```
FFFFFFFFFFFFFFF        111              111        XXX         XXX
FFFFFFFFFFFFFFF        111              111        XXX         XXX
FFFFFFFFFFFFFFF        111              111        XXX         XXX
FFF                 111111           111111        XXX         XXX
FFF                 111111           111111        XXX         XXX
FFF                 111111           111111        XXX         XXX
FFF                    111              111           XXX   XXX
FFF                    111              111           XXX   XXX
FFF                    111              111           XXX   XXX
FFFFFFFF.FFF           111              111              XXX
FFFFFFFFFFFF           111              111              XXX
FFFFFFFFFFFF           111              111              XXX
FFF                    111              111           XXX   XXX
FFF                    111              111           XXX   XXX
FFF                    111              111           XXX   XXX
FFF                    111              111        XXX         XXX
FFF                    111              111        XXX         XXX
FFF                    111              111        XXX         XXX
FFF                 111111111        111111111     XXX         XXX
FFF                 111111111        111111111     XXX         XXX
FFF                 111111111        111111111     XXX         XXX
```

```
_$2

Symb
----
IOC
IO_
IO_
IO_
IO_
IO_
KIL

KIL
KIL
LB_
LB_
LB_
LB_
LB_
LOC
LOC
LOC

LOC
LOC
LOC
LOC_
LOC_
L_C
L_C
L_D
L_D
MAI
MAKE
MAKE
MAKE
MAKE
MAKE
MAKE

MAKE
MAK
MAP_
MAP_

MAP
MAR
MAR
MAR
MAR
MAR
```

FILEID**MODIFY

```
MM      MM    000000    DDDDDDDD      IIIIII    FFFFFFFFFF  YY      YY
MM      MM    000000    DDDDDDDD      IIIIII    FFFFFFFFFF  YY      YY
MMMM  MMMM  00      00  DD      DD      II      FF          YY      YY
MMMM  MMMM  00      00  DD      DD      II      FF          YY      YY
MM  MM  MM  00      00  DD      DD      II      FF            YY  YY
MM  MM  MM  00      00  DD      DD      II      FF            YY  YY
MM      MM  00      00  DD      DD      II      FFFFFFFF        YY
MM      MM  00      00  DD      DD      II      FFFFFFFF        YY
MM      MM  00      00  DD      DD      II      FF              YY
MM      MM  00      00  DD      DD      II      FF              YY
MM      MM  00      00  DD      DD      II      FF              YY          ....
MM      MM  00      00  DD      DD      II      FF              YY          ....
MM      MM    000000    DDDDDDDD      IIIIII    FF              YY          ....
MM      MM    000000    DDDDDDDD      IIIIII    FF              YY          ....

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 MODULE MODIFY (
0               LANGUAGE (BLISS32),
0               IDENT = 'V04-001'
0               ) =
1 BEGIN
1
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:  F11ACP Structure Level 2
1 !
1 ! ABSTRACT:
1 !
1 !       This routine implements the MODIFY function.
1 !
1 ! ENVIRONMENT:
1 !
1 !       STARLET operating system, including privileged system services
1 !       and internal exec routines.
1 !
1 !--
1 !
1 !
1 ! AUTHOR:  Andrew C. Goldstein,  CREATION DATE:  6-Jan-1977  23:03
1 !
1 ! MODIFIED BY:
1 !
1 !       V04-001 ACG65998        Andrew C. Goldstein,    6-Sep-1984  15:33
1 !               Checksum file header after setting revision date
1 !
1 !       V03-014 CDS0007         Christian D. Saether    14-Aug-1984
1 !               Modify handling of extension fcbs.
1 !
1 !       V03-013 CDS0006         Christian D. Saether    22-Apr-1984
```

```
1 !                     Modify access arbitration.
1 !
1 !             V03-012 ACG0412         Andrew C. Goldstein,    22-Mar-1984  18:26
1 !                     Implement agent access mode support; add access mode to
1 !                     protection check call
1 !
1 !             V03-011 CDS0005         Christian D. Saether    19-Dec-1983
1 !                     Use BIND_COMMON macro to reduce number
1 !                     of external COMMON declarations.
1 !
1 !             V03-010 CDS0004         Christian D. Saether    25-Sep-1983
1 !                     Modify SERIAL_FILE interface.
1 !
1 !             V03-009 CDS0003         Christian D. Saether    25-Sep-1983
1 !                     Manually merge ACG0343, ACG56916.
1 !
1 !             V03-008 ACG0343         Andrew C. Goldstein,    19-Jul-1983  16:44
1 !                     Inhibit revision counting if NORECORD is specified
1 !
1 !             V03-007 ACG56916        Andrew C. Goldstein,    21-Jun-1983  14:11
1 !                     Update file revision and expiration when modified
1 !
1 !             V03-006 LMP0149         L. Mark Pilant,         13-Sep-1983  11:27
1 !                     Correct a logic problem that caused problems during the
1 !                     protection check of a write attribute operation.
1 !
1 !             V03-005 CDS0002         Christian D. Saether     4-May-1983
1 !                     Add call to SERIAL_FILE to interlock file processing.
1 !
1 !             V03-004 CDS0001         Christian D. Saether    20-Apr-1983
1 !                     Changes to arbitrate extend and truncate access
1 !                     in a cluster.  Truncate now requires exclusive
1 !                     access to the file to succeed.
1 !                     Change interface to TRUNCATE routine.
1 !
1 !             V03-003 LMP0059         L. Mark Pilant,         21-Dec-1982  11:37
1 !                     Always create an FCB for a file header.  This eliminates
1 !                     a lot of special case FCB handling.
1 !
1 !             V03-002 LMP0036         L. Mark Pilant,         17-Aug-1982  10:15
1 !                     Add support for ACL's.
1 !
1 !             V03-001 ACG0282         Andrew C. Goldstein,    6-Apr-1982  16:08
1 !                     Check for device write-locked before attempting operations
1 !
1 !             V02-007 ACG0223         Andrew C. Goldstein,    17-Nov-1981  21:49
1 !                     Allow modification of directory version limit
1 !
1 !             V02-006 ACG0171         Andrew C. Goldstein,    7-May-1980  18:34
1 !                     Condition check of truncate lock count on presence of FCB
1 !
1 !             V02-005 ACG0167         Andrew C. Goldstein,    16-Apr-1980  19:27
1 !                     Previous revision history moved to F11B.REV
1 !**
1
1
1 LIBRARY 'SYS$LIBRARY:LIB.L32';
```

```
;  115          0115  1 REQUIRE 'SRC$:FCPDEF.B32';
;  116          1106  1
```

```
1 GLOBAL ROUTINE MODIFY  : L_NORM =
1
1 !++
1 !
1 ! FUNCTIONAL DESCRIPTION:
1 !
1 !       This routine implements the MODIFY function.
1 !
1 ! CALLING SEQUENCE:
1 !       MODIFY ()
1 !
1 ! INPUT PARAMETERS:
1 !       NONE
1 !
1 ! IMPLICIT INPUTS:
1 !       CURRENT_WINDOW: window for file
1 !       PRIMARY_FCB: FCB of file
1 !       IO_PACKET: I/O packet in process
1 !       FILE_HEADER: address of current file header
1 !
1 ! OUTPUT PARAMETERS:
1 !       NONE
1 !
1 ! IMPLICIT OUTPUTS:
1 !       NONE
1 !
1 ! ROUTINE VALUE:
1 !       NONE
1 !
1 ! SIDE EFFECTS:
1 !       file and FCB modified
1 !
1 !--
1
2 BEGIN
2
2 LOCAL
2       FCB_CREATED,                    ! flag indicating new FCB created
2       FIND_MODE,                      ! mode for FIND call
2       ABD             : REF BBLOCKVECTOR [,ABD$C_LENGTH],
2                                       ! buffer descriptors
2       FIB             : REF BBLOCK,   ! FIB
2       FCB             : REF BBLOCK,   ! FCB of file
2       HEADER          : REF BBLOCK,   ! file header
2       STATUS;                         ! Routine exit status
2
2 BIND_COMMON;
2
2 EXTERNAL ROUTINE
2       REBLD_PRIM_FCB  : L_NORM NOVALUE, ! rebuild primary fcb from header
2       BUILD_EXT_FCBS  : L_NORM NOVALUE, ! build extension fcb chain.
2       ARBITRATE_ACCESS : [ JSB_2ARGS, ! arbitrate access interlocks.
2       CONV_ACCLOCK    : L_NORM,       ! convert file access lock
2       LOCK_COUNT      : L_NORM,       ! determine number of locks.
2       SERIAL_FILE     : L_NORM,       ! interlock file processing.
2       GET_FIB         : L_NORM,       ! get FIB of request
2       GET_LOC_ATTR    : L_NORM,       ! get placement data from attribute list
```

```
2              GET_LOC          : L_NORM,          ! get placament data
2              FIND             : L_NORM,          ! find name in directory
2              SWITCH_VOLUME    : L_NORM,          ! switch context to right volume
2              SEARCH_FCB       : L_NORM,          ! search FCB list
2              READ_HEADER      : L_NORM,          ! read file header
2              CREATE_FCB       : L_NORM,          ! create an FCB
2              CHECK_PROTECT    : L_NORM,          ! check file protection
2              SET_REVISION     : L_NORM,          ! set file revision end expiration
2              WRITE_ATTRIB     : L_NORM,          ! write file attributes
2              EXTEND           : L_NORM,          ! extend file
2              TRUNCATE         : L_NORM,          ! truncate file
2              CHECKSUM         : L_NORM,          ! checksum the file header
2              UPDATE_FCB       : L_NORM;          ! rebuild fcb from header
2
2
2 ! First find the buffer descriptor, FIB, FCB, etc., and read the header.
2 !
2
2                                                  ! pointer to buffer descriptors
2 ABD = .BBLOCK [.IO_PACKET[IRP$L_SVAPTE], AIB$L_DESCRIPT];
2 FIB = GET_FIB (.ABD);
2 IF .FIB[FIB$W_VERLIMIT] GTRU 32767
2 THEN ERR_EXIT (SS$_BADPARAM);
2
2 IF .FIB[FIB$V_ALLOCATR]
2 THEN GET_LOC_ATTR (.ABD, .FIB);
2 GET_LOC (.FIB, LOC_RVN, LOC_LBN);
2
2 ! If a directory ID is present, do a directory search first.
2 !
2
2 FIND_MODE = 0;
2 IF .FIB[FIB$W_VERLIMIT] NEQ 0
2 THEN FIND_MODE = 2;
2 IF .CLEANUP_FLAGS[CLF_DIRECTORY]
2 THEN FIND (.ABD, .FIB, .FIND_MODE);
2 SWITCH_VOLUME (.FIB[FIB$W_FID_RVN]);
2
2 ! If there is a file open on the channel, check the file ID returned by the
2 ! FIND against the file ID that is open. If they are different, drop the FCB
2 ! and window addresses on the floor.
2 !
2
2 IF .PRIMARY_FCB NEQ 0
2 THEN
2     IF .PRIMARY_FCB[FCB$W_FID_NUM] NEQ .FIB[FIB$W_FID_NUM]
2     OR .PRIMARY_FCB[FCB$W_FID_RVN] NEQ .FIB[FIB$W_FID_RVN]
2     THEN
3         BEGIN
3         PRIMARY_FCB = 0;
3         CURRENT_WINDOW = 0;
2         END;
2
2 ! Synchronize further processing on this FID.
2 !
2
2 PRIM_LCKINDX = SERIAL_FILE (FIB [FIB$W_FID]);
```

```
2
2 FCB = SEARCH_FCB (FIB[FIB$W_FID]);
2 HEADER = READ_HEADER (FIB[FIB$W_FID], .FCB);
2
2 ! At this point build the necessary FCB list if the file is not accessed.
2 ! This is necessary to allow the ACL to be built.
2 !
2
2 FCB_CREATED = 0;
2 IF .FCB EQL 0
2 THEN
3     BEGIN
3     FCB_CREATED = 1;
3     FCB = KERNEL_CALL (CREATE_FCB, .HEADER);
2     END;
2 PRIMARY_FCB =  FCB;                                   ! Record FCB for external use
2
2 ! If the file is multi-header, read the extension headers and create
2 ! extension FCB's as necessary.  Finally, read back the primary header.
2 .
2
2 IF .FCB_CREATED
2 THEN
2     BUILD_EXT_FCBS (.HEADER)
2 ELSE
2     IF .FCB [FCB$V_STALE]
2     THEN
3         BEGIN
3
3         REBLD_PRIM_FCB (.FCB, .HEADER);
3
3         BUILD_EXT_FCBS (.HEADER);
3
2         END;
2
2 ! Check that the volume is write enabled.
2 !
2
2 IF .BBLOCK [CURRENT_UCB[UCB$L_DEVCHAR], DEV$V_SWL]
2 THEN ERR_EXIT (SS$_WRITLCK);
2
2 ! Arbitrate access interlocks. If this is the accessor, then the file
2 ! must be write accessed. Count a write to the file.
2 !
2
2 IF .CURRENT_WINDOW NEQ 0
2 THEN
3     BEGIN
3
3     IF NOT .CURRENT_WINDOW[WCB$V_WRITE]
3     THEN ERR_EXIT (SS$_NOPRIV);
3
3     IF .FIB [FIB$V_TRUNC]
3     THEN
3         IF .FCB [FCB$W_REFCNT] NEQ 1
3             OR LOCK_COUNT (.FCB [FCB$L_ACCLKID]) NEQ 1
3         THEN
```

```
3               ERR_EXIT (SS$_ACCONFLICT);
3
3       IF NOT .FIB[FIB$V_NORECORD]
3       THEN
3           CURRENT_WINDOW [WCB$L_WRITES] = .CURRENT_WINDOW [WCB$L_WRITES] + 1;
3
3       END
3
3 ! If it is not, then the file must not be locked against modification
3 ! and the caller must pass file protection. Count a revision to the file.
3 !
3
2 ELSE
3       BEGIN
3
3       IF .FIB [FIB$V_EXTEND] OR .FIB [FIB$V_TRUNC]
3       THEN
4           BEGIN
4           LOCAL
4               CURR_LKMODE;
4
4           CHECK_PROTECT (WRITE_ACCESS, .HEADER, .FCB,
4                          MAXU (.IO_PACKET[IRP$V_MODE], .FIB[FIB$B_AGENT_MODE]));
4
4           CURR_LKMODE = .FCB [FCB$B_ACCLKMODE];
4
4           IF .FIB [FIB$V_EXTEND]
4           THEN
5               BEGIN
5
5               IF .FIB[FIB$V_TRUNC]
5               THEN ERR_EXIT (SS$_BADPARAM);
5
5               IF NOT ARBITRATE_ACCESS (FIB$M_WRITE, .FCB)
5               THEN
5                   ERR_EXIT (SS$_ACCONFLICT);
5               END
4           ELSE
4
4 ! This is a truncation.  Truncation is only allowed if there is no other
4 ! access to the file whatever.
4 !
4
5               BEGIN
5               IF NOT ARBITRATE_ACCESS (FIB$M_NOREAD, .FCB)
5               THEN
5                   ERR_EXIT (SS$_ACCONFLICT);
5
5               IF .FCB [FCB$W_REFCNT] NEQ 0
5                   OR LOCK_COUNT (.FCB [FCB$L_ACCLKID]) NEQ 1
5               THEN
6                   BEGIN
6                   CONV_ACCLOCK (.CURR_LKMODE, .FCB);
6                   ERR_EXIT (SS$_ACCONFLICT);
5                   END;
5
4               END;          ! of trunc
```

```
4
4           CONV_ACCLOCK (.CURR_LKMODE, .FCB);
4
3           END;            ! of trunc or extend
3
3       IF NOT .FIB[FIB$V_NORECORD]
3       THEN
4           BEGIN
4           SET_REVISION (.HEADER, 1);
4           CHECKSUM (.HEADER);
3           END;
2       END;
2
2   ! If an attribute list exists, perform the write attributes operation.
2   !
2
2   IF .IO_PACKET[IRP$W_BCNT] GTR ABD$C_ATTRIB
2   THEN
3       BEGIN
3       WRITE_ATTRIB (.HEADER, .ABD, 1);
3       HEADER = .FILE_HEADER;
3       CHECKSUM (.HEADER);
2       END;
2
2   ! If the extend enable bit is on, perform the extend operation.
2   ! If the truncate bit is on, perform the truncate operation. If both are
2   ! on, it is an error.
2   !
2
3   IF (.FIB[FIB$V_EXTEND] OR .FIB[FIB$V_TRUNC])
2   AND .CURRENT_VCB[VCB$V_NOALLOC]
2   THEN ERR_EXIT (SS$_WRITLCK);
2
2   IF .FIB[FIB$V_EXTEND]
2   THEN
3       BEGIN
3       EXTEND (.FIB, .HEADER);
2       END;
2
2   IF .FIB[FIB$V_TRUNC]
2   THEN
2       TRUNCATE (.FIB, .HEADER, .FIB [FIB$L_EXVBN]);
2
2   HEADER = .FILE_HEADER;
2   CHECKSUM (.HEADER);                     ! checksum the file header
2   UPDATE_FCB (.HEADER);
2
2   RETURN 1;
2
1   END;                                    ! end of routine MODIFY


                                                    .TITLE  MODIFY
                                                    .IDENT  \V04-001\

                                                    .EXTRN  REBLD_PRIM_FCB, BUILD_EXT_FCBS
                                                    .EXTRN  ARBITRATE_ACCESS
```

```
                                                                  .EXTRN  CONV_ACCLOCK, LOCK_COUNT
                                                                  .EXTRN  SERIAL_FILE, GET_FIB
                                                                  .EXTRN  GET_LOC_ATTR, GET_LOC
                                                                  .EXTRN  FIND, SWITCH_VOLUME
                                                                  .EXTRN  SEARCH_FCB, READ_HEADER
                                                                  .EXTRN  CREATE_FCB, CHECK_PROTECT
                                                                  .EXTRN  SET_REVISION, WRITE_ATTRIB
                                                                  .EXTRN  EXTEND, TRUNCATE
                                                                  .EXTRN  CHECKSUM, UPDATE_FCB

                                                                  .PSECT  $CODE$,NOWRT,2

                                        00FC 00000                .ENTRY  MODIFY, Save R2,R3,R4,R5,R6,R7               ; 1107
                  57        0000G   CF   9E 00002                 MOVAB   CHECKSUM, R7
                  50           90   AA   D0 00007                 MOVL    -112(BASE), R0                              ; 1183
                  56           2C   B0   D0 0000B                 MOVL    @44(R0), ABD                                ;
                                         56   DD 0000F            PUSHL   ABD                                         ; 1184
              0000G   CF                 01   FB 00011            CALLS   #1, GET_FIB                                 ;
                      52                 50   D0 00016            MOVL    R0, FIB                                     ;
               7FFF   8F           2C    A2   B1 00019            CMPW    44(FIB), #32767                             ; 1185
                                         03   1B 0001F            BLEQU   1$                                          ;
                                       0118   31 00021            BRW     17$                                         ;
        09       16   A2                 04   E1 00024  1$:       BBC     #4, 22(FIB), 2$                             ; 1188
                                         52   DD 00029            PUSHL   FIB                                         ; 1189
                                         56   DD 0002B            PUSHL   ABD                                         ;
              0000G   CF                 02   FB 0002D            CALLS   #2, GET_LOC_ATTR                            ;
                                   20    AA   9F 00032  2$:       PUSHAB  32(BASE)                                    ; 1190
                                   1C    AA   9F 00035            PUSHAB  28(BASE)                                    ;
                                         52   DD 00038            PUSHL   FIB                                         ;
              0000G   CF                 03   FB 0003A            CALLS   #3, GET_LOC                                 ;
                                         50   D4 0003F            CLRL    FIND_MODE                                   ; 1195
                                   2C    A2   B5 00041            TSTW    44(FIB)                                     ; 1196
                                         03   13 00044            BEQL    3$                                          ;
                      50                 02   D0 00046            MOVL    #2, FIND_MODE                               ; 1197
        0B            6A                 06   E1 00049  3$:       BBC     #6, (BASE), 4$                              ; 1198
                                         50   DD 0004D            PUSHL   FIND_MODE                                   ; 1199
                                         52   DD 0004F            PUSHL   FIB                                         ;
                                         56   DD 00051            PUSHL   ABD                                         ;
              0000G   CF                 03   FB 00053            CALLS   #3, FIND                                    ;
                      7E           08    A2   3C 00058  4$:       MOVZWL  8(FIB), -(SP)                               ; 1200
              0000G   CF                 01   FB 0005C            CALLS   #1, SWITCH_VOLUME                           ;
                      50           08    AA   D0 00061            MOVL    8(BASE), R0                                 ; 1207
                                         11   13 00065            BEQL    6$                                          ;
                 04   A2           24    A0   B1 00067            CMPW    36(R0), 4(FIB)                              ; 1209
                                         07   12 0006C            BNEQ    5$                                          ;
                 08   A2           28    A0   B1 0006E            CMPW    40(R0), 8(FIB)                              ; 1210
                                         03   13 00073            BEQL    6$                                          ;
                                   08    AA   7C 00075  5$:       CLRQ    8(BASE)                                     ; 1213
                                   04    A2   9F 00078  6$:       PUSHAB  4(FIB)                                      ; 1220
              0000G   CF                 01   FB 0007B            CALLS   #1, SERIAL_FILE                             ;
                 18   AA                 50   D0 00080            MOVL    R0, 24(BASE)                                ;
                                   04    A2   9F 00084            PUSHAB  4(FIB)                                      ; 1222
              0000G   CF                 01   FB 00087            CALLS   #1, SEARCH_FCB                              ;
                      53                 50   D0 0008C            MOVL    R0, FCB                                     ;
                                         53   DD 0008F            PUSHL   FCB                                         ; 1223
                                   04    A2   9F 00091            PUSHAB  4(FIB)                                      ;
              0000G   CF                 02   FB 00094            CALLS   #2, READ_HEADER                             ;
```

```
                                 54          50  D0 00099          MOVL    R0, HEADER                                ;
                                             55  D4 0009C          CLRL    FCB_CREATED                               ; 1229
                                             53  D5 0009E          TSTL    FCB                                       ; 1230
                                             0D  12 000A0          BNEQ    7$                                        ;
                                 55          01  D0 000A2          MOVL    #1, FCB_CREATED                           ; 1233
                                             54  DD 000A5          PUSHL   HEADER                                    ; 1234
                         0000G   CF          01  FB 000A7          CALLS   #1, CREATE_FCB                            ;
                                 53          50  D0 000AC          MOVL    R0, FCB                                   ;
                            08   AA          53  D0 000AF 7$:      MOVL    FCB, 8(BASE)                              ; 1236
                                 0B          55  E8 000B3          BLBS    FCB_CREATED, 8$                           ; 1242
                                 0E     23   A3  E9 000B6          BLBC    35(FCB), 9$                               ; 1246
                                             18  BB 000BA          PUSHR   #^M<R3,R4>                                ; 1250
                         0000G   CF          02  FB 000BC          CALLS   #2, REBLD_PRIM_FCB                        ;
                                             54  DD 000C1 8$:      PUSHL   HEADER                                    ; 1252
                         0000G   CF          01  FB 000C3          CALLS   #1, BUILD_EXT_FCBS                        ;
                                 50     94   AA  D0 000C8 9$:      MOVL    -108(BASE), R0                            ; 1259
                 03         3B   A0          01  E1 000CC          BBC     #1, 59(R0), 10$                           ;
                                           00F4  31 000D1          BRW     28$                                       ;
                                 50     0C   AA  D0 000D4 10$:     MOVL    12(BASE), R0                              ; 1266
                                             2D  13 000D8          BEQL    14$                                       ;
                 03         0B   A0          01  E0 000DA          BBS     #1, 11(R0), 11$                           ; 1270
                                             24  BF 000DF          CHMU    #36                                       ; 1271
                                                 04 000E1          RET                                               ;
                                 13     17   A2  E9 000E2 11$:     BLBC    23(FIB), 12$                              ; 1273
                                 01     18   A3  B1 000E6          CMPW    24(FCB), #1                               ; 1275
                                             61  12 000EA          BNEQ    19$                                       ;
                                        48   A3  DD 000EC          PUSHL   72(FCB)                                   ; 1276
                         0000G   CF          01  FB 000EF          CALLS   #1, LOCK_COUNT                            ;
                                 01          50  D1 000F4          CMPL    R0, #1                                    ;
                                             7F  12 000F7          BNEQ    22$                                       ;
                 07              62          15  E0 000F9 12$:     BBS     #21, (FIB), 13$                           ; 1280
                                 50     0C   AA  D0 000FD          MOVL    12(BASE), R0                              ; 1282
                                        28   A0  D6 00101          INCL    40(R0)                                    ;
                                           0091  31 00104 13$:     BRW     25$                                       ; 1266
                                        16   A2  95 00107 14$:     TSTB    22(FIB)                                   ; 1293
                                             04  19 0010A          BLSS    15$                                       ;
                                 76     17   A2  E9 0010C          BLBC    23(FIB), 24$                              ;
                                 50     90   AA  D0 00110 15$:     MOVL    -112(BASE), R0                            ; 1300
       7E         0B     A0      02          00  EF 00114          EXTZV   #0, #2, 11(R0), -(SP)                     ;
                                 6E     2E   A2  91 0011A          CMPB    46(FIB), (SP)                             ;
                                             04  1B 0011E          BLEQU   16$                                       ;
                                 6E     2E   A2  9A 00120          MOVZBL  46(FIB), (SP)                             ;
                                             53  DD 00124 16$:     PUSHL   FCB                                       ; 1299
                                             54  DD 00126          PUSHL   HEADER                                    ;
                                             01  DD 00128          PUSHL   #1                                        ;
                         0000G   CF          04  FB 0012A          CALLS   #4, CHECK_PROTECT                         ;
                                 55     0B   A3  9A 0012F          MOVZBL  11(FCB), CURR_LKMODE                      ; 1302
                                        16   A2  95 00133          TSTB    22(FIB)                                   ; 1304
                                             17  18 00136          BGEQ    20$                                       ;
                                 03     17   A2  E9 00138          BLBC    23(FIB), 18$                              ; 1308
                                             14  BF 0013C 17$:     CHMU    #20                                       ; 1309
                                                 04 0013E          RET                                               ;
                                 51          53  D0 0013F 18$:     MOVL    FCB, R1                                   ; 1311
                                 50   0100   8F  3C 00142          MOVZWL  #256, R0                                  ;
                                          0000G  30 00147          BSBW    ARBITRATE_ACCESS                          ;
                                 30          50  E8 0014A          BLBS    R0, 23$                                   ;
                                             29  11 0014D 19$:     BRB     22$                                       ; 1313
```

```
                        51             53 D0 0014F 20$:     MOVL     FCB, R1                                        ; 1322
                        50      0400   8F 3C 00152          MOVZWL   #1024, R0                                      ;
                             0000G     30 00157             BSBW     ARBITRATE_ACCESS                               ;
                        1B             50 E9 0015A          BLBC     R0, 22$                                        ;
                                18     A3 B5 0015D          TSTW     24(FCB)                                        ; 1326
                                       0D 12 00160          BNEQ     21$                                            ;
                                48     A3 DD 00162          PUSHL    72(FCB)                                        ; 1327
                   0000G  CF           01 FB 00165          CALLS    #1, LOCK_COUNT                                 ;
                          01           50 D1 0016A          CMPL     R0, #1                                         ;
                                       0E 13 0016D          BEQL     23$                                            ;
                                       53 DD 0016F 21$:     PUSHL    FCB                                            ; 1330
                                       55 DD 00171          PUSHL    CURR_LKMODE                                    ;
                   0000G  CF           02 FB 00173          CALLS    #2, CONV_ACCLOCK                               ;
                                0800   8F BF 00178 22$:     CHMU     #2048                                          ; 1331
                                          04 0017C          RET                                                     ;
                                       53 DD 0017D 23$:     PUSHL    FCB                                            ; 1336
                                       55 DD 0017F          PUSHL    CURR_LKMODE                                    ;
                   0000G  CF           02 FB 00181          CALLS    #2, CONV_ACCLOCK                               ;
        0E                62           15 E0 00186 24$:     BBS      #21, (FIB), 25$                                ; 1340
                                       01 DD 0018A          PUSHL    #1                                             ; 1343
                                       54 DD 0018C          PUSHL    HEADER                                         ;
                   0000G  CF           02 FB 0018E          CALLS    #2, SET_REVISION                               ;
                                       54 DD 00193          PUSHL    HEADER                                         ; 1344
                          67           01 FB 00195          CALLS    #1, CHECKSUM                                   ;
                          50    90     AA D0 00198 25$:     MOVL     -112(BASE), R0                                 ; 1351
                          05    32     A0 B1 0019C          CMPW     50(R0), #5                                     ;
                                       14 1B 001A0          BLEQU    26$                                            ;
                                       01 DD 001A2          PUSHL    #1                                             ; 1354
                                0050   8F BB 001A4          PUSHR    #^M<R4,R6>                                     ;
                   0000G  CF           03 FB 001A8          CALLS    #3, WRITE_ATTRIB                               ;
                          54    04     AA D0 001AD          MOVL     4(BASE), HEADER                                ; 1355
                                       54 DD 001B1          PUSHL    HEADER                                         ; 1356
                          67           01 FB 001B3          CALLS    #1, CHECKSUM                                   ;
                                16     A2 95 001B6 26$:     TSTB     22(FIB)                                        ; 1364
                                       04 19 001B9          BLSS     27$                                            ;
                          0E    17     A2 E9 001BB          BLBC     23(FIB), 29$                                   ;
                          50    98     AA D0 001BF 27$:     MOVL     -104(BASE), R0                                 ; 1365
        05          0B    A0           04 E1 001C3          BBC      #4, 11(R0), 29$                                ;
                                025C   8F BF 001C8 28$:     CHMU     #604                                           ; 1366
                                          04 001CC          RET                                                     ;
                                16     A2 95 001CD 29$:     TSTB     22(FIB)                                        ; 1368
                                       07 18 001D0          BGEQ     30$                                            ;
                                       14 BB 001D2          PUSHR    #^M<R2,R4>                                     ; 1371
                   0000G  CF           02 FB 001D4          CALLS    #2, EXTEND                                     ;
                          0A    17     A2 E9 001D9 30$:     BLBC     23(FIB), 31$                                   ; 1374
                                1C     A2 DD 001DD          PUSHL    28(FIB)                                        ; 1376
                                       14 BB 001E0          PUSHR    #^M<R2,R4>                                     ;
                   0000G  CF           03 FB 001E2          CALLS    #3, TRUNCATE                                   ;
                          54    04     AA D0 001E7 31$:     MOVL     4(BASE), HEADER                                ; 1378
                                       54 DD 001EB          PUSHL    HEADER                                         ; 1379
                          67           01 FB 001ED          CALLS    #1, CHECKSUM                                   ;
                                       54 DD 001F0          PUSHL    HEADER                                         ; 1380
                   0000G  CF           01 FB 001F2          CALLS    #1, UPDATE_FCB                                 ;
                          50           01 D0 001F7          MOVL     #1, R0                                         ; 1382
                                          04 001FA          RET                                                     ; 1384

; Routine Size:  507 bytes,    Routine Base:  $CODE$ + 0000
```

```
MPW
Sym

ACL
AQE
BIT
CAC
CHI
COL
CUR
DAT
DIR
FCE
HEA
IND
IOC
MAF
MVL
QUC
RVT
UCE
UNP
VBN
VCE
WCE
WIN


PSE
---
.
$AE
$CC


Pha
---
Int
Com
Pas
Sym
Pas
Sym
Pse
Cro
Ass

The
661
The
24
10
```

```
;  396          1385  1
;  397          1386  1 END
;  398          1387  0 ELUDOM




;                              PSECT SUMMARY
;
;      Name                      Bytes                   Attributes
;
;  $CODE$                            507  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        50         0        1000          00:02.0




;                               COMMAND QUALIFIERS

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

; Size:          507 code + 0 data bytes
; Run Time:         00:24.2
; Elapsed Time:     00:56.4
; Lines/CPU Min:     3441
; Lexemes/CPU-Min: 39769
; Memory Used:  319 pages
; Compilation Complete
```

```
MPW
VAX



Mac
---
_$2
_$2
TOT

139

The

MAC
```

0171 AH-BT13A-SE
VAX/VMS V4.0

DIGITAL EQUIPMENT CORPORATION
CONFIDENTIAL AND PROPRIETARY

PMS
LIS

QUOTAUTIL
LIS

MAKPTR
LIS

MATCHNAME
LIS

MPWIND
LIS

LOCKDB
LIS

LOCKERS
LIS

MAKACC
LIS

PARSNM
LIS

MAPVBN
LIS

IODONE
LIS

LOCKDN
LIS

MODIFY
LIS

MOUNT
LIS

NXTHDR
LIS

MAKNMB
LIS

MAKSTR
LIS