


```

1 0001 0 MODULE INITIO (
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: INIT Utility Structure Level 1
34 0034 1
35 0035 1 ABSTRACT:
36 0036 1
37 0037 1     These routines do basic disk I/O.
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: 14-Nov-1977 19:42
48 0048 1
49 0049 1 MODIFIED BY:
50 0050 1
51 0051 1     V03-001 ACG0361 Andrew C. Goldstein, 21-Sep-1983 17:06
52 0052 1     Eliminate READ_PHYSICAL routine
53 0053 1
54 0054 1 **
55 0055 1
56 0056 1
57 0057 1 LIBRARY 'SYS$LIBRARY:LIB.L32';

```

INITIO
V04-000

H 1
16-Sep-1984 01:52:40
14-Sep-1984 12:35:18

VAX-11 BLISS-32 V4.0-742
DISK\$VMSMASTER:[INIT.SRC]INITIO.B32;1 Page 2 (1)

```
: 58      0058 1 REQUIRE 'SRCS:INIDEF.B32';  
: 59      0349 1 REQUIRE 'LIBDS:[VMSLIB.OBJ]INITMSG.B32';  
: 60      0481 1  
: 61      0482 1  
: 62      0483 1 FORWARD ROUTINE  
: 63      0484 1      READ_BLOCK,      ! read block by LBN  
: 64      0485 1      WRITE_BLOCK      : NOVALUE;      ! write block by LBN
```

INI
V04

```

66 0486 1 GLOBAL ROUTINE READ_BLOCK (LBN, BUFFER) =
67 0487 1
68 0488 1 :++
69 0489 1
70 0490 1 : FUNCTIONAL DESCRIPTION:
71 0491 1
72 0492 1 : This routine reads a disk block by logical block number.
73 0493 1
74 0494 1
75 0495 1 : CALLING SEQUENCE:
76 0496 1 : READ_BLOCK (ARG1, ARG2)
77 0497 1
78 0498 1 : INPUT PARAMETERS:
79 0499 1 : ARG1: logical block number
80 0500 1 : ARG2: buffer address
81 0501 1
82 0502 1 : IMPLICIT INPUTS:
83 0503 1 : CHANNEL: channel number assigned to disk
84 0504 1
85 0505 1 : OUTPUT PARAMETERS:
86 0506 1 : NONE
87 0507 1
88 0508 1 : IMPLICIT OUTPUTS:
89 0509 1 : NONE
90 0510 1
91 0511 1 : ROUTINE VALUE:
92 0512 1 : status of read
93 0513 1
94 0514 1 : SIDE EFFECTS:
95 0515 1 : block read into buffer
96 0516 1
97 0517 1 :--
98 0518 1
99 0519 2 BEGIN
100 0520 2
101 0521 2 LOCAL
102 0522 2 : STATUS, ! system service status
103 0523 2 : IO_STATUS : VECTOR [4, WORD]; ! I/O status block
104 0524 2
105 0525 2 EXTERNAL
106 0526 2 : CHANNEL; ! I/O channel number
107 0527 2
108 0528 2
109 P 0529 2 STATUS = $QIOW (CHAN = .CHANNEL,
110 P 0530 2 : FUNC = IOS_READBLK,
111 P 0531 2 : IOSB = IO_STATUS[0],
112 P 0532 2 : P1 = .BUFFER,
113 P 0533 2 : P2 = 512,
114 P 0534 2 : P3 = .LBN
115 0535 2 );
116 0536 2 IF .STATUS THEN STATUS = .IO_STATUS[0];
117 0537 2 RETURN .STATUS;
118 0538 2
119 0539 1 END; ! end of routine READ_BLOCK

```

.TITLE INITIO

INITIO
V04-000

IN
V04

```

.IDENT \V04-000\
.EXTRN CHANNEL, SYSSQIOW
.PSECT $CODE$,NOWRT,2
.ENTRY READ_BLOCK, Save nothing          : 0486
SUBL2 #8, SP
CLRQ -(SP)                                : 0535
CLRQ -(SP)
PUSHL LBN
MOVZWL #512, -(SP)
PUSHL BUFFER
CLRQ -(SP)
PUSHAB IO_STATUS
PUSHL #33
PUSHL CHANNEL
CLRQ -(SP)
CALLS #2, SYSSQIOW
BLBC STATUS, 1$                            : 0536
MOVZWL IO_STATUS, STATUS
RET                                          : 0539

```

5E		0000	00000	
		08	C2 00002	
		7E	7C 00005	
		7E	D4 00007	
	04	AC	DD 00009	
7E	0200	8F	3C 0000C	
	08	AC	DD 00011	
		7E	7C 00014	
	20	AE	9F 00016	
		21	DD 00019	
	0000G	CF	DD 0001B	
		7E	D4 0001F	
00000000G	00	0C	FB 00021	
	03	50	E9 00028	
	50	6E	3C 0002B	
		04	0002E 1\$:	

; Routine Size: 47 bytes, Routine Base: \$CODE\$ + 0000

```

121 0540 1 GLOBAL ROUTINE WRITE_BLOCK (LBN, BUFFER) : NOVALUE =
122 0541 1
123 0542 1 !++
124 0543 1
125 0544 1 FUNCTIONAL DESCRIPTION:
126 0545 1
127 0546 1     This routine writes a disk block by logical block number.
128 0547 1
129 0548 1
130 0549 1 CALLING SEQUENCE:
131 0550 1     WRITE_BLOCK (ARG1, ARG2)
132 0551 1
133 0552 1 INPUT PARAMETERS:
134 0553 1     ARG1: logical block number
135 0554 1     ARG2: buffer address
136 0555 1
137 0556 1 IMPLICIT INPUTS:
138 0557 1     CHANNEL: channel number assigned to disk
139 0558 1
140 0559 1 OUTPUT PARAMETERS:
141 0560 1     NONE
142 0561 1
143 0562 1 IMPLICIT OUTPUTS:
144 0563 1     NONE
145 0564 1
146 0565 1 ROUTINE VALUE:
147 0566 1     status of write
148 0567 1
149 0568 1 SIDE EFFECTS:
150 0569 1     block written from buffer
151 0570 1
152 0571 1 !--
153 0572 1
154 0573 2 BEGIN
155 0574 2
156 0575 2 LOCAL
157 0576 2     STATUS,                ! system service status
158 0577 2     IO_STATUS             : VECTOR [4, WORD]; ! I/O status block
159 0578 2
160 0579 2 EXTERNAL
161 0580 2     CHANNEL;                ! I/O channel number
162 0581 2
163 0582 2
164 P 0583 2 STATUS = $QIOW (CHAN = .CHANNEL,
165 PP 0584 2     FUNC = IOS_WRITEBLK OR IOSM_DATACHECK,
166 PP 0585 2     IOSB = IO_STATUS[0],
167 PP 0586 2     P1 = .BUFFER,
168 PP 0587 2     P2 = 512,
169 P 0588 2     P3 = .LBN
170 0589 2 );
171 0590 2 IF .STATUS THEN STATUS = .IO_STATUS[0];
172 0591 2 IF NOT .STATUS
173 0592 2 THEN ERR_EXIT (.STATUS);
174 0593 2
175 0594 1 END;                ! end of routine WRITE_BLOCK.

```

		0000	00000	.ENTRY	WRITE BLOCK, Save nothing	: 0540
5E		08	C2 00002	SUBL2	#8, SP	
		7E	7C 00005	CLRQ	-(SP)	: 0589
		7E	D4 00007	CLRL	-(SP)	
	04	AC	DD 00009	PUSHL	LBN	
7E	0200	8F	3C 0000C	MOVZWL	#512, -(SP)	
	08	AC	DD 00011	PUSHL	BUFFER	
		7E	7C 00014	CLRQ	-(SP)	
	20	AE	9F 00016	PUSHAB	IO STATUS	
7E	4020	8F	3C 00019	MOVZWL	#16416, -(SP)	
	0000G	CF	DD 0001E	PUSHL	CHANNEL	
		7E	D4 00022	CLRL	-(SP)	
00000000G	00	0C	FB 00024	CALLS	#12, SYSSQIOW	
	06	50	E9 0002B	BLBC	STATUS, 1\$: 0590
	50	6E	3C 0002E	MOVZWL	IO STATUS, STATUS	
	09	50	E8 00031	BLBS	STATUS, 2\$: 0591
		50	DD 00034 1\$:	PUSHL	STATUS	: 0592
00000000G	00	01	FB 00036	CALLS	#1, LIB\$STOP	
		04	0003D 2\$:	RET		: 0594

: Routine Size: 62 bytes, Routine Base: \$CODE\$ + 002F

: 176 0595 1
: 177 0596 1 END
: 178 0597 0 ELUDOM

.EXTRN LIB\$STOP

PSECT SUMMARY

Name	Bytes	Attributes
\$CODE\$	109	NOVEC, NOWRT, RD, EXE, NOSHR, LCL, REL, CON, NOPIC, ALIGN(2)

Library Statistics

File	Symbols		Pages Mapped	Processing Time
	Total	Loaded Percent		
_\$255\$DUA28:[SYSLIB]LIB.L32;1	18619	10 0	1000	00:01.9

0188 AH-BT13A-SE
VAX/VMS V4.0

DIGITAL EQUIPMENT CORPORATION
CONFIDENTIAL AND PROPRIETARY

INPSMB
MAP

INSDEF
SQL

INPSMBMSG
LIS

RSXLBDF
SQL

INSCREATE
LIS

INITIO
LIS

INSTAL

INSTALLS
MAP

INSCMD
CLD

INSPREFIX
REQ

INPSMBCLD
CLD

INPSMB
LIS

INSOLDCMD
CLD

INITIO
LIS

RDHOME
LIS

INPSMB

INPSMBCLD
LIS

INSCMD
LIS