G 6

```
PPPPPPPPPPPP        AAAAAAAAA        SSSSSSSSSSSS  RRRRRRRRRRRR     TTTTTTTTTTTTTTT  LLL
PPPPPPPPPPPP        AAAAAAAAA        SSSSSSSSSSSS  RRRRRRRRRRRR     TTTTTTTTTTTTTTT  LLL
PPPPPPPPPPPP        AAAAAAAAA        SSSSSSSSSSSS  RRRRRRRRRRRR     TTTTTTTTTTTTTTT  LLL
PPP         PPP  AAA         AAA  SSS              RRR         RRR        TTT        LLL
PPP         PPP  AAA         AAA  SSS              RRR         RRR        TTT        LLL
PPP         PPP  AAA         AAA  SSS              RRR         RRR        TTT        LLL
PPP         PPP  AAA         AAA  SSS              RRR         RRR        TTT        LLL
PPP         PPP  AAA         AAA  SSS              RRR         RRR        TTT        LLL
PPP         PPP  AAA         AAA  SSS              RRR         RRR        TTT        LLL
PPPPPPPPPPPP     AAA         AAA     SSSSSSSSS     RRRRRRRRRRRR           TTT        LLL
PPPPPPPPPPPP     AAA         AAA     SSSSSSSSS     RRRRRRRRRRRR           TTT        LLL
PPPPPPPPPPPP     AAA         AAA     SSSSSSSSS     RRRRRRRRRRRR           TTT        LLL
PPP              AAAAAAAAAAAAAAA              SSS  RRR   RRR              TTT        LLL
PPP              AAAAAAAAAAAAAAA              SSS  RRR   RRR              TTT        LLL
PPP              AAAAAAAAAAAAAAA              SSS  RRR   RRR              TTT        LLL
PPP              AAA         AAA              SSS  RRR      RRR           TTT        LLL
PPP              AAA         AAA              SSS  RRR      RRR           TTT        LLL
PPP              AAA         AAA              SSS  RRR      RRR           TTT        LLL
PPP              AAA         AAA  SSSSSSSSSSSS     RRR         RRR        TTT        LLLLLLLLLLLLLLL
PPP              AAA         AAA  SSSSSSSSSSSS     RRR         RRR        TTT        LLLLLLLLLLLLLLL
PPP              AAA         AAA  SSSSSSSSSSSS     RRR         RRR        TTT        LLLLLLLLLLLLLLL
```

```
_$2

Sym
---

PAS
PAS
PAS
PAS
PAS
PAS
PAS
PAS
PAS
PAS
PAS
PAS
PAS
PAS
PAS
PAS
PAS
PAS
PAS
PAS
PAS
PAS
PAS
PAS
PAS
PAS
PAS
PAS

PAS

PAS

PAS

PAS
PAS
PAS
PAS
PAS

PAS
PAS
PAS
PAS
PAS
PAS
PAS
PAS
```

J 10

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
```

PA
1-(

```
0 MODULE PAS$$VM ( %TITLE 'Allocate/deallocate virtual memory'
0                IDENT = '1-001'                 ! File: PASVM.B32 Edit: SBL1001
0                ) =
1 BEGIN
1 !
1 !****************************************************************************
1 !*                                                                          *
1 !*  COPYRIGHT (c) 1978, 1980, 1982, 1984 BY                                 *
1 !*  DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASSACHUSETTS.                  *
1 !*  ALL RIGHTS RESERVED.                                                    *
1 !*                                                                          *
1 !*  THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND COPIED   *
1 !*  ONLY IN  ACCORDANCE WITH  THE  TERMS  OF  SUCH  LICENSE  AND WITH THE   *
1 !*  INCLUSION OF THE ABOVE COPYRIGHT NOTICE. THIS SOFTWARE OR  ANY  OTHER   *
1 !*  COPIES THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY   *
1 !*  OTHER PERSON.  NO TITLE TO AND OWNERSHIP OF  THE  SOFTWARE IS  HEREBY   *
1 !*  TRANSFERRED.                                                            *
1 !*                                                                          *
1 !*  THE INFORMATION IN THIS SOFTWARE IS  SUBJECT TO CHANGE WITHOUT NOTICE   *
1 !*  AND  SHOULD  NOT  BE  CONSTRUED AS  A COMMITMENT BY DIGITAL EQUIPMENT   *
1 !*  CORPORATION.                                                            *
1 !*                                                                          *
1 !*  DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE  OR  RELIABILITY OF ITS   *
1 !*  SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL.                 *
1 !*                                                                          *
1 !*                                                                          *
1 !****************************************************************************
1 !
1
1 !++
1 ! FACILITY:     Pascal Language Support
1 !
1 ! ABSTRACT:
1 !
1 !       This module contains procedures to allocate and deallocate
1 !       virtual memory.
1 !
1 ! ENVIRONMENT:  User mode - AST reentrant
1 !
1 ! AUTHOR: Steven B. Lionel, CREATION DATE: 1-April-1981
1 !
1 ! MODIFIED BY:
1 !
1 ! 1-001 - Original.  SBL 1-April-1981
1 !--
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
```

```
1 %SBTTL 'PAS$$GET_VM - Allocate virtual memory'
1 GLOBAL ROUTINE PAS$$GET_VM (
1         PFV: REF $PAS$PFV_FILE_VARIABLE,        ! File variable
1         LENGTH                                  ! Length to allocate
1     ): CALL_VM =
1
1 !++
1 ! FUNCTIONAL DESCRIPTION:
1 !
1 !       This procedure allocates a block of virtual memory and returns
1 !       a pointer to the block as its function value.  The memory allocated
1 !       is zero-filled.
1 !
1 ! CALLING SEQUENCE:
1 !
1 !       block_addr.wa.v = PAS$$GET_VM (PFV.rr.r, LENGTH.rl.v)
1 !
1 ! FORMAL PARAMETERS:
1 !
1 !       PFV             - The Pascal File Variable (PFV) passed by reference.
1 !                         The structure of the PFV is defined in PASPFV.REQ.
1 !
1 !       LENGTH          - The length in bytes of the block to allocate.
1 !                         Must not be greater than 65535.
1 !
1 ! IMPLICIT INPUTS:
1 !
1 !       NONE
1 !
1 ! IMPLICIT OUTPUTS:
1 !
1 !       NONE
1 !
1 ! ROUTINE VALUE:
1 !
1 !       A pointer to the allocated block.
1 !
1 ! SIDE EFFECTS:
1 !
1 !       Allocates virtual memory.
1 !
1 ! SIGNALLED ERRORS:
1 !
1 !       INSVIRMEM - Insufficient virtual memory
1 !
1 !--
1
2     BEGIN
2
2     LOCAL
2         BLOCK_ADDR;                     ! Address of allocated block
2
2     IF .LENGTH GTRU 65535
2     THEN
2         $PAS$BUGCHECK (BUG_BADVMSIZE);
2
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$$GET_VM, Save R2,R3,R4,R5                  ; 0142
                        5E           04     C2 00002             SUBL2   #4, SP
              0000FFFF  8F     04    AC     D1 00005             CMPL    LENGTH, #65535                                 ; 0193
                                     0B     1B 0000D             BLEQU   1$
                                     04     DD 0000F             PUSHL   #4                                             ; 0195
              00000000G 00           01     FB 00011             CALLS   #1, PAS$$BUGCHECK
                                     2A     11 00018             BRB     3$
                                     5E     DD 0001A 1$:         PUSHL   SP                                             ; 0197
                               04    AC     9F 0001C             PUSHAB  LENGTH
              00000000G 00           02     FB 0001F             CALLS   #2, LIB$GET_VM
                        0F           50     E8 00026             BLBS    R0, 2$
                                     7E     D4 00029             CLRL    -(SP)                                          ; 0199
                        7E     00G   8F     9A 0002B             MOVZBL  #PAS$K_INSVIRMEM, -(SP)
              00000000G 00           02     FB 0002F             CALLS   #2, PAS$$SIGNAL
                                     0C     11 00036             BRB     3$
  04   AC          00   6E           00     2C 00038 2$:         MOVC5   #0, (SP), #0, LENGTH, @BLOCK_ADDR              ; 0201
                               00    BE        0003E
                        50           6E     D0 00040             MOVL    BLOCK_ADDR, R0                                 ; 0203
                                            04 00043             RET
                                     50     D4 00044 3$:         CLRL    R0                                             ; 0205
                                            04 00046             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
```

0297 AH-BT13A-SE
VAX/VMS V4.0

DIGITAL EQUIPMENT CORPORATION
CONFIDENTIAL AND PROPRIETARY

PASREWRIT LIS

PASSIGNAL LIS

PASSQR 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