



```

NN      NN  XX      XX  TTTTTTTTTT  HH      HH  DDDDDDDD  RRRRRRRR
NN      NN  XX      XX  TTTTTTTTTT  HH      HH  DDDDDDDD  RRRRRRRR
NN      NN  XX      XX      TT      HH      HH  DD      DD  RR      RR
NN      NN  XX      XX      TT      HH      HH  DD      DD  RR      RR
NNNN    NN  XX  XX  TT      HH      HH  DD      DD  RR      RR
NNNN    NN  XX  XX  TT      HH      HH  DD      DD  RR      RR
NN  NN  NN      XX  TT      HHHHHHHHHH  DD      DD  RRRRRRRR
NN  NN  NN      XX  TT      HHHHHHHHHH  DD      DD  RRRRRRRR
NN      NNNN  XX  XX  TT      HH      HH  DD      DD  RR  RR
NN      NNNN  XX  XX  TT      HH      HH  DD      DD  RR  RR
NN      NN  XX      XX  TT      HH      HH  DD      DD  RR  RR
NN      NN  XX      XX  TT      HH      HH  DD      DD  RR  RR
NN      NN  XX      XX  TT      HH      HH  DDDDDDDD  RR      RR
NN      NN  XX      XX  TT      HH      HH  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
LL.LLLLLL  IIIIII  SSSSSSSS
LLLLLLLLLL  IIIIII  SSSSSSSS

```

```

1 0001 0 MODULE NXTHDR (
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 n. . extension header, if any, of the
38 0038 1     given file.
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: 22-Jul-1977 17:40
49 0049 1
50 0050 1 MODIFIED BY:
51 0051 1
52 0052 1     A0100  ACG001      Andrew C. Goldstein, 10-Oct-1978 20:01
53 0053 1     Previous revision history moved to F11A.REV
54 0054 1
55 0055 1 **
56 0056 1
57 0057 1

```

NXTHDR  
V04-000

C 6  
16-Sep-1984 01:12:24  
14-Sep-1984 12:29:47

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

: 58  
: 59

0058 1 LIBRARY 'SYSSLIBRARY:LIB.L32';  
0059 1 REQUIRE 'SRC\$:FCPDEF.B32';

PM  
VO

```

: 61 0374 1 GLOBAL ROUTINE NEXT_HEADER (HEADER, FCB, EXT_FID, SEGNUM) =
: 62 0375 1
: 63 0376 1 : **
: 64 0377 1
: 65 0378 1 FUNCTIONAL DESCRIPTION:
: 66 0379 1
: 67 0380 1 This routine reads the next extension header, if any, of the
: 68 0381 1 indicated file. Extension data is taken from either the indicated
: 69 0382 1 file header or the arguments.
: 70 0383 1
: 71 0384 1
: 72 0385 1 CALLING SEQUENCE:
: 73 0386 1 NEXT_HEADER (ARG1, ARG2, ARG3, ARG4)
: 74 0387 1
: 75 0388 1 INPUT PARAMETERS:
: 76 0389 1 ARG1: address of current file header or 0
: 77 0390 1 ARG2: address of corresponding FCB or zero
: 78 0391 1 ARG3: extension file ID, if present
: 79 0392 1 ARG4: extension segment number, if present
: 80 0393 1
: 81 0394 1 IMPLICIT INPUTS:
: 82 0395 1 NONE
: 83 0396 1
: 84 0397 1 OUTPUT PARAMETERS:
: 85 0398 1 NONE
: 86 0399 1
: 87 0400 1 IMPLICIT OUTPUTS:
: 88 0401 1 NONE
: 89 0402 1
: 90 0403 1 ROUTINE VALUE:
: 91 0404 1 Address of header read or 0 if none
: 92 0405 1
: 93 0406 1 SIDE EFFECTS:
: 94 0407 1 File header may be read
: 95 0408 1
: 96 0409 1 --
: 97 0410 1
: 98 0411 2 BEGIN
: 99 0412 2
: 100 0413 2 MAP
: 101 0414 2 HEADER : REF BBLOCK, ! file header arg
: 102 0415 2 FCB : REF BBLOCK, ! FCB arg
: 103 0416 2 EXT_FID : REF BBLOCK; ! extension file ID arg
: 104 0417 2
: 105 0418 2 LOCAL
: 106 0419 2 NEW_HEADER : REF BBLOCK, ! address of extension file header read
: 107 0420 2 MAP_AREA : REF BBLOCK, ! address of header map area
: 108 0421 2 EXT_FCB : REF BBLOCK, ! address of extension FCB
: 109 0422 2 FILE_ID : BBLOCK [FID$C_LENGTH], ! file ID of extension header
: 110 0423 2 SEG_NUMBER : BYTE; ! segment number of file header
: 111 0424 2
: 112 0425 2 EXTERNAL ROUTINE
: 113 0426 2 READ_HEADER; ! read a file header
: 114 0427 2
: 115 0428 2
: 116 0429 2 ! Get the extension file number of the file header. If it is zero, then
: 117 0430 2 ! there is no extension header. If it is non-zero, read the header, using

```

```

118 0431 2 ! the extension FCB if one exists.
119 0432 2 !
120 0433 2 !
121 0434 2 IF ACTUALCOUNT LSS 4
122 0435 2 THEN
123 0436 2 BEGIN
124 0437 2 MAP_AREA = .HEADER + .HEADER[FH1$B MPOFFSET]*2;
125 0438 2 FILE_ID[FID$W_NUM] = .MAP_AREA[FM1$W_EX_FILNUM];
126 0439 2 FILE_ID[FID$W_SEQ] = .MAP_AREA[FM1$W_EX_FILSEQ];
127 0440 2 FILE_ID[FID$W_RVN] = 0;
128 0441 2 SEG_NUMBER = .MAP_AREA[FM1$B_EX_SEGNUM] + 1;
129 0442 2 END
130 0443 2 ELSE
131 0444 2 BEGIN
132 0445 2 CH$MOVE (FID$C_LENGTH, .EXT_FID, FILE_ID);
133 0446 2 SEG_NUMBER = .SEGNUM;
134 0447 2 END;
135 0448 2
136 0449 2 IF .FILE_ID[FID$W_NUM] EQL 0 THEN RETURN 0;
137 0450 2 EXT_FCB =
138 0451 2 (IF .FCB NEQ 0
139 0452 2 THEN .FCB[FCB$L_EXFCB]
140 0453 2 ELSE 0
141 0454 2 );
142 0455 2 NEW_HEADER = READ_HEADER (FILE_ID, .EXT_FCB);
143 0456 2
144 0457 2 ! Check the segment number of the header read for consistency.
145 0458 2 !
146 0459 2
147 0460 2 MAP_AREA = .NEW_HEADER + .NEW_HEADER[FH1$B MPOFFSET]*2;
148 0461 2 IF .SEG_NUMBER NEQ .MAP_AREA[FM1$B_EX_SEGNUM]
149 0462 2 THEN ERR_EXIT (SS$_BADFILEHDR);
150 0463 2
151 0464 2 RETURN .NEW_HEADER;
152 0465 2
153 0466 1 END;
! end of routine NEXT_HEADER

```

				.TITLE	NXTHDR	
				.IDENT	\V04-000\	
				.EXTRN	READ_HEADER	
				.PSECT	\$CODE\$,NOWRT,2	
				.ENTRY	NEXT_HEADER, Save R2,R3,R4,R5	: 0374
				SUBL2	#8, SP	
				CMPB	(AP), #4	: 0434
				BGEQU	1\$	
				MOVL	HEADER, R1	: 0437
				MOVZBL	1(R1), R0	
				MOVAW	(R1)[R0], MAP_AREA	
				MOVL	2(MAP_AREA), FILE_ID	: 0438
				CLRW	FILE_ID+4	: 0440
				ADDB3	#1, (MAP_AREA), SEG_NUMBER	: 0441
				BRB	2\$	: 0434
				MOVCS	#6, @EXT_FID, FILE_ID	: 0445

			003C	0000
5E			08	C2 00002
04			6C	91 00005
			19	1E 00008
51	04		AC	D0 0000A
50	01		A1	9A 0000E
52		6140	3E	00012
6E	02		A2	D0 00016
		04	AE	B4 0001A
53	62		01	81 0001D
			09	11 00021
6E	0C	BC	06	28 00023 1\$:

Address	Op Code	Op Name	Operand 1	Operand 2	Instruction	Address
53	10	AC	90	00028	MOVB	SEGNUM, SEG_NUMBER : 0446
		6E	B5	0002C	TSTW	FILE_ID : 0449
		31	13	0002E	BEQL	6\$ : 0451
50	08	AC	D0	00030	MOVL	FCB, R0 : 0452
		06	13	00034	BEQL	3\$ : 0455
50	0C	A0	D0	00036	MOVL	12(R0), EXT_FCB : 0460
		02	11	0003A	BRB	4\$ : 0461
		50	D4	0003C	CLRL	EXT_FCB : 0462
		50	DD	0003E	PUSHL	EXT_FCB : 0464
		AE	9F	00040	PUSHAB	FILE_ID : 0466
0000G	CF	02	FB	00043	CALLS	#2, READ HEADER
	51	50	D0	00048	MOVL	R0, NEW_HEADER
	50	01	A1	9A	MOVZBL	1(NEW_HEADER), R0 : 0460
	52		6140	3E	MOVAW	(NEW_HEADER)[R0], MAP_AREA
	62		53	91	CMPB	SEG_NUMBER, (MAP_AREA) : 0461
			05	13	BEQL	5\$ : 0462
		0810	8F	BF	CHMU	#2064 : 0464
			04	0005C	RET	
	50		51	D0	MOVL	NEW_HEADER, R0 : 0466
			04	00060	RET	
			50	D4	CLRL	R0 : 0466
			04	00063	RET	

; Routine Size: 100 bytes, Routine Base: \$CODE\$ + 0000

```

: 154      0467 1
: 155      0468 1 END
: 156      0469 0 ELUDOM

```

PSECT SUMMARY

Name	Bytes	Attributes
\$CODE\$	100	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	11 0	1000	00:01.9

NXTHDR  
V04-000

G 6  
16-Sep-1984 01:12:24  
14-Sep-1984 12:29:47

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

PM  
V04

COMMAND QUALIFIERS

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

: Size: 100 code + 0 data bytes  
: Run Time: 00:06.5  
: Elapsed Time: 00:17.9  
: Lines/CPU Min: 4315  
: Lexemes/CPU-Min: 13628  
: Memory Used: 88 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, MXTHOR LIS. The background of each window is dark with light-colored text, and the overall image has a dark, almost black, background.