H 5

```
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
FFFFFFFFFFFF            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
```

L 16

```
**FILE**ID**INIFCP
```

```
  IIIIII    NN      NN     IIIIII    FFFFFFFFFF     CCCCCCCC  PPPPPPPP
  IIIIII    NN      NN     IIIIII    FFFFFFFFFF     CCCCCCCC  PPPPPPPP
    II      NN      NN       II      FF           CC          PP      PP
    II      NN      NN       II      FF           CC          PP      PP
    II      NNNN    NN       II      FF           CC          PP      PP
    II      NNNN    NN       II      FF           CC          PP      PP
    II      NN  NN  NN       II      FFFFFFFF     CC          PPPPPPPP
    II      NN  NN  NN       II      FFFFFFFF     CC          PPPPPPPP
    II      NN    NNNN       II      FF           CC          PP
    II      NN    NNNN       II      FF           CC          PP
    II      NN      NN       II      FF           CC          PP               ....
    II      NN      NN       II      FF           CC          PP               ....
  IIIIII    NN      NN     IIIIII    FF             CCCCCCCC  PP               ....
  IIIIII    NN      NN     IIIIII    FF             CCCCCCCC  PP               ....

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 INIFCP (
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 1
1 !
1 ! ABSTRACT:
1 !
1 !       This routine does the one time initialization for FCP.
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 ! AUTHOR:  Andrew C. Goldstein,  CREATION DATE:  20-Dec-1976  16:30
1 !
1 ! MODIFIED BY:
1 !
1 !       V03-011 CDS0007         Christian D. Saether     2-May-1984
1 !               Create bad block scanner mailbox as permanent.
1 !
1 !       V03-010 ACG0415         Andrew C. Goldstein,    12-Apr-1984  12:34
1 !               Fix end points of locked area descriptors
1 !
```

```
;    58        0058    1 !         V03-009 ACG0408          Andrew C. Goldstein,    24-Mar-1984  0:07
;    59        0059    1 !                 Misc bug fixes for storage reorganization
;    60        0060    1 !
;    61        0061    1 !         V03-008 ACG0408          Andrew C. Goldstein,    23-Mar-1984  12:03
;    62        0062    1 !                 Dynamically allocate impure storage on startup
;    63        0063    1 !
;    64        0064    1 !         V03-007 CDS0006          Christian D. Saether    19-Feb-1984
;    65        0065    1 !                 Remove reference to INIT_POOL.
;    66        0066    1 !                 Change external references to general mode.
;    67        0067    1 !
;    68        0068    1 !         V03-006 CDS0005          Christian D. Saether    12-Dec-1983
;    69        0069    1 !                 Move all GLOBAL data declarations to COMMON.
;    70        0070    1 !                 Eliminate most of the initialization routine in
;    71        0071    1 !                 conjunction with the reduction of image sections.
;    72        0072    1 !                 Get channel by calling IOC$FFCHAN directly instead
;    73        0073    1 !                 of using $ASSIGN (it isn't really assigned to
;    74        0074    1 !                 a specific device, anyway).
;    75        0075    1 !
;    76        0076    1 !         V03-005 CDS0004          Christian D. Saether    27-Aug-1983
;    77        0077    1 !                 Only assign one disk i/o channel.  Remember its ccb address.
;    78        0078    1 !
;    79        0079    1 !         V03-004 CDS0003          Christian D. Saether    26-Jul-1983
;    80        0080    1 !                 Eliminate creation of job controller mailbox.
;    81        0081    1 !
;    82        0082    1 !         V03-003 CDS0002          Christian D. Saether    15-Dec-1982
;    83        0083    1 !                 Remove some non-pic references.
;    84        0084    1 !
;    85        0085    1 !         V03-002 CDS0001          C Saether               18-Jul-1982
;    86        0086    1 !                 Changes to support ACP to XQP file system.
;    87        0087    1 !
;    88        0088    1 !         V03-001 LMP0037          L. Mark Pilant,         28-Jun-1982  15:10
;    89        0089    1 !                 Remove the addressing mode module switch.
;    90        0090    1 !
;    91        0091    1 !         V02-004 ACG0245          Andrew C. Goldstein,    23-Dec-1981  21:04
;    92        0092    1 !                 Add job controller mailbox
;    93        0093    1 !
;    94        0094    1 !         V02-003 LMP0004          L. Mark Pilant,          1-Dec-1981  12:10
;    95        0095    1 !                 Make external references use general mode
;    96        0096    1 !
;    97        0097    1 !         V02-002 ACG0167          Andrew C. Goldstein,    16-Apr-1980  19:26
;    98        0098    1 !                 Previous revision history moved to F11B.REV
;    99        0099    1 !**
;   100        0100    1
;   101        0101    1
;   102        0102    1 LIBRARY 'SYS$LIBRARY:LIB.L32';
;   103        0103    1 REQUIRE 'SRC$:FCPDEF.B32';
;   104        1094    1
;   105        1095    1 FORWARD ROUTINE
;   106        1096    1         INIT_FCP,                       ! initialize file system
;   107        1097    1         INIT_STORAGE    : L_NORM NOVALUE; ! initialize global storage
;   108        1098    1
;   109        1099    1 ! Dummy vectors to bracket the locked down code and data psects.
;   110        1100    1 !
;   111        1101    1
;   112        1102    1 PSECT   GLOBAL  = $AAAAA$ (NOWRITE, EXECUTE, ALIGN (9));
;   113        1103    1 GLOBAL  CODE_START      : VECTOR [0];
;   114        1104    1
;   115          1105  1 PSECT    GLOBAL   = $LOCKEDC0$ (NOWRITE, EXECUTE, ALIGN (9));
;   116          1106  1 GLOBAL   L_CODE_START     : VECTOR [0];
;   117          1107  1
;   118          1108  1 PSECT    GLOBAL   = $LOCKEDC9$ (NOWRITE, EXECUTE, ALIGN (2));
;   119          1109  1 GLOBAL   L_CODE_END       : VECTOR [0];
```

```
1 GLOBAL ROUTINE INIT_FCP =
1
1 !++
1 !
1 ! FUNCTIONAL DESCRIPTION:
1 !
1 !       This routine does the one time initialization for FCP.
1 !       It is called during process creation immediately after
1 !       the xqp code is mapped.
1 !
1 ! CALLING SEQUENCE:
1 !       INIT_FCP ()
1 !
1 ! INPUT PARAMETERS:
1 !       NONE
1 !
1 ! IMPLICIT INPUTS:
1 !       system I/O data base
1 !
1 ! OUTPUT PARAMETERS:
1 !       NONE
1 !
1 ! IMPLICIT OUTPUTS:
1 !       IO_CHANNEL: gets channel number of I/O channel
1 !       IO_CCB: gets ccb address of i/o channel
1 !
1 ! ROUTINE VALUE:
1 !       NONE
1 !
1 ! SIDE EFFECTS:
1 !       FCP hooked up to system data base
1 !
1 !--
1
2 BEGIN
2
2 LOCAL
2       STORAGE_DESC    : VECTOR [2],   ! descriptor for allocated storage
2       STATUS;                         ! system status return
2
2 GLOBAL REGISTER
2       BASE            = 10;
2
2 EXTERNAL
2       CTL$GL_CTLBASVA : ADDRESSING_MODE (ABSOLUTE),
2                                       ! base address of permanent P1 space
2       EXE$GL_FLAGS    : BITVECTOR ADDRESSING_MODE (ABSOLUTE);
2                                       ! system flags vector
2
2 EXTERNAL LITERAL
2       STORAGE_SIZE    : UNSIGNED (16), ! size of impure area in bytes
2       STORAGE_OFFSET  : UNSIGNED (16), ! offset to point base register at
2       EXE$V_INIT      : UNSIGNED (6); ! bit position of FCP init flag
2
2 ! First allocate the impure storage region.
2 !
```

```
IODC
Symb

ABD$
ABD$
ABD$
ABD$
ABD$
ABD$
ACB$
ACL_
AQB_
BITM
BUG$
CACH
CHEC
CHIN
CTL$
DATA
DIRE
FCB_
FIB$
HEAD
INDE
IO$_
IO$_
IOC$
IOC$
IO_D
IP[$
IPL$
IRP$
IRP$
IRP$
IRP$
IRP$
IRP$
IRP$
IRP$
IRP$
IRP$
IRP$
IRP$
IRP$
IRP$
IRP$
LOCA
MVL_
PACK
PR$_
PR$_
QUOT
RVT_
SCH$
UCB$
USER
VCB$
VCB_
WCB_
```

```
;  178               1167  2
;  179             P 1168  3 IF NOT (STATUS = $EXPREG (PAGCNT = (STORAGE_SIZE + 511) / 512,
;  180             P 1169  3                           RETADR = STORAGE_DESC,
;  181             P 1170  3                           REGION = 1
;  182               1171  3                           ))
;  183               1172  2 THEN $EXIT (CODE = .STATUS);
;  184               1173  2 BASE = .STORAGE_DESC[1] + STORAGE_OFFSET;
;  185               1174  2 CTL$GL_CTLBASVA = .STORAGE_DESC[1];
;  186             - 1175  2
;  187               1176  2 INIT_STORAGE ();
;  188               1177  2
;  189               1178  2 ! Finally set the FCP inited bit in the system flags word to indicate that
;  190               1179  2 ! a file system now exists (significant only during system startup).
;  191               1180  2 !
;  192               1181  2
;  193               1182  2 IF TESTBITCS (EXE$GL_FLAGS [EXE$V_INIT])
;  194               1183  2 THEN
;  195               1184  2
;  196               1185  2 ! This will happen when the xqp is merged into the sysinit process.
;  197               1186  2 ! It should have all the privileges we need to create this mailbox,
;  198               1187  2 ! so elevating and restoring them is not necessary.
;  199               1188  2 !
;  200               1189  2
;  201               1190  3     BEGIN
;  202               1191  3     LOCAL
;  203               1192  3         MBX_CHAN,
;  204               1193  3         DESC    : VECTOR [2];
;  205               1194  3
;  206               1195  3     PIC_DESC ('ACP$BADBLOCK_MBX', DESC );
;  207               1196  3
;  208             P 1197  3     IF NOT $CREMBX (CHAN = MBX_CHAN,
;  209             P 1198  3                     MAXMSG = BBS$C_LENGTH,
;  210             P 1199  3                     BUFQUO = BBS$C_LENGTH*100,
;  211             P 1200  3                     PROMSK = %X'FFFF',
;  212             P 1201  3                     LOGNAM = DESC,
;  213               1202  4                     PRMFLG = 1)
;  214               1203  3     THEN
;  215               1204  3         BUG_CHECK (XQPERR);
;  216               1205  3
;  217               1206  3     $DASSGN (CHAN = .MBX_CHAN);
;  218               1207  3
;  219               1208  2     END;
;  220               1209  2
;  221               1210  2 SS$_NORMAL
;  222               1211  2
;  223               1212  1 END;                                         ! end of routine INIT_FCP


                                                                          .TITLE  INIFCP
                                                                          .IDENT  \V04-000\

                                                                          .PSECT  $CODE$,NOWRT,2

42  4D  5F  4B  43  4F  4C  42  44  41  42  24  50  43  41  00000 P.AAA:  .ASCII  \ACP$BADBLOCK_MBX\                              ;
                                                        58  0000F                                                                 ;

                                                                          .PSECT  $LOCKEDC9$,NOWRT,2
```

```
IODC
Psec

PSE(
----
.  /
$ABS
$COD

Phas
----
Init
Comm
Pass
Symb
Pass
Symb
Psec
Cros
Asse

The
665C
Ther
302
22 p

Macr
----
_$2
-$2
TOTA

145

Ther

MACI
```

```
                                          00000 L_CODE_END::
                                                      .BLKB    0

                                                      .PSECT   $LOCKEDCO$,NOWRT,9

                                          00000 L_CODE_START::
                                                      .BLKB    0

                                                      .PSECT   $AAAAA$,NOWRT,9

                                          00000 CODE_START::
                                                      .BLKB    0

                                                      .EXTRN   CTL$GL_CTLBASVA
                                                      .EXTRN   EXE$GL_FLAGS, STORAGE_SIZE
                                                      .EXTRN   STORAGE_OFFSET, EXE$V_INIT
                                                      .EXTRN   SYS$EXPREG, SYS$EXIT
                                                      .EXTRN   SYS$CREMBX, BUG$_XQPERR
                                                      .EXTRN   SYS$DASSGN

                                                      .PSECT   $CODE$,NOWRT,2

                                     0400 00000       .ENTRY   INIT_FCP, Save R10                          ; 1110
               5E                14    C2 00002       SUBL2    #20, SP                                     :
                                 01    DD 00005       PUSHL    #1                                          ; 1171
                                 7E    D4 00007       CLRL     -(SP)                                       :
                        14       AE    9F 00009       PUSHAB   STORAGE_DESC                                :
                 00000000*       8F    DD 0000C       PUSHL    #<<STORAGE_SIZE+511>/512>                   :
      00000000G 00                 04  FB 00012       CALLS    #4, SYS$EXPREG                              :
                09                 50  E8 00019       BLBS     STATUS, 1$                                  :
                                   50  DD 0001C       PUSHL    STATUS                                      ; 1172
      00000000G 00                 01  FB 0001E       CALLS    #1, SYS$EXIT                                :
5A           10 AE   00000000G     8F  C1 00025 1$:   ADDL3    #STORAGE_OFFSET, STORAGE_DESC+4, BASE       ; 1173
      00000000G 9F          10     AE  D0 0002E       MOVL     STORAGE_DESC+4, @#CTL$GL_CTLBASVA           ; 1174
          0000V CF                 00  FB 00036       CALLS    #0, INIT_STORAGE                            ; 1176
36    00000000G 9F                00G  E2 0003B       BBSS     S^EXE$V_INIT, @#EXE$GL_FLAGS, 3$            ; 1182
             04 AE                 10  D0 00043       MOVL     #16, DESC                                   ; 1195
             08 AE          A6     AF  9E 00047       MOVAB    P.AAA, DESC+4                               :
                            04     AE  9F 0004C       PUSHAB   DESC                                        ; 1202
                                   7E  D4 0004F       CLRL     -(SP)                                       :
                7E        FFFF     8F  3C 00051       MOVZWL   #65535, -(SP)                               :
                7E        0708     8F  3C 00056       MOVZWL   #1800, -(SP)                                :
                                   12  DD 0005B       PUSHL    #18                                         :
                            14     AE  9F 0005D       PUSHAB   MBX_CHAN                                    :
                                   01  DD 00060       PUSHL    #1                                          :
      00000000G 00                 07  FB 00062       CALLS    #7, SYS$CREMBX                              :
                04                 50  E8 00069       BLBS     R0, 2$                                      :
                                     FEFF 0006C       BUGW                                                 ; 1204
                                    0000* 0006E       .WORD    <BUG$_XQPERR!4>                             :
                                   6E  DD 00070 2$:   PUSHL    MBX_CHAN                                    ; 1206
      00000000G 00                 01  FB 00072       CALLS    #1, SYS$DASSGN                              :
                50                 01  D0 00079 3$:   MOVL     #1, R0                                      ; 1212
                                       04 0007C       RET                                                  :

; Routine Size:  125 bytes,    Routine Base:  $CODE$ + 0010
```

**F]

```
1 GLOBAL ROUTINE INIT_STORAGE : L_NORM NOVALUE =
1
1 !++
1 !
1 ! FUNCTIONAL DESCRIPTION:
1 !
1 !       This routine initializes the file system's global impure area.
1 !
1 ! CALLING SEQUENCE:
1 !       INIT_STORAGE ()
1 !
1 ! INPUT PARAMETERS:
1 !       NONE
1 !
1 ! IMPLICIT INPUTS:
1 !       system I/O data base
1 !
1 ! OUTPUT PARAMETERS:
1 !       NONE
1 !
1 ! IMPLICIT OUTPUTS:
1 !       IO_CHANNEL: gets channel number of I/O channel
1 !       IO_CCB: gets ccb address of i/o channel
1 !
1 ! ROUTINE VALUE:
1 !       NONE
1 !
1 ! SIDE EFFECTS:
1 !       FCP hooked up to system data base
1 !
1 !--
1
2 BEGIN
2
2 LOCAL
2       LOCKED_DESC     : VECTOR [2],   ! descriptor for locked down pages
2       STATUS;                         ! system status return
2
2 BIND_COMMON;
2
2 EXTERNAL
2       CTL$GL_F11BXQP  : ADDRESSING_MODE (ABSOLUTE);
2                                       ! pointer to XQP
2
2 EXTERNAL LITERAL
2       STORAGE_SIZE    : UNSIGNED (16), ! size of impure area in bytes
2       STORAGE_OFFSET  : UNSIGNED (16); ! offset to point base register at
2
2 LINKAGE
2       L_FFCHAN        = JSB   : GLOBAL (CHANNEL=1, CCB=2);
2
2 GLOBAL REGISTER
2       CHANNEL = 1,
2       CCB     = 2     : REF BBLOCK;
2
2 EXTERNAL ROUTINE
2       IOC$FFCHAN      : L_FFCHAN ADDRESSING_MODE (GENERAL),
2                                               ! find free channel
2            DISPATCH;                          ! main dispatch routine
2
2
2 ! Now lock appropriate areas into the working set. These are code and data
2 ! that are used at raised IPL, plus the private kernel stack.
2 !
2
2 LOCKED_DESC [0] = L_CODE_START;
2 LOCKED_DESC [1] = L_CODE_END - 1;
2
2 STATUS = $LKWSET (INADR = LOCKED_DESC);
2 IF NOT .STATUS THEN $EXIT (CODE = .STATUS);
2
2 LOCKED_DESC [0] = L_DATA_START;
2 LOCKED_DESC [1] = L_DATA_END - 1;
2
2 STATUS = $LKWSET (INADR = LOCKED_DESC);
2 IF NOT .STATUS THEN $EXIT (CODE = .STATUS);
2
2 ! Find an I/O channel for use by the file system.
2 !
2
2 IF NOT IOC$FFCHAN ()
2 THEN
2     BUG_CHECK (NOACPCHAN, 'Failed to find channel for XQP');
2
2 CCB [CCB$B_AMOD] = -1;
2
2 IO_CCB = .CCB;
2 IO_CHANNEL = .CHANNEL;
2
2 ! Initialize the rest of the impure storage area.
2 !
2
2 CODE_SIZE = L_CODE_END - CODE_START;
2 CODE_ADDRESS = CODE_START;
2 DATA_SIZE = STORAGE_SIZE;
2 DATA_ADDRESS = STORAGE_START;
2
2 XQP_STKLIM [0] = XQP_QUEUE;
2 XQP_STKLIM [1] = XQP_STACK;
2
2 ! Set up the XQP queue head and dispatcher addresses.
2 !
2
2 XQP_QUEUE [0] = XQP_QUEUE;
2 XQP_QUEUE [1] = XQP_QUEUE;
2 XQP_DISPATCHER = DISPATCH;
2
2 CTL$GL_F11BXQP = XQP_QUEUE;
2
1 END;                                          ! end of routine INIT_STORAGE


                                                        .EXTRN  CTL$GL_F11BXQP, IOC$FFCHAN
```

```
                                                      .EXTRN   DISPATCH, SYS$LKWSET
                                                      .EXTRN   BUG$_NOACPCHAN

                                    0BFC 00000        .ENTRY   INIT_STORAGE, Save R2,R3,R4,R5,R6,R7,R8,R9,-; 1213
                                                               R11                                         ;
                  57 00000000G   00   9E 00002        MOVAB    SYS$EXIT, R7                                ;
                  56 00000000G   00   9E 00009        MOVAB    SYS$LKWSET, R6                              ;
                  5E             04   C2 00010        SUBL2    #4, SP                                      ;
                  55      F540   CA   9E 00013        MOVAB    -2752(BASE), R5                             ; 1249
                  53      FF40   CA   9E 00018        MOVAB    -192(BASE), R3                              ;
                  54      FF68   CA   9E 0001D        MOVAB    -152(BASE), R4                              ;
                          0000'  CF   9F 00022        PUSHAB   L_CODE_START                                ; 1278
            04    AE      0000'  CF   9E 00026        MOVAB    L_CODE_END-1, LOCKED_DESC+4                 ; 1279
                                 7E   7C 0002C        CLRQ     -(SP)                                       ; 1281
                            08   AE   9F 0002E        PUSHAB   LOCKED_DESC                                 ;
                  66             03   FB 00031        CALLS    #3, SYS$LKWSET                              ;
                  52             50   D0 00034        MOVL     R0, STATUS                                  ;
                  05             52   E8 00037        BLBS     STATUS, 1$                                  ; 1282
                                 52   DD 0003A        PUSHL    STATUS                                      ;
                  67             01   FB 0003C        CALLS    #1, SYS$EXIT                                ;
                  6E             55   D0 0003F  1$:   MOVL     R5, LOCKED_DESC                             ; 1284
            04    AE      02B3   CA   9E 00042        MOVAB    691(BASE), LOCKED_DESC+4                    ; 1285
                                 7E   7C 00048        CLRQ     -(SP)                                       ; 1287
                            08   AE   9F 0004A        PUSHAB   LOCKED_DESC                                 ;
                  66             03   FB 0004D        CALLS    #3, SYS$LKWSET                              ;
                  52             50   D0 00050        MOVL     R0, STATUS                                  ;
                  05             52   E8 00053        BLBS     STATUS, 2$                                  ; 1288
                                 52   DD 00056        PUSHL    STATUS                                      ;
                  67             01   FB 00058        CALLS    #1, SYS$EXIT                                ;
                     00000000G   00   16 0005B  2$:   JSB      IOC$FFCHAN                                  ; 1293
                  04             50   E8 00061        BLBS     R0, 3$                                      ;
                                    FEFF 00064        BUGW                                                 ; 1295
                                   0000* 00066        .WORD    <BUG$_NOACPCHAN!4>                          ;
            09    A2             01   8E 00068  3$:   MNEGB    #1, 9(CCB)                                  ; 1297
          FF74    CA             52   D0 0006C        MOVL     CCB, -140(BASE)                             ; 1299
          FF78    CA             51   D0 00071        MOVL     CHANNEL, -136(BASE)                         ; 1300
          FF4C    CA 00000000*   8F   D0 00076        MOVL     #<L_CODE_END-CODE_START>, -180(BASE)        ; 1305
          FF50    CA      0000'  CF   9E 0007F        MOVAB    CODE_START, -176(BASE)                      ; 1306
          FF54    CA      0000G  8F   3C 00086        MOVZWL   #STORAGE_SIZE, -172(BASE)                   ; 1307
          FF58    CA             55   D0 0008D        MOVL     R5, -168(BASE)                              ; 1308
                  64             53   D0 00092        MOVL     R3, (R4)                                    ; 1310
            04    A4             55   D0 00095        MOVL     R5, 4(R4)                                   ; 1311
                  63             53   D0 00099        MOVL     R3, (R3)                                    ; 1316
            04    A3             53   D0 0009C        MOVL     R3, 4(R3)                                   ; 1317
          FF48    CA      0000G  CF   9E 000A0        MOVAB    DISPATCH, -184(BASE)                        ; 1318
     00000000G    9F             53   D0 000A7        MOVL     R3, @#CTL$GL_F11BXQP                        ; 1320
                                      04 000AE        RET                                                  ; 1322

; Routine Size:  175 bytes,    Routine Base:  $CODE$ + 008D


;  335          1323  1
;  336          1324  1 END
;  337          1325  0 ELUDOM
```

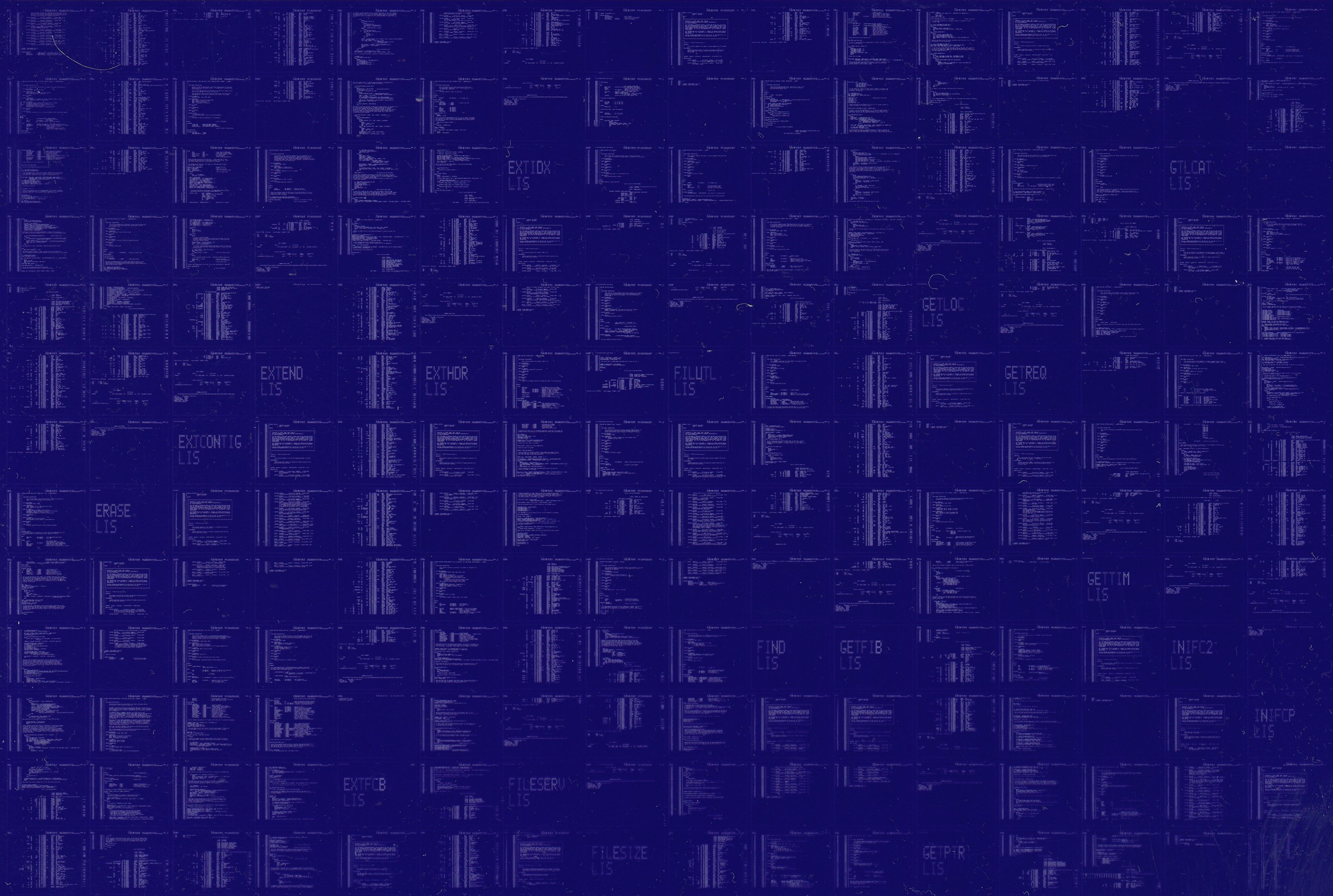
```
LOCK
Symb

ACL_
AQB_
BITM
CACH
CHIP
DATA
DIRE
FCB
HEAD
INDE
IOC$
LOCK
MVL_
PR$
QUOT
RVT
SCH$
SCH$
SCH$
UNLO
VCB_
WCB_



PSEC
----
.  A
$ABS
$LOC



Phas
----
Init
Comm
Pass
Symb
Pass
Symb
Psec
Cros
Asse

The
6388
Ther
234
11 p
```

```
;                                   PSECT SUMMARY
;
;       Name                     Bytes                    Attributes
;
;    $AAAAAA$                        0  NOVEC,NOWRT,  RD ,  EXE,NOSHR,  LCL,  REL,  CON,NOPIC,ALIGN(9)
;    $LOCKEDC0$                      0  NOVEC,NOWRT,  RD ,  EXE,NOSHR,  LCL,  REL,  CON,NOPIC,ALIGN(9)
;    $LOCKEDC9$                      0  NOVEC,NOWRT,  RD ,  EXE,NOSHR,  LCL,  REL,  CON,NOPIC,ALIGN(2)
;    $CODE$                        316  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        27         0      1000         00:01.9





;                                   COMMAND QUALIFIERS

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

; Size:           300 code + 16 data bytes
; Run Time:          00:19.2
; Elapsed Time:      00:36.8
; Lines/CPU Min:      4144
; Lexemes/CPU-Min: 48519
; Memory Used:  201 pages
; Compilation Complete
```

```
LOCK
VAX-



Macr
----
_$25
_$25
TOTA

146

Ther

MACR
```

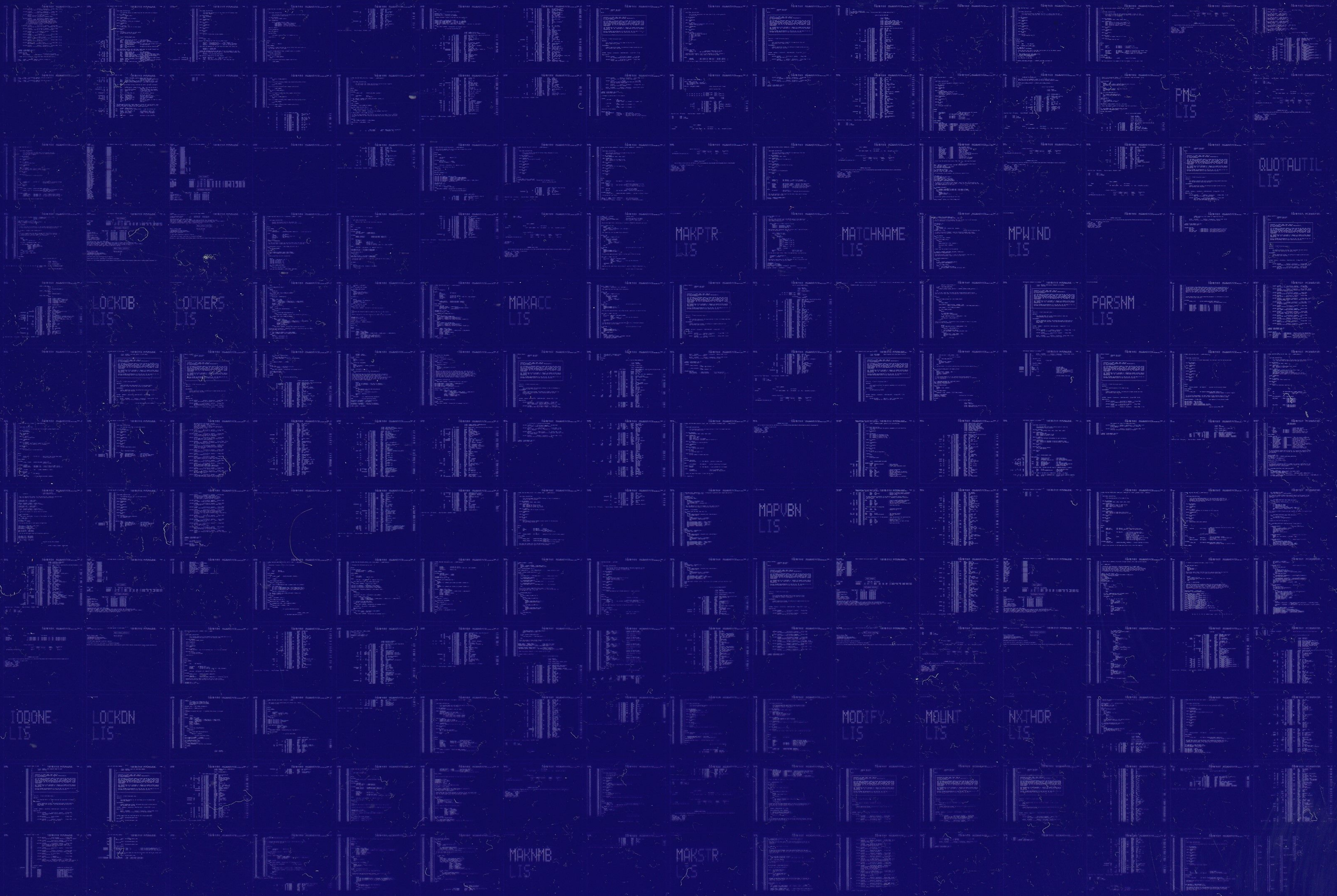
0170 AH-BT13A-SE
VAX/VMS V4.0

DIGITAL EQUIPMENT CORPORATION
CONFIDENTIAL AND PROPRIETARY



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