



```

MM      MM      000000  DDDDDDDD  IIIIII  FFFFFFFF  YY      YY
MM      MM      000000  DDDDDDDD  IIIIII  FFFFFFFF  YY      YY
MMMM    MMMM    00      00  DD      DD  II      FF      YY      YY
MMMM    MMMM    00      00  DD      DD  II      FF      YY      YY
MM      MM      00      00  DD      DD  II      FF      YY      YY
MM      MM      00      00  DD      DD  II      FF      YY      YY
MM      MM      00      00  DD      DD  II      FF      YY      YY
MM      MM      00      00  DD      DD  II      FF      YY      YY
MM      MM      00      00  DD      DD  II      FF      YY      YY
MM      MM      00      00  DD      DD  II      FF      YY      YY
MM      MM      00      00  DD      DD  II      FF      YY      YY
MM      MM      000000  DDDDDDDD  IIIIII  FFFFFFFF  YY      YY
MM      MM      000000  DDDDDDDD  IIIIII  FFFFFFFF  YY      YY

```

```

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

```



```
1 0001 0
2 0002 0 MODULE MODIFY (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 * COPYRIGHT (c) 1978, 1980, 1982, 1984 BY
10 0010 1 * DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASSACHUSETTS.
11 0011 1 * ALL RIGHTS RESERVED.
12 0012 1 *
13 0013 1 * THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND COPIED
14 0014 1 * ONLY IN ACCORDANCE WITH THE TERMS OF SUCH LICENSE AND WITH THE
15 0015 1 * INCLUSION OF THE ABOVE COPYRIGHT NOTICE. THIS SOFTWARE OR ANY OTHER
16 0016 1 * COPIES THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY
17 0017 1 * OTHER PERSON. NO TITLE TO AND OWNERSHIP OF THE SOFTWARE IS HEREBY
18 0018 1 * TRANSFERRED.
19 0019 1 *
20 0020 1 * THE INFORMATION IN THIS SOFTWARE IS SUBJECT TO CHANGE WITHOUT NOTICE
21 0021 1 * AND SHOULD NOT BE CONSTRUED AS A COMMITMENT BY DIGITAL EQUIPMENT
22 0022 1 * CORPORATION.
23 0023 1 *
24 0024 1 * DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE OR RELIABILITY OF ITS
25 0025 1 * SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL.
26 0026 1 *
27 0027 1 *
28 0028 1 *****
29 0029 1
30 0030 1 ++
31 0031 1
32 0032 1 FACILITY: MTAACP
33 0033 1
34 0034 1 ABSTRACT:
35 0035 1 This module executes the modify function.
36 0036 1 ( This module has never been implemented because of design problems
37 0037 1 which caused race conditions in the execution. Maria Nasr)
38 0038 1
39 0039 1 ENVIRONMENT:
40 0040 1
41 0041 1 VMS 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
48 0048 1 AUTHOR: D. H. GILLESPIE, CREATION DATE: 18-JUL-77
49 0049 1
50 0050 1 MODIFIED BY:
51 0051 1
52 0052 1 V02-005 DMW00044 David Michael Walp 28-Oct-1981
53 0053 1 Commented out all code but error checking for bad attributes
54 0054 1
55 0055 1 V02-004 REFORMAT Maria del C. Nasr 30-Jun-1980
56 0056 1
57 0057 1
```

MODIFY  
V04-000

N 8  
16-Sep-1984 02:25:09  
14-Sep-1984 12:46:43

VAX-11 Bliss-32 V4.0-742  
[MTAACP.SRC]MODIFY.B32;1

Page 2  
(1)

```
.. 58      0058 1  !**
.. 59      0059 1
.. 60      0060 1  LIBRARY 'SYSS$LIBRARY:LIB.L32';
.. 61      0061 1
.. 62      0062 1  REQUIRE 'SRC$:MTADEF.B32';
.. 63      0446 1
.. 64      0447 1  FORWARD ROUTINE
.. 65      0448 1      MTA_MODIFY : NOPRES NOVALUE;      ! main control for modify acp function
.. 66      0449 1
.. 67      0450 1  EXTERNAL
.. 68      0451 1      IO_PACKET : REF BBLOCK;      ! address of current IO request packet
```

MOI  
VOI

```

70 0452 1 GLOBAL ROUTINE MTA_MODIFY : NOPRES NOVALUE =
71 0453 1
72 0454 1 |++
73 0455 1
74 0456 1 FUNCTIONAL DESCRIPTION:
75 0457 1     this routine executes the modify function
76 0458 1
77 0459 1 CALLING SEQUENCE:
78 0460 1     MTA_MODIFY()
79 0461 1
80 0462 1 INPUT PARAMETERS:
81 0463 1     NONE
82 0464 1
83 0465 1 IMPLICIT INPUTS:
84 0466 1     IO_PACKET - address of current io request packet
85 0467 1
86 0468 1 OUTPUT PARAMETERS:
87 0469 1     NONE
88 0470 1
89 0471 1 IMPLICIT OUTPUTS:
90 0472 1     NONE
91 0473 1
92 0474 1 ROUTINE VALUE:
93 0475 1     NONE
94 0476 1
95 0477 1 SIDE EFFECTS:
96 0478 1     this routine only handles user labels and end user label processing with
97 0479 1     next ast
98 0480 1
99 0481 1 |--
100 0482 1 BEGIN
101 0483 2
102 0484 2 EXTERNAL REGISTER
103 0485 2     COMMON_REG;
104 0486 2
105 0487 2 LOCAL
106 0488 2
107 0489 2     ABD      : REF BBLOCKVECTOR [, ABD$C_LENGTH],      ! addr of descr vector
108 0490 2     CODE,      ! attribute code
109 0491 2     P,         ! pointer to attribute
110 0492 2     PACKET : REF BBLOCK;      ! io packet address
111 0493 2
112 0494 2 PACKET = .IO_PACKET;
113 0495 2 ABD     = .BBLOCK[.PACKET[IRP$S_SVAPTE], AIB$S_DESCRIPTOR];
114 0496 2
115 0497 2 INCRU I FROM ABD$C_ATTRIB TO .PACKET[IRP$S_BCNT] - 1 DO
116 0498 2     BEGIN
117 0499 2     P     = .ABD[.I, ABD$S_TEXT] + ABD[.I, ABD$S_TEXT];
118 0500 2     CODE = .(.P)<0, 8>;
119 0501 2
120 0502 2     ! check if code is in range
121 0503 2     !
122 0504 2     IF .CODE GTRU ATR$C_MAX_CODE THEN ERR_EXIT(SS$BADATTRIB);
123 0505 2
124 0506 2     ! only attributes that can be modified are user labels and end user
125 0507 2     ! labels and they are not supported
126 0508 2     ! all other attributes are dropped on the floor for device independent

```

```

: 127      0509 3
: 128      0510 3
: 129      0511 4
: 130      0512 3
: 131      0513 3
: 132      0514 2
: 133      0515 2
: 134      0516 1

```

```

:   sake
:   IF (.CODE EQL ATR$C_USERLABEL) OR (.CODE EQL ATR$C_ENDLBLAST)
:   THEN ERR_EXIT(SS$_BADATTRIB);
:   END;
:   END;

```

! end of routine

```

:   50      0000G  CF  D0 00002
:   55      2C  B0  D0 00007
:   53      32  A0  3C 0000B
:           53  D7 0000F
:   50      05  D0 00011
:           22  11 00014
:   51      6540 7E 00016 1$:
:   54      61  3C 0001A
:   54      51  C0 0001D
:   52      64  9A 00020
:   30      52  D1 00023
:           02  1B 00026
:           34  BF 00028
:   CC      52  D1 0002A 2$:
:           05  13 0002D
:   OF      52  D1 0002F
:           02  12 00032
:           34  BF 00034 3$:
:           50  D6 00036 4$:
:   53      50  D1 00038 5$:
:           D9  1B 0003B
:           04 0003D

```

```

:   .TITLE  MODIFY
:   .IDENT  \V04-000\
:   .EXTRN  IO_PACKET
:   .PSECT  $CODE$,NOWRT,2
:   .ENTRY  MTA MODIFY, Save nothing
:   MOVL   IO_PACKET, PACKET
:   MOVL   @44(PACKET), ABD
:   MOVZWL 50(PACKET), R3
:   DECL   R3
:   MOVL   #5, I
:   BRB    5$
:   MOVAQ  (ABD)[I], R1
:   MOVZWL (R1), P
:   ADDL2  R1, P
:   MOVZBL (P), CODE
:   CMPL   CODE, #48
:   BLEQU  2$
:   CHMU   #52
:   CMPL   CODE, #12
:   BEQL   3$
:   CMPL   CODE, #15
:   BNEQ   4$
:   CHMU   #52
:   INCL   I
:   CMPL   I, R3
:   BLEQU  1$
:   RET

```

```

:   0452
:   0494
:   0495
:   0497
:
:   0499
:
:   0500
:   0504
:
:   0511
:
:   0512
:   0497
:
:   0516

```

; Routine Size: 62 bytes, Routine Base: \$CODE\$ + 0000

: 135 0517 1

```
137 0518 1 :ROUTINE TURN_OFF_WRITE : COMMON_CALL NOVALUE =
138 0519 1
139 0520 1 ++
140 0521 1
141 0522 1 :FUNCTIONAL DESCRIPTION:
142 0523 1 :   this routine notes that exclusive writing is no longer taking place
143 0524 1
144 0525 1 :CALLING SEQUENCE:
145 0526 1 :   TURN_OFF_WRITE(), KERNEL MODE
146 0527 1
147 0528 1 :INPUT PARAMETERS:
148 0529 1 :   NONE
149 0530 1
150 0531 1 :IMPLICIT INPUTS:
151 0532 1 :   NONE
152 0533 1
153 0534 1 :OUTPUT PARAMETERS:
154 0535 1 :   NONE
155 0536 1
156 0537 1 :IMPLICIT OUTPUTS:
157 0538 1 :   NONE
158 0539 1
159 0540 1 :ROUTINE VALUE:
160 0541 1 :   NONE
161 0542 1
162 0543 1 :SIDE EFFECTS:
163 0544 1 :   the window no longer maps virtual i/o
164 0545 1
165 0546 1 :USER ERRORS:
166 0547 1 :   NONE
167 0548 1
168 0549 1 --
169 0550 1
170 0551 1 :   BEGIN
171 0552 1
172 0553 1 :   EXTERNAL REGISTER
173 0554 1 :   COMMON_REG;
174 0555 1
175 0556 1 :   CURRENT_WCB[WCB$V_READ] = 1;           ! writing and reading
176 0557 1 :   END;
```

```

178 0558 1 :ROUTINE UPDATE_MUSTCLOS : COMMON_CALL NOVALUE =
179 0559 1
180 0560 1 ++
181 0561 1
182 0562 1 :FUNCTIONAL DESCRIPTION:
183 0563 1 :   this routine notes that the file must be close and inhibits any
184 0564 1 :   more virtual reads or writes
185 0565 1
186 0566 1 :CALLING SEQUENCE:
187 0567 1 :   UPDATE_MUSTCLOS(), KERNEL MODE
188 0568 1
189 0569 1 :INPUT PARAMETERS:
190 0570 1 :   NONE
191 0571 1
192 0572 1 :IMPLICIT INPUTS:
193 0573 1 :   CURRENT_WCB      - address of current window control block
194 0574 1 :   CURRENT_VCB      - address of current volume control block
195 0575 1
196 0576 1 :OUTPUT PARAMETERS:
197 0577 1 :   NONE
198 0578 1
199 0579 1 :IMPLICIT OUTPUTS:
200 0580 1 :   NONE
201 0581 1
202 0582 1 :ROUTINE VALUE:
203 0583 1 :   NONE
204 0584 1
205 0585 1 :SIDE EFFECTS:
206 0586 1 :   NONE
207 0587 1
208 0588 1 :USER ERRORS:
209 0589 1 :   NONE
210 0590 1
211 0591 1 --
212 0592 1
213 0593 1 :   BEGIN
214 0594 1
215 0595 1 :   EXTERNAL REGISTER
216 0596 1 :   COMMON_REG;
217 0597 1
218 0598 1 :   CURRENT_WCB[WCBSW_NMAP] = 0;
219 0599 1 :   CURRENT_VCB[VCBSV_MUSTCLOSE] = 1;
220 0600 1 :   END;

```



```

222 0601 1 ROUTINE INS_USRLBL_ID (ADDR) : COMMON_CALL NOVALUE =
223 0602 1
224 0603 1 ++
225 0604 1
226 0605 1 FUNCTIONAL DESCRIPTION:
227 0606 1     this routine inserts the user label id into the user label
228 0607 1
229 0608 1 CALLING SEQUENCE:
230 0609 1     INS_USRLBL_ID(ARG1), CALLED IN KERNEL MODE
231 0610 1
232 0611 1 INPUT PARAMETERS:
233 0612 1     ARG1 - address of user label
234 0613 1
235 0614 1 IMPLICIT INPUTS:
236 0615 1     NONE
237 0616 1
238 0617 1 OUTPUT PARAMETERS:
239 0618 1     NONE
240 0619 1
241 0620 1 IMPLICIT OUTPUTS:
242 0621 1     first three characters of user label either 'uhl' or 'utl'
243 0622 1
244 0623 1 ROUTINE VALUE:
245 0624 1     NONE
246 0625 1
247 0626 1 SIDE EFFECTS:
248 0627 1     NONE
249 0628 1
250 0629 1 USER ERRORS:
251 0630 1     NONE
252 0631 1
253 0632 1 --
254 0633 1
255 0634 1 BEGIN
256 0635 1
257 0636 1 EXTERNAL REGISTER
258 0637 1     COMMON_REG;
259 0638 1
260 0639 1 IF .CURRENT_VCB[VCBSB_TM] EQL 0
261 0640 1 THEN
262 0641 1     (.ADDR)<0, 24> = 'UHL'
263 0642 1 ELSE
264 0643 1     (.ADDR)<0, 24> = 'UTL';
265 0644 1
266 0645 1 END;
267 0646 1 END

```

! end of routine

```
.. 269 0647 1 ! IF .COUNT GTRU 80 OR .COUNT LSS 18 THEN ERR_EXIT(SS$_BADATTRIB);  
.. 270 0648 1 ! IF .CURRENT_WCB EQL 0 THEN ERR_EXIT(SS$_ILLUSRLBLWT);  
.. 271 0649 1 ! IF .CURRENT_WCB[WCBSV_READ] THEN ERR_EXIT(SS$_ILLUSRLBLWT);  
.. 272 0650 1 ! IF .CURRENT_VCB[VCBSB_TM] EQL 1 THEN  
.. 273 0651 1 ! WHEN REQUEST TO WRITE USER LABEL IS RECEIVED WHEN IN THE DATA AREA  
.. 274 0652 1 ! THEN A FORCED CLOSE TAKES PLACE  
.. 275 0653 1 ! BEGIN  
.. 276 0654 1 ! KERNEL_CALL(UPDATE_MUSTCLOS);  
.. 277 0655 1 ! WRITE_TRAILERS('F');  
.. 278 0656 1 ! END;  
.. 279 0657 1 ! IF NOT .CURRENT_VCB[VCBSV_WAIUSRLBL]  
.. 280 0658 1 ! AND NOT (.HDR1[EO1$L EO1LID] EQL 'EOF1'  
.. 281 0659 1 ! AND .CURRENT_VCB[VCBSB_TM] EQL 2) THEN ERR_EXIT(SS$_ILLUSRLBLWT);  
.. 282 0660 1 ! KERNEL_CALL(INS_USRLBL_ID,.P);  
.. 283 0661 1 ! WRITE_BLOCK(.P,.COUNT); !WRITE USER LABEL
```

```

: 285      0662 1  IF .COUNT NEQ 4 THEN ERR_EXIT(SS$ BADATTRIB);
: 286      0663 1  IF .CURRENT_WCB EQL 0 THEN ERR_EXIT(SS$ ILLUSRLBLWT);
: 287      0664 1  IF .CURRENT_WCB[WCB$V_READ] THEN ERR_EXIT(SS$ ILLUSRLBLWT);
: 288      0665 1  IF NOT .CURRENT_VCB[VCB$V_WAIUSRLBL]
: 289      0666 1  AND NOT .CURRENT_VCB[VCB$V_MUSTCLOSE] THEN ERR_EXIT(SS$ ILLUSRLBLWT);
: 290      0667 1  AST_BLOCK = .(.PT);
: 291      0668 1  BEGIN
: 292      0669 1  BUILTIN PROBER;
: 293      0670 1  MODE = 0;
: 294      0671 1  LENGTH = 4;
: 295      0672 1  IF .AST_BLOCK NEQ 0
: 296      0673 1  AND (NOT PROBER(MODE,LENGTH,.AST_BLOCK)
: 297      0674 1  OR .AST_BLOCK[ACB$B_TYPE] NEQ DYN$C_ACB)
: 298      0675 1  THEN ERR_EXIT(SS$ ILLBLAST);
: 299      0676 1  END;
: 300      0677 1  IF .CURRENT_VCB[VCB$V_MUSTCLOSE] THEN
: 301      0678 1  BEGIN
: 302      0679 1  CLOSE_FILE();
: 303      0680 1  KERNEL_CALL(TURN_OFF_WRITE);
: 304      0681 1  END;
: 305      0682 1  KERNEL_CALL(COMplete_USRLBL,.AST_BLOCK,.I .ABD);
: 306      0683 1
: 307      0684 1  IF UNBLOCK NECESSARY DO IT NOW
: 308      0685 1  IF .CURRENT_VCB[VCB$V_WAIUSRLBL] THEN UNBLOCK(.CURRENT_VCB);
: 309      0686 1
: 310      0687 1
: 311      0688 0 ELUDOM

```

PSECT SUMMARY

Name	Bytes	Attributes
\$CODE\$	62	NOVEC,NOWRT, RD , EXE,NOSHR, LCL, REL, CON,NOPI,ALIGN(2)

Library Statistics

File	Symbols		Pages Mapped	Processing Time
	Total	Loaded		
_\$255\$DUA28:[SYSLIB]LIB.L32;1	18619	14	1000	00:01.8

COMMAND QUALIFIERS

MODIFY  
V04-000

1 9  
16-Sep-1984 02:25:09  
14-Sep-1984 12:46:43

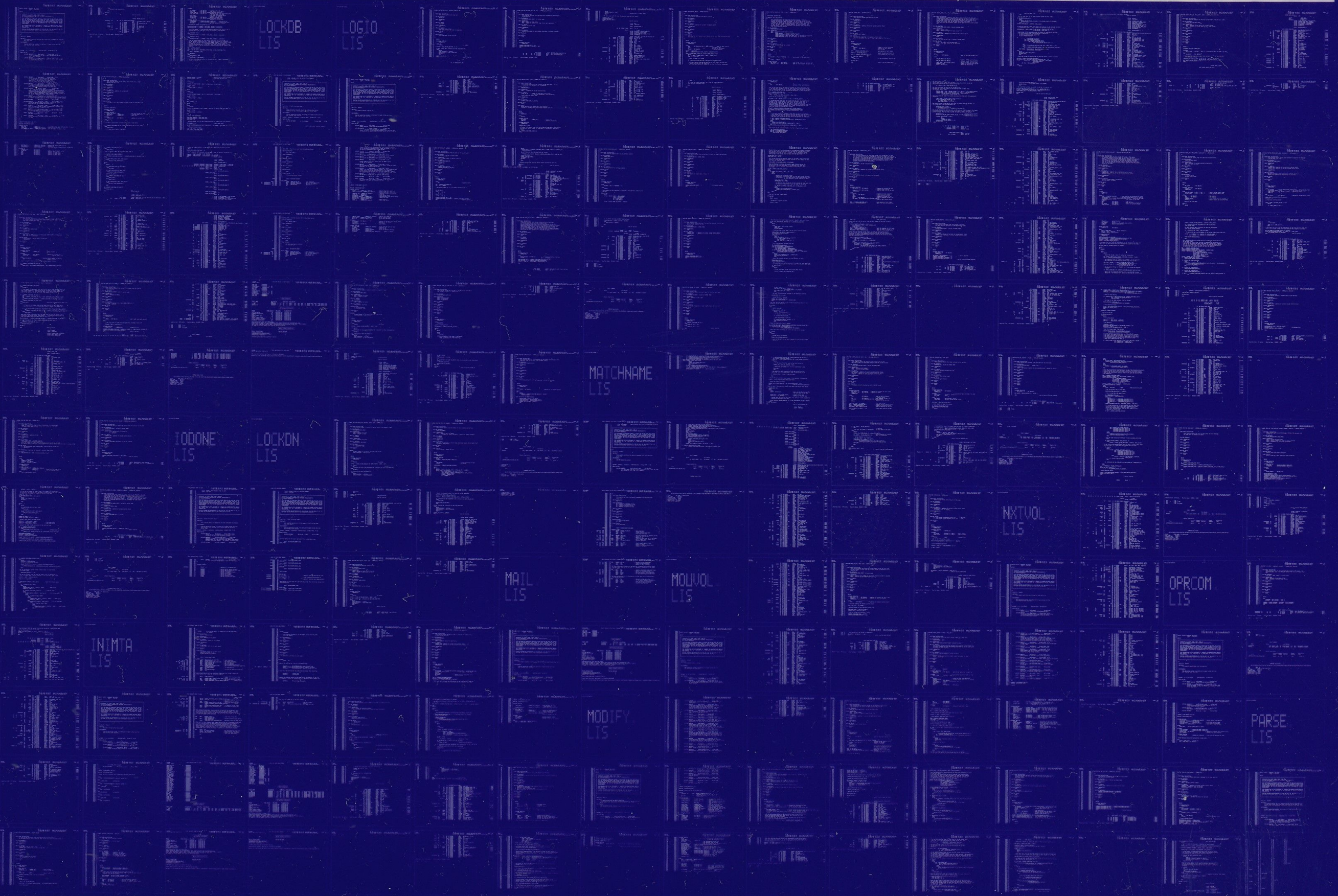
VAX-11 Bliss-32 V4.0-742  
[MTAACP.SRC]MODIFY.B32;1

Page 10  
(7)

MOL  
V04

```
:      BLISS/CHECK=(FIELD,INITIAL,OPTIMIZE)/LIS=LISS:MODIFY/OBJ=OBJ$:MODIFY MSRCS:MODIFY/UPDATE=(ENHS:MODIFY)
: Size:          62 code + 0 data bytes
: Run Time:      00:06.9
: Elapsed Time: 00:16.4
: Lines/CPU Min: 6008
: Lexemes/CPU-Min: 18698
: Memory Used:  76 pages
: Compilation Complete
```





LOCKDB  
LIS

LOGTO  
LIS

MATCHNAME  
LIS

TODONE  
LIS

LOCKDN  
LIS

NXTVOL  
LIS

MAIL  
LIS

MODVOL  
LIS

OPRCOM  
LIS

INIMTA  
LIS

MODIFY  
LIS

PARSE  
LIS