

The image is a large, symmetrical, abstract graphic composed of the letters 'S' and 'Y' arranged in a grid-like pattern. The letters are black and set against a white background. The overall shape is a wide, stylized 'W' or a complex letter 'M'. The top part of the graphic is formed by a series of 'S' characters, while the middle and bottom parts are formed by 'Y' characters. The letters are arranged in a way that creates a sense of depth and perspective, with some letters appearing to be in front of others. The graphic is composed of several vertical columns of letters, with the central column being the tallest and the outer columns being shorter. The letters are arranged in a way that creates a sense of rhythm and balance. The overall effect is a striking, minimalist design that is both visually appealing and intellectually stimulating.

[illegible]

(1)	42	Modification History
(2)	60	Declarations
(3)	83	FIL\$INIWCB - Allocate and Load WCB


```
0000 1 .TITLE FIL$INIWCB - Initialize Window Control Block
0000 2 .IDENT 'V04-000'
0000 3
0000 4 *****
0000 5
0000 6 * COPYRIGHT (c) 1978, 1980, 1982, 1984 BY
0000 7 * DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASSACHUSETTS.
0000 8 * ALL RIGHTS RESERVED.
0000 9
0000 10 * THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND COPIED
0000 11 * ONLY IN ACCORDANCE WITH THE TERMS OF SUCH LICENSE AND WITH THE
0000 12 * INCLUSION OF THE ABOVE COPYRIGHT NOTICE. THIS SOFTWARE OR ANY OTHER
0000 13 * COPIES THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY
0000 14 * OTHER PERSON. NO TITLE TO AND OWNERSHIP OF THE SOFTWARE IS HEREBY
0000 15 * TRANSFERRED.
0000 16
0000 17 * THE INFORMATION IN THIS SOFTWARE IS SUBJECT TO CHANGE WITHOUT NOTICE
0000 18 * AND SHOULD NOT BE CONSTRUED AS A COMMITMENT BY DIGITAL EQUIPMENT
0000 19 * CORPORATION.
0000 20
0000 21 * DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE OR RELIABILITY OF ITS
0000 22 * SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL.
0000 23
0000 24 *****
0000 25
0000 26
0000 27 ++
0000 28 Facility:
0000 29
0000 30 Executive, bootstrap support subroutines
0000 31
0000 32 Abstract:
0000 33
0000 34 This subroutine is called after a file has been opened by the
0000 35 FILEREAD routines to allocate and load a window control block
0000 36 that describes the file.
0000 37
0000 38 Environment:
0000 39
0000 40 This routine executes in kernel mode
0000 41
0000 42 .SBTTL Modification History
0000 43
0000 44 Author:
0000 45
0000 46 Peter H. Lipman
0000 47
0000 48 Creation Date:
0000 49
0000 50 23 September 1976
0000 51
0000 52 Modified By:
0000 53
0000 54 V01-001 LJK0209 Lawrence J. Kenah 21-Jun-1983
0000 55 The routine called MMG$INIWCB was removed from the original
0000 56 image activator system service, renamed to FIL$INIWCB, and
0000 57 placed into its own module.
```


FIL\$INIWCB
V04-000

- Initialize Window Control Block^{H 1}
Modification History

16-SEP-1984 00:12:39 VAX/VMS Macro V04-00
5-SEP-1984 03:42:13 [SYS.SRC]FILINIWCB.MAR;1

Page 2
(1)

0000 58 ;--

FOR
V04


```

0000 60          .SUBTITLE      Declarations
0000 61
0000 62 ; Include Files:
0000 63
0000 64          $DYNDEF          ; Dynamic structure type definitions
0000 65          $WCBDEF          ; Window control block definitions
0000 66
0000 67 ; External Symbols:
0000 68
0000 69          .DISABLE      GLOBAL
0000 70
0000 71 ; External Routine
0000 72
0000 73          .EXTERNAL      EXE$ALONONPAGED
0000 74
0000 75 ; External Global Data Cell
0000 76
0000 77          .EXTERNAL      EXE$GL_SYSWCBFL
0000 78
0000 79 ; External Status Code
0000 80
0000 81          .EXTERNAL      SS$_NORMAL      ; Success status code

```



```
0000 83 .SUBTITLE FIL$INIWCB - Allocate and Load WCB
0000 84 :+
0000 85 : Functional Description:
0000 86 :
0000 87 : When a file is opened by FILEREAD before the full file system is
0000 88 : functional, this routine is called to allocate and load a window
0000 89 : control block to describe the virtual to logical block mapping of that
0000 90 : file.
0000 91 :
0000 92 : Calling Sequence:
0000 93 :
0000 94 : JSB FIL$INIWCB
0000 95 :
0000 96 : Input Parameters:
0000 97 :
0000 98 : R1 = Number of bytes in retrieval pointer buffer
0000 99 :
0000 100 : R2 = Address of retrieval pointer buffer
0000 101 :
0000 102 : Entries are each 8 bytes consisting of a 4-byte block count
0000 103 : and a 4-byte logical block number
0000 104 :
0000 105 : R3 = UCB address
0000 106 :
0000 107 : Output Parameters:
0000 108 :
0000 109 : R2 = Window Control Block address if the routine returns successfully
0000 110 :
0000 111 : Implicit Output:
0000 112 :
0000 113 : The WCB is added to the linked list of system WCBs
0000 114 :
0000 115 : Completion Codes:
0000 116 :
0000 117 : R0 = SS$_NORMAL => WCB successfully allocated
0000 118 :
0000 119 : R0 low bit clear indicates a failure return from EXE$ALONONPAGED
0000 120 :
0000 121 : Side Effects:
0000 122 :
0000 123 : The contents of R1 are destroyed.
0000 124 : -
0000 125 :
00000000 126 .PSECT YFILEREAD BYTE,EXE
0000 127
0000 128 FIL$INIWCB::
51 08 C6 0000 129 DIVL #8,R1 ; Count of 8-byte retrieval pointers
3E BB 0003 130 PUSHF #^M<R1,R2,R3,R4,R5> ; Save input and working registers
54 D4 0005 131 CLRL R4 ; Initialize count of 6-byte retrieval
0007 132 ; pointers
50 62 0000FFFE 8F C1 0007 133 10$: ADDL3 #^XFFFE,(R2),R0 ; Get rounded block count
50 50 0000FFFF 8F C6 000F 134 DIVL #^XFFFF,R0 ; Number of required 6-byte pointers
54 50 C0 0016 135 ADDL R0,R4 ; Accumulate the count
52 08 C0 0019 136 ADDL #8,R2 ; Address of next 8-byte retrieval pointer
E8 51 F5 001C 137 SOBGTR R1,10$ ; Loop through them all
001F 138
51 54 06 C5 001F 139 MULL3 #6,R4,R1 ; Byte count for 6-byte pointers
```



```

51 30 C0 0023 140 ADDL #WCB$K_LENGTH,R1 ; Desired block size to allocate
00000000'GF 16 0026 141 JSB G^EXE$ALONONPAGED ; Allocate nonpaged dynamic memory
5E 50 E9 002C 142 BLBC R0,100$ ; Branch if failed to allocate
002F 143
002F 144 ; R1 = Allocated size
002F 145 ; R2 = Address of allocated area
002F 146
53 51 08 C7 002F 147 DIVL3 #8,R1,R3 ; Get truncated number of quadwords
0033 148 ; to fill with zeros
50 52 D0 0033 149 MOVL R2,R0 ; Make a copy of the WCB address
80 7C 0036 150 20$: CLRQ (R0)+ ; Clear the WCB
FB 53 F5 0038 151 SOBGTR R3,20$ ; (eight bytes at a time)
003B 152
08 A2 51 B0 003B 153 MOVW R1,WCB$W_SIZE(R2) ; Save allocated WCB size
12 90 003F 154 MOVB S^#DYN$C-WCB,-
OA A2 90 0041 155 WCB$B_TYPE(R2) ; Set type field
0043 156 MOVB #<WCB$M_READ ! - ; File accessed for reading
0044 157 WCB$M_NOTFCP>,- ; but not by FCP
0044 158 WCB$B_ACCESS(R2)
0B A2 05 0044 158 MOVW R4,WCB$W_NMAP(R2) ; Retrieval pointer count
16 A2 54 B0 0047 159 INCL WCB$L_STVBN(R2) ; Starting VBN is 1
2C A2 D6 004B 160 INSQUE (R2),G^EXE$GL_SYSWCBFL ; Maintain a list of these WCBs
00000000'GF 62 0E 004E 161 MOVQ (SP)+,R4
54 8E 7D 0055 162
0058 163
0058 164 ; R4 = Number of 8-byte retrieval pointers
0058 165 ; R5 = Addr of first 8-byte retrieval pointer
0058 166
52 52 DD 0058 167 PUSHL R2 ; Save WCB address
30 C0 005A 168 ADDL #WCB$W_P1_COUNT,R2 ; Starting addr in WCB to store
005D 169 ; retrieval pointers
005D 170
005D 171 ; The following double loop consists of an outer loop that iterates through
005D 172 ; the 8-byte retrieval pointers and an inner loop that makes as many 6-byte
005D 173 ; format retrieval pointers as necessary. Note that the 6-byte format is
005D 174 ; 2-bytes of block count and 4-bytes of logical block number. The block count
005D 175 ; holds 2*16-1 blocks because 0 really means 0.
005D 176
53 FFFF 8F 3C 005D 177 30$: MOVZWL #^XFFFF,R3 ; Maximum block count for 6-byte
0062 178 ; retrieval pointers
50 85 7D 0062 179 MOVQ (R5)+,R0
0065 180
0065 181 ; R0 = 4-byte block count
0065 182 ; R1 = Logical block number
0065 183
50 53 D1 0065 184 40$: CMPL R3,R0 ; Use the minimum block count
03 1B 0068 185 BLEQU 50$ ; Branch if maximum is the smaller
53 50 D0 006A 186 MOVL R0,R3 ; Maximum too big, use what's left
82 53 B0 006D 187 50$: MOVW R3,(R2)+ ; Store block count
82 51 D0 0070 188 MOVL R1,(R2)+ ; And starting LBN
51 53 C0 0073 189 ADDL R3,R1 ; Form next LBN
50 53 C2 0076 190 SUBL R3,R0 ; and remaining block count
EA 12 0079 191 BNEQ 40$ ; Branch if more blocks to map
DF 54 F5 007B 192 SOBGTR R4,30$ ; Branch if more 8-byte retrieval pointers
007E 193
3C BA 007E 194 POPR #^M<R2,R3,R4,R5> ; Restore saved registers
0080 195
0080 196 ; R2 = WCB address
```



```
0080 197 ; R3 = UCB address
0080 198
88 0080 199 BISB2 #<WCB$M_COMPLETE ! - ; File is completely mapped
0081 200 WCB$M_CATHEDRAL>,- ; and should remain completely mapped,
0081 201 WCB$B_ACCESS(R2) ; even if the file is extended
0085 202 MOVL R3,WCB$L_ORGUCB(R2) ; Set UCB address in window
0089 203 MOVZWL S^#SS$_NORMAL,R0 ; Indicate successful completion
008C 204 RSB ; and return
008D 205
008D 206 ; Error exit with R0 containing failure status code
008D 207
3E BA 008D 208 100$: POPR #^M<R1,R2,R3,R4,R5> ; Restore saved registers
05 008F 209 RSB ; and return
0090 210
0090 211 .END
```


FIL\$INIWCB
Symbol table

- Initialize Window Control Block M 1

16-SEP-1984 00:12:39 VAX/VMS Macro V04-00
5-SEP-1984 03:42:13 [SYS.SRC]FILINIWCB.MAR;1

Page 7
(3)

DYN\$C_WCB	= 00000012
EXE\$A\$CONONPAGED	***** X 00
EXE\$GL_SYSWCBFL	***** X 00
FIL\$INIWCB	00000000 RG 02
SS\$ NORMAL	***** X 00
WCB\$B_ACCESS	= 0000000B
WCB\$B_TYPE	= 0000000A
WCB\$K_LENGTH	= 00000030
WCB\$L_ORGUCB	= 00000010
WCB\$L_STVBN	= 0000002C
WCB\$M_CATHEDRAL	= 00000040
WCB\$M_COMPLETE	= 00000020
WCB\$M_NOTFCP	= 00000004
WCB\$M_READ	= 00000001
WCB\$W_NMAP	= 00000016
WCB\$W_P1_COUNT	= 00000030
WCB\$W_SIZE	= 00000008

+-----+
! Psect synopsis !
+-----+

PSECT name	Allocation	PSECT No.	Attributes															
. ABS .	00000000 (0.)	00 (0.)	NOPIC	USR	CON	ABS	LCL	NOSHR	NOEXE	NORD	NOWRT	NOVEC	BYTE					
\$AB\$\$	00000000 (0.)	01 (1.)	NOPIC	USR	CON	ABS	LCL	NOSHR	EXE	RD	WRT	NOVEC	BYTE					
YFILEREAD	00000090 (144.)	02 (2.)	NOPIC	USR	CON	REL	LCL	NOSHR	EXE	RD	WRT	NOVEC	BYTE					

+-----+
! Performance indicators !
+-----+

Phase	Page faults	CPU Time	Elapsed Time
Initialization	32	00:00:00.06	00:00:00.63
Command processing	107	00:00:00.57	00:00:03.23
Pass 1	160	00:00:02.51	00:00:11.53
Symbol table sort	0	00:00:00.24	00:00:00.28
Pass 2	53	00:00:00.65	00:00:02.80
Symbol table output	4	00:00:00.03	00:00:00.03
Psect synopsis output	1	00:00:00.01	00:00:00.54
Cross-reference output	0	00:00:00.00	00:00:00.00
Assembler run totals	359	00:00:04.08	00:00:19.04

The working set limit was 1200 pages.
13551 bytes (27 pages) of virtual memory were used to buffer the intermediate code.
There were 20 pages of symbol table space allocated to hold 230 non-local and 6 local symbols.
211 source lines were read in Pass 1, producing 13 object records in Pass 2.
9 pages of virtual memory were used to define 8 macros.

0375 AH-BT13A-SE
VAX/VMS V4.0

DIGITAL EQUIPMENT CORPORATION
CONFIDENTIAL AND PROPRIETARY

