```
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
```

```
  GGGGGGGG  EEEEEEEEEE  TTTTTTTTTT  RRRRRRRR    EEEEEEEEEE    QQQQQQ
  GGGGGGGG  EEEEEEEEEE  TTTTTTTTTT  RRRRRRRR    EEEEEEEEEE    QQQQQQ
GG          EE              TT      RR      RR  EE          QQ      QQ
GG          EE              TT      RR      RR  EE          QQ      QQ
GG          EE              TT      RR      RR  EE          QQ      QQ
GG          EE              TT      RR      RR  EE          QQ      QQ
GG          EEEEEEEE        TT      RRRRRRRR    EEEEEEEE    QQ      QQ
GG          EEEEEEEE        TT      RRRRRRRR    EEEEEEEE    QQ      QQ
GG  GGGGGG  EE              TT      RR  RR      EE          QQ  QQ  QQ
GG  GGGGGG  EE              TT      RR  RR      EE          QQ  QQ  QQ
GG      GG  EE              TT      RR    RR    EE          QQ    QQ        ....
GG      GG  EE              TT      RR    RR    EE          QQ    QQ        ....
  GGGGGG    EEEEEEEEEE      TT      RR      RR  EEEEEEEEEE    QQQQ  QQ      ....
  GGGGGG    EEEEEEEEEE      TT      RR      RR  EEEEEEEEEE    QQQQ  QQ      ....

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
```

```
0
0 MODULE GETREQ (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 !
1 !       This routine gets the next I/O request from the ACP queue.
1 !       If no requests are queued, it hibernates.  When all its work is
1 !       complete, it deletes itself.
1 !
1 ! ENVIRONMENT:
1 !
1 !       Starlet operating system, including privileged system services
1 !       and internal exec routines. This routine must be called
1 !       in kernel mode.
1 !
1 !--
1
1 !
1 !
1 ! AUTHOR:  D. H. GILLESPIE,       CREATION DATE:  11-MAY-1977  17:26
1 !
1 ! MODIFIED BY:
1 !
1 !       V03-001 MMD0161         Meg Dumont,      26-Apr-1983  9:36
1 !               Add references and setup of the HDR4 label.
1 !
1 !       V02-006 DMW00012        David Michael Walp      14-Mar-1981
```

```
1 !                 Added routine GET_CCB to find the address of the channel
1 !                 control block.  Use the routine to find the CCB.
1 !
1 !         V02-005 REFORMAT        Maria del C. Nasr       30-Jun-1980
1 !
1 !         A0004   MCN0003         Maria del C. Nasr       25-Sep-79   16:37
1 !                 Add HDR3 processing
1 !
1 !         A0003   SPR24947        Maria del C. Nasr       04-Sep-79   10:40
1 !                 Fixed bug to allow only one file to be accessed at any
1 !                 one time on a magtape.
1 !
1 !**
1
1 LIBRARY 'SYS$LIBRARY:LIB.L32';
1
1 REQUIRE 'SRC$:MTADEF.B32';
1
1 FORWARD ROUTINE
1     GET_CCB,                                          ! get the channel control block
1     GET_REQ     : L$GET_REQ,                          ! exec mode get request
1     GET_REQUEST : COMMON_CALL;                        ! kernel mode get request
1
1 EXTERNAL ROUTINE
1     DEALLOCATE,                                       ! deallocate system space
1     CHECK_MAIL  : COMMON_CALL,                        ! check for mail from operator
1     LOCK_IODB,                                        ! lock i/o database mutex
1     SYS$HIBER   : ADDRESSING_MODE (ABSOLUTE),
1     UNLOCK_IODB;                                      ! unlock i/o databas mutex
1
1 EXTERNAL
1
1     DISK_UCB    : REF BBLOCK,                         ! UCB of device sys$disk
1     IO_CHANNEL,                                       ! i/o channel number
1     MAIL_CHANNEL;                                     ! mailbox channel number
1
```

```
1 ROUTINE GET_REQUEST : COMMON_CALL =
1
1 !++
1 !
1 ! FUNCTIONAL DESCRIPTION:
1 !       This routine gets the next i/o request from the ACP queue.
1 !       If no more requests, it checks if there are any volumes being
1 !       serviced by the ACP.  If there are none, prepare to delete itself.
1 !
1 ! CALLING SEQUENCE:
1 !       GET_REQUEST (), called in kernel_mode
1 !
1 ! INPUT PARAMETERS:
1 !       none
1 !
1 ! IMPLICIT INPUTS
1 !       QUEUE_HEAD: address of ACP queue block
1 !       IO_CHANNEL: i/o channel number
1 !
1 ! OUTPUT PARAMETERS:
1 !       none
1 !
1 ! IMPLICIT OUTPUTS:
1 !       CURRENT_UCB: address of UCB of request
1 !       CURRENT_VCB: address of vcb of request
1 !       CURRENT_WCB: window of file if accessed
1 !       HDR1:        address of hdr1 in virtual page
1 !       HDR2:        address of hdr2 in virtual page
1 !       HDR3:        address of hdr3 in virtual page
1 !       HDR4:        address of hdr4 in virtual page
1 !
1 ! ROUTINE VALUE:
1 !       address of request i/o packet
1 !
1 ! SIDE EFFECTS:
1 !       I/O channel assigned to device of request
1 !       Result string length set to zero and result string cleared
1 !       Disable write back of channel window
1 !
1 !--
1
2     BEGIN
2
2     EXTERNAL REGISTER
2         COMMON_REG;
2
2     ! This register forces REMQUE to be executed in one instruction
2     !
2
2     REGISTER
2         QUEUE_POINTER   : REF BBLOCK;
2
2     EXTERNAL
2         CURRENT_UCB     : REF BBLOCK,           ! address of current UCB
2         CURRENT_WCB     : REF BBLOCK,           ! address of file window
2         HDR1            : REF BBLOCK,           ! address of hdr1 label
2         HDR2            : REF BBLOCK,           ! address of hdr2 label
```

```
	HDR3		: REF BBLOCK,		! address of hdr3 label
	HDR4		: REF BBLOCK,		! address of hdr4 label
	IOC$GL_AQBLIST	: REF BBLOCK ADDRESSING_MODE (ABSOLUTE),
	QUEUE_HEAD	: REF BBLOCK,		! ACP queue list head
	SCH$GL_CURPCB	: REF BBLOCK ADDRESSING_MODE (ABSOLUTE);

LOCAL
	CCB		: REF BBLOCK,		! pointer to ccb of i/o channel
	PACKET		: REF BBLOCK;		! address of new i/o packet

! Attempt to dequeue a packet.
!

WHILE 1
DO
	BEGIN
	QUEUE_POINTER = .QUEUE_HEAD;

	IF NOT REMQUE(.QUEUE_POINTER[AQB$L_ACPQFL], PACKET)
	THEN

	    ! check that structures are valid
	    !
	    BEGIN

	    LOCAL
		VPAGE	: REF BBLOCK;

	    IF .PACKET[IRP$B_TYPE] NEQ DYN$C_IRP
	    THEN
		BUG_CHECK(NOTIRPAQB);

	    CURRENT_UCB = .PACKET[IRP$L_UCB];

	    IF .CURRENT_UCB[UCB$B_TYPE] NEQ DYN$C_UCB
	    THEN
		BUG_CHECK(NOTUCBIRP);

	    CURRENT_VCB = .CURRENT_UCB[UCB$L_VCB];

	    IF .CURRENT_VCB[VCB$B_TYPE] NEQ DYN$C_VCB
	    THEN
		BUG_CHECK(NOTVCBUCB);

	    ! If the virtual page forward link in the VCB does not point to
	    ! itself, it means there is a page allocated to this volume.  Get
	    ! the address, verify the type, and set the header pointers.
	    !
	    IF .CURRENT_VCB[VCB$L_VPFL] NEQ CURRENT_VCB[VCB$L_VPFL]
	    THEN
		BEGIN
		VPAGE = .CURRENT_VCB[VCB$L_VPFL];

		IF .VPAGE[VVP$B_TYPE] NEQ VVP_TYPE
		THEN
		    BUG_CHECK(NOTVVPVCB);

```

```
5                   HDR1 = VPAGE[VVP$T_HDR1];
5                   HDR2 = VPAGE[VVP$T_HDR2];
5                   HDR3 = VPAGE[VVP$T_HDR3];
5                   HDR4 = VPAGE[VVP$T_HDR4];
4                   END;
4
4               ! If the volume is blocked for rewind or volume mount, put all
4               ! requests other than cancel on the stalled i/o queue.
4               !
4
5               IF NOT (.CURRENT_VCB[VCB$V_WAIREWIND] OR .CURRENT_VCB[VCB$V_WAIMOUVOL])
4               THEN
4                   EXITLOOP;                           ! volume is not blocked
4
4               IF .PACKET[IRP$V_FCODE] EQL IO$_ACPCONTROL
4                   AND
4                   NOT .PACKET[IRP$V_VIRTUAL]
4               THEN
4                   EXITLOOP;                           ! let cancel thru
4
4               ! insert in stalled i/o queue
4               !
4               INSQUE(.PACKET, .VPAGE[VVP$L_STALLIOBL]);
4               END
3           ELSE
3
3               ! If the REMQUE failed and the mount count in the AQB is zero, this
3               ! ACP is potentially idle.  Interlock the i/o database and check
3               ! the queue and the count again.  If the ACP is no longer idle,
3               ! proceed as if nothing had happended. If it still is, unhook the
3               ! AQB from the system AQB list.  Once unhooked, the ACP can no
3               ! longer be found by anyone.  Change the process uic so that a new
3               ! ACP will be successfully created by mount.  Return to exec mode
3               ! GET_REQUEST to wait for outstanding i/o and delete the process.
3               !
4               BEGIN
4
4               IF .QUEUE_POINTER[AQB$B_MNTCNT] EQL 0
4               THEN
5                   BEGIN
5                   LOCK_IODB();                        ! lock i/o data base mutex
5
5                   IF .QUEUE_POINTER[AQB$B_MNTCNT] EQL 0
5                       AND
5                       .QUEUE_POINTER[AQB$L_ACPQFL] EQL QUEUE_POINTER[AQB$L_ACPQFL]
5                   THEN
6                       BEGIN
6
6                       LOCAL
6                           P       : REF BBLOCK;
6
6                       P = .IOC$GL_AQBLIST;
6
6                       IF .P EQL .QUEUE_POINTER
6                       THEN
6                           IOC$GL_AQBLIST = .QUEUE_POINTER[AQB$L_LINK]
6                       ELSE
```

```
7                         BEGIN

7                         UNTIL .P[AQB$L_LINK] EQL .QUEUE_POINTER
7                         DO
7                             P = .P[AQB$L_LINK];
7
7                         P[AQB$L_LINK] = .QUEUE_POINTER[AQB$L_LINK];
6                         END;
6
6                     ! Inc group component of process PCB so the create ACP in
6                     ! mount will not get a duplicate process error.
6                     !
6                     SCH$GL_CURPCB[PCB$W_GRP] = .SCH$GL_CURPCB[PCB$W_GRP] + 1;
6                     UNLOCK_IODB();
6                     END
5                 ELSE
5                     UNLOCK_IODB();
5
4                 END;                                ! end of if no more volumes mounted
4
4             RETURN 0;                               ! no packet
4
3             END;                                    ! end of if packet found
3
2         END;                                        ! end of while loop
2
2     ! The current volume may not be on the unit indicated in the users channel
2     !
3     BEGIN
3
3     LOCAL
3         UCBLST  : REF VECTOR [LONG],                ! pointer to UCBlst in RVT
3         RVT     : REF BBLOCK;                       ! relative volume table
3
3     RVT = .CURRENT_VCB[VCB$L_RVT];
3
3     IF .RVT[RVT$B_TYPE] NEQ DYN$C_RVT
3     THEN
3         BUG_CHECK(NOTRVTVCB);
3
3     UCBLST = RVT[RVT$L_UCBLST];
3     CURRENT_UCB = .UCBLST[.CURRENT_VCB[VCB$W_RVN]];
2     END;
2
2     IF .CURRENT_UCB[UCB$B_TYPE] NEQ DYN$C_UCB
2     THEN
2         BUG_CHECK(NOTUCBRVT);
2
2     CCB = GET_CCB ( .IO_CHANNEL );                  ! point to ccb
2     CCB[CCB$L_UCB] = .CURRENT_UCB;                  ! and assign it by stuffing UCB
2
2     ! Get the WCB address if there is a file open on the channel. The window
2     ! pointer will not be valid if this is a cancel io which is an ACP control
2     ! function with the virtual bit off.
2     !
2     CURRENT_WCB = .CURRENT_VCB[VCB$L_WCB];
2
```

```
! If low order bit is set, then deaccess is pending.  Ignore window.
!

IF .(PACKET[IRP$L_WIND])<0,1>
THEN
    CURRENT_WCB = 0;

! address for window is long word aligned
!

IF .(PACKET[IRP$L_WIND])<1,2> NEQ 0
THEN
    BUG_CHECK(BADWCBPT);

IF .CURRENT_WCB NEQ 0
THEN
    BEGIN

    IF .CURRENT_WCB[WCB$B_TYPE] NEQ DYN$C_WCB
    THEN
        BUG_CHECK(NOTWCBIRP);

    IF .CURRENT_WCB[WCB$V_NOTFCP]
    THEN
        BUG_CHECK(NOTFCPWCB);

    END;

! Prevent write-back of WCB.  Set result string length equal to zero.
! Clear result string buffer.
!

IF .PACKET[IRP$V_COMPLX]
THEN
    BEGIN

    LOCAL
        ABD : REF BBLOCKVECTOR [, ABD$C_LENGTH];

    ABD = .BBLOCK[.PACKET[IRP$L_SVAPTE], AIB$L_DESCRIPT];
    ABD[ABD$C_WINDOW, ABD$W_COUNT] = 0;

    IF .ABD[ABD$C_RESL, ABD$W_COUNT] GEQ 2
    THEN
        (.ABD[ABD$C_RESL, ABD$W_TEXT] + ABD[ABD$C_RESL, ABD$W_TEXT] + 1)<0,16> = 0;

    CH$FILL(0, .ABD[ABD$C_RES, ABD$W_COUNT],
        .ABD[ABD$C_RES, ABD$W_TEXT] + ABD[ABD$C_RES, ABD$W_TEXT] + 1);
    END

    ! If there is no buffer packet, the function must be an ACP control
    ! function.
    !
ELSE
    BEGIN

    IF (.PACKET[IRP$V_FCODE] GTRU IO$_LOGICAL AND .PACKET[IRP$V_FCODE] NEQ IO$_ACPCONTROL)
```

```
;  380          0762  3             OR
;  381          0763  4             (.PACKET[IRP$V_FCODE] EQL IO$_ACPCONTROL AND .PACKET[IRP$V_VIRTUAL])
;  382          0764  3         THEN
;  383          0765  3             BUG_CHECK(NOBUFPCKT);
;  384          0766  3
;  385          0767  2         END;
;  386          0768  2
;  387          0769  2     RETURN .PACKET;
;  388          0770  2
;  389          0771  1     END;                                        ! end of routine


                                                                 .TITLE  GETREQ
                                                                 .IDENT  \V04-000\

                                                                 .EXTRN  DEALLOCATE, CHECK_MAIL
                                                                 .EXTRN  LOCK_IODB, SYS$HIBER
                                                                 .EXTRN  UNLOCK_IODB, DISK_UCB
                                                                 .EXTRN  IO_CHANNEL, MAIL_CHANNEL
                                                                 .EXTRN  CURRENT_UCB, CURRENT_WCB
                                                                 .EXTRN  HDR1, HDR2, HDR3
                                                                 .EXTRN  HDR4, IOC$GL_AQBLIST
                                                                 .EXTRN  QUEUE_HEAD, SCH$GL_CURPCB
                                                                 .EXTRN  BUG$_NOTIRPAQB, BUG$_NOTUCBIRP
                                                                 .EXTRN  BUG$_NOTVCBUCB, BUG$_NOTVVPVCB
                                                                 .EXTRN  BUG$_NOTRVTVCB, BUG$_NOTUCBRVT
                                                                 .EXTRN  BUG$_BADWCBPT, BUG$_NOTWCBIRP
                                                                 .EXTRN  BUG$_NOTFCPWCB, BUG$_NOBUFPCKT

                                                                 .PSECT  $CODE$,NOWRT,2

                                          03FC 00000 GET_REQUEST:
                                                                 .WORD   Save R2,R3,R4,R5,R6,R7,R8,R9                          ; 0477
                          59      0000G  CF   9E 00002           MOVAB   CURRENT_WCB, R9                                       ;
                          58  00000000G  9F   9E 00007           MOVAB   @#IOC$GL_AQBLIST, R8                                  ;
                          57      0000G  CF   9E 0000E           MOVAB   CURRENT_UCB, R7                                       ;
                          52      0000G  CF   D0 00013 1$:       MOVL    QUEUE_HEAD, QUEUE_POINTER                             ; 0550
                          56         00  B2   CF 00018           REMQUE  @0(QUEUE_POINTER), PACKET                             ; 0552
                                         7D   1D 0001C           BVS     9$                                                    ;
                          0A         0A  A6   91 0001E           CMPB    10(PACKET), #10                                       ; 0562
                                         04   13 00022           BEQL    2$                                                    ;
                                            FEFF 00024           BUGW                                                          ; 0564
                                           0000* 00026           .WORD   <BUG$_NOTIRPAQB!4>                                    ;
                          67         1C  A6   D0 00028 2$:       MOVL    28(PACKET), CURRENT_UCB                               ; 0566
                          50             67   D0 0002C           MOVL    CURRENT_UCB, R0                                       ; 0568
                          10         0A  A0   91 0002F           CMPB    10(R0), #16                                           ;
                                         04   13 00033           BEQL    3$                                                    ;
                                            FEFF 00035           BUGW                                                          ; 0570
                                           0000* 00037           .WORD   <BUG$_NOTUCBIRP!4>                                    ;
                          50             67   D0 00039 3$:       MOVL    CURRENT_UCB, R0                                       ; 0572
                          5B         34  A0   D0 0003C           MOVL    52(R0), CURRENT_VCB                                   ;
                          11         0A  AB   91 00040           CMPB    10(CURRENT_VCB), #17                                  ; 0574
                                         04   13 00044           BEQL    4$                                                    ;
                                            FEFF 00046           BUGW                                                          ; 0576
                                           0000* 00048           .WORD   <BUG$_NOTVCBUCB!4>                                    ;
                          50         3C  AB   9E 0004A 4$:       MOVAB   60(CURRENT_VCB), R0                                   ; 0582
                          50         3C  AB   D1 0004E           CMPL    60(CURRENT_VCB), R0                                   ;
```

```
                                     28    13 00052        BEQL    6$                                    ;
                       50      3C    AB    D0 00054        MOVL    60(CURRENT_VCB), VPAGE               ; 0585
                       02      0A    A0    91 00058        CMPB    10(VPAGE), #2                        ; 0587
                                     04    13 0005C        BEQL    5$                                    ;
                                         FEFF 0005E        BUGW                                          ; 0589
                                        0000* 00060        .WORD   <BUG$_NOTVVPVCB!4>                   ;
                 0000G CF      0C    A0    9E 00062 5$:    MOVAB   12(R0), HDR1                         ; 0591
                 0000G CF      5C    A0    9E 00068        MOVAB   92(R0), HDR2                         ; 0592
                 0000G CF    00AC    C0    9E 0006E        MOVAB   172(R0), HDR3                        ; 0593
                 0000G CF    00FC    C0    9E 00075        MOVAB   252(R0), HDR4                        ; 0594
            05      0B AB            03    E0 0007C 6$:    BBS     #3, 11(CURRENT_VCB), 7$              ; 0601
            5D      0B AB            02    E1 00081        BBC     #2, 11(CURRENT_VCB), 15$             ;
  38     20 A6         06            00    ED 00086 7$:    CMPZV   #0, #6, 32(PACKET), #56              ; 0605
                                     05    12 0008C        BNEQ    8$                                    ;
            50      2A A6            04    E1 0008E        BBC     #4, 42(PACKET), 15$                  ; 0607
                  01A8 D0            66    0E 00093 8$:    INSQUE  (PACKET), @424(VPAGE)                ; 0613
                                   FF78    31 00098        BRW     1$                                    ; 0552
                                     53    D4 0009B 9$:    CLRL    R3                                    ; 0628
                               0B    A2    95 0009D        TSTB    11(QUEUE_POINTER)                    ;
                                     3E    12 000A0        BNEQ    14$                                   ;
                                     53    D6 000A2        INCL    R3                                    ;
                 0000G CF            00    FB 000A4        CALLS   #0, LOCK_IODB                        ; 0631
                       2F            53    E9 000A9        BLBC    R3, 13$                              ; 0633
                       52            62    D1 000AC        CMPL    (QUEUE_POINTER), QUEUE_POINTER       ; 0635
                                     2A    12 000AF        BNEQ    13$                                   ;
                       50            68    D0 000B1        MOVL    IOC$GL_AQBLIST, P                    ; 0642
                       52            50    D1 000B4        CMPL    P, QUEUE_POINTER                     ; 0644
                                     06    12 000B7        BNEQ    10$                                   ;
                       68      10    A2    D0 000B9        MOVL    16(QUEUE_POINTER), IOC$GL_AQBLIST    ; 0646
                                     11    11 000BD        BRB     12$                                   ;
                       52      10    A0    D1 000BF 10$:   CMPL    16(P), QUEUE_POINTER                 ; 0650
                                     06    13 000C3        BEQL    11$                                   ;
                       50      10    A0    D0 000C5        MOVL    16(P), P                             ; 0652
                                     F4    11 000C9        BRB     10$                                   ;
                    10 A0      10    A2    D0 000CB 11$:   MOVL    16(QUEUE_POINTER), 16(P)             ; 0654
                       50 00000000G  9F    D0 000D0 12$:   MOVL    @#SCH$GL_CURPCB, R0                  ; 0660
                             00BE    C0    B6 000D7        INCW    190(R0)                               ;
                 0000G CF            00    FB 000DB 13$:   CALLS   #0, UNLOCK_IODB                      ; 0664
                                   00B8    31 000E0 14$:   BRW     27$                                   ; 0668
                       50      20    AB    D0 000E3 15$:   MOVL    32(CURRENT_VCB), RVT                 ; 0682
                       0E      0A    A0    91 000E7        CMPB    10(RVT), #14                         ; 0684
                                     04    13 000EB        BEQL    16$                                   ;
                                         FEFF 000ED        BUGW                                          ; 0686
                                        0000* 000EF        .WORD   <BUG$_NOTRVTVCB!4>                   ;
                       50      44    A0    9E 000F1 16$:   MOVAB   68(R0), UCBLST                       ; 0688
                       51      0E    AB    3C 000F5        MOVZWL  14(CURRENT_VCB), R1                  ; 0689
                       67          6041    D0 000F9        MOVL    (UCBLST)[R1], CURRENT_UCB            ;
                       50            67    D0 000FD        MOVL    CURRENT_UCB, R0                      ; 0692
                       10      0A    A0    91 00100        CMPB    10(R0), #16                          ;
                                     04    13 00104        BEQL    17$                                   ;
                                         FEFF 00106        BUGW                                          ; 0694
                                        0000* 00108        .WORD   <BUG$_NOTUCBRVT!4>                   ;
                            0000G    CF    DD 0010A 17$:   PUSHL   IO_CHANNEL                           ; 0696
                 0000V CF            01    FB 0010E        CALLS   #1, GET_CCB                          ;
                       60            67    D0 00113        MOVL    CURRENT_UCB, (CCB)                   ; 0697
                       69      38    AB    D0 00116        MOVL    56(CURRENT_VCB), CURRENT_WCB         ; 0703
                       02      18    A6    E9 0011A        BLBC    24(PACKET), 18$                      ; 0708
```

```
                                          69  D4 0011E         CLRL    CURRENT_WCB                              ; 0710
                         06         18    A6  93 00120 18$:    BITB    24(PACKET), #6                           ; 0715
                                          04  13 00124         BEQL    19$                                      ;
                                            FEFF 00126         BUGW                                             ; 0717
                                           0000* 00128         .WORD   <BUG$_BADWCBPT!4>                        ;
                         50               69  D0 0012A 19$:    MOVL    CURRENT_WCB, R0                          ; 0719
                                          16  13 0012D         BEQL    21$                                      ;
                         12         0A    A0  91 0012F         CMPB    10(R0), #18                              ; 0723
                                          04  13 00133         BEQL    20$                                      ;
                                            FEFF 00135         BUGW                                             ; 0725
                                           0000* 00137         .WORD   <BUG$_NOTWCBIRP!4>                       ;
                         50               69  D0 00139 20$:    MOVL    CURRENT_WCB, R0                          ; 0727
               04        0B    A0         02  E1 0013C         BBC     #2, 11(R0), 21$                          ;
                                            FEFF 00141         BUGW                                             ; 0729
                                           0000* 00143         .WORD   <BUG$_NOTFCPWCB!4>                       ;
               2C        2A    A6         03  E1 00145 21$:    BBC     #3, 42(PACKET), 23$                      ; 0737
                               50    2C   B6  D0 0014A         MOVL    @44(PACKET), ABD                         ; 0744
                                     02   A0  B4 0014E         CLRW    2(ABD)                                   ; 0745
                               02    1A   A0  B1 00151         CMPW    26(ABD), #2                              ; 0747
                                          0D  1F 00155         BLSSU   22$                                      ;
                               52    18   A0  9E 00157         MOVAB   24(ABD), R2                              ; 0749
                               51         62  3C 0015B         MOVZWL  (R2), R1                                 ;
                                     01 A241  9F 0015E         PUSHAB  1(R2)[R1]                                ;
                                          9E  B4 00162         CLRW    @(SP)+                                   ;
                               52    20   A0  9E 00164 22$:    MOVAB   32(ABD), R2                              ; 0752
                               51         62  3C 00168         MOVZWL  (R2), R1                                 ;
    22   A0        00          6E         00  2C 0016B         MOVC5   #0, (SP), #0, 34(ABD), 1(R2)[R1]         ;
                                     01 A241     00171                                                          ;
                                          21  11 00174         BRB     26$                                      ; 0737
         2F    20  A6          06         00  ED 00176 23$:    CMPZV   #0, #6, 32(PACKET), #47                  ; 0761
                                          08  1B 0017C         BLEQU   24$                                      ;
         38    20  A6          06         00  ED 0017E         CMPZV   #0, #6, 32(PACKET), #56                  ;
                                          0D  12 00184         BNEQ    25$                                      ;
         38    20  A6          06         00  ED 00186 24$:    CMPZV   #0, #6, 32(PACKET), #56                  ; 0763
                                          09  12 0018C         BNEQ    26$                                      ;
                   04    2A    A6         04  E1 0018E         BBC     #4, 42(PACKET), 26$                      ;
                                            FEFF 00193 25$:    BUGW                                             ; 0765
                                           0000* 00195         .WORD   <BUG$_NOBUFPCKT!4>                       ;
                               50         56  D0 00197 26$:    MOVL    PACKET, R0                               ; 0769
                                              04 0019A         RET                                              ;
                                          50  D4 0019B 27$:    CLRL    R0                                       ; 0771
                                              04 0019D         RET                                              ;

; Routine Size:  414 bytes,    Routine Base:  $CODE$ + 0000


;  390          0772  1
```

```
GLOBAL ROUTINE GET_REQ : L$GET_REQ =

!++
!
! FUNCTIONAL DESCRIPTION:
!       This routine gets a request from the io queue. If there are no
!       entries, it then enables ast's for exec mode and hibernates.
!       If an ast is delivered, then  a process may be unblock and continued
!       from point of block.
!
! CALLING SEQUENCE:
!       GET_REQ(), exec mode
!
! INPUT PARAMETERS:
!       none
!
! IMPLICIT INPUTS:
!       none
!
! OUTPUT PARAMETERS:
!       none
!
! IMPLICIT OUTPUTS:
!       none
!
! ROUTINE VALUE:
!       Address of request i/o packet
!
! SIDE EFFECTS:
!       none
!
!--

    BEGIN

    EXTERNAL REGISTER
        COMMON_REG;

    BIND
        SECONDS = UPLIT (-70000000, -1);

    EXTERNAL
        QUEUE_HEAD      : REF BBLOCK;           ! head of ACP queue

    LOCAL
        PACKET;                                 ! packet address

    REGISTER
        QUEUE_POINTER   : REF BBLOCK;

    WHILE 1
    DO
        BEGIN
        $SETAST(ENBFLG = 0);                    ! disable ast's
        PACKET = KERNEL_CALL(GET_REQUEST);      ! get request off queue

        IF .PACKET NEQ 0
```

```
;  449          0830  3             THEN
;  450          0831  3                 RETURN .PACKET;                         ! got a request to process
;  451          0832  3
;  452          0833  3             ! if there are no volumes to service and the ACP's queue is empty, check
;  453          0834  3             ! if all I/O is done, namely rewinds.  Wait for all I/O before deleting
;  454          0835  3             ! process.  If the ACP still has volumes/requests to service, hibernate.
;  455          0836  3             !
;  456          0837  3             QUEUE_POINTER = .QUEUE_HEAD;
;  457          0838  3
;  458          0839  3             IF .QUEUE_POINTER[AQB$B_MNTCNT] EQL 0
;  459          0840  3                 AND
;  460          0841  3                 .QUEUE_POINTER[AQB$L_ACPQFL] EQL QUEUE_POINTER[AQB$L_ACPQFL]
;  461          0842  3             THEN
;  462          0843  4                 BEGIN
;  463          0844  4
;  464          0845  4                 LOCAL
;  465          0846  4                     CCB     : REF BBLOCK;
;  466          0847  4
;  467          0848  4                 WHILE 1
;  468          0849  4                 DO
;  469          0850  5                     BEGIN
;  470          0851  5                     CCB = KERNEL_CALL ( GET_CCB, .IO_CHANNEL );
;  471          0852  5
;  472          0853  5                     IF .CCB[CCB$W_IOC] EQL 0
;  473          0854  5                     THEN
;  474          0855  5                         EXITLOOP;
;  475          0856  5
;  476          0857  6                     IF $SETIMR(EFN = TIMEFN, DAYTIM = SECONDS)
;  477          0858  5                     THEN
;  478          0859  5                         $WAITFR(EFN = TIMEFN);
;  479          0860  5
;  480          0861  4                     END;                                ! end of short while loop
;  481          0862  4
;  482          0863  4                 CCB[CCB$L_UCB] = .DISK_UCB;
;  483          0864  4                 $DASSGN(CHAN = .MAIL_CHANNEL);
;  484          0865  4                 $DASSGN(CHAN = .IO_CHANNEL);
;  485          0866  4                 KERNEL_CALL(DEALLOCATE, .QUEUE_POINTER);
;  486          0867  4                 $DELPRC();
;  487          0868  4                 END
;  488          0869  3             ELSE
;  489          0870  4                 BEGIN
;  490          0871  4                 CHECK_MAIL();
;  491          0872  4                 $SETAST(ENBFLG = 1);                    ! enable before hibernate
;  492          0873  4                 SYS$HIBER();                            ! hibernate
;  493          0874  3                 END;
;  494          0875  3
;  495          0876  2             END;                                        ! end of while loop
;  496          0877  2
;  497          0878  2         RETURN 1;                                       ! Never Execute
;  498          0879  2                                                         !  but gets rid of info error
;  499          0880  2
;  500          0881  1         END;                                            ! end of routine


                                      0019E          .BLKB   2
                 FFFFFFFF  FBD3E280   001A0 P.AAA:   .LONG   -70000000, -1                                 ;
```

```
                                              SECONDS=            P.AAA
                                                       .EXTRN  SYS$SETAST, SYS$CMKRNL
                                                       .EXTRN  SYS$SETIMR, SYS$WAITFR
                                                       .EXTRN  SYS$DASSGN, SYS$DELPRC

                                 1C  BB 00000 GET_REQ::
                                                       PUSHR   #^M<R2,R3,R4>                          ; 0773
                                 7E  D4 00002 1$:      CLRL    -(SP)                                  ; 0826
         00000000G  00           01  FB 00004          CALLS   #1, SYS$SETAST                         ;
                                 7E  D4 0000B          CLRL    -(SP)                                  ; 0827
                                 5E  DD 0000D          PUSHL   SP                                     ;
                          FE45   CF  9F 0000F          PUSHAB  GET_REQUEST                            ;
         00000000G  9F           03  FB 00013          CALLS   #3, @#SYS$CMKRNL                       ;
                    54           50  D0 0001A          MOVL    R0, PACKET                             ;
                                 06  13 0001D          BEQL    2$                                     ; 0829
                    50           54  D0 0001F          MOVL    PACKET, R0                             ; 0831
                               0095  31 00022          BRW     7$                                     ;
                    52    0000G  CF  D0 00025 2$:      MOVL    QUEUE_HEAD, QUEUE_POINTER              ; 0837
                          0B     A2  95 0002A          TSTB    11(QUEUE_POINTER)                      ; 0839
                                 73  12 0002D          BNEQ    5$                                     ;
                    52           62  D1 0002F          CMPL    (QUEUE_POINTER), QUEUE_POINTER         ; 0841
                                 6E  12 00032          BNEQ    5$                                     ;
                          0000G  CF  DD 00034 3$:      PUSHL   IO_CHANNEL                             ; 0851
                                 01  DD 00038          PUSHL   #1                                     ;
                                 5E  DD 0003A          PUSHL   SP                                     ;
                          0000V  CF  9F 0003C          PUSHAB  GET_CCB                                ;
         00000000G  9F           04  FB 00040          CALLS   #4, @#SYS$CMKRNL                       ;
                    53           50  D0 00047          MOVL    R0, CCB                                ;
                            0A   A3  B5 0004A          TSTW    10(CCB)                                ; 0853
                                 1C  13 0004D          BEQL    4$                                     ;
                                 7E  7C 0004F          CLRQ    -(SP)                                  ; 0857
                            A4   AF  9F 00051          PUSHAB  SECONDS                                ;
                                 03  DD 00054          PUSHL   #3                                     ;
         00000000G  00           04  FB 00056          CALLS   #4, SYS$SETIMR                         ;
                    D4           50  E9 0005D          BLBC    R0, 3$                                 ;
                                 03  DD 00060          PUSHL   #3                                     ; 0859
         00000000G  00           01  FB 00062          CALLS   #1, SYS$WAITFR                         ;
                                 C9  11 00069          BRB     3$                                     ; 0848
                    63    0000G  CF  D0 0006B 4$:      MOVL    DISK_UCB, (CCB)                        ; 0863
                          0000G  CF  DD 00070          PUSHL   MAIL_CHANNEL                           ; 0864
         00000000G  00           01  FB 00074          CALLS   #1, SYS$DASSGN                         ;
                          0000G  CF  DD 0007B          PUSHL   IO_CHANNEL                             ; 0865
         00000000G  00           01  FB 0007F          CALLS   #1, SYS$DASSGN                         ;
                                 52  DD 00086          PUSHL   QUEUE_POINTER                          ; 0866
                                 01  DD 00088          PUSHL   #1                                     ;
                                 5E  DD 0008A          PUSHL   SP                                     ;
                          0000G  CF  9F 0008C          PUSHAB  DEALLOCATE                             ;
         00000000G  9F           04  FB 00090          CALLS   #4, @#SYS$CMKRNL                       ;
                                 7E  7C 00097          CLRQ    -(SP)                                  ; 0867
         00000000G  00           02  FB 00099          CALLS   #2, SYS$DELPRC                         ;
                                 15  11 000A0          BRB     6$                                     ; 0839
             0000G  CF           00  FB 000A2 5$:      CALLS   #0, CHECK_MAIL                         ; 0871
                                 01  DD 000A7          PUSHL   #1                                     ; 0872
         00000000G  00           01  FB 000A9          CALLS   #1, SYS$SETAST                         ;
         00000000G  9F           00  FB 000B0          CALLS   #0, @#SYS$HIBER                        ; 0873
                               FF48  31 000B7 6$:      BRW     1$                                     ; 0823
```

```
                                                  1C  BA 000BA 7$:      POPR     #^M<R2,R3,R4>                                   ; 0881
                                                      05 000BC          RSB                                                      ;

; Routine Size:  189 bytes,    Routine Base:  $CODE$ + 01A8


;  501          0882  1
```

```
1 GLOBAL ROUTINE GET_CCB (CHANNEL) =
1
1 !++
1 !
1 ! FUNCTIONAL DESCRIPTION:
1 !
1 !       This routine returns the address of the channel control block
1 !       associated with the given channel.
1 !
1 !
1 ! CALLING SEQUENCE.
1 !       GET_CCB (ARG1) in kernel mode
1 !
1 ! INPUT PARAMETERS:
1 !       ARG1: channel number
1 !
1 ! IMPLICIT INPUTS:
1 !       NONE
1 !
1 ! OUTPUT PARAMETERS:
1 !       NONE
1 !
1 ! IMPLICIT OUTPUTS:
1 !       NONE
1 !
1 ! ROUTINE VALUE:
1 !       address of CCB
1 !
1 ! SIDE EFFECTS:
1 !       NONE
1 !
1 !--
1
1
2 BEGIN
2
2 LINKAGE
2       L_VERIFYCHAN    = JSB (REGISTER = 0) :
2                         GLOBAL (CCB = 1)
2                         NOPRESERVE (2, 3, 4, 5);
2
2 GLOBAL REGISTER
2       CCB             = 1 : REF BBLOCK; ! CCB address returned
2
2 LOCAL
2       STATUS;                         ! status of system call
2
2 EXTERNAL ROUTINE
2       IOC$VERIFYCHAN  : L_VERIFYCHAN ADDRESSING_MODE (ABSOLUTE);
2                                       ! exec routine to find CCB
2
2 STATUS = IOC$VERIFYCHAN (.CHANNEL);
2 IF NOT .STATUS
2 THEN BUG_CHECK (INVCHAN, FATAL, 'Invalid ACP channel number');
2
2 RETURN .CCB;
2
```

```
;  560          0940  1 END;                                         ! end of routine GET_CCB


                                                                              .EXTRN  IOC$VERIFYCHAN, BUG$_INVCHAN

                                                     0FFC 00000               .ENTRY  GET_CCB, Save R2,R3,R4,R5,R6,R7,R8,R9,R10,- ; 0883
                                                                                      R11                                         ;
                                        50       04  AC  D0 00002             MOVL    CHANNEL, R0                                 ; 0934
                                           00000000G  9F  16 00006             JSB     @#IOC$VERIFYCHAN                            ;
                                        04           50  E8 0000C             BLBS    STATUS, 1$                                  ; 0935
                                                     FEFF 0000F             BUGW                                                ; 0936
                                                     0000* 00011             .WORD   <BUG$_INVCHAN!4>                            ;
                                        50           51  D0 00013 1$:         MOVL    CCB, R0                                     ; 0938
                                                         04 00016             RET                                                 ; 0940

; Routine Size:  23 bytes,    Routine Base:  $CODE$ + 0265


;  561          0941  1 END
;  562          0942  0 ELUDOM




;                                        PSECT SUMMARY
;
;       Name                        Bytes                          Attributes
;
;  $CODE$                               636  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        51         0       1000          00:01.9





;                                         COMMAND QUALIFIERS

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

; Size:          626 code + 10 data bytes
; Run Time:         00:16.9
; Elapsed Time:     00:34.4
; Lines/CPU Min:      3350
; Lexemes/CPU-Min: 21698
; Memory Used:  200 pages
; Compilation Complete
```

0254 AH-BT13A-SE
VAX/VMS V4.0

DIGITAL EQUIPMENT CORPORATION
CONFIDENTIAL AND PROPRIETARY

FREEPG
LIS

DEACCS
LIS

FIND
LIS

DATECN
LIS

FRMOD1
LIS

CREATE
LIS

GETREQ
LIS

EXPIRE
LIS

CNTRL
LIS

FRMHDR
LIS

HEADER
LIS

COMLABPRC
LIS

ENDVOL
LIS

GETFIB
LIS