSB(R0),AST$T_BUF+RDP$Q_STATUS(R0)
                   51   12 A0   9A  09BA   888          ADDW2   #RDP$T_TT_RDATA,R1
                52   26 A0   9E  09BD   889          MOVAB   AST$T_BUF(R0),R2                 ; SET ADDRESS OF WRITE MESSAGE
                             09C1   890 60$:
                        6D   10  09C1   891          BSBB    WRITE_TO_NETX
                             09C3   892 70$:
                        04   09C3   893          RET
                             09C4   894
```

```
                                09C4   896          .SBTTL WRITE_TO_NET - WRITE TO LINK
                                09C4   897 ;
                                09C4   898 ;
                                09C4   899 ; INPUTS:
                                09C4   900 ;       R0 - AST BLOCK
                                09C4   901 ;       R1 - length of message
                                09C4   902 ;       R2 - address of message
                                09C4   903 ;       R3 - AST routine to call (if called at WRITE_TO_NET)
                                09C4   904 ;
                                09C4   905
                                09C4   906 WRITE_TO_NET_SYNC::     ; R0,R1,R2 inputs
                                09C4   907
                          53    DD 09C4   908          PUSHL   R3                          ; Save R3
                          53    D4 09C6   909          CLRL    R3                          ; No AST
                          50    DD 09C8   910          PUSHL   R0                          ; Save block
                          70    10 09CA   911          BSBB    WRITE_TO_NET                ; Write to net
                                   09CC   912          $WAITFR_S EFN = #RT$C_LINKEFN       ; Wait for completion
                                   09D5   913          ONERROR QUIT                        ; Failure?
                          50 8ED0 09FD   914          POPL    R0                          ; Restore AST block
                    50    04 A0   3C   0A00   915          MOVZWL  AST$Q_IOSB(R0),R0           ; Fetch status
                                   0A04   916          ONERROR QUIT                        ; Failure?
                          53 8ED0 0A2C   917          POPL    R3                          ; Restore
                             05    0A2F   918          RSB                                 ; Return
                                   0A30   919
                                   0A30   920 WRITE_TO_NETX::                             ; Standard completion routine
                60  00000A9F'EF    9E   0A30   921          MOVAB   LNKWRTDONE,AST$L_STATE(R0)      ; New state
                    53    F7E9 CF    9E   0A37   922          MOVAB   ASTHANDLER,R3               ; AST routine
                                   0A3C   923
                                   0A3C   924 WRITE_TO_NET::
                                   0A3C   925
                                   0A3C   926 .IF DF ASSEM_TRACE
                             01    E0   0A3C   927          BBS     #RTLOG$V_TRACE,-
                          4D 0000'CF        0A3E   928                  W^RTLOG_FLAGS,TRACE_WRITE      ; branch if tracing
                                   0A42   929 TRACE_CONTINUE2:
                                   0A42   930 .ENDC
                                   0A42   931          $QIO_S  CHAN = LINKCHAN -       ;WRITE MESSAGE ON LINK
                                   0A42   932                  EFN  = #RT$C_LINKEFN,-
                                   0A42   933                  FUNC = #IO$_WRITEVBLK -
                                   0A42   934                  IOSB = AST$Q_IOSB(R0) -
                                   0A42   935                  ASTADR = (R3) -
                                   0A42   936                  ASTPRM = R0 -
                                   0A42   937                  P1 = (R2) -
                                   0A42   938                  P2 = R1
                                   0A66   939          ONERROR QUIT
                             05    0A8E   940          RSB
                                   0A8F   941
                                   0A8F   942 .IF DF ASSEM_TRACE
                                   0A8F   943 TRACE_WRITE:
                          3F    BB 0A8F   944          PUSHR   #^M<R0,R1,R2,R3,R4,R5>  ; Trace code
                53  00000000'EF    9E   0A91   945          MOVAB   DBG$LINKWRITE,R3
                          F565'    30   0A98   946          BSBW    DBG$TRACE_IO
                          3F    BA 0A9B   947          POPR    #^M<R0,R1,R2,R3,R4,R5>
                          A3    11 0A9D   948          BRB     TRACE_CONTINUE2
                                   0A9F   949 .ENDC
```

```
                                        .SBTTL LNKWRTDONE - A WRITE TO THE LINK HAS COMPLETED
                                ;++
                                ; FUNCTIONAL DESCRIPTION:
                                ;
                                ;       HANDLE THE AST INDICATING THAT A WRITE HAS COMPLETED ON THE LINK
                                ;       BY FREEING THE BUFFER.
                                ;
                                ; CALLING SEQUENCE:
                                ;
                                ;       CALLS   #0,LNKWRTDONE
                                ;
                                ; INPUT PARAMETERS:
                                ;
                                ;       R0 POINTS TO AN AST CONTROL BLOCK
                                ;
                                ; IMPLICIT INPUTS:
                                ;
                                ;       NONE
                                ;
                                ; OUTPUT PARAMETERS:
                                ;
                                ;       NONE
                                ;
                                ; IMPLICIT OUTPUTS:
                                ;
                                ;       NONE
                                ;
                                ; COMPLETION CODES:
                                ;
                                ;
                                ; SIDE EFFECTS:
                                ;
                                ;       A BUFFER IS FREED
                                ;       THE QUEUE OF PENDING QIO'S IS CHECKED
                                ;
                                ;--
                                ;
                                LNKWRTDONE::
                        0000    .WORD   0
      51 04 A0    E8            BLBS    AST$Q_IOSB(R0),20$              ;ERROR ON LINK WRITE ??
   52    04 A0    3C            MOVZWL  AST$Q_IOSB(R0),R2               ;SAVE ERROR STATUS
0000'8F    52    B1            CMPW    R2,#S$$_ABORT
             01    12            BNEQ    10$
                   04            RET                                     ;LINK BROKE - LINK MBX WILL GET REASON
                                 $PUTMSG_S MSGVEC = DECNETERR            ;LINK ERROR
                                 QUIT    R2
06 00000000'EF    E9            BLBC    CTERM_FLAG,20$
          F50D'   30            BSBW    CTERM_LNKWRTDONE
             0042 31            BRW     40$
                                ;
   53    2A A0    D0            MOVL    AST$T_BUF+RDP$L_REFID(R0),R3    ;SAVE ID
            0562  30            BSBW    BUFFREE                 ;RELEASE THE BUFFER
000005A7'EF   53  D1            CMPL    R3,READQIO              ;WAS THIS A READ (OR SET MODE)
             15   12            BNEQ    30$                     ;NO
   000005A7'EF   D4            CLRL    READQIO                 ;FORGET ABOUT THE PREVIOUS ONE
50 000005AF'FF   0F            REMQUE  @READQ,R0
```

```
                 06   1D  0B13  1008         BVS     30$              ;NO MORE
        F715 CF  00   FB  0B15  1009         CALLS   #0,PROCMSG       ;GO PROCESS IT
                      04  0B1A  1010         RET
   000005AB'EF   53   D1  0B1B  1011 30$:    CMPL    R3,WRITEQIO      ;WAS THIS A WRITE
                 14   12  0B22  1012         BNEQ    40$              ;NO
     000005AB'EF      D4  0B24  1013         CLRL    WRITEQIO         ;FORGET ABOUT THE PREVIOUS ONE
   50  000005B7'FF    0F  0B2A  1014         REMQUE  @WRITEQ,R0
                 05   1D  0B31  1015         BVS     40$              ;NO MORE
        F6F7 CF  00   FB  0B33  1016         CALLS   #0,PROCMSG       ;GO PROCESS IT
                      04  0B38  1017 40$:    RET
```

VMS
V04

```
                     0B39  1019          .SBTTL LNKMBXDONE - MESSAGE RECEIVED ON THE LINK MAILBOX
                     0B39  1020 ;++
                     0B39  1021 ; FUNCTIONAL DESCRIPTION:
                     0B39  1022 ;
                     0B39  1023 ;       HANDLE THE AST INDICATING THAT A MESSAGE WAS RECEIVED ON THE LINK
                     0B39  1024 ;       MAILBOX
                     0B39  1025 ;
                     0B39  1026 ; CALLING SEQUENCE:
                     0B39  1027 ;
                     0B39  1028 ;       CALLS   #0,LNKMBXDONE
                     0B39  1029 ;
                     0B39  1030 ; INPUT PARAMETERS:
                     0B39  1031 ;
                     0B39  1032 ;       R0 POINTS TO AN AST CONTROL BLOCK
                     0B39  1033 ;
                     0B39  1034 ; IMPLICIT INPUTS:
                     0B39  1035 ;
                     0B39  1036 ;       NONE
                     0B39  1037 ;
                     0B39  1038 ; OUTPUT PARAMETERS:
                     0B39  1039 ;
                     0B39  1040 ;       NONE
                     0B39  1041 ;
                     0B39  1042 ; IMPLICIT OUTPUTS:
                     0B39  1043 ;
                     0B39  1044 ;       NONE
                     0B39  1045 ;
                     0B39  1046 ; COMPLETION CODES:
                     0B39  1047 ;
                     0B39  1048 ;
                     0B39  1049 ; SIDE EFFECTS:
                     0B39  1050 ;
                     0B39  1051 ;       THE PROGRAM CAN BE ABORTED.
                     0B39  1052 ;
                     0B39  1053 ;--
                     0B39  1054 ;
                     0B39  1055 LNKMBXDONE:
             0004    0B39  1056          .WORD   ^M<R2>
                     0B3B  1057          ;
                     0B3B  1058          ; First, check valid disconnects
                     0B3B  1059          ;
  52    26 A0   3C   0B3B  1060          MOVZWL  AST$T_BUF(R0),R2        ; Fetch MSG code
0000'8F    52   B1   0B3F  1061          CMPW    R2,#MSG$_DISCON         ; Disconnect?
           0E   13   0B44  1062          BEQL    10$                     ; Branch if yes
0000'8F    52   B1   0B46  1063          CMPW    R2,#MSG$_EXIT           ; Exit?
           07   13   0B4B  1064          BEQL    10$                     ; Branch if yes
0000'8F    52   B1   0B4D  1065          CMPW    R2,#MSG$_ABORT          ; Log out - ignore it
           29   12   0B52  1066          BNEQ    20$                     ; Not a valid shutdown message...
                     0B54  1067 10$:
                     0B54  1068          QUIT    #SS$_NORMAL             ; Exit, no status message
                     0B7D  1069          ;
                     0B7D  1070          ; Either a serious error or something like a CONFIRM,
                     0B7D  1071          ; which isn't important.
                     0B7D  1072          ;
                     0B7D  1073 20$:
0000'8F    52   B1   0B7D  1074          CMPW    R2,#MSG$_THIRDPARTY     ; Third party disconnect?
           29   12   0B82  1075          BNEQ    30$                     ; Branch if not
```

```
                     0B84  1076                QUIT    #SS$_THIRDPARTY          ; Abort program
                     0BAD  1077 30$:
0000'8F    52   B1  0BAD  1078                CMPW    R2,#MSG$_PATHLOST        ; Path lost?
           29   12  0BB2  1079                BNEQ    40$                      ; Branch if not
                     0BB4  1080                QUIT    #SS$_PATHLOST            ; Abort program
                     0BBD  1081                ;
                     0BBD  1082                ; Unimportant message, just requeue read
                     0BBD  1083                ;
                     0BBD  1084 40$:
                     0BBD  1085                $QIO_S  CHAN = MAILCHAN -        ;LINK MAILBOX READ
                     0BBD  1086                        FUNC = #IO$_READVBLK -
                     0BBD  1087                        IOSB = LINKMAIL+AST$Q_IOSB -
                     0BBD  1088                        ASTADR = ASTHANDLER -
                     0BBD  1089                        ASTPRM = #LINKMAIL -
                     0BBD  1090                        P1 = LINKMAIL+AST$T_BUF -
                     0BBD  1091                        P2 = #40
                     0C0E  1092                ONERROR QUIT
                  04 0C36  1093                RET
```

```
                        0C37  1095          .SBTTL OUTBANDAST - OUT OF BAND CHARACTER AST RECEIVED
                        0C37  1096 ;++
                        0C37  1097 ; FUNCTIONAL DESCRIPTION:
                        0C37  1098 ;
                        0C37  1099 ;       HANDLES THE AST RESULTING FROM AN OUT OF BAND CHARACTER
                        0C37  1100 ;
                        0C37  1101 ; CALLING SEQUENCE:
                        0C37  1102 ;
                        0C37  1103 ;       CALLS   #0,OUTBANDAST
                        0C37  1104 ;
                        0C37  1105 ; INPUT PARAMETERS:
                        0C37  1106 ;
                        0C37  1107 ;       NONE
                        0C37  1108 ;
                        0C37  1109 ; IMPLICIT INPUTS:
                        0C37  1110 ;
                        0C37  1111 ;       AST PARAMETER - CHARACTER
                        0C37  1112 ;
                        0C37  1113 ; OUTPUT PARAMETERS:
                        0C37  1114 ;
                        0C37  1115 ;       NONE
                        0C37  1116 ;
                        0C37  1117 ; IMPLICIT OUTPUTS:
                        0C37  1118 ;
                        0C37  1119 ;       RETSTATUS
                        0C37  1120 ;
                        0C37  1121 ; COMPLETION CODES:
                        0C37  1122 ;
                        0C37  1123 ;
                        0C37  1124 ; SIDE EFFECTS:
                        0C37  1125 ;
                        0C37  1126 ;       A MESSAGE SENT ON LINK
                        0C37  1127 ;
                        0C37  1128 ;--
                        0C37  1129 ;
                        0C37  1130 OUTBANDAST:
                0000    0C37  1131          .WORD   0
    0000056C'EF   04 AC   90  0C39  1132          MOVB    4(AP),OUTBANDCHAR       ;GET THE CHARACTER THAT CAUSED THE AST
00000562'EF   FFFF 8F   B0  0C41  1133          MOVW    #RDP$C_ATTN,OUTBANDMSG+RDP$W_OPCODE
    00000564'EF   06    B0  0C4A  1134          MOVW    #RDP$C_TT_OUTBAND,OUTBANDMSG+RDP$W_MOD
0000056A'EF   00000000'EF   B0  0C51  1135          MOVW    TERMUNIT,OUTBANDMSG+RDP$W_UNIT
                        0C5C  1136          $QIO_S  CHAN = LINKCHAN -       ;SEND ON LINK
                        0C5C  1137                  FUNC = #IO$_WRITEVBLK -
                        0C5C  1138                  P1 = OUTBANDMSG -
                        0C5C  1139                  P2 = #RDP$K_HEADERLEN+1
                        0C81  1140          ONERROR QUIT
                  04    0CA9  1141          RET
```

```
                        OCAA  1143          .SBTTL LINKGONE - TIMER EXPIRED SO LINK IS GONE
                        OCAA  1144 ;++
                        OCAA  1145 ; FUNCTIONAL DESCRIPTION:
                        OCAA  1146 ;
                        OCAA  1147 ;       HANDLES THE AST RESULTING FROM THE TIMER STARTED TO WAIT FOR THE
                        OCAA  1148 ;         MAILBOX TO GET THE REASON WHY THE LINK IS GONE
                        OCAA  1149 ;
                        OCAA  1150 ; CALLING SEQUENCE:
                        OCAA  1151 ;
                        OCAA  1152 ;       CALLS   #0,LINKGONE
                        OCAA  1153 ;
                        OCAA  1154 ; INPUT PARAMETERS:
                        OCAA  1155 ;
                        OCAA  1156 ;       NONE
                        OCAA  1157 ;
                        OCAA  1158 ; IMPLICIT INPUTS:
                        OCAA  1159 ;
                        OCAA  1160 ;       LAST LINK MAILBOX MESSAGE
                        OCAA  1161 ;
                        OCAA  1162 ; OUTPUT PARAMETERS:
                        OCAA  1163 ;
                        OCAA  1164 ;       NONE
                        OCAA  1165 ;
                        OCAA  1166 ; IMPLICIT OUTPUTS:
                        OCAA  1167 ;
                        OCAA  1168 ;       RETSTATUS
                        OCAA  1169 ;
                        OCAA  1170 ; COMPLETION CODES:
                        OCAA  1171 ;
                        OCAA  1172 ;
                        OCAA  1173 ; SIDE EFFECTS:
                        OCAA  1174 ;
                        OCAA  1175 ;       A $WAKE WILL FORCE THE PROGRAM TO EXIT
                        OCAA  1176 ;
                        OCAA  1177 ;--
                        OCAA  1178 ;
                        OCAA  1179 LINKGONE:
                  0000  OCAA  1180          .WORD   0
00000593'EF  0000005C'EF  3C  OCAC  1181          MOVZWL  LINKMAIL+AST$T_BUF,MBXMSGTYP    ;GET REASON FROM THE LAST MAILBOX ME
                        OCB7  1182          $PUTMSG_S MSGVEC = MBXMSG                   ;OUTPUT MESSAGE TYPE
                        OCCA  1183          QUIT
```

```
                                        OCEF  1185          .SBTTL UNSDATMBX - MESSAGE IN TERMINAL MAILBOX
                                        OCEF  1186 ;++
                                        OCEF  1187 ; FUNCTIONAL DESCRIPTION:
                                        OCEF  1188 ;
                                        OCEF  1189 ;       HANDLES THE AST RESULTING FROM UNSOLICITED TERMINAL DATA OR HANGUP
                                        OCEF  1190 ;
                                        OCEF  1191 ; CALLING SEQUENCE:
                                        OCEF  1192 ;
                                        OCEF  1193 ;       CALLS   #0,UNSDATMBX
                                        OCEF  1194 ;
                                        OCEF  1195 ; INPUT PARAMETERS:
                                        OCEF  1196 ;
                                        OCEF  1197 ;       NONE
                                        OCEF  1198 ;
                                        OCEF  1199 ; IMPLICIT INPUTS:
                                        OCEF  1200 ;
                                        OCEF  1201 ;       UNSDAT
                                        OCEF  1202 ;
                                        OCEF  1203 ; OUTPUT PARAMETERS:
                                        OCEF  1204 ;
                                        OCEF  1205 ;       NONE
                                        OCEF  1206 ;
                                        OCEF  1207 ; IMPLICIT OUTPUTS:
                                        OCEF  1208 ;
                                        OCEF  1209 ;       RETSTATUS
                                        OCEF  1210 ;
                                        OCEF  1211 ; COMPLETION CODES:
                                        OCEF  1212 ;
                                        OCEF  1213 ;
                                        OCEF  1214 ; SIDE EFFECTS:
                                        OCEF  1215 ;
                                        OCEF  1216 ;       A MESSAGE IS SENT ON THE LINK.  IF THIS QIO FAILS, A $WAKE FORCES THE
                                        OCEF  1217 ;       PROGRAM TO EXIT.
                                        OCEF  1218 ;
                                        OCEF  1219 ;--
                                        OCEF  1220 ;
                                        OCEF  1221 UNSDATMBX:
                                 000C   OCEF  1222          .WORD   ^M<R2,R3>
                                        OCF1  1223
                                  51 0A DO    OCF1  1224          MOVL    #RDP$K_HEADERLEN,R1     ; Assume we send this much
                              52  00000144'EF 9E    OCF4  1225          MOVAB   UNSDAT+AST$T_BUF,R2     ; base address of RDP
                                  53 0C A2  3C    OCFB  1226          MOVZWL  RDP$K_HEADERLEN+2(R2),R3 ; Message code
                                0000'8F  53 B1    OCFF  1227          CMPW    R3,#MSG$_TRMUNSOLIC     ; Unsolicted data?
                                         06 12    0D04  1228          BNEQ    10$                     ; Branch if not
                                         00 B0    0D06  1229          MOVW    #RDP$C_TT_UNSOL,-
                                      02 A2       0D08  1230                  RDP$W_MOD(R2)           ; Unsolicited data
                                         23 11    0D0A  1231          BRB     20$                     ; Back to common code
                                                  0D0C  1232 10$:
                                0000'8F  53 B1    0D0C  1233          CMPW    R3,#MSG$_TRMBRDCST      ; Broadcast message?
                                         11 12    0D11  1234          BNEQ    18$                     ; branch if not
                                                  0D13  1235
                                  51 20 A2  3C    0D13  1236          MOVZWL  RDP$K_HEADERLEN+22(R2),R1 ; Length of broadcast
                                     51 22  C0    0D17  1237          ADDL2   #RDP$K_HEADERLEN+24,R1  ; Add rest + header
                                0A A2  51 B0    0D1A  1238          MOVW    R1,RDP$K_HEADERLEN(R2)  ; Save it in msg
                                02 A2  05 B0    0D1E  1239          MOVW    #RDP$C_TT_BRDCST, -
                                                  0D22  1240                  RDP$W_MOD(R2)           ; Broadcast
                                         0B 11    0D22  1241          BRB     20$
```

```
                              0D24   18$:
      0000'8F    53   B1      0D24           CMPW    R3,#MSG$_TRMHANGUP             ; Hangup?
                 25'  12      0D29           BNEQ    UNSMSGDONE+2                   ; Ignore it
                 01   B0      0D2B           MOVW    #RDP$C_TT_HANGUP,-
              02 A2           0D2D                   RDP$W_MOD(R2)                  ; Hangup
                              0D2F   20$:
      62     FFFF 8F  B0      0D2F           MOVW    #RDP$C_ATTN,RDP$W_OPCODE(R2) ; message type
08 A2  00000000'EF    B0      0D34           MOVW    TERMUNIT,RDP$W_UNIT(R2) ; Term unit num *** ?
                              0D3C
      50    011E'CF   9E      0D3C           MOVAB   W^UNSDAT,R0                    ; Address of AST block
               4E'AF  9E      0D41           MOVAB   B^UNSMSGDONE,-
                 60           0D44                   AST$L_STATE(R0)                ; Next state
      53    F4DB CF   9E      0D45           MOVAB   W^ASTHANDLER,R3                ; Ast address
               FCEF   30      0D4A           BSBW    WRITE_TO_NET                   ; Write message
                 04           0D4D           RET
                              0D4E
```

```
                        0D4E  1259          .SBTTL UNSMSGDONE - DO A NEW TERMINAL MAILBOX READ
                        0D4E  1260 ;++
                        0D4E  1261 ; FUNCTIONAL DESCRIPTION:
                        0D4E  1262 ;
                        0D4E  1263 ;       WHEN THE WRITE TO THE LINK COMPLETES,DO A NEW TERMINAL MAILBOX READ.
                        0D4E  1264 ;
                        0D4E  1265 ; CALLING SEQUENCE:
                        0D4E  1266 ;
                        0D4E  1267 ;       CALLS   #0,UNSMSGDONE
                        0D4E  1268 ;
                        0D4E  1269 ; INPUT PARAMETERS:
                        0D4E  1270 ;
                        0D4E  1271 ;       NONE
                        0D4E  1272 ;
                        0D4E  1273 ; IMPLICIT INPUTS:
                        0D4E  1274 ;
                        0D4E  1275 ;       NONE
                        0D4E  1276 ;
                        0D4E  1277 ; OUTPUT PARAMETERS:
                        0D4E  1278 ;
                        0D4E  1279 ;       NONE
                        0D4E  1280 ;
                        0D4E  1281 ; IMPLICIT OUTPUTS:
                        0D4E  1282 ;
                        0D4E  1283 ;       NONE
                        0D4E  1284 ;
                        0D4E  1285 ; COMPLETION CODES:
                        0D4E  1286 ;
                        0D4E  1287 ;
                        0D4E  1288 ; SIDE EFFECTS:
                        0D4E  1289 ;
                        0D4E  1290 ;       IF THE MAILBOX READ QIO FAILS, A $WAKE IS ISSUED TO CAUSE THE PROGRAM
                        0D4E  1291 ;       TO EXIT
                        0D4E  1292 ;
                        0D4E  1293 ;--
                        0D4E  1294 ;
                        0D4E  1295 UNSMSGDONE:
                    0000  0D4E  1296          .WORD   0
0000011E'EF   9C AF   9E  0D50  1297          MOVAB   UNSDATMBX,UNSDAT          ;NEW STATE
                        0D58  1298          $QIO_S  CHAN = TERMMBXCHAN -      ;SET UP UNSOLICITED DATA MBX READ
                        0D58  1299                  FUNC = #IO$_READVBLK -
                        0D58  1300                  IOSB = UNSDAT+AST$Q_IOSB -
                        0D58  1301                  ASTADR = ASTHANDLER -
                        0D58  1302                  ASTPRM = #UNSDAT -
                        0D58  1303                  P1 = UNSDAT+AST$T_BUF+RDP$K_HEADERLEN+2 -
                        0D58  1304                  P2 = #MAXMSG
                        0D8D  1305          ONERROR QUIT
                      04  0DB5  1306          RET
```

```
                     0DB6  1308          .SBTTL CNTRLC_AST - CONTROL-C & CONTROL-Y
                     0DB6  1309 ;++
                     0DB6  1310 ; FUNCTIONAL DESCRIPTION:
                     0DB6  1311 ;
                     0DB6  1312 ;       HANDLE THE AST RESULTING FROM A CONTROL-C OR A CONTROL-Y
                     0DB6  1313 ;
                     0DB6  1314 ; CALLING SEQUENCE:
                     0DB6  1315 ;
                     0DB6  1316 ;       CALLS   #0,CNTRLC_AST
                     0DB6  1317 ;
                     0DB6  1318 ; INPUT PARAMETERS:
                     0DB6  1319 ;
                     0DB6  1320 ;       NONE
                     0DB6  1321 ;
                     0DB6  1322 ; IMPLICIT INPUTS:
                     0DB6  1323 ;
                     0DB6  1324 ;       CNTRLYTIM
                     0DB6  1325 ;
                     0DB6  1326 ; OUTPUT PARAMETERS:
                     0DB6  1327 ;
                     0DB6  1328 ;       NONE
                     0DB6  1329 ;
                     0DB6  1330 ; IMPLICIT OUTPUTS:
                     0DB6  1331 ;
                     0DB6  1332 ;       CNTRLYTIM
                     0DB6  1333 ;
                     0DB6  1334 ; COMPLETION CODES:
                     0DB6  1335 ;
                     0DB6  1336 ;
                     0DB6  1337 ; SIDE EFFECTS:
                     0DB6  1338 ;
                     0DB6  1339 ;       A MESSAGE IS SENT ON THE LINK AND FOR ^Y THE AST IS REENABLED.
                     0DB6  1340 ;       TWO QUICK (LESS THAN 3 SEC) ^Y'S WILL ABORT THIS PROGRAM.
                     0DB6  1341 ;--
                     0DB6  1342 ;
                0000 0DB6  1343 CNTRLC_AST::    .WORD   0
00000114'EF   FFFF 8F  B0  0DB8  1344           MOVW    #RDP$C_ATTN,CNTRLCY+RDP$W_OPCODE
     0000'8F   04 AC   B1  0DC1  1345           CMPW    4(AP),#IO$M_CTRLCAST
               10      12  0DC7  1346           BNEQ    10$
00000116'EF    02     B0  0DC9  1347           MOVW    #RDP$C_TT_CTRLC,CNTRLCY+RDP$W_MOD
                      0DD0  1348
      000005A6'EF     94  0DD0  1349           CLRB    CNTRCFLAG               ; NO CONTROL-C ENABLES
             0083     31  0DD6  1350           BRW     30$
     0000'8F   04 AC   B1  0DD9  1351 10$:     CMPW    4(AP),#IO$M_CTRLYAST
               26      13  0DDF  1352           BEQL    20$
                      0DE1  1353           QUIT    4(AP)                   ; PROBABLY A HANGUP
             0088     30  0E07  1354 20$:     BSBW    CNTRLYTEST              ; CHECK FOR RECENT ^Y
                      0E0A  1355           $QIO_S  CHAN = CNTRLCHAN -      ; RE-ENABLE IT
                      0E0A  1356                   FUNC = #IO$_SETMODE!IO$M_CTRLYAST -
                      0E0A  1357                   P1 = CNTRLC_AST -
                      0E0A  1358                   P2 = #IO$M_CTRLYAST
                      0E30  1359
                      0E30  1360 ; WE WILL START TIMING WAITING FOR A SECOND ONE
                      0E30  1361
      0000059B'EF     95  0E30  1362           TSTB    CNTRLYTIM               ; ONLY SET ONE TIMER AT A TIME
               1D     12  0E36  1363           BNEQ    25$                     ; SKIP IF TIMER ENABLED
      0000059B'EF     96  0E38  1364           INCB    CNTRLYTIM               ; INDICATE WE HAVE ONE ^Y
```

```
                        0E3E  1365          $SETIMR_S DAYTIM = THREESEC -   ; TIME THREE SECONDS
                        0E3E  1366                    ASTADR = YTIMEDONE
00000116'EF   03    B0  0E55  1367 25$:     MOVW    #RDP$C_TT_CTRLY,CNTRLCY+RDP$W_MOD
                        0E5C  1368 30$:
0000011C'EF   00000000'EF   B0  0E5C  1369          MOVW    TERMUNIT,CNTRLCY+RDP$W_UNIT
                        0E67  1370
      07 00000000'EF    E9  0E67  1371          BLBC    CTERM_FLAG,40$          ; Branch if VAX
        0000'CF    6C   FA  0E6E  1372          CALLG   (AP),@^CTERM_CTRL_CY    ; Notify cterm module
                  13    11  0E73  1373          BRB     50$                     ; Exit
                        0E75  1374
50    000000EE'EF    9E  0E75  1375 40$:     MOVAB   CNTRLCY_MSG,R0          ; AST block
            51    0A    D0  0E7C  1376          MOVL    #RDP$K_HEADERLEN,R1     ; and length
                        0E7F  1377
      52    26 A0    9E  0E7F  1378          MOVAB   AST$T_BUF(R0),R2                ; Address of message
                  53    D4  0E83  1379          CLRL    R3                      ; No ast
                FBB4    30  0E85  1380          BSBW    WRITE_TO_NET            ; Write to net
                        0E88  1381 50$:
                  04    0E88  1382          RET
                        0E89  1383 ;
                        0E89  1384 ;
                        0E89  1385 ; ^Y TIMER HAS EXPIRED
                        0E89  1386 ;
                        0E89  1387 YTIMEDONE:
                0000    0E89  1388          .WORD   0
      0000059B'EF   94  0E8B  1389          CLRB    CNTRLYTIM       ;NO RECENT ^Y
                  04    0E91  1390          RET
                        0E92  1391 ;
                        0E92  1392 ;
                        0E92  1393 ; ASK ABORT QUESTION IF THERE WAS A RECENT ^Y
                        0E92  1394 ;
                        0E92  1395 CNTRLYTEST:
      0000059B'EF   95  0E92  1396          TSTB    CNTRLYTIM
                  01    12  0E98  1397          BNEQ    10$
                  05    0E9A  1398          RSB                     ;NO RECENT ^Y - CONTINUE
                        0E9B  1399 10$:     $GETMSG_S MSGID = #REM$_CNTRLY -
                        0E9B  1400                    MSGLEN = CNTRLYMSGBUF -
                        0E9B  1401                    BUFADR = CNTRLYMSGBUF -
                        0E9B  1402                    FLAGS = #1
                        0EB8  1403          $FAO_S  CTRSTR = CNTRLYMSGBUF -
                        0EB8  1404                  OUTLEN = CNTRLYQUESLEN -
                        0EB8  1405                  OUTBUF = CNTRLYQUESBUF -
                        0EB8  1406                  P1 = #REMOTENODE
                        0ED7  1407          $QIOW_S CHAN = READCHAN -       ;ASK ABOUT THE ^Y JUST TYPED
                        0ED7  1408                  EFN = #1 -
                        0ED7  1409                  FUNC = #IO$_READPROMPT!IO$M_CVTLOW -
                        0ED7  1410                  P1 = ANSBUF -
                        0ED7  1411                  P2 = #10 -
                        0ED7  1412                  P5 = #CNTRLYQUES -
                        0ED7  1413                  P6 = CNTRLYQUESLEN
59 8F   0000059C'EF   91  0F06  1414          CMPB    ANSBUF,#^A/Y/           ;DID HE SAY YES
                  01    13  0F0E  1415          BEQL    20$
                  05    0F10  1416          RSB                     ;HE SAID NO - SO CONTINUE
                        0F11  1417 ; ABORT
                        0F11  1418 20$:     QUIT    #SS$_NORMAL     ;NO STATUS MESSAGE
```

```
                                                .SBTTL VMS_INDREAD - READ INDIRECT COMMAND FILE
                                ;++
                                ; FUNCTIONAL DESCRIPTION:
                                ;
                                ;       READS FROM AN INDIRECT COMMAND FILE
                                ;
                                ; CALLING SEQUENCE:
                                ;
                                ;       JSB     VMS_INDREAD
                                ;
                                ; INPUT PARAMETERS:
                                ;
                                ;       R0 - AST Block
                                ;       R1 - Address to put input data
                                ;       R2 - QIO read modifiers (only IO$M_CVTLOW checked for)
                                ;       R3 - Size of request in bytes
                                ;
                                ; IMPLICIT INPUTS:
                                ;
                                ;       SYSINRAB
                                ;
                                ; OUTPUT PARAMETERS:
                                ;
                                ;       NONE
                                ;
                                ; IMPLICIT OUTPUTS:
                                ;
                                ;       INDFLAG
                                ;
                                ; COMPLETION CODES:
                                ;
                                ; SIDE EFFECTS:
                                ;
                                ;       ON AN EOF, FURTHER READS FROM THE INDIRECT FILE ARE DISABLED.
                                ;       THE PROGRAM WILL EXIT ON AN INDIRECT FILE READ.
                                ;
                                ;--
                                ;
                                VMS_INDREAD::
                                
                0043 8F     BB  PUSHR   #^M<R0,R1,R6>
00000000'EF     51  D0      MOVL    R1,SYSINRAB+RAB$L_UBF   ; Data address
00000000'EF     53  B0      MOVW    R3,SYSINRAB+RAB$W_USZ   ; Requested size
                            $GET    RAB = SYSINRAB          ; Read a record
00000000'8F     50  D1      CMPL    R0,#RMS$_NORMAL
                69  12      BNEQ    10$                     ; Problem
                0043 8F     POPR    #^M<R0,R1,R6>
                   04 A0    CLRQ    AST$Q_IOSB(R0)
51      00000000'EF     3C  MOVZWL  SYSINRAB+RAB$W_RSZ,R1
        00000000'EF     B0  MOVW    SYSINRAB+RAB$W_RSZ,-
                   06 A0            AST$Q_IOSB+2(R0)        ; Size of read
51      00000000'EF     C0  ADDL    SYSINRAB+RAB$L_UBF,R1   ; Find where to put terminator
                61  0D      MOVB    #^X0D,(R1)
              08 A0 0D      MOVB    #^X0D,AST$Q_IOSB+4(R0)  ; Set <CR> as terminator
              0A A0 01      MOVW    #1,AST$Q_IOSB+6(R0)     ; Terminator size
 04 A0    0000'8F     B0  MOVW    #SS$_NORMAL,AST$Q_IOSB(R0)
52      00000000'8F     D3  BITL    #IO$M_CVTLOW,R2
```

```
                     22  13  0F97  1477          BEQL    9$                              ;NO CASE CONVERSION
56   00000000'EF     D0  0F99  1478          MOVL    SYSINRAB+RAB$L_UBF,R6           ;BUFFER ADDRESS
51   00000000'EF     3C  0FA0  1479          MOVZWL  SYSINRAB+RAB$W_RSZ,R1           ;CHARACTERS TO CHECK
      61 8F    66     91  0FA7  1480 5$:      CMPB    (R6),#^A/a/
                      09  19  0FAB  1481          BLSS    8$                              ;NOT LOWER CASE
      7A 8F    66     91  0FAD  1482          CMPB    (R6),#^A/z/
                      03  14  0FB1  1483          BGTR    8$                              ;NOT LOWER CASE
          66   20     82  0FB3  1484          SUBB    #^X20,(R6)                      ;MAKE IT UPPER CASE
                      56  D6  0FB6  1485 8$:      INCL    R6                              ;NEXT
             EC 51    F5  0FB8  1486          SOBGTR  R1,5$
                         0FBB  1487 9$:      $DCLAST_S ASTADR = ASTHANDLER, -        ;SIMULATE A COMPETION AST
                         0FBB  1488                    ASTPRM = R0
                      04  0FCA  1489          RET
00000000'8F   50     D1  0FCB  1490 10$:     CMPL    R0,#RMS$_EOF                    ;ARE WE JUST DONE WITH THE FILE
                      18  12  0FD2  1491          BNEQ    20$                             ;REAL PROBLEM
                         0FD4  1492          $CLOSE  FAB = SYSINFAB                  ;DON'T NEED IT
      00000000'EF    94  0FE1  1493          CLRB    INDFLAG
          0043 8F    BA  0FE7  1494          POPR    #^M<R0,R1,R6>
                      05  0FEB  1495          RSB                                     ;GO DO THE REAL QIO
                         0FEC  1496 20$:     QUIT
```

```
                     1011  1498          .SBTTL GETBUF - GET A BUFFER
                     1011  1499 ;++
                     1011  1500 ; FUNCTIONAL DESCRIPTION:
                     1011  1501 ;
                     1011  1502 ;       GET A FREE BUFFER OR ALLOCATE ONE IF THERE ARE NONE.
                     1011  1503 ;
                     1011  1504 ; CALLING SEQUENCE:
                     1011  1505 ;
                     1011  1506 ;       JSB     GETBUF
                     1011  1507 ;
                     1011  1508 ; INPUT PARAMETERS:
                     1011  1509 ;
                     1011  1510 ;       NONE
                     1011  1511 ;
                     1011  1512 ; IMPLICIT INPUTS:
                     1011  1513 ;
                     1011  1514 ;       BUFQUEUE
                     1011  1515 ;
                     1011  1516 ; OUTPUT PARAMETERS:
                     1011  1517 ;
                     1011  1518 ;       R0 POINTS TO THE BUFFER
                     1011  1519 ;
                     1011  1520 ; IMPLICIT OUTPUTS:
                     1011  1521 ;
                     1011  1522 ;       NONE
                     1011  1523 ;
                     1011  1524 ; COMPLETION CODES:
                     1011  1525 ;
                     1011  1526 ;
                     1011  1527 ; SIDE EFFECTS:
                     1011  1528 ;
                     1011  1529 ;       MORE VIRTUAL MEMORY MAY BE ALLOCATED
                     1011  1530 ;
                     1011  1531 ;--
                     1011  1532 ;
                     1011  1533 GETBUF::
50   00000026'FF  0F 1011  1534          REMQUE  @BUFQUEUE,R0    ;GET A BUFFER
               01  1D 1018  1535          BVS     10$             ; BRANCH IF NONE
                     101A  1536 5$:
                  05 101A  1537          RSB                     ; RETURN
                     101B  1538
                     101B  1539 ; ALLOCATE A BUFFER
                     101B  1540
     0000002E'EF  9F 101B  1541 10$:     PUSHAB  BUFADR          ;BUFFER ADDRESS WILL BE RETURNED HERE
     00000032'EF  9F 1021  1542          PUSHAB  BUFSIZE         ;REQUESTED SIZE
00000000'GF   02  FB 1027  1543          CALLS   #2,G^LIB$GET_VM
                     102E  1544          ONERROR QUIT
                     1056  1545
                     1056  1546 .IF DF debug
                     1056  1547          movab   gotvm,R0        ; set message address
                     1056  1548          bsbw    log_ascic       ; log message
                     1056  1549 .endc
50   0000002E'EF  D0 1056  1550          MOVL    BUFADR,R0
               BB  11 105D  1551          BRB     5$              ; BRANCH TO EXIT
                     105F  1552
```

```
                     105F  1554          .SBTTL BUFFREE - FREE A BUFFER
                     105F  1555 ;++
                     105F  1556 ; FUNCTIONAL DESCRIPTION:
                     105F  1557 ;
                     105F  1558 ;       FREE A BUFFER.
                     105F  1559 ;
                     105F  1560 ; CALLING SEQUENCE:
                     105F  1561 ;
                     105F  1562 ;       JSB     BUFFREE
                     105F  1563 ;
                     105F  1564 ; INPUT PARAMETERS:
                     105F  1565 ;
                     105F  1566 ;       R0 POINTS TO THE BUFFER
                     105F  1567 ;
                     105F  1568 ; IMPLICIT INPUTS:
                     105F  1569 ;
                     105F  1570 ;       BUFQUEUE
                     105F  1571 ;
                     105F  1572 ; OUTPUT PARAMETERS:
                     105F  1573 ;
                     105F  1574 ;       NONE
                     105F  1575 ;
                     105F  1576 ; IMPLICIT OUTPUTS:
                     105F  1577 ;
                     105F  1578 ;       NONE
                     105F  1579 ;
                     105F  1580 ; COMPLETION CODES:
                     105F  1581 ;
                     105F  1582 ;
                     105F  1583 ; SIDE EFFECTS:
                     105F  1584 ;
                     105F  1585 ;       NONE
                     105F  1586 ;
                     105F  1587 ;--
                     105F  1588 ;
                     105F  1589 BUFFREE::
0C000026'EF    60    0E  105F  1590          INSQUE  (R0),BUFQUEUE           ;PUT BUFFER IN FREE LIST
                     05  1066  1591          RSB
```

```
                        1067  1593 .SBTTL  READ ONLY DATA
                        1067  1594
                000F'   1067  1595 TERMOPS:    .WORD   <2$-1$>/4
           0001 0000'   1069  1596 1$:         .WORD   IO$_READVBLK,OP_READ
           0001 0000'   106D  1597             .WORD   IO$_READLBLK,OP_READ
           0001 0000'   1071  1598             .WORD   IO$_READPBLK,OP_READ
           0101 0000'   1075  1599             .WORD   IO$_READPROMPT,OP_READ!OP_PRMPT
           0001 0000'   1079  1600             .WORD   IO$_TTYREADALL,OP_READ
           0101 0000'   107D  1601             .WORD   IO$_TTYREADPALL,OP_READ!OP_PRMPT
           0002 0000'   1081  1602             .WORD   IO$_WRITEVBLK,OP_WRITE
           0002 0000'   1085  1603             .WORD   IO$_WRITELBLK,OP_WRITE
           0002 0000'   1089  1604             .WORD   IO$_WRITEPBLK,OP_WRITE
           0003 0000'   108D  1605             .WORD   IO$_SETMODE,OP_SETMODE
           0003 0000'   1091  1606             .WORD   IO$_SETCHAR,OP_SETMODE
           0004 0000'   1095  1607             .WORD   IO$_SENSEMODE,OP_SENSEMODE
           0004 0000'   1099  1608             .WORD   IO$_SENSECHAR,OP_SENSEMODE
           0005 0000'   109D  1609             .WORD   IO$_ACPCONTROL,OP_CANCEL
           0006 FFFF    10A1  1610             .WORD   -1,OP_BRDCST
                        10A5  1611 2$:
                        10A5  1612
00000000 00000000'00000002   10A5  1613 ACSIGNORE: .LONG  2,REM$_ACSIGN,0
                        10B1  1614
00000000 00000000'00000002   10B1  1615 BADOUTBAND: .LONG 2,REM$_BADOUTBAND,0
                        10BD  1616
        FFFFFFFF FE363C80   10BD  1617 THREESEC:   .LONG  -10*1000*1000*3,-1          ;THREE SECOND TIMER
                        10C5  1618
```

```
                        10C5  1620          .SBTTL  READ WRITE DATA
                    00000000  1621          .PSECT  _RTPAD,LONG
                        0000  1622
                    0000000C  0000  1623  CHARBUF:         .BLKB   12
                        000C  1624
                        0000  000C  1625  OUTBANDINC::    .WORD   0           ;OUT OF BAND (INCLUDE) AST CHANNEL
                        0000  000E  1626  OUTBANDEXC::    .WORD   0           ;OUT OF BAND (EXCLUDE) AST CHANNEL
                        0000  0010  1627  OUTBANDABO::    .WORD   0           ;OUT OF BAND (ABORT)   AST CHANNEL
                        0012  1628
                    00000000  0012  1629  CTRLO_CHAN:      .LONG   0           ; CHANNEL FOR ^O OUT OF BANDS
                    00000000  0016  1630  CTRLO_MASK:      .LONG   0
                    00008000  001A  1631                   .LONG   1@<^A/O/-^A/@/>
                        001E  1632
                    00000026  001E  1633  RTERMDES:   .BLKL  2               ;TERMINATOR CHARACTER DESCRIPTOR
                        0026  1634
00000026'00000026'  0026  1635  BUFQUEUE:   .LONG  BUFQUEUE,BUFQUEUE      ;EMPTY BUFFER QUEUE
                        002E  1636
                    00000000  002E  1637  BUFADR:     .LONG  0               ;THE ADDRESS OF AN ALLOCATED BUFFER GOES HERE
                        0032  1638
                    00000440  0032  1639  BUFSIZE:    .LONG  AST$T_BUF+MAXMSG       ;BUFFER SIZE
                        0036  1640
                    00000B39' 0036  1641  LINKMAIL:         .LONG   LNKMBXDONE
                    00000084  003A  1642                     .BLKB   AST$T_BUF+40-4
```

```
                        0084  1644 .SBTTL  Protocol Message buffers
                        0084  1645
                        0084  1646 ;
                        0084  1647 ; BIND and configuration data message
                        0084  1648 ;
                        0084  1649
          000000AA      0084  1650 CONFIG_MSG:      .BLKB   AST$T_BUF          ;CONFIGURATION MESSAGE
    00 01 01 01         00AA  1651 CONFIG_MSG_ST:   .BYTE   1,1,1,0            ; BIND,V1, ECO=1,customer ECO
      0004 0007         00AE  1652                  .WORD   7,4                ; VMS, support mask
                        00B2  1653 CONFIG_CHAR:
          000000BE      00B2  1654                  .BLKL   3                  ; Characteristics buffer
          00000014      00BE  1655 CONFIG_MSGLEN = .-CONFIG_MSG_ST
                        00BE  1656
                        00BE  1657 ;
                        00BE  1658 ; Unsolicited data message (init)
                        00BE  1659 ;
                        00BE  1660
          000000E4      00BE  1661 INIT_MSG:        .BLKB   AST$T_BUF
              FFFF      00E4  1662 1$:              .WORD   RDP$C_ATTN
0000 0000 0000 0000     00E6  1663                  .WORD   RDP$C_TT_UNSOL,0,0,0    ; Fake unsolicited data
          0000000A      00EE  1664 INIT_MSGLEN = .-1$
                        00EE  1665
                        00EE  1666 ;
                        00EE  1667 ; CONTROL C or CONTROL Y (^C or ^Y) out of band message
                        00EE  1668 ;
                        00EE  1669
          00000114      00EE  1670 CNTRLCY_MSG:     .BLKB   AST$T_BUF
          0000011E      0114  1671 CNTRLCY:         .BLKB   RDP$K_HEADERLEN  ; LINK MESSAGE FOR CONTROL C OR Y
          0000000A      011E  1672 CNTRLCY_MSGLEN = .-CNTRLCY
                        011E  1673
          00000CEF'     011E  1674 UNSDAT::         .LONG   UNSDATMBX        ; VMS AND CTERMRT ONLY???***
          00000562      0122  1675                  .BLKB   AST$T_BUF+MAXMSG ; FOR UNSOLICITED DATA MAILBOX
                        0562  1676
                        0562  1677 ;
                        0562  1678 ; Out of band message
                        0562  1679 ;
                        0562  1680
          0000056C      0562  1681 OUTBANDMSG: .BLKB  RDP$K_HEADERLEN
                00      056C  1682 OUTBANDCHAR: .BYTE 0
                        056D  1683 ;
                        056D  1684 ; Cancel message
                        056D  1685 ;
                        056D  1686
00000000 FFFFFFFE       056D  1687 CANMSG:     .LONG   RDP$C_END,0
              0000      0575  1688             .WORD   0
          0000057F      0577  1689             .BLKQ   1
                        057F  1690 ;
          00000587      057F  1691 BRDDESC:    .BLKL   2               ;DESCRIPTOR FOR BROADCASTS
                        0587  1692 ;
00000001 00000000'00000004  0587  1693 MBXMSG:         .LONG   4,REM$_NETMBX,1
          00000000      0593  1694 MBXMSGTYP:      .LONG   0
          00000000      0597  1695 LINKERR:        .LONG    0
                        059B  1696 ;
                00      059B  1697 CNTRLYTIM: .BYTE   0                 ;RECENT ^Y INDICATOR
                        059C  1698 ;
          000005A6      059C  1699 ANSBUF:     .BLKB   10               ;RECEIVE ANSWER TO ^Y QUESTION
                        05A6  1700 ;
```

```
                00  05A6  1701 CNTRCFLAG:: .BYTE  0            ;INDICATE CONTROL-C ENABLES
                    05A7  1702
          00000000  05A7  1703 READQIO::  .LONG   0            ;ID OF CURRENT READ REQUEST
                    05AB  1704 ;
          00000000  05AB  1705 WRITEQIO:: .LONG   0            ;ID OF CURRENT WRITE REQUEST
                    05AF  1706 ;
000005AF'000005AF'  05AF  1707 READQ::    .LONG   READQ,READQ  ;QUEUE OF PENDING READS
                    05B7  1708 ;
000005B7'000005B7'  05B7  1709 WRITEQ::   .LONG   WRITEQ,WRITEQ        ;QUEUE OF PENDING WRITES
                    05BF  1710 ;
          000005C7  05BF  1711 REMOTENODE::    .BLKL   2               ;REMOTE NODE NAME
                    05C7  1712 ;
                    05C7  1713
                    05C7  1714 ;
          000005CF  05C7  1715 INCMASK:   .BLKL  2             ;OUT OF BAND INCLUDE MASK
          000005D7  05CF  1716 EXCMASK:   .BLKL  2             ;OUT OF BAND EXCLUDE MASK
                    05D7  1717 ;
              0000' 05D7  1718 FIRST_READ: .WORD IO$M_PURGE    ; Don't purge type ahead (on the first read)
                    05D9  1719 ;
                    05D9  1720 CNTRLYMSGBUF:
000005E1'00000100   05D9  1721          .LONG   256,1$               ;BUFFER TO HOLD CONTROL Y QUESTION (BEFORE F
          000006E1  05E1  1722 1$:      .BLKB   256
                    06E1  1723 ;
                    06E1  1724 CNTRLYQUESLEN:
          00000000  06E1  1725          .LONG   0                    ;FINAL LENGTH OF QUESTION
                    06E5  1726 ;
                    06E5  1727 CNTRLYQUESBUF:
000006ED'00000100   06E5  1728          .LONG   256,CNTRLYQUES       ;BUFFER TO HOLD THE FINAL QUESTION
                    06ED  1729 CNTRLYQUES:
          000007ED  06ED  1730          .BLKB   256
                    07ED  1731
          00000000  07ED  1732 MAX_SENDSIZE::  .LONG   0            ; Maximum network send size ***
                    07F1  1733
                    07F1  1734
                    07F1  1735 .END    ; of module VMSRT
```

```
$$.TMP1                 =  00000001
$$.TMP2                 =  000000EF
$$T1                    =  00000000
$$T2                    =  00000004
ACSIGNORE                  000010A5  R      02
ANSBUF                     0000059C  R      03
ASSEM_TRACE             =  00000001
AST$L_STATE             =  00000000
AST$Q_IOSB              =  00000004
AST$T_BUF               =  00000026
AST$W_OPCODE            =  0000000C
ASTHANDLER                 00000224  RG     02
BADOUTBAND                 000010B1  R      02
BRDDESC                    0000057F  R      03
BROADCAST                  000008BC  R      02
BUFADR                     0000002E  R      03
BUFFREE                    0000105F  RG     02
BUFQUEUE                   00000026  R      03
BUFSIZE                    00000032  R      03
CANCELIO                   000007BD  R      02
CANMSG                     0000056D  R      03
CHARBUF                    00000000  R      03
CHAR_BLOCK                 ********    X    02
CNTRCFLAG                  000005A6  RG     03
CNTRLCHAN                  ********    X    02
CNTRLCY                    00000114  R      03
CNTRLCY_MSG                000000EE  R      03
CNTRLCY_MSGLEN          =  0000000A
CNTRLC_AST                 00000DB6  RG     02
CNTRLYMSGBUF               000005D9  R      03
CNTRLYQUES                 000006ED  R      03
CNTRLYQUESBUF              000006E5  R      03
CNTRLYQUESLEN              000006E1  R      03
CNTRLYTEST                 00000E92  R      02
CNTRLYTIM                  0000059B  R      03
CONFIG_CHAR                000000B2  R      03
CONFIG_MSG                 00000084  R      03
CONFIG_MSGLEN           =  00000014
CONFIG_MSG_ST              000000AA  R      03
CTERM_CTRLO_AST            ********    X    02
CTERM_CTRL_CY              ********    X    02
CTERM_FLAG                 ********    X    02
CTERM_LINKRECV             ********    X    02
CTERM_LNKWRTDONE           ********    X    02
CTERM_Q;ODONE              ********    X    02
CTERM_UNSMSGDONE           ********    X    02
CTP$B_PRO_MSGTYPE       =  00000026
CTRLO_CHAN                 00000012  R      03
CTRLO_MASK                 00000016  R      03
CT_BIND_ACC_MSG            ********    X    02
CT_BIND_MSGLEN             ********    X    02
DBG$LINKRECV               ********    X    02
DBG$LINKWRITE              ********    X    02
DBG$TRACE_IO               ********    X    02
DECNETERR                  ********    X    02
DEVNAM                     ********    X    02
DEVNAMLEN                  ********    X    02
EXCMASK                    000005CF  R      03
FINALACS                   ********    X    02
FIRST_READ                 000005D7  R      03
GETBUF                     00001011  RG     02
INCMASK                    000005C7  R      03
INDFLAG                    ********    X    02
INIT_MSG                   000000BE  R      03
INIT_MSGLEN             =  0000000A
IO$M_CTRLCAST              ********    X    02
IO$M_CTRLYAST              ********    X    02
IO$M_CVTLOW                ********    X    02
IO$M_HANGUP                ********    X    02
IO$M_INCLUDE               ********    X    02
IO$M_OUTBAND               ********    X    02
IO$M_PURGE                 ********    X    03
IO$V_EXTEND                ********    X    02
IO$_ACPCONTROL             ********    X    02
IO$_READLBLK               ********    X    02
IO$_READPBLK               ********    X    02
IO$_READPROMPT             ********    X    02
IO$_READVBLK               ********    X    02
IO$_SENSECHAR              ********    X    02
IO$_SENSEMODE              ********    X    02
IO$_SETCHAR                ********    X    02
IO$_SETMODE                ********    X    02
IO$_TTYREADALL             ********    X    02
IO$_TTYREADPALL            ********    X    02
IO$_WRITELBLK              ********    X    02
IO$_WRITEPBLK              ********    X    02
IO$_WRITEVBLK              ********    X    02
LIB$GET_VM                 ********    X    02
LINKCHAN                   ********    X    02
LINKERR                    00000597  R      03
LINKGONE                   00000CAA  R      02
LINKMAIL                   00000036  R      03
LINKRECV                   000006DA  RG     02
LINKRECV_ERR               00000672  R      02
LNKMBXDONE                 00000B39  R      02
LNKWRTDONE                 00000A9F  RG     02
MAILCHAN                   ********    X    02
MAXMSG                  =  0000041A
MAX_SENDSIZE               000007ED  RG     03
MBXMSG                     00000587  R      03
MBXMSGTYP                  00000593  R      03
MSG$_ABORT                 ********    X    02
MSG$_DISCON                ********    X    02
MSG$_EXIT                  ********    X    02
MSG$_PATHLOST              ********    X    02
MSG$_THIRDPARTY            ********    X    02
MSG$_TRMBRDCST             ********    X    02
MSG$_TRMHANGUP             ********    X    02
MSG$_TRMUNSOLIC            ********    X    02
OP_BRDCST               =  00000006
OP_CANCEL               =  00000005
OP_PRMPT                =  00000100
OP_READ                 =  00000001
OP_SENSEMODE            =  00000004
```

```
OP_SETMODE                      =  00000003
OP_WRITE                        =  00000002
OUTBANDABO                         00000010 RG     03
OUTBANDAST                         00000C37 R      02
OUTBANDCHAR                        0000056C R      03
OUTBANDEXC                         0000000E RG     03
OUTBANDINC                         0000000C RG     03
OUTBANDMSG                         00000562 R      03
OUTBAND_ERR                        00000622 R      02
PROCMSG                            0000022F R      02
PROCMSG_EXIT                       00000621 R      02
PROTO_ECO                          ********   X    02
QIODONE                            0000095B RG     02
QIO_ERR                            000005B3 R      02
RAB$L_UBF                          ********   X    02
RAB$W_RSZ                          ********   X    02
RAB$W_USZ                          ********   X    02
RDP$B_TT_OUTBAND                =  0000000A
RDP$C_ATTN                      =  FFFFFFFF
RDP$C_END                       =  FFFFFFFE
RDP$C_TT_BRDCST                 =  00000005
RDP$C_TT_CTRLC                  =  00000002
RDP$C_TT_CTRLY                  =  00000003
RDP$C_TT_HANGUP                 =  00000001
RDP$C_TT_OUTBAND                =  00000006
RDP$C_TT_UNSOL                  =  00000000
RDP$K_HEADERLEN                 =  0000000A
RDP$L_REFID                     =  00000004
RDP$L_TT_ASTPRM                 =  0000000A
RDP$L_TT_BCNT                   =  0000000A
RDP$L_TT_CARCON                 =  0000000E
RDP$L_TT_CHAR2                  =  0000001E
RDP$L_TT_FILL                   =  00000016
RDP$L_TT_PARITY                 =  0000001A
RDP$L_TT_SPEED                  =  00000012
RDP$L_TT_TIMOUT                 =  0000000E
RDP$Q_STATUS                    =  0000000A
RDP$Q_TT_CHAR                   =  0000000A
RDP$Q_TT_SCHAR                  =  00000012
RDP$T_TT_RDATA                  =  00000012
RDP$T_TT_TERM                   =  00000012
RDP$T_TT_WDATA                  =  00000012
RDP$W_MOD                       =  00000002
RDP$W_OPCODE                    =  00000000
RDP$W_UNIT                      =  00000008
READCHAN                           ********   X    02
READMSG                            0000024A R      02
READQ                              000005AF RG     03
READQIO                            000005A7 RG     03
RECORD_QUIT                        ********     X  02
REM$_ACSIGN                        ********     X  02
REM$_BADOUTBAND                    ********     X  02
REM$_CNTRLY                        ********     X  02
REM$_NETMBX                        ********     X  03
REMOTENODE                         000005BF RG     03
RETSTATUS                          ********     X  02
RMS$_EOF                           ********     X  02
RMS$_NORMAL                        ********     X  02
RT$C_LINKEFN                    =  00000001
RTERMDES                           0000001E R      03
RTLOG$V_TRACE                   =  00000001
RTLOG_FLAGS                        ********     X  02
SENSEMSG                           000005C8 R      02
SETMSG                             000003A3 R      02
SS$_ABORT                          ********     X  02
SS$_EXQUOTA                        ********     X  02
SS$_NORMAL                         ********     X  02
SS$_PATHLOST                       ********     X  02
SS$_THIRDPARTY                     ********     X  02
SYS$ASSIGN                         ********    GX  02
SYS$BRDCST                         ********    GX  02
SYS$CANCEL                         ********    GX  02
SYS$CLOSE                          ********    GX  02
SYS$DCLAST                         ********    GX  02
SYS$FAO                            ********     X  02
SYS$GET                            ********    GX  02
SYS$GETMSG                         ********    GX  02
SYS$PUTMSG                         ********    GX  02
SYS$QIO                            ********    GX  02
SYS$QIOW                           ********    GX  02
SYS$SETAST                         ********    GX  02
SYS$SETIMR                         ********    GX  02
SYS$WAITFR                         ********    GX  02
SYS$WAKE                           ********    GX  02
SYSINFAB                           ********     X  02
SYSINRAB                           ********     X  02
TERMMBXCHAN                        ********     X  02
TERMOPS                            00001067 R      02
TERMUNIT                           ********     X  02
THREESEC                           000010BD R      02
TRACE_CONTINUE                     000006E9 R      02
TRACE_CONTINUE2                    00000A42 R      02
TRACE_RECV                         0000065A R      02
TRACE_WRITE                        00000A8F R      02
TTYDESC                            ********     X  02
UNSDAT                             0000011E RG     03
UNSDATMBX                          00000CEF R      02
UNSMSGDONE                         00000D4E R      02
VMSRT                              00000000 RG     02
VMS_INDREAD                        00000F3A RG     02
WAKEFLAG                           ********     X  02
WRITECHAN                          ********     X  02
WRITEMSG                           00000335 R      02
WRITEQ                             000005B7 RG     03
WRITEQIO                           000005AB RG     03
WRITE_TO_NET                       00000A3C RG     02
WRITE_TO_NETX                      00000A30 RG     02
WRITE_TO_NET_SYNC                  000009C4 RG     02
YTIMEDONE                          00000E89 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	00000000 (0.)	01 (1.)	NOPIC USR CON ABS LCL NOSHR EXE RD WRT NOVEC BYTE
RTPAD	000010C5 (4293.)	02 (2.)	NOPIC USR CON REL LCL NOSHR EXE RD NOWRT NOVEC BYTE
_RTPAD	000007F1 (2033.)	03 (3.)	NOPIC USR CON REL LCL NOSHR EXE RD WRT NOVEC LONG

Performance indicators

Phase	Page faults	CPU Time	Elapsed Time
Initialization	37	00:00:00.05	00:00:02.30
Command processing	123	00:00:00.46	00:00:03.35
Pass 1	401	00:00:10.19	00:00:38.89
Symbol table sort	0	00:00:00.98	00:00:02.22
Pass 2	288	00:00:03.24	00:00:10.96
Symbol table output	26	00:00:00.11	00:00:00.30
Psect synopsis output	2	00:00:00.02	00:00:00.02
Cross-reference output	0	00:00:00.00	00:00:00.00
Assembler run totals	880	00:00:15.05	00:00:58.04

The working set limit was 1950 pages.
91005 bytes (178 pages) of virtual memory were used to buffer the intermediate code.
There were 60 pages of symbol table space allocated to hold 859 non-local and 100 local symbols.
1735 source lines were read in Pass 1, producing 40 object records in Pass 2.
36 pages of virtual memory were used to define 33 macros.

Macro library statistics

Macro library name	Macros defined
_255DUA28:[RTPAD.OBJ]RTPAD.MLB;1	5
_255DUA28:[SYS.OBJ]LIB.MLB;1	1
_255DUA28:[SYSLIB]STARLET.MLB;2	24
TOTALS (all libraries)	30

968 GETS were required to define 30 macros.

There were no errors, warnings or information messages.

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