```
FFFFFFFFFFFFFFFF         111               111        XXX          XXX
FFFFFFFFFFFFFFFF         111               111        XXX          XXX
FFFFFFFFFFFFFFFF         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
```

```
_$25

Symb
----
IOC$
IO_C
IO_C
IO_D
IO_F
IO_S
KILL

KILL
KILL
LB_E
LB_D
LB_F
LB_H
LB_L
LOCA
LOCA
LOCK

LOCK
LOCK
LOCK
LOC_
LOC_
L_CO
L_CO
L_DA
L_DA
MAIN
MAKE
MAKE
MAKE
MAKE
MAKE
MAKE

MAKE
MAKE
MAP_
MAP_

MAP_
MARK
MARK
MARK
MARK
MARK
```

FILEID**EXTCONTIG

```
EEEEEEEEEE  XX      XX  TTTTTTTTTT    CCCCCCCC    OOOOOO    NN      NN  TTTTTTTTTT    IIIIII      GGGGGGGG
EEEEEEEEEE  XX      XX  TTTTTTTTTT    CCCCCCCC    OOOOOO    NN      NN  TTTTTTTTTT    IIIIII      GGGGGGGG
EE          XX      XX      TT      CC          OO      OO  NN      NN      TT          II      GG
EE          XX      XX      TT      CC          OO      OO  NN      NN      TT          II      GG
EE            XX  XX        TT      CC          OO      OO  NNNN    NN      TT          II      GG
EE            XX  XX        TT      CC          OO      OO  NNNN    NN      TT          II      GG
EEEEEEEE        XX          TT      CC          OO      OO  NN  NN  NN      TT          II      GG
EEEEEEEE        XX          TT      CC          OO      OO  NN  NN  NN      TT          II      GG
EE            XX  XX        TT      CC          OO      OO  NN    NNNN      TT          II      GG  GGGGGG
EE            XX  XX        TT      CC          OO      OO  NN    NNNN      TT          II      GG  GGGGGG
EE          XX      XX      TT      CC          OO      OO  NN      NN      TT          II      GG      GG      ....
EE          XX      XX      TT      CC          OO      OO  NN      NN      TT          II      GG      GG      ....
EEEEEEEEEE  XX      XX      TT        CCCCCCCC    OOOOOO    NN      NN      TT        IIIIII      GGGGGG        ....
EEEEEEEEEE  XX      XX      TT        CCCCCCCC    OOOOOO    NN      NN      TT        IIIIII      GGGGGG        ....

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 EXTCONTIG (
0               LANGUAGE (BLISS32),
0               IDENT = 'V04-000'
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 extends a file, keeping it contiguous by actually
1 !       reallocating and copying the blocks.
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:  13-Jun-1979  17:39
1 !
1 ! MODIFIED BY:
1 !
1 !       V03-003 CDS0001         Christian D. Saether    29-Dec-1983
1 !               Use L_NORM linkage and BIND_COMMON macro.
1 !
1 !       V03-002 ACG0367         Andrew C. Goldstein,    26-Oct-1983  19:50
1 !               Update highwater mark of extended file
1 !
```

```
0058  1 !        V03-001 STJ3070         Steven T. Jeffreys,   23-Mar-1983
0059  1 !                Remove unnecessary reference to RETURN_BLOCKS.
0060  1 !
0061  1 !        V02-004 STJ41739        Steven T. Jeffreys,   24-Nov-1981
0062  1 !                Explicitly set the allocation control bits when
0063  1 !                extending the quota file.  This will prevent the
0064  1 !                extend from suceededing when it should have failed.
0065  1 !
0066  1 !        V02-003 STJ33788        Steven T. Jeffreys,   27-Feb-1981
0067  1 !                Signal error if extend fails.
0068  1 !
0069  1 !        B0102   ACG0055         Andrew C. Goldstein,  25-Jul-1979  18:41
0070  1 !        Interface changes to TRUNCATE_HEADER
0071  1 !
0072  1 !        B0101   ACG0053         Andrew C. Goldstein,  19-Jul-1979  17:51
0073  1 !        Disk quota bug fixes
0074  1 !
0075  1 !**
0076  1
0077  1
0078  1 LIBRARY 'SYS$LIBRARY:LIB.L32';
0079  1 REQUIRE 'SRC$:FCPDEF.B32';
1070  1
1071  1
1072  1 FORWARD ROUTINE
1073  1         EXTEND_CONTIG   : L_NORM,       ! extend a file contiguously
1074  1         HANDLER;                        ! local condition handler
```

```
1 GLOBAL ROUTINE EXTEND_CONTIG (FIB, FCB, SIZE) : L_NORM =
1
1 !++
1 !
1 ! FUNCTIONAL DESCRIPTION:
1 !
1 !       This routine extends a file. If allocated but unused space is
1 !       present, this means simply pushing back the EOF and materializing a
1 !       block of zeroes. If the file is to be physically extended, it is
1 !       copied to a new location on the disk to keep it contiguous.
1 !
1 ! CALLING SEQUENCE:
1 !       EXTEND_CONTIG (ARG1, ARG2, ARG3)
1 !
1 ! INPUT PARAMETERS:
1 !       ARG1: scratch FIB for operation
1 !       ARG2: FCB on which file is open
1 !       ARG3: size by which to extend the file (0 means exponentially)
1 !
1 ! IMPLICIT INPUTS:
1 !       CURRENT_RVN: RVN of current volume
1 !
1 ! OUTPUT PARAMETERS:
1 !       NONE
1 !
1 ! IMPLICIT OUTPUTS:
1 !       PRIMARY_FCB: FCB of file
1 !
1 ! ROUTINE VALUE:
1 !       address of buffer containing next block to use
1 !
1 ! SIDE EFFECTS:
1 !       file extended, storage map altered, FCB & windows altered
1 !
1 !--
1
2 BEGIN
2
2 LINKAGE
2       L_MAKE_POINTER  = CALL :
2                         GLOBAL (MAP_POINTER = 9);
2
2 MAP
2       FIB             : REF BBLOCK,   ! address of FIB for this operation
2       FCB             : REF BBLOCK;   ! address of FCB for file
2
2 BUILTIN
2       ROT,
2       FP;
2
2 GLOBAL REGISTER
2       MAP_POINTER     = 9 : REF BBLOCK; ! pointer to current retrieval pointer
2
2 LOCAL
2       HEADER          : REF BBLOCK,   ! address of file file header
2       NEXT_VBN,                       ! next file VBN to use
2       NEW_SIZE,                       ! size to extend file to
```

```
2               NEW_LBN,                        ! starting LBN of new space
2               BUFFER,                         ! buffer address of current file block
2               NEXT_LBN;                       ! LBN of next block to use
2
2       BIND_COMMON;
2
2       EXTERNAL ROUTINE
2               READ_HEADER     : L_NORM,       ! read file header
2               CHARGE_QUOTA    : L_NORM,       ! charge space to user's quota
2               ALLOC_BLOCKS    : L_NORM,       ! allocate blocks from storage map
2               MAKE_POINTER    : L_MAKE_POINTER, ! build header map pointer
2               READ_BLOCK      : L_NORM,       ! read a disk block
2               RESET_LBN       : L_NORM,       ! assign new LBN to buffer
2               WRITE_BLOCK     : L_NORM,       ! write block to disk
2               CREATE_BLOCK    : L_NORM,       ! fabricate a block buffer
2               INVALIDATE      : L_NORM,       ! invalidate a buffer
2               TRUNCATE_HEADER : L_NORM,       ! truncate file header
2               CHECKSUM        : L_NORM,       ! compute file header checksum
2               WRITE_HEADER    : L_NORM,       ! write file header
2               INIT_FCB2       : L_NORM,       ! update file control block
2               ZERO_WINDOWS    : L_NORM;       ! invalidate related file windows
2
2
2       ! Set up context and read the file header. Note that the file must be contiguous.
2       !
2
2       CH$MOVE (FIB$S_FID, FCB[FCB$W_FID], FIB[FIB$W_FID]);
2       PRIMARY_FCB = .FCB;
2       IF .FCB[FCB$L_STLBN] EQL 0
2       THEN ERR_EXIT (SS$_FILESTRUCT);
2       HEADER = READ_HEADER (0, .FCB);
2
2       ! The next VBN to use is the current file eof block number. If the block
2       ! is not present in the file, the file must be physically extended.
2       !
2
2       NEW_SIZE = 0;
2       NEXT_VBN = .FCB[FCB$L_EFBLK] + 1;
2
2       IF .NEXT_VBN GTRU .FCB[FCB$L_FILESIZE]
2       THEN
3           BEGIN
3
3       ! Compute the number of blocks needed (50% of the current file size),
3       ! or as specified if non-zero, and allocate the new space contiguously.
3       !
3
3           IF .SIZE NEQ 0
3           THEN NEW_SIZE = .SIZE + .FCB[FCB$L_FILESIZE]
3           ELSE NEW_SIZE = .FCB[FCB$L_FILESIZE] + MAXU (.FCB[FCB$L_FILESIZE]/2, 1);
3           CHARGE_QUOTA (.HEADER[FH2$L_FILEOWNER], .NEW_SIZE - .FCB[FCB$L_FILESIZE],
3                       BITLIST (QUOTA_CHECK));
3
3           CLEANUP_FLAGS[CLF_FIXFCB] = 1;
3           FIB[FIB$W_EXCTL] = (FIB$M_ALCON OR FIB$M_FILCON);
3           IF NOT ALLOC_BLOCKS (.FIB, .NEW_SIZE, NEW_LBN, NEW_SIZE)
3           THEN
```

```
3             ERR_EXIT (SS$_DEVICEFULL);
3         UNREC_COUNT = .NEW_SIZE;
3         UNREC_LBN = .NEW_LBN;
3         UNREC_RVN = .CURRENT_RVN;
3
3     ! Now copy the file data from the old file to the newly allocated space.
3     !
3
3         INCR VBN FROM 1 TO .FCB[FCB$L_FILESIZE] DO
4             BEGIN
4             BUFFER = READ_BLOCK (.VBN + .FCB[FCB$L_STLBN] - 1, 1, DATA_TYPE);
4             RESET_LBN (.BUFFER, .VBN + .NEW_LBN - 1);
4             WRITE_BLOCK (.BUFFER);
3             END;
3
3     ! Now deallocate the old file blocks. Then build retrieval pointers
3     ! for the new blocks in the file header. Do the truncation with a local
3     ! condition handler enabled for special error recovery.
3     !
3
3         .FP = HANDLER;
3         TRUNCATE_HEADER (.FIB, .HEADER);
3         .FP = 0;
3
3         HEADER[FH2$B_MAP_INUSE] = 0;
3         CH$FILL (0, (.HEADER[FH2$B_ACOFFSET] - .HEADER[FH2$B_MPOFFSET])*2,
3                 .HEADER + .HEADER[FH2$B_MPOFFSET]*2);
3         MAP_POINTER = .HEADER + .HEADER[FH2$B_MPOFFSET]*2;
3         MAKE_POINTER (.NEW_SIZE, .NEW_LBN, .HEADER);
3         UNREC_COUNT = 0;
3         NEW_SIZE = .NEW_SIZE - .FCB[FCB$L_FILESIZE];
3         KERNEL_CALL (ZERO_WINDOWS, .FCB);
2         END;                                  ! end of file extension
2
2     ! Now that we have enough space in the file, push the end of file
2     ! mark back one block and materialize the new block in memory. Also
2     ! update the FCB and flush any windows on it.
2     ! If this file header supports it, stuff the high water field to
2     ! be the allocated size.
2     !
2
2     IF .HEADER [FH2$B_IDOFFSET] GEQU ($BYTEOFFSET (FH2$L_HIGHWATER)+4)/2
2     THEN
2         HEADER [FH2$L_HIGHWATER] = .NEXT_VBN + 1;
2
2     BBLOCK [HEADER[FH2$W_RECATTR], FAT$L_EFBLK] = ROT (.NEXT_VBN + 1, 16);
2     BBLOCK [HEADER[FH2$W_RECATTR], FAT$W_FFBYTE] = 0;
2     KERNEL_CALL (INIT_FCB2, .FCB, .HEADER);
2     BBLOCK [HEADER[FH2$W_RECATTR], FAT$W_HIBLKL] = .FCB[FCB$L_FILESIZE];
2     CHECKSUM (.HEADER);
2     WRITE_HEADER ();
2     IF .NEW_SIZE NEQ 0
2     THEN CHARGE_QUOTA (.HEADER[FH2$L_FILEOWNER], .NEW_SIZE, BITLIST (QUOTA_CHARGE));
2
2     NEXT_LBN = .FCB[FCB$L_STLBN] + .NEXT_VBN - 1;
2     BUFFER = CREATE_BLOCK (.NEXT_LBN, 1, DATA_TYPE);
2
```

```
;  257          1246  2 RETURN .BUFFER;
;  258          1247  2
;  259          1248  1 END;                                   ! end of routine EXTEND_CONTIG


                                                                          .TITLE  EXTCONTIG
                                                                          .IDENT  \V04-000\

                                                                          .EXTRN  READ_HEADER, CHARGE_QUOTA
                                                                          .EXTRN  ALLOC_BLOCKS, MAKE_POINTER
                                                                          .EXTRN  READ_BLOCK, RESET_LBN
                                                                          .EXTRN  WRITE_BLOCK, CREATE_BLOCK
                                                                          .EXTRN  INVALIDATE, TRUNCATE_HEADER
                                                                          .EXTRN  CHECKSUM, WRITE_HEADER
                                                                          .EXTRN  INIT_FCB2, ZERO_WINDOWS

                                                                          .PSECT  $CODE$,NOWRT,2

                                                    03FC 00000            .ENTRY  EXTEND_CONTIG, Save R2,R3,R4,R5,R6,R7,R8,R9 ; 1075
                               5E               08    C2 00002            SUBL2   #8, SP                                      ;
                               50        04     AC    7D 00005            MOVQ    FIB, R0                                     ; 1158
             04   A0     24    A1               06    28 00009            MOVC3   #6, 36(R1), 4(R0)                           ;
                         08    AA        08     AC    D0 0000F            MOVL    FCB, 8(BASE)                                ; 1159
                               50        08     AC    D0 00014            MOVL    FCB, R0                                     ; 1160
                                         30     A0    D5 00018            TSTL    48(R0)                                      ;
                                                05    12 0001B            BNEQ    1$                                          ;
                                       08C0     8F    BF 0001D            CHMU    #2240                                       ; 1161
                                                      04 00021            RET                                                 ;
                                         08     AC    DD 00022 1$:        PUSHL   FCB                                         ; 1162
                                                7E    D4 00025            CLRL    -(SP)                                       ;
                        0000G  CF               02    FB 00027            CALLS   #2, READ_HEADER                             ;
                               57               50    D0 0002C            MOVL    R0, HEADER                                  ;
                                                6E    D4 0002F            CLRL    NEW_SIZE                                    ; 1168
                               50        08     AC    D0 00031            MOVL    FCB, R0                                     ; 1169
                  56     3C    A0               01    C1 00035            ADDL3   #1, 60(R0), NEXT_VBN                        ;
                         38    A0               56    D1 0003A            CMPL    NEXT_VBN, 56(R0)                            ; 1171
                                                03    1A 0003E            BGTRU   2$                                          ;
                                              00EE    31 00040            BRW     9$                                          ;
                                         0C     AC    D5 00043 2$:        TSTL    SIZE                                        ; 1179
                                                08    13 00046            BEQL    3$                                          ;
                  6E     0C    AC        38     A0    C1 00048            ADDL3   56(R0), SIZE, NEW_SIZE                      ; 1180
                                                0F    11 0004E            BRB     5$                                          ;
                  51     38    A0               02    C7 00050 3$:        DIVL3   #2, 56(R0), R1                              ; 1181
                                                03    12 00055            BNEQ    4$                                          ;
                               51               01    D0 00057            MOVL    #1, R1                                      ;
                               6E      38 B041    9E 0005A 4$:            MOVAB   @56(R0)[R1], NEW_SIZE                       ;
                                                01    DD 0005F 5$:        PUSHL   #1                                          ; 1183
                  7E     04    AE        38     A0    C3 00061            SUBL3   56(R0), NEW_SIZE, -(SP)                     ; 1182
                                         3C     A7    DD 00067            PUSHL   60(HEADER)                                  ;
                        0000G  CF               03    FB 0006A            CALLS   #3, CHARGE_QUOTA                            ;
                               6A               02    88 0006F            BISB2   #2, (BASE)                                  ; 1185
                               50        04     AC    D0 00072            MOVL    FIB, R0                                     ; 1186
                         16    A0               05    B0 00076            MOVW    #5, 22(R0)                                  ;
                                                5E    DD 0007A            PUSHL   SP                                          ; 1187
                                         08     AE    9F 0007C            PUSHAB  NEW_LBN                                     ;
                                         08     AE    DD 0007F            PUSHL   NEW_SIZE                                    ;
                                         04     AC    DD 00082            PUSHL   FIB                                         ;
```

```
                    0000G  CF          04  FB 00085        CALLS   #4, ALLOC_BLOCKS                  :
                           05          50  E8 0008A        BLBS    R0, 6$                            :
                                 0850  8F  BF 0008D        CHMU    #2128                             : 1189
                                           04 00091        RET                                       :
                       28  AA          6E  D0 00092 6$:    MOVL    NEW_SIZE, 40(BASE)                : 1190
                       24  AA    04    AE  D0 00096        MOVL    NEW_LBN, 36(BASE)                 : 1191
                       2C  AA    A0    AA  D0 0009B        MOVL    -96(BASE), 44(BASE)               : 1192
                           50    08    AC  D0 000A0        MOVL    FCB, R0                           : 1197
                           53    38    A0  D0 000A4        MOVL    56(R0), R3                        :
                                       52  D4 000A8        CLRL    VBN                               :
                                       2E  11 000AA        BRB     8$                                :
                                       04  DD 000AC 7$:    PUSHL   #4                                : 1199
                                       01  DD 000AE        PUSHL   #1                                :
                           50    08    AC  D0 000B0        MOVL    FCB, R0                           :
              50           52    30    A0  C1 000B4        ADDL3   48(R0), VBN, R0                   :
                                 FF    A0  9F 000B9        PUSHAB  -1(R0)                            :
                    0000G  CF          03  FB 000BC        CALLS   #3, READ_BLOCK                    :
                           58          50  D0 000C1        MOVL    R0, BUFFER                        :
              50           52    04    AE  C1 000C4        ADDL3   NEW_LBN, VBN, R0                  : 1200
                                 FF    A0  9F 000C9        PUSHAB  -1(R0)                            :
                                       58  DD 000CC        PUSHL   BUFFER                            :
                    0000G  CF          02  FB 000CE        CALLS   #2, RESET_LBN                     :
                                       58  DD 000D3        PUSHL   BUFFER                            : 1201
                    0000G  CF          01  FB 000D5        CALLS   #1, WRITE_BLOCK                   :
              CE           52          53  F3 000DA 8$:    AOBLEQ  R3, VBN, 7$                       : 1197
                           6D   0000V  CF  9E 000DE        MOVAB   HANDLER, (FP)                     : 1209
                                       57  DD 000E3        PUSHL   HEADER                            : 1210
                                 04    AC  DD 000E5        PUSHL   FIB                               :
                    0000G  CF          02  FB 000E8        CALLS   #2, TRUNCATE_HEADER               :
                                       6D  D4 000ED        CLRL    (FP)                              : 1211
                                 3A    A7  94 000EF        CLRB    58(HEADER)                        : 1213
                           51    01    A7  9A 000F2        MOVZBL  1(HEADER), R1                     : 1214
                           50    02    A7  9A 000F6        MOVZBL  2(HEADER), R0                     :
                           50          51  C2 000FA        SUBL2   R1, R0                            :
                           50          02  C4 000FD        MULL2   #2, R0                            :
                           59        6741  3E 00100        MOVAW   (HEADER)[R1], R9                  : 1215
  50          00           6E          00  2C 00104        MOVC5   #0, (SP), #0, R0, (R9)            :
                                       69     00109                                                  :
                           50    01    A7  9A 0010A        MOVZBL  1(HEADER), R0                     : 1216
                           59        6740  3E 0010E        MOVAW   (HEADER)[R0], MAP_POINTER         :
                                       57  DD 00112        PUSHL   HEADER                            : 1217
                                 08    AE  DD 00114        PUSHL   NEW_LBN                           :
                                 08    AE  DD 00117        PUSHL   NEW_SIZE                          :
                    0000G  CF          03  FB 0011A        CALLS   #3, MAKE_POINTER                  :
                                 28    AA  D4 0011F        CLRL    40(BASE)                          : 1218
                           50    08    AC  D0 00122        MOVL    FCB, R0                           : 1219
                           6E    38    A0  C2 00126        SUBL2   56(R0), NEW_SIZE                  :
                                       50  DD 0012A        PUSHL   R0                                : 1220
                    0000G  CF          01  FB 0012C        CALLS   #1, ZERO_WINDOWS                  :
                           28          67  91 00131 9$:    CMPB    (HEADER), #40                     : 1230
                                       05  1F 00134        BLSSU   10$                               :
                       4C  A7    01    A6  9E 00136        MOVAB   1(R6), 76(HEADER)                 : 1232
                           50    01    A6  9E 0013B 10$:   MOVAB   1(R6), R0                         : 1234
         1C   A7           50          10  9C 0013F        ROTL    #16, R0, 28(HEADER)               :
                                 20    A7  B4 00144        CLRW    32(HEADER)                        : 1235
                                       57  DD 00147        PUSHL   HEADER                            : 1236
                                 08    AC  DD 00149        PUSHL   FCB                               :
```

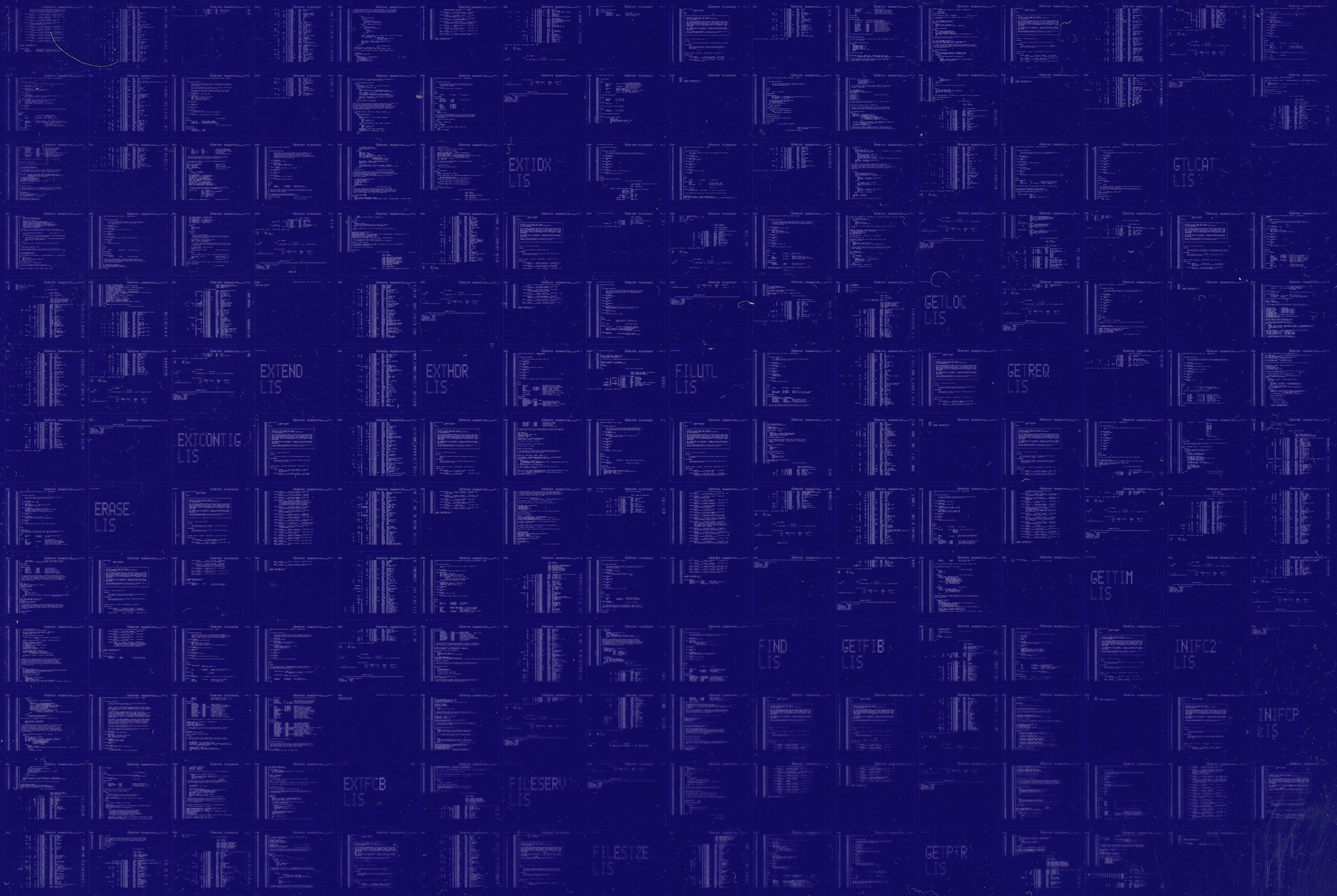
```
                          0000G  CF              02  FB 0014C          CALLS   #2, INIT_FCB2                          ;
                                 50       08     AC  D0 00151          MOVL    FCB, R0                                ; 1237
                            1A   A7       38     A0  B0 00155          MOVW    56(R0), 26(HEADER)                     ;
                                                 57  DD 0015A          PUSHL   HEADER                                 ; 1238
                          0000G  CF              01  FB 0015C          CALLS   #1, CHECKSUM                           ;
                          0000G  CF              00  FB 00161          CALLS   #0, WRITE_HEADER                       ; 1239
                                                 6E  D5 00166          TSTL    NEW_SIZE                               ; 1240
                                                 0D  13 00168          BEQL    11$                                    ;
                                                 02  DD 0016A          PUSHL   #2                                     ; 1241
                                          04     AE  DD 0016C          PUSHL   NEW_SIZE                               ;
                                          3C     A7  DD 0016F          PUSHL   60(HEADER)                             ;
                          0000G  CF              03  FB 00172          CALLS   #3, CHARGE_QUOTA                       ;
                                 50       08     AC  D0 00177 11$:     MOVL    FCB, R0                                ; 1243
                  50             56       30     A0  C1 0017B          ADDL3   48(R0), NEXT_VBN, R0                   ;
                                                 04  DD 00180          PUSHL   #4                                     ; 1244
                                                 01  DD 00182          PUSHL   #1                                     ;
                                                 70  9F 00184          PUSHAB  -(NEXT_LBN)                            ;
                          0000G  CF              03  FB 00186          CALLS   #3, CREATE_BLOCK                       ;
                                 58              50  D0 0018B          MOVL    R0, BUFFER                             ;
                                                 04  0018E             RET                                            ; 1248

; Routine Size:  399 bytes,    Routine Base:  $CODE$ + 0000
```

```
1 ROUTINE HANDLER (SIGNAL, MECHANISM) =
1
1 !++
1 !
1 ! FUNCTIONAL DESCRIPTION:
1 !
1 !       This routine is the condition handler for file extension. It is
1 !       enabled only during the truncate call (deallocating the old file
1 !       blocks). Normal error handling would cause the entire file to
1 !       be dropped on the floor. Since we already have a new good copy, we
1 !       should forge ahead. Note that no error status is returned to the user,
1 !       although we will log a system error.
1 !
1 !
1 ! CALLING SEQUENCE:
1 !       HANDLER (ARG1, ARG2)
1 !
1 ! INPUT PARAMETERS:
1 !       ARG1: address of signal array
1 !       ARG2: address of mechanism array
1 !
1 ! IMPLICIT INPUTS:
1 !       FILE_HEADER: address of file file header
1 !
1 ! OUTPUT PARAMETERS:
1 !       NONE
1 !
1 ! IMPLICIT OUTPUTS:
1 !       NONE
1 !
1 ! ROUTINE VALUE:
1 !       SS$_RESIGNAL or none if unwind
1 !
1 ! SIDE EFFECTS:
1 !       file header map area cleaned out
1 !
1 !--
1
2 BEGIN
2
2 MAP
2       SIGNAL          : REF BBLOCK,   ! signal array arg
2       MECHANISM       : REF BBLOCK;   ! mechanism array arg
2
2
2 ! Check the condition code for FCP error exit and check that it is not a
2 ! write error. Then initialize the header's map area and unwind. On other
2 ! signals we simply resignal.
2 !
2
2 IF .SIGNAL[CHF$L_SIG_NAME] EQL SS$_CMODUSER
2 THEN $UNWIND (DEPADR = MECHANISM[CHF$L_MCH_DEPTH]);
2
2 RETURN SS$_RESIGNAL;                          ! status is irrelevant if unwind
2
1 END;                                          ! end of routine handler
```

```
                                                              .EXTRN  SYS$UNWIND

                                       0000 00000 HANDLER:.WORD   Save nothing                          ; 1249
                      50        04    AC   D0 00002          MOVL     SIGNAL, R0                            ; 1299
           00000424   8F        04    A0   D1 00006          CMPL     4(R0), #1060                          ;
                                      0E   12 0000E          BNEQ     1$                                    ;
                                      7E   D4 00010          CLRL     -(SP)                                 ; 1300
               7E         08  AC      08   C1 00012          ADDL3    #8, MECHANISM, -(SP)                  ;
           00000000G  00              02   FB 00017          CALLS    #2, SYS$UNWIND                        ;
                      50      0918    8F   3C 0001E 1$:      MOVZWL   #2328, R0                             ; 1302
                                           04 00023          RET                                            ; 1304

; Routine Size:  36 bytes,    Routine Base:  $CODE$ + 018F


;  317          1305  1
;  318          1306  1 END
;  319          1307  0 ELUDOM



;                                 PSECT SUMMARY
;
;       Name                           Bytes                        Attributes
;
;  $CODE$                                   435  NOVEC,NOWRT,  RD ,  EXE,NOSHR,  LCL,  REL,  CON,NOPIC,ALIGN(2)



;                              Library Statistics
;
;                                              -------- Symbols --------       Pages       Processing
;       File                                   Total    Loaded   Percent       Mapped      Time
;
;  _$255$DUA28:[SYSLIB]LIB.L32;1               18619        45         0       1000          00:02.0




;                                  COMMAND QUALIFIERS

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

; Size:          435 code + 0 data bytes
; Run Time:         00:21.4
; Elapsed Time:     00:50.1
; Lines/CPU Min:     3669
; Lexemes/CPU-Min: 43544
; Memory Used:  266 pages
; Compilation Complete
```

0170 AH-BT13A-SE
VAX/VMS V4.0

DIGITAL EQUIPMENT CORPORATION
CONFIDENTIAL AND PROPRIETARY



EXTIDX
LIS

GTLCAT
LIS

GETLOC
LIS

EXTEND
LIS

EXTHDR
LIS

FILUTL
LIS

GETREQ
LIS

EXTCONTIG
LIS

ERASE
LIS

GETTIM
LIS

FIND
LIS

GETFIB
LIS

INIFC2
LIS

INIFCP
LIS

EXTFCB
LIS

FILESERV
LIS

FILESIZE
LIS

GETPTR
LIS