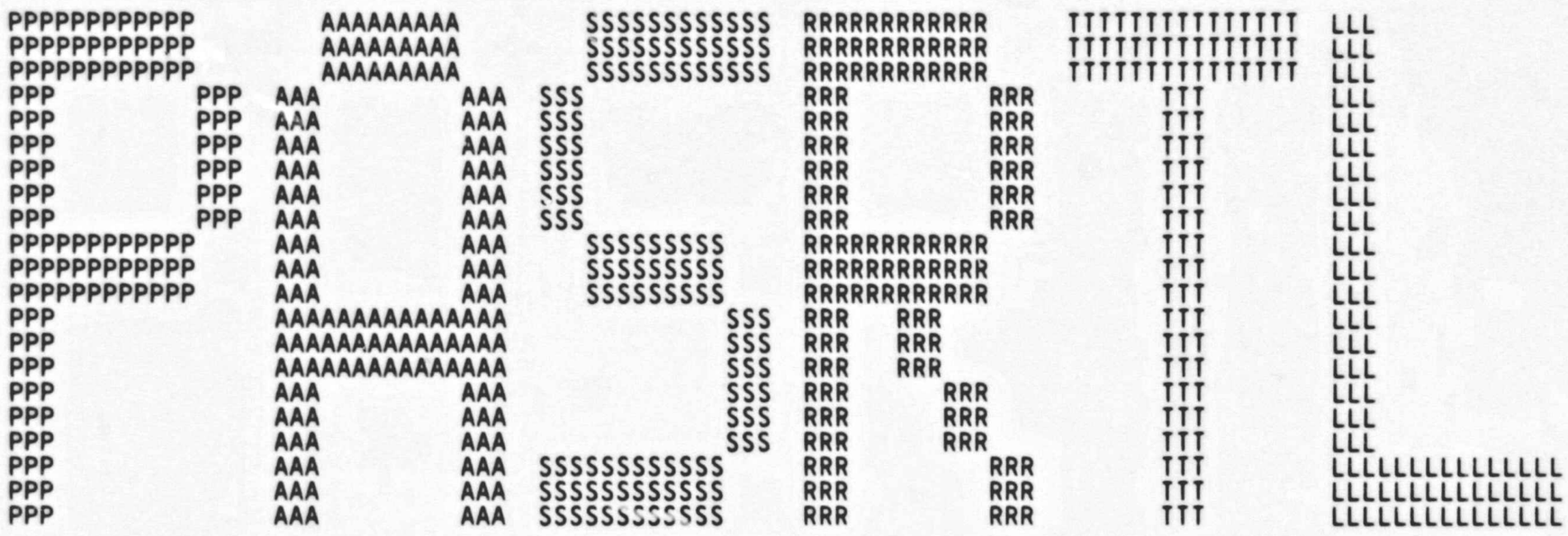
PPPPPPPPPPPP AAAAAAAAA SSSSSSSSSSS RRRRRRRRRRR TTTTTTTTTTTTTTT LLL
PPP PPP AAA AAA SSS RRR RRR TTT LLL
PPPPPPPPPPPP AAAAAAAAAAAAAAA SSS RRR RRR TTT LLL
PPP AAA AAA SSSSSSSSS RRRRRRRRRRR TTT LLL
PPP AAAAAAAAAAAAAAA SSS RRR RRR TTT LLL
PPP AAA AAA SSS RRR RRR TTT LLL
PPP AAA AAA SSSSSSSSSSS RRR RRR TTT LLLLLLLLLLLLLLL

FILEID**PASVM

```
PPPPPPPP      AAAAAA      SSSSSSSS  VV      VV  MM      MM
PPPPPPPP      AAAAAA      SSSSSSSS  VV      VV  MM      MM
PP      PP  AA      AA  SS          VV      VV  MMMM  MMMM
PP      PP  AA      AA  SS          VV      VV  MMMM  MMMM
PP      PP  AA      AA  SS          VV      VV  MM  MM  MM
PP      PP  AA      AA  SS          VV      VV  MM  MM  MM
PPPPPPPP    AA      AA    SSSSSS    VV      VV  MM      MM
PPPPPPPP    AA      AA    SSSSSS    VV      VV  MM      MM
PP          AAAAAAAAAA          SS  VV      VV  MM      MM
PP          AAAAAAAAAA          SS  VV      VV  MM      MM
PP          AA      AA          SS    VV  VV    MM      MM      ....
PP          AA      AA          SS    VV  VV    MM      MM      ....
PP          AA      AA  SSSSSSSS        VV      MM      MM      ....
PP          AA      AA  SSSSSSSS        VV      MM      MM      ....

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 PAS$$VM ( %TITLE 'Allocate/deallocate virtual memory'
;     2          0002  0                  IDENT = '1-001'                    ! File: PASVM.B32 Edit: SBL1001
;     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:     Pascal Language Support
;    32          0032  1 !
;    33          0033  1 ! ABSTRACT:
;    34          0034  1 !
;    35          0035  1 !       This module contains procedures to allocate and deallocate
;    36          0036  1 !       virtual memory.
;    37          0037  1 !
;    38          0038  1 ! ENVIRONMENT:  User mode - AST reentrant
;    39          0039  1 !
;    40          0040  1 ! AUTHOR: Steven B. Lionel, CREATION DATE: 1-April-1981
;    41          0041  1 !
;    42          0042  1 ! MODIFIED BY:
;    43          0043  1 !
;    44          0044  1 ! 1-001 - Original.  SBL 1-April-1981
;    45          0045  1 !--
;    46          0046  1
```

```
;    48     0047  1 %SBTTL 'Declarations'
;    49     0048  1 !
;    50     0049  1 ! PROLOGUE DEFINITIONS:
;    51     0050  1 !
;    52     0051  1
;    53     0052  1 REQUIRE 'RTLIN:PASPROLOG';                  ! Linkages, externals, PSECTs, structures
;    54     0116  1
;    55     0117  1 !
;    56     0118  1 ! TABLE OF CONTENTS:
;    57     0119  1 !
;    58     0120  1
;    59     0121  1 FORWARD ROUTINE
;    60     0122  1     PAS$$GET_VM: CALL_VM,                  ! Allocate virtual memory
;    61     0123  1     PAS$$FREE_VM: NOVALUE;                 ! Deallocate virtual memory
;    62     0124  1
;    63     0125  1 !
;    64     0126  1 ! MACROS:
;    65     0127  1 !
;    66     0128  1 !       NONE
;    67     0129  1 !
;    68     0130  1 ! EQUATED SYMBOLS:
;    69     0131  1 !
;    70     0132  1 !       NONE
;    71     0133  1 !
;    72     0134  1 ! FIELDS:
;    73     0135  1 !
;    74     0136  1 !       NONE
;    75     0137  1 !
;    76     0138  1 ! OWN STORAGE:
;    77     0139  1 !
;    78     0140  1 !    NONE
```

```
;    80          0141  1 %SBTTL 'PAS$$GET_VM - Allocate virtual memory'
;    81          0142  1 GLOBAL ROUTINE PAS$$GET_VM (
;    82          0143  1         PFV: REF $PAS$PFV_FILE_VARIABLE,        ! File variable
;    83          0144  1         LENGTH                                  ! Length to allocate
;    84          0145  1     ): CALL_VM =
;    85          0146  1
;    86          0147  1 !++
;    87          0148  1 ! FUNCTIONAL DESCRIPTION:
;    88          0149  1 !
;    89          0150  1 !       This procedure allocates a block of virtual memory and returns
;    90          0151  1 !       a pointer to the block as its function value.  The memory allocated
;    91          0152  1 !       is zero-filled.
;    92          0153  1 !
;    93          0154  1 ! CALLING SEQUENCE:
;    94          0155  1 !
;    95          0156  1 !       block_addr.wa.v = PAS$$GET_VM (PFV.rr.r, LENGTH.rl.v)
;    96          0157  1 !
;    97          0158  1 ! FORMAL PARAMETERS:
;    98          0159  1 !
;    99          0160  1 !       PFV             - The Pascal File Variable (PFV) passed by reference.
;   100          0161  1 !                         The structure of the PFV is defined in PASPFV.REQ.
;   101          0162  1 !
;   102          0163  1 !       LENGTH          - The length in bytes of the block to allocate.
;   103          0164  1 !                         Must not be greater than 65535.
;   104          0165  1 !
;   105          0166  1 ! IMPLICIT INPUTS:
;   106          0167  1 !
;   107          0168  1 !       NONE
;   108          0169  1 !
;   109          0170  1 ! IMPLICIT OUTPUTS:
;   110          0171  1 !
;   111          0172  1 !       NONE
;   112          0173  1 !
;   113          0174  1 ! ROUTINE VALUE:
;   114          0175  1 !
;   115          0176  1 !       A pointer to the allocated block.
;   116          0177  1 !
;   117          0178  1 ! SIDE EFFECTS:
;   118          0179  1 !
;   119          0180  1 !       Allocates virtual memory.
;   120          0181  1 !
;   121          0182  1 ! SIGNALLED ERRORS:
;   122          0183  1 !
;   123          0184  1 !       INSVIRMEM - Insufficient virtual memory
;   124          0185  1 !
;   125          0186  1 !--
;   126          0187  1
;   127          0188  2     BEGIN
;   128          0189  2
;   129          0190  2     LOCAL
;   130          0191  2         BLOCK_ADDR;                         ! Address of allocated block
;   131          0192  2
;   132          0193  2     IF .LENGTH GTRU 65535
;   133          0194  2     THEN
;   134          0195  2         $PAS$BUGCHECK (BUG_BADVMSIZE);
;   135          0196  2
;   136          0197  2     IF NOT LIB$GET_VM (LENGTH, BLOCK_ADDR)
```

```
;  137          0198  2      THEN
;  138          0199  2          $PAS$IO_ERROR (PAS$_INSVIRMEM,0);
;  139          0200  2
;  140          0201  2      CH$FILL (0, .LENGTH, .BLOCK_ADDR);  ! Zero-fill
;  141          0202  2
;  142          0203  2      RETURN .BLOCK_ADDR;
;  143          0204  2
;  144          0205  1      END;                                    ! End of routine PAS$$GET_VM


                                                                     .TITLE  PAS$$VM Allocate/deallocate virtual memory
                                                                     .IDENT  \1-001\

                                                                     .EXTRN  PAS$$GET_VM, PAS$$FREE_VM
                                                                     .EXTRN  PAS$$BUGCHECK, LIB$GET_VM
                                                                     .EXTRN  PAS$$SIGNAL, PAS$K_INSVIRMEM

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

                                    003C 00000                       .ENTRY  PAS$$GET_VM, Save R2,R3,R4,R5                ; 0142
                          5E        04   C2 00002                     SUBL2   #4, SP
                0000FFFF  8F   04   AC   D1 00005                     CMPL    LENGTH, #65535                               ; 0193
                                    0B   1B 00008                     BLEQU   1$
                                    04   DD 0000A                     PUSHL   #4                                           ; 0195
                00000000G 00        01   FB 0000B                     CALLS   #1, PAS$$BUGCHECK
                                    2A   11 00012                     BRB     3$
                                    5E   DD 00014           1$:       PUSHL   SP                                           ; 0197
                              04    AC   9F 00016                     PUSHAB  LENGTH
                00000000G 00        02   FB 00019                     CALLS   #2, LIB$GET_VM
                          0F        50   E8 0001C                     BLBS    R0, 2$
                                    7E   D4 0001F                     CLRL    -(SP)                                        ; 0199
                          7E   00G  8F   9A 00021                     MOVZBL  #PAS$K_INSVIRMEM, -(SP)
                     00000000G 00        02   FB 00025                CALLS   #2, PAS$$SIGNAL
                                    0C   11 0002B                     BRB     3$
     04   AC             00        6E   00   2C 0002D           2$:   MOVC5   #0, (SP), #0, LENGTH, @BLOCK_ADDR           ; 0201
                                                00   BE 00033
                                    50   6E   D0 00036                MOVL    BLOCK_ADDR, R0                               ; 0203
                                              04 00039                RET
                                    50   D4 0003A           3$:       CLRL    R0                                           ; 0205
                                              04 0003C                RET

; Routine Size:  71 bytes,    Routine Base:  _PAS$CODE + 0000

;  145          0206  1
;  146          0207  1 !<BLF/PAGE>
```

```
;  148     0208  1 %SBTTL 'PAS$$FREE_VM - Deallocate virtual memory'
;  149     0209  1 GLOBAL ROUTINE PAS$$FREE_VM (
;  150     0210  1         LENGTH,                                 ! Length to allocate
;  151     0211  1         BLOCK_ADDR: REF VECTOR [, LONG]         ! Address of block to deallocate
;  152     0212  1     ): NOVALUE =
;  153     0213  1
;  154     0214  1 !++
;  155     0215  1 ! FUNCTIONAL DESCRIPTION:
;  156     0216  1 !
;  157     0217  1 !       This procedure deallocates a block of virtual memory.
;  158     0218  1 !
;  159     0219  1 ! CALLING SEQUENCE:
;  160     0220  1 !
;  161     0221  1 !       CALL PAS$$FREE_VM (LENGTH.rl.v, BLOCK_ADDR.ma.r)
;  162     0222  1 !
;  163     0223  1 ! FORMAL PARAMETERS:
;  164     0224  1 !
;  165     0225  1 !       LENGTH          - The length in bytes of the block to deallocate.
;  166     0226  1 !
;  167     0227  1 !       BLOCK_ADDR      - The address of the block to deallocate
;  168     0228  1 !
;  169     0229  1 ! IMPLICIT INPUTS:
;  170     0230  1 !
;  171     0231  1 !       NONE
;  172     0232  1 !
;  173     0233  1 ! IMPLICIT OUTPUTS:
;  174     0234  1 !
;  175     0235  1 !       NONE
;  176     0236  1 !
;  177     0237  1 ! ROUTINE VALUE:
;  178     0238  1 !
;  179     0239  1 !       NONE
;  180     0240  1 !
;  181     0241  1 ! SIDE EFFECTS:
;  182     0242  1 !
;  183     0243  1 !       Deallocates virtual memory.
;  184     0244  1 !
;  185     0245  1 ! SIGNALLED ERRORS:
;  186     0246  1 !
;  187     0247  1 !       BUGCHECK
;  188     0248  1 !
;  189     0249  1 !--
;  190     0250  1
;  191     0251  2     BEGIN
;  192     0252  2
;  193     0253  2     IF NOT LIB$FREE_VM (LENGTH, BLOCK_ADDR [0])
;  194     0254  2     THEN
;  195     0255  2         $PAS$BUGCHECK (BUG_FREEVMFAIL);
;  196     0256  2
;  197     0257  2     BLOCK_ADDR [0] = 0;         ! Invalidate pointer
;  198     0258  2
;  199     0259  2     RETURN;
;  200     0260  2
;  201     0261  1     END;                                        ! End of routine PAS$$FREE_VM
```

```
                                                                .EXTRN  LIB$FREE_VM

                                                    0000 00000  .ENTRY  PAS$$FREE_VM, Save nothing                ; 0209
                                                08  AC   DD 00002  PUSHL   BLOCK_ADDR                           ; 0253
                                                04  AC   9F 00005  PUSHAB  LENGTH                               ;
                                 00000000G  00      02   FB 00008  CALLS   #2, LIB$FREE_VM                      ;
                                            0A      50   E8 0000F  BLBS    R0, 1$                              ;
                                                    02   DD 00012  PUSHL   #2                                   ; 0255
                                 00000000G  00      01   FB 00014  CALLS   #1, PAS$$BUGCHECK                    ;
                                                    04 0001B      RET                                           ;
                                                08  BC   D4 0001C 1$:    CLRL    @BLOCK_ADDR                   ; 0257
                                                    04 0001F      RET                                           ; 0261

; Routine Size:  32 bytes,    Routine Base:  _PAS$CODE + 0047

;  202          0262  1
;  203          0263  1 !<BLF/PAGE>
```

```
;  205          0264  1 END                                   ! End of module PAS$$VM
;  206          0265  1
;  207          0266  0 ELUDOM
```

```
;                                  PSECT SUMMARY
;
;       Name                    Bytes                           Attributes
;
;  _PAS$CODE                        103  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         0         0         581        00:01.0
;  _$255$DUA28:[PASRTL.OBJ]PASLIB.L32;1      427        33         7          33        00:00.4
```

```
;                                  COMMAND QUALIFIERS

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

; Size:         103 code + 0 data bytes
; Run Time:       00:04.4
; Elapsed Time:   00:16.9
; Lines/CPU Min:   3668
; Lexemes/CPU-Min:  7586
; Memory Used:  47 pages
; Compilation Complete
```

PASREWRIT LIS

PASSIGNAL LIS

PASSOR LIS

PASTRUNCA LIS

PASUNLOCK LIS

PASWRIBOO LIS

PASVECTOR LIS

PASWRIBIN LIS

PASUNDEFI LIS

PASVALIDA LIS

PASWRIHEX LIS

PASVM LIS

PASWRIFFI LIS

PASTIME LIS

PASWRIFDI LIS

PASUPDATE LIS

PASWRIENU LIS

PASSTATUS LIS

PASUFB LIS

PASWRICHA LIS