J 6

```
MMM         MMM   TTTTTTTTTTTTTTT     AAAAAAAAA         AAAAAAAAA        CCCCCCCCCCCC   PPPPPPPPPPPP
MMM         MMM   TTTTTTTTTTTTTTT     AAAAAAAAA         AAAAAAAAA        CCCCCCCCCCCC   PPPPPPPPPPPP
MMM         MMM   TTTTTTTTTTTTTTT     AAAAAAAAA         AAAAAAAAA        CCCCCCCCCCCC   PPPPPPPPPPPP
MMMMMM   MMMMMM         TTT        AAA         AAA   AAA         AAA  CCC               PPP         PPP
MMMMMM   MMMMMM         TTT        AAA         AAA   AAA         AAA  CCC               PPP         PPP
MMMMMM   MMMMMM         TTT        AAA         AAA   AAA         AAA  CCC               PPP         PPP
MMM   MMM   MMM         TTT        AAA         AAA   AAA         AAA  CCC               PPP         PPP
MMM   MMM   MMM         TTT        AAA         AAA   AAA         AAA  CCC               PPP         PPP
MMM   MMM   MMM         TTT        AAA         AAA   AAA         AAA  CCC               PPP         PPP
MMM         MMM         TTT        AAA         AAA   AAA         AAA  CCC               PPPPPPPPPPPP
MMM         MMM         TTT        AAA         AAA   AAA         AAA  CCC               PPPPPPPPPPPP
MMM         MMM         TTT        AAA         AAA   AAA         AAA  CCC               PPPPPPPPPPPP
MMM         MMM         TTT        AAAAAAAAAAAAAAA   AAAAAAAAAAAAAAA  CCC               PPP
MMM         MMM         TTT        AAAAAAAAAAAAAAA   AAAAAAAAAAAAAAA  CCC               PPP
MMM         MMM         TTT        AAAAAAAAAAAAAAA   AAAAAAAAAAAAAAA  CCC               PPP
MMM         MMM         TTT        AAA         AAA   AAA         AAA  CCC               PPP
MMM         MMM         TTT        AAA         AAA   AAA         AAA  CCC               PPP
MMM         MMM         TTT        AAA         AAA   AAA         AAA  CCC               PPP
MMM         MMM         TTT        AAA         AAA   AAA         AAA     CCCCCCCCCCCC   PPP
MMM         MMM         TTT        AAA         AAA   AAA         AAA     CCCCCCCCCCCC   PPP
MMM         MMM         TTT        AAA         AAA   AAA         AAA     CCCCCCCCCCCC   PPP
```

```
**FILE**ID**LOGIO                                B  5



LL            OOOOOO      GGGGGGGG    IIIIII      OOOOOO
LL            OOOOOO      GGGGGGGG    IIIIII      OOOOOO
LL          OO      OO  GG              II      OO      OO
LL          OO      OO  GG              II      OO      OO
LL          OO      OO  GG              II      OO      OO
LL          OO      OO  GG              II      OO      OO
LL          OO      OO  GG              II      OO      OO
LL          OO      OO  GG              II      OO      OO
LL          OO      OO  GG  GGGGGG      II      OO      OO
LL          OO      OO  GG  GGGGGG      II      OO      OO
LL          OO      OO  GG      GG      II      OO      OO      ....
LL          OO      OO  GG      GG      II      OO      OO      ....
LLLLLLLLLL    OOOOOO      GGGGGG      IIIIII      OOOOOO        ....
LLLLLLLLLL    OOOOOO      GGGGGG      IIIIII      OOOOOO        ....

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
```

LOG
V04

```
0
0 MODULE LOGIO (LANGUAGE (BLISS32) ,
0               IDENT = 'V04-000'
0               ) =
1 BEGIN
1 !
1 !****************************************************************************
1 !*                                                                          *
1 !*  COPYRIGHT (c) 1978, 1980, 1982, 1984 BY                                 *
1 !*  DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASSACHUSETTS.                  *
1 !*  ALL RIGHTS RESERVED.                                                    *
1 !*                                                                          *
1 !*  THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND COPIED   *
1 !*  ONLY IN  ACCORDANCE WITH  THE  TERMS  OF  SUCH  LICENSE  AND WITH THE   *
1 !*  INCLUSION OF THE ABOVE COPYRIGHT NOTICE. THIS SOFTWARE OR  ANY  OTHER   *
1 !*  COPIES THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY   *
1 !*  OTHER PERSON.  NO TITLE TO AND OWNERSHIP OF  THE  SOFTWARE IS  HEREBY   *
1 !*  TRANSFERRED.                                                            *
1 !*                                                                          *
1 !*  THE INFORMATION IN THIS SOFTWARE IS  SUBJECT TO CHANGE WITHOUT NOTICE   *
1 !*  AND  SHOULD  NOT  BE  CONSTRUED AS  A COMMITMENT BY DIGITAL EQUIPMENT   *
1 !*  CORPORATION.                                                            *
1 !*                                                                          *
1 !*  DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE  OR  RELIABILITY OF ITS   *
1 !*  SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL.                 *
1 !*                                                                          *
1 !*                                                                          *
1 !****************************************************************************
1
1 !++
1 !
1 ! FACILITY:  MTAACP
1 !
1 ! ABSTRACT:
1 !       This module handles logical IO.
1 !
1 !
1 ! ENVIRONMENT:
1 !
1 !       Starlet operating system, including privileged system services
1 !       and internal exec routines.
1 !
1 !--
1
1 !
1 !
1 ! AUTHOR:  D. H. GILLESPIE,      CREATION DATE:  14-JUL-1977
1 !
1 ! MODIFIED BY:
1 !
1 !       V03-009 HH0041           Hai Huang                 24-Jul-1984
1 !               Remove REQUIRE 'LIBD$:[VMSLIB.OBJ]MOUNTMSG.B32'.
1 !
1 !       V03-008 ROW0258          Ralph O. Weber            21-NOV-1983
1 !                       The Paul Painter Memorial Enhancement
1 !               Named for one of the unfortunate customers who suffered much
1 !               to determine the great UCB$L_MT_RECORD secret while trying to
```

```
;  58      0058  1 !                  create a user-written magtape driver, this change eliminates
;  59      0059  1 !                  use of the device dependent field, UCB$L_MT_RECORD in favor of
;  60      0060  1 !                  the device independent field, UCB$L_RECORD.
;  61      0061  1 !
;  62      0062  1 !        V03-007 STJ3101          Steven T. Jeffreys,      24-May-1983
;  63      0063  1 !                Removed reference to obsolete IO$M_INTSKIP.
;  64      0064  1 !
;  65      0065  1 !        V03-006 MMD0172          Meg Dumont,       9-May-1983  15:14
;  66      0066  1 !                Fix to make USER_STATUS defined consistently within module
;  67      0067  1 !
;  68      0068  1 !        V03-005 MMD0101          Meg Dumont,       17-Feb-1983  12:56
;  69      0069  1 !                Use routine GET_DEV_NAME to get tape unit device name.
;  70      0070  1 !
;  71      0071  1 !        V03-004 MMD0004          Meg Dumont,       21-Jan-1983  12:32
;  72      0072  1 !
;  73      0073  1 !                Change COMPLETE_VIO so it always puts an ABORT status
;  74      0074  1 !                in the IO to complete
;  75      0075  1 !
;  76      0076  1 !        V03-003 MMD0003          Meg Dumont,       3-Jan-1983  15:39
;  77      0077  1 !                Add the modifier IO$M_CLRSEREXCP to all QIO's issued by MTAACP,
;  78      0078  1 !                necessary for the MSCP tape drives.
;  79      0079  1 !
;  80      0080  1 !        V03-002 MMD0002          Meg Dumont,       5-Nov-1982  16:38
;  81      0081  1 !                Support for read record reverse. Support for the streaming tape
;  82      0082  1 !                drives, which forces all outstanding I/O's to complete
;  83      0083  1 !                before processing continues on a Serious Exception.  Support
;  84      0084  1 !                for completing all I/O's to the user when USER EOT mode is set.
;  85      0085  1 !
;  86      0086  1 !        V03-001 MMD0001          Meg Dumont,       2-Jul-1982  12:08
;  87      0087  1 !                Add RETURN to ISSUE_IO to fix getting INFO message
;  88      0088  1 !
;  89      0089  1 !        V02-004 REFORMAT         Maria del C. Nasr        30-Jun-1980
;  90      0090  1 !
;  91      0091  1 !        00003   MCN0009          Maria del C. Nasr        20-Nov-1979   12:00
;  92      0092  1 !                The STATUS code from a QIO is returned in low 16 bits of
;  93      0093  1 !                longword.  Fix check for STATUS to <0,16>.
;  94      0094  1 !
;  95      0095  1 !!
;  96      0096  1 !**
;  97      0097  1
;  98      0098  1 LIBRARY 'SYS$LIBRARY:LIB.L32';
;  99      0099  1
; 100      0100  1
; 101      0101  1 REQUIRE 'SRC$:MTADEF.B32';
; 102      0485  1
; 103      0486  1 FORWARD ROUTINE
; 104      0487  1     ADJTM        : COMMON_CALL NOVALUE,            ! adjust tape mark count
; 105      0488  1     CHCK_IO_CLR_EXCP : COMMON_CALL NOVALUE,        ! Check to get all io's from
; 106      0489  1                                                    ! device and clear serious expt
; 107      0490  1     COMPLETE_VIO : COMMON_CALL NOVALUE,            ! complete virtual io in error
; 108      0491  1     ISSUE_IO     : L$ISSUE_IO,                     ! issue IO
; 109      0492  1     READ_BLOCK   : COMMON_CALL,                    ! read logical block
; 110      0493  1     READ_BLOCK_REVERSE  : COMMON_CALL,             ! read backwards one logical block
; 111      0494  1     REPOSITION   : L$REPOSITION NOVALUE,           ! reposition tape
; 112      0495  1     RESTORE_POS  : NOVALUE COMMON_CALL,
; 113      0496  1     SPACE        : COMMON_CALL,                    ! space blocks
; 114      0497  1     SPACE_TM     : COMMON_CALL NOVALUE,            ! space tape marks
;  115        0498  1        UNBLOCK_SPACE          : COMMON_CALL NOVALUE,   ! unblock for SPACE_TM
;  116        0499  1        WRITE_BLOCK : COMMON_CALL NOVALUE,               ! write logical block
;  117        0500  1        WRITE_TM     : NOVALUE L$WRITE_TM;               ! write one tape mark
;  118        0501  1
;  119        0502  1   EXTERNAL
;  120        0503  1        CURRENT_UCB : REF BBLOCK,                        ! address of current unit control block
;  121        0504  1        IO_CHANNEL,                                      ! address of IO channel
;  122        0505  1        IO_STATUS,                                       ! IO status
;  123        0506  1        USER_STATUS : VECTOR [2];                        ! user status
;  124        0507  1
;  125        0508  1   EXTERNAL ROUTINE
;  126        0509  1        GET_DEV_NAME           : COMMON_CALL NOVALUE,   ! given UCB addr get dev name
;  127        0510  1        IO_DONE,                                         ! complete IO
;  128        0511  1        MOUNT_VOL              : COMMON_CALL,           ! mount volume
;  129        0512  1        PRINT_OPR_MSG          : L$PRINT_OPR_MSG,       ! print an operator message
;  130        0513  1        RESET_UNIT             : COMMON_CALL,
;  131        0514  1        SYS$QIOW               : ADDRESSING_MODE (ABSOLUTE);   ! queue io request
;  132        0515  1
```

```
1 GLOBAL ROUTINE READ_BLOCK (ADDR, LEN) : COMMON_CALL =
1
1 !++
1 !
1 ! FUNCTIONAL DESCRIPTION:
1 !       This routine reads a logical record from magnetic tape.
1 !
1 ! CALLING SEQUENCE:
1 !       READ_BLOCK(ARG1,ARG2)
1 !
1 ! INPUT PARAMETERS:
1 !       ARG1 - address for data
1 !       ARG2 - length to read
1 !
1 ! IMPLICIT INPUTS:
1 !       IO_CHANNEL
1 !
1 ! OUTPUT PARAMETERS:
1 !       ARG1 - address for data
1 !
1 ! IMPLICIT OUTPUTS:
1 !       USER_STATUS, IO_STATUS
1 !
1 ! ROUTINE VALUE:
1 !       0 - tm encountered
1 !       1 - successful read
1 !
1 ! SIDE EFFECTS:
1 !       none
1 !
1 ! ERRORS:
1 !       Primary status is I/O error returned from driver
1 !       SS$_FCPREADERR - read failure
1 !
1 !--
1
2     BEGIN
2
2     EXTERNAL REGISTER
2         COMMON_REG;
2
2     LOCAL
2         STATUS;                                   ! IO status
2
2     STATUS = ISSUE_IO(IO$_READLBLK, .ADDR, .LEN);
2
2     IF .STATUS
2         OR
2         .STATUS<0,16> EQLU SS$_DATAOVERUN OR .STATUS<0,16> EQLU SS$_ENDOFTAPE
2     THEN
2         RETURN 1;
2
2     IF .STATUS<0,16> NEQU SS$_ENDOFFILE
2     THEN
3         BEGIN
3         USER_STATUS[0] = .STATUS;
3         USER_STATUS[1]  = SS$_FCPREADERR;
```

```
;  191        0573  3          ERR_EXIT();
;  192        0574  2          END;
;  193        0575  2
;  194        0576  2      KERNEL_CALL(ADJTM, 1);
;  195        0577  2      RETURN 0;                                          ! tm encountered
;  196        0578  2
;  197        0579  1      END;                                               ! end of routine


                                                            .TITLE  LOGIO
                                                            .IDENT  \V04-000\

                                                            .EXTRN  CURRENT_UCB, IO_CHANNEL
                                                            .EXTRN  IO_STATUS, USER_STATUS
                                                            .EXTRN  GET_DEV_NAME, IO_DONE
                                                            .EXTRN  MOUNT_VOL, PRINT_OPR_MSG
                                                            .EXTRN  RESET_UNIT, SYS$QIOW
                                                            .EXTRN  SYS$CMKRNL

                                                            .PSECT  $CODE$,NOWRT,2

                                         0000 00000         .ENTRY  READ_BLOCK, Save nothing             ; 0516
                    7E          04    AC   7D 00002         MOVQ    ADDR, -(SP)                          ; 0560
                                      21   DD 00006         PUSHL   #33                                  ;
                                   0000V   30 00008         BSBW    ISSUE_IO                             ;
                    5E                0C   C0 0000B         ADDL2   #12, SP                              ;
                    0E                50   E8 0000E         BLBS    STATUS, 1$                           ; 0562
            0838    8F                50   B1 00011         CMPW    STATUS, #2104                        ; 0564
                                      07   13 00016         BEQL    1$                                   ;
            0878    8F                50   B1 00018         CMPW    STATUS, #2168                        ;
                                      04   12 0001D         BNEQ    2$                                   ;
                    50                01   D0 0001F  1$:    MOVL    #1, R0                               ; 0566
                                           04 00022         RET                                          ;
            0870    8F                50   B1 00023  2$:    CMPW    STATUS, #2160                        ; 0568
                                      0E   13 00028         BEQL    3$                                   ;
           0000G    CF                50   D0 0002A         MOVL    STATUS, USER_STATUS                  ; 0571
           0000G    CF        0888    8F   3C 0002F         MOVZWL  #2184, USER_STATUS+4                 ; 0572
                                      00   BF 00036         CHMU    #0                                   ; 0573
                                      01   DD 00038  3$:    PUSHL   #1                                   ; 0576
                                      01   DD 0003A         PUSHL   #1                                   ;
                                      5E   DD 0003C         PUSHL   SP                                   ;
                              0000V   CF   9F 0003E         PUSHAB  ADJTM                                ;
       00000000G    9F                04   FB 00042         CALLS   #4, @#SYS$CMKRNL                     ;
                                      50   D4 00049         CLRL    R0                                   ; 0577
                                           04 0004B         RET                                          ; 0579

; Routine Size:  76 bytes,    Routine Base:  $CODE$ + 0000
```

```
1 GLOBAL ROUTINE READ_BLOCK_REVERSE (ADDR, LEN) : COMMON_CALL =
1
1 !++
1 !
1 ! FUNCTIONAL DESCRIPTION:
1 !       This routine reads in reverse a logical record from magnetic tape.
1 !
1 ! CALLING SEQUENCE:
1 !       READ_BLOCK(ARG1,ARG2)
1 !
1 ! INPUT PARAMETERS:
1 !       ARG1 - address for data
1 !       ARG2 - length to read
1 !
1 ! IMPLICIT INPUTS:
1 !       IO_CHANNEL
1 !
1 ! OUTPUT PARAMETERS:
1 !       ARG1 - address for data
1 !
1 ! IMPLICIT OUTPUTS:
1 !       USER_STATUS, IO_STATUS
1 !
1 ! ROUTINE VALUE:
1 !       0 - tm encountered
1 !       1 - successful read
1 !
1 ! SIDE EFFECTS:
1 !       none
1 !
1 ! ERRORS:
1 !       Primary status is I/O error returned from driver
1 !       SS$_FCPREADERR - read failure
1 !
1 !--
1
2     BEGIN
2
2     EXTERNAL REGISTER
2         COMMON_REG;
2
2     LOCAL
2         STATUS;                                 ! IO status
2
2     STATUS = ISSUE_IO(IO$_READLBLK OR IO$M_REVERSE, .ADDR, .LEN);
2
2     IF .STATUS
2         OR
2         .STATUS<0,16> EQLU SS$_DATAOVERUN OR .STATUS<0,16> EQLU SS$_ENDOFTAPE
2     THEN
2         RETURN 1;
2
2     IF .STATUS<0,16> NEQU SS$_ENDOFFILE
2     THEN
3         BEGIN
3         USER_STATUS[0] = .STATUS;
3         USER_STATUS[1]  = SS$_FCPREADERR;
```

```
;   256         0637  3           ERR_EXIT();
;   257         0638  2           END;
;   258         0639  2
;   259         0640  2       KERNEL_CALL(ADJTM, 1);
;   260         0641  2       RETURN 0;                                  ! tm encountered
;   261         0642  2
;   262         0643  1       END;                                       ! end of routine



                                       0000 00000          .ENTRY  READ_BLOCK_REVERSE, Save nothing          ; 0580
                    7E        04    AC   7D 00002          MOVQ    ADDR, -(SP)                               ; 0624
                    7E        61    8F   9A 00006          MOVZBL  #97, -(SP)                                ;
                               0000V     30 0000A          BSBW    ISSUE_IO                                  ;
                    5E               0C  C0 0000D          ADDL2   #12, SP                                   ;
                    0E               50  E8 00010          BLBS    STATUS, 1$                                ; 0626
          0838      8F               50  B1 00013          CMPW    STATUS, #2104                             ; 0628
                                     07  13 00018          BEQL    1$                                        ;
          0878      8F               50  B1 0001A          CMPW    STATUS, #2168                             ;
                                     04  12 0001F          BNEQ    2$                                        ;
                    50               01  D0 00021  1$:     MOVL    #1, R0                                    ; 0630
                                         04 00024          RET                                               ;
          0870      8F               50  B1 00025  2$:     CMPW    STATUS, #2160                             ; 0632
                                     0E  13 0002A          BEQL    3$                                        ;
          0000G     CF               50  D0 0002C          MOVL    STATUS, USER_STATUS                       ; 0635
          0000G     CF        0888   8F  3C 00031          MOVZWL  #2184, USER_STATUS+4                      ; 0636
                                     00  BF 00038          CHMU    #0                                        ; 0637
                                     01  DD 0003A  3$:     PUSHL   #1                                        ; 0640
                                     01  DD 0003C          PUSHL   #1                                        ;
                                     5E  DD 0003E          PUSHL   SP                                        ;
                              0000V  CF  9F 00040          PUSHAB  ADJTM                                     ;
      00000000G     9F               04  FB 00044          CALLS   #4, @#SYS$CMKRNL                          ;
                                     50  D4 0004B          CLRL    R0                                        ; 0641
                                         04 0004D          RET                                               ; 0643

; Routine Size:  78 bytes,    Routine Base:  $CODE$ + 004C


;   263         0644  1
;   265         0645  1 GLOBAL ROUTINE WRITE_BLOCK (ADDR, LEN) : COMMON_CALL NOVALUE =
;   266         0646  1
;   267         0647  1 !++
;   268         0648  1 !
;   269         0649  1 ! FUNCTIONAL DESCRIPTION:
;   270         0650  1 !       This routine writes one logical block.
;   271         0651  1 !
;   272         0652  1 ! CALLING SEQUENCE:
;   273         0653  1 !       WRITE_BLOCK(ARG1,ARG2)
;   274         0654  1 !
;   275         0655  1 ! INPUT PARAMETERS:
;   276         0656  1 !       ARG1 - address of data block to write
;   277         0657  1 !       ARG2 - length of data block to write
;   278         0658  1 !
;   279         0659  1 ! IMPLICIT INPUTS:
;   280         0660  1 !       IO_CHANNEL
;   281         0661  1 !
;   282         0662  1 ! OUTPUT PARAMETERS:
;   283         0663  1 !       one block written
;   284         0664  1 !
;   285         0665  1 ! IMPLICIT OUTPUTS:
;   286         0666  1 !       IO_STATUS, USER_STATUS
;   287         0667  1 !
;   288         0668  1 ! ROUTINE VALUE:
;   289         0669  1 !       none
;   290         0670  1 !
;   291         0671  1 ! SIDE EFFECTS:
;   292         0672  1 !       SS$_FCPWRITERR - write failure
;   293         0673  1 !
;   294         0674  1 !--
;   295         0675  1
;   296         0676  2     BEGIN
;   297         0677  2
;   298         0678  2     EXTERNAL REGISTER
;   299         0679  2         COMMON_REG;
;   300         0680  2
;   301         0681  2     LOCAL
;   302         0682  2         STATUS;                                 ! IO status
;   303         0683  2
;   304         0684  2     STATUS = ISSUE_IO(IO$_WRITELBLK, .ADDR, .LEN);
;   305         0685  2
;   306         0686  2     IF NOT .STATUS AND  .STATUS<0,16> NEQ SS$_ENDOFTAPE
;   307         0687  2     THEN
;   308         0688  3         BEGIN
;   309         0689  3         USER_STATUS[0] = .STATUS;
;   310         0690  3         USER_STATUS[1]  = SS$_FCPWRITERR;
;   311         0691  3         ERR_EXIT();
;   312         0692  2         END;
;   313         0693  2
;   314         0694  1     END;                                        ! end of routine





                                         0000 00000           .ENTRY  WRITE_BLOCK, Save nothing                        ; 0645
                          7E       04    AC  7D 00002           MOVQ    ADDR, -(SP)                                      ; 0684
```

```
                                          20  DD 00006              PUSHL    #32                              ;
                                       0000V  30 00008              BSBW     ISSUE_IO                         ;
                      5E                  0C  C0 0000B              ADDL2    #12, SP                          ;
                      15                  50  E8 0000E              BLBS     STATUS, 1$                       ; 0686
              0878    8F                  50  B1 00011              CMPW     STATUS, #2168                    ;
                                          0E  13 00016              BEQL     1$                               ;
             0000G    CF                  50  D0 00018              MOVL     STATUS, USER_STATUS              ; 0689
             0000G    CF        08A0      8F  3C 0001D              MOVZWL   #2208, USER_STATUS+4             ; 0690
                                          00  BF 00024              CHMU     #0                               ; 0691
                                              04 00026  1$:         RET                                       ; 0694

; Routine Size:  39 bytes,    Routine Base:  $CODE$ + 009A


;  315          0695  1
```

```
1 GLOBAL ROUTINE SPACE (NUMBER) : COMMON_CALL =
1
1 !++
1 !
1 ! FUNCTIONAL DESCRIPTION:
1 !       This routine spaces a given number of records in either direction.
1 !
1 ! CALLING SEQUENCE:
1 !       SPACE(ARG1)
1 !
1 ! INPUT PARAMETERS:
1 !       ARG1 - number of records to space
1 !       ( positive means forward space, negative means backspace )
1 !
1 ! IMPLICIT INPUTS:
1 !       IO_CHANNEL
1 !
1 ! OUTPUT PARAMETERS:
1 !       none
1 !
1 ! IMPLICIT OUTPUTS:
1 !       IO_STATUS, USER_STATUS
1 !       Tape positioned accordingly
1 !
1 ! ROUTINE VALUE:
1 !       0 - end of file
1 !       1 - successful
1 !
1 ! SIDE EFFECTS:
1 !       SS$_FCPSPACERR - space failure
1 !
1 !--
1
2     BEGIN
2
2     EXTERNAL REGISTER
2         COMMON_REG;
2
2     LOCAL
2         TM,                                  ! number of tape marks
2         STATUS;                              ! io status
2
2     STATUS = ISSUE_IO(IO$_SKIPRECORD, .NUMBER, 0);
2
2     IF NOT .STATUS
2     THEN
3         BEGIN
3
3         IF .STATUS<0,16> EQL SS$_ENDOFFILE
3         THEN
4             BEGIN
4             TM = 1;                          ! encountered one spacing forward
4
4             IF .NUMBER LSS 0
4             THEN
4                 TM = -1;                     ! encountered one backspacing
4
```

```
;  374        0753  4              ! end of file indicates tape mark encountered
;  375        0754  4              !
;  376        0755  4              KERNEL_CALL(ADJTM, .TM);
;  377        0756  4              RETURN 0;
;  378        0757  4
;  379        0758  3              END;
;  380        0759  3
;  381        0760  3          IF .STATUS<0,16> EQL SS$_ENDOFTAPE
;  382        0761  3          THEN
;  383        0762  3              RETURN 1;
;  384        0763  3
;  385        0764  3          USER_STATUS[0] = .STATUS;
;  386        0765  3          USER_STATUS[1]  = SS$_FCPSPACERR;
;  387        0766  3          ERR_EXIT();
;  388        0767  2          END;
;  389        0768  2
;  390        0769  2      RETURN 1;
;  391        0770  2
;  392        0771  1      END;                                        ! end of routine



                                      0004 00000        .ENTRY  SPACE, Save R2                          ; 0696
                                   7E  D4 00002         CLRL    -(SP)                                   ; 0738
                          04       AC  DD 00004         PUSHL   NUMBER                                  ;
                                   26  DD 00007         PUSHL   #38                                     ;
                                0000V  30 00009         BSBW    ISSUE_IO                                ;
                    5E             0C  C0 0000C         ADDL2   #12, SP                                 ;
                    52             50  D0 0000F         MOVL    R0, STATUS                              ;
                    3A             52  E8 00012         BLBS    STATUS, 3$                              ; 0740
          0870      8F             52  B1 00015         CMPW    STATUS, #2160                           ; 0744
                                   1E  12 0001A         BNEQ    2$                                      ;
                    50             01  D0 0001C         MOVL    #1, TM                                  ; 0747
                          04       AC  D5 0001F         TSTL    NUMBER                                  ; 0749
                                   03  18 00022         BGEQ    1$                                      ;
                    50             01  CE 00024         MNEGL   #1, TM                                  ; 0751
                                   50  DD 00027 1$:     PUSHL   TM                                      ; 0755
                                   01  DD 00029         PUSHL   #1                                      ;
                                   5E  DD 0002B         PUSHL   SP                                      ;
                        0000V      CF  9F 0002D         PUSHAB  ADJTM                                   ;
   00000000G        9F             04  FB 00031         CALLS   #4, @#SYS$CMKRNL                        ;
                                   19  11 00038         BRB     4$                                      ; 0756
          0878      8F             52  B1 0003A 2$:     CMPW    STATUS, #2168                           ; 0760
                                   0E  13 0003F         BEQL    3$                                      ;
         0000G      CF             52  D0 00041         MOVL    STATUS, USER_STATUS                     ; 0764
         0000G      CF      0898   8F  3C 00046         MOVZWL  #2200, USER_STATUS+4                    ; 0765
                                   00  BF 0004D         CHMU    #0                                      ; 0766
                    50             01  D0 0004F 3$:     MOVL    #1, R0                                  ; 0769
                                       04 00052         RET                                             ;
                                   50  D4 00053 4$:     CLRL    R0                                      ; 0771
                                       04 00055         RET                                             ;

; Routine Size:  86 bytes,    Routine Base:  $CODE$ + 00C1
;  393          0772  1
```

```
; Ro

;  7
;  395          0773 1 GLOBAL ROUTINE WRITE_TM : NOVALUE L$WRITE_TM =
;  396          0774 1
;  397          0775 1 !++
;  398          0776 1 !
;  399          0777 1 ! FUNCTIONAL DESCRIPTION:
;  400          0778 1 !       This routine writes one tape mark.
;  401          0779 1 !
;  402          0780 1 ! CALLING SEQUENCE:
;  403          0781 1 !       WRITE_TM()
;  404          0782 1 !
;  405          0783 1 ! INPUT PARAMETERS:
;  406          0784 1 !       none
;  407          0785 1 !
;  408          0786 1 ! IMPLICIT INPUTS:
;  409          0787 1 !       IO_CHANNEL
;  410          0788 1 !
;  411          0789 1 ! OUTPUT PARAMETERS:
;  412          0790 1 !       none
;  413          0791 1 !
;  414          0792 1 ! IMPLICIT OUTPUTS:
;  415          0793 1 !       IO_STATUS, USER_STATUS
;  416          0794 1 !       Tape mark written, tm count incremented.
;  417          0795 1 !
;  418          0796 1 ! ROUTINE VALUE:
;  419          0797 1 !       none
;  420          0798 1 !
;  421          0799 1 ! SIDE EFFECTS:
;  422          0800 1 !       SS$_FCPWRITERR - write failure
;  423          0801 1 !
;  424          0802 1 !--
;  425          0803 1
;  426          0804 2     BEGIN
;  427          0805 2
;  428          0806 2     EXTERNAL REGISTER
;  429          0807 2         COMMON_REG;
;  430          0808 2
;  431          0809 2     LOCAL
;  432          0810 2         STATUS;                                 ! io status
;  433          0811 2
;  434          0812 2     STATUS = ISSUE_IO(IO$_WRITEOF, 0, 0);
;  435          0813 2
;  436          0814 2     IF NOT .STATUS AND .STATUS<0,16> NEQ SS$_ENDOFTAPE
;  437          0815 2     THEN
;  438          0816 3         BEGIN
;  439          0817 3         USER_STATUS[0] = .STATUS;
;  440          0818 3         USER_STATUS[1] = SS$_FCPWRITERR;
;  441          0819 3         ERR_EXIT();
;  442          0820 2         END;
;  443          0821 2
;  444          0822 2     KERNEL_CALL(ADJTM, 1);
;  445          0823 1     END;                                        ! end of routine


                                   7E  7C 00000 WRITE_TM::
```

```
                                                         CLRQ    -(SP)                        ; 0812
                                   28  DD 00002          PUSHL   #40                          ;
                                0000V  30 00004          BSBW    ISSUE_IO                     ;
                   5E              0C  C0 00007          ADDL2   #12, SP                      ;
                   15              50  E8 0000A          BLBS    STATUS, 1$                   ; 0814
           0878    8F              50  B1 0000D          CMPW    STATUS, #2168                ;
                                   0E  13 00012          BEQL    1$                           ;
          C000G    CF              50  D0 00014          MOVL    STATUS, USER_STATUS          ; 0817
          0000G    CF    08A0      8F  3C 00019          MOVZWL  #2208, USER_STATUS+4         ; 0818
                                   00  BF 00020          CHMU    #0                           ; 0819
                                   01  DD 00022 1$:      PUSHL   #1                           ; 0822
                                   01  DD 00024          PUSHL   #1                           ;
                                   5E  DD 00026          PUSHL   SP                           ;
                         0000V     CF  9F 00028          PUSHAB  ADJTM                        ;
      00000000G    9F              04  FB 0002C          CALLS   #4, @#SYS$CMKRNL             ;
                                   05  00033             RSB                                  ; 0823

; Routine Size:  52 bytes,    Routine Base:  $CODE$ + 0117

;  446        0824  1
;  448          0825  1 GLOBAL ROUTINE ADJTM (NUMBER) : COMMON_CALL NOVALUE =
;  449          0826  1
;  450          0827  1 !++
;  451          0828  1 !
;  452          0829  1 ! FUNCTIONAL DESCRIPTION:
;  453          0830  1 !       This routine adjusts the tm count by the given number.
;  454          0831  1 !
;  455          0832  1 ! CALLING SEQUENCE:
;  456          0833  1 !       ADJTM(ARG1), called in kernel mode
;  457          0834  1 !
;  458          0835  1 ! INPUT PARAMETERS:
;  459          0836  1 !       ARG1 - signed number to adjust count by
;  460          0837  1 !
;  461          0838  1 ! IMPLICIT INPUTS:
;  462          0839  1 !       CURRENT_VCB[VCB$B_TM]
;  463          0840  1 !
;  464          0841  1 ! OUTPUT PARAMETERS:
;  465          0842  1 !       none
;  466          0843  1 !
;  467          0844  1 ! IMPLICIT OUTPUTS:
;  468          0845  1 !       CURRENT_VCB[VCB$B_TM]
;  469          0846  1 !       CURRENT_VCB[VCB$L_ST_RECORD]
;  470          0847  1 !
;  471          0848  1 ! ROUTINE VALUE:
;  472          0849  1 !       none
;  473          0850  1 !
;  474          0851  1 ! SIDE EFFECTS:
;  475          0852  1 !       none
;  476          0853  1 !
;  477          0854  1 !--
;  478          0855  1
;  479          0856  2     BEGIN
;  480          0857  2
;  481          0858  2     EXTERNAL REGISTER
;  482          0859  2         COMMON_REG;
;  483          0860  2
;  484          0861  2     LOCAL
;  485          0862  2         TM;                                         ! number of tm's
;  486          0863  2
;  487          0864  2     TM = .CURRENT_VCB[VCB$B_TM];
;  488          0865  2     TM = .TM + .NUMBER;
;  489          0866  2
;  490          0867  2     ! Now adjust number so it is a number between 0 and 2
;  491          0868  2     !
;  492          0869  2
;  493          0870  2     IF .TM GEQ 3
;  494          0871  2     THEN
;  495          0872  2         TM = .TM - 3;
;  496          0873  2
;  497          0874  2     IF .TM LSS 0
;  498          0875  2     THEN
;  499          0876  2         TM = .TM + 3;
;  500          0877  2
;  501          0878  2     CURRENT_VCB[VCB$B_TM] = .TM;
;  502          0879  2     CURRENT_VCB[VCB$L_ST_RECORD] = .CURRENT_UCB[UCB$L_RECORD];
;  503          0880  1     END;                                            ! end of routine
```

```
                                      0000 00000         .ENTRY   ADJTM, Save nothing                        ; 0825
                 50        2E     AB    9A 00002         MOVZBL   46(CURRENT_VCB), TM                        ; 0864
                 50        04     AC    C0 00006         ADDL2    NUMBER, TM                                 ; 0865
                 03               50    D1 0000A         CMPL     TM, #3                                     ; 0870
                                  03    19 0000D         BLSS     1$                                         ;
                 50               03    C2 0000F         SUBL2    #3, TM                                     ; 0872
                                  50    D5 00012 1$:     TSTL     TM                                         ; 0874
                                  03    18 00014         BGEQ     2$                                         ;
                 50               03    C0 00016         ADDL2    #3, TM                                     ; 0876
           2E    AB               50    90 00019 2$:     MOVB     TM, 46(CURRENT_VCB)                        ; 0878
                 50     0000G     CF    D0 0001D         MOVL     CURRENT_UCB, R0                            ; 0879
           30    AB     00B0      C0    D0 00022         MOVL     176(R0), 48(CURRENT_VCB)                   ;
                                        04 00028         RET                                                 ; 0880

; Routine Size:  41 bytes,    Routine Base:  $CODE$ + 014B


;  504          0881  1
;   506         0882  1 GLOBAL ROUTINE ISSUE_IO (FUNCTION, P1, P2) : L$ISSUE_IO =
;   507         0883  1
;   508         0884  1 !++
;   509         0885  1 !
;   510         0886  1 ! FUNCTIONAL DESCRIPTION:
;   511         0887  1 !       This routine issues the I/O and if the device is offline or
;   512         0888  1 !       the volume is invalid it repositions it.
;   513         0889  1 !
;   514         0890  1 !
;   515         0891  1 ! CALLING SEQUENCE:
;   516         0892  1 !       ISSUE_IO(FUNCTION,P1,P2)
;   517         0893  1 !
;   518         0894  1 ! INPUT PARAMETERS:
;   519         0895  1 !       ARG1 - function code
;   520         0896  1 !       ARG2 - P1 parameter
;   521         0897  1 !       ARG3 - P2 parameter
;   522         0898  1 !
;   523         0899  1 ! IMPLICIT INPUTS:
;   524         0900  1 !       none
;   525         0901  1 !
;   526         0902  1 ! OUTPUT PARAMETERS:
;   527         0903  1 !       none
;   528         0904  1 !
;   529         0905  1 ! IMPLICIT OUTPUTS:
;   530         0906  1 !       none
;   531         0907  1 !
;   532         0908  1 ! ROUTINE VALUE:
;   533         0909  1 !       I/O status
;   534         0910  1 !
;   535         0911  1 ! SIDE EFFECTS:
;   536         0912  1 !       none
;   537         0913  1 !
;   538         0914  1 !--
;   539         0915  1
;   540         0916  2     BEGIN
;   541         0917  2
;   542         0918  2     EXTERNAL REGISTER
;   543         0919  2         COMMON_REG;
;   544         0920  2
;   545         0921  2     LOCAL
;   546         0922  2         CUR_RECORD;
;   547         0923  2
;   548         0924  2     ! save current position
;   549         0925  2     !
;   550         0926  2     CUR_RECORD = .CURRENT_UCB[UCB$L_RECORD];
;   551         0927  2
;   552         0928  2     WHILE 1
;   553         0929  2     DO
;   554         0930  3         BEGIN
;   555         0931  3
;   556         0932  4         BEGIN
;   557         0933  4         LOCAL
;   558         0934  4             STATUS;
;   559         0935  4
;   560       P 0936  4         STATUS = $QIOW(EFN = EFN, CHAN = .IO_CHANNEL,
;   561       P 0937  4             FUNC = .FUNCTION  OR IO$M_CLSEREXCP,
;   562         0938  4             IOSB = IO_STATUS, P1 = .P1, P2 = .P2);
;  563          0939  4
;  564          0940  4             IF NOT .STATUS
;  565          0941  4             THEN
;  566          0942  4                 IO_STATUS = .STATUS;
;  567          0943  3             END;
;  568          0944  3
;  569          0945  3             IF .IO_STATUS<0,16> NEQ SS$_MEDOFL AND .IO_STATUS<0,16> NEQ SS$_VOLINV
;  570          0946  3             THEN
;  571          0947  3                 RETURN .IO_STATUS;
;  572          0948  3
;  573          0949  3             REPOSITION(.CUR_RECORD);
;  574          0950  2             END;
;  575          0951  2
;  576          0952  2         RETURN 1;
;  577          0953  1         END;                                          ! end of routine ISSUE_IO



                             50      0000G  CF   D0 00000 ISSUE_IO::
                                                                  MOVL     CURRENT_UCB, R0                                   ; 0926
                                      00B0  C0   DD 00005         PUSHL    176(R0)                                           ;
                   7E        08   AE 00000200  8F   C9 00009         BISL3    #512, FUNCTION, -(SP)                             ; 0938
                                              7E   7C 00012 1$:     CLRQ     -(SP)                                             ;
                                              7E   7C 00014         CLRQ     -(SP)                                             ;
                                          24  AE   DD 00016         PUSHL    P2                                                ;
                                          24  AE   DD 00019         PUSHL    P1                                                ;
                                              7E   7C 0001C         CLRQ     -(SP)                                             ;
                                       0000G  CF   9F 0001E         PUSHAB   IO_STATUS                                         ;
                                          24  AE   DD 00022         PUSHL    36(SP)                                            ;
                                       0000G  CF   DD 00025         PUSHL    IO_CHANNEL                                        ;
                                              01   DD 00029         PUSHL    #1                                                ;
                      00000000G  00          0C   FB 0002B         CALLS    #12, SYS$QIOW                                     ;
                                 05          50   E8 00032         BLBS     STATUS, 2$                                        ; 0940
                          0000G  CF          50   D0 00035         MOVL     STATUS, IO_STATUS                                 ; 0942
                          01A4   8F   0000G  CF   B1 0003A 2$:     CMPW     IO_STATUS, #420                                   ; 0945
                                              10   13 00041         BEQL     3$                                                ;
                          0254   8F   0000G  CF   B1 00043         CMPW     IO_STATUS, #596                                   ;
                                              07   13 0004A         BEQL     3$                                                ;
                             50      0000G  CF   D0 0004C         MOVL     IO_STATUS, R0                                     ; 0947
                                              0B   11 00051         BRB      4$                                                ;
                                          04  AE   DD 00053 3$:     PUSHL    CUR_RECORD                                        ; 0949
                                           0000V   30 00056         BSBW     REPOSITION                                        ;
                             5E               04   C0 00059         ADDL2    #4, SP                                            ;
                                              B4   11 0005C         BRB      1$                                                ; 0928
                             5E               08   C0 0005E 4$:     ADDL2    #8, SP                                            ; 0953
                                                   05 00061         RSB                                                        ;

; Routine Size:  98 bytes,    Routine Base:  $CODE$ + 0174


;  578          0954  1
;    1 GLOBAL ROUTINE SPACE_TM (NUMBER) : COMMON_CALL NOVALUE =
;    1
;    1 !++
;    1 !
;    1 ! FUNCTIONAL DESCRIPTION:
;    1 !       This routine spaces a given number of tm's in either direction.
;    1 !
;    1 ! CALLING SEQUENCE:
;    1 !       SPACE_TM(NUMBER)
;    1 !
;    1 ! INPUT PARAMETERS:
;    1 !       ARG1 - number of tm's to space
;    1 !               (if negative, space backward. if positive, space forward.)
;    1 !
;    1 ! IMPLICIT INPUTS:
;    1 !       IO_CHANNEL
;    1 !
;    1 ! OUTPUT PARAMETERS:
;    1 !       none
;    1 !
;    1 ! IMPLICIT OUTPUTS:
;    1 !       TM count incremented to reflect tape postioned beyond the tm specified
;    1 !       IO_STATUS, USER_STATUS
;    1 !
;    1 ! ROUTINE VALUE:
;    1 !       none
;    1 !
;    1 ! SIDE EFFECTS:
;    1 !       SS$_FCPSPACERR - space failure
;    1 !
;    1 !--
;    1
;    2     BEGIN
;    2
;    2     EXTERNAL REGISTER
;    2         COMMON_REG;
;    2
;    2     EXTERNAL ROUTINE
;    2         BLOCK,
;    2         SYS$QIO : ADDRESSING_MODE (ABSOLUTE);
;    2
;    2     LOCAL
;    2         CUR_RECORD,                                 ! current position of tape
;    2         STATUS;                                     ! io status
;    2
;    2     CUR_RECORD = .CURRENT_UCB[UCB$L_RECORD];
;    2
;    2     WHILE 1
;    2     DO
;    3         BEGIN
;    3         BBLOCK[.CURRENT_VCB[VCB$L_VPFL], VVP$L_NO_TM] = .NUMBER;
;  P 3         $QIO( CHAN = .IO_CHANNEL,
;  P 3                 FUNC = IO$_SKIPFILE OR IO$M_CLSEREXCP,
;  P 3                 IOSB = BBLOCK[.CURRENT_VCB[VCB$L_VPFL], VVP$L_STATUS],
;  P 3                 ASTADR = UNBLOCK_SPACE,
;  P 3                 ASTPRM = .CURRENT_VCB,
;    3                 P1 = .NUMBER );
;  637      1012 3          ! Block the process to wait for function to be completed
;  638      1013 3          !
;  639      1014 3          BLOCK($FIELDMASK(VCB$V_WAIREWIND));
;  640      1015 3
;  641      1016 3          STATUS = .BBLOCK[.CURRENT_VCB[VCB$L_VPFL], VVP$L_STATUS];
;  642      1017 3
;  643      1018 3          IF .STATUS<0,16> NEQ SS$_MEDOFL AND .STATUS<0,16> NEQ SS$_VOLINV
;  644      1019 3          THEN
;  645      1020 3              EXITLOOP;
;  646      1021 3
;  647      1022 3          REPOSITION(.CUR_RECORD);
;  648      1023 3          END;
;  649      1024 2
;  650      1025 2      IF NOT .STATUS AND .STATUS<0,16> NEQ SS$_ENDOFTAPE
;  651      1026 2      THEN
;  652      1027 2          BEGIN
;  653      1028 3          USER_STATUS[0] = .STATUS;
;  654      1029 3          USER_STATUS[1] = SS$_FCPSPACERR;
;  655      1030 3          ERR_EXIT();
;  656      1031 3          END;
;  657      1032 2
;  658      1033 2
;  659      1034 2      KERNEL_CALL(ADJTM, .NUMBER);
;  660      1035 1      END;                                                  ! end of routine


                                                        .EXTRN  BLOCK, SYS$QIO

                              000C 00000            .ENTRY  SPACE_TM, Save R2,R3                       ; 0955
              50   0000G  CF   D0 00002            MOVL    CURRENT_UCB, R0                             ; 1000
              53   00B0   C0   D0 00007            MOVL    176(R0), CUR_RECORD
              50     3C   AB   D0 0000C 1$:        MOVL    60(CURRENT_VCB), R0                         ; 1005
      01C4    C0     04   AC   D0 00010            MOVL    NUMBER, 452(R0)
                          7E   7C 00016            CLRQ    -(SP)                                       ; 1011
                          7E   7C 00018            CLRQ    -(SP)                                       ;
                          7E   D4 0001A            CLRL    -(SP)                                       ;
                     04   AC   DD 0001C            PUSHL   NUMBER                                      ;
                          5B   DD 0001F            PUSHL   CURRENT_VCB                                 ;
                   0000V  CF   9F 00021            PUSHAB  UNBLOCK_SPACE                               ;
                   019C   C0   9F 00025            PUSHAB  412(R0)                                     ;
              7E   0225   8F   3C 00029            MOVZWL  #549, -(SP)                                 ;
                   0000G  CF   DD 0002E            PUSHL   IO_CHANNEL                                  ;
                          7E   D4 00032            CLRL    -(SP)                                       ;
 00000000G    00          0C   FB 00034            CALLS   #12, SYS$QIO                                ;
                          08   DD 0003B            PUSHL   #8                                          ; 1015
      0000G   CF          01   FB 0003D            CALLS   #1, BLOCK                                   ;
              50     3C   AB   D0 00042            MOVL    60(CURRENT_VCB), R0                         ; 1017
              52   019C   C0   D0 00046            MOVL    412(R0), STATUS                             ;
      01A4    8F          52   B1 0004B            CMPW    STATUS, #420                                ; 1019
                          07   13 00050            BEQL    2$                                          ;
      0254    8F          52   B1 00052            CMPW    STATUS, #596                                ;
                          0A   12 00057            BNEQ    3$                                          ;
                          53   DD 00059 2$:        PUSHL   CUR_RECORD                                  ; 1023
                       0000V   30 0005B            BSBW    REPOSITION                                  ;
              5E          04   C0 0005E            ADDL2   #4, SP                                      ;
                          A9   11 00061            BRB     1$                                          ; 1002
```

```
                        15            52  E8 00063 3$:     BLBS    STATUS, 4$                          ; 1026
               0878     8F            52  B1 00066         CMPW    STATUS, #2168                       ;
                                      0E  13 0006B         BEQL    4$                                  ;
               0000G    CF            52  D0 0006D         MOVL    STATUS, USER_STATUS                 ; 1029
               0000G    CF     0898   8F  3C 00072         MOVZWL  #2200, USER_STATUS+4                ; 1030
                                      00  BF 00079         CHMU    #0                                  ; 1031
                                 04   AC  DD 0007B 4$:     PUSHL   NUMBER                              ; 1034
                                      01  DD 0007E         PUSHL   #1                                  ;
                                      5E  DD 00080         PUSHL   SP                                  ;
                               FEEF   CF  9F 00082         PUSHAB  ADJTM                               ;
          00000000G     9F            04  FB 00086         CALLS   #4, @#SYS$CMKRNL                    ;
                                          04 0008D         RET                                         ; 1035

; Routine Size:  142 bytes,    Routine Base:  $CODE$ + 01D6


;  661          1036  1
```

```
1 GLOBAL ROUTINE REPOSITION (NO_RECORD) : L$REPOSITION NOVALUE =
1
1 !++
1 !
1 ! FUNCTIONAL DESCRIPTION:
1 !       This routine mounts the device that is offline and repositions
1 !       to the current position.
1 !
1 ! CALLING SEQUENCE:
1 !       REPOSITION(ARG1)
1 !
1 ! INPUT PARAMETERS:
1 !       ARG1 - number of record to position to
1 !
1 ! IMPLICIT INPUTS:
1 !       CURRENT_UCB - address of current unit control block
1 !       CURRENT_VCB - address of current volume control block
1 !
1 ! OUTPUT PARAMETERS:
1 !       none
1 !
1 ! IMPLICIT OUTPUTS:
1 !       none
1 !
1 ! ROUTINE VALUE:
1 !       none
1 !
1 ! SIDE EFFECTS:
1 !       none
1 !
1 ! USER ERRORS:
1 !       none
1 !
1 !--
1
2     BEGIN
2
2     EXTERNAL REGISTER
2         COMMON_REG;
2
2     LABEL
2         OFFLINE;
2
2     LOCAL
2         CVT_DEVNAM        : VECTOR [MAX_DEVNAM_LENGTH,BYTE], ! Converted dev name
2         CVT_DEVNAM_LENGTH : BYTE,                  ! and length of dev name
2         SAV_TM,
2         SAV_ST_REC,
2         MVL_ENTRY,                                 ! address of cur volume MVL entry
2         VOL;                                       ! current volume
2
2     VOL = .CURRENT_VCB[VCB$B_CUR_RVN];
2     SAV_TM = .CURRENT_VCB[VCB$B_TM];
2     SAV_ST_REC = .CURRENT_VCB[VCB$L_ST_RECORD];
2
2     ! This next call will use the UCB address to get the device's name and
2     ! will fill in the fields with that name and the length of the name.
```

```
;     2
;     2      GET_DEV_NAME(CVT_DEVNAM_LENGTH,CVT_DEVNAM);
;     2
;     2      ! Set device not mounted since rewind does not currently recognize device
;     2      ! offline
;     2      !
;     2      MVL_ENTRY = .CURRENT_VCB[VCB$L_MVL] + MVL$K_FIXLEN + ((.VOL -  1)*MVL$K_LENGTH);
;     2
;     2  OFFLINE :
;     3      BEGIN
;     3
;     3      WHILE 1
;     3      DO
;     4          BEGIN
;     4
;     4          ! Send message to operator informing that the device is offline
;     4          !
;     4          PRINT_OPR_MSG(MOUN$_OFFLINE, 0, .CVT_DEVNAM_LENGTH,CVT_DEVNAM);
;     4          KERNEL_CALL(RESET_UNIT);
;     4
;     4          ! Mount volume again
;     4          MOUNT_VOL(.VOL,
;     4              $FIELDMASK(MOU$V_REWIND) + $FIELDMASK(MOU$V_LBLCHECK) +
;     4                  $FIELDMASK(MOU$V_MOUNTERR));
;     4
;     4          WHILE 1
;     4          DO
;     5              BEGIN
;     6              (
;     6
;     6              LOCAL
;     6                  STATUS;
;     6
;     6              ! Space the number of blocks left to space
;     6              !
;   P 6              STATUS = $QIOW(EFN = EFN,
;   P 6                        CHAN = .IO_CHANNEL,
;   P 6                        FUNC = IO$_SKIPRECORD OR IO$M_CLSEREXCP,
;   P 6                        IOSB = IO_STATUS,
;     6                        P1 = .NO_RECORD - .CURRENT_UCB[UCB$L_RECORD]);
;     6
;     6              IF NOT .STATUS
;     6              THEN
;     5                  .IO_STATUS = .STATUS);            ! directive status
;     5
;     5              IF .NO_RECORD EQL .CURRENT_UCB[UCB$L_RECORD]
;     5              THEN
;     5                  LEAVE OFFLINE;                    ! repositioning complete
;     5
;     5              IF .IO_STATUS<0,16> EQL SS$_MEDOFL OR .IO_STATUS<0,16> EQL SS$_VOLINV
;     5                  OR
;     5                  .CURRENT_UCB[UCB$L_RECORD] GEQ .NO_RECORD
;     5              THEN
;     5                  EXITLOOP                          ! start again
;     5              ELSE
;     5
;     5                  IF .IO_STATUS<0,16> NEQ SS$_ENDOFFILE
;  777       1151  5                  THEN
;  778       1152  6                      BEGIN
;  779       1153  6                      USER_STATUS[0] = .IO_STATUS;
;  780       1154  6                      USER_STATUS[1] = SS$_FCPREPSTN;
;  781       1155  6                      ERR_EXIT();
;  782       1156  5                      END;
;  783       1157  5
;  784       1158  4              END;
;  785       1159  4
;  786       1160  3          END;
;  787       1161  3
;  788       1162  2      END;                                    ! end offline
;  789       1163  2      KERNEL_CALL(RESTORE_POS, .SAV_TM, .SAV_ST_REC);
;  790       1164  1      END;                                    ! end of routine




                     5E             14  C2 00000 REPOSITION::
                                                         SUBL2   #20, SP                                    ; 1037
                     7E       2F    AB  9A 00003         MOVZBL  47(CURRENT_VCB), VOL                       ; 1088
                     7E       2E    AB  9A 00007         MOVZBL  46(CURRENT_VCB), SAV_TM                    ; 1089
                              30    AB  DD 0000B         PUSHL   48(CURRENT_VCB)                            ; 1090
                              10    AE  9F 0000E         PUSHAB  CVT_DEVNAM                                 ; 1095
                              10    AE  9F 00011         PUSHAB  CVT_DEVNAM_LENGTH                          ;
             0000G   CF             02  FB 00014         CALLS   #2, GET_DEV_NAME                           ;
                     51       08    AE  D0 00019         MOVL    VOL, R1                                    ; 1100
                     50       34  BB41  7E 0001D         MOVAQ   @52(CURRENT_VCB)[R1], MVL_ENTRY            ;
                     50             1C  C0 00022         ADDL2   #28, MVL_ENTRY                             ;
                              10    AE  9F 00025 1$:     PUSHAB  CVT_DEVNAM                                 ; 1111
                     7E       10    AE  9A 00028         MOVZBL  CVT_DEVNAM_LENGTH, -(SP)                   ;
                                    7E  D4 0002C         CLRL    -(SP)                                      ;
                 0072811C   8F      DD 0002E             PUSHL   #7504156                                   ;
                              0000G 30 00034             BSBW    PRINT_OPR_MSG                              ;
                     5E             0C  C0 00037         ADDL2   #12, SP                                    ;
                                    6E  D4 0003A         CLRL    (SP)                                       ; 1112
                                    5E  DD 0003C         PUSHL   SP                                         ;
                           0000G    CF  9F 0003E         PUSHAB  RESET_UNIT                                 ;
         00000000G   9F             03  FB 00042         CALLS   #3, @#SYS$CMKRNL                           ;
                                    0B  DD 00049         PUSHL   #11                                        ; 1116
                              0C    AE  DD 0004B         PUSHL   VOL                                        ; 1115
             0000G   CF             02  FB 0004E         CALLS   #2, MOUNT_VOL                              ;
                                    7E  7C 00053 2$:     CLRQ    -(SP)                                      ; 1133
                                    7E  7C 00055         CLRQ    -(SP)                                      ;
                                    7E  D4 00057         CLRL    -(SP)                                      ;
                     50    0000G    CF  D0 00059         MOVL    CURRENT_UCB, R0                            ;
 7E          38      AE    00B0     C0  C3 0005E         SUBL3   176(R0), NO_RECORD, -(SP)                  ;
                                    7E  7C 00065         CLRQ    -(SP)                                      ;
                           0000G    CF  9F 00067         PUSHAB  IO_STATUS                                  ;
                     7E    0226     8F  3C 0006B         MOVZWL  #550, -(SP)                                ;
                           0000G    CF  DD 00070         PUSHL   IO_CHANNEL                                 ;
                                    01  DD 00074         PUSHL   #1                                         ;
         00000000G   00             0C  FB 00076         CALLS   #12, SYS$QIOW                              ;
                     05             50  E8 0007D         BLBS    STATUS, 3$                                 ; 1135
             0000G   DF             50  D0 00080         MOVL    STATUS, @IO_STATUS                         ; 1137
                     50    0000G    CF  D0 00085 3$:     MOVL    CURRENT_UCB, R0                            ; 1139
```

```
        00B0    C0        24    AE  D1 0008A          CMPL    NO_RECORD, 176(R0)                    ;
                                37  13 00090          BEQL    5$                                    ;
                51     0000G    CF  3C 00092          MOVZWL  IO_STATUS, R1                         ; 1143
        01A4    8F              51  B1 00097          CMPW    R1, #420                              ;
                                87  13 0009C          BEQL    1$                                    ;
        0254    8F              51  B1 0009E          CMPW    R1, #596                              ;
                                80  13 000A3          BEQL    1$                                    ;
          24    AE      00B0    C0  D1 000A5          CMPL    176(R0), NO_RECORD                    ; 1145
                                03  19 000AB          BLSS    4$                                    ;
                              FF75  31 000AD          BRW     1$                                    ;
        0870    8F              51  B1 000B0  4$:     CMPW    R1, #2160                             ; 1150
                                9C  13 000B5          BEQL    2$                                    ;
       0000G    CF     0000G    CF  D0 000B7          MOVL    IO_STATUS, USER_STATUS                ; 1153
       0000G    CF      0988    8F  3C 000BE          MOVZWL  #2440, USER_STATUS+4                  ; 1154
                                00  BF 000C5          CHMU    #0                                    ; 1155
                                8A  11 000C7          BRB     2$                                    ; 1119
                                6E  DD 000C9  5$:     PUSHL   SAV_ST_REC                            ; 1163
                          08    AE  DD 000CB          PUSHL   SAV_TM                                ;
                                02  DD 000CE          PUSHL   #2                                    ;
                                5E  DD 000D0          PUSHL   SP                                    ;
                       0000V    CF  9F 000D2          PUSHAB  RESTORE_POS                           ;
   00000000G    9F              05  FB 000D6          CALLS   #5, @#SYS$CMKRNL                      ;
                5E              20  C0 000DD          ADDL2   #32, SP                               ; 1164
                                    05 000E0          RSB                                           ;

; Routine Size:  225 bytes,    Routine Base:  $CODE$ + 0264


;  791          1165  1
;  793         1166  1 GLOBAL ROUTINE RESTORE_POS (TM, REC) : COMMON_CALL NOVALUE =
;  794         1167  1
;  795         1168  1 !++
;  796         1169  1 !
;  797         1170  1 ! FUNCTIONAL DESCRIPTION:
;  798         1171  1 !       This routine restores the tape position information destroyed by
;  799         1172  1 !       ASSUME_MOUNTED.
;  800         1173  1 !
;  801         1174  1 ! CALLING SEQUENCE:
;  802         1175  1 !       RESTORE_POS(ARG1,ARG2), in kernel mode
;  803         1176  1 !
;  804         1177  1 ! INPUT PARAMETERS:
;  805         1178  1 !       ARG1 - number or tape marks
;  806         1179  1 !       ARG2 - number of blocks into tape since last tape mark
;  807         1180  1 !
;  808         1181  1 ! IMPLICIT INPUTS:
;  809         1182  1 !       address of current VCB
;  810         1183  1 !
;  811         1184  1 ! OUTPUT PARAMETERS:
;  812         1185  1 !       none
;  813         1186  1 !
;  814         1187  1 ! IMPLICIT OUTPUTS:
;  815         1188  1 !       CURRENT_VCB[VCB$B_TM] and CURRENT_VCB[VCB$L_ST_RECORD] updated
;  816         1189  1 !
;  817         1190  1 ! ROUTINE VALUE:
;  818         1191  1 !       none
;  819         1192  1 !
;  820         1193  1 ! SIDE EFFECTS:
;  821         1194  1 !       none
;  822         1195  1 !
;  823         1196  1 !--
;  824         1197  1
;  825         1198  2     BEGIN
;  826         1199  2
;  827         1200  2     EXTERNAL REGISTER
;  828         1201  2         COMMON_REG;
;  829         1202  2
;  830         1203  2     CURRENT_VCB[VCB$B_TM] = .TM;
;  831         1204  2     CURRENT_VCB[VCB$L_ST_RECORD] = .REC;
;  832         1205  1     END;



                                          0000 00000        .ENTRY  RESTORE_POS, Save nothing                   ; 1166
              2E   AB       04   AC  90 00002        MOVB    TM, 46(CURRENT_VCB)                          ; 1203
              30   AB       08   AC  D0 00007        MOVL    REC, 48(CURRENT_VCB)                         ; 1204
                                          04 0000C        RET                                                 ; 1205

; Routine Size:  13 bytes,    Routine Base:  $CODE$ + 0345
```

```
1 ROUTINE UNBLOCK_SPACE (VCB) : COMMON_CALL NOVALUE =
1
1 !++
1 !
1 ! FUNCTIONAL DESCRIPTION:
1 !       This routine unblocks after a SPACE_TM has been done.  If I/O
1 !       is canceled, the tape position is updated.
1 !
1 ! calling sequence:
1 !       UNBLOCK_SPACE()
1 !
1 ! INPUT PARAMETERS:
1 !       ARG1     - address of volume control block
1 !
1 ! IMPLICIT INPUTS:
1 !       VVP$L_NO_TM       - number of tape marks positioned.
1 !                          ( If neg then backwards,  else forwards.)
1 !       Saved stack and impure area
1 !
1 ! OUTPUT PARAMETERS:
1 !       none
1 !
1 ! IMPLICIT OUTPUTS:
1 !       CURRENT_VCB[VCB$L_ST_RECORD]
1 !
1 ! ROUTINE VALUE:
1 !       none
1 !
1 ! SIDE EFFECT:
1 !       Never returns to PC where AST's were enabled.
1 !       Instead it resumes where the blocked request left off.
1 !
1 !--
1
2     BEGIN
2
2     EXTERNAL
2         IO_PACKET;
2
2     LOCAL
2         STATUS;
2
2     EXTERNAL ROUTINE
2         DO_CANCEL       : COMMON_CALL,            ! cancel i/o
2         UNBLOCK;                                  ! unblock processing
2
2     EXTERNAL REGISTER
2         COMMON_REG;
2
2     CURRENT_VCB = .VCB;
2
2     ! If cancel I/O request came thru while spacing tape mark, then restore
2     ! UCB address, adjust number of tape marks if successful, and cancel
2     ! request
2     !
2     IF .CURRENT_VCB[VCB$V_CANCELIO]
2     THEN
```

```
;  891        1263  3             BEGIN
;  892        1264  5             CURRENT_UCB = .(.CURRENT_VCB[VCB$L_VPBL] + VVP$K_LENGTH + (CURRENT_UCB
;  893        1265  3             - USER_STATUS[0]));
;  894        1266  3
;  895        1267  3             IF .STATUS<0,16> EQL SS$_ENDOFTAPE OR .STATUS
;  896        1268  3             THEN
;  897        1269  3                 KERNEL_CALL(ADJTM, .BBLOCK[.CURRENT_VCB[VCB$L_VPFL], VVP$L_NO_TM]);
;  898        1270  3
;  899        1271  3             ERROR(SS$_CANCEL);
;  900        1272  3             KERNEL_CALL(DO_CANCEL);
;  901        1273  3             IO_PACKET = 0;
;  902        1274  3             RETURN;
;  903        1275  3
;  904        1276  2             END;
;  905        1277  2
;  906        1278  2         ! Unblock process and continue where request processing left off.
;  907        1279  2         !
;  908        1280  2         UNBLOCK();
;  909        1281  1         END;                                                      ! end of routine
; INFO#250                L1:1267
; Referenced LOCAL symbol STATUS is probably not initialized


                                                              .EXTRN  IO_PACKET, DO_CANCEL
                                                              .EXTRN  UNBLOCK

                                   0004 00000 UNBLOCK_SPACE:
                                                              .WORD   Save R2                                       ; 1206
                    52 00000000G   9F   9E 00002              MOVAB   @#SYS$CMKRNL, R2                              ;
                    5B          04 AC   D0 00009              MOVL    VCB, CURRENT_VCB                              ; 1255
          42     0B AB             05   E1 0000D              BBC     #5, 11(CURRENT_VCB), 3$                       ; 1261
          50     40 AB 00000000*   8F   C1 00012              ADDL3   #<<CURRENT_UCB-USER_STATUS>+12>, -            ; 1264
                                                                      64(CURRENT_VCB), R0                           ;
               0000G  CF           60   D0 0001B              MOVL    (R0), CURRENT_UCB                             ;
               0878   8F           50   B1 00020              CMPW    STATUS, #2168                                 ; 1267
                                   03   13 00025              BEQL    1$                                            ;
                    13             50   E9 00027              BLBC    STATUS, 2$                                    ;
                    50          3C AB   D0 0002A  1$:         MOVL    60(CURRENT_VCB), R0                           ; 1269
                              01C4 C0   DD 0002E              PUSHL   452(R0)                                       ;
                                   01   DD 00032              PUSHL   #1                                            ;
                                   5E   DD 00034              PUSHL   SP                                            ;
                              FDBF CF   9F 00036              PUSHAB  ADJTM                                         ;
                    62             04   FB 0003A              CALLS   #4, SYS$CMKRNL                                ;
               0000G  CF      0830 8F   B0 0003D  2$:         MOVW    #2096, USER_STATUS                            ; 1271
                                   7E   D4 00044              CLRL    -(SP)                                         ; 1272
                                   5E   DD 00046              PUSHL   SP                                            ;
                              0000G CF  9F 00048              PUSHAB  DO_CANCEL                                     ;
                    62             03   FB 0004C              CALLS   #3, SYS$CMKRNL                                ;
                              0000G CF  D4 0004F              CLRL    IO_PACKET                                     ; 1273
                                        04 00053              RET                                                   ; 1263
               0000G  CF           00   FB 00054  3$:         CALLS   #0, UNBLOCK                                   ; 1280
                                        04 00059              RET                                                   ; 1281

; Routine Size:  90 bytes,    Routine Base:  $CODE$ + 0352
;  911        1282  1 GLOBAL ROUTINE CHCK_IO_CLR_EXCP : COMMON_CALL NOVALUE =
;  912        1283  1
;  913        1284  1 !++
;  914        1285  1 !
;  915        1286  1 ! FUNCTIONAL DESCRIPTION:
;  916        1287  1 !       This routine saves the  drives characteristics than does
;  917        1288  1 !       a QIOW set mode to the device to ensure that
;  918        1289  1 !       all outstanding reads or writes have been posted to the VCB
;  919        1290  1 !       before processing continues. This is necessary to ensure consistant
;  920        1291  1 !       behaviour between the old class of tape drives and the new type
;  921        1292  1 !       which speak tape protocol. The old tape drivers will still
;  922        1293  1 !       put all oustanding IO's on the VCB's bolocked IO queue. The new
;  923        1294  1 !       drivers will complete these IO's with an error of SS$_SERIOUSEXCP
;  924        1295  1 !       which the ACP will queue on it's blocked IO queue.
;  925        1296  1 !
;  926        1297  1 ! CALLING SEQUENCE:
;  927        1298  1 !       CHCK_IO_CLR_EXCP()
;  928        1299  1 !
;  929        1300  1 ! INPUT PARAMETERS:
;  930        1301  1 !       none
;  931        1302  1 !
;  932        1303  1 ! IMPLICIT INPUTS:
;  933        1304  1 !       IO_CHANNEL
;  934        1305  1 !       CURRENT_UCB
;  935        1306  1 !
;  936        1307  1 ! OUTPUT PARAMETERS:
;  937        1308  1 !       none
;  938        1309  1 !
;  939        1310  1 ! IMPLICIT OUTPUTS:
;  940        1311  1 !       none
;  941        1312  1 !
;  942        1313  1 ! ROUTINE VALUE:
;  943        1314  1 !       none
;  944        1315  1 !
;  945        1316  1 !--
;  946        1317  1
;  947        1318  2     BEGIN
;  948        1319  2
;  949        1320  2     EXTERNAL REGISTER
;  950        1321  2         COMMON_REG;
;  951        1322  2
;  952        1323  2     LOCAL
;  953        1324  2         SAVE_DEVCHAR : VECTOR [2],              ! Characteristics of drive
;  954        1325  2         STATUS;                                 ! io status
;  955        1326  2
;  956        1327  2     SAVE_DEVCHAR [0] = .(CURRENT_UCB[UCB$B_DEVCLASS])<0,32>;
;  957        1328  2     SAVE_DEVCHAR [1] = .CURRENT_UCB[UCB$L_DEVDEPEND];
;  958        1329  2     STATUS = ISSUE_IO ( IO$_SETMODE, SAVE_DEVCHAR, 0);
;  959        1330  2
;  960        1331  1     END;




                                0000 00000          .ENTRY  CHCK_IO_CLR_EXCP, Save nothing            ; 1282
                 5E          04  C2 00002          SUBL2   #4, SP                                     ;
```

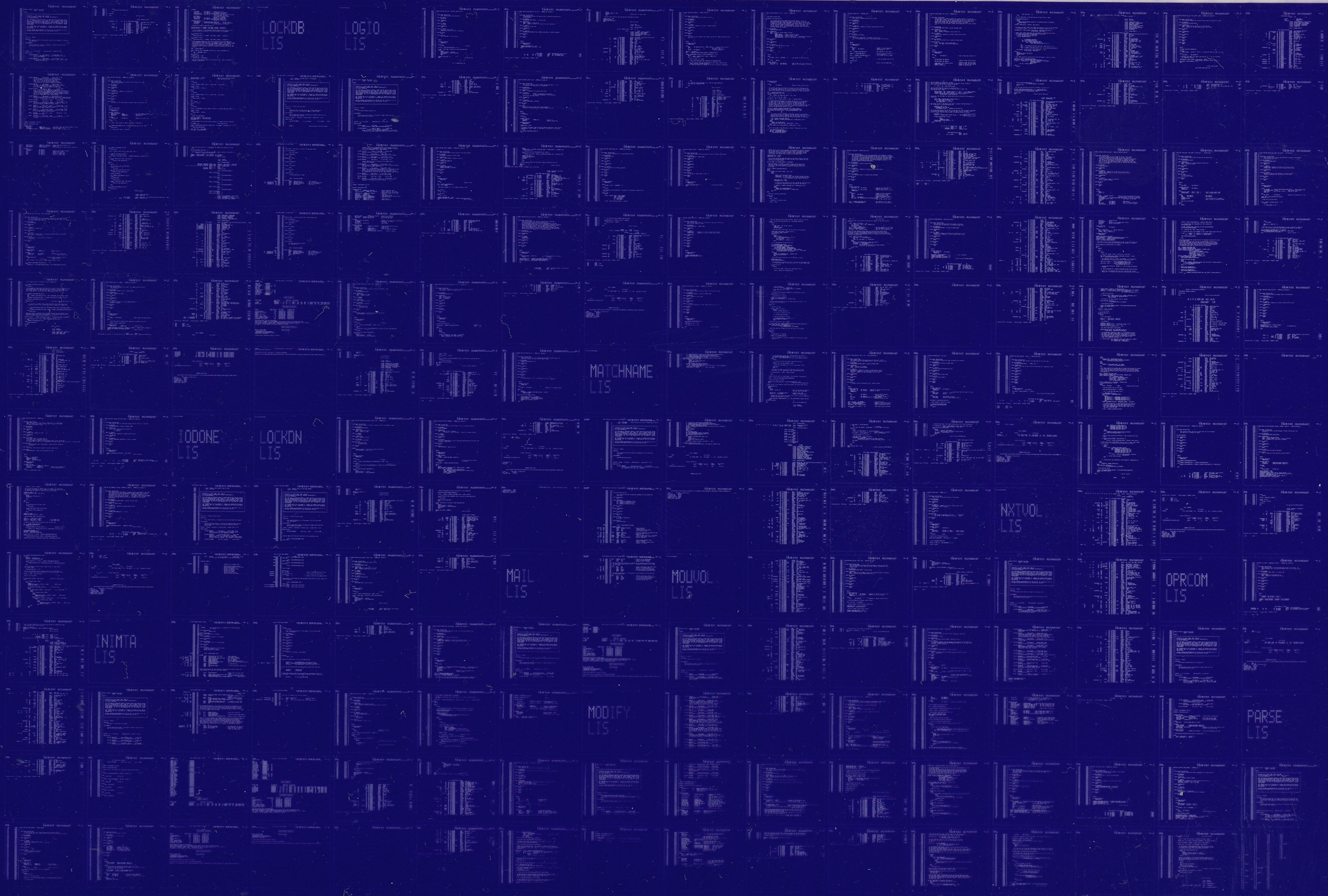
```
                     50    0000G  CF   D0 00005          MOVL    CURRENT_UCB, R0                       ; 1327
                              40  A0   DD 0000A          PUSHL   64(R0)                                 ;
              04     AE       44  A0   D0 0000D          MOVL    68(R0), SAVE_DEVCHAR+4                 ; 1328
                                  7E   D4 00012          CLRL    -(SP)                                  ; 1329
                              04  AE   9F 00014          PUSHAB  SAVE_DEVCHAR                           ;
                                  23   DD 00017          PUSHL   #35                                    ;
                           FDAC   30   00019          BSBW    ISSUE_IO                               ;
                                  04   0001C          RET                                            ; 1331

; Routine Size:  29 bytes,    Routine Base:  $CODE$ + 03AC


;  961          1332  1
```

```
1 GLOBAL ROUTINE COMPLETE_VIO : COMMON_CALL NOVALUE =
1
1 !++
1 !
1 ! FUNCTIONAL DESCRIPTION:
1 !       This routine gets all I/O's queued off the VCB's blocked queue and
1 !       completes them to the user with an ABORT status
1 !
1 ! CALLING SEQUENCE:
1 !       KERNEL_CALL(COMPLETE_VIO)
1 !
1 ! INPUT PARAMETERS:
1 !       none
1 !
1 ! IMPLICIT INPUTS:
1 !       none
1 !
1 ! OUTPUT PARAMETERS:
1 !       none
1 !
1 ! IMPLICIT OUTPUTS:
1 !       none
1 !
1 ! ROUTINE VALUE:
1 !       none
1 !
1 ! SIDE EFFECTS:
1 !       All outstanding IO's will be completed in error to the user.
1 !
1 !--
1
2     BEGIN
2
2     EXTERNAL REGISTER
2         COMMON_REG;
2
2    LOCAL
2         PACKET : REF BBLOCK;                    ! address of io request packet
2
2    WHILE 1
2    DO
3         BEGIN
3         IF REMQUE (.CURRENT_VCB[VCB$L_BLOCKFL], PACKET)
3             THEN EXITLOOP;
3
3 ! make the error an ABORT status
3
3         PACKET[IRP$L_IOST1] = SS$_ABORT;
3         USER_STATUS[0] = .PACKET[IRP$L_IOST1];
3         USER_STATUS[1] = .PACKET[IRP$L_IOST2];
3         KERNEL_CALL(IO_DONE, .PACKET);
3         END
1    END;
```

```
                                    0004 00000          .ENTRY   COMPLETE_VIO, Save R2                     ; 1333
                52       00    BB   0F 00002 1$:        REMQUE   @0(CURRENT_VCB), PACKET                   ; 1375
                               1D   1D 00006            BVS      2$                                        ;
            38  A2                  2C   D0 00008       MOVL     #44, 56(PACKET)                           ; 1380
         0000G  CF       38    A2   7D 0000C            MOVQ     56(PACKET), USER_STATUS                   ; 1381
                               52   DD 00012            PUSHL    PACKET                                    ; 1383
                               01   DD 00014            PUSHL    #1                                        ;
                               5E   DD 00016            PUSHL    SP                                        ;
                      0000G    CF   9F 00018            PUSHAB   IO_DONE                                   ;
     00000000G  9F             04   FB 0001C            CALLS    #4, @#SYS$CMKRNL                          ;
                               DD   11 00023            BRB      1$                                        ; 1372
                                    04 00025 2$:        RET                                                ; 1385

; Routine Size:  38 bytes,    Routine Base:  $CODE$ + 03C9


; 1016          1386  1 END
; 1017          1387  0 ELUDOM



;                               PSECT SUMMARY
;
;       Name                    Bytes                       Attributes
;
;   $CODE$                          1007  NOVEC,NOWRT,  RD ,  EXE,NOSHR,  LCL,  REL,  CON,NOPIC,ALIGN(2)


;                          Library Statistics
;
;                                        -------- Symbols --------       Pages        Processing
;       File                             Total    Loaded   Percent       Mapped       Time
;
;   _$255$DUA28:[SYSLIB]LIB.L32;1        18619       45         0        1000          00:01.9


; Information:  1
; Warnings:     0
; Errors:       0


;                               COMMAND QUALIFIERS

;       BLISS/CHECK=(FIELD,INITIAL,OPTIMIZE)/LIS=LIS$:LOGIO/OBJ=OBJ$:LOGIO MSRC$:LOGIO/UPDATE=(ENH$:LOGIO)

; Size:          1007 code + 0 data bytes
; Run Time:         00:22.4
; Elapsed Time:     01:05.6
; Lines/CPU Min:      3720
; Lexemes/CPU-Min: 19424
; Memory Used:  126 pages
; Compilation Complete
```

0255 AH-BT13A-SE
VAX/VMS V4.0

DIGITAL EQUIPMENT CORPORATION
CONFIDENTIAL AND PROPRIETARY



LOCKDB
LIS

LOGIO
LIS

MATCHNAME
LIS

IODONE
LIS

LOCKDN
LIS

NXTVOL
LIS

MAIL
LIS

MOUVOL
LIS

OPRCOM
LIS

INIMTA
LIS

MODIFY
LIS

PARSE
LIS