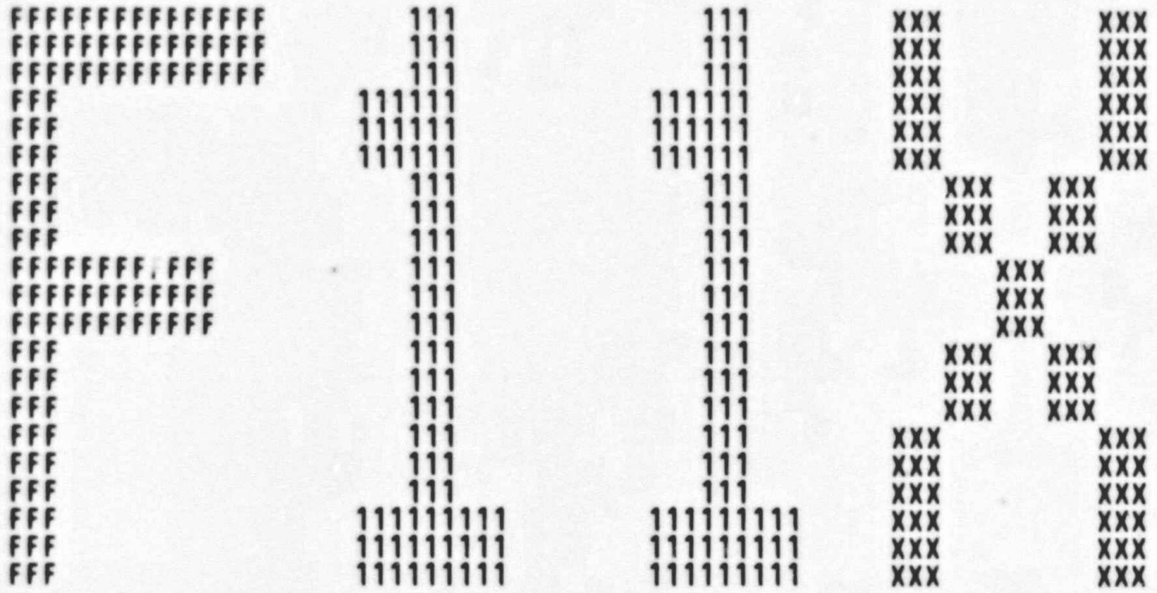

_$2

Symb

IOCS
IO_C
IO_C
IO_D
IO_F
IO_S
KICL

KILL
KILL
LB_B
LB_D
LB_F
LB_K
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

```
  GGGGGGGG  EEEEEEEEEE  TTTTTTTTTT  PPPPPPPP    TTTTTTTTTT  RRRRRRRR
  GGGGGGGG  EEEEEEEEEE  TTTTTTTTTT  PPPPPPPP    TTTTTTTTTT  RRRRRRRR
GG          EE              TT      PP      PP      TT      RR      RR
GG          EE              TT      PP      PP      TT      RR      RR
GG          EE              TT      PP      PP      TT      RR      RR
GG          EE              TT      PP      PP      TT      RR      RR
GG          EEEEEEEE        TT      PPPPPPPP        TT      RRRRRRRR
GG          EEEEEEEE        TT      PPPPPPPP        TT      RRRRRRRR
GG  GGGGGG  EE              TT      PP              TT      RR  RR
GG  GGGGGG  EE              TT      PP              TT      RR  RR
GG      GG  EE              TT      PP              TT      RR    RR        ....
GG      GG  EE              TT      PP              TT      RR    RR        ....
  GGGGGG    EEEEEEEEEE      TT      PP              TT      RR      RR      ....
  GGGGGG    EEEEEEEEEE      TT      PP              TT      RR      RR      ....

LL              IIIIII      SSSSSSSS
LL              IIIIII      SSSSSSSS
LL                II      SS
LL                II      SS
LL                II      SS
LL                II      SS
LL                II        SSSSSS
LL                II        SSSSSS
LL                II              SS
LL                II              SS
LL                II              SS
LL                II              SS
LLLLLLLLLL      IIIIII    SSSSSSSS
LLLLLLLLLL      IIIIII    SSSSSSSS
```

```
;     1          0001  0 MODULE GETPTR (
;     2          0002  0                 LANGUAGE (BLISS32),
;     3          0003  0                 IDENT = 'V04-000'
;     4          0004  0                 ) =
;     5          0005  1 BEGIN
;     6          0006  1
;     7          0007  1 !
;     8          0008  1 !**********************************************************************************
;     9          0009  1 !*                                                                                *
;    10          0010  1 !*  COPYRIGHT (c) 1978, 1980, 1982, 1984 BY                                       *
;    11          0011  1 !*  DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASSACHUSETTS.                         *
;    12          0012  1 !*  ALL RIGHTS RESERVED.                                                           *
;    13          0013  1 !*                                                                                *
;    14          0014  1 !*  THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND COPIED          *
;    15          0015  1 !*  ONLY IN  ACCORDANCE WITH  THE  TERMS  OF  SUCH  LICENSE  AND WITH THE        *
;    16          0016  1 !*  INCLUSION OF THE ABOVE COPYRIGHT NOTICE. THIS SOFTWARE OR  ANY  OTHER        *
;    17          0017  1 !*  COPIES THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY        *
;    18          0018  1 !*  OTHER PERSON.  NO TITLE TO AND OWNERSHIP OF  THE  SOFTWARE IS  HEREBY        *
;    19          0019  1 !*  TRANSFERRED.                                                                   *
;    20          0020  1 !*                                                                                *
;    21          0021  1 !*  THE INFORMATION IN THIS SOFTWARE IS  SUBJECT TO CHANGE WITHOUT NOTICE        *
;    22          0022  1 !*  AND  SHOULD  NOT  BE  CONSTRUED AS  A COMMITMENT BY DIGITAL EQUIPMENT        *
;    23          0023  1 !*  CORPORATION.                                                                   *
;    24          0024  1 !*                                                                                *
;    25          0025  1 !*  DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE  OR  RELIABILITY OF ITS        *
;    26          0026  1 !*  SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL.                        *
;    27          0027  1 !*                                                                                *
;    28          0028  1 !*                                                                                *
;    29          0029  1 !**********************************************************************************
;    30          0030  1
;    31          0031  1 !++
;    32          0032  1 !
;    33          0033  1 ! FACILITY:  F11ACP Structure Level 2
;    34          0034  1 !
;    35          0035  1 ! ABSTRACT:
;    36          0036  1 !
;    37          0037  1 !       This routine returns the value of a header map pointer.
;    38          0038  1 !
;    39          0039  1 ! ENVIRONMENT:
;    40          0040  1 !
;    41          0041  1 !       STARLET operating system, including privileged system services
;    42          0042  1 !       and internal exec routines.
;    43          0043  1 !
;    44          0044  1 !--
;    45          0045  1 !
;    46          0046  1 !
;    47          0047  1 ! AUTHOR:  Andrew C. Goldstein,  CREATION DATE:  21-Nov-1977  17:12
;    48          0048  1 !
;    49          0049  1 ! MODIFIED BY:
;    50          0050  1 !
;    51          0051  1 !       V03-001 CDS0001         Christian D. Saether    31-July-1984
;    52          0052  1 !               Define linkage in require file, remove local definition.
;    53          0053  1 !
;    54          0054  1 !       B0101   ACG0008         Andrew C. Goldstein,  26-Dec-1978  19:20
;    55          0055  1 !       Skip placement pointers (for placement support)
;    56          0056  1 !
;    57          0057  1 !       B0100   ACG00001        Andrew C. Goldstein,  10-Oct-1978  20:00
```

```
;    58          0058  1 !       Previous revision history moved to [F11B.SRC]F11B.REV
;    59          0059  1 !**
;    60          0060  1
;    61          0061  1
;    62          0062  1 LIBRARY 'SYS$LIBRARY:LIB.L32';
;    63          0063  1 REQUIRE 'SRC$:FCPDEF.B32';
;    64          1054  1
```

```
;    66          1055  1 GLOBAL ROUTINE GET_MAP_POINTER : L_MAP_POINTER NOVALUE =
;    67          1056  1
;    68          1057  1 !++
;    69          1058  1 !
;    70          1059  1 ! FUNCTIONAL DESCRIPTION:
;    71          1060  1 !
;    72          1061  1 !       This routine returns the contents of a file header map pointer and
;    73          1062  1 !       advances the map area pointer to the next map pointer.
;    74          1063  1 !
;    75          1064  1 !
;    76          1065  1 ! CALLING SEQUENCE:
;    77          1066  1 !       GET_MAP_POINTER ()
;    78          1067  1 !
;    79          1068  1 ! INPUT PARAMETERS:
;    80          1069  1 !       NONE
;    81          1070  1 !
;    82          1071  1 ! IMPLICIT INPUTS:
;    83          1072  1 !       R8 = address of header map pointer
;    84          1073  1 !
;    85          1074  1 ! OUTPUT PARAMETERS:
;    86          1075  1 !       NONE
;    87          1076  1 !
;    88          1077  1 ! IMPLICIT OUTPUTS:
;    89          1078  1 !       R6 = block count
;    90          1079  1 !       R7 = starting LBN
;    91          1080  1 !
;    92          1081  1 ! ROUTINE VALUE:
;    93          1082  1 !       NONE
;    94          1083  1 !
;    95          1084  1 ! SIDE EFFECTS:
;    96          1085  1 !       R8 advanced to next pointer (placement pointers are transparently skipped)
;    97          1086  1 !
;    98          1087  1 !--
;    99          1088  1
;   100          1089  2 BEGIN
;   101          1090  2
;   102          1091  2 EXTERNAL REGISTER
;   103          1092  2         COUNT           = 6,            ! retrieval pointer block count
;   104          1093  2         LBN             = 7,            ! retrieval pointer start LBN
;   105          1094  2         MAP_POINTER     = 8 : REF BBLOCK; ! address of map pointer
;   106          1095  2
;   107          1096  2
;   108          1097  2 ! Determine the type of the map pointer and interpret it appropriately.
;   109          1098  2 !
;   110          1099  2
;   111          1100  2 IF .MAP_POINTER[FM2$V_FORMAT] EQL FM2$C_PLACEMENT
;   112          1101  2 THEN MAP_POINTER = .MAP_POINTER + FM2$C_LENGTH0;
;   113          1102  2
;   114          1103  2 CASE .MAP_POINTER[FM2$V_FORMAT] FROM 0 TO 3 OF
;   115          1104  2     SET
;   116          1105  2
;   117          1106  3     [FM2$C_PLACEMENT]:  BEGIN
;   118          1107  3                         COUNT = 0;
;   119          1108  3                         LBN = 0;
;   120          1109  3                         MAP_POINTER = .MAP_POINTER + 2;
;   121          1110  2                         END;
;   122          1111  2
```

```
;  123        1112 3       [FM2$C_FORMAT1]:       BEGIN
;  124        1113 3                              COUNT = .MAP_POINTER[FM2$B_COUNT1];
;  125        1114 3                              LBN<16,16> = .MAP_POINTER[FM2$V_HIGHLBN];
;  126        1115 3                              LBN<0,16> = .MAP_POINTER[FM2$W_LOWLBN];
;  127        1116 3                              MAP_POINTER = .MAP_POINTER + 4;
;  128        1117 3                              COUNT = .COUNT + 1;
;  129        1118 2                              END;
;  130        1119 2
;  131        1120 3       [FM2$C_FORMAT2]:       BEGIN
;  132        1121 3                              COUNT = .MAP_POINTER[FM2$V_COUNT2];
;  133        1122 3                              LBN = .MAP_POINTER[FM2$L_LBN2];
;  134        1123 3                              MAP_POINTER = .MAP_POINTER + 6;
;  135        1124 3                              COUNT = .COUNT + 1;
;  136        1125 2                              END;
;  137        1126 2
;  138        1127 3       [FM2$C_FORMAT3]:       BEGIN
;  139        1128 3                              COUNT = (ROT (..MAP_POINTER, 16) AND (1^30-1));
;  140        1129 3                              LBN = .MAP_POINTER[FM2$L_LBN3];
;  141        1130 3                              MAP_POINTER = .MAP_POINTER + 8;
;  142        1131 3                              COUNT = .COUNT + 1;
;  143        1132 2                              END;
;  144        1133 2
;  145        1134 2       TES;
;  146        1135 2
;  147        1136 1 END;                                 ! end of routine GET_MAP_POINTER


                                                          .TITLE  GETPTR
                                                          .IDENT  \V04-000\


                                                          .PSECT  $CODE$,NOWRT,2

            C0     8F        01   A8  93 00000 GET_MAP_POINTER::
                                                          BITB    1(MAP_POINTER), #192                        ; 1100
                                      03  12 00005        BNEQ    1$                                          ;
                           58         02  C0 00007        ADDL2   #2, MAP_POINTER                             ; 1101
   51             68       02         0E  EF 0000A 1$:    EXTZV   #14, #2, (MAP_POINTER), R1                  ; 1103
                  03       00         51  CF 0000F        CASEL   R1, #0, #3                                  ;
 002E           0020     000E       0008     00013 2$:    .WORD   3$-2$,-                                     ;
                                                                  4$-2$,-                                     ;
                                                                  5$-2$,-                                     ;
                                                                  6$-2$                                       ;
                                      56  7C 0001B 3$:    CLRQ    COUNT                                       ; 1107
                           58         02  C0 0001D        ADDL2   #2, MAP_POINTER                             ; 1109
                                      05  00020           RSB                                                 ; 1103
                           56         88  9A 00021 4$:    MOVZBL  (MAP_POINTER)+, COUNT                       ; 1113
   50             88       06         00  EF 00024        EXTZV   #0, #6, (MAP_POINTER)+, R0                  ; 1114
   57             10       10         50  F0 00029        INSV    R0, #16, #16, LBN                           ;
                           57         88  B0 0002E        MOVW    (MAP_POINTER)+, LBN                         ; 1115
                                      1A  11 00031        BRB     7$                                          ; 1117
   56             88       0E         00  EF 00033 5$:    EXTZV   #0, #14, (MAP_POINTER)+, COUNT              ; 1121
                           57   01    A8  D0 00038        MOVL    1(MAP_POINTER), LBN                         ; 1122
                           58         05  C0 0003C        ADDL2   #5, MAP_POINTER                             ; 1123
                                      0C  11 0003F        BRB     7$                                          ; 1124
                  50       88         10  9C 00041 6$:    ROTL    #16, (MAP_POINTER)+, R0                     ; 1128
```

```
      56          50              1E            00  EF 00045          EXTZV   #0, #30, R0, COUNT          ;
                                  57            88  D0 0004A          MOVL    (MAP_POINTER)+, LBN         ; 1129
                                                56  D6 0004D 7$:      INCL    COUNT                       ; 1131
                                                    05 0004F          RSB                                 ; 1136

; Routine Size:  80 bytes,    Routine Base:  $CODE$ + 0000


;  148          1137  1
;  149          1138  1 END
;  150          1139  0 ELUDOM



;                          PSECT SUMMARY
;
;       Name                      Bytes                   Attributes
;
;  $CODE$                            80  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        29         0      1000          00:01.9



;                          COMMAND QUALIFIERS

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

; Size:           80 code + 0 data bytes
; Run Time:          00:08.9
; Elapsed Time:      00:19.8
; Lines/CPU Min:     7722
; Lexemes/CPU-Min:  26942
; Memory Used:  113 pages
; Compilation Complete
```

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