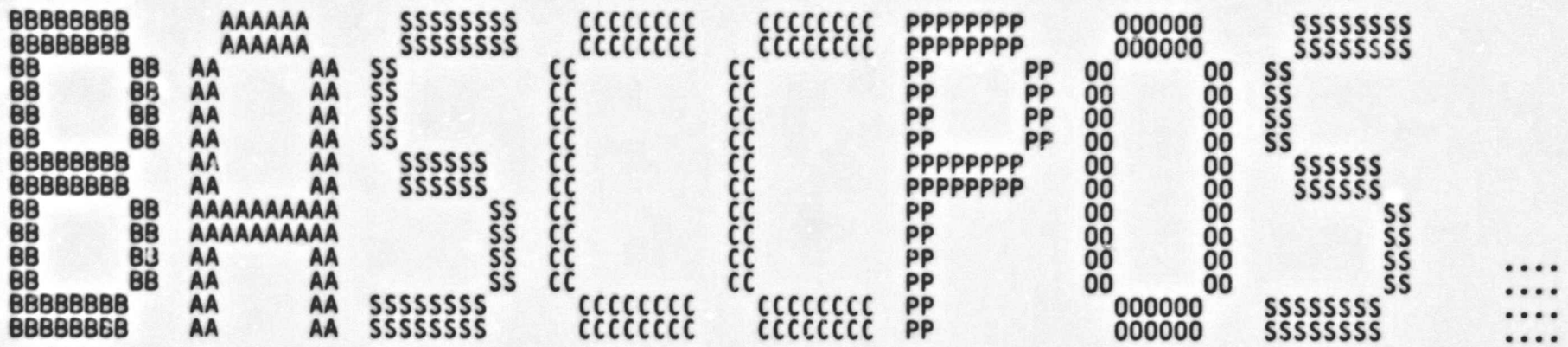
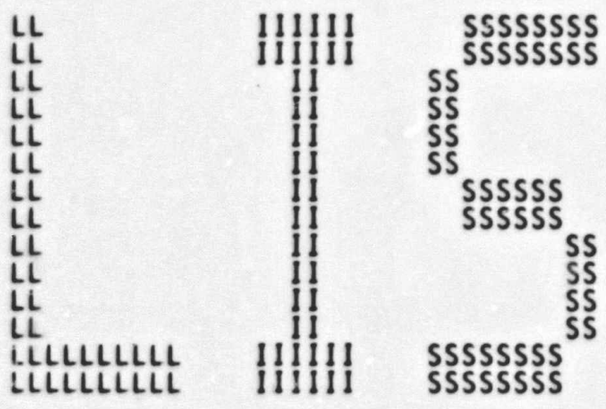
```
BBBBBBBBBBBB        AAAAAAAAA       SSSSSSSSSSS  RRRRRRRRRRR     TTTTTTTTTTTTTTT  LLL
BBBBBBBBBBBB        AAAAAAAAA       SSSSSSSSSSS  RRRRRRRRRRR     TTTTTTTTTTTTTTT  LLL
BBBBBBBBBBBB        AAAAAAAAA       SSSSSSSSSSS  RRRRRRRRRRR     TTTTTTTTTTTTTTT  LLL
BBB          BBB   AAA         AAA  SSS          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     SSSSSSSSS    RRRRRRRRRRR       TTT        LLL
BBBBBBBBBBBB       AAA         AAA     SSSSSSSSS    RRRRRRRRRRR       TTT        LLL
BBBBBBBBBBBB       AAA         AAA     SSSSSSSSS    RRRRRRRRRRR       TTT        LLL
BBB          BBB   AAAAAAAAAAAAAAA           SSS  RRR   RRR         TTT        LLL
BBB          BBB   AAAAAAAAAAAAAAA           SSS  RRR   RRR         TTT        LLL
BBB          BBB   AAAAAAAAAAAAAAA           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  SSSSSSSSSSS  RRR         RRR    TTT        LLLLLLLLLLLLLLL
BBBBBBBBBBBB       AAA         AAA  SSSSSSSSSSS  RRR         RRR    TTT        LLLLLLLLLLLLLLL
BBBBBBBBBBBB       AAA         AAA  SSSSSSSSSSS  RRR         RRR    TTT        LLLLLLLLLLLLLLL
```


BASIC POS


LISP

```
;     1          0001  0 MODULE BAS$CCPOS (                          ! BASIC Current Cursor Position
;     2          0002  0                 IDENT = '1-009'             ! File: BASCCPOS.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 ! FACILITY: BASIC Support Library - user callable
;    31          0031  1 !
;    32          0032  1 ! ABSTRACT:
;    33          0033  1 !
;    34          0034  1 !       This module supports the BASIC function CCPOS.  It returns the current
;    35          0035  1 !       cursor position on the given channel number.  For the time being it
;    36          0036  1 !       returns the value in LUB$B_PRINT_POS.  Eventually it is hoped that the
;    37          0037  1 !       terminal driver can be interrogated directly for the cursor position.
;    38          0038  1 !
;    39          0039  1 ! ENVIRONMENT: User access mode; mixture of AST level or not
;    40          0040  1 !
;    41          0041  1 ! AUTHOR: Donald G. Petersen, CREATION DATE: 29-Nov-78
;    42          0042  1 !
;    43          0043  1 ! MODIFIED BY:
;    44          0044  1 !
;    45          0045  1 ! 1-01  - Original.  DGP 29-Nov-78
;    46          0046  1 ! 1-002 - Use OPEN$K_LUN_BPRI in call to CB_PUSH.  DGP 05-Dec-78
;    47          0047  1 ! 1-003 - Change REQUIRE file names from FOR... to OTS...  JBS 06-DEC-78
;    48          0048  1 ! 1-004 - Change OPEN prefix to LUB.  JBS 13-DEC-78
;    49          0049  1 ! 1-005 - Change FOR$$_B... to BAS$$CB... JBS 02-JAN-1979
;    50          0050  1 ! 1-006 - Change PRINT_POS to longword.  DGP 19-Mar-79
;    51          0051  1 ! 1-007 - Allow CB_PUSH to reach back to the PRINT LUB.  JBS 01-MAY-1979
;    52          0052  1 ! 1-008 - Call STOP_IO if the channel is not open.  JBS 01-MAY-1979
;    53          0053  1 ! 1-009 - Set up ISB$A_USER_FP.  JBS 25-JUL-1979
;    54          0054  1 !--
;    55          0055  1
;    56          0056  1 !<BLF/PAGE>
```

```
;    58          0057  1
;    59          0058  1 SWITCHES ADDRESSING_MODE (EXTERNAL = GENERAL, NONEXTERNAL = WORD_RELATIVE);
;    60          0059  1
;    61          0060  1 !
;    62          0061  1 !       LINKAGES
;    63          0062  1 !
;    64          0063  1
;    65          0064  1 REQUIRE 'RTLIN:OTSLNK';                         ! Initialize all linkages
;    66          0493  1
;    67          0494  1 !
;    68          0495  1 ! TABLE OF CONTENTS:
;    69          0496  1 !
;    70          0497  1
;    71          0498  1 FORWARD ROUTINE
;    72          0499  1     BAS$CCPOS;                                  ! Current Cursor Position
;    73          0500  1
;    74          0501  1 !
;    75          0502  1 ! INCLUDE FILES:
;    76          0503  1 !
;    77          0504  1
;    78          0505  1 REQUIRE 'RTLIN:RTLPSECT';                       ! Psect definitions
;    79          0600  1
;    80          0601  1 REQUIRE 'RTLML:OTSLUB';                         ! needed for LUB$L_PRINT_POS
;    81          0741  1
;    82          0742  1 REQUIRE 'RTLML:OTSISB';                         ! needed for ISB$A_USER_FP
;    83          0910  1
;    84          0911  1 LIBRARY 'RTLSTARLE';                            ! STARLET library for macros and symbols
;    85          0912  1
;    86          0913  1 !
;    87          0914  1 ! MACROS:
;    88          0915  1 !
;    89          0916  1 !       NONE
;    90          0917  1 !
;    91          0918  1 !
;    92          0919  1 ! EQUATED SYMBOLS:
;    93          0920  1 !
;    94          0921  1 !       NONE
;    95          0922  1 !
;    96          0923  1 ! PSECT DECLARATIONS:
;    97          0924  1 !
;    98          0925  1 DECLARE_PSECTS (BAS);                           ! declare PSECTs for BAS$ facility
;    99          0926  1 !
;   100          0927  1 ! OWN STORAGE:
;   101          0928  1 !
;   102          0929  1 !       NONE
;   103          0930  1 !
;   104          0931  1 !
;   105          0932  1 ! EXTERNAL REFERENCES:
;   106          0933  1 !
;   107          0934  1 !
;   108          0935  1
;   109          0936  1 EXTERNAL ROUTINE
;   110          0937  1     BAS$$CB_PUSH : JSB_CB_PUSH,                 ! Push control block
;   111          0938  1     BAS$$CB_POP : JSB_CB_POP NOVALUE,           ! Pop control block
;   112          0939  1     BAS$$STOP_IO : NOVALUE;                     ! Signal fatal I/O errors
;   113          0940  1
;   114          0941  1 EXTERNAL LITERAL
```

```
;  115          0942  1     BAS$K_IO_CHANOT : UNSIGNED (8);          ! I/O Channel not open
;  116          0943  1
```

```
;  118          0944  1 GLOBAL ROUTINE BAS$CCPOS (                    ! Get cursor position
;  119          0945  1         UNIT                                   ! Channel number for which cursor position is desired
;  120          0946  1     ) : =
;  121          0947  1
;  122          0948  1 !++
;  123          0949  1 ! FUNCTIONAL DESCRIPTION:
;  124          0950  1 !
;  125          0951  1 !       Return the current cursor position of the indicated channel.
;  126          0952  1 !       Channel 0 is split into 2 data bases, input and output.  If channel 0
;  127          0953  1 !       is specified, then the cursor position for the output data base will be
;  128          0954  1 !       returned.  This problem will go away if and when the terminal driver supplies
;  129          0955  1 !       the cursor position for files opened on a terminal.
;  130          0956  1 !
;  131          0957  1 ! FORMAL PARAMETERS:
;  132          0958  1 !
;  133          0959  1 !       UNIT.rl.v               cursor position of this unit is returned
;  134          0960  1 !
;  135          0961  1 ! IMPLICIT INPUTS:
;  136          0962  1 !
;  137          0963  1 !       LUB$L_PRINT_POS         Current cursor position for a channel
;  138          0964  1 !
;  139          0965  1 ! IMPLICIT OUTPUTS:
;  140          0966  1 !
;  141          0967  1 !       NONE
;  142          0968  1 !
;  143          0969  1 ! ROUTINE VALUE:
;  144          0970  1 !
;  145          0971  1 !       CURSOR_POS.wl.v         Current cursor position of indicated channel
;  146          0972  1 !
;  147          0973  1 ! SIDE EFFECTS:
;  148          0974  1 !
;  149          0975  1 !       Errors are signalled if a channel number less than K_DLUN_MIN or
;  150          0976  1 !       greater than 99 by BAS$$CB_PUSH. If the channel is not 0 and
;  151          0977  1 !       it has not been opened, this routine signals.
;  152          0978  1 !
;  153          0979  1 !--
;  154          0980  1
;  155          0981  2     BEGIN
;  156          0982  2
;  157          0983  2     BUILTIN
;  158          0984  2         FP;
;  159          0985  2
;  160          0986  2     GLOBAL REGISTER
;  161          0987  2         CCB = K_CCB_REG : REF BLOCK [, BYTE];
;  162          0988  2
;  163          0989  2     MAP
;  164          0990  2         UNIT : REF BLOCK;
;  165          0991  2
;  166          0992  2     LOCAL
;  167          0993  2         FMP : REF BLOCK [, BYTE],
;  168          0994  2         CCPOS;                                 ! holds cursor position
;  169          0995  2
;  170          0996  2     FMP = .FP;
;  171          0997  2 !+
;  172          0998  2 ! If unit 0 was requested then change the unit number to indicate the
;  173          0999  2 ! PRINT logical unit.  Call CB_PUSH to get a pointer to the data base.  If
;  174          1000  2 ! the unit number is not in the valid range of unit numbers, CB_PUSH will
```

```
;  175          1001  2 ! signal an error.  Check to see if the channel has been opened and if the
;  176          1002  2 ! logical unit is not 0 then signal.
;  177          1003  2 !-
;  178          1004  2     BAS$$CB_PUSH ((IF .UNIT EQL 0 THEN LUB$K_LUN_BPRI ELSE .UNIT), LUB$K_ILUN_MIN);
;  179          1005  2     CCB [ISB$A_USER_FP] = .FMP [SF$L_SAVE_FP];
;  180          1006  2
;  181          1007  3     IF (( NOT .CCB [LUB$V_OPENED]) AND (.UNIT GTR 0))
;  182          1008  2     THEN
;  183          1009  2 !+
;  184          1010  2 ! The unit is not 0 and it is not opened.  Pop the control blocks back
;  185          1011  2 ! and signal the error.
;  186          1012  2 !-
;  187          1013  2         BAS$$STOP_IO (BAS$K_IO_CHANOT)
;  188          1014  2     ELSE
;  189          1015  2 !+
;  190          1016  2 ! Everything is valid.  Pick up the cursor position from the channel
;  191          1017  2 ! data base
;  192          1018  2 !-
;  193          1019  2         CCPOS = .CCB [LUB$L_PRINT_POS];
;  194          1020  2
;  195          1021  2 !+
;  196          1022  2 ! Pop the control block back and return the cursor position.
;  197          1023  2 !-
;  198          1024  2     BAS$$CB_POP ();
;  199          1025  2     RETURN .CCPOS
;  200          1026  1     END;                                    ! End of routine BAS$CCPOS

                                                        .TITLE  BAS$CCPOS
                                                        .IDENT  \1-009\

                                                        .EXTRN  BAS$$CB_PUSH, BAS$$CB_POP
                                                        .EXTRN  BAS$$STOP_IO, BAS$K_IO_CHANOT

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

                                    081C 00000          .ENTRY  BAS$CCPOS, Save R2,R3,R4,R11            ; 0944
                            53      5D   D0 00002       MOVL    FP, FMP                                 ; 0996
                            54  04  AC   D0 00005       MOVL    UNIT, R4                                ; 1004
                                    05   12 00009       BNEQ    1$                                      ;
                            52      08   CE 0000B       MNEGL   #8, R2                                  ;
                                    03   11 0000E       BRB     2$                                      ;
                            52      54   D0 00010 1$:   MOVL    R4, R2                                  ;
                            50      08   CE 00013 2$:   MNEGL   #8, R0                                  ;
                        00000000G   00   16 00016       JSB     BAS$$CB_PUSH                            ;
                    FF4C    CB  0C  A3   D0 0001C       MOVL    12(FMP), -180(CCB)                      ; 1005
                            11  FC  AB   E8 00022       BLBS    -4(CCB), 3$                             ; 1007
                                    54   D5 00026       TSTL    R4                                      ;
                                    0D   15 00028       BLEQ    3$                                      ;
                            7E      00G  8F   9A 0002A  MOVZBL  #BAS$K_IO_CHANOT, -(SP)                 ; 1013
                00000000G   00          01   FB 0002E  CALLS   #1, BAS$$STOP_IO                        ;
                                    04   11 00035       BRB     4$                                      ;
                            52  C8  AB   D0 00037 3$:   MOVL    -56(CCB), CCPOS                         ; 1019
                        00000000G   00   16 0003B 4$:   JSB     BAS$$CB_POP                             ; 1024
                            50          52   D0 00041   MOVL    CCPOS, R0                               ; 1025
                                        04 00044        RET                                             ; 1026
```

```
; Routine Size:  69 bytes,    Routine Base:  _BAS$CODE + 0000

;  201          1027  1
;  202          1028  1 END                                    !End of module BAS$CCPOS
;  203          1029  1
;  204          1030  0 ELUDOM



;                                PSECT SUMMARY
;
;      Name                      Bytes                         Attributes
;
;  _BAS$CODE                       69  NOVEC,NOWRT,  RD ,  EXE,  SHR,  LCL,  REL,  CON,  PIC,ALIGN(2)


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



;                                      COMMAND QUALIFIERS

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

; Size:          69 code + 0 data bytes
; Run Time:          00:08.3
; Elapsed Time:      00:20.3
; Lines/CPU Min:      7454
; Lexemes/CPU-Min:  44156
; Memory Used:   112 pages
; Compilation Complete
```

BASCLOSE
LIS

BASCONCAT
LIS

BASCTRLO
LIS

BASCHANGE
LIS

BASCTRLC
LIS

BASCHAIN
LIS

BASCOPYFD
LIS

BASCHR
LIS

BASCMPAPP
LIS

BASCUTOUT
LIS

BASCCPOS
LIS