

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	XXX
FFFFFFFFFFFF	111	111	XXX	XXX
FFFFFFFFFFFF	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	111	111	XXX	XXX
FFF	111	111	XXX	XXX
FFF	1111111111	1111111111	XXX	XXX
FFF	1111111111	1111111111	XXX	XXX
FFF	1111111111	1111111111	XXX	XXX

_\$25
Symt

IOCI
IO_C
IO_C
IO_C
IO_F
IO_S
KICL
KILL
KILL
LB_E
LB_C
LB_F
LB_P
LB_L
LOCA
LOCK
LOCK
LOCK
LOCK
LOC_
LOC_
L_CC
L_CC
L_DA
L_DA
MAIA
MAKE
MAKE
MAKE
MAKE
MAKE
MAKE
MAKE
MAKE
MAKE
MAP_
MAP_
MAP_
MARI
MARI
MARI
MARI

```

IIIIII  NN  NN  IIIIII  FFFFFFFF  CCCCCC  PPPPPPP
IIIIII  NN  NN  IIIIII  FFFFFFFF  CCCCCC  PPPPPPP
  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  II       FFFFFFFF  CC      PPPPPPP
  II    NN  NN  II       FFFFFFFF  CC      PPPPPPP
  II    NN  NN  II       FF          CC      PP
  II    NN  NN  II       FF          CC      PP
  II    NN  NN  II       FF          CC      PP
  II    NN  NN  II       FF          CC      PP
  II    NN  NN  IIIIII  FF          CC      PP
IIIIII  NN  NN  IIIIII  FF          CCCCCC  PP
IIIIII  NN  NN  IIIIII  FF          CCCCCC  PP

```

```

LL      IIIIII  SSSSSSS
LL      IIIIII  SSSSSSS
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
LLLLLLLL  IIIIII  SSSSSSS
LLLLLLLL  IIIIII  SSSSSSS

```

```

1 0001 0 MODULE INIFCP (
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 1
34 0034 1
35 0035 1 ABSTRACT:
36 0036 1
37 0037 1 This routine does the one time initialization for FCP.
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. This routine must be called
43 0043 1 in kernel mode.
44 0044 1
45 0045 1 --
46 0046 1
47 0047 1
48 0048 1 AUTHOR: Andrew C. Goldstein, CREATION DATE: 20-Dec-1976 16:30
49 0049 1
50 0050 1 MODIFIED BY:
51 0051 1
52 0052 1 V03-011 CDS0007 Christian D. Saether 2-May-1984
53 0053 1 Create bad block scanner mailbox as permanent.
54 0054 1
55 0055 1 V03-010 ACG0415 Andrew C. Goldstein, 12-Apr-1984 12:34
56 0056 1 Fix end points of locked area descriptors
57 0057 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 (ot channel by calling IOCSFFCHAN directly instead
73 0073 1 of using SASSIGN (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 ACG0165 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 'SYSS$LIBRARY:LIB.L32';
103 0103 1 REQUIRE 'SRCB:F0PDEF.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 = SAAAAAS (NOWRITE, EXECUTE, ALIGN (9));
113 1103 1 GLOBAL CODE_START . VECTOR [0];
114 1104 1

```

INIFCP
V04-000

6 1
16-Sep-1984 00:37:40
14-Sep-1984 12:30:32

VAX-11 Bliss-32 V4.0-742 Page 3
DISK&VMSMASTER:[FIX.SRC]INIFCP.B32;1 (1)

1001
V04.

```
: 115      1105 1 PSECT GLOBAL = BLOCKEDCS (NOWRITE, EXECUTE, ALIGN (9));  
: 116      1106 1 GLOBAL L_CODE_START : VECTOR [0];  
: 117      1107 1  
: 118      1108 1 PSECT GLOBAL = BLOCKEDCS (NOWRITE, EXECUTE, ALIGN (2));  
: 119      1109 1 GLOBAL L_CODE_END : VECTOR [0];
```

```

121 1110 1 GLOBAL ROUTINE INIT_FCP =
122 1111 1
123 1112 1 **
124 1113 1
125 1114 1 FUNCTIONAL DESCRIPTION:
126 1115 1
127 1116 1 This routine does the one time initialization for FCP.
128 1117 1 It is called during process creation immediately after
129 1118 1 the xqp code is mapped.
130 1119 1
131 1120 1 CALLING SEQUENCE:
132 1121 1 INIT_FCP ()
133 1122 1
134 1123 1 INPUT PARAMETERS:
135 1124 1 NONE
136 1125 1
137 1126 1 IMPLICIT INPUTS:
138 1127 1 system I/O data base
139 1128 1
140 1129 1 OUTPUT PARAMETERS:
141 1130 1 NONE
142 1131 1
143 1132 1 IMPLICIT OUTPUTS:
144 1133 1 IO_CHANNEL: gets channel number of I/O channel
145 1134 1 IO_CCB: gets ccb address of i/o channel
146 1135 1
147 1136 1 ROUTINE VALUE:
148 1137 1 NONE
149 1138 1
150 1139 1 SIDE EFFECTS:
151 1140 1 FCP hooked up to system data base
152 1141 1
153 1142 1 --
154 1143 1
155 1144 2 BEGIN
156 1145 2
157 1146 2 LOCAL
158 1147 2 STORAGE_DESC : VECTOR [2], ! descriptor for allocated storage
159 1148 2 STATUS; ! system status return
160 1149 2
161 1150 2 GLOBAL REGISTER
162 1151 2 BASE = 10;
163 1152 2
164 1153 2 EXTERNAL
165 1154 2 CTL$GL_CTLBASVA : ADDRESSING_MODE (ABSOLUTE),
166 1155 2 ! base address of permanent P1 space
167 1156 2 EXE$GL_FLAGS : BITVECTOR ADDRESSING_MODE (ABSOLUTE);
168 1157 2 ! system flags vector
169 1158 2
170 1159 2 EXTERNAL LITERAL
171 1160 2 STORAGE_SIZE : UNSIGNED (16), ! size of impure area in bytes
172 1161 2 STORAGE_OFFSET : UNSIGNED (16), ! offset to point base register at
173 1162 2 EXE$V_INIT : UNSIGNED (6); ! bit position of FCP init flag
174 1163 2
175 1164 2
176 1165 2 ; First allocate the impure storage region.
177 1166 2

```

ABDI
ABDI
ABDI
ABDI
ACBI
ACL
AGB
BIT
BUG
CAC
CHE
CHI
CTL
DAT
DIR
FCB
FIB
HEA
IND
IOS
IOS
IOC
IOC
IO
IPC
IPL
IRP
IRP
IRP
IRP
IRP
IRP
IRP
IRP
IRP
IRP
IRP
LOC
MVL
PAC
PR
PR
QUO
RVT
SCH
UCB
USE
VCB
VCB
WCB

```
178  
179  
180  
181  
182  
183  
184  
185  
186  
187  
188  
189  
190  
191  
192  
193  
194  
195  
196  
197  
198  
199  
200  
201  
202  
203  
204  
205  
206  
207  
208  
209  
210  
211  
212  
213  
214  
215  
216  
217  
218  
219  
220  
221  
222  
223
```

```
1167  
1168  
1169  
1170  
1171  
1172  
1173  
1174  
1175  
1176  
1177  
1178  
1179  
1180  
1181  
1182  
1183  
1184  
1185  
1186  
1187  
1188  
1189  
1190  
1191  
1192  
1193  
1194  
1195  
1196  
1197  
1198  
1199  
1200  
1201  
1202  
1203  
1204  
1205  
1206  
1207  
1208  
1209  
1210  
1211  
1212
```

```
IF NOT (STATUS = $EXPREG (PAGCNT = (STORAGE_SIZE + 511) / 512,  
RETADR = STORAGE_DESC,  
REGION = 1  
))  
THEN $EXIT (CODE = .STATUS);  
BASE = .STORAGE_DESC[1] + STORAGE_OFFSET;  
CTL$GL_CTLBASVA = .STORAGE_DESC[1];  
  
INIT_STORAGE ();  
  
! Finally set the FCP init'd bit in the system flags word to indicate that  
! a file system now exists (significant only during system startup).  
  
IF TESTBITS (EXESGL_FLAGS [EXESV_INIT])  
THEN  
  
! This will happen when the xgp is merged into the sysinit process.  
! It should have all the privileges we need to create this mailbox,  
! so elevating and restoring them is not necessary.  
  
BEGIN  
LOCAL  
  MBX_CHAN,  
  DESC : VECTOR [2];  
  
PIC_DESC ('ACPS$BADBLOCK_MBX', DESC );  
  
IF NOT $CREMBX (CHAN = MBX_CHAN,  
  MAXMSG = BBSSC_LENGTH,  
  BUFQUO = BBSSC_LENGTH*100,  
  PROMSK = 'X'FFFF',  
  LOGNAM = DESC,  
  PRMFLG = 1)  
THEN  
  BUG_CHECK (XQPERR);  
  
$DASSGN (CHAN = .MBX_CHAN);  
  
END;  
  
SS$NORMAL  
  
END;
```

! end of routine INIT_FCP

```
.TITLE INIFCP  
.IDENT \V04-000\  
.PSECT $CODE$,NOWRT,2  
.ASCII \ACPS$BADBLOCK_MBX\  
.PSECT $LOCKEDC9$,NOWRT,2
```

```
42 4D 5F 4B 43 4F 4C 42 44 41 42 24 50 43 41 0000 P.AAA:  
58 000F
```

1000
Psect
PSECT
\$AB
\$CO
Pha
Init
Com
Pas
Sym
Pas
Sym
Psect
Cro
Ass
The
665
The
302
22
Mac
\$2
\$2
TOT
145
The
MAC

```

00000 L_CODE_END::
        .BLKB 0
        .PSECT $LOCKEDCOS,NOWRT,9

00000 L_CODE_START::
        .BLKB 0
        .PSECT $AAAAAS,NOWRT,9

00000 CODE_START::
        .BLKB 0

        .EXTRN CTLSGL_CTLBASVA
        .EXTRN EXESGL_FLAGS, STORAGE_SIZE
        .EXTRN STORAGE_OFFSET, EXESV_INIT
        .EXTRN SYS$EXPREG, SYS$EXIT
        .EXTRN SYS$CREMBX, BUGS_XQPERR
        .EXTRN SYSSDASSGN

        .PSECT $CODE$,NOWRT,2

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

```

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

INIFCP
V04-000

6 1
16-Sep-1984 00:37:40
14-Sep-1984 12:30:32

VAX-11 Bliss-32 V4.0-742
DISK\$VMSMASTER:[FIX.SRC]INIFCP.B32;1 Page 7 (2)

LOCK
V04-

```

225 1213 1 GLOBAL ROUTINE INIT_STORAGE : L_NORM NOVALUE =
226 1214 1
227 1215 1 **
228 1216 1
229 1217 1 FUNCTIONAL DESCRIPTION:
230 1218 1
231 1219 1     This routine initializes the file system's global impure area.
232 1220 1
233 1221 1 CALLING SEQUENCE:
234 1222 1     INIT_STORAGE ()
235 1223 1
236 1224 1 INPUT PARAMETERS:
237 1225 1     NONE
238 1226 1
239 1227 1 IMPLICIT INPUTS:
240 1228 1     system I/O data base
241 1229 1
242 1230 1 OUTPUT PARAMETERS:
243 1231 1     NONE
244 1232 1
245 1233 1 IMPLICIT OUTPUTS:
246 1234 1     IO_CHANNEL: gets channel number of I/O channel
247 1235 1     IO_CCB: gets ccb address of i/o channel
248 1236 1
249 1237 1 ROUTINE VALUE:
250 1238 1     NONE
251 1239 1
252 1240 1 SIDE EFFECTS:
253 1241 1     FCP hooked up to system data base
254 1242 1
255 1243 1 --
256 1244 1
257 1245 2 BEGIN
258 1246 2
259 1247 2 LOCAL
260 1248 2     LOCKED_DESC      : VECTOR [2],      ! descriptor for locked down pages
261 1249 2     STATUS:          : system status return
262 1250 2
263 1251 2 BIND_COMMON;
264 1252 2
265 1253 2 EXTERNAL
266 1254 2     CTL$GL_F11BXQP  : ADDRESSING_MODE (ABSOLUTE);
267 1255 2                       ! pointer to XQP
268 1256 2
269 1257 2 EXTERNAL LITERAL
270 1258 2     STORAGE_SIZE    : UNSIGNED (16), ! size of impure area in bytes
271 1259 2     STORAGE_OFFSET  : UNSIGNED (16); ! offset to point base register at
272 1260 2
273 1261 2 LINKAGE
274 1262 2     L_FFCHAN        = JSB      : GLOBAL (CHANNEL=1, CCB=2);
275 1263 2
276 1264 2 GLOBAL REGISTER
277 1265 2     CHANNEL = 1,
278 1266 2     CCB     = 2      : REF BBLOCK;
279 1267 2
280 1268 2 EXTERNAL ROUTINE
281 1269 2     IOC$FFCHAN      : L_FFCHAN ADDRESSING_MODE (GENERAL),

```

```
282 1270 2
283 1271 DISPATCH;
284 1272
285 1273
286 1274 ! Now lock appropriate areas into the working set. These are code and data
287 1275 ! that are used at raised IPL, plus the private kernel stack.
288 1276
289 1277
290 1278 LOCKED_DESC [0] = L_CODE_START;
291 1279 LOCKED_DESC [1] = L_CODE_END - 1;
292 1280
293 1281 STATUS = $LKWSET (INADR = LOCKED_DESC);
294 1282 IF NOT .STATUS THEN $EXIT (CODE = .STATUS);
295 1283
296 1284 LOCKED_DESC [0] = L_DATA_START;
297 1285 LOCKED_DESC [1] = L_DATA_END - 1;
298 1286
299 1287 STATUS = $LKWSET (INADR = LOCKED_DESC);
300 1288 IF NOT .STATUS THEN $EXIT (CODE = .STATUS);
301 1289
302 1290 ! Find an I/O channel for use by the file system.
303 1291
304 1292
305 1293 IF NOT IOC$FFCHAN ()
306 1294 THEN
307 1295     BUG_CHECK (NOACPCHAN, 'Failed to find channel for XQP');
308 1296
309 1297 CCB [CCBSB_AMOD] = -1;
310 1298
311 1299 IO_CCB = .CCB;
312 1300 IO_CHANNEL = .CHANNEL;
313 1301
314 1302 ! Initialize the rest of the impure storage area.
315 1303
316 1304
317 1305 CODE_SIZE = L_CODE_END - CODE_START;
318 1306 CODE_ADDRESS = CODE_START;
319 1307 DATA_SIZE = STORAGE_SIZE;
320 1308 DATA_ADDRESS = STORAGE_START;
321 1309
322 1310 XQP_STKLIM [0] = XQP_QUEUE;
323 1311 XQP_STKLIM [1] = XQP_STACK;
324 1312
325 1313 ! Set up the XQP queue head and dispatcher addresses.
326 1314
327 1315
328 1316 XQP_QUEUE [0] = XQP_QUEUE;
329 1317 XQP_QUEUE [1] = XQP_QUEUE;
330 1318 XQP_DISPATCHER = DISPATCH;
331 1319
332 1320 CTL$GL_F11BXQP = XQP_QUEUE;
333 1321
334 1322 1 END;
! end of routine INIT_STORAGE
```

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

OBFC 00000				.EXTRN DISPATCH, SYS\$LKWSET	
				.EXTRN BUG\$_NOACPCHAN	
				.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	FF7C	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	CLRO -(SP)	1281
	08	AE	9F 0002E	PUSHAB LOCKED_DESC	
66		03	FB 00031	CALLS #3, SYS\$LKWSET	
52		50	DO 00034	MOVL R0, STATUS	
05		52	EB 00037	BLBS STATUS, 1\$	1282
		52	DD 0003A	PUSHL STATUS	
67		01	FB 0003C	CALLS #1, SYS\$EXIT	
04	AE 02B3	55	DO 0003F 1\$:	MOVL R5, LOCKED_DESC	1284
		CA	9E 00042	MOVAB 691(BASE), LOCKED_DESC+4	1285
		7E	7C 00048	CLRO -(SP)	1287
	08	AE	9F 0004A	PUSHAB LOCKED_DESC	
66		03	FB 0004D	CALLS #3, SYS\$LKWSET	
52		50	DO 00050	MOVL R0, STATUS	
05		52	EB 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	EB 00061	BLBS R0, 3\$	
		FEFF	00064	BUGW	1295
		0000*	00066	.WORD <BUG\$_NOACPCHAN!4>	
09	A2	01	8E 00068 3\$:	MNEGB #1, 9T(CB)	1297
FF74	CA	52	DO 0006C	MOVL (CB, -140(BASE)	1299
FF78	CA	51	DO 00071	MOVL CHANNEL, -136(BASE)	1300
FF4C	CA 00000000*	8F	DO 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	DO 0008D	MOVL R5, -168(BASE)	1308
		53	DO 00092	MOVL R3, (R4)	1310
04	A4	55	DO 00095	MOVL R5, 4(R4)	1311
		53	DO 00099	MOVL R3, (R3)	1316
04	A3	53	DO 0009C	MOVL R3, 4(R3)	1317
FF48	CA 0000G	CF	9E 000A0	MOVAB DISPATCH, -184(BASE)	1318
00000000G	9F	53	DO 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

PSECT SUMMARY

Name	Bytes	Attributes
SAAAAAS	0	NOVEC,NOWRT, RD, EXE,NOSHR, LCL, REL, CON,NOPIC,ALIGN(9)
SLOCKEDC0S	0	NOVEC,NOWRT, RD, EXE,NOSHR, LCL, REL, CON,NOPIC,ALIGN(9)
SLOCKEDC9S	0	NOVEC,NOWRT, RD, EXE,NOSHR, LCL, REL, CON,NOPIC,ALIGN(2)
S_CODES	316	NOVEC,NOWRT, RD, EXE,NOSHR, LCL, REL, CON,NOPIC,ALIGN(2)

Macr

-S25
-S25
TOTA

146

Ther

MACR

Library Statistics

File	Symbols		Pages Mapped	Processing Time
	Total	Loaded Percent		
_S255\$DUA28:[SYSLIB]LIB.L32;1	18619	27 0	1000	00:01.9

COMMAND QUALIFIERS

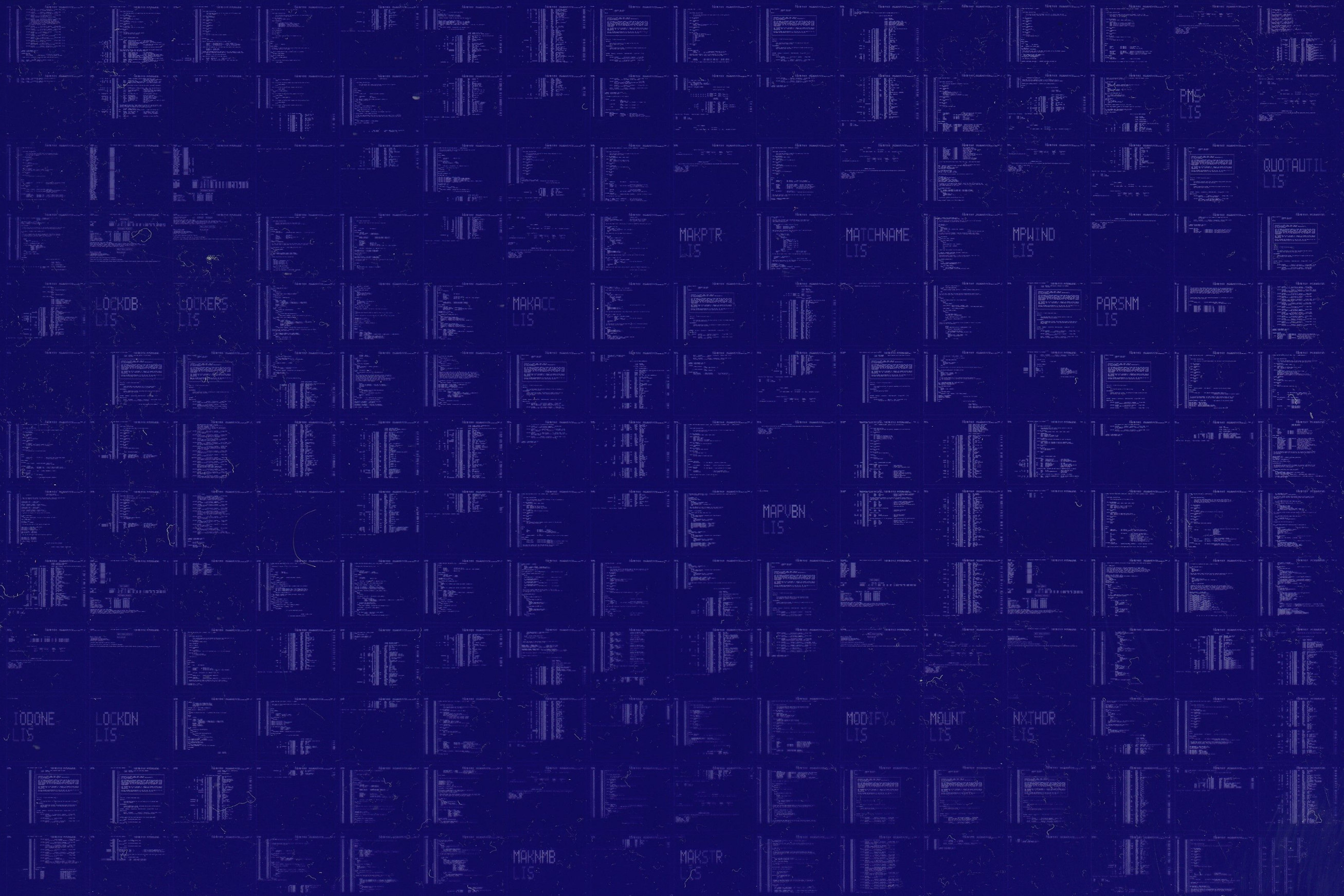
BLISS/CHECK=(FIELD,INITIAL,OPTIMIZE)/LIS=LIS\$:INIFCP/OBJ=OBJ\$:INIFCP MSRC\$:INIFCP/UPDATE=(ENH\$:INIFCP)

: 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

SCREENSHOT 1	SCREENSHOT 2	SCREENSHOT 3	SCREENSHOT 4	SCREENSHOT 5	SCREENSHOT 6	SCREENSHOT 7	SCREENSHOT 8	SCREENSHOT 9	SCREENSHOT 10
SCREENSHOT 11	SCREENSHOT 12	SCREENSHOT 13	SCREENSHOT 14	SCREENSHOT 15	SCREENSHOT 16	SCREENSHOT 17	SCREENSHOT 18	SCREENSHOT 19	SCREENSHOT 20
SCREENSHOT 21	SCREENSHOT 22	SCREENSHOT 23	SCREENSHOT 24	SCREENSHOT 25	SCREENSHOT 26	SCREENSHOT 27	SCREENSHOT 28	SCREENSHOT 29	SCREENSHOT 30
SCREENSHOT 31	SCREENSHOT 32	SCREENSHOT 33	SCREENSHOT 34	SCREENSHOT 35	SCREENSHOT 36	SCREENSHOT 37	SCREENSHOT 38	SCREENSHOT 39	SCREENSHOT 40
SCREENSHOT 41	SCREENSHOT 42	SCREENSHOT 43	SCREENSHOT 44	SCREENSHOT 45	SCREENSHOT 46	SCREENSHOT 47	SCREENSHOT 48	SCREENSHOT 49	SCREENSHOT 50
SCREENSHOT 51	SCREENSHOT 52	SCREENSHOT 53	SCREENSHOT 54	SCREENSHOT 55	SCREENSHOT 56	SCREENSHOT 57	SCREENSHOT 58	SCREENSHOT 59	SCREENSHOT 60
SCREENSHOT 61	SCREENSHOT 62	SCREENSHOT 63	SCREENSHOT 64	SCREENSHOT 65	SCREENSHOT 66	SCREENSHOT 67	SCREENSHOT 68	SCREENSHOT 69	SCREENSHOT 70
SCREENSHOT 71	SCREENSHOT 72	SCREENSHOT 73	SCREENSHOT 74	SCREENSHOT 75	SCREENSHOT 76	SCREENSHOT 77	SCREENSHOT 78	SCREENSHOT 79	SCREENSHOT 80
SCREENSHOT 81	SCREENSHOT 82	SCREENSHOT 83	SCREENSHOT 84	SCREENSHOT 85	SCREENSHOT 86	SCREENSHOT 87	SCREENSHOT 88	SCREENSHOT 89	SCREENSHOT 90
SCREENSHOT 91	SCREENSHOT 92	SCREENSHOT 93	SCREENSHOT 94	SCREENSHOT 95	SCREENSHOT 96	SCREENSHOT 97	SCREENSHOT 98	SCREENSHOT 99	SCREENSHOT 100

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

TODONE
LIS

LOCKDN
LIS

MODIFY
LIS

MOUNT
LIS

MYTHOR
LIS

MAKNMB
LIS

MAKSTR
LIS