M 2

XF
VO

```
IIIIIIIII        OOOOOOOOO        SSSSSSSSSSSS  UUU         UUU  PPPPPPPPPPPP
IIIIIIIII        OOOOOOOOO        SSSSSSSSSSSS  UUU         UUU  PPPPPPPPPPPP
IIIIIIIII        OOOOOOOOO        SSSSSSSSSSSS  UUU         UUU  PPPPPPPPPPPP
   III        OOO         OOO  SSS              UUU         UUU  PPP         PPP
   III        OOO         OOO  SSS              UUU         UUU  PPP         PPP
   III        OOO         OOO  SSS              UUU         UUU  PPP         PPP
   III        OOO         OOO  SSS              UUU         UUU  PPP         PPP
   III        OOO         OOO  SSS              UUU         UUU  PPP         PPP
   III        OOO         OOO  SSS              UUU         UUU  PPP         PPP
   III        OOO         OOO     SSSSSSSSS     UUU         UUU  PPPPPPPPPPPP
   III        OOO         OOO     SSSSSSSSS     UUU         UUU  PPPPPPPPPPPP
   III        OOO         OOO     SSSSSSSSS     UUU         UUU  PPPPPPPPPPPP
   III        OOO         OOO              SSS  UUU         UUU  PPP
   III        OOO         OOO              SSS  UUU         UUU  PPP
   III        OOO         OOO              SSS  UUU         UUU  PPP
   III        OOO         OOO              SSS  UUU         UUU  PPP
   III        OOO         OOO              SSS  UUU         UUU  PPP
   III        OOO         OOO              SSS  UUU         UUU  PPP
IIIIIIIII        OOOOOOOOO     SSSSSSSSSSSS     UUUUUUUUUUUUUUU  PPP
IIIIIIIII        OOOOOOOOO     SSSSSSSSSSSS     UUUUUUUUUUUUUUU  PPP
IIIIIIIII        OOOOOOOOO     SSSSSSSSSSSS     UUUUUUUUUUUUUUU  PPP
```

```
**FILE**ID**LABUFFER                                             G  7                                                             LP
                                                                                                                                  V0
```

```
LL            AAAAAA    BBBBBBBB    UU      UU  FFFFFFFFFF  FFFFFFFFFF  EEEEEEEEEE  RRRRRRRR
LL            AAAAAA    BBBBBBBB    UU      UU  FFFFFFFFFF  FFFFFFFFFF  EEEEEEEEEE  RRRRRRRR
LL          AA      AA  BB      BB  UU      UU  FF          FF          EE          RR      RR
LL          AA      AA  BB      BB  UU      UU  FF          FF          EE          RR      RR
LL          AA      AA  BB      BB  UU      UU  FF          FF          EE          RR      RR
LL          AA      AA  BB      BB  UU      UU  FF          FF          EE          RR      RR
LL          AA      AA  BBBBBBBB    UU      UU  FFFFFFFF    FFFFFFFF    EEEEEEEE    RRRRRRRR
LL          AA      AA  BBBBBBBB    UU      UU  FFFFFFFF    FFFFFFFF    EEEEEEEE    RRRRRRRR
LL          AAAAAAAAAA  BB      BB  UU      UU  FF          FF          EE          RR  RR
LL          AAAAAAAAAA  BB      BB  UU      UU  FF          FF          EE          RR  RR
LL          AA      AA  BB      BB  UU      UU  FF          FF          EE          RR    RR        ....
LL          AA      AA  BB      BB  UU      UU  FF          FF          EE          RR    RR        ....
LLLLLLLLLL  AA      AA  BBBBBBBB    UUUUUUUUUU  FF          FF          EEEEEEEEEE  RR      RR      ....
LLLLLLLLLL  AA      AA  BBBBBBBB    UUUUUUUUUU  FF          FF          EEEEEEEEEE  RR      RR      ....

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
```

```
0000            .TITLE  LPA$BUFFER
0000            .IDENT  'V04-000'
0000
0000    ;
0000    ;****************************************************************************
0000    ;*                                                                          *
0000    ;*  COPYRIGHT (c) 1978, 1980, 1982, 1984 BY                                 *
0000    ;*  DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASSACHUSETTS.                  *
0000    ;*  ALL RIGHTS RESERVED.                                                    *
0000    ;*                                                                          *
0000    ;*  THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND COPIED   *
0000    ;*  ONLY IN  ACCORDANCE WITH  THE  TERMS  OF  SUCH  LICENSE  AND WITH THE   *
0000    ;*  INCLUSION OF THE ABOVE COPYRIGHT NOTICE. THIS SOFTWARE OR  ANY  OTHER   *
0000    ;*  COPIES THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY   *
0000    ;*  OTHER PERSON.  NO TITLE TO AND OWNERSHIP OF  THE  SOFTWARE IS  HEREBY   *
0000    ;*  TRANSFERRED.                                                            *
0000    ;*                                                                          *
0000    ;*  THE INFORMATION IN THIS SOFTWARE IS  SUBJECT TO CHANGE WITHOUT NOTICE   *
0000    ;*  AND  SHOULD  NOT  BE  CONSTRUED AS  A COMMITMENT BY DIGITAL EQUIPMENT   *
0000    ;*  CORPORATION.                                                            *
0000    ;*                                                                          *
0000    ;*  DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE  OR  RELIABILITY OF ITS   *
0000    ;*  SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL.                 *
0000    ;*                                                                          *
0000    ;*                                                                          *
0000    ;****************************************************************************
0000    ;
0000
0000    ;++
0000    ; FACILITY:     LPA-11 PROCEDURE LIBRARY
0000    ;
0000    ; ABSTRACT:
0000    ;       THIS MODULE CONTAINS THE ROUTINES WHICH MANIPULATE BUFFERS
0000    ;       AS PART OF THE LPA-11 PROCEDURE LIBRARY.  THIS INCLUDES THE
0000    ;       AST ROUTINES.
0000    ;
0000    ; ENVIRONMENT:  USER MODE, SHARED OR NON-SHARED LIBRARY
0000    ;
0000    ; AUTHOR:  STEVE BECKHARDT,     CREATION DATE:  28-AUG-78
0000    ;
0000    ; MODIFIED BY:
0000    ;
0000    ;       V03-001 SBL3001         Steven B. Lionel        30-March-1982
0000    ;               Change module name to LPA$BUFFER.
0000    ;--
```

```
    0000            .SBTTL  DECLARATIONS
    0000    ;
    0000    ; INCLUDE FILES:
    0000    ;
    0000
    0000    ;
    0000    ; MACROS:
    0000    ;
    0000
    0000    ;
    0000    ; EQUATED SYMBOLS:
    0000    ;
    0000
    0000    ;
    0000    ; OWN STORAGE:
    0000    ;
    0000
00000000            .PSECT  _LPA$CODE,PIC,SHR,EXE,NOWRT,LONG
    0000
    0000
    0000   LPA$BFSTART::                            ; START OF BUFFER ROUTINES (MAY BE
    0000                                            ; USED TO LOCK THESE ROUTINES INTO
    0000                                            ; THE PROCESS'S WORKING SET).
```

```
                            0000          .SBTTL  LPA$IBFSTS - GET BUFFER STATUS
                            0000  ;++
                            0000  ; FUNCTIONAL DESCRIPTION
                            0000  ;
                            0000  ;       THIS ROUTINE RETURNS THE STATUS OF EACH OF THE BUFFERS.
                            0000  ;       THE STATUS IS DEFINED AS WHICH QUEUE THE BUFFER IS ON.
                            0000  ;
                            0000  ; CALLING SEQUENCE:
                            0000  ;
                            0000  ;       CALLS/G
                            0000  ;
                            0000  ; INPUT PARAMETERS:
                            0000  ;
                            0000  ;       IBUF(AP)                ADDRESS OF IBUF ARRAY
                            0000  ;       ISTAT(AP)               ADDRESS OF LONGWORD ARRAY WITH AS MANY
                            0000  ;                               ELEMENTS AS THERE ARE BUFFERS INVOLVED
                            0000  ;                               IN THE SWEEP
                            0000  ;
                            0000  ; IMPLICIT INPUTS:
                            0000  ;
                            0000  ;       VARIOUS FIELDS IN THE IBUF ARRAY
                            0000  ;
                            0000  ; OUTPUT PARAMETERS:
                            0000  ;
                            0000  ;       ISTAT(AP)               ADDRESS OF LONGWORD ARRAY WITH AS MANY
                            0000  ;                               ELEMENTS AS THERE ARE BUFFERS INVOLVED
                            0000  ;                               IN THE SWEEP.  EACH ELEMENT IS FILLED WITH
                            0000  ;                               THE STATUS OF THE CORRESPONDING BUFFER.
                            0000  ;                               THE STATUS CODES ARE AS FOLLOWS:
                            0000  ;
                            0000  ;                                       2       BUFFER IS ON DEVICE QUEUE
                            0000  ;                                       1       BUFFER IS ON USER QUEUE
                            0000  ;                                       0       BUFFER IS NOT ON ANY QUEUE
                            0000  ;                                      -1       BUFFER IS ON INUSE QUEUE
                            0000  ;
                            0000  ; IMPLICIT OUTPUTS:
                            0000  ;
                            0000  ;       NONE
                            0000  ;
                            0000  ; COMPLETION CODES:
                            0000  ;
                            0000  ;       NONE
                            0000  ;
                            0000  ; SIDE EFFECTS:
                            0000  ;
                            0000  ;       VARIOUS FIELDS IN THE IBUF ARRAY ARE MODIFIED
                            0000  ;
                            0000  ;--
                            0000
                      005C  0000          .ENTRY  LPA$IBFSTS,^M<R2,R3,R4,R6>
                            0002
            56   04 AC  D0  0002          MOVL    4(AP),R6                ; GET ADDRESS OF IBUF ARRAY
            54   08 AC  D0  0006          MOVL    8(AP),R4                ; GET ADDRESS OF ISTAT ARRAY
                            000A
   50   22 A6   03   00  EF  000A          EXTZV   #0,#3,IBF$L_CMDTBL+CMT$B_VBFRMASK(R6),R0  ; GET HIGHEST BUF. #
                            0010
                            0010          ; INITIALIZE ALL ENTRIES IN ISTAT ARRAY TO 0
            6440   D4  0010  10$:    CLRL    (R4)[R0]
           FA 50   F4  0013          SOBGEQ  R0,10$
                       0016
                       0016          ; *** TURN OFF ASTS ***
                       0016          $SETAST_S       #0
                       001F
                       001F          ; STORE STATUS FOR ALL BUFFERS ON THE DEVICE QUEUE
        53    02   D0  001F          MOVL    #2,R3                    ; STATUS = 2
     52   58 A6    DE  0022          MOVAL   IBF$L_DEVQFL(R6),R2      ; ADDRESS OF DEVICE QUEUE
              1B   10  0026          BSBB    FOLLOWQ                  ; FOLLOW THE QUEUE
                       0028
                       0028          ; STORE STATUS FOR ALL BUFFERS ON THE USER QUEUE
              53   D7  0028          DECL    R3                       ; STATUS = 1
     52   50 A6    DE  002A          MOVAL   IBF$L_USRQFL(R6),R2      ; ADDRESS OF USER QUEUE
              13   10  002E          BSBB    FOLLOWQ                  ; FOLLOW THE QUEUE
                       0030
                       0030          ; STORE STATUS FOR ALL BUFFERS ON THE INUSE QUEUE
        53    01   CE  0030          MNEGL   #1,R3                    ; STATUS = -1
     52   60 A6    DE  0033          MOVAL   IBF$L_INUQFL(R6),R2      ; ADDRESS OF INUSE QUEUE
              0A   10  0037          BSBB    FOLLOWQ                  ; FOLLOW THE QUEUE
                       0039
                       0039          ; *** TURN ON ASTS ***
                       0039          $SETAST_S       #1
                   04  0042          RET
                       0043
                       0043
                       0043          ;
                       0043          ; LOCAL SUBROUTINE TO FOLLOW QUEUE AND STORE STATUS IN ISTAT ARRAY
                       0043          ;
                       0043          ;       INPUT:  R2 CONTAINS ADDRESS OF QUEUE HEAD
                       0043          ;               R3 CONTAINS STATUS TO STORE IN ISTAT ARRAY
                       0043          ;
                       0043  FOLLOWQ:
        51    52   D0  0043          MOVL    R2,R1                    ; COPY ADDRESS OF QUEUE HEAD
                       0046
        51    61   D0  0046  10$:    MOVL    (R1),R1                  ; GET ADDRESS OF NEXT LINK
        52    51   D1  0049          CMPL    R1,R2                    ; BACK TO QUEUE HEAD YET?
              09   13  004C          BEQL    20$                      ; YES, DONE
          01EA     30  004E          BSBW    CVTADINDX                ; NO, CONVERT ADDRESS TO BUFFER INDEX
     6440     53   D0  0051          MOVL    R3,(R4)[R0]              ; STORE STATUS IN CORRESPONING ENTRY
              EF   11  0055          BRB     10$                      ; OF ISTAT AND BRANCH BACK FOR NEXT ONE
                   05  0057  20$:    RSB
```

```
                         0058          .SBTTL  LPA$IGTBUF - GET A BUFFER
                         0058  ;++
                         0058  ; FUNCTIONAL DESCRIPTION:
                         0058  ;
                         0058  ;       THIS ROUTINE IS CALLED BY THE USER'S PROGRAM TO GET A BUFFER
                         0058  ;       FROM THE HEAD OF THE USER QUEUE.  UNLIKE LPA$IWTBUF, THIS ROUTINE
                         0058  ;       RETURNS IMMEDIATELY IF THE QUEUE IS EMPTY.  THIS ROUTINE IS INTENDED
                         0058  ;       TO BE CALLED FROM THE USER'S COMPLETION ROUTINE (WHICH IS CALLED BY
                         0058  ;       THE AST HANDLERS).  THIS ROUTINE SHOULD BE CALLED IF THE USER
                         0058  ;       IS SYNCHRONIZING WITH A COMPLETION ROUTINE, RATHER THAN WITH EVENT
                         0058  ;       FLAGS.
                         0058  ;
                         0058  ; CALLING SEQUENCE:
                         0058  ;
                         0058  ;       CALLS/G
                         0058  ;       MAY ALSO BE CALLED AS A FUNCTION
                         0058  ;
                         0058  ; INPUT PARAMETERS:
                         0058  ;
                         0058  ;       IBUF(AP)                ADDRESS OF IBUF ARRAY
                         0058  ;       IBUFNO(AP)              ADDRESS OF LONGWORD WHICH RECEIVES BUFFER
                         0058  ;                               INDEX OR -1 WHICH INDICATES NO BUFFER IN QUEUE
                         0058  ;
                         0058  ; IMPLICIT INPUTS:
                         0058  ;
                         0058  ;       VARIOUS FIELDS IN THE IBUF ARRAY
                         0058  ;
                         0058  ; OUTPUT PARAMETERS:
                         0058  ;
                         0058  ;       IBUFNO(AP)              ADDRESS OF LONGWORD WHICH RECEIVES BUFFER
                         0058  ;                               INDEX OR -1 WHICH INDICATES NO BUFFFR IN QUEUE
                         0058  ;
                         0058  ; IMPLICIT OUTPUTS:
                         0058  ;
                         0058  ;       OFFSET IBF$Q_IOST IN THE IBUF ARRAY CONTAINS ADDITIONAL STATUS
                         0058  ;
                         0058  ; COMPLETION CODES:
                         0058  ;
                         0058  ;       NONE
                         0058  ;
                         0058  ; SIDE EFFECTS:
                         0058  ;
                         0058  ;       VARIOUS FIELDS IN THE IBUF ARRAY ARE MODIFIED
                         0058  ;
                         0058  ;--
                         0058
                   0040  0058          .ENTRY  LPA$IGTBUF,^M<R6>
                         005A
       56   04 AC    D0  005A          MOVL    4(AP),R6                ; GET ADDRESS OF IBUF ARRAY
                         005E
                         005E          ; REMOVE BUFFER FROM HEAD OF USER QUEUE
       51   50 B6    0F  005E          REMQUE  @IBF$L_USRQFL(R6),R1    ; R1 GETS ADDRESS OF LINK
                 0C  1D  0062          BVS     10$                     ; NO ENTRY IN QUEUE
               01D4  30  0064          BSBW    CVTADINDX               ; CONVERT ADDRESS TO INDEX IN R0
                 66  B4  0067          CLRW    IBF$Q_IOST(R6)          ; CLEAR STATUS
    02 A6   14 A6    B0  0069          MOVW    IBF$L_LBUF(R6),IBF$Q_IOST+2(R6) ; STORE BUFFER LENGTH
                 07  11  006E          BRB     20$
```

```
                      0070
                      0070   10$:    ; USER QUEUE IS EMPTY - RETURN REAL I/O STATUS
66    08 A6     7D    0070           MOVQ    IBF$Q_IOSB(R6),IBF$Q_IOST(R6)
   50    01     CE    0074           MNEGL   #1,R0                        ; RETURN -1
                      0077
   02    6C     91    0077   20$:    CMPB    (AP),#2                      ; IBUFNO SUPPLIED?
         09     1F    007A           BLSSU   40$                          ; NO
51    08 AC     D0    007C           MOVL    8(AP),R1                     ; GET ADDRESS
         03     13    0080           BEQL    40$                          ; DEFAULTED
   61    50     D0    0082           MOVL    R0,(R1)                      ; STORE INDEX
                04    0085   40$:    RET                                  ; INDEX IS IN R0 FOR FUNCTION CALL
```

```
                      0086          .SBTTL  LPA$INXTBF - SET NEXT BUFFER
                      0086          .SBTTL  LPA$RMVBUF - REMOVE BUFFER
                      0086  ;++
                      0086  ; FUNCTIONAL DESCRIPTION:
                      0086  ;
                      0086  ;       LPA$INXTBF CHANGES THE ORDER OF BUFFERS IN THE DEVICE QUEUE
                      0086  ;       BY INSERTING THE INDICATED BUFFER AT THE HEAD OF THE QUEUE.
                      0086  ;       NOTE THAT THE BUFFER MUST ALREADY BE ON THE DEVICE QUEUE.
                      0086  ;       LPA$RMVBUF REMOVES A BUFFER FROM THE DEVICE QUEUE.
                      0086  ;
                      0086  ; CALLING SEQUENCE:
                      0086  ;
                      0086  ;       CALLS/G
                      0086  ;       THESE ROUTINES MAY ALSO BE CALLED AS FUNCTIONS
                      0086  ;
                      0086  ; INPUT PARAMETERS:
                      0086  ;
                      0086  ;       IBUF(AP)                ADDRESS OF IBUF ARRAY
                      0086  ;       IBUFNO(AP)              ADDRESS OF LONGWORD WHICH CONTAINS INDEX
                      0086  ;                               OF BUFFER TO BE REMOVED OR MADE NEXT
                      0086  ;       IND(AP)                 ADDRESS OF LONGWORD TO RECEIVE RETURN STATUS
                      0086  ;
                      0086  ; IMPLICIT INPUTS:
                      0086  ;
                      0086  ;       VARIOUS FIELDS IN THE IBUF ARRAY
                      0086  ;
                      0086  ; OUTPUT PARAMETERS:
                      0086  ;
                      0086  ;       IND(AP)                 ADDRESS OF LONGWORD TO RECEIVE RETURN STATUS
                      0086  ;
                      0086  ; IMPLICIT OUTPUTS:
                      0086  ;
                      0086  ;       NONE
                      0086  ;
                      0086  ; COMPLETION CODES:
                      0086  ;
                      0086  ;       0       INDICATES THAT THE SPECIFIED BUFFER WAS NOT IN THE DEVICE QUEUE
                      0086  ;       1       INDICATES THAT THE SPECIFIED BUFFER WAS REMOVED OR MADE NEXT
                      0086  ;
                      0086  ; SIDE EFFECTS:
                      0086  ;
                      0086  ;       VARIOUS FIELDS IN THE IBUF ARRAY ARE MODIFIED
                      0086  ;
                      0086  ;--
                      0086
                005C  0086          .ENTRY  LPA$INXTBF,^M<R2,R3,R4,R6>
      54    01    D0  0088          MOVL    #1,R4                   ; INDICATES WHICH CALL
            04    11  008B          BRB     COMMON
                      008D
                005C  008D          .ENTRY  LPA$RMVBUF,^M<R2,R3,R4,R6>
            54    D4  008F          CLRL    R4                      ; INDICATES WHICH CALL
                      0091
                      0091
  56     04 AC    D0  0091  COMMON: MOVL    4(AP),R6                ; GET ADDRESS OF IBUF ARRAY
            52    D4  0095          CLRL    R2                      ; RETURN CODE
                      0097
  53     08 BC    D0  0097          MOVL    @8(AP),R3               ; GET BUFFER INDEX
  53    68 A643    7E  009B             MOVAQ   IBF$Q_BFRLNKS(R6)[R3],R3; CONVERT TO ADDRESS
                       00A0
                       00A0             ; *** TURN OFF ASTS ***
                       00A0             $SETAST_S       #0
                       00A9
   51    58 A6     DE  00A9             MOVAL   IBF$L_DEVQFL(R6),R1     ; GET ADDRESS OF DEVICE QUEUE HEAD
      50    51     D0  00AD             MOVL    R1,R0
                       00B0
                       00B0     10$:    ; GET NEXT LINK IN QUEUE
      50    60     D0  00B0             MOVL    (R0),R0
      51    50     D1  00B3             CMPL    R0,R1                   ; BACK TO QUEUE HEAD YET?
            11     13  00B6             BEQL    30$                     ; YES, BUFFER NOT ON QUEUE
                       00B8
      53    50     D1  00B8             CMPL    R0,R3                   ; IS THIS THE BUFFER?
            F3     12  00BB             BNEQ    10$                     ; NO
                       00BD
                       00BD             ; HAVE BUFFER.  ADDRESS IS IN R0.  REMOVE FROM QUEUE
      50    60     0F  00BD             REMQUE  (R0),R0
                       00C0
                       00C0             ; IF LPA$INXTBF WAS ENTRY, THEN INSERT BUFFER AT HEAD OF QUEUE.
         04 54     E9  00C0             BLBC    R4,20$                  ; BR. IF LPA$RMVBUF WAS ENTRY
   58 A6    60     0E  00C3             INSQUE  (R0),IBF$L_DEVQFL(R6)   ; INSERT AT HEAD OF DEVICE QUEUE
                       00C7
            52     D6  00C7     20$:    INCL    R2                      ; INDICATE SUCCESS
                       00C9
                       00C9     30$:    ;*** TURN ON ASTS ***
                       00C9             $SETAST_S       #1
                       00D2
      50    52     D0  00D2             MOVL    R2,R0                   ; RETURN STATUS
      03    6C     91  00D5             CMPB    (AP),#3                 ; IND SPECIFIED?
            09     1F  00D8             BLSSU   40$                     ; NO
   51    0C AC     D0  00DA             MOVL    12(AP),R1               ; GET ADDRESS OF IND
            03     13  00DE             BEQL    40$                     ; DEFAULTED
      61    50     D0  00E0             MOVL    R0,(R1)                 ; STORE STATUS
                   04  00E3     40$:    RET
```

```
00E4           .SBTTL  LPA$IWTBUF - WAIT FOR A BUFFER
00E4     ;++
00E4     ;FUNCTIONAL DESCRIPTION:
00E4     ;
00E4     ;       THIS ROUTINE IS CALLED BY THE USER'S PROGRAM TO GET A BUFFER
00E4     ;       FROM THE HEAD OF THE USER QUEUE.  HOWEVER, UNLIKE LPA$IGTBUF,
00E4     ;       THIS ROUTINE WAITS FOR ONE IF THE QUEUE IS EMPTY.  THIS ROUTINE
00E4     ;       SHOULD BE USED IF THE USER IS SYNCHRONIZING WITH EVENT FLAGS
00E4     ;       RATHER THAN WITH A COMPLETION ROUTINE.  THIS ROUTINE DOES THE FOLLOWING:
00E4     ;               1)  REMOVES A BUFFER FROM THE HEAD OF THE USER QUEUE.
00E4     ;                   IF THERE IS ONE, IT RETURNS WITH THE BUFFER INDEX.
00E4     ;               2)  OTHERWISE, THE EVENT FLAG IS CLEARED.
00E4     ;               3)  ANOTHER ATTEMPT IS MADE TO REMOVE A BUFFER FROM THE
00E4     ;                   HEAD OF THE USER QUEUE.  IF THERE IS ONE, IT IS RETURNED.
00E4     ;               4)  OTHERWISE, THE I/O STATUS BLOCK IS CHECKED TO SEE IF THE
00E4     ;                   SWEEP IS STILL IN PROGRESS.  IF NOT, THE STATUS IS RETURNED.
00E4     ;               5)  OTHERWISE, WE WAIT FOR THE EVENT FLAG AND THEN GO TO STEP 1.
00E4     ;       NOTE THAT STEP 1 IS ONLY AN OPTIMIZATION:  IT SAVES CLEARING THE
00E4     ;       EVENT FLAG IF A BUFFER IS ALREADY AVAILABLE.
00E4     ;
00E4     ; CALLING SEQUENCE:
00E4     ;
00E4     ;       CALLS/G
00E4     ;       MAY ALSO BE CALLED AS A FUNCTION
00E4     ;
00E4     ; INPUT PARAMETERS:
00E4     ;
00E4     ;       IBUF(AP)                ADDRESS OF IBUF ARRAY
00E4     ;       IEFN(AP)                UNUSED (PRESENT FOR COMPATIBILITY ONLY)
00E4     ;       IBUFNO(AP)              ADDRESS OF A LONGWORD WHICH RECEIVES BUFFER
00E4     ;                               INDEX OR -1 IF THERE ARE NO BUFFERS AND THE
00E4     ;                               SWEEP HAS STOPPED.
00E4     ;
00E4     ; IMPLICIT INPUTS:
00E4     ;
00E4     ;       VARIOUS FIELDS IN THE IBUF ARRAY
00E4     ;
00E4     ; OUTPUT PARAMETERS:
00E4     ;
00E4     ;       IBUFNO(AP)              ADDRESS OF A LONGWORD WHICH RECEIVES BUFFER
00E4     ;                               INDEX OR -1 IF THERE ARE NO BUFFERS AND THE
00E4     ;                               SWEEP HAS STOPPED.
00E4     ;
00E4     ; IMPLICIT OUTPUTS:
00E4     ;
00E4     ;       OFFSET IBF$Q_IOST IN THE IBUF ARRAY CONTAINS ADDITIONAL STATUS
00E4     ;
00E4     ; COMPLETION CODES:
00E4     ;
00E4     ;       NONE
00E4     ;
00E4     ; SIDE EFFECTS:
00E4     ;
00E4     ;       VARIOUS FIELDS IN THE IBUF ARRRAY ARE MODIFIED
00E4     ;
00E4     ;--
00E4
```

```
                 0044  00E4         .ENTRY  LPA$IWTBUF,^M<R2,R6>
                       00E6
   56   04 AC     D0   00E6         MOVL    4(AP),R6                 ; GET ADDRESS OF IBUF ARRAY
   52   4E A6     9A   00EA         MOVZBL  IBF$B_EFN(R6),R2         ; GET EVENT FLAG BEING USED
                       00EE
                       00EE   10$:  ; REMOVE BUFFER FROM HEAD OF USER QUEUE
   51   50 B6     0F   00EE         REMQUE  @IBF$L_USRQFL(R6),R1     ; ADDRESS IN R1
           31     1C   00F2         BVC     70$                      ; HAVE ONE
                       00F4
                       00F4   30$:  ; USER QUEUE IS EMPTY.  CLEAR EVENT FLAG
                       00F4         $CLREF_S        R2               ; CLEAR EVENT FLAG
        17 50     E9   00FD         BLBC    R0,40$                   ; ERROR
                       0100
                       0100         ; TRY AGAIN TO REMOVE A BUFFER FROM HEAD OF USER QUEUE
   51   50 B6     0F   0100         REMQUE  @IBF$L_USRQFL(R6),R1     ; ADDRESS IN R1
           1F     1C   0104         BVC     70$                      ; HAVE ONE
                       0106
                       0106         ; QUEUE IS STILL EMPTY.  CHECK FOR ERROR OR DONE BEFORE WAITING
        08 A6     B5   0106         TSTW    IBF$Q_IOSB(R6)           ; CHECK I/O STATUS BLOCK
           11     12   0109         BNEQ    50$                      ; ERROR OR DONE
                       010B
                       010B         $WAITFR_S       R2               ; SWEEP IS STILL ON SO WAIT
        D7 50     E8   0114         BLBS    R0,10$                   ; TRY AGAIN
                       0117
                       0117   40$:  ; ERROR IN CLEARING OR WAITING FOR EVENT FLAG
     66    50     D0   0117         MOVL    R0,IBF$Q_IOST(R6)        ; RETURN ERROR CODE IN I/O STATUS
           04     11   011A         BRB     60$
                       011C
                       011C   50$:  ; SWEEP FINISHED NORMALLY OR WITH ERROR.  RETURN I/O STATUS BLOCK
   66   08 A6     7D   011C         MOVQ    IBF$Q_IOSB(R6),IBF$Q_IOST(R6)
                       0120
     50    01     CE   0120   60$:  MNEGL   #1,R0                    ; RETURN -1 INSTEAD OF INDEX
           0A     11   0123         BRB     80$
                       0125
                       0125   70$:  ; HAVE A BUFFER ADDRESS IN R1.  CONVERT TO INDEX IN R0
         0113     30   0125         BSBW    CVTADINDX
           66     B4   0128         CLRW    IBF$Q_IOST(R6)           ; RETURN 0 STATUS
02 A6   14 A6     B0   012A         MOVW    IBF$L_LBUF(R6),IBF$Q_IOST+2(R6) ; RETURN BUFFER LENGTH
                       012F
     03    6C     91   012F   80$:  CMPB    (AP),#3                  ; IBUFNO SPECIFIED?
           09     1F   0132         BLSSU   90$                      ; NO
   51   0C AC     D0   0134         MOVL    12(AP),R1                ; GET ADDRESS OF IBUFNO
           03     13   0138         BEQL    90$                      ; DEFAULTED
     61    50     D0   013A         MOVL    R0,(R1)                  ; STORE INDEX IN IBUFNO
                  04   013D   90$:  RET                              ; INDEX IN R0 FOR FUNCTION CALL
```

```
013E          .SBTTL  LPA$RLSBUF - RELEASE BUFFER
013E    ;++
013E    ; FUNCTIONAL DESCRIPTION:
013E    ;
013E    ;         THIS ROUTINE IS CALLED BY THE USER'S PROGRAM TO RELEASE A BUFFER
013E    ;         (OR BUFFERS) ONTO THE DEVICE QUEUE.  IT WORKS AS FOLLOWS:
013E    ;                 1)  EACH BUFFER INDEX IS CONVERTED TO AN ADDRESS
013E    ;                 2)  IF THE USW IS NOT CURRENTLY SET WITH A NEXT BUFFER INDEX
013E    ;                     THEN IT IS LOADED WITH THIS BUFFER'S INDEX AND THE BUFFER
013E    ;                     IS INSERTED ON THE TAIL OF THE INUSE QUEUE.
013E    ;                 3)  OTHERWISE, THE BUFFER IS INSERTED ON THE TAIL OF THE
013E    ;                     DEVICE QUEUE.
013E    ;         IF BUFFER OVERRUN IS NON-FATAL, THEN BUFFER 0 IS HANDLED AS A
013E    ;         SPECIAL CASE.  INSTEAD OF BEING INSERTED ON THE DEVICE QUEUE,
013E    ;         A FLAG IS SET.  IF THIS FLAG IS NOT SET WHEN BUFFER OVERRUN OCCURS
013E    ;         THEN THE LPA-11 HAS USED BUFFER 0 BEFORE IT HAS BEEN RELEASED.
013E    ;         THIS IS RETURNED AS A DOUBLE BUFFER OVERRUN ERROR.
013E    ;
013E    ; CALLING SEQUENCE:
013E    ;
013E    ;         CALLS/G
013E    ;
013E    ; INPUT PARAMETERS:
013E    ;
013E    ;         IBUF(AP)                ADDRESS OF IBUF ARRAY
013E    ;         IND(AP)                 ADDRESS OF LONGWORD TO STORE RETURN STATUS
013E    ;         N0(AP)                  ADDRESS OF LONGWORD CONTAINING BUFFER INDEX
013E    ;           .
013E    ;           .
013E    ;         N7(AP)                  ADDRESS OF LONGWORD CONTAINING BUFFER INDEX
013E    ;
013E    ; IMPLICIT INPUTS:
013E    ;
013E    ;         VARIOUS FIELDS IN THE IBUF ARRAY
013E    ;
013E    ; OUTPUT PARAMETERS:
013E    ;
013E    ;         IND(AP)                 ADDRESS OF LONGWORD TO RECEIVE RETURN STATUS
013E    ;
013E    ; IMPLICIT OUTPUTS:
013E    ;
013E    ;         NONE
013E    ;
013E    ; COMPLETION CODES:
013E    ;
013E    ;         0               INDICATES ILLEGAL BUFFER NUMBER, INCORRECT # OF ARGS,
013E    ;                         OR DOUBLE BUFFER OVERRUN OCCURED (THE LAST CASE CAN
013E    ;                         ONLY OCCUR IF BUFFER OVERRUN IS NON-FATAL AND BUFFER 0
013E    ;                         WAS RELEASED).
013E    ;         1               INDICATES BUFFER(S) SUCCESSFULLY RELEASED
013E    ;
013E    ; SIDE EFFECTS:
013E    ;
013E    ; VARIOUS FIELDS IN THE IBUF ARRAY ARE MODIFIED
013E    ;
013E    ;--
013E
```

```
                    005C  013E           .ENTRY  LPA$RLSBUF,^M<R2,R3,R4,R6>
                          0140
              52     D4   0140           CLRL    R2                              ; RETURN CODE
         53   6C     9A   0142           MOVZBL  (AP),R3                         ; GET # OF ARGUMENTS
      56   04 AC     D0   0145           MOVL    4(AP),R6                        ; GET ADDRESS OF IBUF
      54   0C AC     DE   0149           MOVAL   12(AP),R4                       ; POINT TO FIRST BUFFER INDEX ARG
         53   03     C2   014D           SUBL    #3,R3                           ; GET # OF BUFFERS TO RELEASE
         07   53     D1   0150           CMPL    R3,#7                           ; COMPARE WITH MAXIMUM
              3F     1A   0153           BGTRU   60$                             ; TOO MANY OR TOO FEW ARGS SUPPLIED
                          0155
                          0155     20$:  ; RELEASE NEXT BUFFER
         50   94     D0   0155           MOVL    @(R4)+,R0                       ; GET INDEX
              10     12   0158           BNEQ    25$                             ; NOT BUFFER 0
                          015A
                          015A           ; HAVE BUFFER 0.  IS OVERRUN NOT FATAL?
  0B 20 A6    17     E1   015A           BBC     #LA$V_BFROVRN,IBF$L_CMDTBL(R6),25$      ; BR. IF NO
                          015F
                          015F           ; BUFFER OVERRUN IS NOT FATAL AND THIS IS BUFFER 0.
     4C A6    10     A8   015F           BISW    #FLG_M_BFRORLSD,IBF$W_FLAGS(R6) ; SET BUFFER 0 RELEASED FLAG
  27 4C A6    05     E5   0163           BBCC    #FLG_V_DBFROVRN,IBF$W_FLAGS(R6),50$  ; BR. IF NO DBL BFR OVRN
              2A     11   0168           BRB     60$                             ; DOUBLE BUFFER OVERRUN OCCURRED
                          016A
                          016A     25$:  ; VERIFY BUFFER # IS WITHIN RANGE
         03   00     ED   016A           CMPZV   #0,#3,-                         ; COMPARE WITH HIGHEST BUFFER # ALLOWED
      50   22 A6          016D                   IBF$L_CMDTBL+CMT$B_VBFRMASK(R6),R0
              22     1F   0170           BLSSU   60$                             ; SUPPLIED INDEX IS TOO HIGH - ERROR
   51    68 A640     7E   0172           MOVAQ   IBF$Q_BFRLNKS(R6)[R0],R1; GET ADDRESS OF BUFFER LINK
                          0177
                          0177           ; SHOULD WE RELEASE THIS BUFFER ONTO DEVICE QUEUE OR ONTO
                          0177           ; INUSE QUEUE AND LOAD USW
  0F 4C A6    00     E2   0177           BBSS    #FLG_V_USWSET,IBF$W_FLAGS(R6),30$  ; BR. IF USW IS ALREADY SET
  0A 48 A6    0E     E0   017C           BBS     #14,IBF$W_USW(R6),30$   ; BR. IF REQUEST IS BEING STOPPED
                          0181
                          0181           ; INSERT ONTO TAIL OF INUSE QUEUE AND LOAD USW
      64 B6   61     0E   0181           INSQUE  (R1),@IBF$L_INUQBL(R6)  ; INSERT ONTO TAIL OF INUSE QUEUE
      49 A6   50     90   0185           MOVB    R0,IBF$W_USW+1(R6)      ; LOAD NEXT BFR INDEX AND CLEAR DONE BIT
              04     11   0189           BRB     50$
                          018B
                          018B     30$:  ; INSERT BUFFER ONTO TAIL OF DEVICE QUEUE
      5C B6   61     0E   018B           INSQUE  (R1),@IBF$L_DEVQBL(R6)
                          018F
           C3 53     F4   018F     50$:  SOBGEQ  R3,20$                          ; DO NEXT ONE
              52     D6   0192           INCL    R2                              ; INDICATE SUCCESS
                          0194
      50   08 AC     D0   0194     60$:  MOVL    8(AP),R0                        ; GET ADDRESS OF IND
              03     13   0198           BEQL    70$                             ; DEFAULTED
         60   52     D0   019A           MOVL    R2,(R0)                         ; STORE RETURN STATUS
                     04   019D     70$:  RET
```

```
                          019E          .SBTTL  LPA$$BFRAST - BUFFER FULL/EMPTY AST HANDLER
                          019E    ;++
                          019E    ; FUNCTIONAL DESCRIPTION:
                          019E    ;
                          019E    ;       THIS ROUTINE IS THE AST HANDLER FOR NORMAL BUFFER FULL OR
                          019E    ;       EMPTY ASTS.  PRIMARILY IT DOES THE FOLLOWING:
                          019E    ;               1)  REMOVE A BUFFER FROM HEAD OF DEVICE QUEUE
                          019E    ;               2)  LOAD THAT BUFFER'S INDEX INTO THE USW
                          019E    ;               3)  INSERT THAT BUFFER INTO TAIL OF INUSE QUEUE
                          019E    ;               4)  REMOVE A BUFFER FROM HEAD OF INUSE QUEUE
                          019E    ;               5)  INSERT THAT BUFFER INTO TAIL OF USER QUEUE
                          019E    ;
                          019E    ; CALLING SEQUENCE:
                          019E    ;
                          019E    ;       CALLS/G
                          019E    ;
                          019E    ; INPUT PARAMETERS:
                          019E    ;
                          019E    ;       IBUF(AP)        ADDRESS OF IBUF ARRAY (AST PARAMETER)
                          019E    ;       8(AP)           SAVED R0 (UNUSED)
                          019E    ;       12(AP)          SAVED R1 (UNUSED)
                          019E    ;       16(AP)          SAVED PC (UNUSED)
                          019E    ;       20(AP)          SAVED PSL (UNUSED)
                          019E    ;
                          019E    ; IMPLICIT INPUTS:
                          019E    ;
                          019E    ;       VARIOUS FIELDS IN THE IBUF ARRAY
                          019E    ;
                          019E    ; OUTPUT PARAMETERS:
                          019E    ;
                          019E    ;       NONE
                          019E    ;
                          019E    ; IMPLICIT OUTPUTS:
                          019E    ;
                          019E    ;       NONE
                          019E    ;
                          019E    ; COMPLETION CODES:
                          019E    ;
                          019E    ;       NONE
                          019E    ;
                          019E    ; SIDE EFFECTS:
                          019E    ;
                          019E    ;       VARIOUS FIELDS IN THE IBUF ARRAY ARE MODIFIED
                          019E    ;
                          019E    ;--
                          019E
                    0040  019E          .ENTRY  LPA$$BFRAST,^M<R6>
                          01A0
     56   04 AC      D0   01A0          MOVL    4(AP),R6                        ; GET ADDRESS OF IBUF
                          01A4
  27 48 A6     0E    E0   01A4          BBS     #14,IBF$W_USW(R6),40$                   ; BR. IF STOP BIT SET IN USW
  0B 4C A6     02    E1   01A9          BBC     #FLG_V_CNTBFRS,IBF$W_FLAGS(R6),10$ ; BR. IF NOT COUNTING BUFFERS
     07 18 A6        F5   01AE          SOBGTR  IBF$L_NBUF(R6),10$              ; BR. IF WE HAVEN'T FINISHED NBUF BFRS
                          01B2
                          01B2          ; NBUF BUFFERS HAVE BEEN FILLED OR EMPTIED.  STOP SWEEP
  49 A6      40 8F   90   01B2          MOVB    #^X40,IBF$W_USW+1(R6)           ; SET STOP BIT IN USW
              17     11   01B7          BRB     40$
```

```
                        01B9    599
                        01B9    600 10$:    ; REMOVE BFR FROM DEV. Q, LOAD INTO USW, AND INSERT INTO INUSE Q.
     51   58 B6    0F   01B9    601         REMQUE  @IBF$L_DEVQFL(R6),R1     ; REMOVE FROM HEAD OF DEVICE QUEUE
             06    1C   01BD    602         BVC     20$                      ; HAVE ONE
     4C A6   01    AA   01BF    603         BICW    #FLG_M_USWSET,IBF$W_FLAGS(R6)  ; NONE IN Q. CLEAR BIT FOR RLSBUF
             0B    11   01C3    604         BRB     40$
           0073    30   01C5    605 20$:    BSBW    CVTADINDX                ; CONVERT ADDRESS IN R1 TO INDEX IN R0
     49 A6   50    90   01C8    606         MOVB    R0,IBF$W_USW+1(R6)       ; LOAD USW (THIS ALSO CLEARS DONE BIT)
     64 B6   61    0E   01CC    607         INSQUE  (R1),@IBF$L_INUQBL(R6)   ; INSERT IN TAIL OF INUSE QUEUE
                        01D0    608
                        01D0    609 40$:    ; IF THIS IS THE FIRST AST (REQUEST STARTED) DON'T PUT A BUFFER
                        01D0    610         ; ON THE USER QUEUE
  03 4C A6   01    E3   01D0    611         BBCS    #FLG_V_SWPSTRTD,IBF$W_FLAGS(R6),80$  ; BR. IF FIRST TIME THROUGH
                        01D5    612
                        01D5    613         ; NOW REMOVE FROM HEAD OF INUSE QUEUE, INSERT ON TAIL OF USER QUEUE
                        01D5    614         ; AND CALL USER'S COMPLETION ROUTINE IF ONE WAS SPECIFIED.
           006F    30   01D5    615         BSBW    GIVUSRBFR                ; DOES IT ALL
                        01D8    616
                   04   01D8    617 80$:    RET
```

```
                         01D9          .SBTTL  LPA$$OVRAST - BFR OVER/UNDERRUN AST HANDLER
                         01D9   ;++
                         01D9   ; FUNCTIONAL DESCRIPTION:
                         01D9   ;
                         01D9   ;      THIS ROUTINE IS THE AST HANDLER FOR BUFFER OVER/UNDERRUN ASTS.
                         01D9   ;      PRIMARILY, IT DOES THE FOLLOWING:
                         01D9   ;              1)  REMOVE A BUFFER FROM HEAD OF INUSE QUEUE
                         01D9   ;              2)  INSERT THAT BUFFER INTO TAIL OF USER QUEUE
                         01D9   ;              3)  IF BUFFER 0 HAS BEEN RELEASED, THEN IT IS INSERTED INTO
                         01D9   ;                  THE HEAD (!) OF THE INUSE QUEUE.  OTHERWISE, THE
                         01D9   ;                  DOUBLE BUFFER OVERRUN BIT IS SET.
                         01D9   ;
                         01D9   ; CALLING SEQUENCE:
                         01D9   ;
                         01D9   ;      CALLS/G
                         01D9   ;
                         01D9   ; INPUT PARAMETERS:
                         01D9   ;
                         01D9   ;      IBUF(AP)        ADDRESS OF IBUF ARRAY (AST PARAMETER)
                         01D9   ;      8(AP)           SAVED R0 (UNUSED)
                         01D9   ;      12(AP)          SAVED R1 (UNUSED)
                         C'D9   ;      16(AP)          SAVED PC (UNUSED)
                         01D9   ;      20(AP)          SAVED PSL (UNUSED)
                         01D9   ;
                         01D9   ; IMPLICIT INPUTS:
                         01D9   ;
                         01D9   ;      VARIOUS FIELDS IN THE IBUF ARRAY
                         01D9   ;
                         01D9   ; OUTPUT PARAMETERS:
                         01D9   ;
                         01D9   ;      NONE
                         01D9   ;
                         01D9   ; IMPLICIT CUTPUTS:
                         01D9   ;
                         01D9   ;      NONE
                         01D9   ;
                         01D9   ; COMPLETION CODES:
                         01D9   ;
                         01D9   ;      NONE
                         01D9   ;
                         01D9   ; SIDE EFFECTS:
                         01D9   ;
                         01D9   ;      VARIOUS FIELDS IN THE IBUF ARRAY ARE MODIFIED
                         01D9   ;--
                         01D9
                   0040  01D9          .ENTRY  LPA$$OVRAST,^M<R6>
                         01DB
   56    04 AC       D0  01DB          MOVL    4(AP),R6                ; GET ADDRESS OF IBUF
                         01DF
                         01DF          ; REMOVE HEAD OF INUSE QUEUE, INSERT ON TAIL OF USER QUEUE,
                         01DF          ; AND CALL USER'S COMPLETION ROUTINE IF ONE WAS SPECIFIED.
              0065   30  01DF          BSBW    GIVUSRBFR               ; DOES IT ALL
                         01E2
                         01E2          ; MAKE SURE BUFFER 0 HAS BEEN RELEASED.
06 4C A6    04       E4  01E2          BBSC    #FLG_V_BFRORLSD,IBF$W_FLAGS(R6),10$     ; BR. IF IT HAS BEEN
   4C A6    20       A8  01E7          BISW    #FLG_M_DBFROVRN,IBF$W_FLAGS(R6)         ; DBL BFR OVRN OCCURRED
            05       11  01EB          BRB     20$
```

```
                            01ED     676
                            01ED     677 10$:     ; INSERT BUFFER 0 INTO HEAD OF INUSE QUEUE
60 A6    68 A6      0E      01ED     678          INSQUE  IBF$Q_BFRLNKS(R6),IBF$L_INUQFL(R6)
                            01F2     679
09 4C A6     02     E1      01F2     680 20$:     BBC     #FLG_V_CNTBFRS,IBF$W_FLAGS(R6),30$  ; BR. IF NOT COUNTING BFRS
      05 18 A6      F5      01F7     681          SOBGTR  IBF$L_NBUF(R6),30$           ; BR. IF WE HAVEN'T FINISHED NBUF BFRS
49 A6    40 8F      90      01FB     682          MOVB    #^X40,IBF$W_USW+1(R6)       ; SET STOP BIT IN USW
                            0200     683
                    04      0200     684 30$:     RET
```

```
                              0201        .SBTTL  LPA$$CMPLTAST - QIO COMPLETE AST HANDLER
                              0201  ;++
                              0201  ; FUNCTIONAL DESCRIPTION:
                              0201  ;
                              0201  ;       THIS ROUTINE IS THE QIO COMPLETE AST HANDLER.  PRIMARILY, IT DOES
                              0201  ;       THE FOLLOWING:
                              0201  ;       1)  UNDER CERTAIN CONDITIONS THERE IS AN EXTRA BUFFER TO BE PLACED
                              0201  ;           ON THE USER QUEUE.  IF THERE IS, A BUFFER IS REMOVED FROM THE
                              0201  ;           HEAD OF THE INUSE QUEUE AND INSERTED ON THE TAIL OF THE USER QUEUE.
                              0201  ;       2)  IF THE USER SPECIFIED A COMPLETION ROUTINE, IT IS CALLED.
                              0201  ;       3)  THE CHANNEL IS DEASSIGNED.
                              0201  ;
                              0201  ; CALLING SEQUENCE:
                              0201  ;
                              0201  ;       CALLS/G
                              0201  ;
                              0201  ; INPUT PARAMETERS:
                              0201  ;
                              0201  ;       IBUF(AP)        ADDRESS OF IBUF ARRAY (AST PARAMETER)
                              0201  ;       8(AP)           SAVED R0 (UNUSED)
                              0201  ;       12(AP)          SAVED R1 (UNUSED)
                              0201  ;       16(AP)          SAVED PC (UNUSED)
                              0201  ;       20(AP)          SAVED PSL (UNUSED)
                              0201  ;
                              0201  ; IMPLICIT INPUTS:
                              0201  ;
                              0201  ;       VARIOUS FIELDS IN THE IBUF ARRAY
                              0201  ;
                              0201  ; OUTPUT PARAMETERS:
                              0201  ;
                              0201  ;       NONE
                              0201  ;
                              0201  ; IMPLICIT OUTPUTS:
                              0201  ;
                              0201  ;       NONE
                              0201  ;
                              0201  ; COMPLETION CODES:
                              0201  ;
                              0201  ;       NONE
                              0201  ;
                              0201  ; SIDE EFFECTS:
                              0201  ;
                              0201  ;       VARIOUS FIELDS IN THE IBUF ARRAY ARE MODIFIED
                              0201  ;
                              0201  ;--
                              0201
                        0040  0201        .ENTRY  LPA$$CMPLTAST,^M<R6>
                              0203
      56    04 AC       D0    0203        MOVL    4(AP),R6                  ; GET ADDRESS OF IBUF
                              0207
                              0207        ; WE HAVE AN EXTRA BUFFER IF THE STATUS IS NORMAL, BUFFER OVERRUN,
                              0207        ; OR NON-FATAL ERROR COUNT EXCEEDED (SAME AS BUFFER OVERRUN).
     00'    08 A6       B1    0207        CMPW    IBF$Q_IOSB(R6),S^#SS$_NORMAL    ; NORMAL STATUS?
               16       13    020B        BEQL    20$                             ; YES
 0000'8F    08 A6       B1    020D        CMPW    IBF$Q_IOSB(R6),#SS$_DEVREQERR   ; REQUEST ERROR?
               11       12    0213        BNEQ    40$                             ; NO, SO IT CAN'T BE EITHER OTHER ERROR
   A3 8F    0D A6       91    0215        CMPB    IBF$Q_IOSB+5(R6),#BFROVRN       ;BUFFER OVER/UNDERRUN?
```

```
              07   13  021A               BEQL    20$                        ; YES
A0 8F      0D A6   91  021C               CMPB    IBF$Q_IOSB+5(R6),#NFECEX          ; NON-FATAL ERROR COUNT EXC.?
              03   12  0221               BNEQ    40$                        ; NO
                       0223
                       0223      20$:     ; HAVE AN EXTRA BUFFER TO PLACE ON USER QUEUE.  REMOVE FROM HEAD
                       0223               ; OF INUSE QUEUE, INSERT ON TAIL OF USER QUEUE, AND CALL USER'S
                       0223               ; COMPLETION ROUTINE IF ONE WAS SPECIFIED.
            0021   30  0223               BSBW    GIVUSRBFR                  ; DOES IT ALL
                       0226
                       0226      40$:     ; NOW CALL USER'S COMPLETION ROUTINE (POSSIBLY FOR THE SECOND TIME)
                       0226               ; IF ONE WAS SPECIFIED.
    50     10 A6   D0  0226               MOVL    IBF$L_COMPLADDR(R6),R0     ; GET ADDRESS OF ROUTINE
              03   13  022A               BEQL    50$                        ; NONE SPECIFIED
       60     00   FB  022C               CALLS   #0,(R0)                    ; CALL IT
                       022F
                       022F      50$:     ; DEASSIGN CHANNEL
                       022F               $DASSGN_S       IBF$W_CHAN(R6)
                       023A
                   04  023A               RET
```

```
                        023B                .SBTTL  CVTADINDX - CONVERT ADDRESS TO INDEX
                        023B      ;++
                        023B      ; FUNCTIONAL DESCRIPTION:
                        023B      ;
                        023B      ;         THIS ROUTINE CONVERTS A BUFFER LINK ADDRESS TO A BUFFER INDEX
                        023B      ;
                        023B      ; CALLING SEQUENCE:
                        023B      ;
                        023B      ;         BSBW/B
                        023B      ;
                        023B      ; INPUT PARAMETERS:
                        023B      ;
                        023B      ;         R1      ADDRESS OF BUFFER LINK
                        023B      ;         R6      ADDRESS OF IBUF ARRAY
                        023B      ;
                        023B      ; IMPLICIT INPUTS:
                        023B      ;
                        023B      ;         NONE
                        023B      ;
                        023B      ; OUTPUT PARAMETERS:
                        023B      ;
                        023B      ;         R0      CONTAINS BUFFER INDEX
                        023B      ;
                        023B      ; IMPLICIT OUTPUTS:
                        023B      ;
                        023B      ;         NONE
                        023B      ;
                        023B      ; COMPLETION CODES:
                        023B      ;
                        023B      ;         NONE
                        023B      ;
                        023B      ; SIDE EFFECTS:
                        023B      ;
                        023B      ;         NONE
                        023B      ;
                        023B      ;--
                        023B
                        023B      CVTADINDX:
     50   68 A6    9E   023B              MOVAB   IBF$Q_BFRLNKS(R6),R0    ; GET ADDRESS OF START OF BUFFER LINKS
  50   51    50    C3   023F              SUBL3   R0,R1,R0                ; SUBTRACT THAT FROM GIVEN BFR LINK ADDR
       50    08    C6   0243              DIVL    #8,R0                   ; DIVIDE BY 8 TO CONVERT TO INDEX
                   05   0246              RSB
```

```
                           0247          .SBTTL  GIVUSRBFR - GIVE USER BUFFER
                           0247  ;++
                           0247  ; FUNCTIONAL DESCRIPTION:
                           0247  ;
                           0247  ;       THIS ROUTINE REMOVES A BUFFER FROM THE HEAD OF THE INUSE QUEUE,
                           0247  ;       INSERTS IT ON THE TAIL OF THE USER QUEUE, AND CALLS THE USER'S
                           0247  ;       COMPLETION ROUTINE, IF ONE WAS SPECIFIED.
                           0247  ;
                           0247  ; CALLING SEQUENCE:
                           0247  ;
                           0247  ;       BSBW/B
                           0247  ;
                           0247  ; INPUT PARAMETERS:
                           0247  ;
                           0247  ;       R6      ADDRESS OF IBUF ARRAY
                           0247  ;
                           0247  ; IMPLICIT INPUTS:
                           0247  ;
                           0247  ;       VARIOUS FIELDS IN THE IBUF ARRAY
                           0247  ;
                           0247  ; OUTPUT PARAMETERS:
                           0247  ;
                           0247  ;       NONE
                           0247  ;
                           0247  ; IMPLICIT OUTPUTS:
                           0247  ;
                           0247  ;       NONE
                           0247  ;
                           0247  ; COMPLETION CODES:
                           0247  ;
                           0247  ;       NONE
                           0247  ;
                           0247  ; SIDE EFFECTS:
                           0247  ;
                           0247  ;       R0,R1 ARE NOT PRESERVED
                           0247  ;       VARIOUS FIELDS IN THE IBUF ARRAY ARE MODIFIED
                           0247  ;
                           0247  ;--
                           0247
                           0247  GIVUSRBFR:
                           0247          ; REMOVE BUFFER FROM HEAD OF INUSE QUEUE
50    60 B6        0F      0247          REMQUE  @IBF$L_INUQFL(R6),R0    ; R0 CONTAINS ADDRESS
           0D      1D      024B          BVS     20$                     ; QUEUE IS EMPTY - SHOULD NEVER HAPPEN!
                           024D
                           024D          ; INSERT BUFFER ON TAIL OF USER QUEUE
54 B6      60      0E      024D          INSQUE  (R0),@IBF$L_USRQBL(R6)
                           0251
                           0251          ; CALL USER'S COMPLETION ROUTINE IF ONE WAS SPECIFIED
50    10 A6        D0      0251          MOVL    IBF$L_COMPLADDR(R6),R0  ; GET ADDRESS OF COMPLETION ROUTINE
           03      13      0255          BEQL    20$                     ; NONE SPECIFIED
   60      00      FB      0257          CALLS   #0,(R0)                 ; CALL IT
                   05      025A  20$:    RSB
                           025B
                           025B
                           025B
                           025B  LPA$BFEND::                             ; END OF BUFFER ROUTINES (MAY BE USED
                           025B                                          ; TO LOCK THESE ROUTINES INTO THE
```

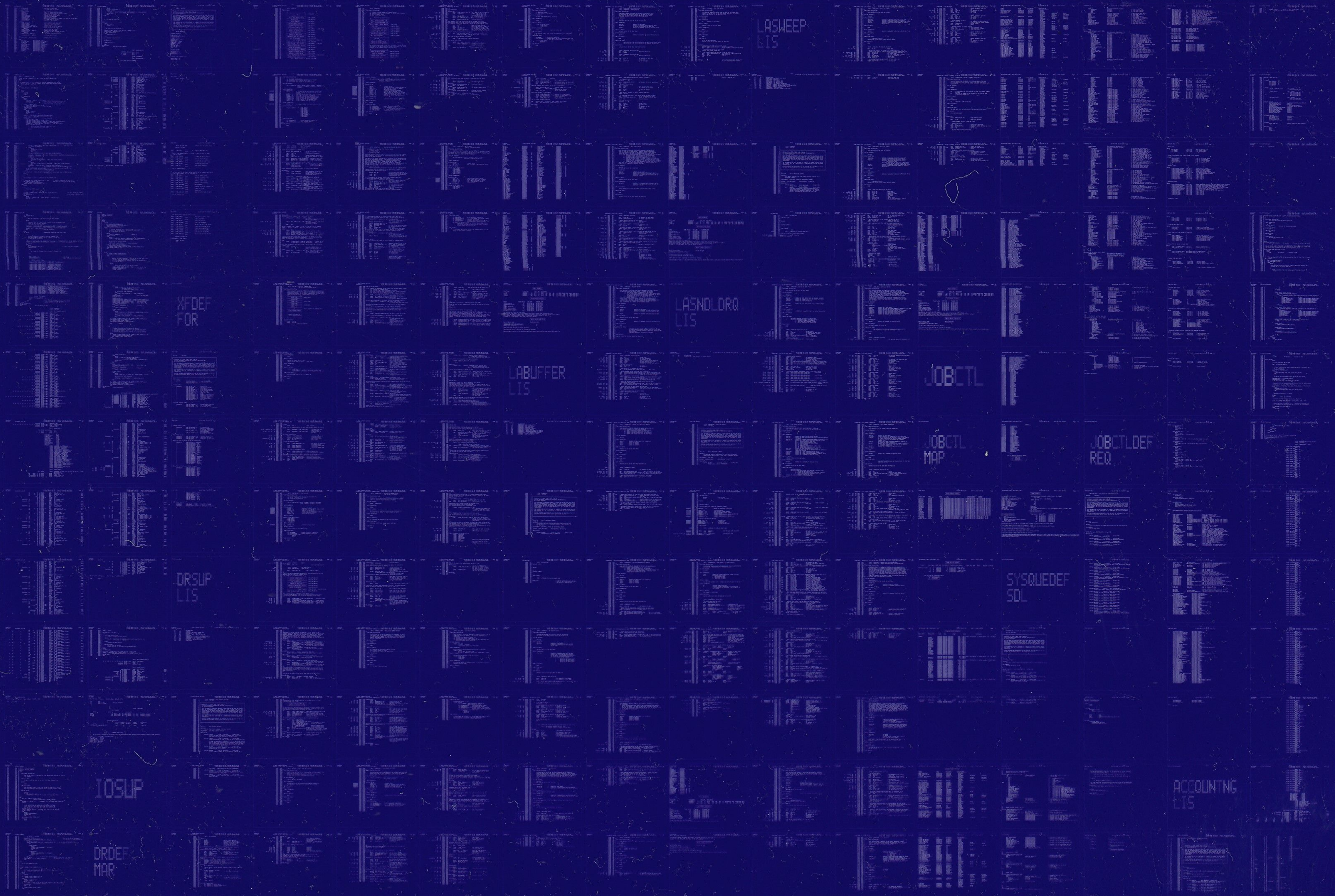
```
                     025B   863                                      ; PROCESS'S WORKING SET).
                     025B   864
                     025B   865
                     025B   866        .END
```

```
BFROVRN         = 000000A3                        LPA$IGTBUF         00000058 RG     02
CLKEVFLG        = 00000017                        LPA$INXTBF         00000086 RG     02
CMT$B_EVMRKN      00000021                        LPA$IWTBUF         000000E4 RG     02
CMT$B_ICHN        0000001A                        LPA$RLSBUF         0000013E RG     02
CMT$B_INC         0000001B                        LPA$RMVBUF         0000008D RG     02
CMT$B_STWRDN      00000020                        NFECEX           = 000000A0
CMT$B_VBFRMASK    00000002                        SIZ...           = 00000001
CMT$L_BFRADDR     0000000C                        SS$_DEVREQERR      ********   X    02
CMT$L_BFRLEN      00000008                        SS$_NORMAL         ********   X    02
CMT$L_RCLADDR     00000014                        SYS$CLREF          ********  GX    02
CMT$L_RCLLEN      00000010                        SYS$DASSGN         ********  GX    02
CMT$L_USWADDR     00000004                        SYS$SETAST         ********  GX    02
CMT$W_DELAY       00000018                        SYS$WAITFR         ********  GX    02
CMT$W_DWELL       0000001E
CMT$W_EVMRKM      00000024
CMT$W_MODE        00000000
CMT$W_NCHN        0000001C
CMT$W_STWRDM      00000022
COMMON            00000091 R       02
CVTADINDX         0000023B R       02
DEFEVFLG        = 00000016
FLG_M_BFRORLSD  = 00000010
FLG_M_DBFROVRN  = 00000020
FLG_M_USWSET    = 00000001
FLG_V_BFRORLSD  = 00000004
FLG_V_CNTBFRS   = 00000002
FLG_V_DBFROVRN  = 00000005
FLG_V_SWPS'RTD  = 00000001
FLG_V_USWSET    = 00000000
FOLCOWQ           00000043 R       02
GIVUSRBFR         00000247 R       02
IBF$B_EFN         0000004E
IBF$L_CMDTBL      00000020
IBF$L_COMPLADDR   00000010
IBF$L_DEVQBL      0000005C
IBF$L_DEVQFL      00000058
IBF$L_INUQBL      00000064
IBF$L_INUQFL      00000060
IBF$L_LAMSKB      0000001C
IBF$L_LBUF        00000014
IBF$L_NBUF        00000018
IBF$L_USRQBL      00000054
IBF$L_USRQFL      00000050
IBF$Q_BFRLNKS     00000068
IBF$Q_IOSB        00000008
IBF$Q_IOST        00000000
IBF$W_CHAN        0000004A
IBF$W_FLAGS       0000004C
IBF$W_USW         00000048
INITCODE        = 00001234
LA$V_BFROVRN    = 00000017
LPA$$BFRAST       0000019E RG      02
LPA$$CMPLTAST     00000201 RG      02
LPA$$OVRAST       000001D9 RG      02
LPA$BFEND         0000025B RG      02
LPA$BFSTART       00000000 RG      02
LPA$IBFSTS        00000000 RG      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
$ABS$                           000000A8  (  168.)  01 (  1.)  NOPIC   USR   CON   ABS   LCL NOSHR   EXE   RD     WRT NOVEC BYTE
_LPA$CODE                       0000025B  (  603.)  02 (  2.)    PIC   USR   CON   REL   LCL   SHR   EXE   RD   NOWRT NOVEC LONG

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

Phase                    Page faults    CPU Time       Elapsed Time
-----                    ----------    --------       ------------
Initialization                   34    00:00:00.06    00:00:00.93
Command processing              139    00:00:00.63    00:00:02.32
Pass 1                          155    00:00:03.25    00:00:07.09
Symbol table sort                 0    00:00:00.15    00:00:00.27
Pass 2                          150    00:00:01.86    00:00:03.63
Symbol table output               7    00:00:00.07    00:00:00.28
Psect synopsis output             3    00:00:00.02    00:00:00.02
Cross-reference output            0    00:00:00.00    00:00:00.00
Assembler run totals            490    00:00:06.06    00:00:14.57

The working set limit was 1200 pages.
18381 bytes (36 pages) of virtual memory were used to buffer the intermediate code.
There were 10 pages of symbol table space allocated to hold 111 non-local and 35 local symbols.
995 source lines were read in Pass 1, producing 37 object records in Pass 2.
15 pages of virtual memory were used to define 14 macros.

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

Macro library name                          Macros defined
------------------                          --------------
_$255$DUA28:[SYSLIB]STARLET.MLB;2                    9

121 GETS were required to define 9 macros.

There were no errors, warnings or information messages.

MACRO/DISABLE=TRACE/LIS=LIS$:LABUFFER/OBJ=OBJ$:LABUFFER MSRC$:LADEF/UPDATE=(ENH$:LADEF)+MSRC$:LABUFFER/UPDATE=(ENH$:LABUFFER)
```

0190 AH-BT13A-SE
VAX/VMS V4.0

DIGITAL EQUIPMENT CORPORATION
CONFIDENTIAL AND PROPRIETARY



LASWEEP
LIS

XFDEF
FOR

LASNDLDRQ
LIS

LABUFFER
LIS

JOBCTL

JOBCTL
MAP

JOBCTLDEF
REQ

DRSUP
LIS

SYSQUEDEF
SDL

IOSUP

ACCOUNTNG
LIS

DRDEF
MAR