

```

FFFFFFFFFFFFFFFF
FFFFFFFFFFFFFFFF
FFFFFFFFFFFFFFFF
FFF
FFF
FFF
FFF
FFF
FFF
FFFFFFFFFFFFFF
FFFFFFFFFFFFFF
FFFFFFFFFFFFFF
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
111111111
111111111
111111111

```

```

111
111
111
111111
111111
111111
111
111
111
111
111
111
111
111
111
111
111
111
111
111
111111111
111111111
111111111

```

```

AAAAAAAAA
AAAAAAAAA
AAAAAAAAA
AAA      AAA
AAA      AAA
AAA      AAA
AAA      AAA
AAA      AAA
AAA      AAA
AAA      AAA
AAA      AAA
AAA      AAA
AAA      AAA
AAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAA
AAA      AAA
AAA      AAA
AAA      AAA
AAA      AAA
AAA      AAA
AAA      AAA
AAA      AAA

```

```

RRRRRRRR      DDDDDDDD      HH      HH      EEEEEEEEEEE      DDDDDDDD      RRRRRRRR
RRRRRRRR      DDDDDDDD      HH      HH      EEEEEEEEEEE      DDDDDDDD      RRRRRRRR
RR      RR      DD      DD      HH      HH      EE      DD      DD      RR      RR
RR      RR      DD      DD      HH      HH      EE      DD      DD      RR      RR
RR      RR      DD      DD      HH      HH      EE      DD      DD      RR      RR
RR      RR      DD      DD      HH      HH      EE      DD      DD      RR      RR
RRRRRRRR      DD      DD      HHHHHHHHHH      EEEEEEEEE      DD      DD      RRRRRRRR
RRRRRRRR      DD      DD      HHHHHHHHHH      EEEEEEEEE      DD      DD      RRRRRRRR
RR      RR      DD      DD      HH      HH      EE      DD      DD      RR      RR
RR      RR      DD      DD      HH      HH      EE      DD      DD      RR      RR
RR      RR      DD      DD      HH      HH      EE      DD      DD      RR      RR
RR      RR      DD      DD      HH      HH      EE      DD      DD      RR      RR
RR      RR      DDDDDDDD      HH      HH      EEEEEEEEEEE      DDDDDDDD      RR      RR
RR      RR      DDDDDDDD      HH      HH      EEEEEEEEEEE      DDDDDDDD      RR      RR

```

```

LL      IIIIII      SSSSSSSS
LL      IIIIII      SSSSSSSS
LL      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  
V  
  
M  
-  
T  
9  
T  
M

```

1 0001 0 MODULE RDHEDR (
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 * 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 1
34 0034 1
35 0035 1 ABSTRACT:
36 0036 1
37 0037 1 This routine reads the desired file header, checks it for
38 0038 1 validity and correctness, and returns its address.
39 0039 1
40 0040 1 ENVIRONMENT:
41 0041 1
42 0042 1 STARLET operating system, including privileged system services
43 0043 1 and internal exec routines.
44 0044 1
45 0045 1 --
46 0046 1
47 0047 1
48 0048 1 AUTHOR: Andrew C. Goldstein, CREATION DATE: 13-Dec-1976 22:00
49 0049 1
50 0050 1 MODIFIED BY:
51 0051 1
52 0052 1 A0102 ACG0120 Andrew C. Goldstein, 16-Jan-1980 21:16
53 0053 1 Reorder header consistency checking
54 0054 1
55 0055 1 A0101 ACG0083 Andrew C. Goldstein, 15-Nov-1979 1:02
56 0056 1 Invalidate file header if bad
57 0057 1

```

RDHEDR  
V04-000

J 10  
16-Sep-1984 01:14:50  
14-Sep-1984 12:29:48

VAX-11 Bliss-32 V4.0-742  
DISK\$VMSMASTER:[F11A.SRC]RDHEDR.B32;1 Page (1) 2

:	58	0058	1	:	A0100	ACG00001	Andrew C. Goldstein,	10-Oct-1978	20:03
:	59	0059	1	:			Previous revision history moved to F11A.REV		
:	60	0060	1	:	**				
:	61	0061	1	:					
:	62	0062	1	:					
:	63	0063	1	:	LIBRARY	'SYSS\$LIBRARY:LIB.L32';			
:	64	0064	1	:	REQUIRE	'SRC\$:FCPDEF.B32';			

```

: 66 0379 1 GLOBAL ROUTINE READ_HEADER (FILE_ID, FCB) =
: 67 0380 1
: 68 0381 1 !++
: 69 0382 1
: 70 0383 1 FUNCTIONAL DESCRIPTION:
: 71 0384 1
: 72 0385 1 This routine reads the desired file header, checks it for
: 73 0386 1 validity and correctness, and returns its address.
: 74 0387 1
: 75 0388 1 CALLING SEQUENCE:
: 76 0389 1 READ_HEADER (ARG1, ARG2)
: 77 0390 1
: 78 0391 1 INPUT PARAMETERS:
: 79 0392 1 ARG1: address of file ID or 0
: 80 0393 1 ARG2: FCB address or 0 if none
: 81 0394 1
: 82 0395 1 IMPLICIT INPUTS:
: 83 0396 1 CURRENT_VCB contains address of VCB in process
: 84 0397 1
: 85 0398 1 OUTPUT PARAMETERS:
: 86 0399 1 NONE
: 87 0400 1
: 88 0401 1 IMPLICIT OUTPUTS:
: 89 0402 1 HEADER_LBN contains LBN of header read
: 90 0403 1 FILE_HEADER contains address of header buffer
: 91 0404 1
: 92 0405 1 ROUTINE VALUE:
: 93 0406 1 address of file header
: 94 0407 1
: 95 0408 1 SIDE EFFECTS:
: 96 0409 1 index file window may be turned
: 97 0410 1
: 98 0411 1 --
: 99 0412 1
100 0413 2 BEGIN
101 0414 2
102 0415 2 MAP
103 0416 2 FILE_ID : REF BBLOCK, ! file ID arg
104 0417 2 FCB : REF BBLOCK; ! FCB arg
105 0418 2
106 0419 2 LOCAL
107 0420 2 VBN, ! VBN of header
108 0421 2 LBN, ! LBN of header
109 0422 2 HEADER : REF BBLOCK, ! address of header block
110 0423 2 FID : REF BBLOCK, ! local file ID pointer
111 0424 2 IDX_FCB : REF BBLOCK; ! address of index file FCB
112 0425 2
113 0426 2 EXTERNAL
114 0427 2 CLEANUP_FLAGS : BITVECTOR, ! cleanup_action flags
115 0428 2 HEADER_LBN, ! longword to get LBN of header
116 0429 2 FILE_HEADER : REF BBLOCK, ! longword to get buffer address
117 0430 2 CURRENT_VCB : REF BBLOCK; ! address of VCB in process
118 0431 2
119 0432 2 EXTERNAL ROUTINE
120 0433 2 MAP_VBN, ! map virtual to logical
121 0434 2 READ_BLOCK, ! read a disk block
: 122 0435 2 CHECK_HEADER, ! check header for correctness

```

```

: 123 0436 2 INVALIDATE; ! invalidate block buffer
: 124 0437 2
: 125 0438 2 ! Get the LBN of the file header. If an FCB is supplied, it contains
: 126 0439 2 ! the LBN. If not, derive it from the file number.
: 127 0440 2
: 128 0441 2
: 129 0442 2 LBN =
: 130 0443 2 BEGIN
: 131 0444 2 IF .FCB NEQ 0
: 132 0445 2 THEN .FCB[FCB$L_HDLBN]
: 133 0446 2 ELSE
: 134 0447 2 BEGIN
: 135 0448 2 IF .FILE_ID[FID$W_NUM] EQL 0 THEN ERR_EXIT (SS$ NOSUCHFILE);
: 136 0449 2 VBN = .FILE_ID[FID$W_NUM] + .CURRENT_VCB[VCB$B_IBMAPSIZE] + 2;
: 137 0450 2 IDX_FCB = .CURRENT_VCB[VCB$L_FCBFL];
: 138 0451 2 MAP_VBN (.VBN, .IDX_FCB[FCB$L_WLFL]);
: 139 0452 2 END
: 140 0453 2 END;
: 141 0454 2
: 142 0455 2 IF .LBN EQL -1 THEN ERR_EXIT (SS$ NOSUCHFILE);
: 143 0456 2
: 144 0457 2 ! Now read the header and check it for correctness. If a file ID
: 145 0458 2 ! was supplied, use the file number and file sequence number from
: 146 0459 2 ! it; else use the arguments. If the file operation is being done on a
: 147 0460 2 ! spooled device, the file must be marked as spooled.
: 148 0461 2
: 149 0462 2
: 150 0463 2 HEADER = READ_BLOCK (.LBN, 1, HEADER_TYPE);
: 151 0464 2
: 152 0465 2 IF .FILE_ID NEQ 0
: 153 0466 2 THEN
: 154 0467 2 FID = .FILE_ID
: 155 0468 2 ELSE
: 156 0469 2 FID = FCB[FCB$W_FID];
: 157 0470 2
: 158 0471 2 IF NOT CHECK_HEADER (.HEADER, .FID)
: 159 0472 2 THEN
: 160 0473 2 BEGIN
: 161 0474 2 INVALIDATE (.HEADER);
: 162 0475 2 ERR_EXIT ();
: 163 0476 2 END;
: 164 0477 2
: 165 0478 2 IF .CLEANUP_FLAGS[CLF_SPOOLFILE]
: 166 0479 2 AND NOT .HEADER[FH1$V_SPOOL]
: 167 0480 2 AND .HEADER[FH1$W_FID_NUM] GTRU 5
: 168 0481 2 THEN ERR_EXIT (SS$ NOSUCHFILE);
: 169 0482 2
: 170 0483 2 HEADER_LBN = .LBN; ! return LBN of header
: 171 0484 2 FILE_HEADER = .HEADER; ! and address
: 172 0485 2 RETURN .HEADER; ! and the header itself
: 173 0486 1 END; ! end of routine READ_HEADER

```

```

.TITLE RDHEDR
.IDENT \V04-000\
.EXTRN CLEANUP_FLAGS, HEADER_LBN

```

					.EXTRN	FILE HEADER, CURRENT_VCB		
					.EXTRN	MAP_VBN, READ_BLOCK		
					.EXTRN	CHECK_HEADER, INVALIDATE		
					.PSECT	\$CODE\$,NOWRT,2		
			000C	00000	.ENTRY	READ_HEADER, Save R2,R3		0379
	50	08	AC	D0 00002	MOVL	FCB, R0		0444
			06	13 00006	BEQL	1\$		
	53	34	A0	D0 00008	MOVL	52(R0), LBN		0445
			28	11 0000C	BRB	2\$		
		04	BC	B5 0000E 1\$:	TSTW	@FILE_ID		0448
			70	13 00011	BEQL	6\$		
	51	0000G	CF	D0 00013	MOVL	CURRENT_VCB, R1		0449
	50		04	BC 3C 00018	MOVZWL	@FILE_ID, R0		
	52		38	A1 9A 0001C	MOVZBL	56(R1), R2		
	50		52	C0 00020	ADDL2	R2, R0		
	50		02	C0 00023	ADDL2	#2, VBN		
	51		61	D0 00026	MOVL	(R1), IDX_FCB		0450
		10	A1	DD 00029	PUSHL	16(IDX_FCB)		0451
			50	DD 0002C	PUSHL	VBN		
	0000G	CF	02	FB 0002E	CALLS	#2, MAP_VBN		
	53		50	D0 00033	MOVL	R0, LBN		
	FFFFFFF	8F	53	D1 00036 2\$:	CML	LBN, #-1		0455
			44	13 0003D	BEQL	6\$		
		7E	01	7D 0003F	MOVQ	#1, -(SP)		0463
			53	DD 00042	PUSHL	LBN		
	0000G	CF	03	FB 00044	CALLS	#3, READ_BLOCK		
	52		50	D0 00049	MOVL	R0, HEADER		
			04	AC D5 0004C	TSTL	FILE_ID		0465
			06	13 0004F	BEQL	3\$		
	50		04	AC D0 00051	MOVL	FILE_ID, FID		0467
			05	11 00055	BRB	4\$		
	50	08	AC	C1 00057 3\$:	ADDL3	#36, FCB, FID		0469
			50	DD 0005C 4\$:	PUSHL	FID		0471
			52	DD 0005E	PUSHL	HEADER		
	0000G	CF	02	FB 00060	CALLS	#2, CHECK_HEADER		
		0A	50	E8 00065	BLBS	R0, 5\$		
			52	DD 00068	PUSHL	HEADER		0474
	0000G	CF	01	FB 0006A	CALLS	#1, INVALIDATE		
			00	BF 0006F	CHMU	#0		0475
			04	00071	RET			
		0000G	CF	95 00072 5\$:	TSTB	CLEANUP_FLAGS		0478
			10	18 00076	BGEQ	7\$		
	0B	0D	A2	04 E0 00078	BBS	#4, 13(HEADER), 7\$		0479
			05	02 A2 B1 0007D	CMPW	2(HEADER), #5		0480
			05	1B 00081	BLEQU	7\$		
		0910	8F	BF 00083 6\$:	CHMU	#2320		0481
			04	00087	RET			
	0000G	CF	53	D0 00088 7\$:	MOVL	LBN, HEADER_LBN		0483
	0000G	CF	52	D0 0008D	MOVL	HEADER, FILE_HEADER		0484
		50	52	D0 00092	MOVL	HEADER, R0		0485
			04	00095	RET			0486

; Routine Size: 150 bytes, Routine Base: \$CODE\$ + 0000

RDHEDR  
V04-000

N 10  
16-Sep-1984 01:14:50  
14-Sep-1984 12:29:48

VAX-11 Bliss-32 V4.0-742  
DISK\$VMMASTER:[F11A.SRC]RDHEDR.B32;1 Page 6 (2)

: 174 0487 1  
: 175 0488 1 END  
: 176 0489 0 ELUDOM

PSECT SUMMARY

: Name Bytes Attributes  
: \$CODE\$ 150 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	10	0	1000	00:01.9

COMMAND QUALIFIERS

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

: Size: 150 code + 0 data bytes  
: Run Time: 00:07.4  
: Elapsed Time: 00:21.5  
: Lines/CPU Min: 3970  
: Lexemes/CPU-Min: 12738  
: Memory Used: 94 pages  
: Compilation Complete



The image displays a grid of 100 terminal windows, arranged in 10 rows and 10 columns. Each window contains a program name followed by the letters 'LIS'. The programs are: Row 1: REQUEL LIS, RWATTR LIS; Row 2: MOOTFY LIS; Row 3: SCHFCB LIS; Row 4: MAKACC LIS; Row 5: MPWIND LIS; Row 6: MAPUBN LIS, PMS LIS, RDHEDR LIS, RWJB LIS; Row 7: RETDIR LIS; Row 8: ROBLOK LIS; Row 9: SMALOC LIS; Row 10: MAKMBE LIS, MAKSTR LIS, MATHOR LIS. The background of each window is dark with light-colored text, and the overall image has a dark, almost black, background.