

1
2
:001 :CDS0004
4-1
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
:001 :CDS0004
:002 :CDS0004
:003 :CDS0004
:004 :CDS0004
52
53

0001
0002
0003
0004
0005
0006
0007
0008
0009
0010
0011
0012
0013
0014
0015
0016
0017
0018
0019
0020
0021
0022
0023
0024
0025
0026
0027
0028
0029
0030
0031
0032
0033
0034
0035
0036
0037
0038
0039
0040
0041
0042
0043
0044
0045
0046
0047
0048
0049
0050
0051
0052
0053
0054
0055
0056
0057

```
0 MODULE MAKACC (
0     LANGUAGE (BLISS32),
0     IDENT = 'V04-001'
0 ) =
1 BEGIN
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 FACILITY: F11ACP Structure Level 1
1
1 ABSTRACT:
1
1 This routine makes the necessary hookups in the I/O database to
1 reflect a new file access.
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 AUTHOR: Andrew C. Goldstein, CREATION DATE: 20-Dec-1976 17:28
1
1 MODIFIED BY:
1
1 V04-001 CDS0004 Christian D. Saether 14-Nov-1984
1 Remove test for directory here. It must check the
1 file header instead in ACCESS.
1
1 V03-004 CDS0003 Christian D. Saether 19-Apr-1984
1 Bump REFCNT in Fcb. Do not bump other counts if
```

```
.. 54      0058 1  : this is NOACCLOCK. Remove reference to old dirfcb index.
... 55      0059 1  :
... 56      0060 1  : V03-003 CDS0002 Christian D. Saether 2-Mar-1984
... 57      0061 1  : Set WRITE_TURN flag in WCB if index file, storage bitmap,
... 58      0062 1  : or a directory is being write accessed.
... 59      0063 1  :
... 60      0064 1  : V03-002 CDS0001 Christian D. Saether 30-Dec-1983
... 61      0065 1  : Use L_NORM linkage and BIND_COMMON macro.
... 62      0066 1  :
... 63      0067 1  : V03-001 LMP0059 L. Mark Pilant, 4-Jan-1983 12:28
... 64      0068 1  : Don't insert the FCB into the queue as it is done when the
... 65      0069 1  : FCB is created.
... 66      0070 1  :
... 67      0071 1  : V02-002 LMP0003 L. Mark Pilant, 20-Nov-1981 9:30
... 68      0072 1  : Modify so that all the segments to a window get inserted into
... 69      0073 1  : the window queue.
... 70      0074 1  :
... 71      0075 1  : V02-001 ACG0167 Andrew C. Goldstein, 16-Apr-1980 19:26
... 72      0076 1  : Previous revision history moved to F11B.REV
... 73      0077 1  : **
... 74      0078 1  :
... 75      0079 1  :
... 76      0080 1 LIBRARY 'SYSS$LIBRARY:LIB.L32';
... 77      0081 1 REQUIRE 'SRCS:FCPDEF.B32';
```

```

79 1072 1 GLOBAL ROUTINE MAKE_ACCESS (FCB, WINDOW, ABD) : L_NORM NOVALUE =
80 1073 1
81 1074 1 !++
82 1075 1
83 1076 1 FUNCTIONAL DESCRIPTION:
84 1077 1
85 1078 1 This routine makes the necessary hookups in the I/O database to
86 1079 1 reflect a new file access.
87 1080 1
88 1081 1 CALLING SEQUENCE:
89 1082 1 MAKE_ACCESS (ARG1, ARG2, ARG3)
90 1083 1
91 1084 1 INPUT PARAMETERS:
92 1085 1 ARG1: address of FCB to access
93 1086 1 ARG2: address of window to link up
94 1087 1 ARG3: address of buffer descriptors
95 1088 1
96 1089 1 IMPLICIT INPUTS:
97 1090 1 CURRENT_VCB: VCB of volume in process
98 1091 1
99 1092 1 OUTPUT PARAMETERS:
100 1093 1 NONE
101 1094 1
102 1095 1 IMPLICIT OUTPUTS:
103 1096 1 NONE
104 1097 1
105 1098 1 ROUTINE VALUE:
106 1099 1 NONE
107 1100 1
108 1101 1 SIDE EFFECTS:
109 1102 1 VCB transaction count bumped, access counts in FCB adjusted,
110 1103 1 FCB and window hooked up.
111 1104 1
112 1105 1 --
113 1106 1
114 1107 2 BEGIN
115 1108 2
116 1109 2 MAP
117 1110 2 FCB : REF BBLOCK, ! FCB arg
118 1111 2 WINDOW : REF BBLOCK, ! window arg
119 1112 2 ABD : REF BBLOCKVECTOR [,ABD$C_LENGTH];
120 1113 2 ! buffer descriptor arg
121 1114 2
122 1115 2 LOCAL
123 1116 2 WINDOW_SEGMENT : REF BBLOCK; ! address of the current window segment
124 1117 2
125 1118 2 BIND_COMMON;
126 1119 2
127 1120 2 EXTERNAL
128 1121 2 PMS$GL_OPEN : ADDRESSING_MODE (ABSOLUTE),
129 1122 2 ! system count of currently open files
130 1123 2 PMS$GL_OPENS : ADDRESSING_MODE (ABSOLUTE);
131 1124 2 ! system count of files opened
132 1125 2
133 1126 2 ! Now hook the window onto the FCB and adjust the access counts
134 1127 2 ! according to the access control bits in the window.
135 1128 2

```

```

136      1129      2 WINDOW_SEGMENT = .WINDOW;
137      1130      2 DO
138      1131      2 BEGIN
139      1132      2     INSQUE (.WINDOW_SEGMENT, .FCB[FCBSL_WLBL]);
140      1133      2     WINDOW_SEGMENT = .WINDOW_SEGMENT[WCBSL_LINK];
141      1134      2     END
142      1135      2 UNTIL .WINDOW_SEGMENT EQL 0;
143      1136      2
144      1137      2 FCB [FCBSW_REFCNT] = .FCB [FCBSW_REFCNT] + 1;    ! bump reference count
145      1138      2
146      1139      2 IF NOT .WINDOW [WCBSV_NOACLOCK]
147      1140      2 THEN
148      1141      2     BEGIN
149      1142      2         FCB[FCBSW_ACNT] = .FCB[FCBSW_ACNT] + 1;    ! bump access count
150      1143      2
151      1144      2         IF .WINDOW[WCBSV_NOREAD]
152      1145      2         THEN FCB[FCBSV_EXCL] = 1;                ! set exclusive access
153      1146      2
154      1147      2         IF .WINDOW[WCBSV_NOWRITE]
155      1148      2         THEN FCB[FCBSW_LCNT] = .FCB[FCBSW_LCNT] + 1; ! no writers
156      1149      2
157      1150      2         IF .WINDOW[WCBSV_NOTRUNC]
158      1151      2         THEN FCB[FCBSW_TCNT] = .FCB[FCBSW_TCNT] + 1; ! no truncates
159      1152      2
160      1153      2     END;
161      1154      2
162      1155      2     ! For a write access, bump the writer count. If this is the
163      1156      2     ! first write, and the file is the index file or the storage map, set
164      1157      2     ! the appropriate flag in the VCB.
165      1158      2     !
166      1159      2     !
167      1160      2     !
168      1161      2     !
169      1162      2 IF .WINDOW[WCBSV_WRITE]
170      1163      2 THEN
171      1164      2     BEGIN
172      1165      2         IF .FCB [FCBSB_FID_NMX] EQL 0
173      1166      2         THEN
174      1167      2             BEGIN
175      1168      2                 IF .FCB[FCBSW_FID_NUM] EQL 1
176      1169      2                 THEN
177      1170      2                     BEGIN
178      1171      2                         CURRENT_VCB[VCBSV_WRITE_IF] = 1;
179      1172      2                         WINDOW [WCBSV_WRITE_TURN] = 1;
180      1173      2                     END;
181      1174      2                 END;
182      1175      2             IF .FCB[FCBSW_FID_NUM] EQL 2
183      1176      2             THEN
184      1177      2                 BEGIN
185      1178      2                     CURRENT_VCB[VCBSV_WRITE_SM] = 1;
186      1179      2                     WINDOW [WCBSV_WRITE_TURN] = 1;
187      1180      2                 END;
188      1181      2             END;
189      1182      2         END;
190      1183      2     IF NOT .WINDOW [WCBSV_NOACLOCK]
191      1184      2     THEN
192      1185      2         FCB[FCBSW_WCNT] = .FCB[FCBSW_WCNT] + 1;

```

```

: 200      1186      3
: 201      1187      2      END;
: 202      1188      2
: 203      1189      2      ! Count the access in the volume transaction count and return
: 204      1190      2      ! the window address for the user's CCB.
: 205      1191      2
: 206      1192      2
: 207      1193      2      PMSSGL_OPEN = .PMSSGL_OPEN + 1;      ! bump open file count
: 208      1194      2      PMSSGL_OPENS = .PMSSGL_OPENS + 1;      ! bump count of opens
: 209      1195      2      CURRENT_VCB[VCB$W_TRANS] = .CURRENT_VCB[VCB$W_TRANS] + 1;
: 210      1196      2
: 211      1197      2      ABD[ABD$C_WINDOW, ABD$W_COUNT] = 4;      ! enable write-back
: 212      1198      2      .ABD[ABD$C_WINDOW, ABD$W_TEXT] + ABD[ABD$C_WINDOW, ABD$W_TEXT] + 1 = .WINDOW;
: 213      1199      2      ! put window address in buffer text
: 214      1200      2
: 215      1201      2      ! Mark the access complete in the cleanup action flags.
: 216      1202      2
: 217      1203      2
: 218      1204      2      CLEANUP_FLAGS[CLF_DEACCESS] = 1;
: 219      1205      2
: 220      1206      1      END;      ! end of routine MAKE_ACCESS

```

```

.TITLE MAKACC
.IDENT \V04-001\
.EXTRN PMSSGL_OPEN, PMSSGL_OPENS
.PSECT $CODE$,NOWRT,2

```

			0000 0000	.ENTRY MAKE ACCESS, Save nothing	: 1072
	51	08	AC D0 00002	MOVL WINDOW, WINDOW_SEGMENT	: 1130
	50	04	AC D0 00006 1\$:	MOVL FCB, R0	: 1133
14	B0	61	0E 0000A	INSQUE (WINDOW_SEGMENT), @20(R0)	
	51	20	A1 D0 0000E	MOVL 32(WINDOW_SEGMENT), WINDOW_SEGMENT	: 1134
			F2 12 00012	BNEQ 1\$: 1136
	50	04	AC D0 00014	MOVL FCB, R0	: 1138
		18	A0 B6 00018	INCW 24(R0)	
	50	08	AC D0 0001B	MOVL WINDOW, R0	: 1140
37	14	A0	02 E0 0001F	BBS #2, 20(R0), 4\$	
	50	04	AC D0 00024	MOVL FCB, R0	: 1144
		1A	A0 B6 00028	INCW 26(R0)	
	50	08	AC D0 0002B	MOVL WINDOW, R0	: 1146
08	15	A0	02 E1 0002F	BBC #2, 21(R0), 2\$	
	50	04	AC D0 00034	MOVL FCB, R0	: 1147
	22	A0	08 B8 00038	BISB2 #8, 34(R0)	
	50	08	AC D0 0003C 2\$:	MOVL WINDOW, R0	: 1149
	07	14	A0 E9 00040	BLBC 20(R0), 3\$	
	50	04	AC D0 00044	MOVL FCB, R0	: 1150
		1E	A0 B6 00048	INCW 30(R0)	
	50	08	AC D0 0004B 3\$:	MOVL WINDOW, R0	: 1152
07	15	A0	03 E1 0004F	BBC #3, 21(R0), 4\$	
	50	04	AC D0 00054	MOVL FCB, R0	: 1153
		20	A0 B6 00058	INCW 32(R0)	
	50	08	AC D0 0005B 4\$:	MOVL WINDOW, R0	: 1162
49	08	A0	01 E1 0005F	BBC #1, 11(R0), 7\$	
	50	04	AC D0 00064	MOVL FCB, R0	: 1165

		29	A0	95	00068	TSTB	41(R0)	
		30	12	00068	BNEQ	6\$		
	01	24	AC	B1	0006D	CMPW	36(R0), #1	1168
		10	12	00071	BNEQ	5\$		
	50	98	AA	D0	00073	MOVL	-104(BASE), R0	1171
08	A0	01	88	00077	BISB2	#1, 11(R0)		
	50	08	AC	D0	0007B	MOVL	WINDOW, R0	1172
15	A0	10	88	0007F	BISB2	#16, 21(R0)		
	50	04	AC	D0	00083	MOVL	FCB, R0	1175
	02	24	A0	B1	00087	CMPW	36(R0), #2	
		10	12	0008B	BNEQ	6\$		
	50	98	AA	D0	0008D	MOVL	-104(BASE), R0	1178
08	A0	02	88	00091	BISB2	#2, 11(R0)		
	50	08	AC	D0	00095	MOVL	WINDOW, R0	1179
15	A0	10	88	00099	BISB2	#16, 21(R0)		
	50	08	AC	D0	0009D	MOVL	WINDOW, R0	1183
07	14	A0	02	E0	000A1	BBS	#2, 20(R0), 7\$	
		50	04	AC	D0	000A6	MOVL	FCB, R0
		1C	A0	B6	000AA	INCL	23(R0)	1185
		00000000G	9F	D6	000AD	INCL	@#PMS\$GL_OPEN	1193
		00000000G	9F	D6	000B3	INCL	@#PMS\$GL_OPENS	1194
	50	98	AA	D0	000B9	MOVL	-104(BASE), R0	1195
		0C	A0	B6	000BD	INCL	12(R0)	
	50	0C	AC	D0	000C0	MOVL	ABD, R0	1197
02	A0	04	B0	000C4	MOVW	#4, 2(R0)		
	50	0C	BC	3C	000C8	MOVZWL	@ABD, R0	1198
	50	0C	AC	C0	000CC	ADDL2	ABD, R0	
01	A0	08	AC	D0	000D0	MOVL	WINDOW, 1(R0)	
02	AA	01	88	000D5	BISB2	#1, 2(BASE)		1204
		04	000D9	RET				1206

: Routine Size: 218 bytes, Routine Base: \$CODE\$ + 0000

```
: 221      1207  1
: 222      1208  1 END
: 223      1209  0 ELUDOM
```

PSECT SUMMARY

Name	Bytes	Attributes
\$CODE\$	218	NOVEC, NOWRT, RD, EXE, NOSHR, LCL, REL, CON, NOPIC, ALIGN(2)

Library Statistics

File	----- Symbols -----		Pages Mapped	Processing Time
	Total	Loaded Percent		

MAKACC
V04-001

D 12
8-Jan-1985 18:12:07
2-Oct-1984 12:43:34

VAX-11 Bliss-32 V4.0-742
[F11X.BUGSRC]MAKACC.B32;1

Page 7
(2)

: _\$255\$DUA18:[SYSLIB]LIB.L32;1 18619 40 0 1000 00:02.0

COMMAND QUALIFIERS

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

: Size: 218 code + 0 data bytes
: Run Time: 00:19.0
: Elapsed Time: 00:35.0
: Lines/CPU Min: 3809
: Lexemes/CPU-Min: 46856
: Memory Used: 232 pages
: Compilation Complete

