

FFFFFFFFFFFFFFFF
FFFFFFFFFFFFFFFF
FFFFFFFFFFFFFFFF
FFF
FFF
FFF
FFF
FFF
FFF
FFFFFFFF.FFF
FFFFFFFFFFFFFF
FFFFFFFFFFFFFF
FFF
FFF
FFF
FFF
FFF
FFF
FFF
FFF
FFF
FFF
FFF
FFF
FFF

111
111
111
111111
111111
111111
111
111
111
111
111
111
111
111
111
111
111
111
111
111
111
1111111111
1111111111
1111111111

111
111
111
111111
111111
111111
111
111
111
111
111
111
111
111
111
111
111
111
111
111
1111111111
1111111111
1111111111

XXX XXX
XXX XXX
XXX XXX
XXX XXX
XXX XXX
XXX
XXX XXX XXX
XXX XXX
XXX XXX
XXX
XXX XXX XXX
XXX XXX XXX
XXX XXX
XXX XXX XXX
XXX XXX XXX
XXX XXX XXX
XXX XXX XXX
XXX XXX XXX
XXX XXX XXX
XXX XXX XXX
XXX XXX XXX

_S25

Symb

IOCI
IO_C
IO_C
IO_C
IO_F
IO_S
KICL

KILL
KILL
LB_E
LB_C
LB_F
LB_P
LB_L
LOCAL
LOCAL
LOCK

LOCK
LOCK
LOCK
LOC_
LOC_
L_CC
L_CC
L_D
L_DA
L_DA
MAIN
MAKE
MAKE
MAKE
MAKE
MAKE

MAKE
MAKE
MAP_
MAP_

M.P
MAR
MAR
MAR
MAR

```

RRRRRRRR      EEEEEEEEEEE  MM      MM      000000      VV      VV      EEEEEEEEEEE
RRRRRRRR      EEEEEEEEEEE  MM      MM      000000      VV      VV      EEEEEEEEEEE
RR      RR      EE          MMMM  MMMM  00      00      VV      VV      EE
RR      RR      EE          MMMM  MMMM  00      00      VV      VV      EE
RR      RR      EE          MM  MM  MM  00      00      VV      VV      EE
RR      RR      EE          MM  MM  MM  00      00      VV      VV      EE
RRRRRRRR      EEEEEEEEEEE  MM      MM      00      00      VV      VV      EEEEEEEEEEE
RRRRRRRR      EEEEEEEEEEE  MM      MM      00      00      VV      VV      EEEEEEEEEEE
RR  RR        EE          MM      MM      00      00      VV      VV      EE
RR  RR        EE          MM      MM      00      00      VV      VV      EE
RR      RR      EE          MM      MM      00      00      VV      VV      EE
RR      RR      EE          MM      MM      00      00      VV      VV      EE
RR      RR      EE          MM      MM      00      00      VV      VV      EE
RR      RR      EEEEEEEEEEE MM      MM      000000      VV      VV      EEEEEEEEEEE
RR      RR      EEEEEEEEEEE MM      MM      000000      VV      VV      EEEEEEEEEEE

```

```

LL      IIIIII      SSSSSSSS
LL      IIIIII      SSSSSSSS
L      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
LLLLLLLLLLLL IIIIII SSSSSSSS
LLLLLLLLLLLL IIIIII SSSSSSSS

```

RE
Sy
AC
AQ
BI
CA
CH
CU
DA
DI
EX
FC
HE
IN
IO
IR
IR
LB
MV
PA
QU
RE
RV
UC
UN
VC
WC

PS
--
SA
SC

Ph
--
In
Co
Pa
Sy
Pa
Sy
Ps
Cr
As
Th
41
Th
23
12

```

1 0001 0 MODULE REMOVE (
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 removes a directory entry.
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: 15-Jan-1978 20:19
48 0048 1
49 0049 1 MODIFIED BY:
50 0050 1
51 0051 1 V03-001 CDS0001 Christian D. Saether 30-Dec-1983
52 0052 1 Use L_NORM linkage and BIND_COMMON macro.
53 0053 1
54 0054 1 V02-007 ACG0259 Andrew C. Goldstein, 26-Jan-1982 19:14
55 0055 1 Add mode to keep a name with no versions for autopurge;
56 0056 1 remove updating of directory index.
57 0057 1

```

REMOVE
V04-C00

N 9
16-Sep-1984 01:02:17 VAX-11 Bliss-32 V4.0-742 Page 2
14-Sep-1984 12:30:43 DISK\$VMSMASTER:[F11X.SRC]REMOVE.B32;1 (1)

```

: 58      0058 1 | V02-006 ACG0208      Andrew C. Goldstein, 12-Nov-1981 11:24
: 59      0059 1 |      Change declaration of directory context
: 60      0060 1 |
: 61      0061 1 | V02-005 ACG0167      Andrew C. Goldstein, 16-Apr-1980 19:27
: 62      0062 1 |      Previous revision history moved to f11B.REV
: 63      0063 1 | **
: 64      0064 1 |
: 65      0065 1 |
: 66      0066 1 | LIBRARY 'SYSS$LIBRARY:LIB.L32';
: 67      0067 1 | REQUIRE 'SRC$:FCPDEF.B32';
```

```

69 1058 1 GLOBAL ROUTINE REMOVE (KEEP_NAME) : L_NORM NOVALUE =
70 1059 1
71 1060 1 |++
72 1061 1 |
73 1062 1 | FUNCTIONAL DESCRIPTION:
74 1063 1 |
75 1064 1 |     This routine removes a directory entry.
76 1065 1 |
77 1066 1 |
78 1067 1 | CALLING SEQUENCE:
79 1068 1 |     REMOVE (ARG1)
80 1069 1 |
81 1070 1 | INPUT PARAMETERS:
82 1071 1 |     ARG1: 0 to remove name if all versions are gone
83 1072 1 |           1 to keep a name with no versions
84 1073 1 |
85 1074 1 | IMPLICIT INPUTS:
86 1075 1 |     DIR_VBN: VBN of current directory block
87 1076 1 |     DIR_BUFFER: address of directory block buffer
88 1077 1 |     DIR_ENTRY: address of directory record found
89 1078 1 |     DIR_VERSION: address of directory version entry
90 1079 1 |
91 1080 1 | OUTPUT PARAMETERS:
92 1081 1 |     NONE
93 1082 1 |
94 1083 1 | IMPLICIT OUTPUTS:
95 1084 1 |     DIR_END: address of end of directory data
96 1085 1 |     PREV_NAME: file name of directory entry
97 1086 1 |     PREV_VERSION: version number of directory entry
98 1087 1 |
99 1088 1 | ROUTINE VALUE:
100 1089 1 |     NONE
101 1090 1 |
102 1091 1 | SIDE EFFECTS:
103 1092 1 |     directory entry removed, block(s) written
104 1093 1 |
105 1094 1 | --
106 1095 1 |
107 1096 2 BEGIN
108 1097 2
109 1098 2 LOCAL
110 1099 2     P,                ! character scan pointer
111 1100 2     RECORD_SIZE;   ! size of directory record
112 1101 2
113 1102 2 BIND_COMMON;
114 1103 2
115 1104 2 DIR_CONTEXT_DEF;
116 1105 2
117 1106 2 EXTERNAL ROUTINE
118 1107 2     NEXT_REC           : L_NORM,      ! get next directory record
119 1108 2     MARK_DIRTY        : L_NORM,      ! mark buffer for write back
120 1109 2     SHUFFLE_DIR      : L_NORM,      ! compress directory
121 1110 2     UPDATE_DIRSEQ    : L_NORM ADDRESSING_MODE (GENERAL);
122 1111 2                                     ! update directory sequence count
123 1112 2
124 1113 2
125 1114 2 ! First scan to the end of the directory records and compute the size of the

```

```
126 1115 2 ! current record. Also save name and version for the cleanup.
127 1116 2 !
128 1117 2 !
129 1118 2 DIR_END = .DIR_ENTRY;
130 1119 2 UNTIL .DIR_END[DIR$W_SIZE] EQL 65535
131 1120 2 DO DIR_END = NEXT_REC (.DIR_END);
132 1121 2 DIR_END = .DIR_END + 2;
133 1122 2 !
134 1123 2 RECORD_SIZE = .DIR_ENTRY[DIR$W_SIZE] + 2;
135 1124 2 IF NOT .CLEANUP_FLAGS[CLF_CLEANUP]
136 1125 2 THEN
137 1126 2     BEGIN
138 1127 2     PREV_VERSION = .DIR_VERSION[DIR$W_VERSION];
139 1128 2     CH$MOVE (.DIR_ENTRY[DIR$B_NAMECOUNT]+1, DIR_ENTRY[DIR$B_NAMECOUNT], PREV_NAME);
140 1129 2     CLEANUP_FLAGS[CLF_REENTER] = 1;
141 1130 2     END;
142 1131 2 !
143 1132 2 ! Determine if the entry being removed is of the form xxx.DIR;1. If so, bump
144 1133 2 ! the directory sequence count in the UCB to invalidate RMS caches.
145 1134 2 !
146 1135 2 !
147 1136 2 IF .DIR_VERSION[DIR$W_VERSION] EQL 1 ! simple tests first
148 1137 2 THEN
149 1138 2     BEGIN
150 1139 2     P = CH$FIND_SUB (.DIR_ENTRY[DIR$B_NAMECOUNT], DIR_ENTRY[DIR$T_NAME],
151 1140 2     4, UPLIT_BYTE ('.DIR'));
152 1141 2     IF NOT CH$FAIL (.P)
153 1142 2     AND .P + 4 EQL DIR_ENTRY[DIR$T_NAME] + .DIR_ENTRY[DIR$B_NAMECOUNT]
154 1143 2     THEN KERNEL_CALL (UPDATE_DIRSEQ);
155 1144 2     END;
156 1145 2 !
157 1146 2 ! Determine if the record contains one or more versions. If there are
158 1147 2 ! more than one or if we are to keep the name, we just squish out the
159 1148 2 ! indicated version.
160 1149 2 !
161 1150 2 !
162 1151 2 IF .KEEP_NAME
163 1152 2 OR .RECORD_SIZE GTRU (.DIR_ENTRY[DIR$B_NAMECOUNT]
164 1153 2     + DIR$C_LENGTH + DIR$C_VERSION + 1 AND NOT 1)
165 1154 2 THEN
166 1155 2     BEGIN
167 1156 2     CH$COPY (.DIR_END - .DIR_VERSION - DIR$C_VERSION, .DIR_VERSION + DIR$C_VERSION,
168 1157 2     0, .DIR_END - .DIR_VERSION, .DIR_VERSION);
169 1158 2     DIR_ENTRY[DIR$W_SIZE] = .DIR_ENTRY[DIR$W_SIZE] - DIR$C_VERSION;
170 1159 2     END
171 1160 2 !
172 1161 2 ! If the record contains only one version, we remove the entire record.
173 1162 2 !
174 1163 2 !
175 1164 2 ELSE
176 1165 2     BEGIN
177 1166 2     CH$COPY (.DIR_END - .DIR_ENTRY - .RECORD_SIZE, .DIR_ENTRY + .RECORD_SIZE,
178 1167 2     0, .DIR_END - .DIR_ENTRY, .DIR_ENTRY);
179 1168 2     DIR_VERSION = 0;
180 1169 2     END;
181 1170 2 !
182 1171 2 ! If this leaves us with an empty block, squish it out. Otherwise, just
```

```

: 183 1172 2 ! write it. But don't squish out the last block of an empty directory.
: 184 1173 2 !
: 185 1174 2 !
: 186 1175 2 IF .DIR_BUFFER[DIR$W SIZE] EQL 65535
: 187 1176 2 AND .DIR_FCB[FCB$L_EFBLK] GTRU 1
: 188 1177 2 THEN
: 189 1178 2     SHUFFLE_DIR (-1)
: 190 1179 2 ELSE
: 191 1180 2     MARK_DIRTY (.DIR_BUFFER);
: 192 1181 2
: 193 1182 1 END;

```

! end of routine REMOVE

```

.TITLE REMOVE
.IDENT \V04-000\
.PSECT $CODE$,NOWRT,2

```

```

52 49 44 2E 0000 P.AAA: .ASCII \.DIR\

```

```

.EXTRN NEXT_REC, MARK_DIRTY
.EXTRN SHUFFLE_DIR, UPDATE_DIRSEQ

```

```

.ENTRY REMOVE, Save R2,R3,R4,R5,R6,R7,R8,R9

```

				03FC 00000	MOVAB	220(BASE), R7	1058
		57	00DC	CA 9E 00002	MOVAB	8(R7), R6	1100
		56	08	A7 9E 00007	MOVAB	16(R7), R8	1102
		58	10	A7 9E 0000B	MOVL	(R6), (R8)	1118
		68		66 D0 0000F	CMPW	@0(R8), #65535	1119
	FFFF	8F	00	B8 B1 00012	BEQL	2\$	
				0C 13 00018	PUSHL	(R8)	1120
				68 DD 0001A	CALLS	#1, NEXT_REC	
	0000G	CF		01 FB 0001C	MOVL	R0, (R8)	
		68		50 D0 00021	BRB	1\$	
				EC 11 00024	ADDL2	#2, (R8)	1121
		68		02 C0 00026	MOVZWL	@0(R6), RECORD_SIZE	1123
		59	00	B6 3C 00029	ADDL2	#2, RECORD_SIZE	
		59		02 C0 0002D	BBS	#9, (BASE)-3\$	1124
	1B	6A		09 E0 00030	CVTWL	@12(R7), 338(BASE)	1127
		0152	CA	0C B7 32 00034	MOVL	(R6), R0	1128
				50 66 D0 0003A	MOVZBL	5(R0), R1	
				51 05 A0 9A 0003D	INCL	R1	
				51 D6 00041	MOVC3	R1, 5(R0), 342(BASE)	
	0156	CA	05	A0 51 28 00043	BISB2	#128, 2(BASE)	1129
			02	AA 80 8F 88 0004A	CMPW	@12(R7), #1	1136
				01 0C B7 B1 0004F	BNEQ	5\$	
				33 12 00053	MOVL	(R6), R0	1139
				50 66 D0 00055	MOVZBL	5(R0), R1	
				51 05 A0 9A 00058	MATCHC	#4, P.AAA, R1, 6(R0)	1140
	06	A0	51	9C AF 04 39 0005C	BEQL	4\$	
				03 13 00063	MOVL	#4, R3	
				53 04 D0 00065	SUBL2	#4, R3	
				53 04 C2 00068	BEQL	5\$	1141
				1B 13 0006B	ADDL2	#4, R3	1142
				53 04 C0 0006D	MOVL	(R6), R1	
				51 66 D0 00070	MOVZBL	5(R1), R0	
				50 05 A1 9A 00073	MOVAB	6(R1)[R0], R0	
				50 06 A140 9E 00077			

		50		53	D1	0007C		CMP	R3, R0		
				07	12	0007F		BNEQ	5\$		
	00000000G	00		00	FB	00081		CALLS	#0, UPDATE_DIRSEQ		1143
		12	04	AC	EB	00088	5\$:	BLBS	KEEP_NAME, -6\$		1151
		50		66	D0	0008C		MOVL	(R6), R0		1152
		50	05	A0	9A	0008F		MOVZBL	5(R0), R0		1153
		50		0F	C0	00093		ADDL2	#15, R0		
		50		01	8A	00096		BICB2	#1, R0		
		50		59	D1	00099		CMP	RECORD_SIZE, R0		1152
		50		19	1B	0009C		BLEQU	7\$		
		50	0C	A7	D0	0009E	6\$:	MOVL	12(R7), R0		1156
	58	68		50	C3	000A2		SUBL3	R0, (R8), R8		
		51	F8	A8	9E	000A6		MOVAB	-8(R8), R1		
58	00	08	A0	51	2C	000AA		MOVCS	R1, 8(R0), #0, R8, (R0)		1157
		00	B6	60		000B0					
		00		08	A2	000B1		SUBW2	#8, @0(R6)		1158
		58	68	14	11	000B5		BRB	8\$		1151
	58	58		66	C3	000B7	7\$:	SUBL3	(R6), (R8), R8		1166
	50	58		59	C3	000BB		SUBL3	RECORD_SIZE, R8, R0		
58	00	00	B649	50	2C	000BF		MOVCS	R0, @0(R6)[RECORD_SIZE], #0, R8, @0(R6)		1167
		00		B6		000C6					
		0C		A7	D4	000C8		CLRL	12(R7)		1168
	FFFF	8F		04	B7	000CB	8\$:	CMPW	@4(R7), #65535		1175
		50	00D0	14	12	000D1		BNEQ	9\$		
		01		CA	D0	000D3		MOVL	208(BASE), R0		1176
		01	3C	A0	D1	000D8		CMP	60(R0), #1		
		7E		09	1B	000DC		BLEQU	9\$		
	0000G	CF		01	CE	000DE		MNEGL	#1, -(SP)		1178
				01	FB	000E1		CALLS	#1, SHUFFLE_DIR		
				04	04	000E6		RET			
	0000G	CF	04	A7	DD	000E7	9\$:	PUSHL	4(R7)		1180
				01	FB	000EA		CALLS	#1, MARK_DIRTY		
				04	04	000EF		RET			1182

; Routine Size: 240 bytes, Routine Base: \$CODE\$ + 0004

```

: 194      1183  1
: 195      1184  1 END
: 196      1185  0 ELUDOM

```

PSECT SUMMARY

Name	Bytes	Attributes
\$CODE\$	244	NOVEC, NOWRT, RD, EXE, NOSHR, LCL, REL, CON, NOPIC, ALIGN(2)

Library Statistics

REMOVE
V04-000

F 10
16-Sep-1984 01:02:17
14-Sep-1984 12:30:43

VAX-11 Bliss-32 V4.0-742 Page 7
DISK\$VMSMASTER:[F11X.SRC]REMOVE.B32;1 (2)

File	----- Total	Symbols Loaded	----- Percent	Pages Mapped	Processing Time
:_\$255\$DUA28:[SYSLIB]LIB.L32;1	18619	24	0	1000	00:02.0

COMMAND QUALIFIERS

BLISS/CHECK=(FIELD,INITIAL,OPTIMIZE)/LIS=LIS\$:REMOVE/OBJ=OBJ\$:REMOVE MSRCS\$:REMOVE/UPDATE=(ENHS\$:REMOVE)

: Size: 240 code + 4 data bytes
: Run Time: 00:18.0
: Elapsed Time: 01:07.2
: Lines/CPU Min: 3956
: Lexemes/CPU-Min: 49255
: Memory Used: 227 pages
: Compilation Complete

0172 AH-BT13A-SE
VAX/VMS V4.0

DIGITAL EQUIPMENT CORPORATION
CONFIDENTIAL AND PROPRIETARY

Grid of 130 terminal screens (10 rows by 13 columns) displaying various system messages and user prompts. Visible text includes:

- ROWB LIS
- ROBLOK LIS
- REQUEL LIS
- RWATTR LIS
- REMOVE LIS
- ROHEDR LIS
- RETDTR LIS

Each screen contains a mix of alphanumeric characters, including status indicators, error codes, and system identifiers. The overall appearance is that of a dense array of active terminals in a mainframe environment.