```
LLL                 IIIIIIIII     NNN         NNN  KKK         KKK  EEEEEEEEEEEEEEE   RRRRRRRRRRR
LLL                 IIIIIIIII     NNN         NNN  KKK         KKK  EEEEEEEEEEEEEEE   RRRRRRRRRRR
LLL                 IIIIIIIII     NNN         NNN  KKK         KKK  EEEEEEEEEEEEEEE   RRRRRRRRRRR
LLL                    III        NNN         NNN  KKK         KKK  EEE               RRR         RRR
LLL                    III        NNN         NNN  KKK         KKK  EEE               RRR         RRR
LLL                    III        NNN         NNN  KKK         KKK  EEE               RRR         RRR
LLL                    III        NNNNNN      NNN  KKK       KKK    EEE               RRR         RRR
LLL                    III        NNNNNN      NNN  KKK       KKK    EEE               RRR         RRR
LLL                    III        NNNNNN      NNN  KKK       KKK    EEE               RRR         RRR
LLL                    III        NNN   NNN   NNN  KKKKKKKKK        EEEEEEEEEEE       RRRRRRRRRRR
LLL                    III        NNN   NNN   NNN  KKKKKKKKK        EEEEEEEEEEE       RRRRRRRRRRR
LLL                    III        NNN   NNN   NNN  KKKKKKKKK        EEEEEEEEEEE       RRRRRRRRRRR
LLL                    III        NNN      NNNNNN  KKK       KKK    EEE               RRR   RRR
LLL                    III        NNN      NNNNNN  KKK       KKK    EEE               RRR   RRR
LLL                    III        NNN      NNNNNN  KKK       KKK    EEE               RRR   RRR
LLL                    III        NNN         NNN  KKK         KKK  EEE               RRR      RRR
LLL                    III        NNN         NNN  KKK         KKK  EEE               RRR      RRR
LLL                    III        NNN         NNN  KKK         KKK  EEE               RRR      RRR
LLLLLLLLLLLLLLLL    IIIIIIIII     NNN         NNN  KKK         KKK  EEEEEEEEEEEEEEE   RRR         RRR
LLLLLLLLLLLLLLLL    IIIIIIIII     NNN         NNN  KKK         KKK  EEEEEEEEEEEEEEE   RRR         RRR
LLLLLLLLLLLLLLLL    IIIIIIIII     NNN         NNN  KKK         KKK  EEEEEEEEEEEEEEE   RRR         RRR
```

FILEID**LNKVMALLO

```
LL          NN      NN  KK      KK  VV      VV  MM      MM    AAAAAA    LL          LL            000000
LL          NN      NN  KK      KK  VV      VV  MM      MM    AAAAAA    LL          LL            000000
LL          NN      NN  KK      KK  VV      VV  MMMM  MMMM  AA      AA  LL          LL          00      00
LL          NN      NN  KK      KK  VV      VV  MMMM  MMMM  AA      AA  LL          LL          00      00
LL          NNNN    NN  KK    KK    VV      VV  MM  MM  MM  AA      AA  LL          LL          00      00
LL          NNNN    NN  KK    KK    VV      VV  MM  MM  MM  AA      AA  LL          LL          00      00
LL          NN  NN  NN  KKKKKK      VV      VV  MM      MM  AA      AA  LL          LL          00      00
LL          NN  NN  NN  KKKKKK      VV      VV  MM      MM  AA      AA  LL          LL          00      00
LL          NN    NNNN  KK    KK    VV      VV  MM      MM  AAAAAAAAAA  LL          LL          00      00
LL          NN    NNNN  KK    KK    VV      VV  MM      MM  AAAAAAAAAA  LL          LL          00      00
LL          NN      NN  KK      KK    VV  VV    MM      MM  AA      AA  LL          LL          00      00      ....
LL          NN      NN  KK      KK    VV  VV    MM      MM  AA      AA  LL          LL          00      00      ....
LLLLLLLLLL  NN      NN  KK      KK      VV      MM      MM  AA      AA  LLLLLLLLLL  LLLLLLLLLL    000000        ....
LLLLLLLLLL  NN      NN  KK      KK      VV      MM      MM  AA      AA  LLLLLLLLLL  LLLLLLLLLL    000000        ....

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 lnk_vmallo
;  2          0002  0                 (ident = 'V04-000'
;  3          0003  0                 ,addressing_mode
;  4          0004  0                         (external    = general
;  5          0005  0                         ,nonexternal = long_relative
;  6          0006  0                         )
;  7          0007  0                 ) =
;  8          0008  0
;  9          0009  1 begin
; 10          0010  1 !
; 11          0011  1 !
; 12          0012  1 !*****************************************************************************
; 13          0013  1 !*                                                                           *
; 14          0014  1 !*  COPYRIGHT (c) 1978, 1980, 1982, 1984 BY                                 *
; 15          0015  1 !*  DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASSACHUSETTS.                   *
; 16          0016  1 !*  ALL RIGHTS RESERVED.                                                     *
; 17          0017  1 !*                                                                           *
; 18          0018  1 !*  THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND COPIED   *
; 19          0019  1 !*  ONLY IN  ACCORDANCE WITH  THE  TERMS  OF  SUCH  LICENSE  AND WITH THE  *
; 20          0020  1 !*  INCLUSION OF THE ABOVE COPYRIGHT NOTICE. THIS SOFTWARE OR  ANY  OTHER   *
; 21          0021  1 !*  COPIES THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY   *
; 22          0022  1 !*  OTHER PERSON.  NO TITLE TO AND OWNERSHIP OF  THE  SOFTWARE IS  HEREBY   *
; 23          0023  1 !*  TRANSFERRED.                                                             *
; 24          0024  1 !*                                                                           *
; 25          0025  1 !*  THE INFORMATION IN THIS SOFTWARE IS  SUBJECT TO CHANGE WITHOUT NOTICE   *
; 26          0026  1 !*  AND  SHOULD  NOT  BE  CONSTRUED AS  A COMMITMENT BY DIGITAL EQUIPMENT   *
; 27          0027  1 !*  CORPORATION.                                                             *
; 28          0028  1 !*                                                                           *
; 29          0029  1 !*  DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE  OR  RELIABILITY OF ITS   *
; 30          0030  1 !*  SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL.                  *
; 31          0031  1 !*                                                                           *
; 32          0032  1 !*                                                                           *
; 33          0033  1 !*****************************************************************************
; 34          0034  1 !
; 35          0035  1 !
; 36          0036  1 !
; 37          0037  1 !
; 38          0038  1 !
; 39          0039  1 !++
; 40          0040  1 !
; 41          0041  1 ! MODULE: LNK_VMALLO
; 42          0042  1 !
; 43          0043  1 ! FACILITY: LINKER
; 44          0044  1 !
; 45          0045  1 ! ABSTRACT: ALLOCATION OF VIRTUAL MEMORY TO IMAGE.
; 46          0046  1 !
; 47          0047  1 ! HISTORY:
; 48          0048  1 !
; 49          0049  1 !       VERSION: X01.00
; 50          0050  1 !
; 51          0051  1 !       AUTHOR: T.J. PORTER 30-MAR-77
; 52          0052  1 !
; 53          0053  1 ! MODIFIED BY:
; 54          0054  1 !
; 55          0055  1 !       V03-017 ADE0001         Alan D. Eldridge        30-Jul-1984
; 56          0056  1 !               Use error returned by LNK$REQUESTMEM rather than assuming
; 57          0057  1 !               LIN$_MEMFUL.
```

```
;    58          0058  1 !
;    59          0059  1 !       V03-016 JWT0152         Jim Teague              20-Feb-1984
;    60          0060  1 !               Adjustment for longer global section names.
;    61          0061  1 !
;    62          0062  1 !       V03-015 JWT0088         Jim Teague              12-Jan-1983
;    63          0063  1 !               Filter 0-length psects out of DMT.
;    64          0064  1 !
;    65          0065  1 !       V03-014 JWT0083         Jim Teague              07-Jan-1983
;    66          0066  1 !               Don't let Linker fill in vacant low address space in
;    67          0067  1 !               in based executable images.
;    68          0068  1 !
;    69          0069  1 !       V03-013 JWT0078         Jim Teague              15-Dec-1982
;    70          0070  1 !               Change implementation of DMT.
;    71          0071  1 !
;    72          0072  1 !       V03-012 JWT0074         Jim Teague              09-Dec-1982
;    73          0073  1 !               Finalize DMT info.
;    74          0074  1 !
;    75          0075  1 !       V03-011 JWT0061         Jim Teague              22-Oct-1982
;    76          0076  1 !               Add debugger image section for images linked /debug.
;    77          0077  1 !
;    78          0078  1 !--
```

```
;    80          0079  1 !
;    81          0080  1 !++
;    82          0081  1 !
;    83          0082  1 ! FUNCTIONAL DESCRIPTION:
;    84          0083  1 !
;    85          0084  1 !       THIS MODULE CONTAINS THE VIRTUAL MEMORY ALLOCATION LOGIC
;    86          0085  1 !       OF THE LINKER. THERE IS ONE ROUTINE:
;    87          0086  1 !               LNK$VMALLO()
;    88          0087  1 !       WHICH ALLOCATES MEMORY TO P-SECTIONS OF EACH OF THE
;    89          0088  1 !       FOUR POSSIBLE IMAGE SECTIONS, READ-ONLY, READ-WRITE
;    90          0089  1 !       EXECUTE-ONLY AND EXECUTE-WRITE, AS WELL AS TO
;    91          0090  1 !       THE USER STACK.
;    92          0091  1 !       AS EACH IMAGE-SECTION IS BUILT, THE P-SECTIONS WHOSE
;    93          0092  1 !       ATTRIBUTES MATCH THOSE OF THE IMAGE SECTION ARE
;    94          0093  1 !       ASSIGNED A BASE ADDRESS IN LEXICAL ORDER. AS EACH
;    95          0094  1 !       IS ASSIGNED ITS BASE ADDRESS, ANY RELOCATABLE SYMBOLS OWNED
;    96          0095  1 !       BY THAT P-SECTION ARE RELOCATED BY ADDITION OF
;    97          0096  1 !       THE ALIGNED P-SECTION BASE ADDRESS.
;    98          0097  1 !       AFTER ALL REQUIRED IMAGE SECTIONS HAVE BEEN BUILT
;    99          0098  1 !       THE HEADER SIZE IS CALCULATED AND THE BASE VIRTUAL
;   100          0099  1 !       BLOCK NUMBER IN EACH ISD (ON SINGLY LINKED LIST)
;   101          0100  1 !       IS RELOCATED BY THE NUMBER OF HEADER BLOCKS.
;   102          0101  1 !
;   103          0102  1 !--
;   104          0103  1 !
;   105          0104  1 library
;   106          0105  1         'LIBL32';                               ! SYSTEM USER DATA STRUCTURES
;   107          0106  1 require
;   108          0107  1         'PREFIX';                               ! GENERAL DATA, MACROS ETC.
;   109          0222  1 library
;   110          0223  1         'DATBAS';                               ! DATA BASE DESCRIPTION
;   111          0224  1 require
;   112          0225  1         'ISGENC';                               ! GENERATION CONTROL TABLES
;   113          0609  1 !
;   114          0610  1 forward routine
;   115          0611  1         allocluster : novalue,                  ! GATHER P-SECTIONS INTO I-SECTIONS FOR CURRENT CLUSTER
;   116          0612  1         compare_bases,                          ! ACTION ROUTINE FOR LIB$INSERT_TREE
;   117          0613  1         alloc_node,                             ! ACTION ROUTINE FOR LIB$INSERT_TREE
;   118          0614  1         relocluster : novalue,                  ! RELOCATE CONTENTS OF CURRENT CLUSTER
;   119          0615  1         lnk$crefixisd,                          ! BUILD FIXUP SECTION ISD
;   120          0616  1         lnk$cluvirmem,                          ! ALLOCATE VIRTUAL MEMORY FOR A CLUSTER
;   121          0617  1         lnk$setlastclu,                         ! TEST/SET LAST CLUSTER
;   122          0618  1         stack_isectbld : novalue,               ! BUILD USER STACK IMAGE SECTION
;   123          0619  1         normal_isectbld : novalue,              !       NORMAL USER ISECT
;   124          0620  1         buildisd : novalue;                     ! CREATE ISD
;   125          0621  1 !
;   126          0622  1 external
;   127          0623  1         lnk$gl_omddst,                          ! COUNT OF OBJMODS CONTRIBUTING TO DST
;   128          0624  1         lnk$gw_pscdst: word,                    ! COUNT OF PSECTS IN ABOVE OBJMODS
;   129          0625  1         lnk$gl_shrsyms,                         ! NUMBER OF SYMBOLS IN OTHER SHAREABLE IMAGES
;   130          0626  1         lnk$gl_shrimgs,                         ! NUMBER OF SHAREABLE IMAGES REFERENCED
;   131          0627  1         lnk$gl_shrclstrs,                       ! NUMBER OF PIC/NON-PIC SHR IMAGES IN LINK
;   132          0628  1         lnk$al_valctltb,                        ! ADDRESS OF CREF BY VALUE CONTROL TABLE
;   133          0629  1         lnk$al_sytblfmt,                        ! CREF CONTROL TABLE
;   134          0630  1         lnk$gl_tfrpsc : ref block[,byte],       ! POINTER TO P-SECT CONTAING USER
;   135          0631  1         lnk$gl_tfradr,                          ! TRANSFER ADDRESS
;   136          0632  1         lnk$gl_dbgtfps : ref block[,byte],      ! POINTER TO PSECTION CONTAINING DEBUG
```

```
;  137          0633  1          lnk$gl_dbgtfr,                          ! TRANSFER ADDRESS
;  138          0634  1          lnk$gl_ctlmsk : block[,byte],           ! LINK CONTROL MASK
;  139          0635  1          lnk$gl_minva,                           ! LOWEST VIRTUAL ADDRESS ALLOCATED
;  140          0636  1          lnk$gl_clulst : vector[2],              ! CLUSTER DESCRIPTOR LISTHEAD
;  141          0637  1          lnk$gl_lastclu : ref block[,byte],      ! POINTER TO LAST CLUSTER DESCRIPTOR ALLOCATED
;  142          0638  1          lnk$gl_curclu : ref block[,byte],       ! CURRENT CLUSTER DESCRIPTOR POINTER
;  143          0639  1          lnk$gl_defclu : block[,byte],           ! DEFAULT CLUSTER
;  144          0640  1          lnk$gl_fvmlst : ref block[,byte];       ! FREE V.M. LIST ENTRY
;  145          0641  1  !
;  146          0642  1  external routine
;  147          0643  1          crf$insrtkey,                           ! INSERT KEY
;  148          0644  1          crf$insrtref,                           ! INSERT REF TO KEY IN CREF TABLE
;  149          0645  1          lib$insert_tree,                        ! INSERT INTO BALANCED BINARY TREE
;  150          0646  1          lnk$alloblk : novalue,                  ! DYNAMIC MEMORY ALLOCATOR
;  151          0647  1          lnk$allovirmem,                         ! ALLOCATE SPECIFICE REGION OF VIRTUAL MEMORY
;  152          0648  1          lnk$findvirmem,                         ! FIND A LARGE ENOUGH REGION OF   ''      ''
;  153          0649  1          lnk$fndpscmsk,                          ! FIND P-SECTIONS BY MASK
;  154          0650  1          lnk$requestmem;                         ! REQUEST SOME VM FOR IMAGE BUFFER
;  155          0651  1  !
;  156          0652  1  external literal
;  157          0653  1          lin$_basesym,                           ! BAD BASE ADDRESS SYMBOL
;  158          0654  1          lin$_clubelbas,                         ! CLUSTER= BASED CLUSTER BELOW BASE= BASE
;  159          0655  1          lin$_confmem,                           ! CONFLICTING MEMORY REQUIREMENTS
;  160          0656  1          lin$_insvirmem,                         ! INSUFFICIENT VIRTUAL MEMORY
;  161          0657  1          lin$_memful,
;  162          0658  1          lin$_noimgfil,                          ! NO IMAGE FILE CREATED
;  163          0659  1          lin$_stkovflo;                          ! STACK OVERFLOW
;  164          0660  1  !
;  165          0661  1  global
;  166          0662  1          lnk$gl_dmtbuffer: vector[2,long],       ! BUFFER FOR DMT
;  167          0663  1          lnk$gl_dmtbytes,                        ! SIZE OF DBG MODULE/PSECT INFO TABLE
;  168          0664  1          lnk$gl_lowclu : ref block[,byte],       ! ADDR OF CLUSTER DESCRIPTOR OF LOWEST VA CLUSTER (NO SHR IM
;  169          0665  1          lnk$gl_highclu : ref block[,byte],      ! ADDR OF CLUSTER DESCRIPTOR OF HIGHEST VA CLUSTER (NO SHR I
;  170          0666  1          lnk$gl_lstclstr : ref block[,byte],     ! ADDR OF CLUSTER DESCRIPTOR OF LAST CLUSTER (INCLUDES SHR I
;  171          0667  1          lnk$gl_fixisd : ref block[,byte],       ! ADDR OF FIXUP ISD
;  172          0668  1          lnk$gl_1stgadr,                         ! VA OF FIRST SPECIAL G^ FIXUP
;  173          0669  1          lnk$gl_lastgadr,                        ! VA OF LAST  ''
;  174          0670  1          lnk$gw_shriscts : word,                 ! NUMBER OF ISECTS FROM PIC SHAREABLE IMAGES
;  175          0671  1          lnk$gl_maplst,                          ! LIST OF P-SECTIONS FOR MAP OUTPUT (IN V.A. ORDER)
;  176          0672  1          lnk$gw_stack : word,                    ! STACK SIZE (PAGES)
;  177          0673  1          lnk$gw_nisects : word;                  ! NUMBER OF IMAGE SECTIONS
;  178          0674  1  !
;  179          0675  1  global literal
;  180          0676  1          def$c_stack = 20,                       ! DEFAULT STACK SIZE (PAGES)
;  181          0677  1          def$c_base = %x'200';                   ! DEFAULT BASE ADDRESS
;  182          0678  1  !
;  183          0679  1  !       GENERATE IMAGE SECTION GENERATION CONTROL TABLES
;  184          0680  1  !
;  185          0681  1          psect own = $plit$ (nopic,concatenate,local,noshare,noexecute,nowrite);
;  186          0682  1
;  187        P 0683  1  isgentbl(exe,                                   ! EXECUTABLE IMAGE
;  188        P 0684  1          _u_r_,  rdwrtexerelvec,                 ! READ ONLY PART
;  189        P 0685  1          _u_rw,  rdwrtexerelvec,                 ! READ WRITE DATA
;  190        P 0686  1          _u_x_,  rdwrtexerelvec,                 ! EXECUTE ONLY
;  191        P 0687  1          _u_xw,  rdwrtexerelvec,                 ! OVERWRITTEN CODE
;  192        P 0688  1          _u_r_v, rdwrtexerelvec,                 ! READ ONLY VECTOR
;  193        P 0689  1          _u_rwv, rdwrtexerelvec,                 ! READ WRITE VECTOR
```

```
;   194         P 0690  1           _u_x_v, rdwrtexerelvec,                  ! EXECUTE ONLY VECTOR
;   195         P 0691  1           _u_xwv, rdwrtexerelvec,                  ! OVERWRITTEN CODE VECTOR
;   196           0692  1           _stack,0);                               ! STACK -- MUST BE LAST
;   197         P 0693  1   isgentbl(sys,                                    ! SYSTEM IMAGE
;   198           0694  1           s_alone,gps$m_rel);                      ! IS A SINGLE SECTION
;   199         P 0695  1   isgentbl(shr,                                    ! SHAREABLE IMAGE
;   200         P 0696  1           _s_r_,  shrpicvec,
;   201         P 0697  1           _s_rw,  shrpicvec,
;   202         P 0698  1           _s_x_,  shrpicvec,
;   203         P 0699  1           _s_xw,  shrpicvec,
;   204         P 0700  1           ___r_,  shrpicvec,
;   205         P 0701  1           ___rw,  shrpicvec,
;   206         P 0702  1           ___x_,  shrpicvec,
;   207         P 0703  1           ___xw,  shrpicvec,
;   208         P 0704  1           _spr_,  shrpicvec,
;   209         P 0705  1           _sprw,  shrpicvec,
;   210         P 0706  1           _spx_,  shrpicvec,
;   211         P 0707  1           _spxw,  shrpicvec,
;   212         P 0708  1           __pr_,  shrpicvec,
;   213         P 0709  1           __prw,  shrpicvec,
;   214         P 0710  1           __px_,  shrpicvec,
;   215         P 0711  1           __pxw,  shrpicvec,
;   216         P 0712  1           _s_r_v, shrpicvec,                       ! VECTOR IMAGE SECTIONS
;   217         P 0713  1           _s_rwv, shrpicvec,
;   218         P 0714  1           _s_x_v, shrpicvec,
;   219         P 0715  1           _s_xwv, shrpicvec,
;   220         P 0716  1           ___r_v, shrpicvec,
;   221         P 0717  1           ___rwv, shrpicvec,
;   222         P 0718  1           ___x_v, shrpicvec,
;   223         P 0719  1           ___xwv, shrpicvec,
;   224         P 0720  1           _spr_v, shrpicvec,
;   225         P 0721  1           _sprwv, shrpicvec,
;   226         P 0722  1           _spx_v, shrpicvec,
;   227         P 0723  1           _spxwv, shrpicvec,
;   228         P 0724  1           __pr_v, shrpicvec,
;   229         P 0725  1           __prwv, shrpicvec,
;   230         P 0726  1           __px_v, shrpicvec,
;   231         P 0727  1           __pxwv, shrpicvec,
;   232         P 0728  1           __pxwv, shrpicvec,
;   233           0729  1           _stack,0);                                   ! MUST BE THE LAST
;   234           0730  1
;   235           0731  1           psect own = $own$ (nopic,concatenate,local,noshare,noex .ute,write);
;   236           0732  1
;   237           0733  1   own
;   238           0734  1           defclubased,                                 ! REMEMBER IF DEFAULT CLUSTER IS BASED
;   239           0735  1           curpsect : ref block[,byte],                 ! POINTER TO CURRENT PSECT DESCRIPTOR
;   240           0736  1           current_base,                                ! CURRENT BASE IN ISECT
;   241           0737  1           isect_base,                                  ! BASE OF THE IMAGE SECTION
;   242           0738  1           cntrl_reg_addr : initial(system_space),      ! CONTROL REGION ADDRESS (FOR NEXT ISECT ALL
;   243           0739  1           isect : byte,                                ! I-SECT CONTROL TABLE INDEX
;   244           0740  1           numisects : byte,                            ! POTENTIAL NUMBER OF ISECTS IN THIS CLUSTER
;   245           0741  1           isectgentbl : ref blockvector[,isc$c_size,byte],  ! ISECT GENERATION TABLE FOR CURRENT CLUSTER
;   246           0742  1           previsect : ref block[,byte],                ! POINTER TO PREVIOUS ISECT DESCRIPTOR BLOCK
;   247           0743  1           isctdesc : ref block[,byte];                 ! I-SECT DESCRIPTOR BLOCK
```

```
;  249     0744  1 global routine lnk$vmallo : novalue =
;  250     0745  2 begin
;  251     0746  2 !++
;  252     0747  2 !       THIS ROUTINE IS THE DRIVER FOR MEMORY ALLOCATION.
;  253     0748  2 !
;  254     0749  2 !       TWO PASES OF THE CLUSTER LIST ARE MADE. THE FIRST PASS ALLOCATES
;  255     0750  2 !       SPECIFIC ADDRESS SPACE TO CLUSTERS WHICH HAVE BEEN GIVEN A
;  256     0751  2 !       BASE ADDRESS . THE SECOND PASS
;  257     0752  2 !       ALLOCATES MEMORY TO POSITION INDEPENDENT SHAREABLE IMAGES AND
;  258     0753  2 !       OTHER USER CLUSTERS NOT EXPLICITLY POSITIONED. THIS IS DONE IN
;  259     0754  2 !       CLUSTER LIST ORDER AND THE FIRST SPACE LARGE ENOUGH FROM THE BASE
;  260     0755  2 !       OF THE REGION IS ALLOCATED.
;  261     0756  2 !--
;  262     0757  2 local   status,
;  263     0758  2         baseaddrsym     : ref block[,byte],                     ! THE SYMBOL TABLE ENTRY FOR THE SYMBOL
;  264     0759  2         basesymsnb      : ref block[,byte];
;  265     0760  2 !
;  266     0761  2 ! SELECT THE CORRECT ISECT GENERATION CONTROL TABLE
;  267     0762  2 !
;  268     0763  2 if .lnk$gl_ctlmsk[lnk$v_exe]                                    ! IF NORMAL EXE IMAGE POINT TO
;  269     0764  3 then begin
;  270     0765  3      isectgentbl =  exe_isgentbl;                               ! ITS CONTROL TABLE AND SET
;  271     0766  3      numisects   = .exe_isects;                                 ! NUMBER OF POTENTIAL ISECTS PER CLUSTER
;  272     0767  3      end
;  273     0768  2 else if .lnk$gl_ctlmsk[lnk$v_sys]                               ! POINT TO SYSTEM IMAGE CONTROL TABLE
;  274     0769  3      then begin
;  275     0770  3           isectgentbl =  sys_isgentbl;                          ! AND THEN
;  276     0771  3           numisects   = .sys_isects;                            ! POTENTIAL NUMBER OF ISECTS
;  277     0772  3           end
;  278     0773  3      else begin                                                 ! IT MUST BE A SHAREABLE IMAGE
;  279     0774  3           isectgentbl =  shr_isgentbl;                          ! SO SET THE CONTROL TABLE
;  280     0775  3           numisects   = .shr_isects;                            ! AND NUMBER OF ISECTS PER CLUSTER
;  281     0776  2           end;
;  282     0777  2
;  283     0778  2 defclubased    = .lnk$gl_defclu [clu$v_based];                  ! REMEMBER IF DEFAULT CLUSTER BASED
;  284     0779  2 lnk$gl_highclu = 0;                                             ! INITIALLY POINTS TO NO CLUSTER
;  285     0780  2 lnk$gl_lowclu  = 0;                                             !
;  286     0781  2 lnk$gl_curclu  = lnk$gl_clulst [0];                             ! START AT TOP OF CLUSTER LIST
;  287     0782  2 while (lnk$gl_curclu = .lnk$gl_curclu [clu$l_nxtclu]) neq 0     ! GET NEXT CLUSTER DESCRIPTOR
;  288     0783  2 do if not .lnk$gl_curclu [clu$v_shrimg]                         ! AND IF A USER CLUSTER WITH A
;  289     0784  2    and    .lnk$gl_curclu [clu$v_based]                          ! BASE ADDRESS SPECIFIED
;  290     0785  3    then begin                                                   ! ITS TIME HAS COME
;  291     0786  3         lnk$gl_ctlmsk [lnk$v_picimg] = false;                   ! THE IMAGE IS NON-PIC
;  292     0787  3         baseaddrsym = .lnk$gl_curclu [clu$l_base];              ! CELL CONTAINS POINTER TO THE SYMBOL TABLE
;  293     0788  3         if  .lnk$gl_curclu [clu$v_symbas]                       ! IF THE ADDRESS IS SYMBOLICALLY SPECIFIED
;  294     0789  3         and .baseaddrsym neq 0                                  !  AND IT WAS DEFINED
;  295     0790  4         then begin                                              ! FIRST FIGURE OUT IF VALID
;  296     0791  4              basesymsnb = .baseaddrsym - .baseaddrsym [sym$b_namlng]
;  297     0792  4                                       - snb$c_fxdlen;           ! POINT TO NAME BLOCK
;  298     0793  6              if (.baseaddrsym [sym$w_flags] and (gsy$m_def      ! FIRST CHECK ABSOLUTE DEFINITION
;  299     0794  4                        or gsy$m_rel)) neq gsy$m_def
;  300     0795  5              then begin
;  301     0796  5                   signal (lin$_basesym,1,                       ! AND ISSUE ERROR IF NOT
;  302     0797  5                           basesymsnb [snb$b_namlng]);
;  303     0798  5                   if .lnk$gl_ctlmsk [lnk$v_sys]                 ! IF A SYSTEM IMAGE SET THE
;  304     0799  5                   then lnk$gl_curclu [clu$l_base] = system_space  ! BASE ADDRESS FOR IT
;  305     0800  5                   else lnk$gl_curclu [clu$l_base] = 0;          ! OTHERWISE DEFAULT TO ZERO
```

```
;  306     0801  5                end
;  307     0802  5           else begin
;  308     0803  6                if not (.lnk$gl_ctlmsk [lnk$v_sys] and not    ! OTHERWISE UNLESS SYSTEM IMAGE WITHOUT A HE
;  309     0804  6                        .lnk$gl_ctlmsk [lnk$v_sysheadr])
;  310     0805  5                then lnk$gl_curclu [clu$l_base] = .baseaddrsym [sym$l_value] + ! THEN SET TO NEXT PAGE BEY
;  311     0806  5                                                  511 and not 511
;  312     0807  5                else lnk$gl_curclu [clu$l_base] = .baseaddrsym [sym$l_value]; ! IF NO /HEADER, THEN DON'T
;  313     0808  4                end;
;  314     0809  4           if  .lnk$gl_ctlmsk [lnk$v_sys]                                    ! IF A SYSTEM IMAGE SYMBOLIC
;  315     0810  4           then lnk$gl_fvmlst [fvm$l_address] = .lnk$gl_curclu [clu$l_base];  ! BASED THEN SET V.M. BASE
;  316     0811  4           end;
;  317     0812  3      if  .defclubased                                                 ! IF DEFAULT CLUSTER IS BASED
;  318     0813  3      and .lnk$gl_curclu [clu$l_base] lssu .lnk$gl_defclu [clu$l_base]
;  319     0814  3      then   signal   ( lin$_clubelbas, 3
;  320     0815  3                     ,.lnk$gl_curclu [clu$l_base]
;  321     0816  3                     , lnk$gl_curclu [clu$b_namlng]
;  322     0817  3                     ,.lnk$gl_defclu [clu$l_base]
;  323     0818  3                     );
;  324     0819  3      allocluster ();                                                  ! GO ALLOCATE THE CLUSTER
;  325     0820  3      if not lnk$allovirmem (.lnk$gl_curclu [clu$l_base]               ! CONTENTS TO VIRTUAL MEMORY
;  326     0821  3                            ,.lnk$gl_curclu [clu$l_pages])            ! THEN ATTEMPT TO ALLOCATE THAT REGION
;  327     0822  4      then begin
;  328     0823  4           lnk$gl_ctlmsk[lnk$v_image] = false;                         ! TURN OFF IMAGE PRODUCTION
;  329     0824  4           signal ( lin$_confmem, 3                                    ! AND ISSUE ERROR ON FAILURE
;  330     0825  4                  ,.lnk$gl_curclu[clu$l_base]
;  331     0826  4                  ,.lnk$gl_curclu[clu$l_pages]
;  332     0827  4                  , lnk$gl_curclu [clu$b_namlng]
;  333     0828  4                  , lin$_noimgfil
;  334     0829  4                  );
;  335     0830  3           end;
;  336     0831  3      lnk$setlastclu (.lnk$gl_curclu);                                ! SEE IF LAST CLUSTER
;  337     0832  3      relocluster ();                                                 !
;  338     0833  3      end
;  339     0834  3
;  340     0835  2        else if  .lnk$gl_curclu [clu$v_shrimg]                        ! IF THIS IS A BASED SHAREABLE IMAGE
;  341     0836  2             and .lnk$gl_curclu [clu$v_based]
;  342     0837  2             then lnk$cluvirmem (.lnk$gl_curclu);                    ! ALLOCATE VIRTUAL MEMORY FOR CLUSTER
;  343     0838  2 !
;  344     0839  2 ! ALL EXPLICITLY BASED VIRTUAL MEMORY HAS NOW BEEN ALLOCATED. NOW MAKE
;  345     0840  2 ! ANOTHER PASS DOWN THE CLUSTER LIST, ALLOCATING MEMORY TO THE USER
;  346     0841  2 ! DEFINED CLUSTERS WHICH HAVE NO EXPLICIT ADDRESS SPECIFICATION.
;  347     0842  2 !
;  348     0843  2 baseaddrsym = .lnk$gl_minva;                                        ! SAVE MINIMUM ADDRESS
;  349     0844  2
;  350     0845  2 if not .lnk$gl_ctlmsk [lnk$v_shr]                                   ! IF NOT CREATING SHAREABLE IMAGE
;  351     0846  2 then if lnk$allovirmem (0,1)                                        !  FORCE ALLOCATE FIRST PAGE IF STILL FREE
;  352     0847  2      then lnk$gl_minva = .baseaddrsym;                              ! RESTORE MINIMUM ADDRESS IF SUCCEEDED
;  353     0848  2
;  354     0849  2 lnk$gl_curclu = lnk$gl_clulst[0];                                   ! START AT TOP OF LIST
;  355     0850  2
;  356     0851  2 while (lnk$gl_curclu = .lnk$gl_curclu[clu$l_nxtclu]) neq 0          ! GET NEXT CLUSTER DESCRIPTOR
;  357     0852  3 do  begin
;  358     0853  3     if  not .lnk$gl_curclu [clu$v_based]                            ! IF BASE ADDRESS NOT YET ESTABLISHED
;  359     0854  3     and not .lnk$gl_curclu [clu$v_shrimg]                           ! AND NOT FROM A SHAREABLE IMAGE
;  360     0855  4     then begin
;  361     0856  4          allocluster ();                                            ! GATHER P-SECTIONS
;  362     0857  4          lnk$cluvirmem (.lnk$gl_curclu);                            ! ALLOCATE VIRT. MEMORY FOR CLUSTER
```

```
;  363     0858  3            end;
;  364     0859  2        end;
;  365     0860  2
;  366     0861  2    ! ALLOCATE BUFFER FOR DMT INFORMATION
;  367     0862  2
;  368     0863  2    if (lnk$gl_dmtbytes = .lnk$gl_omddst * dcm$c_size +.lnk$gw_pscdst * dcp$c_size) gtr 0
;  369     0864  3    then if not (status = lnk$requestmem ( ((.lnk$gl_dmtbytes or 511)+1)/512, lnk$gl_dmtbuffer))
;  370     0865  3         then begin
;  371     0866  3              signal (.status, 0, lin$_noimgfil);
;  372     0867  3              lnk$gl_ctlmsk [lnk$v_image] = false;
;  373     0868  2              end;
;  374     0869  2
;  375     0870  2    !
;  376     0871  2    ! IF THIS IMAGE IS A BASED IMAGE, THEN ALLOCATE THE FIXUP SECTION
;  377     0872  2    ! NOW, IF THERE ARE ANY SHAREABLE IMAGES LINKED IN ALSO, SO THAT WHEN THE
;  378     0873  2    ! VIRTUAL MEMORY IS ALLOCATED FOR THESE SHAREABLE IMAGES, THEY DON'T OVERMAP
;  379     0874  2    ! THE FIXUP SECTION.  WE CAN'T ALLOCATE VIRTUAL MEMORY FOR THE FIXUP SECTION
;  380     0875  2    ! YET, SINCE WE DON'T KNOW THE EXTENT OF IT'S SIZE (DUE TO NOT KNOWING THE
;  381     0876  2    ! NUMBER OF .ADDRESS FIXUPS YET TO BE FOUND), WE CAN ONLY SPECIFY IT'S
;  382     0877  2    ! BASE, AND THEN COMPUTE THE SIZE AT LNK$FLUSHIMG AFTER PASS 2.
;  383     0878  2    !
;  384     0879  2    ! WE CAN DO THIS IF THE IMAGE IS BASED, BECAUSE THE .ADDRESS FIXUPS ARE NOT
;  385     0880  2    ! DONE THEN.
;  386     0881  2    !
;  387     0882  2    if  not .lnk$gl_ctlmsk[lnk$v_sys]
;  388     0883  2    and not .lnk$gl_ctlmsk[lnk$v_shr]
;  389     0884  2    and .defclubased
;  390     0885  3    then begin
;  391     0886  3         local  fixisdhdr : ref block[,byte];
;  392     0887  3
;  393     0888  3         lnk$crefixisd();
;  394     0889  3         fixisdhdr = lnk$gl_fixisd[isl$t_hdrisd];
;  395     0890  3         if not lnk$allovirmem(.fixisdhdr[isd$v_vpn]^9,.fixisdhdr[isd$w_pagcnt])
;  396     0891  4         then begin
;  397     0892  4              signal(lin$_confmem,3,
;  398     0893  4                              .fixisdhdr[isd$v_vpn]^9,.fixisdhdr[isd$w_pagcnt],
;  399     0894  4                             cstring('FIXUP SECTION'),lin$_noimgfil);
;  400     0895  4              lnk$gl_ctlmsk [lnk$v_image] = false;
;  401     0896  3              end;
;  402     0897  3         lnk$gl_fixisd [isl$v_memalo] = true;                    ! FLAG MEMORY ALLOCATED
;  403     0898  2         end;
;  404     0899  2
;  405     0900  2    !
;  406     0901  2    ! IF IMAGE IS BASED, MAKE ONE MORE PASS DOWN THE LIST AND ALLOCATE VM
;  407     0902  2    ! FOR PIC SHAREABLE IMAGES
;  408     0903  2    !
;  409     0904  2    lnk$gl_curclu = lnk$gl_clulst[0];                           ! START AT TOP OF LIST
;  410     0905  2
;  411     0906  2    if .defclubased                                             ! IF DEFAULT CLUSTER IS BASED (BASE=)
;  412     0907  2        and not .lnk$gl_ctlmsk[lnk$v_sys]
;  413     0908  2    then while (lnk$gl_curclu = .lnk$gl_curclu[clu$l_nxtclu]) neq 0         ! GET NEXT CLUSTER DESCRIPTOR
;  414     0909  3    do begin
;  415     0910  3        if not .lnk$gl_curclu[clu$v_based]                      ! IF BASE ADDRESS NOT YET ESTABLISHE
;  416     0911  3            and .lnk$gl_curclu[clu$v_shrimg]                    ! AND A SHAREABLE IMAGE
;  417     0912  4        then begin
;  418     0913  4            lnk$cluvirmem(.lnk$gl_curclu);                      ! ALLOCATE VIRT. MEMORY FOR CLUSTER
;  419     0914  3            end;
```

```
;  420          0915  2      end;
;  421          0916  2
;  422          0917  2  !
;  423          0918  2  ! IF LNK$GL_HIGHCLU IS 0, THEN THERE WERE PROBABLY NO OBJECT INPUT FILES,
;  424          0919  2  ! ONLY SHAREABLE IMAGES.  MAKE THE DEFAULT CLUSTER THE HIGHEST CLUSTER.
;  425          0920  2  !
;  426          0921  2  if .lnk$gl_highclu eql 0
;  427          0922  3  then begin
;  428          0923  3      if .lnk$gl_defclu[clu$l_fstfdb] eql 0                          ! IF IT HAS NO FILES IN IT
;  429          0924  4      then begin
;  430          0925  4          lnk$gl_lastclu[clu$l_nxtclu] = lnk$gl_defclu;              ! LINK IT ON THE END OF THE LIST
;  431          0926  4          if (lnk$gl_defclu[clu$l_prevclu] = .lnk$gl_lastclu) eql 0
;  432          0927  4              then lnk$gl_defclu[clu$l_prevclu] = lnk$gl_clulst;
;  433          0928  4          lnk$gl_lastclu = lnk$gl_defclu;
;  434          0929  3          end;
;  435          0930  3      lnk$allovirmem(0,1);                                       ! ALLOCATE FIRST PAGE
;  436          0931  3      lnk$gl_highclu = lnk$gl_defclu;                            ! MAKE DEFAULT CLUSTER THE HIGHEST C
;  437          0932  3      lnk$gl_lowclu = lnk$gl_defclu;
;  438          0933  2      end;
;  439          0934  2  !
;  440          0935  2  ! NOW CREATE THE FIXUP SECTION WITH THE ISD AS THE LAST ISECT IN THE IMAGE.
;  441          0936  2  ! THEN, GO THROUGH ALL THE CLUSTERS AND DEFINE THE SYMBOLS IN THE FIXUP SECTION
;  442          0937  2  ! AS NEEDED.
;  443          0938  2  !
;  444          0939  3  if (.lnk$gl_fixisd eql 0)
;  445          0940  4      and ((not .lnk$gl_ctlmsk[lnk$v_sys]
;  446          0941  5          and (.lnk$gl_shrclstrs neq 0)                          ! CREATE SECTION ONLY IF NEEDED
;  447          0942  4          and not .defclubased)
;  448          0943  3      or .lnk$gl_ctlmsk[lnk$v_exe])                              ! UNLESS EXECUTABLE, THEN ALWAYS CRE
;  449          0944  2      then lnk$crefixisd();
;  450          0945  2
;  451          0946  2  if .lnk$gl_tfrpsc neq 0                                        ! IF A TRANSFER ADDRESS EXISTS
;  452          0947  2      then lnk$gl_tfradr = .lnk$gl_tfradr + .lnk$gl_tfrpsc[psc$l_base];   ! THEN RELOCATE IT
;  453          0948  2  if .lnk$gl_dbgtfps neq 0 and .lnk$gl_dbgtfps neq T              ! IF A DEBUG TRANSFER ADDRESS (AND N
;  454          0949  2      then lnk$gl_dbgtfr = .lnk$gl_dbgtfr + .lnk$gl_dbgtfps[psc$l_base];  ! RELOCATE IT ALSO
;  455          0950  2  return;                                                        ! MEMORY ALLOCATION/RELOCATION COMPL
;  456          0951  1  end;                                          ! OF ROUTINE.


                                        .TITLE  LNK_VMALLO
                                        .IDENT  \V04-000\

                                        .PSECT  $PLIT$,NOWRT,NOEXE,2

                      09  00000 EXE_ISECTS:
                                        .BYTE   9                                      ;
                          00001         .BLKB   3
                    0348  00004 EXE_ISGENTBL:
                                        .WORD   840                                    ;
                    0008  00006         .WORD   8                                      ;
                      00  00008         .BYTE   0                                      ;
                      10  00009         .BYTE   16                                     ;
                      00  0000A         .BYTE   0                                      ;
                      00  0000B         .BYTE   0                                      ;
                      00  0000C         .BYTE   0                                      ;
                    0348  0000D         .WORD   840                                    ;
                    0108  0000F         .WORD   264                                    ;
```

```
                00011          .BYTE   0
                00012          .BYTE   16
                00013          .BYTE   0
                00014          .BYTE   10
                00015          .BYTE   0
                00016          .WORD   840
                00018          .WORD   72
                0001A          .BYTE   0
                0001B          .BYTE   16
                0001C          .BYTE   0
                0001D          .BYTE   0
                0001E          .BYTE   0
                0001F          .WORD   840
                00021          .WORD   328
                00023          .BYTE   0
                00024          .BYTE   16
                00025          .BYTE   0
                00026          .BYTE   10
                00027          .BYTE   0
                00028          .WORD   840
                0002A          .WORD   520
                0002C          .BYTE   0
                0002D          .BYTE   16
                0002E          .BYTE   0
                0002F          .BYTE   0
                00030          .BYTE   0
                00031          .WORD   840
                00033          .WORD   776
                00035          .BYTE   0
                00036          .BYTE   16
                00037          .BYTE   0
                00038          .BYTE   10
                00039          .BYTE   0
                0003A          .WORD   840
                0003C          .WORD   584
                0003E          .BYTE   0
                0003F          .BYTE   16
                00040          .BYTE   0
                00041          .BYTE   0
                00042          .BYTE   0
                00043          .WORD   840
                00045          .WORD   840
                00047          .BYTE   0
                00048          .BYTE   16
                00049          .BYTE   0
                0004A          .BYTE   10
                0004B          .BYTE   0
                0004C          .WORD   0
                0004E          .WORD   8
                00050          .BYTE   -3
                00051          .BYTE   12
                00052          .BYTE   0
                00053          .BYTE   12
                00054          .BYTE   0
                00055  SYS_ISECTS:
                               .BYTE   1
                00056          .BLKB   2
```

```
0008   00058 SYS_ISGENTBL:
                     .WORD   8                                ;
0008   0005A         .WORD   8                                ;
  00   0005C         .BYTE   0                                ;
  10   0005D         .BYTE   16                               ;
  00   0005E         .BYTE   0                                ;
  00   0005F         .BYTE   0                                ;
  00   00060         .BYTE   0                                ;
  22   00061 SHR_ISECTS:
                     .BYTE   34                               ;
       00062         .BLKB   2
0369   00064 SHR_ISGENTBL:
                     .WORD   873                              ;
0028   00066         .WORD   40                               ;
  01   00068         .BYTE   1                                ;
  10   00069         .BYTE   16                               ;
  00   0006A         .BYTE   0                                ;
  00   0006B         .BYTE   0                                ;
  00   0006C         .BYTE   0                                ;
0369   0006D         .WORD   873                              ;
0128   0006F         .WORD   296                              ;
  01   00071         .BYTE   1                                ;
  10   00072         .BYTE   16                               ;
  00   00073         .BYTE   0                                ;
  08   00074         .BYTE   8                                ;
  00   00075         .BYTE   0                                ;
0369   00076         .WORD   873                              ;
0068   00078         .WORD   104                              ;
  01   0007A         .BYTE   1                                ;
  10   0007B         .BYTE   16                               ;
  00   0007C         .BYTE   0                                ;
  00   0007D         .BYTE   0                                ;
  00   0007E         .BYTE   0                                ;
0369   0007F         .WORD   873                              ;
0168   00081         .WORD   360                              ;
  01   00083         .BYTE   1                                ;
  10   00084         .BYTE   16                               ;
  00   00085         .BYTE   0                                ;
  08   00086         .BYTE   8                                ;
  00   00087         .BYTE   0                                ;
0369   00088         .WORD   873                              ;
0008   0008A         .WORD   8                                ;
  02   0008C         .BYTE   2                                ;
  10   0008D         .BYTE   16                               ;
  00   0008E         .BYTE   0                                ;
  00   0008F         .BYTE   0                                ;
  00   00090         .BYTE   0                                ;
0369   00091         .WORD   873                              ;
0108   00093         .WORD   264                              ;
  02   00095         .BYTE   2                                ;
  10   00096         .BYTE   16                               ;
  00   00097         .BYTE   0                                ;
  0A   00098         .BYTE   10                               ;
  00   00099         .BYTE   0                                ;
0369   0009A         .WORD   873                              ;
0048   0009C         .WORD   72                               ;
  02   0009E         .BYTE   2                                ;
```

LN
V0
;

```
                10  0009F          .BYTE   16
                00  000A0          .BYTE   0
                00  000A1          .BYTE   0
                00  000A2          .BYTE   0
              0369  000A3          .WORD   873
              0148  000A5          .WORD   328
                02  000A7          .BYTE   2
                10  000A8          .BYTE   16
                00  000A9          .BYTE   0
                0A  000AA          .BYTE   10
                00  000AB          .BYTE   0
              0369  000AC          .WORD   873
              0029  000AE          .WORD   41
                03  000B0          .BYTE   3
                10  000B1          .BYTE   16
                00  000B2          .BYTE   0
                00  000B3          .BYTE   0
                00  000B4          .BYTE   0
              0369  000B5          .WORD   873
              0129  000B7          .WORD   297
                03  000B9          .BYTE   3
                10  000BA          .BYTE   16
                00  000BB          .BYTE   0
                08  000BC          .BYTE   8
                00  000BD          .BYTE   0
              0369  000BE          .WORD   873
              0069  000C0          .WORD   105
                03  000C2          .BYTE   3
                10  000C3          .BYTE   16
                00  000C4          .BYTE   0
                00  000C5          .BYTE   0
                00  000C6          .BYTE   0
              0369  000C7          .WORD   873
              0169  000C9          .WORD   361
                03  000CB          .BYTE   3
                10  000CC          .BYTE   16
                00  000CD          .BYTE   0
                08  000CE          .BYTE   8
                00  000CF          .BYTE   0
              0369  000D0          .WORD   873
              0009  000D2          .WORD   9
                04  000D4          .BYTE   4
                10  000D5          .BYTE   16
                00  000D6          .BYTE   0
                00  000D7          .BYTE   0
                00  000D8          .BYTE   0
              0369  000D9          .WORD   873
              0109  000DB          .WORD   265
                04  000DD          .BYTE   4
                10  000DE          .BYTE   16
                00  000DF          .BYTE   0
                0A  000E0          .BYTE   10
                00  000E1          .BYTE   0
              0369  000E2          .WORD   873
              0049  000E4          .WORD   73
                04  000E6          .BYTE   4
                10  000E7          .BYTE   16
```

```
                00  000E8          .BYTE   0
                00  000E9          .BYTE   0
                00  000EA          .BYTE   0
              0369  000EB          .WORD   873
              0149  000ED          .WORD   329
                04  000EF          .BYTE   4
                10  000F0          .BYTE   16
                00  000F1          .BYTE   0
                0A  000F2          .BYTE   10
                00  000F3          .BYTE   0
              0369  000F4          .WORD   873
              0228  000F6          .WORD   552
                01  000F8          .BYTE   1
                10  000F9          .BYTE   16
                00  000FA          .BYTE   0
                00  000FB          .BYTE   0
                00  000FC          .BYTE   0
              0369  000FD          .WORD   873
              0328  000FF          .WORD   808
                01  00101          .BYTE   1
                10  00102          .BYTE   16
                00  00103          .BYTE   0
                08  00104          .BYTE   8
                00  00105          .BYTE   0
              0369  00106          .WORD   873
              0268  00108          .WORD   616
                01  0010A          .BYTE   1
                10  0010B          .BYTE   16
                00  0010C          .BYTE   0
                00  0010D          .BYTE   0
                00  0010E          .BYTE   0
              0369  0010F          .WORD   873
              0368  00111          .WORD   872
                01  00113          .BYTE   1
                10  00114          .BYTE   16
                00  00115          .BYTE   0
                08  00116          .BYTE   8
                00  00117          .BYTE   0
              0369  00118          .WORD   873
              0208  0011A          .WORD   520
                02  0011C          .BYTE   2
                10  0011D          .BYTE   16
                00  0011E          .BYTE   0
                00  0011F          .BYTE   0
                00  00120          .BYTE   0
              0369  00121          .WORD   873
              0308  00123          .WORD   776
                02  00125          .BYTE   2
                10  00126          .BYTE   16
                00  00127          .BYTE   0
                0A  00128          .BYTE   10
                00  00129          .BYTE   0
              0369  0012A          .WORD   873
              0248  0012C          .WORD   584
                02  0012E          .BYTE   2
                10  0012F          .BYTE   16
                00  00130          .BYTE   0
```

```
                00131          .BYTE   0
                00132          .BYTE   0
0369            00133          .WORD   873
0348            00135          .WORD   840
                00137          .BYTE   2
                00138          .BYTE   16
                00139          .BYTE   0
                0013A          .BYTE   10
                0013B          .BYTE   0
0369            0013C          .WORD   873
0229            0013E          .WORD   553
                00140          .BYTE   3
                00141          .BYTE   16
                00142          .BYTE   0
                00143          .BYTE   0
                00144          .BYTE   0
0369            00145          .WORD   873
0329            00147          .WORD   809
                00149          .BYTE   3
                0014A          .BYTE   16
                0014B          .BYTE   0
                0014C          .BYTE   8
                0014D          .BYTE   0
0369            0014E          .WORD   873
0269            00150          .WORD   617
                00152          .BYTE   3
                00153          .BYTE   16
                00154          .BYTE   0
                00155          .BYTE   0
                00156          .BYTE   0
0369            00157          .WORD   873
0369            00159          .WORD   873
                0015B          .BYTE   3
                0015C          .BYTE   16
                0015D          .BYTE   0
                0015E          .BYTE   8
                0015F          .BYTE   0
0369            00160          .WORD   873
0209            00162          .WORD   521
                00164          .BYTE   4
                00165          .BYTE   16
                00166          .BYTE   0
                00167          .BYTE   0
                00168          .BYTE   0
0369            00169          .WORD   873
0309            0016B          .WORD   777
                0016D          .BYTE   4
                0016E          .BYTE   16
                0016F          .BYTE   0
                00170          .BYTE   10
                00171          .BYTE   0
0369            00172          .WORD   873
0249            00174          .WORD   585
                00176          .BYTE   4
                00177          .BYTE   16
                00178          .BYTE   0
                00179          .BYTE   0
```

```
                                    00  0017A          .BYTE   0
                                  0369  0017B          .WORD   873
                                  0349  0017D          .WORD   841
                                    04  0017F          .BYTE   4
                                    10  00180          .BYTE   16
                                    00  00181          .BYTE   0
                                    0A  00182          .BYTE   10
                                    00  00183          .BYTE   0
                                  0369  00184          .WORD   873
                                  0349  00186          .WORD   841
                                    04  00188          .BYTE   4
                                    10  00189          .BYTE   16
                                    00  0018A          .BYTE   0
                                    0A  0018B          .BYTE   10
                                    00  0018C          .BYTE   0
                                  0000  0018D          .WORD   0
                                  0008  0018F          .WORD   8
                                    FD  00191          .BYTE   -3
                                    0C  00192          .BYTE   12
                                    00  00193          .BYTE   0
                                    0C  00194          .BYTE   12
                                    00  00195          .BYTE   0
                                    0D  00196 P.AAA:   .BYTE   13
4E  4F  49  54  43  45  53  20  50  55  58  49  46  00197          .ASCII  \FIXUP SECTION\

                                                       .PSECT  $OWN$,NOEXE,2

                                        00000 DEFCLUBASED:
                                                       .BLKB   4
                                        00004 CURPSECT:
                                                       .BLKB   4
                                        00008 CURRENT_BASE:
                                                       .BLKB   4
                                        0000C ISECT_BASE:
                                                       .BLKB   4
                                    80000000  00010 CNTRL_REG_ADDR:
                                                       .LONG   -2147483648
                                        00014 ISECT:   .BLKB   1
                                        00015 NUMISECTS:
                                                       .BLKB   1
                                        00016          .BLKB   2
                                        00018 ISECTGENTBL:
                                                       .BLKB   4
                                        0001C PREVISECT:
                                                       .BLKB   4
                                        00020 ISCTDESC:
                                                       .BLKB   4

                                                       .PSECT  $GLOBAL$,NOEXE,2

                                        00000 LNK$GL_DMTBUFFER::
                                                       .BLKB   8
                                        00008 LNK$GL_DMTBYTES::
                                                       .BLKB   4
                                        0000C LNK$GL_LOWCLU::
                                                       .BLKB   4
                                        00010 LNK$GL_HIGHCLU::
```

```
                        .BLKB   4
00014 LNK$GL_LSTCLSTR::
                        .BLKB   4
00018 LNK$GL_FIXISD::
                        .BLKB   4
0001C LNK$GL_1STGADR::
                        .BLKB   4
00020 LNK$GL_LASTGADR::
                        .BLKB   4
00024 LNK$GW_SHRISCTS::
                        .BLKB   2
00026                   .BLKB   2
00028 LNK$GL_MAPLST::
                        .BLKB   4
0002C LNK$GW_STACK::
                        .BLKB   2
0002E LNK$GW_NISECTS::
                        .BLKB   2

            ISD$C_SIZE==            88
            HDR$K_FILLCHR==         255
            IHD$K_SHR==             2
            IHD$K_ACTIVOFF==        48
            IHD$K_SYMDBGOFF==       68
            IHD$K_IMGIDOFF==        88
            IHD$K_PATCHOFF==        168
            IHD$K_MAXLENGTH==       168
            HDR$K_MINFILL==         2
            DEF$C_STACK==           20
            DEF$C_BASE==            512
                        .EXTRN  LNK$GL_OMDDST, LNK$GW_PSCDST
                        .EXTRN  LNK$GL_SHRSYMS, LNK$GL_SHRIMGS
                        .EXTRN  LNK$GL_SHRCLSTRS
                        .EXTRN  LNK$AL_VALCTLTB
                        .EXTRN  LNK$AL_SYTBLFMT
                        .EXTRN  LNK$GL_TFRPSC, LNK$GL_TFRADR
                        .EXTRN  LNK$GL_DBGTFPS, LNK$GL_DBGTFR
                        .EXTRN  LNK$GL_CTLMSK, LNK$GL_MINVA
                        .EXTRN  LNK$GL_CLULST, LNK$GL_LASTCLU
                        .EXTRN  LNK$GL_CURCLU, LNK$GL_DEFCLU
                        .EXTRN  LNK$GL_FVMLST, CRF$INSRTKEY
                        .EXTRN  CRF$INSRTREF, LIB$INSERT_TREE
                        .EXTRN  LNK$ALLOBLK, LNK$ALLOVIRMEM
                        .EXTRN  LNK$FINDVIRMEM, LNK$FNDPSCMSK
                        .EXTRN  LNK$REQUESTMEM, LIN$_BASESYM
                        .EXTRN  LIN$_CLUBELBAS, LIN$_CONFMEM
                        .EXTRN  LIN$_INSVIRMEM, LIN$_MEMFUL
                        .EXTRN  LIN$_NOIMGFIL, LIN$_STKOVFLO

                        .PSECT  $CODE$,NOWRT,2

                0FFC 00000      .ENTRY  LNK$VMALLO, Save R2,R3,R4,R5,R6,R7,R8,R9,-  ; 0744
                                        R10,R11                                     ;
5B 00000000G   00  9E 00002     MOVAB   LIB$SIGNAL, R11                             ;
5A 00000000'   EF  9E 00009     MOVAB   EXE_ISGENTBL, R10                           ;
59 00000000'   EF  9E 00010     MOVAB   LNK$GL_FIXISD, R9                           ;
58 00000000G   00  9E 00017     MOVAB   LNK$GL_DEFCLU, R8                           ;
```

```
                                57 00000000'   EF  9E 0001E          MOVAB   DEFCLUBASED, R7                                  ;
                                56 00000000G   00  9E 00025          MOVAB   LNK$GL_CTLMSK, R6                               ;
                                55 00000000G   00  9E 0002C          MOVAB   LNK$GL_CURCLU, R5                               ;
                          0B                   66  01  E1 00033          BBC     #1, LNK$GL_CTLMSK, 1$                   ; 0763
                       18 A7                   6A  9E 00037          MOVAB   EXE_ISGENTBL, ISECTGENTBL                       ; 0765
                       15 A7            FC     AA  90 0003B          MOVB    EXE_ISECTS, NUMISECTS                           ; 0766
                                               1A  11 00040          BRB     3$                                              ; 0763
                          0C                   66  03  E1 00042 1$:      BBC     #3, LNK$GL_CTLMSK, 2$                   ; 0768
                       18 A7            54     AA  9E 00046          MOVAB   SYS_ISGENTBL, ISECTGENTBL                       ; 0770
                       15 A7            51     AA  90 0004B          MOVB    SYS_ISECTS, NUMISECTS                           ; 0771
                                               0A  11 00050          BRB     3$                                              ; 0768
                       18 A7            60     AA  9E 00052 2$:      MOVAB   SHR_ISGENTBL, ISECTGENTBL                       ; 0774
                       15 A7            5D     AA  90 00057          MOVB    SHR_ISECTS, NUMISECTS                           ; 0775
                67  58 A8                   01  00  EF 0005C 3$:      EXTZV   #0, #1, LNK$GL_DEFCLU+88, DEFCLUBASED           ; 0778
                                            F4     A9  7C 00062          CLRQ    LNK$GL_LOWCLU                           ; 0780
                                65 00000000G   00  9E 00065          MOVAB   LNK$GL_CLULST, LNK$GL_CURCLU                    ; 0781
                                50             65  D0 0006C 4$:      MOVL    LNK$GL_CURCLU, R0                               ; 0782
                                65             60  D0 0006F          MOVL    (R0), LNK$GL_CURCLU                             ;
                                               03  12 00072          BNEQ    5$                                              ;
                                             0115  31 00074          BRW     19$                                             ;
                                54             65  D0 00077 5$:      MOVL    LNK$GL_CURCLU, R4                               ; 0783
                          03    58 A4          02  E1 0007A          BBC     #2, 88(R4), 6$                                  ;
                                             00FA  31 0007F          BRW     17$                                             ;
                                03    58 A4          E8 00082 6$:      BLBS    88(R4), 7$                                      ; 0784
                                             00EE  31 00086          BRW     16$                                             ;
                       02 A6                   02  8A 00089 7$:      BICB2   #2, LNK$GL_CTLMSK+2                             ; 0786
                                52       4C     A4  D0 0008D          MOVL    76(R4), BASEADDRSYM                             ; 0787
                          75    58 A4                01  E1 00091          BBC     #1, 88(R4), 13$                         ; 0788
                                               73  13 00096          BEQL    13$                                             ; 0789
                                50       0F     A2  9A 00098          MOVZBL  15(BASEADDRSYM), R0                             ; 0791
                             50                52  C3 0009C          SUBL3   R0, BASEADDRSYM, R0                             ;
                                53       FB     A0  9E 000A0          MOVAB   -5(R0), BASESYMSNB                              ; 0792
                                50       0A     A2  3C 000A4          MOVZWL  10(BASEADDRSYM), R0                             ; 0793
                                50 FFFFFFF5     8F  CA 000A8          BICL2   #-11, R0                                        ;
                                02                 50  D1 000AF          CMPL    R0, #2                                  ; 0794
                                               24  13 000B2          BEQL    9$                                              ;
                                         04     A3  9F 000B4          PUSHAB  4(BASESYMSNB)                                   ; 0797
                                               01  DD 000B7          PUSHL   #1                                              ;
                                   00000000G   8F  DD 000B9          PUSHL   #LIN$_BASESYM                                   ;
                                6B             03  FB 000BF          CALLS   #3, LIB$SIGNAL                                  ;
                                50             65  D0 000C2          MOVL    LNK$GL_CURCLU, R0                               ; 0799
                          0A    66                03  E1 000C5          BBC     #3, LNK$GL_CTLMSK, 8$                   ; 0798
                       4C A0 80000000          8F  D0 000C9          MOVL    #-2147483648, 76(R0)                            ; 0799
                                               25  11 000D1          BRB     12$                                             ;
                                         4C    A0  D4 000D3 8$:      CLRL    76(R0)                                          ; 0800
                                               20  11 000D6          BRB     12$                                             ; 0793
                          05    66                03  E1 000D8 9$:      BBC     #3, LNK$GL_CTLMSK, 10$                  ; 0803
                          13 03 A6                01  E1 000DC          BBC     #1, LNK$GL_CTLMSK+3, 11$                ; 0804
                          50    62 000001FF       8F  C1 000E1 10$:     ADDL3   #511, (BASEADDRSYM), R0                 ; 0805
                    4C    A4    50 000001FF       8F  CB 000E9          BICL3   #511, R0, 76(R4)                        ; 0806
                                               04  11 000F2          BRB     12$                                             ; 0805
                       4C A4                   62  D0 000F4 11$:     MOVL    (BASEADDRSYM), 76(R4)                   ; 0807
                          0F    66                03  E1 000F8 12$:     BBC     #3, LNK$GL_CTLMSK, 13$                  ; 0809
                                51 00000000G   00  D0 000FC          MOVL    LNK$GL_FVMLST, R1                               ; 0810
                                50             65  D0 00103          MOVL    LNK$GL_CURCLU, R0                               ;
                       04 A1            4C     A0  D0 00106          MOVL    76(R0), 4(R1)                                   ;
                                               67  E9 0010B 13$:     BLBC    DEFCLUBASED, 14$                                ; 0812
```

```
                50                65  D0 0010E             MOVL    LNK$GL_CURCLU, R0                       ; 0813
          4C    A8          4C    A0  D1 00111             CMPL    76(R0), LNK$GL_DEFCLU+76
                                    17  1E C0116           BGEQU   14$
                            4C    A8  DD 00118             PUSHL   LNK$GL_DEFCLU+76                        ; 0817
                50                65  D0 0011B             MOVL    LNK$GL_CURCLU, R0                       ; 0816
                            5C    A0  9F 0011E             PUSHAB  92(R0)
                            4C    A0  DD 00121             PUSHL   76(R0)
                                      03  DD 00124         PUSHL   #3
                      00000000G     8F  DD 00126           PUSHL   #LIN$_CLUBELBAS
                6B                    05  FB 0012C         CALLS   #5, LIB$SIGNAL
    00000000V   EF                    00  FB 0012F 14$:    CALLS   #0, ALLOCLUSTER                         ; 0819
                50                65  D0 00136             MOVL    LNK$GL_CURCLU, R0                       ; 0821
                7E          4C    A0  7D 00139             MOVQ    76(R0), -(SP)                           ; 0820
    00000000G   00                02  FB 0013D             CALLS   #2, LNK$ALLOVIRMEM
                1E                    50  E8 00144         BLBS    R0, 15$
                66                    01  8A 00147         BICB2   #1, LNK$GL_CTLMSK                       ; 0823
                      00000000G     8F  DD 0014A           PUSHL   #LIN$_NOIMGFIL                          ; 0827
                50                65  D0 00150             MOVL    LNK$GL_CURCLU, R0
                            5C    A0  9F 00153             PUSHAB  92(R0)
                7E          4C    A0  7D 00156             MOVQ    76(R0), -(SP)
                                      03  DD 0015A         PUSHL   #3
                      00000000G     8F  DD 0015C           PUSHL   #LIN$_CONFMEM
                6B                    06  FB 00162         CALLS   #6, LIB$SIGNAL
                                      65  DD 00165 15$:     PUSHL   LNK$GL_CURCLU                           ; 0831
    00000000V   EF                    01  FB 00167         CALLS   #1, LNK$SETLASTCLU
    00000000V   EF                    00  FB 0016E         CALLS   #0, RELOCLUSTER                         ; 0832
                                      12  11 00175         BRB     18$                                     ; 0783
    0D    58    A4                    02  E1 00177 16$:    BBC     #2, 88(R4), 18$                         ; 0835
                09          58    A4  E9 0017C 17$:        BLBC    88(R4), 18$                             ; 0836
                                      54  DD 00180         PUSHL   R4                                      ; 0837
    00000000V   EF                    01  FB 00182         CALLS   #1, LNK$CLUVIRMEM
                                FEE0  31  00189 18$:        BRW     4$                                      ; 0783
                52 00000000G    00    D0  0018C 19$:        MOVL    LNK$GL_MINVA, BASEADDRSYM               ; 0843
    15          66                    02  E0 00193         BBS     #2, LNK$GL_CTLMSK, 20$                  ; 0845
                                      01  DD 00197         PUSHL   #1                                      ; 0846
                                      7E  D4 00199         CLRL    -(SP)
    00000000G   00                    02  FB 0019B         CALLS   #2, LNK$ALLOVIRMEM
                07                    50  E9 001A2         BLBC    R0, 20$
    00000000G   00                    52  D0 001A5         MOVL    BASEADDRSYM, LNK$GL_MINVA               ; 0847
                65 00000000G    00    9E  001AC 20$:        MOVAB   LNK$GL_CLULST, LNK$GL_CURCLU            ; 0849
                50                65  D0 001B3 21$:        MOVL    LNK$GL_CURCLU, R0                       ; 0851
                65                    60  D0 001B6         MOVL    (R0), LNK$GL_CURCLU
                                      1E  13 001B9         BEQL    22$
                50                65  D0 001BB             MOVL    LNK$GL_CURCLU, R0                       ; 0853
                F1          58    A0  E8 001BE             BLBS    88(R0), 21$
    EC    58    A0                    02  E0 001C2         BBS     #2, 88(R0), 21$                         ; 0854
    00000000V   EF                    00  FB 001C7         CALLS   #0, ALLOCLUSTER                         ; 0856
                                      65  DD 001CE         PUSHL   LNK$GL_CURCLU                           ; 0857
    00000000V   EF                    01  FB 001D0         CALLS   #1, LNK$CLUVIRMEM
                                      DA  11 001D7         BRB     21$                                     ; 0851
    51 00000000G    00                0C  C5 001D9 22$:     MULL3   #12, LNK$GL_OMDDST, R1                  ; 0863
                50 00000000G    00    3C  001E1             MOVZWL  LNK$GW_PSCDST, R0
          F0    A9                6140  7E  001E8         MOVAQ   (R1)[R0], LNK$GL_DMTBYTES
                                      30  15 001ED         BLEQ    23$
                            E8    A9  9F 001EF             PUSHAB  LNK$GL_DMTBUFFER                        ; 0864
    50    F0    A9 000001FF     8F  C9 001F2             BISL3   #511, LNK$GL_DMTBYTES, R0
                                      50  D6 001FB         INCL    R0
```

```
                        7E                      50 00000200   8F   C7 001FD          DIVL3   #512, R0, -(SP)                                 ;
                    00000000G                   00            02   FB 00205          CALLS   #2, LNK$REQUESTMEM                              ;
                                                10            50   E8 0020C          BLBS    STATUS, 23$                                     ;
                                                          00000000G  8F   DD 0020F          PUSHL   #LIN$_NOIMGFIL                          ; 0866
                                                                     7E   D4 00215          CLRL    -(SP)                                   ;
                                                                     50   DD 00217          PUSHL   STATUS                                  ;
                                                                6B        03   FB 00219          CALLS   #3, LIB$SIGNAL                          ;
                                                                66        01   8A 0021C          BICB2   #1, LNK$GL_CTLMSK                       ; 0867
                        51                                      66        03   E0 0021F 23$:     BBS     #3, LNK$GL_CTLMSK, 25$          ; 0882
                        4D                                      66        02   E0 00223          BBS     #2, LNK$GL_CTLMSK, 25$          ; 0883
                                                                4A        67   E9 00227          BLBC    DEFCLUBASED, 25$                ; 0884
                    00000000V                                   EF        00   FB 0022A          CALLS   #0, LNK$CREFIXISD               ; 0888
                        52                                      69        18   C1 00231          ADDL3   #24, LNK$GL_FIXISD, FIXISDHDR   ; 0889
                                                                7E          02   A2 3C 00235          MOVZWL  2(FIXISDHDR), -(SP)             ; 0890
                53          04                                  A2        15   00   EF 00239          EXTZV   #0, #21, 4(FIXISDHDR), R3       ;
                                                                53        53   09   78 0023F          ASHL    #9, R3, R3                      ;
                                                                          53   DD 00243          PUSHL   R3                              ;
                                                    00000000G   00            02   FB 00245          CALLS   #2, LNK$ALLOVIRMEM              ;
                                                                1E            50   E8 0024C          BLBS    R0, 24$                         ;
                                                                          00000000G  8F   DD 0024F          PUSHL   #LIN$_NOIMGFIL                  ; 0892
                                                                               0192  CA   9F 00255          PUSHAB  P.AAA                           ; 0894
                                                                7E          02   A2 3C 00259          MOVZWL  2(FIXISDHDR), -(SP)             ; 0893
                                                                          53   DD 0025D          PUSHL   R3                              ;
                                                                          03   DD 0025F          PUSHL   #3                              ; 0892
                                                                          00000000G  8F   DD 00261          PUSHL   #LIN$_CONFMEM                   ;
                                                                6B                   06   FB 00267          CALLS   #6, LIB$SIGNAL                  ;
                                                                66                   01   8A 0026A          BICB2   #1, LNK$GL_CTLMSK               ; 0895
                                                                50                   69   D0 0026D 24$:     MOVL    LNK$GL_FIXISD, R0               ; 0897
                                                    14          A0                   02   88 00270          BISB2   #2, 20(R0)                      ;
                                                                65        00000000G  00   9E 00274 25$:     MOVAB   LNK$GL_CLULST, LNK$GL_CURCLU    ; 0904
                                                                23                   67   E9 0027B          BLBC    DEFCLUBASED, 27$                ; 0906
                    1F                                          66                   03   E0 0027E          BBS     #3, LNK$GL_CTLMSK, 27$          ; 0907
                                                                50                   65   D0 00282 26$:     MOVL    LNK$GL_CURCLU, R0               ; 0908
                                                                65                   60   D0 00285          MOVL    (R0), LNK$GL_CURCLU             ;
                                                                                     17   13 00288          BEQL    27$                             ;
                                                                50                   65   D0 0028A          MOVL    LNK$GL_CURCLU, R0               ; 0910
                                                                F1                58   A0 E8 0028D          BLBS    88(R0), 26$                     ;
                    EC                  58                      A0                   02   E1 00291          BBC     #2, 88(R0), 26$                 ; 0911
                                                                50                   DD 00296          PUSHL   R0                              ; 0913
                        00000000V       EF                                           01   FB 00298          CALLS   #1, LNK$CLUVIRMEM               ;
                                                                                     E1   11 0029F          BRB     26$                             ; 0908
                                                                          F8         A9   D5 002A1 27$:     TSTL    LNK$GL_HIGHCLU                  ; 0921
                                                                                     37   12 002A4          BNEQ    30$                             ;
                                                                          08         A8   D5 002A6          TSTL    LNK$GL_DEFCLU+8                 ; 0923
                                                                                     1F   12 002A9          BNEQ    29$                             ;
                                                                50        00000000G  00   D0 002AB          MOVL    LNK$GL_LASTCLU, R0              ; 0925
                                                                60                   68   9E 002B2          MOVAB   LNK$GL_DEFCLU, (R0)             ;
                                                          04            A8           50   D0 002B5          MOVL    R0, LNR$GL_DEFCLU+4             ; 0926
                                                                                     08   12 002B9          BNEQ    28$                             ;
                                                          04            A8 00000000G 00   9E 002BB          MOVAB   LNK$GL_CLULST, LNK$GL_DEFCLU+4  ; 0927
                                                    00000000G           00           68   9E 002C3 28$:     MOVAB   LNK$GL_DEFCLU, LNK$GL_LASTCLU   ; 0928
                                                                                     01   DD 002CA 29$:     PUSHL   #1                              ; 0930
                                                                                     7E   D4 002CC          CLRL    -(SP)                           ;
                                                    00000000G           00           02   FB 002CE          CALLS   #2, LNK$ALLOVIRMEM              ;
                                                          F8            A9           68   9E 002D5          MOVAB   LNK$GL_DEFCLU, LNK$GL_HIGHCLU   ; 0931
                                                          F4            A9           68   9E 002D9          MOVAB   LNK$GL_DEFCLU, LNK$GL_LOWCLU    ; 0932
                                                                                     69   D5 002DD 30$.     TSTL    LNK$GL_FIXISD                   ; 0939
```

```
                                1A  12 002DF         BNEQ    33$                            ;
      0B          66            03  E0 002E1         BBS     #3, LNK$GL_CTLMSK, 31$         ; 0940
                     00000000G  00  D5 002E5         TSTL    LNK$GL_SHRCLSTRS               ; 0941
                                03  13 002EB         BEQL    31$                            ;
                  04            67  E9 002ED         BLBC    DEFCLUBASED, 32$               ; 0942
      07          66            01  E1 002F0  31$:   BBC     #1, LNK$GL_CTLMSK, 33$         ; 0943
         00000000V EF           00  FB 002F4  32$:   CALLS   #0, LNK$CREFIXISD              ; 0944
                  50 00000000G  00  D0 002FB  33$:   MOVL    LNK$GL_TFRPSC, R0              ; 0946
                                08  13 00302         BEQL    34$                            ;
         00000000G 00       18  A0  C0 00304         ADDL2   24(R0), LNK$GL_TFRADR          ; 0947
                  50 00000000G  00  D0 0030C  34$:   MOVL    LNK$GL_DBGTFPS, R0             ; 0948
                                0D  13 00313         BEQL    35$                            ;
                  01            50  D1 00315         CMPL    R0, #1                         ;
                                08  13 00318         BEQL    35$                            ;
         00000000G 00       18  A0  C0 0031A         ADDL2   24(R0), LNK$GL_DBGTFR          ; 0949
                                    04 00322  35$:   RET                                    ; 0951

; Routine Size:  803 bytes,    Routine Base:  $CODE$ + 0000
```

```
;  458          0952  1 routine allocluster : novalue =
;  459          0953  2 begin
;  460          0954  2 !
;  461          0955  2 !       THIS ROUTINE INDEXES DOWN THE I-SECTION GENERATION CONTROL
;  462          0956  2 !       TABLE, CALLING THE I-SECT SPECIFIC ROUTINE.
;  463          0957  2 !       FIRST HOWEVER ESTABLISH THE BASE ADDRESS IF IT WAS SYMBOLIC
;  464          0958  2 !
;  465          0959  2 isect_base = .lnk$gl_curclu[clu$l_base];                 ! START AT BASE DETERMINED BY COMMAND
;  466          0960  2 isctdesc = lnk$gl_curclu[clu$l_fstisd];                  ! INITIALIZE TO LISTHEAD FOR ISECTIONS
;  467          0961  2 previsect = lnk$gl_curclu[clu$l_fstisd];
;  468          0962  2 !
;  469          0963  2 incr iloop from 1 to .numisects do                      ! INCREMENT THROUGH GENERATION
;  470          0964  3 begin                                                   ! CONTROL TABLE
;  471          0965  3         isect = .iloop;                                 ! REMEMBER INDEX
;  472          0966  4         if not (.lnk$gl_ctlmsk[lnk$v_sys]               ! UNLESS A SYSTEM IMAGE WITHOUT A HEADER
;  473          0967  4                 and not .lnk$gl_ctlmsk[lnk$v_sysheadr])
;  474          0968  3             then isect_base = (.isect_base+511) and not 511;    ! ROUND UP NEXT PAGE
;  475          0969  3         if .isectgentbl[.isect-1,isc$b_code] eql stack_code     ! CALL THE BUILD ROUTINE BASED ON
;  476          0970  3         then stack_isectbld()                           ! THE KIND OF ISECT .. STACK
;  477          0971  3         else normal_isectbld();                         ! AND ALL OTHERS
;  478          0972  2 end;
;  479          0973  2 if .isctdesc[isl$l_nxtisd] eql -1                       ! IF AN ALLOCATED ISD IS UNUSED
;  480          0974  3 then begin
;  481          0975  3         isctdesc = .lnk$gl_curclu[clu$l_lstisd];        ! THEN DISCONNECT IT FROM LIST
;  482          0976  3         isctdesc[isl$l_nxtisd] = 0;
;  483          0977  2     end;
;  484          0978  2 return;
;  485          0979  1 end;


                                    003C 00000 ALLOCLUSTER:
                                                .WORD   Save R2,R3,R4,R5                ; 0952
                        55 00000000G    00  9E 00002        MOVAB   LNK$GL_CURCLU, R5
                        54 00000000'    EF  9E 00009        MOVAB   ISCTDESC, R4
                        50              65  D0 00010        MOVL    LNK$GL_CURCLU, R0   ; 0959
                    EC  A4      4C      A0  D0 00013        MOVL    76(R0), ISECT_BASE
                        50              18  C0 00018        ADDL2   #24, R0             ; 0960
                        64              50  D0 0001B        MOVL    R0, ISCTDESC
                    FC  A4              50  D0 0001E        MOVL    R0, PREVISECT       ; 0961
                        53      F5      A4  9A 00022        MOVZBL  NUMISECTS, R3       ; 0963
                                        52  D4 00026        CLRL    ILOOP
                                        48  11 00028        BRB     5$
                    F4  A4              52  90 0002A 1$:    MOVB    ILOOP, ISECT        ; 0965
                08 00000000G    00      03  E1 0002E        BBC     #3, LNK$GL_CTLMSK, 2$   ; 0966
                12 00000000G    00      01  E1 00036        BBC     #1, LNK$GL_CTLMSK+3, 3$ ; 0967
                50      EC      A4  000001FF    8F  C1 0003E 2$:    ADDL3   #511, ISECT_BASE, R0    ; 0968
            EC  A4              50  000001FF    8F  CB 00047        BICL3   #511, R0, ISECT_BASE
                        50      F4      A4  9A 00050 3$:    MOVZBL  ISECT, R0           ; 0969
                        50              09  C4 00054        MULL2   #9, R0
                        50      F8      A4  C0 00057        ADDL2   ISECTGENTBL, R0
                    FD  8F      FB      A0  91 0005B        CMPB    -5(R0), #253
                                        09  12 00060        BNEQ    4$
                00000000V   EF              00  FB 00062        CALLS   #0, STACK_ISECTBLD  ; 0970
                                        07  11 00069        BRB     5$
```

```
                00000000V EF              00 FB 0006B 4$:     CALLS   #0, NORMAL_ISECTBLD                 ; 0971
            B4          52                53 F3 000 2 5$:     AOBLEQ  R3, ILOOP, 1$                       ; 0963
                FFFFFFFF  8F        00    B4 D1 00076         CMPL    @ISCTDESC, #-1                      ; 0973
                                          0A 12 0007E         BNEQ    6$                                  ;
                          50              65 D0 00080         MOVL    LNK$GL_CURCLU, R0                   ; 0975
                          64        1C    A0 D0 00083         MOVL    28(R0), ISCTDESC                    ;
                                    00    B4 D4 00087         CLRL    @ISCTDESC                           ; 0976
                                          04 0008A 6$:        RET                                         ; 0979

; Routine Size:  139 bytes,    Routine Base:  $CODE$ + 0323
```

```
;      0980  1 routine compare_bases (psectbase, currentnode, psectdesc) =
;      0981  2 begin
;      0982  2 !
;      0983  2 ! COMPARE PSECT BASE WITH BASE OF CURRENT NODE
;      0984  2 !
;      0985  2 map
;      0986  2     currentnode : ref block[,byte],
;      0987  2     psectdesc : ref block[,byte];
;      0988  2
;      0989  2 local
;      0990  2     pdesc : ref block[,byte];
;      0991  2
;      0992  2 pdesc = .currentnode[node$l_ptr];                        ! GET POINTER TO PSECT DESCRIPTOR
;      0993  2 if .psectbase lssu .pdesc[psc$l_base]
;      0994  2     then return -1
;      0995  2     else if .psectbase eql .pdesc[psc$l_base]
;      0996  3         then begin
;      0997  3             if .psectdesc[psc$l_length]                 ! BASES THE SAME...DOES CURRENT NODE EXTEND FARTHER?
;      0998  3                     lssu .pdesc[psc$l_length]
;      0999  3                 then return -1
;      1000  3                 else return ch$compare(.psectdesc[psc$b_namlng],! IF THEY ARE EQUAL COMPARE THE NAMES
;      1001  3                                 psectdesc[psc$t_name],
;      1002  3                                 .pdesc[psc$b_namlng],
;      1003  3                                 pdesc[psc$t_name]);
;      1004  3             end
;      1005  2         else return 1
;      1006  1 end;


                                  001C 00000 COMPARE_BASES:
                                                  .WORD   Save R2,R3,R4                               ; 0980
                  50       08  AC  D0 00002        MOVL    CURRENTNODE, R0                             ; 0992
                  51       0A  A0  D0 00006        MOVL    10(R0), PDESC                               ;
             18   A1       04  AC  D1 0000A        CMPL    PSECTBASE, 24(PDESC)                        ; 0993
                               05  1E 0000F        BGEQU   1$                                          ;
                  54           01  CE 00011        MNEGL   #1, R4                                      ; 0995
                               35  11 00014        BRB     6$                                          ;
                               2D  12 00016 1$:    BNEQ    4$                                          ;
                  50       0C  AC  D0 00018        MOVL    PSECTDESC, R0                               ; 0997
             1C   A1       1C  A0  D1 0001C        CMPL    28(R0), 28(PDESC)                           ; 0998
                               05  1E 00021        BGEQU   2$                                          ;
                  50           01  CE 00023        MNEGL   #1, R0                                      ; 1000
                               20  11 00026        BRB     5$                                          ;
                  53       2D  A0  9A 00028 2$:    MOVZBL  45(R0), R3                                  ;
                  52       2D  A1  9A 0002C        MOVZBL  45(PDESC), R2                               ; 1002
                  54           01  D0 00030        MOVL    #1, R4                                      ; 1003
    52        00       2E   A0     53  2D 00033        CMPC5   R3, 46(R0), #0, R2, 46(PDESC)               ;
                           2E  A1     00039                                                            ;
                               03  1A 0003B        BGTRU   3$                                          ;
                  54           01  D9 0003D        SBWC    #1, R4                                      ;
                  50           54  D0 00040 3$:    MOVL    R4, R0                                      ; 1000
                               03  11 00043        BRB     5$                                          ;
                  50           01  DC 00045 4$:    MOVL    #1, R0                                      ; 1005
                  54           50  D0 00048 5$:    MOVL    R0, R4                                      ; 0995
```

```
                        50          54  D0 0004B 6$:    MOVL    R4, R0
                                        04 0004E        RET

; Routine Size:  79 bytes,    Routine Base:  $CODE$ + 03AE
```

; 1006

```
;  515          1007  1 routine alloc_node (keyvalue, retadr, psectdesc) =
;  516          1008  2 begin
;  517          1009  2 !
;  518          1010  2 ! ALLOCATE A NODE AND POINT TO CURRENT PSECT DESCRIPTOR
;  519          1011  2 !
;  520          1012  2 local
;  521          1013  2     blockaddr : ref block[,byte];
;  522          1014  2
;  523          1015  2 lnk$alloblk(node$c_long,blockaddr);
;  524          1016  2 blockaddr[node$l_ptr] = .psectdesc;
;  525          1017  2 .retadr = .blockaddr;
;  526          1018  2 return true
;  527          1019  1 end;


                                      0000 00000 ALLOC_NODE:
                                                       .WORD   Save nothing                          ; 1007
                  5E               04  C2 00002        SUBL2   #4, SP
                                   5E  DD 00005        PUSHL   SP                                    ; 1015
                                   0E  DD 00007        PUSHL   #14
          00000000G   00           02  FB 00009        CALLS   #2, LNK$ALLOBLK
                      50           6E  D0 00010        MOVL    BLOCKADDR, R0                         ; 1016
              0A      A0      0C   AC  D0 00013        MOVL    PSECTDESC, 10(R0)
              08      BC           50  D0 00018        MOVL    R0, @RETADR                           ; 1017
                      50           01  D0 0001C        MOVL    #1, R0                                ; 1018
                                       04 0001F        RET                                           ; 1019

; Routine Size:  32 bytes,    Routine Base:  $CODE$ + 03FD
```

```
;  529     1020  1 routine relocluster : novalue =
;  530     1021  2 begin
;  531     1022  2 !
;  532     1023  2 !       THIS ROUTINE RELOCATES THE CONTENTS OF THE CURRENT CLUSTER. THIS MEANS
;  533     1024  2 !       GOING DOWN EACH OF THE I-SECTION AND P-SECTION LISTS AND ADDING
;  534     1025  2 !       THE CLUSTER BASE TO THEIR BASES. IN ADDITION, FOR EACH P-SECTION,
;  535     1026  2 !       THE LIST OF SYMBOLS IS SCANNED AND THOSE THAT ARE RELOCATABLE ARE
;  536     1027  2 !       ALSO RELOCATED BY THE CLUSTER BASE ADDRESS. IN THE PROCESS OF DOING THIS
;  537     1028  2 !       EACH P-SECTION DESCRIPTOR IS REMOVE FROM THE CLUSTER LIST AND PUT
;  538     1029  2 !       ON THE MAPPING LIST IN VIRTUAL ADDRESS ORDER FOR LATER MAP
;  539     1030  2 !       OUTPUT.
;  540     1031  2 !
;  541     1032  2 !               IF THE CURRENT CLUSTER HAS ALREADY BEEN BASED, HOWEVER, THE
;  542     1033  2 !       RELOCATION IS NOT DONE - THE P-SECTIONS ARE MERELY MOVED TO THE
;  543     1034  2 !       MAPPING LIST.
;  544     1035  2 !
;  545     1036  2 routine relocpsects (psectdesc) =
;  546     1037  3 begin
;  547     1038  3 !
;  548     1039  3 ! THIS LOCAL ROUTINE RELOCATES THE PSECTS ON THE SPECIFIED PSECT
;  549     1040  3 ! LIST AND TRANSFERS THEM TO THE MAPPING LIST.
;  550     1041  3 !
;  551     1042  3 map
;  552     1043  3         psectdesc : ref block[,byte];
;  553     1044  3
;  554     1045  3 local
;  555     1046  3         blockaddr,
;  556     1047  3         symsnb : ref block[,byte],
;  557     1048  3         symbol : ref block[,byte];
;  558     1049  3
;  559     1050  3 if .psectdesc neq 0
;  560     1051  3     then return true;
;  561     1052  3 if .psectdesc[psc$l_left] neq 0
;  562     1053  3     then relocpsects(.psectdesc[psc$l_left]);
;  563     1054  3 if not .psectdesc[psc$v_deleted]                                 ! IF PSECT IS NOT DELETED
;  564     1055  4     and (.psectdesc[psc$l_isect] neq 0                          ! IF P-SECTION ALLOCATED TO MEMORY
;  565     1056  4     or (.psectdesc[psc$w_flags] and (gps$m_lib or gps$m_gbl)) neq 0   ! OR IS GLOBAL OR IN SHAREABLE
;  566     1057  4     or .lnk$gl_ctlmsk[lnk$v_shr]                                ! OR A SHAREABLE OR
;  567     1058  4     or .lnk$gl_ctlmsk[lnk$v_dbg]                                ! DEBUGGABLE IMAGE
;  568     1059  4     or .lnk$gl_ctlmsk[lnk$v_symtbl])                            ! OR CREATING A SYMBOL TABLE
;  569     1060  4     then begin
;  570     1061  4         if .psectdesc[psc$v_rel]                                ! AND THIS P-SECTION IS RELOCATABLE
;  571     1062  4             and not .lnk$gl_curclu[clu$v_based]                 ! AND CLUSTER IS NOT BASED
;  572     1063  5             or (.lnk$gl_curclu[clu$v_shrimg] and .lnk$gl_curclu[clu$v_pic])
;  573     1064  4             then psectdesc[psc$l_base] = .psectdesc[psc$l_base] +
;  574     1065  4                                          .lnk$gl_curclu[clu$l_base];   ! ADD CLUSTER BASE TO ITS BASE
;  575     1066  4         symbol = .psectdesc[psc$l_symlst];                      ! GO DOWN LIST OF
;  576     1067  4         while .symbol neq 0                                     ! SYMBOLS OWNED BY
;  577     1068  5         do begin                                                ! P-SECTION AND ADD ITS
;  578     1069  5                 if .symbol[sym$v_rel]                           ! BASE TO ANY THAT
;  579     1070  5 !                   OR (.LNK$GL_CURCLU[CLU$V_SHRIMG] AND .LNK$GL_CURCLU[CLU$V_PIC])
;  580     1071  6                 then begin                                      ! ARE RELOCATABLE
;  581     1072  6                         bind
;  582     1073  6                             symsnb = .symbol - .symbol[sym$b_namlng]    ! POINT TO NAME BLOCK
;  583     1074  6                                              - snb$c_fxdlen : block[,byte];
;  584     1075  6                         if .lnk$gl_curclu[clu$v_shrimg]         ! IF THIS IS SHAREABLE IMAGE
;  585     1076  6                         then symbol[sym$v_rel] = false;         !  IT'S NOT RELOCATABLE ANYMORE
```

```
;  586          1077  6                        if .symbol[sym$v_gref]                                  ! IF SYMBOL WAS IN PIC, NON-BASED IM
;  587          1078  6                            then symbol[sym$l_value] = .symbol[sym$l_offset]
;  588          1079  6                                                     + .lnk$gl_curclu[clu$l_base]
;  589          1080  6                            else symbol[sym$l_value] = .symbol[sym$l_value]
;  590          1081  6                                                     + .psectdesc[psc$l_base];
;  591          1082  6                        if .lnk$gl_ctlmsk[lnk$v_long]                           ! IF A LONG FORM MAP IS REQUIRED
;  592          1083  6                            and .symbol[sym$v_crosref]                          ! AND SYMBOL HAS BEEN CROSS REFERENC
;  593          1084  6                            and not .symbol[sym$v_supres]                       ! AND THE SYMBOL IS NOT  SUPPRESSED
;  594          1085  6                            and not .symbol[sym$v_lclsym]                       ! AND IT'S NOT A LOCAL SYMBOL
;  595          1086  7                        then begin
;  596          1087  7                            crf$insrtkey(lnk$al_sytblfmt,symsnb[snb$b_namlng],
;  597          1088  7                                         symbol[sym$l_value],
;  598          1089  7                                         .symbol[sym$w_flags]);
;  599          1090  7                            crf$insrtref(lnk$al_valctlb,                        ! THEN INSERT THIS SYMBOL
;  600          1091  7                                symbol[sym$l_va[..],symsnb[snb$b_namlng], ! AS A REFERENCE TO ITS
;  601          1092  7                                .symbol[sym$w_flags],0);                        ! OWN VALUE RELOCATED
;  602          1093  6                            end;
;  603          1094  5                    end;
;  604          1095  5                symbol = .symbol[sym$l_psclst];                                 ! MOVE ON TO NEXT
;  605          1096  4            end;
;  606          1097  4            lib$insert_tree(lnk$gl_maplst,.psectdesc[psc$l_base],%ref(1),   !  INSERT NODE INTO VA
;  607          1098  4                            compare_bases, alloc_node, blockaddr,.psectdesc);! ORDERED TREE (PSECT MAPPING LIST)
;  608          1099  4                                                                            ! (IF ABS, SUPRESSED IN MAP, BUT NEE
;  609          1100  4                                                                            ! IN LIST FOR SYMBOL TABLE FILE OUTP
;  610          1101  3    end;
;  611          1102  3
;  612          1103  3 if .psectdesc[psc$l_right] neq 0                                            ! PROCESS RIGHT SUBTREE IF PRESENT
;  613          1104  3     then relocpsects(.psectdesc[psc$l_right]);
;  614          1105  3
;  615          1106  3 return true
;  616          1107  2 end;                                                                         ! OF RELOCPSECTS


                                01FC 00000 RELOCPSECTS:
                                           .WORD   Save R2,R3,R4,R5,R6,R7,R8                        ; 1036
                            58      FB  AF 9E 00002        MOVAB   RELOCPSECTS, R8
                            57 00000000G 00 9E 00006       MOVAB   LNK$GL_CURCLU, R7
                            56 00000000G 00 9E 0000D       MOVAB   LNK$GL_CTLMSK, R6
                            5E           08 C2 00014       SUBL2   #8, SP
                            53      04   AC D0 00017       MOVL    PSECTDESC, R3                    ; 1050
                                         03 12 0001B       BNEQ    1$
                                    00EF 31 0001D          BRW     16$
                                         63 D5 00020 1$:   TSTL    (R3)                             ; 1052
                                         05 13 00022       BEQL    2$
                                         63 DD 00024       PUSHL   (R3)                             ; 1053
                            68           01 FB 00026       CALLS   #1, RELOCPSECTS
                     03     0B   A3      06 E1 00029 2$:   BBC     #6, 11(R3), 4$                   ; 1054
                                    00D3 31 0002E 3$:      BRW     15$
                                20   A3 D5 00031 4$:       TSTL    32(R3)                           ; 1055
                                         13 12 00034       BNEQ    5$
                            12      0A   A3 93 00036       BITB    10(R3), #18                      ; 1056
                                         0D 12 0003A       BNEQ    5$
                     09     66           02 E0 0003C       BBS     #2, LNK$GL_CTLMSK, 5$            ; 1057
                     05     66           06 E0 00040       BBS     #6, LNK$GL_CTLMSK, 5$            ; 1058
```

```
                E5      01      A6              05  E1  00044           BBC     #5, LNK$GL_CTLMSK+1, 3$                 ; 1059
                07      0A      A3              03  E1  00049 5$:       BBC     #3, 10(R3), 6$                          ; 1061
                                50              67  D0  0004E           MOVL    LNK$GL_CURCLU, R0                       ; 1062
                                0D      58      A0  E9  00051           BLBC    88(R0), 7$                              ;
                                50              67  D0  00055 6$:       MOVL    LNK$GL_CURCLU, R0                       ; 1063
                0D      58      A0              02  E1  00058           BBC     #2, 88(R0), 8$                          ;
                08      58      A0              03  E1  0005D           BBC     #3, 88(R0), 8$                          ;
                                50              67  D0  00062 7$:       MOVL    LNK$GL_CURCLU, R0                       ; 1065
                        18      A3      4C      A0  C0  00065           ADDL2   76(R0), 24(R3)                          ;
                                52      14      A3  D0  0006A 8$:       MOVL    20(R3), SYMBOL                          ; 1066
                                                72  13  0006E 9$:       BEQL    14$                                     ; 1067
                                55      0A      A2  9E  00070           MOVAB   10(SYMBOL), R5                          ; 1069
                64              65              03  E1  00074           BBC     #3, (R5), 13$                           ;
                                50      0F      A2  9A  00078           MOVZBL  15(SYMBOL), R0                          ; 1073
                50              52              50  C3  0007C           SUBL3   R0, SYMBOL, R0                          ;
                                54      FB      A0  9E  00080           MOVAB   -5(R0), R4                              ; 1074
                                50              67  D0  00084           MOVL    LNK$GL_CURCLU, R0                       ; 1075
                03      58      A0              02  E1  00087           BBC     #2, 88(R0), 10$                         ;
                                65              08  8A  0008C           BICB2   #8, (R5)                                ; 1076
                08              65              0E  E1  0008F 10$:      BBC     #14, (R5), 11$                          ; 1077
                62      10      A2      4C      A0  C1  00093           ADDL3   76(R0), 16(SYMBOL), (SYMBOL)            ; 1079
                                                04  11  00099           BRB     12$                                     ; 1078
                                62      18      A3  C0  0009B 11$:      ADDL2   24(R3), (SYMBOL)                        ; 1081
                                39      01      A6  E9  0009F 12$:      BLBC    LNK$GL_CTLMSK+1, 13$                    ; 1082
                34      0C      A2              01  E1  000A3           BBC     #1, 12(SYMBOL), 13$                     ; 1083
                30              65              0D  E0  000A8           BBS     #13, (R5), 13$                          ; 1084
                                2C      01      A5  E8  000AC           BLBS    1(R5), 13$                              ; 1085
                                7E              65  3C  000B0           MOVZWL  (R5), -(SP)                             ; 1089
                                                52  DD  000B3           PUSHL   SYMBOL                                  ; 1088
                                        04      A4  9F  000B5           PUSHAB  4(R4)                                   ; 1087
                                00000000G       00  9F  000B8           PUSHAB  LNK$AL_SYTBLFMT                         ;
                    00000000G   00                  04  FB  000BE           CALLS   #4, CRF$INSRTKEY                    ; 1088
                                7E                  7E  D4  000C5           CLRL    -(SP)                               ; 1091
                                        7E          65  3C  000C7           MOVZWL  (R5), -(SP)                         ; 1092
                                                04  A4  9F  000CA           PUSHAB  4(R4)                               ; 1091
                                                52  DD  000CD           PUSHL   SYMBOL                                  ;
                                00000000G       00  9F  000CF           PUSHAB  LNK$AL_VALCTLTB                         ; 1090
                    00000000G   00                  05  FB  000D5           CALLS   #5, CRF$INSRTREF                    ; 1091
                                52      04          A2  D0  000DC 13$:      MOVL    4(SYMBOL), SYMBOL                   ; 1095
                                                    8C  11  000E0           BRB     9$                                  ; 1067
                                                    53  DD  000E2 14$:      PUSHL   R3                                  ; 1098
                                                08  AE  9F  000E4           PUSHAB  BLOCKADDR                           ; 1097
                                                E0  A8  9F  000E7           PUSHAB  ALLOC_NODE                          ;
                                                91  A8  9F  000EA           PUSHAB  COMPARE_BASES                       ;
                                        10      AE      01  D0  000ED           MOVL    #1, 16(SP)                      ;
                                                10  AE  9F  000F1           PUSHAB  16(SP)                              ;
                                                18  A3  DD  000F4           PUSHL   24(R3)                              ;
                                00000000'       EF  9F  000F7           PUSHAB  LNK$GL_MAPLST                       ;
                    00000000G   00                  07  FB  000FD           CALLS   #7, LIB$INSERT_TREE                 ;
                                                04  A3  D5  00104 15$:      TSTL    4(R3)                               ; 1103
                                                06  13  00107           BEQL    16$                                 ;
                                                04  A3  DD  00109           PUSHL   4(R3)                               ; 1104
                                        68              01  FB  0010C           CALLS   #1, RELOCPSECTS                 ;
                                        50              01  D0  0010F 16$:      MOVL    #1, R0                          ; 1106
                                                            04  00112           RET                                     ; 1107

; Routine Size:  275 bytes,    Routine Base:  $CODE$ + 041D
```

```
;  617          1108  2 !
;  618          1109  2 ! MAIN BODY OF RELOCLUSTER
;  619          1110  2 !
;  620          1111  2
;  621          1112  2 if not .lnk$gl_curclu[clu$v_shrimg]
;  622          1113  3 then begin
;  623          1114  4     if (.lnk$gl_highclu eql 0                                  ! IF NO HIGHEST CLUSTER
;  624          1115  4         or .lnk$gl_curclu[clu$l_base] gtru .lnk$gl_highclu[clu$l_base])! OR THIS IS NEW HIGHEST
;  625          1116  3         then lnk$gl_highclu = .lnk$gl_curclu;                  ! THEN MAKE IT NEW HIGHEST CLUSTER
;  626          1117  4     if (.lnk$gl_lowclu eql 0                                   ! IF NO LOWEST CLUSTER
;  627          1118  4         or .lnk$gl_curclu[clu$l_base] lssu .lnk$gl_lowclu[clu$l_base]) ! OR THIS ONE IS LOWER
;  628          1119  3         then lnk$gl_lowclu = .lnk$gl_curclu;
;  629          1120  2     end;
;  630          1121  2 !
;  631          1122  2 ! RELOCATE THE PSECTS IN THIS CLUSTER
;  632          1123  2 !
;  633          1124  2 relocpsects(.lnk$gl_curclu[clu$l_lpslst]);
;  634          1125  2
;  635          1126  2 return true
;  636          1127  1 end;                                                          ! OF RELOCLUSTER


                                0004 00000 RELOCLUSTER:
                                                .WORD   Save R2                                 ; 1020
                          52 00000000'  EF  9E 00002            MOVAB   LNK$GL_HIGHCLU, R2
                          50 00000000G  00  D0 00009            MOVL    LNK$GL_CURCLU, R0       ; 1112
                    20    58    A0      02  E0 00010            BBS     #2, 88(R0), 4$
                          51            62  D0 00015            MOVL    LNK$GL_HIGHCLU, R1      ; 1114
                                        07  13 00018            BEQL    1$
                          4C    A1    4C  A0 D1 0001A           CMPL    76(R0), 76(R1)          ; 1115
                                        03  1B 0001F            BLEQU   2$
                          62            50  D0 00021 1$:       MOVL    R0, LNK$GL_HIGHCLU      ; 1116
                          51        FC  A2  D0 00024 2$:       MOVL    LNK$GL_LOWCLU, R1       ; 1117
                                        07  13 00028            BEQL    3$
                          4C    A1    4C  A0 D1 0002A           CMPL    76(R0), 76(R1)          ; 1118
                                        04  1E 0002F            BGEQU   4$
                          FC    A2          50  D0 00031 3$:    MOVL    R0, LNK$GL_LOWCLU       ; 1119
                                    10      A0  DD 00035 4$:    PUSHL   16(R0)                  ; 1124
                        FEB0    CF          01  FB 00038        CALLS   #1, RELOCPSECTS
                                            04  0003D          RET                             ; 1127

; Routine Size:  62 bytes,    Routine Base:  $CODE$ + 0530
```

```
;  638          1128  1 global routine lnk$crefixisd =
;  639          1129  2 begin
;  640          1130  2 !
;  641          1131  2 ! THIS ROUTINE CREATES THE FIXUP IMAGE SECTION AT THE END OF THE HIGHEST
;  642          1132  2 ! USER CLUSTER
;  643          1133  2 !
;  644          1134  2
;  645          1135  2     local
;  646          1136  2         lastisd : ref block[,byte],
;  647          1137  2         lasthdrisd : ref block[,byte],
;  648          1138  2         stackisd : ref block[,byte],
;  649          1139  2         symbolblock : ref block[,byte],
;  650          1140  2         symbolsnb : ref block[,byte],
;  651          1141  2         hdrisd : ref block[,byte],
;  652          1142  2         fixadr,
;  653          1143  2         savecurclu,
;  654          1144  2         tempisd : block[isl$c_size+isd$c_maxlenglbl,byte];
;  655          1145  2
;  656          1146  2     lastisd = stackisd = .lnk$gl_highclu[clu$l_lstisd];          ! POINT TO LAST ISD IN CLUSTER
;  657          1147  2     lasthdrisd = lastisd[isl$t_hdrisd];                          ! POINT TO PART OF LAST ISD BOUND FO
;  658          1148  2     if .lastisd eql lnk$gl_highclu[clu$l_fstisd]                 ! IF CLUSTER HAS NO IMAGE SECTIONS
;  659          1149  3     then begin
;  660          1150  3         ch$fill(0,isl$c_size+isd$c_maxlenglbl,tempisd);          ! ZERO TEMPORARY ISECT
;  661          1151  3         lasthdrisd = tempisd[isl$t_hdrisd];                      ! USE IT FOR THE PAGE COUNT
;  662          1152  3         lasthdrisd[isd$w_pagcnt] = 1;                            ! SET ONE PAGE (SO FIXUP SECTION NOT
;  663          1153  2         end;
;  664          1154  2     if .lasthdrisd[isd$b_type] neq isd$k_usrstack                ! IF THE LAST SECTION IS NOT A STACK
;  665          1155  2         then stackisd = 0                                        ! THEN REMEMBER THAT
;  666          1156  3         else begin
;  667          1157  3             lastisd = .lastisd[isl$l_previsd];                   ! POINT TO ONE JUST BEFORE STACK ISD
;  668          1158  3             lasthdrisd = lastisd[isl$t_hdrisd];                  ! AND IT'S HEADER PART
;  669          1159  2             end;
;  670          1160  2     savecurclu = .lnk$gl_curclu;                                 ! SAVE CURRENT CLUSTER DESCRIPTOR PO
;  671          1161  2     lnk$gl_curclu = .lnk$gl_highclu;                             ! MAKE HIGHEST CLUSTER THE CURRENT C
;  672          1162  3     isect_base = fixadr = (.lasthdrisd[isd$v_vpg]^9)             ! SET BASE OF ISECT WE WILL CREATE
;  673          1163  2                                 +(.lasthdrisd[isd$w_pagcnt]*512);
;  674          1164  2     if .isect_base eql 0                                         ! IF BASE IS 0, MAKE IT X'200', THIS
;  675          1165  2         then isect_base = %x'200';                               !  NULL.
;  676          1166  2     lnk$alloblk(isd$c_size,isctdesc);                            ! ALLOCATE ISECT DESCRIPTOR
;  677          1167  2     lnk$gl_fixisd = lastisd[isl$l_nxtisd] = .isctdesc;           ! LINK INTO ISD LIST AND REMEMBER DE
;  678          1168  2     if .lnk$gl_ctlmsk[lnk$v_shr]                                 ! SET THE CORRECT INDEX INTO ISGENTB
;  679          1169  2         then isect = 6                                           ! (IF A SHAREABLE IMAGE)
;  680          1170  2         else isect = 2;                                          ! (IF AN EXECUTABLE IMAGE)
;  681          1171  2     previsect = .lastisd;                                        ! POINT TO PREVIOUS ISECT DESCRIPTOR
;  682          1172  4     buildisd(((((.lnk$gl_shrimgs*2)+.lnk$gl_shrsyms+1)*4)        ! FILL IN AND LINK INTO CURRENT CLUS
;  683          1173  2                                         +iaf$c_length+511)/512);
;  684          1174  2     if .stackisd neq 0
;  685          1175  3     then begin
;  686          1176  3         isctdesc[isl$l_nxtisd] = .stackisd;                      ! PUT STACK ISD BACK IN IF IT WAS TH
;  687          1177  3         stackisd[isl$l_previsd] = .isctdesc;
;  688          1178  3         lnk$gl_curclu[clu$l_lstisd] = .stackisd;
;  689          1179  2         end;
;  690          1180  2     hdrisd = isctdesc[isl$t_hdrisd];                             ! POINT TO PART IN HEADER
;  691          1181  2     fixadr = .fixadr + iaf$c_length;                             ! POINT TO START OF FIXUP AREA (SKIP
;  692          1182  2     !
;  693          1183  2     ! NOW LOOP THROUGH ALL THE CLUSTERS, AND FOR EACH CLUSTER THAT HAS SYMBOLS
;  694          1184  2     ! IN THE SHR LST, REDEFINE THE SYMBOL AT THE PROPER LOCATION IN THE FIXUP
```

```
;  695          1185  2      ! AREA.
;  696          1186  2      !
;  697          1187  2      lnk$gl_curclu = lnk$gl_clulst[0];                                   ! START AT TOP OF CLUSTER LIST
;  698          1188  2
;  699          1189  2      if .lnk$gl_shrsyms neq 0
;  700          1190  2      then while (lnk$gl_curclu = .lnk$gl_curclu[clu$l_nxtclu]) neq 0
;  701          1191  2      do if .lnk$gl_curclu[clu$l_shrsyms] neq 0                           ! IF THIS CLUSTER HAS SHAREABLE IMAG
;  702          1192  3      then begin
;  703          1193  3          fixadr = .fixadr + 8;                                           ! LEAVE ROOM FOR # OF REFS, AND SHR
;  704          1194  3          symbolblock = .lnk$gl_curclu[clu$l_shrlst];                     ! NOW RUN THROUGH ALL THE SYMBOLS
;  705          1195  3          while .symbolblock neq 0
;  706          1196  4          do begin
;  707          1197  4              symbolsnb = .symbolblock - .symbolblock[sym$b_namlng]       ! POINT TO NAME BLOCK PART
;  708          1198  4                                       - snb$c_fxdlen;
;  709          1199  4              symbolblock[sym$l_offset] = .symbolblock[sym$l_value];      ! SAVE OFFSET
;  710          1200  4              symbolblock[sym$l_value] = .fixadr;                         ! DEFINE SYMBOL TO BE HERE INSTEAD
;  711          1201  4              fixadr = .fixadr + 4;                                       ! NEXT SLOT
;  712          1202  4              symbolblock = .symbolblock[sym$l_shrlnk];                   ! AND LINK TO NEXT
;  713          1203  3              end;
;  714          1204  2          end;
;  715          1205  2
;  716          1206  2      lnk$gl_lstgadr = .fixadr+4;
;  717          1207  2      lnk$gl_lastgadr = .fixadr+4;
;  718          1208  2      lnk$gl_curclu = .savecurclu;                                        ! RESTORE CURRENT CLUSTER
;  719          1209  2
;  720          1210  2  return true
;  721          1211  1  end;


                                                  0FFC 00000          .ENTRY  LNK$CREFIXISD, Save R2,R3,R4,R5,R6,R7,R8,-      ; 1128
                                                                              R9,R10,R11
                                  5B 00000000G    00   9E 00002      MOVAB   LNK$GL_CURCLU, R11
                                  5A 00000000'    EF   9E 00009      MOVAB   ISCTDESC, R10
                                  5E        A8    AE   9E 00010      MOVAB   -88(SP), SP
                                  58 00000000'    EF   D0 00014      MOVL    LNK$GL_HIGHCLU, R8                              ; 1146
                                  59        1C    A8   D0 0001B      MOVL    28(R8), STACKISD
                                  57              59   D0 0001F      MOVL    STACKISD, LASTISD
                                  56        18    A7   9E 00022      MOVAB   24(R7), LASTHDRISD                              ; 1147
                                  50        18    A8   9E 00026      MOVAB   24(R8), R0                                      ; 1148
                                  50              57   D1 0002A      CMPL    LASTISD, R0
                                                  10   12 0002D      BNEQ    1$
    0058  8F             00       6E              00   2C 0002F      MOVC5   #0, (SP), #0, #88, TEMPISD                      ; 1150
                                                  6E      00036
                                  56        18    AE   9E 00037      MOVAB   TEMPISD+24, LASTHDRISD                          ; 1151
                            02    A6              01   B0 0003R      MOVW    #1, 2(LASTHDRISD)                               ; 1152
                            FD    8F        0B    A6   91 0003F 1$:  CMPB    11(LASTHDRISD), #253                            ; 1154
                                                  04   13 00044      BEQL    2$
                                                  59   D4 00046      CLRL    STACKISD                                        ; 1155
                                                  08   11 00048      BRB     3$
                                  57        04    A7   D0 0004A 2$:  MOVL    4(LASTISD), LASTISD                             ; 1157
                                  56        18    A7   9E 0004E      MOVAB   24(R7), LASTHDRISD                              ; 1158
                                  55              6B   D0 00052 3$:  MOVL    LNK$GL_CURCLU, SAVECURCLU                       ; 1160
                                  6B              58   D0 00055      MOVL    R8, LNK$GL_CURCLU                               ; 1161
        50        04    A6        17              00   EF 00058      EXTZV   #0, #23, 4(LASTHDRISD), R0                      ; 1162
```

```
  50                 50              09   78 0005E          ASHL    #9, R0, R0                                  ;
                     56        02    A6   3C 00062          MOVZWL  2(LASTHDRISD), R6                           ; 1163
  56                 56              09   78 00066          ASHL    #9, R6, R6                                  ;
  52                 50              56   C1 0006A          ADDL3   R6, R0, FIXADR                              ;
              EC     AA              52   D0 0006E          MOVL    FIXADR, ISECT_BASE                          ; 1162
                                     06   12 00072          BNEQ    4$                                          ; 1164
              EC     AA      0200    8F   3C 00074          MOVZWL  #512, ISECT_BASE                            ; 1165
                                     5A   DD 0007A 4$:      PUSHL   R10                                         ; 1166
                     7E        58    8F   9A 0007C          MOVZBL  #88, -(SP)                                  ;
      00000000G      00              02   FB 00080          CALLS   #2, LNK$ALLOBLK                             ;
                     67              6A   D0 00087          MOVL    ISCTDESC, (LASTISD)                         ; 1167
      00000000'      EF              67   D0 0008A          MOVL    (LASTISD), LNK$GL_FIXISD                    ;
   06 00000000G      00              02   E1 00091          BBC     #2, LNK$GL_CTLMSK, 5$                       ; 1168
              F4     AA              06   90 00099          MOVB    #6, ISECT                                   ; 1169
                                     04   11 0009D          BRB     6$                                          ;
              F4     AA              02   90 0009F 5$:      MOVB    #2, ISECT                                   ; 1170
              FC     AA              57   D0 000A3 6$:      MOVL    LASTISD, PREVISECT                          ; 1171
                     50 00000000G    00   D0 000A7          MOVL    LNK$GL_SHRIMGS, R0                          ; 1172
                     51 00000000G    00   D0 000AE          MOVL    LNK$GL_SHRSYMS, R1                          ;
                     50          6140     3E 000B5          MOVAW   (R1)[R0], R0                                ;
                     50              04   C4 000B9          MULL2   #4, R0                                      ;
                     50      0243    C0   9E 000BC          MOVAB   579(R0), R0                                 ;
  7E                 50 00000200     8F   C7 000C1          DIVL3   #512, R0, -(SP)                             ; 1173
      00000000V      EF              01   FB 000C9          CALLS   #1, BUILDISD                                ;
                                     59   D5 000D0          TSTL    STACKISD                                    ; 1174
                                     0F   13 000D2          BEQL    7$                                          ;
              00     BA              59   D0 000D4          MOVL    STACKISD, @ISCTDESC                         ; 1176
              04     A9              6A   D0 000D8          MOVL    ISCTDESC, 4(STACKISD)                       ; 1177
                     50              6B   D0 000DC          MOVL    LNK$GL_CURCLU, R0                           ; 1178
              1C     A0              59   D0 000DF          MOVL    STACKISD, 28(R0)                            ;
  50                 6A              18   C1 000E3 7$:      ADDL3   #24, ISCTDESC, HDRISD                       ; 1180
                     52        40    A2   9E 000E7          MOVAB   64(R2), FIXADR                              ; 1181
                     6B 00000000G    00   9E 000EB          MOVAB   LNK$GL_CLULST, LNK$GL_CURCLU                ; 1187
                        00000000G    00   D5 000F2          TSTL    LNK$GL_SHRSYMS                              ; 1189
                                     32   13 000F8          BEQL    10$                                         ;
                     53              6B   D0 000FA          MOVL    LNK$GL_CURCLU, R3                           ; 1190
                     6B              63   D0 000FD 8$:      MOVL    (R3), LNK$GL_CURCLU                         ;
                                     2A   13 00100          BEQL    10$                                         ;
                     53              6B   D0 00102          MOVL    LNK$GL_CURCLU, R3                           ; 1191
                               2C    A3   D5 00105          TSTL    44(R3)                                      ;
                                     F3   13 00108          BEQL    8$                                          ;
                     52              08   C0 0010A          ADDL2   #8, FIXADR                                  ; 1193
                     50        30    A3   D0 0010D          MOVL    48(R3), SYMBOLBLOCK                         ; 1194
                                     EA   13 00111 9$:      BEQL    8$                                          ; 1195
                     54        0F    A0   9A 00113          MOVZBL  15(SYMBOLBLOCK), R4                         ; 1197
  54                 50              54   C3 00117          SUBL3   R4, SYMBOLBLOCK, R4                         ;
                     51        FB    A4   9E 0011B          MOVAB   -5(R4), SYMBOLSNB                           ; 1198
              10     A0              60   D0 0011F          MOVL    (SYMBOLBLOCK), 16(SYMBOLBLOCK)              ; 1199
                     60              82   DE 00123          MOVAL   (FIXADR)+, (SYMBOLBLOCK)                    ; 1200
                     50        1C    A0   D0 00126          MOVL    28(SYMBOLBLOCK), SYMBOLBLOCK                ; 1202
                                     E5   11 0012A          BRB     9$                                          ; 1195
                     52              04   C0 0012C 10$:     ADDL2   #4, R2                                      ; 1206
      00000000'      EF              52   D0 0012F          MOVL    R2, LNK$GL_1STGADR                          ;
      00000000'      EF              52   D0 00136          MOVL    R2, LNK$GL_LASTGADR                         ; 1207
                     6B              55   D0 0013D          MOVL    SAVECURCLU, LNK$GL_CURCLU                   ; 1208
                     50              01   D0 00140          MOVL    #1, R0                                      ; 1210
                                          04 00143          RET                                                 ; 1211
```

; Routine Size: 324 bytes,   Routine Base: $CODE$ + 056E

```
;  723          1212  1 global routine lnk$cluvirmem (clusterdesc) =
;  724          1213  2 begin
;  725          1214  2 !
;  726          1215  2 !       THIS ROUTINE ALLOCATES VIRTUAL MEMORY FOR A CLUSTER
;  727          1216  2 !       IF THE MEMORY CANNOT BE ALLOCATED THE IMAGE PRODUCTION IS
;  728          1217  2 !       DISABLED.
;  729          1218  2 !
;  730          1219  2 ! INPUTS:
;  731          1220  2 !
;  732          1221  2 !       CLUSTERDESC     A POINTER TO THE CLUSTER DESCRIPTOR FOR CLUSTER TO
;  733          1222  2 !                       BE ALLOCATED
;  734          1223  2 !
;  735          1224  2 ! OUTPUTS:
;  736          1225  2 !
;  737          1226  2 !       VIRTUAL MEMORY ALLOCATED. [CLU$L_BASE] IN THE CLUSTER DESCRIPTOR
;  738          1227  2 !       SET TO VA OF ALLOCATED MEMORY.
;  739          1228  2 !
;  740          1229  2 !
;  741          1230  2 map
;  742          1231  2     clusterdesc : ref block[,byte];
;  743          1232  2
;  744          1233  2 local
;  745          1234  2     savecluster,
;  746          1235  2     status;
;  747          1236  2
;  748          1237  4 if not (status = (if not .clusterdesc[clu$v_based]                ! IF THIS CLUSTER NOT BASED
;  749          1238  4 !           AND .LNK$GL_CTLMSK[LNK$V_SHR]                       ! AND WE ARE PRODUCING A SHAREABLE IMAGE
;  750          1239  4             and .defclubased                                    ! THAT IS BASED
;  751          1240  4         then lnk$findvirmem(clusterdesc[clu$l_base],            ! THEN FIND THE VIRTUAL MEMORY
;  752          1241  4                             .clusterdesc[clu$l_pages],
;  753          1242  4                             .lnk$gl_defclu[clu$l_base])        ! BUT INSIST THAT IT BE ABOVE THE BASE= BASE
;  754          1243  4         else if .clusterdesc[clu$v_based]                       ! IF THIS CLUSTER IS BASED
;  755          1244  4             then lnk$allovirmem(.clusterdesc[clu$l_base],.clusterdesc[clu$l_pages])
;  756          1245  4             else lnk$findvirmem(clusterdesc[clu$l_base],        ! ATTEMPT TO FIND ENOUGH MEMORY
;  757          1246  3                         .clusterdesc[clu$l_pages])))           ! FOR CLUSTER, RETURNING BASE ADDRES
;  758          1247  3 then begin
;  759          1248  3     signal(lin$_insvirmem,2,                                    ! BUT ISSUE ERROR IF INSUFFICIENT
;  760          1249  3                 .clusterdesc[clu$l_pages],clusterdesc[clu$b_namlng],lin$_noimgfil);
;  761          1250  3     lnk$gl_ctlmsk[lnk$v_image] = false;                         ! AND DISABLE IMAGE OUTPUT
;  762          1251  2     end;
;  763          1252  2 if .clusterdesc[clu$v_shrimg]
;  764          1253  2     then clusterdesc[clu$v_based] = true;                       ! IT'S BASED NOW
;  765          1254  2 isctdesc = clusterdesc[clu$l_fstisd];                         ! INITIALIZE TO LISTHEAD FOR ISECTIONS
;  766          1255  3 if (if .clusterdesc[clu$v_shrimg]
;  767          1256  3         then .clusterdesc[clu$v_pic]
;  768          1257  3         else not .clusterdesc[clu$v_based])
;  769          1258  2 then while (isctdesc = .isctdesc[isl$l_nxtisd]) neq 0   ! GET NEXT ISD IN LIST
;  770          1259  3 do begin
;  771          1260  3     bind
;  772          1261  3         hdrisd = isctdesc[isl$t_hdrisd] : block[,byte]; ! POINT TO HEADER PART
;  773          1262  3
;  774          1263  3     if not .hdrisd[isd$v_p1]                                    ! IF NOT CONTROL REGION
;  775          1264  3         then hdrisd[isd$v_vpg] = .hdrisd[isd$v_vpg] +           ! ADD THE CLUSTER BASE PAGE
;  776          1265  3                 (.clusterdesc[clu$l_base] ^ -9);            ! TO I-SECTION PAGE NUMBER
;  777          1266  3     if .clusterdesc[clu$v_shrimg]                               ! IF SHAREABLE IMAGE
;  778          1267  3         then hdrisd[isd$v_based] = true;                        ! FLAG SECTION AS BASED
;  779          1268  2     end;
```

```
;  780          1269  2 !
;  781          1270  2 ! SEE IF LAST CLUSTER
;  782          1271  2 !
;  783          1272  2 lnk$setlastclu(.clusterdesc);
;  784          1273  2 !
;  785          1274  2 ! RELOCATE CONTENTS OF CLUSTER
;  786          1275  2 !
;  787          1276  3 if (if .clusterdesc[clu$v_shrimg]
;  788          1277  3         then .clusterdesc[clu$v_pic]
;  789          1278  3         else not .clusterdesc[clu$v_based])
;  790          1279  3 then begin
;  791          1280  3     savecluster = .lnk$gl_curclu;
;  792          1281  3     lnk$gl_curclu = .clusterdesc;
;  793          1282  3     relocluster();
;  794          1283  3     lnk$gl_curclu = .savecluster;
;  795          1284  2     end;
;  796          1285  2
;  797          1286  2 return .status
;  798          1287  1 end;

                                  03FC 00000          .ENTRY  LNK$CLUVIRMEM, Save R2,R3,R4,R5,R6,R7,R8,R9 ; 1212
                59 00000000G   00  9E 00002          MOVAB   LNK$FINDVIRMEM, R9                          ;
                58 00000000G   00  9E 00009          MOVAB   LNK$GL_CURCLU, R8                           ;
                57 00000000'   EF  9E 00010          MOVAB   ISCTDESC, R7                                ;
                52        04   AC  D0 00017          MOVL    CLUSTERDESC, R2                             ; 1237
                54        58   A2  9E 0001B          MOVAB   88(R2), R4                                  ;
                18             64  E8 0001F          BLBS    (R4), 2$                                    ;
                11        E0   A7  E9 00022          BLBC    DEFCLUBASED, 1$                             ; 1239
                     00000000G 00  DD 00026          PUSHL   LNK$GL_DEFCLU+76                            ; 1242
                           50  A2  DD 0002C          PUSHL   80(R2)                                      ; 1241
                           4C  A2  9F 0002F          PUSHAB  76(R2)                                      ; 1240
                69             03  FB 00032          CALLS   #3, LNK$FINDVIRMEM                          ;
                               19  11 00035          BRB     4$                                          ;
                0D             64  E9 00037 1$:      BLBC    (R4), 3$                                    ; 1243
                7E        4C   A2  7D 0003A 2$:      MOVQ    76(R2), -(SP)                               ; 1244
          00000000G 00         02  FB 0003E          CALLS   #2, LNK$ALLOVIRMEM                          ;
                               09  11 00045          BRB     4$                                          ;
                           50  A2  DD 00047 3$:      PUSHL   80(R2)                                      ; 1246
                           4C  A2  9F 0004A          PUSHAB  76(R2)                                      ; 1245
                69             02  FB 0004D          CALLS   #2, LNK$FINDVIRMEM                          ;
                55             50  D0 00050 4$:      MOVL    R0, STATUS                                  ;
                22             55  E8 00053          BLBS    STATUS, 5$                                  ; 1237
                     00000000G 8F  DD 00056          PUSHL   #LIN$_NOIMGFIL                              ; 1249
                           5C  A2  9F 0005C          PUSHAB  92(R2)                                      ;
                           50  A2  DD 0005F          PUSHL   80(R2)                                      ;
                               02  DD 00062          PUSHL   #2                                          ;
                     00000000G 8F  DD 00064          PUSHL   #LIN$_INSVIRMEM                             ;
          00000000G 00         05  FB 0006A          CALLS   #5, LIB$SIGNAL                              ;
          00000000G 00         01  8A 00071          BICB2   #1, LNK$GL_CTLMSK                           ; 1250
03                64             02  E1 00078 5$:      BBC     #2, (R4), 6$                                ; 1252
                  64             01  88 0007C          BISB2   #1, (R4)                                    ; 1253
                  67        18   A2  9E 0007F 6$:      MOVAB   24(R2), ISCTDESC                            ; 1254
06                64             02  E1 00083          BBC     #2, (R4), 7$                                ; 1255
```

```
                        38              64              03  E1  00087         BBC     #3, (R4), 11$                           ; 1256
                                                        03  11  0008B         BRB     8$
                                        33              64  E8  0008D 7$:     BLBS    (R4), 11$                               ; 1257
                                        51              67  D0  00090 8$:     MOVL    ISCTDESC, R1                            ; 1258
                                        67              61  D0  00093 9$:     MOVL    (R1), ISCTDESC
                                                        2B  13  00096         BEQL    11$
                                        51              67  D0  00098         MOVL    ISCTDESC, R1                            ; 1261
                                        50      18      A1  9E  0009B         MOVAB   24(R1), R0
                        15      06      A0              05  E0  0009F         BBS     #5, 6(R0), 10$                          ; 1263
                        53      4C      A2      F7      8F  78  000A4         ASHL    #-9, 76(R2), R3                         ; 1265
                56              04      A0              17  00  EF  000AA     EXTZV   #0, #23, 4(R0), R6
                                        56              53  C0  000B0         ADDL2   R3, R6
            04  A0                      17              00  56  F0  000B3     INSV    R6, #0, #23, 4(R0)
                                        D6              64  02  E1  000B9 10$:  BBC     #2, (R4), 9$                            ; 1266
                                                09      A0  02  88  000BD     BISB2   #2, 9(R0)                               ; 1267
                                                        D0  11  000C1         BRB     9$                                      ; 1258
                                                        52  DC  000C3 11$:     PUSHL   R2                                      ; 1272
                                00000000V       EF      01  FB  000C5         CALLS   #1, LNK$SETLASTCLU
                        06                      64  02  E1  000CC         BBC     #2, (R4), 12$                           ; 1276
                        13                      64  03  E1  000D0         BBC     #3, (R4), 14$                           ; 1277
                                                        03  11  000D4         BRB     13$
                                        0E      64  E8  000D6 12$:     BLBS    (R4), 14$                               ; 1278
                                        54      68  D0  000D9 13$:     MOVL    LNK$GL_CURCLU, SAVECLUSTER                      ; 1280
                                        68      52  D0  000DC         MOVL    R2, LNK$GL_CURCLU                       ; 1281
                                FD9A    CF      00  FB  000DF         CALLS   #0, RELOCLUSTER                         ; 1282
                                        68      54  D0  000E4         MOVL    SAVECLUSTER, LNK$GL_CURCLU              ; 1283
                                        50      55  D0  000E7 14$:     MOVL    STATUS, R0                              ; 1286
                                                    04  000EA         RET                                             ; 1287

; Routine Size:  235 bytes,    Routine Base:  $CODE$ + 06B2

;  799          1288  1
```

```
;  801          1289  1 global routine lnk$setlastclu (cludesc) =
;  802          1290  2 begin
;  803          1291  2 !
;  804          1292  2 !       THIS ROUTINE TESTS IF THE CLUSTER PASSED TO IT IS INDEED
;  805          1293  2 !       THE LAST CLUSTER.  IF SO, IT SETS LNK$GL_LSTCLSTR
;  806          1294  2 !
;  807          1295  2 map
;  808          1296  2     cludesc : ref block[,byte];
;  809          1297  2
;  810          1298  3 if (.lnk$gl_lstclstr eql 0)
;  811          1299  3     or (.cludesc[clu$l_base] gtru .lnk$gl_lstclstr[clu$l_base])
;  812          1300  2     then lnk$gl_lstclstr = .cludesc;
;  813          1301  2
;  814          1302  2 return true
;  815          1303  1 end;


                                        0004 00000          .ENTRY  LNK$SETLASTCLU, Save R2                 ; 1289
                    52 00000000'    EF  9E 00002            MOVAB   LNK$GL_LSTCLSTR, R2                     ;
                    51              62  D0 00009            MOVL    LNK$GL_LSTCLSTR, R1                     ; 1298
                                    0B  13 0000C            BEQL    1$                                      ;
                    50        04    AC  D0 0000E            MOVL    CLUDESC, R0                             ; 1299
                4C  A1        4C    A0  D1 00012            CMPL    76(R0), 76(R1)                          ;
                                    04  1B 00017            BLEQU   2$                                      ;
                    62        04    AC  D0 00019 1$:        MOVL    CLUDESC, LNK$GL_LSTCLSTR                ; 1300
                    50              01  D0 0001D 2$:        MOVL    #1, R0                                  ; 1302
                                        04 00020            RET                                             ; 1303

; Routine Size:  33 bytes,    Routine Base:  $CODE$ + 079D
```

```
;  817          1304  1 routine stack_isectbld : novalue =
;  818          1305  2 begin
;  819          1306  2 !
;  820          1307  2 !       THIS ROUTINE BUILDS A DEMAND ZERO IMAGE SECTION FOR
;  821          1308  2 !       USER STACK, IF ANY AND IF ON THE LAST CLUSTER.
;  822          1309  2 !
;  823          1310  2 if .lnk$gw_stack neq 0                                    ! IF THERE IS SOME STACK
;  824          1311  3 and (.lnk$gl_curclu[clu$l_nxtclu] eql 0                   ! AND WE ARE ON THE LAST CLUSTER
;  825          1312  3     or .lnk$gl_curclu eql lnk$gl_defclu)
;  826          1313  3 then begin
;  827          1314  4         if (isect_base = .cntrl_reg_addr - .lnk$gw_stack*512)   ! SET BASE OF STACK
;  828          1315  3                 lssu control_region                     ! IF IT GOES BELOW CONTROL REGION
;  829          1316  3         then signal_stop(lin$_stkovflo,2,               ! AN ERROR
;  830          1317  3                         .lnk$gw_stack,.isect_base);
;  831          1318  3         if .isctdesc[isl$l_nxtisd] eql 0                ! IF CURRENT ISD IS THE LAST
;  832          1319  4         then begin                                      ! ONE IN USE
;  833          1320  4                 lnk$alloblk(isd$c_size,isctdesc[isl$l_nxtisd]); ! GO ALLOCATE A NEW DESCRIPTOR
;  834          1321  4                 isctdesc = .isctdesc[isl$l_nxtisd];     ! AND LINK TO THE PREVIOUS ONE
;  835          1322  3             end;
;  836          1323  3         buildisd(.lnk$gw_stack);                        ! GO CREATE ISD
;  837          1324  3         cntrl_reg_addr = .isect_base;                   ! UPDATE CONTROL REGION POINTER
;  838          1325  2     end;
;  839          1326  2 return;                                                 ! AND ALL DONE
;  840          1327  1 end.                                                    ! OF STACK ALLOCATOR

                                001C 00000 STACK_ISECTBLD:
                                            .WORD   Save R2,R3,R4               ;  1304
                    54 00000000'  EF  9E 00002      MOVAB   LNK$GW_STACK, R4
                    53 00000000'  EF  9E 00009      MOVAB   ISECT_BASE, R3
                    52            64  3C 00010      MOVZWL  LNK$GW_STACK, R2      ;  1310
                                  67  13 00013      BEQL    4$
                    51 00000000G  00  D0 00015      MOVL    LNK$GL_CURCLU, R1     ;  1311
                                  61  D5 0001C      TSTL    (R1)
                                  0C  13 0001E      BEQL    1$
                    50 00000000G  00  9E 00020      MOVAB   LNK$GL_DEFCLU, R0     ;  1312
                    50            51  D1 00027      CMPL    R1, R0
                                  50  12 0002A      BNEQ    4$
51                  52            09  78 0002C 1$:  ASHL    #9, R2, R1            ;  1314
51          04      A3            51  C3 00030      SUBL3   R1, CNTRL_REG_ADDR, R1
                    63            51  D0 00035      MOVL    R1, ISECT_BASE
    40000000        8F            51  D1 00038      CMPL    R1, #1073741824       ;  1315
                                  13  1E 0003F      BGEQU   2$
                                  63  DD 00041      PUSHL   ISECT_BASE            ;  1317
                                  52  DD 00043      PUSHL   R2
                                  02  DD 00045      PUSHL   #2                    ;  1316
                      00000000G   8F  DD 00047      PUSHL   #LIN$_STKOVFLO
    00000000G         00          04  FB 0004D      CALLS   #4, LIB$STOP
                    50      14    A3  D0 00054 2$:  MOVL    ISCTDESC, R0          ;  1318
                                  60  D5 00058      TSTL    (R0)
                                  12  12 0005A      BNEQ    3$
                                  50  DD 0005C      PUSHL   R0                    ;  1320
                    7E      58    8F  9A 0005E      MOVZBL  #88, -(SP)
    00000000G       00            02  FB 00062      CALLS   #2, LNK$ALLOBLK
```

```
                     14   A3        14   B3  D0 00069          MOVL    @ISCTDESC, ISCTDESC                     ; 1321
                          7E             64  3C 0006E 3$:     MOVZWL  LNK$GW_STACK, -(SP)                     ; 1323
            00000000V     EF             01  FB 00071         CALLS   #1, BUILDISD                            ;
                     04   A3             63  D0 00078         MOVL    ISECT_BASE, CNTRL_REG_ADDR              ; 1324
                                             04 0007C 4$:     RET                                             ; 1327

; Routine Size:  125 bytes,    Routine Base:  $CODE$ + 07BE
```

```
;  842          1328  1 routine normal_isectbld : novalue =
;  843          1329  2 begin
;  844          1330  2 !
;  845          1331  2 !       THIS ROUTINE BUILDS THE NORMAL IMAGE-SECTIONS OF THE
;  846          1332  2 !       PROGRAM REGION BY GATHERING COMPATIBLE P-SECTIONS IN
;  847          1333  2 !       LEXICAL ORDER
;  848          1334  2 !
;  849          1335  2 routine findmask_action (psctdesc) =
;  850          1336  3 begin
;  851          1337  3 !
;  852          1338  3 ! THIS ROUTINE IS CALLED BY LNK$FNDPSCMSK FOR EACH PSECT
;  853          1339  3 ! THAT MATCHES THE MASK
;  854          1340  3 !
;  855          1341  3 map
;  856          1342  3     psctdesc : ref block[,byte];
;  857          1343  3
;  858          1344  3 local
;  859          1345  3         around,
;  860          1346  3         isect_blks,
;  861          1347  3         symsnb : ref block[,byte],                    ! PTR TO NAME BLOCK
;  862          1348  3         psect_length,
;  863          1349  3         pscpart_base,
;  864          1350  3         symbol : ref block[,byte];
;  865          1351  3 !
;  866          1352  3 bind
;  867          1353  3         isd$c_maxblk = %x'FFFF',                      ! MAX ISD BLOCK SIZE
;  868          1354  3         isd$c_maxblkbyt = isd$c_maxblk*512;           ! EXPRESSED IN BYTES ALSO
;  869          1355  3 !
;  870          1356  4 begin                                                 ! WITH APPROPRIATE ATTRIBUTES
;  871          1357  4         if .isctdesc[isl$l_nxtisd] eql 0              ! IF WE HAVEN'T ALLOCATED AN ISECT
;  872          1358  5         then begin                                    ! DESCRIPTOR THAT WASN'T USED
;  873          1359  5                 lnk$alloblk(isd$c_size,isctdesc[isl$l_nxtisd]); ! GO ALLOCATE DESCRIPTOR
;  874          1360  5                 isctdesc = .isctdesc[isl$l_nxtisd];   ! AND LINK TO THE PREVIOUS ONE
;  875          1361  5                 isctdesc[isl$l_nxtisd] = -1;          ! TEMPORARILY MARK AS ALLOCATED
;  876          1362  4              end;
;  877          1363  4         around = (1^.psctdesc[psc$b_align]) - 1;      ! COMPUTE THE ADJUSTMENT FOR ALIGNING
;  878          1364  4         current_base = (.current_base + .around) and not .around;! AND DO IT THEN
;  879          1365  4         pscpart_base = .current_base;                 ! SET BASE OF PSECT SEGMENT
;  880          1366  4         psect_length = .psctdesc[psc$l_length];       ! SET LENGTH OF PSECT
;  881          1367  4         psctdesc[psc$l_isect] = 0;                    ! ENSURE 0 ISECT POINTER
;  882          1368  5         while ((.pscpart_base - .isect_base + .psect_length) gtru isd$c_maxblkbyt
;  883          1369  5                 and (.psect_length gtru 0))
;  884          1370  5         do begin
;  885          1371  5             isect_blks = isd$c_maxblk;                ! SET TO MAXIMUM ISD
;  886          1372  5             buildisd(.isect_blks);                    ! BUILD ISD WE HAVE
;  887          1373  5             lnk$gl_curclu[clu$l_pages] = .lnk$gl_curclu[clu$l_pages] + .isect_blks; ! ADD ISECT SIZE TO CLUS
;  888          1374  5             if .psctdesc[psc$l_isect] eql 0 then psctdesc[psc$l_isect] = .isctdesc; ! IF PSECT NOT BOUND TO
;  889          1375  5             lnk$alloblk(isd$c_size,isctdesc[isl$l_nxtisd]);  ! ALLOCATE A NEW ISD
;  890          1376  5             isctdesc = .isctdesc[isl$l_nxtisd];       ! LINK INTO LIST
;  891          1377  5             isctdesc[isl$l_nxtisd] = -1;              ! TEMPORARILY MARK AS ALLOCA
;  892          1378  5             psect_length = (.pscpart_base - .isect_base + .psect_length) - isd$c_maxblkbyt; ! REDUCE   ECT L
;  893          1379  5             isect_base = (.isect_base + isd$c_maxblkbyt);   ! SET NEW ISECT BASE
;  894          1380  5             pscpart_base = .isect_base;               ! SET BASE OF PART OF PSECT
;  895          1381  4             end;
;  896          1382  4         psctdesc[psc$l_base] = .current_base;         ! SET P-SECT BASE
;  897          1383  4         if .psctdesc[psc$l_isect] eql 0 then psctdesc[psc$l_isect] = .isctdesc; ! BIND PSECT TO ISECT IF NOT
;  898          1384  4         current_base = .current_base + .psctdesc[psc$l_length];  ! THEN UPDATE FOR THE NEXT
```

```
;  899          1385  4          pscpart_base = .current_base;                    ! UPDATE PSECT BASE POINTER
;  900          1386  3 end;
;  901          1387  3 return true                                              ! AND DONE
;  902          1388  2 end;                                                     ! OF FINDMASK_ACTION


                                                ISD$C_MAXBLK=       65535
                                                ISD$C_MAXBLKBYT=    33553920

                                    03FC 00000 FINDMASK_ACTION:
                                                .WORD   Save R2,R3,R4,R5,R6,R7,R8,R9                ; 1335
                59 00000000G    00  9E 00002    MOVAB   LNK$ALLOBLK, R9                             ;
                58 00000000'    EF  9E 00009    MOVAB   ISCTDESC, R8                                ;
                50              68  D0 00010    MOVL    ISCTDESC, R0                                ; 1357
                                60  D5 00013    TSTL    (R0)                                        ;
                                10  12 00015    BNEQ    1$                                          ;
                                50  DD 00017    PUSHL   R0                                          ; 1359
                7E      58      8F  9A 00019    MOVZBL  #88, -(SP)                                  ;
                69              02  FB 0001D    CALLS   #2, LNK$ALLOBLK                             ;
                78              98  D0 00020    MOVL    @ISCTDESC, ISCTDESC                         ; 1360
             00 B8              01  CE 00023    MNEGL   #1, @ISCTDESC                               ; 1361
                53      04      AC  D0 00027 1$:MOVL    PSCTDESC, R3                                ; 1363
          50            01      2C  A3 78 0002B ASHL    44(R3), #1, R0                              ;
                                50  D7 00030    DECL    AROUND                                      ;
          51            50      E8  A8 C1 00032 ADDL3   CURRENT_BASE, AROUND, R1                    ; 1364
       E8 A8            51          50 CB 00037 BICL3   AROUND, R1, CURRENT_BASE                    ;
                52      E8      A8  D0 0003C    MOVL    CURRENT_BASE, PSCPART_BASE                  ; 1365
                54      1C      A3  D0 00040    MOVL    28(R3), PSECT_LENGTH                        ; 1366
                55      20      A3  9E 00044    MOVAB   32(R3), R5                                  ; 1367
                                65  D4 00048    CLRL    (R5)                                        ;
                56      EC      A8  D0 0004A    MOVL    ISECT_BASE, R6                              ; 1368
          50    52              56  C3 0004E 2$:SUBL3   R6, PSCPART_BASE, R0                        ;
                50              54  C0 00052    ADDL2   PSECT_LENGTH, R0                            ;
    01FFFE00    8F              50  D1 00055    CMPL    R0, #33553920                               ;
                                52  1B 0005C    BLEQU   4$                                          ;
                                54  D5 0005E    TSTL    PSECT_LENGTH                                ; 1369
                                4E  13 00060    BEQL    4$                                          ;
                57      FFFF    8F  3C 00062    MOVZWL  #65535, ISECT_BLKS                          ; 1371
                                57  DD 00067    PUSHL   ISECT_BLKS                                  ; 1372
    00000000V   EF              01  FB 00069    CALLS   #1, BUILDISD                                ;
                50 00000000G    00  D0 00070    MOVL    LNK$GL_CURCLU, R0                           ; 1373
          50    A0              57  C0 00077    ADDL2   ISECT_BLKS, 80(R0)                          ;
                                65  D5 0007B    TSTL    (R5)                                        ; 1374
                                03  12 0007D    BNEQ    3$                                          ;
                65              68  D0 0007F    MOVL    ISCTDESC, (R5)                              ;
                                68  DD 00082 3$:PUSHL   ISCTDESC                                    ; 1375
                7E      58      8F  9A 00084    MOVZBL  #88, -(SP)                                  ;
                69              02  FB 00088    CALLS   #2, LNK$ALLOBLK                             ;
                78              98  D0 0008B    MOVL    @ISCTDESC, ISCTDESC                         ; 1376
          00    B8              01  CE 0008E    MNEGL   #1, @ISCTDESC                               ; 1377
50              52      EC      A8  C3 00092    SUBL3   ISECT_BASE, PSCPART_BASE, R0                ; 1378
                54 FE000200 E440    9E 00097    MOVAB   -33553920(PSECT_LENGTH)[R0], PSECT_LENGTH   ;
          EC    A8 01FFFE00     8F  C0 0009F    ADDL2   #33553920, ISECT_BASE                       ; 1379
                56      EC      A8  D0 000A7    MOVL    ISECT_BASE, R6                              ; 1380
                52              56  D0 000AB    MOVL    R6, PSCPART_BASE                            ;
```

```
                                 9E  11 000AE          BRB     2$                                      ; 1368
                      18  A3     E8  A8 D0 000B0 4$:   MOVL    CURRENT_BASE, 24(R3)                    ; 1382
                                     65 D5 000B5       TSTL    (R5)                                    ; 1383
                                     03 12 000B7       BNEQ    5$                                      ;
                          65         68 D0 000B9       MOVL    ISCTDESC, (R5)                          ;
                      E8  A8     1C  A3 C0 000BC 5$:   ADDL2   28(R3), CURRENT_BASE                    ; 1384
                          52     E8  A8 D0 000C1       MOVL    CURRENT_BASE, PSCPART_BASE              ; 1385
                          50         01 D0 000C5       MOVL    #1, R0                                  ; 1387
                                     04 000C8          RET                                             ; 1388

; Routine Size:  201 bytes,    Routine Base:  $CODE$ + 083B

;  903          1389  2 !                                                                              ; R
;  904          1390  2 ! MAIN BODY OF NORMAL_ISECTBLD                                                 ;
;  905          1391  2 !                                                                              ;
;  906          1392  2 local
;  907          1393  2     isect_blks;
;  908          1394  2
;  909          1395  2 bind
;  910          1396  2         attribmask = .isectgentbl[.isect-1,isc$w_mask],      ! CREATE MORE
;  911          1397  2         attribvalue = .isectgentbl[.isect-1,isc$w_match];    ! CONVENIENT NAMES
;  912          1398  2
;  913          1399  2 current_base = .isect_base;                                  ! SET FIRST PSECT BASE
;  914          1400  2 lnk$fndpscmsk(attribmask,attribvalue,findmask_action);      ! FIND COMPATIBLE PSECTS
;  915          1401  2 isect_blks = (.current_base - .isect_base+511)/512;         ! COMPUTE NUMBER OF BLOCKS
;  916          1402  2 if .isctdesc[isl$l_nxtisd] eql -1                           ! IF A DESCRIPTOR WAS ALLOCATED
;  917          1403  2 then buildisd(.isect_blks);                                 ! THEN CREATE ISD
;  918          1404  2 lnk$gl_curclu[clu$l_pages] = .lnk$gl_curclu[clu$l_pages] + ! ADD I-SECTION SIZE TO CLUSTER SIZE
;  919          1405  2                                             .isect_blks;
;  920          1406  2 isect_base = .current_base;                                 ! SET BASE OF NEXT I-SECT
;  921          1407  2 return true
;  922          1408  1 end;


                                 000C 00000 NORMAL_ISECTBLD:
                                            .WORD   Save R2,R3                                         ; 1328
                          53 00000000'   EF  9E 00002       MOVAB   ISECT_BASE, R3                   ;
                          51         08  A3  9A 00009       MOVZBL  ISECT, R1                        ; 1396
                          51             09  C4 0000C       MULL2   #9, R1                           ;
                50        51         0C  A3  C1 00010       ADDL3   ISECTGENTBL, R1, R0              ;
                          52             50  D0 00015       MOVL    R0, R2                           ;
                50        51         0C  A3  C1 00018       ADDL3   ISECTGENTBL, R1, R0              ; 1397
                      FC  A3             63  D0 0001D       MOVL    ISECT_BASE, CURRENT_BASE         ; 1399
                              FF12       CF  9F 00021       PUSHAB  FINDMASK_ACTION                  ; 1400
                          7E     F9      A0  3C 00025       MOVZWL  -7(R0), -(SP)                    ;
                          7E     F7      A2  3C 00029       MOVZWL  -9(R2), -(SP)                    ;
                 00000000G    00         C   FB 0002D       CALLS   #3, LNK$FNDPSCMSK                ;
                50        FC  A3             63  C3 00034   SUBL3   ISECT_BASE, CURRENT_BASE, R0     ; 1401
                          50     01FF    C0  9E 00039       MOVAB   511(R0), R0                      ;
                52        50 00000200    8F  C7 0003E       DIVL3   #512, R0, ISECT_BLKS             ;
                FFFFFFFF  8F         14  B3  D1 00046       CMPL    @ISCTDESC, #-1                   ; 1402
                                         09  12 0004E       BNEQ    1$                               ;
                                         52  DD 00050       PUSHL   ISECT_BLKS                       ; 1403
```

```
                00000000V  EF                01  FB 00052              CALLS   #1, BUILDISD                          ;
                        50 00000000G          00  D0 00059 1$:          MOVL    LNK$GL_CURCLU, R0                    ; 1404
                        50  A0                 52  C0 00060              ADDL2   ISECT_BLKS, 80(R0)                   ; 1405
                            63        FC      A3  D0 00064              MOVL    CURRENT_BASE, ISECT_BASE             ; 1406
                                                  04 00068              RET                                          ; 1408

; Routine Size:  105 bytes,    Routine Base:  $CODE$ + 0904
```

```
;  924          1409  1 routine buildisd (npages) : novalue =
;  925          1410  2 begin
;  926          1411  2 !
;  927          1412  2 !       ROUTINE TO ALLOCATE AN IMAGE SECTION DESCRIPTOR AND
;  928          1413  2 !       FILL IN ITS FIELDS, THEN LINK ON TO THE ISD LIST
;  929          1414  2 !       NPAGES = NUMBER OF PAGES TO ALLOCATE IN SECTION.
;  930          1415  2 !
;  931          1416  2 bind
;  932          1417  2         hdrisd = isctdesc[isl$t_hdrisd] : block[,byte],         ! SET POINTER TO PART FOR HEADER
;  933          1418  2         matchctrl = .isectgentbl[.isect-1,isc$b_matctl],        ! ISD MATCH CONTROL
;  934          1419  2         isectflags = .isectgentbl[.isect-1,isc$b_flags],        ! I-SECT FLAGS
;  935          1420  2         isectpscflags = .isectgentbl[.isect-1, isc$w_match],    ! MATCH CONTROL FLAGS FOR ISECT
;  936          1421  2         iscpfc = .isectgentbl[.isect-1,isc$b_pfc],              ! PAGE FAULT CLUSTER PARAMETER
;  937          1422  2         isdsize = .isectgentbl[.isect-1,isc$b_size],            ! ISD SIZE PARAMETER
;  938          1423  2         isdtype = .isectgentbl[.isect-1,isc$b_code];            ! ISECTION TYPE
;  939          1424  2
;  940          1425  2 if isdsize eql 0
;  941          1426  2     then return;                                               ! SKIP NULL ISECT DESCRIPTOR
;  942          1427  2 isctdesc[isl$l_nxtisd] = 0;                                   ! ENSURE END OF LIST
;  943          1428  2 isctdesc[isl$l_previsd] = .previsect;                         ! SET POINTER TO PREVIOUS ISECT
;  944          1429  2 previsect = .isctdesc;                                        ! SET THIS AS NEW PREVIOUS
;  945          1430  2 isctdesc[isl$l_cludsc] = .lnk$gl_curclu;                     ! SET POINTER TO CLUSTER DESCRIPTOR
;  946          1431  2 isctdesc[isl$l_bufadr] = 0;                                  ! NO BUFFER FOR ITS BINARY YET
;  947          1432  2 isctdesc[isl$l_bufend] = 0;                                  ! SO ZERO THE QUADWORD
;  948          1433  2 lnk$gw_nisects = .lnk$gw_nisects + 1;                        ! COUNT THIS ISECTION
;  949          1434  2 hdrisd[isd$w_size] = isdsize;                                ! AND COPY INTO ISD
;  950          1435  2 hdrisd[isd$w_pagcnt] = .npages;                             ! INSERT PAGE COUNT
;  951          1436  2 hdrisd[isd$l_vpnpfc] = .isect_base ^-9;                      ! INSERT VIRTUAL PAGE ADDRESS
;  952          1437  2 if (hdrisd[isd$b_pfc] = minu(.npages,.lnk$gl_curclu[clu$b_pfc])) eql 0 ! IF NO PFC GIVEN FOR CLUSTER
;  953          1438  2 then hdrisd[isd$b_pfc] = minu(iscpfc,.npages);               ! PFC IS MINIMUM OF DEFAULT
;  954          1439  2                                                             ! VALUE AND NUMBER OF PAGES
;  955          1440  2 hdrisd[isd$l_flags] = isectflags;                           ! INSERT FLAGS
;  956          1441  2 hdrisd[isd$v_matchctl] = matchctrl;                         ! INSERT THE MATCH CONTROL FIELD
;  957          1442  2 hdrisd[isd$b_type] = isdtype;                               ! AND TYPE
;  958          1443  2 hdrisd[isd$l_vbn] = 0;                                      ! SET NO RELATIVE DISK BLOCK YET ALLOCATED
;  959          1444  2 lnk$gl_curclu[clu$l_lstisd] = .isctdesc;                    ! WHICH WILL BE ADJUSTED LATER
;  960          1445  2 lnk$gl_curclu[clu$l_nisects] = .lnk$gl_curclu[clu$l_nisects] + 1;! COUNT THIS ISECT IN CLUSTER
;  961          1446  2 if (isectpscflags and gps$m_vec) neq 0                      ! IF THIS IS A VECTOR ISECT
;  962          1447  2         then hdrisd[isd$v_vector] = true;                    ! THEN SET THE FLAG
;  963          1448  2 return;                                                     ! SAVE END POINTER
;  964          1449  1 end;              ! OF ISECT DESCRIPTOR BUILDER


                                007C 00000 BUILDISD:
                                                .WORD   Save R2,R3,R4,R5,R6                             ; 1409
56 00000000'   EF   9E 00002                    MOVAB   PREVISECT, R6                                   ;
55         04   A6   D0 00009                    MOVL    ISCTDESC, R5                                    ; 1417
52         18   A5   9E 0000D                    MOVAB   24(R5), R2                                      ;
50         FC   A6   D0 00011                    MOVL    ISECTGENTBL, R0                                 ; 1418
54         F8   A6   9A 00015                    MOVZBL  ISECT, R4                                       ;
54              09   C4 00019                    MULL2   #9, R4                                          ;
           FC A440   95 0001C                    TSTB    -4(R4)[R0]                                      ; 1425
                01   12 00020                    BNEQ    1$                                              ;
                     04 00022                    RET                                                     ;
```

```
                                                65  D4 00023 1$:    CLRL    (R5)                                    ; 1427
                                        04  A5      66  D0 00025        MOVL    PREVISECT, 4(R5)                        ; 1428
                                            66      55  D0 00029        MOVL    R5, PREVISECT                           ; 1429
                                            51 00000000G    00  D0 0002C        MOVL    LNK$GL_CURCLU, R1               ; 1430
                                        10  A5          51  D0 00033        MOVL    R1, 16(R5)
                                                08  A5  7C 00037        CLRQ    8(R5)                                   ; 1431
                                             00000000'  EF  B6 0003A        INCW    LNK$GW_NISECTS                          ; 1433
                                            62      FC A440  9B 00040        MOVZBW  -4(R4)[R0], (R2)                        ; 1434
                                        02  A2      04  AC  B0 00045        MOVW    NPAGES, 2(R2)                           ; 1435
                                04  A2      F0  A6      F7  8F 78 0004A        ASHL    #-9, ISECT_BASE, 4(R2)                  ; 1436
                                            53      04  AC  D0 00051        MOVL    NPAGES, R3                              ; 1437
                        53      5A  A1          08          00  ED 00055        CMPZV   #0, #8, 90(R1), R3
                                                            04  1E 0005B        BGEQU   2$
                                            53      5A  A1  9A 0005D        MOVZBL  90(R1), R3
                                        07  A2          53  90 00061 2$:     MOVB    R3, 7(R2)
                                                        53  D5 00065        TSTL    R3
                                                        13  12 00067        BNEQ    4$
                                            53      FD A440  9A 00069        MOVZBL  -3(R4)[R0], R3                          ; 1438
                                        04  AC          53  D1 0006E        CMPL    R3, NPAGES
                                                        04  1B 00072        BLEQU   3$
                                            53      04  AC  D0 00074        MOVL    NPAGES, R3
                                        07  A2          53  90 00078 3$:     MOVB    R3, 7(R2)
                                        08  A2      FE A440  9A 0007C 4$:     MOVZBL  -2(R4)[R0], 8(R2)                       ; 1440
                                                FF A440  9F 00082        PUSHAB  -1(R4)[R0]                              ; 1441
                    08  A2          03          04          9E  F0 00086        INSV    @(SP)+, #4, #3, 8(R2)
                                        0B  A2      FB A440  90 0008C        MOVB    -5(R4)[R0], 11(R2)                      ; 1442
                                                0C  A2  D4 00092        CLRL    12(R2)                                  ; 1443
                                        1C  A1          55  D0 00095        MOVL    R5, 28(R1)                              ; 1444
                                                48  A1  D6 00099        INCL    72(R1)                                  ; 1445
                                04          F9 A440          09  E1 0009C        BBC     #9, -7(R4)[R0], 5$              ; 1446
                                        0A  A2          02  88 000A2        BISB2   #2, 10(R2)                              ; 1447
                                                        04 000A6 5$:     RET                                             ; 1449

; Routine Size:  167 bytes,    Routine Base:  $CODE$ + 096D

;  965          1450  0 end eludom


                                                                        .EXTRN  LIB$SIGNAL, LIB$STOP

;                                       PSECT SUMMARY
;
;       Name                    Bytes                           Attributes
;
;  $GLOBAL$                       48  NOVEC,  WRT,  RD ,NOEXE,NOSHR,  LCL,  REL,  CON,NOPIC,ALIGN(2)
;  $PLIT$                        420  NOVEC,NOWRT,  RD ,NOEXE,NOSHR,  LCL,  REL,  CON,NOPIC,ALIGN(2)
;  $OWN$                          36  NOVEC,  WRT,  RD ,NOEXE,NOSHR,  LCL,  REL,  CON,NOPIC,ALIGN(2)
;  $CODE$                       2580  NOVEC,NOWRT,  RD ,  EXE,NOSHR,  LCL,  REL,  CON,NOPIC,ALIGN(2)
;  .  ABS  .                       0  NOVEC,NOWRT,NORD ,NOEXE,NOSHR,  LCL,  ABS,  CON,NOPIC,ALIGN(0)


;                               Library Statistics
```

```
;
;                                  -------- Symbols --------      Pages       Processing
;       File                       Total    Loaded   Percent      Mapped      Time
;
;  _$255$DUA28:[SYSLIB]LIB.L32;1    18619        46         0        1000        00:01.9
;  _$255$DUA28:[LINKER.OBJ]DATBAS.L32;1   538        76        14          28        00:00.8
```

```
;                                   COMMAND QUALIFIERS

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

; Size:         2580 code + 504 data bytes
; Run Time:        01:06.0
; Elapsed Time:    02:27.5
; Lines/CPU Min:    1317
; Lexemes/CPU-Min: 37661
; Memory Used:  319 pages
; Compilation Complete
```

```
Mo
--
RD
RD
FI
SY
AL
AC
LI
LI

LI
LI
LI
LI
LI
SY
ST
ST
ST
SY
LI
SY
```

```
Mo
--
SY
```

LNKPROLTB
LIS

LNKSYMTBL
LIS

LNKSYMOUT
LIS

LNKVMALLO
LIS

LNKPSCTBL
LIS

LNKPROSHR
LIS

LNKSTATSO
LIS

LNKUMCTRL
LIS

RDBSHR
LIS

SYSACLSRV
LIS

RDBDISP
LIS

LOADSS

SECURESHR
MAP

FINDHELD
LIS