

BBBBBBBBBBBB		AAAAAAAAA		SSSSSSSSSS		RRRRRRRRRR		TTTTTTTTTT		LLL
BBBBBBBBBBBB		AAAAAAAAA		SSSSSSSSSS		RRRRRRRRRR		TTTTTTTTTT		LLL
BBBBBBBBBBBB		AAAAAAAAA		SSSSSSSSSS		RRRRRRRRRR		TTTTTTTTTT		LLL
BBB	BBB	AAA	AAA	SSS		RRR	RRR	TTT		LLL
BBB	BBB	AAA	AAA	SSS		RRR	RRR	TTT		LLL
BBB	BBB	AAA	AAA	SSS		RRR	RRR	TTT		LLL
BBB	BBB	AAA	AAA	SSS		RRR	RRR	TTT		LLL
BBB	BBB	AAA	AAA	SSS		RRR	RRR	TTT		LLL
BBB	BBB	AAA	AAA	SSS		RRR	RRR	TTT		LLL
BBBBBBBBBBBB		AAA	AAA	SSS	SSSSSSSS	RRRRRRRRRR		TTT		LLL
BBBBBBBBBBBB		AAA	AAA	SSS	SSSSSSSS	RRRRRRRRRR		TTT		LLL
BBBBBBBBBBBB		AAA	AAA	SSS	SSSSSSSS	RRRRRRRRRR		TTT		LLL
BBB	BBB	AAAAAAAAAAAA		SSS		RRR	RRR	TTT		LLL
BBB	BBB	AAAAAAAAAAAA		SSS		RRR	RRR	TTT		LLL
BBB	BBB	AAAAAAAAAAAA		SSS		RRR	RRR	TTT		LLL
BBB	BBB	AAA	AAA	SSS		RRR	RRR	TTT		LLL
BBB	BBB	AAA	AAA	SSS		RRR	RRR	TTT		LLL
BBB	BBB	AAA	AAA	SSS		RRR	RRR	TTT		LLL
BBB	BBB	AAA	AAA	SSS		RRR	RRR	TTT		LLL
BBBBBBBBBBBB		AAA	AAA	SSSSSSSSSS		RRR	RRR	TTT		LLLLLLLLLLLL
BBBBBBBBBBBB		AAA	AAA	SSSSSSSSSS		RRR	RRR	TTT		LLLLLLLLLLLL
BBBBBBBBBBBB		AAA	AAA	SSSSSSSSSS		RRR	RRR	TTT		LLLLLLLLLLLL

```

BBBBBBBBB      AAAAAA      SSSSSSSSS  UU      UU  NN      NN  LL      000000      CCCCCCCC  KK      KK
BBBBBBBBB      AAAAAA      SSSSSSSSS  UU      UU  NN      NN  LL      000000      CCCCCCCC  KK      KK
BB      BB  AA      AA  SS      UU      UU  NN      NN  LL      00      00  CC      KK      KK
BB      BB  AA      AA  SS      UU      UU  NN      NN  LL      00      00  CC      KK      KK
BB      BB  AA      AA  SS      UU      UU  NNNN     NN  LL      00      00  CC      KK      KK
BB      BB  AA      AA  SS      UU      UU  NNNN     NN  LL      00      00  CC      KK      KK
BBBBBBBBB      AA      AA  SSSSSSS  UU      UU  NN  NN  NN  LL      00      00  CC      KKKKKK
BBBBBBBBB      AA      AA  SSSSSSS  UU      UU  NN  NN  NN  LL      00      00  CC      KKKKKK
BB      BB  AAAAAAAAAA      SS      UU      UU  NN      NN  LL      00      00  CC      KK      KK
BB      BB  AAAAAAAAAA      SS      UU      UU  NN      NN  LL      00      00  CC      KK      KK
BB      BB  AA      AA  SS      UU      UU  NN      NN  LL      00      00  CC      KK      KK
BB      BB  AA      AA  SS      UU      UU  NN      NN  LL      00      00  CC      KK      KK
BBBBBBBBB      AA      AA  SSSSSSS  UUUUUUUUU  NN      NN  LLLLLLLLLL  000000  CCCCCCCC  KK      KK
BBBBBBBBB      AA      AA  SSSSSSS  UUUUUUUUU  NN      NN  LLLLLLLLLL  000000  CCCCCCCC  KK      KK

```

```

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
LLLLLLLLLL  IIIIII      SSSSSSSS
LLLLLLLLLL  IIIIII      SSSSSSSS

```

```

1 0001 0 MODULE BASSUNLOCK (          ! Basic UNLOCK construct
2 0002 0                               ! File: BASUNLOCK.B32
3 0003 0                               ) =
4 0004 1 BEGIN
5 0005 1
6 0006 1 *****
7 0007 1 *
8 0008 1 *  COPYRIGHT (c) 1978, 1980, 1982, 1984 BY *
9 0009 1 *  DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASSACHUSETTS. *
10 0010 1 *  ALL RIGHTS RESERVED. *
11 0011 1 *
12 0012 1 *  THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND COPIED *
13 0013 1 *  ONLY IN ACCORDANCE WITH THE TERMS OF SUCH LICENSE AND WITH THE *
14 0014 1 *  INCLUSION OF THE ABOVE COPYRIGHT NOTICE. THIS SOFTWARE OR ANY OTHER *
15 0015 1 *  COPIES THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY *
16 0016 1 *  OTHER PERSON. NO TITLE TO AND OWNERSHIP OF THE SOFTWARE IS HEREBY *
17 0017 1 *  TRANSFERRED. *
18 0018 1 *
19 0019 1 *  THE INFORMATION IN THIS SOFTWARE IS SUBJECT TO CHANGE WITHOUT NOTICE *
20 0020 1 *  AND SHOULD NOT BE CONSTRUED AS A COMMITMENT BY DIGITAL EQUIPMENT *
21 0021 1 *  CORPORATION. *
22 0022 1 *
23 0023 1 *  DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE OR RELIABILITY OF ITS *
24 0024 1 *  SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL. *
25 0025 1 *
26 0026 1 *
27 0027 1 *****
28 0028 1
29 0029 1
30 0030 1 ++
31 0031 1 FACILITY:
32 0032 1     Basic support library - user callable
33 0033 1
34 0034 1 ABSTRACT:
35 0035 1     This module is the UPI level of the Basic UNLOCK construct.
36 0036 1     This module will setup the I/O data base for the LUN and go directly to
37 0037 1     the REC level.
38 0038 1
39 0039 1
40 0040 1 ENVIRONMENT:
41 0041 1     User access mode - AST reentrant.
42 0042 1
43 0043 1 AUTHOR: Donald G. Petersen, CREATION DATE: 28-Feb-79
44 0044 1
45 0045 1 MODIFIED BY:
46 0046 1
47 0047 1     DGP, 28-Feb-79 : VERSION 01
48 0048 1     1-001 - original. DGP 28-Feb-79
49 0049 1     1-002 - Set up ISB$A_USER_FP. JBS 25-JUL-1979
50 0050 1 --
51 0051 1
52 0052 1 !<BLF/PAGE>
    
```

```

: 54      0053 1  |
: 55      0054 1  | SWITCHES:
: 56      0055 1  |
: 57      0056 1  |
: 58      0057 1  | SWITCHES ADDRESSING_MODE (EXTERNAL = GENERAL, NONEXTERNAL = WORD_RELATIVE);
: 59      0058 1  |
: 60      0059 1  |
: 61      0060 1  | LINKAGES
: 62      0061 1  |
: 63      0062 1  |
: 64      0063 1  | REQUIRE 'RTLIN:OTSLNK';           ! Define all linkages
: 65      0492 1  |
: 66      0493 1  |
: 67      0494 1  | TABLE OF CONTENTS:
: 68      0495 1  |
: 69      0496 1  |
: 70      0497 1  | FORWARD ROUTINE
: 71      0498 1  |     BASSUNLOCK : NOVALUE;       ! UPI level Sequential UNLOCK
: 72      0499 1  |
: 73      0500 1  |
: 74      0501 1  | INCLUDE FILES:
: 75      0502 1  |
: 76      0503 1  |
: 77      0504 1  | REQUIRE 'RTLML:OTISISB';       ! ISB definitions
: 78      0672 1  |
: 79      0673 1  | REQUIRE 'RTLML:OTSLUB';       ! LUB definitions
: 80      0813 1  |
: 81      0814 1  | REQUIRE 'RTLIN:RTLPSECT';     ! Define DECLARE_PSECTS macro
: 82      0909 1  |
: 83      0910 1  | LIBRARY 'RTLSTARLE';         ! Starlet system macros
: 84      0911 1  |
: 85      0912 1  |
: 86      0913 1  | MACROS:
: 87      0914 1  |
: 88      0915 1  |     NONE
: 89      0916 1  |
: 90      0917 1  | EQUATED SYMBOLS:
: 91      0918 1  |     NONE
: 92      0919 1  |
: 93      0920 1  |
: 94      0921 1  | PSECT DECLARATIONS:
: 95      0922 1  |
: 96      0923 1  | DECLARE_PSECTS (BAS);
: 97      0924 1  |
: 98      0925 1  | OWN STORAGE:
: 99      0926 1  |
: 100     0927 1  |     NONE
: 101     0928 1  |
: 102     0929 1  | EXTERNAL REFERENCES:
: 103     0930 1  |
: 104     0931 1  |
: 105     0932 1  | EXTERNAL ROUTINE
: 106     0933 1  |     BASS$REC_UNL : JSB_REC0 NOVALUE, ! REC level processing - RMS interface
: 107     0934 1  |                                     ! UNLOCK
: 108     0935 1  |     BASS$CB_PUSH : JSB_CB_PUSH NOVALUE, ! Load register CCB
: 109     0936 1  |     BASS$CB_POP  : JSB_CB_POP NOVALUE, ! Done with register CCB
: 110     0937 1  |     BASS$STOP_IO : NOVALOE;       ! Signal fatal BASIC I/O error

```

```
: 111      0938 1
: 112      0939 1 !+
: 113      0940 1 !- The following are the error codes used by this module.
: 114      0941 1 !-
: 115      0942 1
: 116      0943 1 EXTERNAL LITERAL
: 117      0944 1     BASSK_IO_CHANOT : UNSIGNED (8);
: 118      0945 1     ! I/O channel not open
```

```

: 120      0946 1 GLOBAL ROUTINE BASSUNLOCK (           : UNLOCK sequential
: 121      0947 1     UNIT                               : logical unit number
: 122      0948 1     ) : NOVALUE =
: 123      0949 1
: 124      0950 1
: 125      0951 1     ++
: 126      0952 1     FUNCTIONAL DESCRIPTION:
: 127      0953 1           This routine will set up the I/O data base for this LUN if necessary
: 128      0954 1           and then go directly to the REC level.  When control is returned to
: 129      0955 1           this routine, it pops the CCB off of the I/O system.  The actual inter-
: 130      0956 1           face to RMS is done at the REC level.  The current record is unlocked.
: 131      0957 1
: 132      0958 1     FORMAL PARAMETERS:
: 133      0959 1
: 134      0960 1           UNIT.rlu.v           logical unit number
: 135      0961 1
: 136      0962 1     IMPLICIT INPUTS:
: 137      0963 1
: 138      0964 1           NONE
: 139      0965 1
: 140      0966 1     IMPLICIT OUTPUTS:
: 141      0967 1
: 142      0968 1           ISBSB_STM_TYPE           the statement
: 143      0969 1
: 144      0970 1     COMPLETION CODES:
: 145      0971 1
: 146      0972 1           NONE
: 147      0973 1
: 148      0974 1     SIDE EFFECTS:
: 149      0975 1
: 150      0976 1           NONE
: 151      0977 1
: 152      0978 1     --
: 153      0979 1
: 154      0980 2     BEGIN
: 155      0981 2
: 156      0982 2     BUILTIN
: 157      0983 2     FP;
: 158      0984 2
: 159      0985 2     GLOBAL REGISTER
: 160      0986 2     CCB = K_CCB_REG : REF BLOCK [, BYTE];
: 161      0987 2
: 162      0988 2     LOCAL
: 163      0989 2     FMP : REF BLOCK [, BYTE];
: 164      0990 2
: 165      0991 2     FMP = .FP;
: 166      0992 2
: 167      0993 2     +
: 168      0994 2     Allocate the LUB/ISB/RAB for this unit if necessary.  Store new CB (con-
: 169      0995 2     trol block) in OTS$$A_CUR_LUB.  Store signed unit number in LUB$W_LUN.
: 170      0996 2     -
: 171      0997 2     BASS$CB_PUSH (.UNIT, LUB$K_ILUN_MIN);
: 172      0998 2     CCB [ISB$A_USER_FP] = .FMP-[SF$[SAVE_FP];
: 173      0999 2     +
: 174      1000 2     If the channel is not open, give an error message.
: 175      1001 2     -
: 176      1002 2     IF ( NOT .CCB [LUB$V_OPENED]) THEN BASS$STOP_IO (BASS$K_IO_CHANOT);

```

```

: 177      1003
: 178      1004
: 179      1005
: 180      1006
: 181      1007
: 182      1008
: 183      1009
: 184      1010
: 185      1011
: 186      1012
: 187      1013
: 188      1014

```

Now that the data base is in place, store the statement type and go directly to the REC level.

```

      CCB [ISBSB STTM_TYPE] = ISBSK_ST_TY_UNL;
      BASS$REC_UNL ();

```

Now that the UNLOCK has been done, pop the CCB off the I/O system.

```

      BASS$CB_POP ();
      END;

```

!End of BASSUNLOCK

```

.TITLE BASSUNLOCK
.IDENT \1-002\

.EXTRN BASS$REC_UNL, BASS$CB_PUSH
.EXTRN BASS$CB_POP, BASS$STOP_IO
.EXTRN BASSK_IO_CHANOT

.PSECT _BAS$CODE, NOWRT, SHR, PIC, 2

.ENTRY BASSUNLOCK, Save R2,R3,R4,R5,R11
MOVL  FP, FMP
MNEGL #8, R0
MOVL  UNIT, R2
JSB   BASS$CB_PUSH
MOVL  12(FMP), -180(CCB)
BLBS  -4(CCB), 1$
MOVZBL #BASSK_IO_CHANOT, -(SP)
CALLS #1, BASS$STOP_IO
MOVB  #42, -143(CCB)
JSB   BASS$REC_UNL
JSB   BASS$CB_POP
RET

```

```

      083C 00000
      53   5D  D0 00002
      50   08  CE 00005
      52   04  AC  D0 00008
      FF4C CB  00000000G 00 16 0000C
      0B   0C  A3  D0 00012
      7E   FC  AB  E8 00018
      00000000G 00 01  FB 00020
      FF71 CB  2A  90 00027 1$:
      00000000G 00 16 0002C
      00000000G 00 16 00032
      04 00038

```

```

: 0946
: 0991
: 0996
: 0997
: 1002
: 1008
: 1009
: 1013
: 1014

```

; Routine Size: 57 bytes, Routine Base: _BAS\$CODE + 0000

```

: 189      1015 1
: 190      1016 1 END
: 191      1017 1
: 192      1018 0 ELUDOM

```

! End of module BASSUNLOCK

PSECT SUMMARY

Name	Bytes	Attributes
_BAS\$CODE	57	NOVEC, NOWRT, RD, EXE, SHR, LCL, REL, CON, PIC, ALIGN(2)

Library Statistics

File	----- Total	Symbols Loaded	----- Percent	Pages Mapped	Processing Time
:_\$255\$DUA28:[SYSLIB]STARLET.L32;1	9776	1	0	581	00:01.2

COMMAND QUALIFIERS

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

: Size: 57 code + 0 data bytes
: Run Time: 00:08.0
: Elapsed Time: 00:17.2
: Lines/CPU Min: 7597
: Lexemes/CPU-Min: 45350
: Memory Used: 108 pages
: Compilation Complete

0033 AH-BT13A-SE
VAX/VMS V4.0

DIGITAL EQUIPMENT CORPORATION
CONFIDENTIAL AND PROPRIETARY

The image displays a grid of 100 terminal windows, arranged in 10 rows and 10 columns. Each window shows a different view of the VAX/VMS operating system, including system utilities, data tables, and diagnostic screens. Several windows are prominently labeled with titles:

- Row 1, Column 8: **BASVIRTUA LIS**
- Row 2, Column 1: **BASUDFWL LIS**
- Row 3, Column 3: **BASUNLOCK LIS**
- Row 3, Column 5: **BASVECTOR LIS**
- Row 5, Column 4: **BASVAL LIS**
- Row 5, Column 6: **BASVRTIO LIS**
- Row 7, Column 2: **BASUNWIND LIS**
- Row 7, Column 3: **BASUPDATE LIS**
- Row 8, Column 5: **BASVECTR2 LIS**

The background of the grid is a dark, textured pattern, and the overall appearance is that of a high-density terminal display from the early 1980s.