

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

_S25

Symb

IOCS

IO_C

IO_C

IO_D

IO_F

IO_S

KICL

KILL

KILL

LB_E

LB_C

LB_F

LB_H

LB_L

LOCAL

LOCK

LOCK

LOCK

LOCK

LOC_

LOC_

L_CC

L_CC

L_DA

L_DA

MAIN

MAKE

MAKE

MAKE

MAKE

MAKE

MAKE

MAKE

MAKE

MAP_

MAP_

MAP

MARI

MARI

MARI

MARI

MARI


```

1 0001 0 MODULE CREWIN (
2 0002 0
3 0003 0     LANGUAGE (BLISS32),
4 0004 0     IDENT = 'V04-000'
5 0005 1 BEGIN
6 0006 1
7 0007 1
8 0008 1
9 0009 1
10 0010 1
11 0011 1
12 0012 1
13 0013 1
14 0014 1
15 0015 1
16 0016 1
17 0017 1
18 0018 1
19 0019 1
20 0020 1
21 0021 1
22 0022 1
23 0023 1
24 0024 1
25 0025 1
26 0026 1
27 0027 1
28 0028 1
29 0029 1
30 0030 1
31 0031 1
32 0032 1
33 0033 1
34 0034 1
35 0035 1
36 0036 1
37 0037 1
38 0038 1
39 0039 1
40 0040 1
41 0041 1
42 0042 1
43 0043 1
44 0044 1
45 0045 1
46 0046 1
47 0047 1
48 0048 1
49 0049 1
50 0050 1
51 0051 1
52 0052 1
53 0053 1
54 0054 1
55 0055 1
56 0056 1
57 0057 1

*****
*
*  COPYRIGHT (c) 1978, 1980, 1982, 1984 BY
*  DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASSACHUSETTS.
*  ALL RIGHTS RESERVED.
*
*  THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND COPIED
*  ONLY IN ACCORDANCE WITH THE TERMS OF SUCH LICENSE AND WITH THE
*  INCLUSION OF THE ABOVE COPYRIGHT NOTICE. THIS SOFTWARE OR ANY OTHER
*  COPIES THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY
*  OTHER PERSON. NO TITLE TO AND OWNERSHIP OF THE SOFTWARE IS HEREBY
*  TRANSFERRED.
*
*  THE INFORMATION IN THIS SOFTWARE IS SUBJECT TO CHANGE WITHOUT NOTICE
*  AND SHOULD NOT BE CONSTRUED AS A COMMITMENT BY DIGITAL EQUIPMENT
*  CORPORATION.
*
*  DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE OR RELIABILITY OF ITS
*  SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL.
*
*****

++
FACILITY:  F11ACP Structure Level 2
ABSTRACT:
    This routine creates and initializes a file window.
ENVIRONMENT:
    STARLET operating system, including privileged system services
    and internal exec routines. This routine must be called
    in kernel mode.

--

AUTHOR:  Andrew C. Goldstein,  CREATION DATE:  14-Dec-1976  17:10
MODIFIED BY:
    V03-003  CDS0002      Christian D. Saether      18-Apr-1984
    Mask off unused bits when copying access control flags
    to wcb$w_acon, and copy FIB$V_NOLOCK state to WCB$V_NOACLOCK.
    V03-002  CDS0001      Christian D. Saether      29-Dec-1983
    Use L_NORM linkage and BIND_COMMON macro.
    
```

CREWIN
V04-000

K 5
16-Sep-1984 00:11:54
14-Sep-1984 12:30:15

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

```

: 58 0058 1 | V03-001 LMP0016 L. Mark Pilant 25-Mar-1982 13:20
: 59 0059 1 | Remove diddling of the COMPLETE bit in the window segments.
: 60 0060 1 |
: 61 0061 1 | V02-002 LMP0003 L. Mark Pilant 17-Nov-1981 14:35
: 62 0062 1 | Add support for segmented windows.
: 63 0063 1 |
: 64 0064 1 | V02-001 ACG0167 Andrew C. Goldstein 16-Apr-1980 19:25
: 65 0065 1 | Previous revision history moved to F11B.REV
: 66 0066 1 | **
: 67 0067 1 |
: 68 0068 1 |
: 69 0069 1 | LIBRARY 'SYSS$LIBRARY:LIB.L32';
: 70 0070 1 | REQUIRE 'SRC$:FCPDEF.B32';
```



```

: 72      1061 1 GLOBAL ROUTINE CREATE_WINDOW (ACCTL, SIZE, HEADER, PID, FCB) : L_NORM =
: 73      1062 1
: 74      1063 1 |++
: 75      1064 1
: 76      1065 1 FUNCTIONAL DESCRIPTION:
: 77      1066 1
: 78      1067 1     This routine creates a file access window.
: 79      1068 1
: 80      1069 1 CALLING SEQUENCE:
: 81      1070 1     CREATE_WINDOW (ARG1, ARG2, ARG3, ARG4, ARG5)
: 82      1071 1
: 83      1072 1 INPUT PARAMETERS:
: 84      1073 1     ARG1: access control word (from FIB, usually)
: 85      1074 1     ARG2: size of window in # of pointers
: 86      1075 1     ARG3: address of file header
: 87      1076 1     ARG4: PID of accessor
: 88      1077 1     ARG5: address of file FCB
: 89      1078 1
: 90      1079 1 IMPLICIT INPUTS:
: 91      1080 1     CURRENT_VCB: address of VCB of volume in process
: 92      1081 1     CURRENT_UCB: address of UCB of disk in process
: 93      1082 1
: 94      1083 1 OUTPUT PARAMETERS:
: 95      1084 1     NONE
: 96      1085 1
: 97      1086 1 IMPLICIT OUTPUTS:
: 98      1087 1     NONE
: 99      1088 1
: 100     1089 1 ROUTINE VALUE:
: 101     1090 1     address of window
: 102     1091 1
: 103     1092 1 SIDE EFFECTS:
: 104     1093 1     window block created
: 105     1094 1
: 106     1095 1 |--
: 107     1096 1
: 108     1097 2 BEGIN
: 109     1098 2
: 110     1099 2 MAP
: 111     1100 2     ACCTL          : BBLOCK,      : access control flags
: 112     1101 2     HEADER       : REF BBLOCK,  : file header arg
: 113     1102 2     FCB          : REF BBLOCK; : FCB arg
: 114     1103 2
: 115     1104 2 LOCAL
: 116     1105 2     WINDOW_SIZE,      : actual size of window
: 117     1106 2     WINDOW         : REF BBLOCK,  : window created
: 118     1107 2     PRIMARY_WINDOW : REF BBLOCK; : address of the primary window
: 119     1108 2
: 120     1109 2 BIND_COMMON;
: 121     1110 2
: 122     1111 2 EXTERNAL ROUTINE
: 123     1112 2     ALLOCATE         : L_NORM,      : allocate dynamic memory
: 124     1113 2     TURN_WINDOW     : L_NORM;      : window turner routine
: 125     1114 2
: 126     1115 2 ! Compute the size of the window. If fixed, allocate it and turn it to
: 127     1116 2 ! map VBN 1. If a maximal window is requested (indicated by a size of -1),
: 128     1117 2 ! the window turner will allocate the window.

```

```

: 129      1118 2 !
: 130      1119 2
: 131      1120 2 WINDOW SIZE = .SIZE;
: 132      1121 2 IF .WINDOW_SIZE EQL 0
: 133      1122 2 THEN WINDOW_SIZE = .CURRENT_VCB[VCB$B_WINDOW];
: 134      1123 2
: 135      1124 2 IF .WINDOW_SIZE NEQ -1
: 136      1125 2 THEN
: 137      1126 2 BEGIN
: 138      1127 2 IF .WINDOW_SIZE GTRU MAX_WINDOW
: 139      1128 2 THEN WINDOW_SIZE = MAX_WINDOW;
: 140      1129 2 IF .WINDOW_SIZE LSSU MIN_WINDOW
: 141      1130 2 THEN WINDOW_SIZE = MIN_WINDOW;
: 142      1131 2 WINDOW = ALLOCATE (.WINDOW_SIZE * 6 + WCB$C_LENGTH, WCB_TYPE);
: 143      1132 2 IF .WINDOW NEQ 0
: 144      1133 2 THEN TURN_WINDOW (.WINDOW, .HEADER, 1, 1);
: 145      1134 2 END
: 146      1135 2 ELSE
: 147      1136 2 WINDOW = TURN_WINDOW (0, .HEADER, 1, 1);
: 148      1137 2
: 149      1138 2 ! Init cells within the window
: 150      1139 2 !
: 151      1140 2
: 152      1141 2 PRIMARY_WINDOW = .WINDOW;
: 153      1142 2 UNTIL .WINDOW EQL 0
: 154      1143 2 DO
: 155      1144 2 BEGIN
: 156      1145 2 WINDOW[WCB$P_PID] = .PID; ! accessor PID
: 157      1146 2 WINDOW[WCB$P_ORGUCB] = .CURRENT_UCB; ! original device UCB
: 158      1147 2 WINDOW[WCB$P_ACON] = .ACCTL [0,0,16,0]; ! access control bits
: 159      1148 2 WINDOW[WCB$P_ACON] = .WINDOW[WCB$P_ACON] ! mask off unused flags
: 160      1149 2 AND (FIB$M_NOWRITE+FIB$M_DLOCK+FIB$M_SPOOL+FIB$M_WRITECK+FIB$M_SEQONLY
: 161      1150 2 +FIB$M_WRITE+FIB$M_READCK+FIB$M_NOREAD+FIB$M_NOTRUNC);
: 162      1151 2 IF (.ACCTL [FIB$V_NOLOCK] ! This assumes FIB$P_ACON
: 163      1152 2 AND .CLEANUP_FLAGS [CLF_SYSPRV])
: 164      1153 2 THEN WINDOW [WCB$V_NOACCLOCK] = 1; ! is first longword in FIB.
: 165      1154 2 WINDOW[WCB$P_FCB] = .FCB; ! FCB address
: 166      1155 2 WINDOW[WCB$V_READ] = 1; ! read access always allowed
: 167      1156 2 WINDOW[WCB$V_WRITE] = .WINDOW[WCB$V_WRITEAC]; ! write access sometimes
: 168      1157 2 IF .CURRENT_RVN NEQ 0 THEN WINDOW[WCB$P_RVT] = .CURRENT_VCB[VCB$P_RVT];
: 169      1158 2 IF .HEADER[FH2$V_READCHECK] THEN WINDOW[WCB$V_READCK] = 1;
: 170      1159 2 IF .HEADER[FH2$V_WRITECHECK] THEN WINDOW[WCB$V_WRITECK] = 1;
: 171      1160 2 WINDOW = .WINDOW[WCB$P_LINK]; ! set address of next segment
: 172      1161 2 END;
: 173      1162 2
: 174      1163 2 RETURN .PRIMARY_WINDOW;
: 175      1164 2
: 176      1165 2 END;

```

! end of routine CREATE_WINDOW

```

.TITLE CREWIN
.IDENT \V04-000\
.EXTRN ALLOCATE, TURN_WINDOW
.PSECT $CODE$,NOWRT,2

```


CREWIN
V04-000

N 5
16-Sep-1984 00:11:54
14-Sep-1984 12:30:15

VAX-11 Bliss-32 V4.0-742
DISK\$VMSMASTER:[F11X.SRC]CREWIN.B32;1

Page 5
(2)

	50	08	AC	0004	00000	.ENTRY	CREATE WINDOW, Save R2	: 1061
	51	98	AA	D0	00002	MOVL	SIZE, WINDOW_SIZE	: 1120
	50	48	A1	9A	00006	BNEQ	1\$: 1121
FFFFFFF	8F		A1	9A	00008	MOVL	-104(BASE), R1	: 1122
	8F		A1	9A	0000C	MOVZBL	72(R1), WINDOW_SIZE	
0000050			D1	00010	1\$:	CMPL	WINDOW_SIZE, #-1	: 1124
			13	00017		BEQL	4\$	
			D1	00019		CMPL	WINDOW_SIZE, #80	: 1127
			1B	00020		BLEQU	2\$	
			9A	00022	2\$:	MOVZBL	#80, WINDOW_SIZE	: 1128
			D5	00026		TSTL	WINDOW_SIZE-	: 1129
			12	00028		BNEQ	3\$	
			D0	0002A	3\$:	MOVL	#1, WINDOW_SIZE	: 1130
			DD	0002D		PUSHL	#1	: 1131
			C4	0002F		MULL2	#6, R0	
			9F	00032		PUSHAB	48(R0)	
0000G	CF		FB	00035		CALLS	#2, ALLOCATE	
	52		D0	0003A		MOVL	R0, WINDOW	
			13	0003D		BEQL	5\$: 1132
			DD	0003F		PUSHL	#1	: 1133
			DD	00041		PUSHL	#1	
			DD	00043		PUSHL	HEADER	
			DD	00046		PUSHL	WINDOW	
0000G	CF		FB	00048		CALLS	#4, TURN_WINDOW	
			11	0004D		BRB	5\$: 1124
			DD	0004F	4\$:	PUSHL	#1	: 1136
			DD	00051		PUSHL	#1	
			DD	00053		PUSHL	HEADER	
			D4	00056		CLRL	-(SP)	
0000G	CF		FB	00058		CALLS	#4, TURN_WINDOW	
	52		D0	0005D		MOVL	R0, WINDOW	
	51		D0	00060	5\$:	MOVL	WINDOW, PRIMARY_WINDOW	: 1141
			13	00063	6\$:	BEQL	11\$: 1142
			D0	00065		MOVL	PID, 12(WINDOW)	: 1145
0C	A2	10	AA	0006A		MOVL	-108(BASE), 16(WINDOW)	: 1146
10	A2	94	AA	0006A		MOVL	-108(BASE), 16(WINDOW)	: 1147
14	A2	04	AC	0006F		MOVW	ACCTL, 20(WINDOW)	: 1149
14	A2	F08C	8F	00074		BICW2	#61580, 20(WINDOW)	: 1151
08	06		AA	0007A		BBC	#4, ACCTL+2, 7\$: 1152
	04		AA	0007F		BLBC	1(BASE), 7\$: 1153
	14		AA	0007F		BISB2	#4, 20(WINDOW)	: 1154
	18		AC	00083	7\$:	MOVL	FCB, 24(WINDOW)	: 1155
	OB		AC	00087		BISB2	#1, 11(WINDOW)	: 1156
	0B		AC	0008C		BISB2	#1, 11(WINDOW)	: 1157
OB	A2	01	A2	00090		INSV	21(WINDOW), #1, #1, 11(WINDOW)	: 1158
			AA	00097		TSTL	-96(BASE)	: 1159
			13	0009A		BEQL	8\$	
	50		AA	0009C		MOVL	-104(BASE), R0	
	1C		AA	000A0		MOVL	32(R0), 28(WINDOW)	
	50		AC	000A5	8\$:	MOVL	HEADER, R0	: 1158
04	34		E1	000A9		BBC	#3, 52(R0), 9\$	
	15		88	000AE		BISB2	#2, 21(WINDOW)	
	50		AC	000B2	9\$:	MOVL	HEADER, R0	: 1159
04	34		E1	000B6		BBC	#4, 52(R0), 10\$	
	14		88	000BB		BISB2	#32, 20(WINDOW)	
	52		AC	000BF	10\$:	MOVL	32(WINDOW), WINDOW	: 1160
			11	000C3		BRB	6\$: 1142
	50		D0	000C5	11\$:	MOVL	PRIMARY_WINDOW, R0	: 1163
			04	000C8		RET		: 1165

; Routine Size: 201 bytes, Routine Base: \$CODE\$ + 0000

: 177 1166 1
: 178 1167 1 END
: 179 1168 0 ELUDOM

PSECT SUMMARY

Name	Bytes	Attributes
\$CODE\$	201	NOVEC,NOWRT, RD, EXE,NOSHR, LCL, REL, CON,NOPIC,ALIGN(2)

Library Statistics

File	Total	Symbols Loaded	Percent	Pages Mapped	Processing Time
_\$255\$DUA28:[SYSLIB]LIB.L32;1	18619	44	0	1000	00:01.9

COMMAND QUALIFIERS

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

; Size: 201 code + 0 data bytes
; Run Time: 00:17.3
; Elapsed Time: 00:37.2
; Lines/CPU Min: 4048
; Lexemes/CPU-Min: 50253
; Memory Used: 221 pages
; Compilation Complete

