



```

LL      NN      NN  KK      KK      GGGGGGGG  EEEEEEEEE  TTTTTTTTT  CCCCCC  MM      MM  DDDDDDD
LL      NN      NN  KK      KK      GGGGGGGG  EEEEEEEEE  TTTTTTTTT  CCCCCC  MM      MM  DDDDDDD
LL      NN      NN  KK      KK      GG          EE          TT          CC          MMMM  MMMM  DD      DD
LL      NN      NN  KK      KK      GG          EE          TT          CC          MM  MM  MM  DD      DD
LL      NN      NN  KK      KK      GG          EE          TT          CC          MM  MM  MM  DD      DD
LL      NN      NN  KK      KK      GG          EE          TT          CC          MM  MM  MM  DD      DD
LL      NN      NN  KKKKKK  KK      GG          EEEEEEE    TT          CC          MM  MM  MM  DD      DD
LL      NN      NN  KKKKKK  KK      GG          EEEEEEE    TT          CC          MM  MM  MM  DD      DD
LL      NN      NN  KKKKKK  KK      GG          EEEEEEE    TT          CC          MM  MM  MM  DD      DD
LL      NN      NN  KKKKKK  KK      GG          EEEEEEE    TT          CC          MM  MM  MM  DD      DD
LL      NN      NN  KK      KK      GG          EE          TT          CC          MM  MM  MM  DD      DD
LL      NN      NN  KK      KK      GG          EE          TT          CC          MM  MM  MM  DD      DD
LL      NN      NN  KK      KK      GG          EE          TT          CC          MM  MM  MM  DD      DD
LLLLLLLL  NN      NN  KK      KK      GGGGGG  EEEEEEEEE  TT          CCCCCC  MM      MM  DDDDDDD
LLLLLLLL  NN      NN  KK      KK      GGGGGG  EEEEEEEEE  TT          CCCCCC  MM      MM  DDDDDDD

```

```

LL      IIIIII  SSSSSSS
LL      IIIIII  SSSSSSS
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
LLLLLLLL  IIIIII  SSSSSSS
LLLLLLLL  IIIIII  SSSSSSS

```

1  
2  
3  
4  
5  
6  
7  
8  
9  
10  
11  
12  
13  
14  
15  
16  
17  
18  
19  
20  
21  
22  
23  
24  
25  
26  
27  
28  
29  
30  
31  
32  
33  
34  
35  
36  
37  
38  
39  
40  
41  
42  
43  
44  
45  
46  
47  
48  
49  
50  
51  
52  
53  
54  
55  
56  
57

```
0001 0 module lnk_getcmdinp
0002 0      (ident='V04-000'
0003 0      ,addressing_mode
0004 0      (external = general
0005 0      ,nonexternal = long_relative
0006 0      )
0007 0      ) =
0008 1 begin
0009 1
0010 1
0011 1 *****
0012 1 *
0013 1 *  COPYRIGHT (c) 1978, 1980, 1982, 1984 BY
0014 1 *  DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASSACHUSETTS.
0015 1 *  ALL RIGHTS RESERVED.
0016 1 *
0017 1 *  THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND COPIED
0018 1 *  ONLY IN ACCORDANCE WITH THE TERMS OF SUCH LICENSE AND WITH THE
0019 1 *  INCLUSION OF THE ABOVE COPYRIGHT NOTICE. THIS SOFTWARE OR ANY OTHER
0020 1 *  COPIES THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY
0021 1 *  OTHER PERSON. NO TITLE TO AND OWNERSHIP OF THE SOFTWARE IS HEREBY
0022 1 *  TRANSFERRED.
0023 1 *
0024 1 *  THE INFORMATION IN THIS SOFTWARE IS SUBJECT TO CHANGE WITHOUT NOTICE
0025 1 *  AND SHOULD NOT BE CONSTRUED AS A COMMITMENT BY DIGITAL EQUIPMENT
0026 1 *  CORPORATION.
0027 1 *
0028 1 *  DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE OR RELIABILITY OF ITS
0029 1 *  SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL.
0030 1 *
0031 1 *
0032 1 *****
0033 1
0034 1
0035 1
0036 1
0037 1
0038 1 ++
0039 1
0040 1 MODULE:      LNK_GETCMDINP
0041 1
0042 1 FACILITY:    LINKER
0043 1
0044 1 ABSTRACT:    Command Language Interface For The LINK Command
0045 1
0046 1 HISTORY:
0047 1
0048 1 VERSION:    X01.00
0049 1
0050 1 AUTHOR:      T.J. PORTER 03-MAY-77
0051 1
0052 1 MODIFIED BY:
0053 1
0054 1          V03-026 ADE0005      Alan D. Eldridge      24-Jul-1984
0055 1          Save proper STV value on LIB$FIND_FILE error.
0056 1
0057 1          V03-025 ADE0004      Alan D. Eldridge      30-Jun-1984
```

```

58 0058 1 Fix bug in /NOEXE.
59 0059 1
60 0060 1 V03-024 ADE0003 Alan D. Eldridge 22-Jun-1984
61 0061 1 Add LNK$GB_LOCNOV_MAP, LNK$GB_LOCNOV_EXE, and LNK$GB_LOCNOV_SYM
62 0062 1 for proper "Grammar Rule" support for constructing output file
63 0063 1 names. These are really flags which, when set, indicate that
64 0064 1 appropriate switch had was used locally but had no value
65 0065 1 associated with it
66 0066 1
67 0067 1 V03-023 ADE0002 Alan D. Eldridge 14-Apr-1984
68 0068 1 More fixes to positional qualifier support.
69 0069 1
70 0070 1 V03-022 ADE0001 Alan D. Eldridge 4-Apr-1984
71 0071 1 Convert routine INPUTFILE to use LIB$FIND_FILE rather than
72 0072 1 $SEARCH and $PARSE.
73 0073 1
74 0074 1 Add support for proper defaulting when /EXE, /MAP, or /SYM
75 0075 1 are used as local, rather than command, qualifiers.
76 0076 1
77 0077 1 V03-021 MCN0001 Maria del C. Nasr 20-Jul-1983
78 0078 1 Determine if the options file is being read from
79 0079 1 SYSS$INPUT or not. This is needed so that there is
80 0080 1 no duplication of input line on output when SET
81 0081 1 VERIFY is on.
82 0082 1
83 0083 1 V03-020 JWT0125 Jim Teague 21-Jun-1983
84 0084 1 Allow UNIV=* when accompanied by /NOEXE. It can thus
85 0085 1 be used only for STB generation from shareable image
86 0086 1 links.
87 0087 1
88 0088 1 V03-019 JWT0119 Jim Teague 11-May-1983
89 0089 1 Tweak new CLI stuff again... adjust map qualifiers.
90 0090 1
91 0091 1 V03-018 JWT0117 Jim Teague 29-Apr-1983
92 0092 1 /NOEXE is equivalent to /EXE=NL:
93 0093 1
94 0094 1 V03-017 JWT0113 Jim Teague 20-Apr-1983
95 0095 1 Call $getjpi to get number of open files left.
96 0096 1
97 0097 1 V03-016 JWT0108 Jim Teague 13-Apr-1983
98 0098 1 More adjustments to new CLI interface.
99 0099 1
100 0100 1 V03-015 JWT0107 Jim Teague 07-Apr-1983
101 0101 1 Fix problems with /MAP qualifier defaults.
102 0102 1
103 0103 1 V03-014 JWT0099 Jim Teague 03-Mar-1983
104 0104 1 New cli interface.
105 0105 1
106 0106 1 V03-013 JWT0059 Jim Teague 19-Oct-1982
107 0107 1 Corrected problem with basing system images explicitly.
108 0108 1
109 0109 1 V03-012 JWT0054 Jim Teague 15-Sep-1982
110 0110 1 INPUTFILE is now able to more intelligently specify
111 0111 1 a file to close.
112 0112 1
113 0113 1 V03-011 JWT0049 Jim Teague 06-Aug-1982
114 0114 1 Call LNK$CLOSEFILE to close files, rather than $CLOSE.

```

LNK\_GETCMDINP  
V04=000

N 4  
15-Sep-1984 23:57:45 VAX-11 Bliss-32 V4.0-742  
14-Sep-1984 12:40:30 [LINKER.SRC]LNKGETCMD.B32;1

Page 3  
(1)

LNK  
V04

: 115 0115 1 :  
: 116 0116 1 :  
: 117 0117 1 :  
: 118 0118 1 :  
: 119 0119 1 :  
: 120 0120 1 :  
: 121 0121 1 :  
: 122 0122 1 :--

V03-010 JWT0044 Jim Teague 30-Jul-1982  
Open file performance boost.  
V03-009 BLS0166 Benn Schreiber 29-Mar-1982  
Set SQ0 when opening input files

000

```
124 0123 1 |  
125 0124 1 | ++  
126 0125 1 |  
127 0126 1 | FUNCTIONAL DESCRIPTION:  
128 0127 1 |  
129 0128 1 | This module contains all the routines and data structures to process the  
130 0129 1 | input command parameters and options. There is a global routine:  
131 0130 1 |  
132 0131 1 |     LNK$GETCMD (arglist)  
133 0132 1 |  
134 0133 1 | Where arglist is the address of the original argument list with which the  
135 0134 1 | linker was called. This argument list is used to perform calls back to  
136 0135 1 | the CLI for result parsing.  
137 0136 1 |  
138 0137 1 | This module also contains parameter request blocks which are passed with  
139 0138 1 | each call to the CLI, as well as parameter qualifier description blocks and  
140 0139 1 | a set of action routines which are called by the CLI on the appropriate  
141 0140 1 | conditions.  
142 0141 1 |  
143 0142 1 | A global routine LNK$ALLOFDB is also included in this module to allocate  
144 0143 1 | and initialize file descriptors and resultant file name string areas. It is  
145 0144 1 | described in detail below.  
146 0145 1 |  
147 0146 1 | --
```

```

149 0147 1 forward routine
150 0148 1 lnk$srcpscdef,
151 0149 1 lnk$collectpsct : novalue,
152 0150 1 setuserlib,
153 0151 1 clierror : novalue,
154 0152 1 call_tparse,
155 0153 1 lnk$allocluster : novalue,
156 0154 1 lnk$allofdb : novalue,
157 0155 1 outputfile : novalue,
158 0156 1 inputfile : novalue,
159 0157 1 specialimage : novalue,
160 0158 1 lnk$setlibrin : novalue,
161 0159 1 lnk$insert_clu,
162 0160 1 optionfile : novalue,
163 0161 1 get_image : novalue,
164 0162 1 get_map : novalue,
165 0163 1 get_sym : novalue,
166 0164 1 get_input,
167 0165 1 setup_inrelnam ;
168 0166 1
169 0167 1 LIBRARY
170 0168 1 'STARLETL32' ;
171 0169 1 require
172 0170 1 'PREFIX' ;
173 0285 1 library
174 0286 1 'DATBAS' ;
175 0287 1 library
176 0288 1 'CLIMACL32' ;
177 0289 1 library
178 0290 1 'TPAMACL32' ;
179 0291 1
180 0292 1 Set up string descriptors for qualifers
181 0293 1
182 P 0294 1 sd ('BRIEF', 'CONTIGUOUS', 'CROSS_REFERENCE', 'DEBUG', 'EXECUTABLE'
183 P 0295 1 'FULL', 'HEADER', 'MAP', 'POIMAGÉ', 'PROTECT'
184 P 0296 1 'SHAREABLE', 'SYMBOL_TABLE', 'SYSLIB', 'SYSSHR', 'SYSTEM'
185 P 0297 1 'TRACE', 'USERLIBRARY', 'INCLUDE', 'LIBRARY', 'OPTIONS'
186 P 0298 1 'SELECTIVE_SEARCH', 'PI'
187 0299 1 ) ;
188 0300 1
189 M 0301 1 macro copy_desc (src, dst) =
190 0302 1 begin
191 0303 1 dst [dsc$w_length] = .src [dsc$w_length] ;
192 0304 1 dst [dsc$b_class] = .src [dsc$b_class] ;
193 0305 1 dst [dsc$b_dtype] = .src [dsc$b_dtype] ;
194 0306 1 dst [dsc$a_pointer] = .src [dsc$a_pointer] ;
195 0307 1 end
196 0308 1 % ;
197 0309 1
198 0310 1 external literal
199 0311 1 cli$negated,
200 0312 1 cli$locneg,
201 0313 1 cli$defaulted,
202 0314 1 cli$locpres,
203 0315 1 cli$present,
204 0316 1 lin$univstar,
205 0317 1 lin$closein,

! Look for psectname in psect def by opt list
! Perform PSECT collection/definition
! Selectively set logical name tables to search
! called for all cli errors
! Call tparse to parse a value
! Allocate a cluster descriptor
! Allocate file descriptors and resultant string areas
! Called for all output files
! Called for each input file
! Called for system and shareable images
! Called for library files
! Insert cluster into cluster tree

! Process image qualifiers
! Process map qualifiers
! Process /SYM
! Process input file name
! Setup related name for output files

! System structure def'ns
! Utility macros etc.
! Internal data base definitions
! Command language structures
! Tparse macros

```

```

: 206      0318 1      lln$_cmdsyntax,      ! Command syntax error
: 207      0319 1      lln$_firstmod,      ! First module requires extraction since library
: 208      0320 1      lln$_illqualval,    ! Illegal qualifier value
: 209      0321 1      lln$_linerr,      ! Display segment in error
: 210      0322 1      lln$_nomods,      ! No input modules specified
: 211      0323 1      lln$_openin,      ! Error opening input file
: 212      0324 1      lln$_tracign ;    ! Traceback supression overridden
: 213      0325 1
: 214      0326 1      external routine
: 215      0327 1      str$copy_r,
: 216      0328 1      lib$cvr_otb,
: 217      0329 1      lib$cvr_dtb,
: 218      0330 1      lib$cvr_htb,
: 219      0331 1      cli$present,
: 220      0332 1      cli$get_value,
: 221      0333 1      lib$find_file,      ! Search list compatible related name processing
: 222      0334 1      lib$find_file_end,  ! Cleanup after LIB$FIND_FILE
: 223      0335 1      lib$insert_tree,   ! Insert into balanced binary tree
: 224      0336 1      lib$lookup_tree,   ! Lookup entry in balanced binary tree
: 225      0337 1      lib$parse,        ! Table driven parser
: 226      0338 1      lln$findpsectnam, ! Lookup/enter psect names
: 227      0339 1      lln$alloblk      : novalue, ! Allocate dynamic memory
: 228      0340 1      lln$dealkb      : novalue, ! Deallocate dynamic memory
: 229      0341 1      lln$procopts,     ! Option file processor
: 230      0342 1      lln$filnamdsc,    ! Return string descriptor for file name
: 231      0343 1      lln$insudfsym,    ! Insert symbol descriptor in undefined symbol list
: 232      0344 1      lln$insert,      ! Symbol table insert routine
: 233      0345 1      lln$closefile,    ! Routine to close a file
: 234      0346 1      lln$openlib,     ! Routine to open a library
: 235      0347 1      lln$search ;     ! Symbol table search routine
: 236      0348 1      !
: 237      0349 1      external
: 238      0350 1      lln$gl_inclst    : ref block [,byte], ! Include list descriptor
: 239      0351 1      lln$gl_stack    : word,           ! Stack size
: 240      0352 1      lln$gl_curfil   : ref block [,byte],
: 241      0353 1      lln$gl_ctlmsk   : block [,byte], ! Control flags
: 242      0354 1      lln$gl_lastclu  : ref block [,byte], ! Pointer to last cluster in list
: 243      0355 1      lln$gl_clulst   : vector [2],    ! Cluster descriptor listhead
: 244      0356 1      lln$gl_defclu   : block [,byte], ! Default cluster descriptor
: 245      0357 1      lln$gl_curclu   : ref block [,byte], ! Current cluster descriptor
: 246      0358 1      lln$gl_clutree  : ref block [,byte], ! Listhead for cluster tree
: 247      0359 1      lln$gl_fmvlst   : ref block [,byte], ! Free virtual memory listhead
: 248      0360 1      lln$gl_imgfil   : ref block [,byte], ! Pointer to image file descriptor slot
: 249      0361 1      lln$gl_mapfil   : ref block [,byte], ! Pointer to the map file descriptor
: 250      0362 1      lln$gl_nclusters : word,           ! Number of clusters allocated
: 251      0363 1      lln$gl_nsymbols : word,           ! Number of symbols
: 252      0364 1      lln$gl_symfil   : ref block [,byte], ! Pointer to symbol table output file
: 253      0365 1      lln$gl_cclulst,  ! List head (singly linked)
: 254      0366 1      ! For clusters in collect option
: 255      0367 1      lln$gl_pscdfilst ; ! Singly linked listhead
: 256      0368 1      ! For psects defined with psect option
: 257      0369 1      !
: 258      0370 1      Bind firstcluster = lln$gl_clulst [0] : ref block [,byte], ! Pointer to first
: 259      0371 1      null_device = $descriptor ('NL:') : block [,byte] ; ! Null device for /NOEXE
: 260      0372 1      !
: 261      0373 1      global
: 262      0374 1      lln$gl_filesleft, ! Number of files left to open

```



```

263 0375 1 lnk$gl_ulibmask,      ! Mask of enabled name tables to search
264 0376 1 lnk$gl_inrelnam,     ! Address of nam block of first input file
265 0377 1 lnk$gl_relnam_exe,  ! Address of related nam block for .EXE
266 0378 1 lnk$gl_relnam_map,   ! Address of related nam block for .MAP
267 0379 1 lnk$gl_relnam_sym,   ! Address of related nam block for .STB
268 0380 1 lnk$gb_locnov_exe  : byte,      ! Set if /EXE used locally with no specified value
269 0381 1 lnk$gb_locnov_map   : byte,      ! Set if /MAP used locally with no specified value
270 0382 1 lnk$gb_locnov_sym  : byte,      ! Set if /SYM used locally with no specified value
271 0383 1 lnk$gb_re*cmd_spl  : byte,      ! Spare, used for alignment
272 0384 1 lnk$gl_sysinput_flag, ! Set when option file is sys$input
273 0385 1 lnk$gl_cliwork ;    ! Pointer to work area for cli
274 0386 1 !
275 0387 1 global bind
276 0388 1 lnk$gt_jpilst = $itmlst_uplit ((itmcod = jpi$_filcnt, bufadr = lnk$gl_filesleft)),
277 0389 1
278 0390 1 lnk$gt_imgsta = cstring ('SYS$IMGSTA'), ! Name of debugger activator
279 0391 1 libfiltype = uplit (byte ('.OLB')), ! Library file type
280 0392 1 defiltype = uplit (byte ('.OBJ')), ! Default file type
281 0393 1 shrfiltype = uplit (byte ('.EXE')), ! Shr img file type
282 0394 1
283 0395 1 map
284 0396 1 lnk$gl_ulibmask : bitvector [] ; ! Pretend its a bitvector
285 0397 1 !
286 0398 1 ! OWN STORAGE
287 0399 1 !
288 0400 1 !
289 0401 1 literal
290 0402 1 imgtabsiz = 7,
291 0403 1 system_regor = %x'80000000',
292 0404 1 rsa_allosize = nam$*_maxrss*3/2, ! Rsa allocation quantity
293 0405 1 typelen = 4 ; ! Length of file type string
294 0406 1 !
295 0407 1 own
296 0408 1 univstar_valid : initial (false), ! UNIV=* valid only with /NOEXE
297 0409 1 local_qualifier : initial (false), ! Set to TRUE when /MAP, /EXE, and
298 0410 1 ! /SYM are parsed as local qualifiers
299 0411 1 listptr, ! Dynamic include list ptr
300 0412 1 dflt_name : vector [nam$*_maxrss, byte], ! Default input related name text
301 0413 1 dflt_inrel : block [nam$*_bln, byte], ! Default input related name nam$
302 0414 1 last_fnm_desc : ref block [,byte] initial (0), ! Last input file descriptor
303 0415 1 rlf_context : ref block [,byte] initial (0), ! Related filename context
304 0416 1 an_open_fdb : initial (0),
305 0417 1 debug_desc : dynamic_descriptor,
306 0418 1 module_desc : dynamic_descriptor,
307 0419 1 share_desc : dynamic_descriptor, ! /SHAREABLE descriptor name
308 0420 1 token_desc : dynamic_descriptor, ! Dynamic desc for general usage
309 0421 1 systembase, ! Base specified in /SYSTEM=NUMBER
310 0422 1 basedflag, ! Set true if /SYSTEM=NUMBER
311 0423 1 outfile, ! Ptr to ptr for the current output file
312 0424 1 nextrsa, ! Ptr to slot for next resultant string on o
313 0425 1 typeadr : initial (ch$ptr (defiltype)), ! Address of the file type string
314 0426 1 infilflgs, ! Flags of each input file
315 0427 1 tpdata, ! Data from tparse
316 0428 1 rsaremsize : byte, ! Size remaining in rsa
317 0429 1 image_qual_table : vector [ imgtabsiz, long ]
318 0430 1 initial
319 0431 1 (sd_trace, sd_contiguous, sd_p0image, sd_protect

```

```
320 0432 1      } sd_syslib, sd_sysshr, sd_header
321 0433 1
322 0434 1      bit_table : vector [imgtabsiz, byte]
323 0435 1      initial
324 0436 1      (byte ($bitposition (lnk$V_trace), $bitposition (lnk$V_contig)
325 0437 1      , $bitposition (lnk$V_p0image), $bitposition (lnk$V_protect)
326 0438 1      , $bitposition (lnk$V_syslib), $bitposition (lnk$V_sysshr)
327 0439 1      , $bitposition (lnk$V_sysheadr)
328 0440 1      ) ;
329 0441 1
330 0442 1
331 0443 1      ; Tparse tables to parse /userlib qualifier values
332 0444 1
333 0445 1 $init_state (userlib_states, userlib_keys) ;
334 0446 1
335 P 0447 1 $state (, ('SYSTEM', tpa$_exit,,1,lnk$gl_ulibmask),
336 P 0448 1 ('GROUP', tpa$_exit,,2,lnk$gl_ulibmask),
337 P 0449 1 ('PROCESS', tpa$_exit,,4,lnk$gl_ulibmask),
338 P 0450 1 ('ALL', tpa$_exit,,7,lnk$gl_ulibmask),
339 P 0451 1 ('NONE', tpa$_exit,,%x'80000000',lnk$gl_ulibmask)
340 0452 1 ) ;
341 0453 1
```

```
343 0454 1 global routine lnk$getcmd (arglist) : novalue =
344 0455 1
345 0456 1 !+
346 0457 1
347 0458 1 This routine initializes the CLI result parser then re-calls it for
348 0459 1 processing of each command parameter. It first calls for the primary output
349 0460 1 parameter (the image), then the secondary output parameter (the map),
350 0461 1 followed by the third output parameter (the symbol table file). Next it
351 0462 1 loops, calling for input parameters (primary only) until there are no more.
352 0463 1 the actions taken on presence or absence of any parameter and its qualifiers
353 0464 1 are taken in the other routines (below) which are called by CLI on the
354 0465 1 appropriate condition.
355 0466 1
356 0467 1
357 0468 1 ARGLIST is the address of the argument list with which the linker was
358 0469 1 originally called.
359 0470 1
360 0471 1 -
361 0472 2 begin
362 0473 2 map
363 0474 2     arglist : ref block [,byte] ;           ! Really a vector of longwords
364 0475 2
365 0476 2 lnk$gl_ctlmsk [lnk$v_verify] = .arglist [cli$v_verify] ;       ! Copy verify context of process
366 0477 2
367 0478 2 rsaremsize = rsa_allosize*3 ;           ! Set up the resultant string area
368 0479 2 lnk$alloblk (.rsaremsize,nextrsa) ;       ! For the file opens (triple size at first)
369 0480 2
370 0481 2
371 0482 2
372 0483 2 Loop through qualifiers that do nothing but set a bit in LNK$GL_CTLMSK...
373 0484 2
374 0485 2
375 0486 2 incr i to imgtabsize - 1
376 0487 2 do lnk$gl_ctlmsk [0,.bit_table [.i],1,0] = cli$present (.image_qual_table [.i]) ;
377 0488 2
378 0489 2 if cli$present (sd_userlibrary)           ! /USERLIBRARY
379 0490 2 then setuserlib () ;
380 0491 2
381 0492 2 if cli$present (sd_system)                 ! /SYSTEM
382 0493 2 then begin
383 0494 2     lnk$gl_ctlmsk [lnk$v_sys] = true ;       ! Set/clear bits
384 0495 2     lnk$gl_ctlmsk [lnk$v_trace] = false ;    ! and call
385 0496 2     lnk$gl_ctlmsk [lnk$v_exe] = false ;     ! specialimage
386 0497 2     specialimage () ;
387 0498 2     end
388 0499 2 else if cli$present (sd_shareable)        ! /SHAREABLE
389 0500 2 then begin
390 0501 2     lnk$gl_ctlmsk [lnk$v_shr] = true ;       ! Set/clear bits
391 0502 2     lnk$gl_ctlmsk [lnk$v_trace] = false ;    ! and call
392 0503 2     lnk$gl_ctlmsk [lnk$v_exe] = false ;     ! specialimage
393 0504 2     specialimage () ;
394 0505 2     end ;
395 0506 2
396 0507 2 if cli$present (sd_debug)                 ! If /DEBUG present,
397 0508 2 then begin                               ! then attempt to grab a value for
398 0509 2     cli$get_value (sd_debug, debug_desc) ;   ! the qualifier (no value is ok)
399 0510 2     lnk$gl_ctlmsk [lnk$v_dbg] = true ;
```

```

400 0511      end ;
401 0512
402 0513      get_image () ;           ! Go get image file name
403 0514      get_map  () ;           ! Go get map file name and qualifiers
404 0515      get_sym  () ;           ! Get symbol table file name
405 0516
406 0517      local_qualifier = true ;           ! /MAP/EXE/SYM are now local qualifiers
407 0518      while get_input ()           ! Loop while more input files...
408 0519      do begin
409 0520          get_image () ;
410 0521          get_map  () ;
411 0522          get_sym  () ;
412 0523      end ;
413 0524
414 0525
415 0526      lib$find_file_end (rlf_context) ;           ! Cleanup after input processing
416 0527
417 0528      lnk$gl_ctlmsk [lnk$v_image] = .lnk$gl_ctlmsk [lnk$v_sys] or
418 0529                                     .lnk$gl_ctlmsk [lnk$v_shr] or
419 0530                                     .lnk$gl_ctlmsk [lnk$v_exe] ;
420 0531
421 0532
422 0533      !
423 0534      ! If a MAP was requested, process MAP qualifiers. This could be done
424 0535      ! sooner since /MAP is positional and the other map qualifiers are
425 0536      ! global.
426 0537
427 0538      if .lnk$gl_ctlmsk [lnk$v_map]
428 0539      then begin
429 0540          if cli$present (sd_full)           ! Process map qualifiers
430 0541          then begin
431 0542              lnk$gl_ctlmsk [lnk$v_long] = true ;
432 0543              lnk$gl_ctlmsk [lnk$v_supsys] = false ;
433 0544          end ;
434 0545
435 0546          if cli$present (sd_brief)
436 0547          then lnk$gl_ctlmsk [lnk$v_brief] = true ;
437 0548
438 0549          if cli$present (sd_cross_reference)
439 0550          then lnk$gl_ctlmsk [lnk$v_cros] = true ;
440 0551          end ;
441 0552
442 0553      !
443 0554
444 0555      ! If /SYSTEM=number, then set the base now
445 0556
446 0557
447 0558      if .basedflag
448 0559      then
449 0560          begin
450 0561              if .lnk$gl_ctlmsk [lnk$v_sysheadr]
451 0562              then lnk$gl_defclu [clu$l_base] = (.systembase + 511) and not 511 ! Only round if /SYSTEM/HEADER
452 0563              else lnk$gl_defclu [clu$l_base] = .systembase ;
453 0564
454 0565              if .lnk$gl_defclu [clu$l_base] gequ control_region           ! Set correct allocation limits
455 0566              then if .lnk$gl_defclu [clu$l_base] gequ system_region
456 0567                  then lnk$gl_fmvlst [fvm$l_address] = system_region

```

```

457 0568 3      else      lnk$gl_fvmlst [fvmlst_address] = control_region
458 0569 3      else      lnk$gl_fvmlst [fvmlst_address] = 0 ;
459 0570 3      end ;
460 0571 2
461 0572 2
462 0573 2      Check that there was at least one input module
463 0574 2
464 0575 2
465 0576 2      if .firstcluster eql 0                                ! if no clusters allocated
466 0577 2      and
467 0578 2      .lnk$gl_defclu [clu$l_fstfdb] eql 0                ! And no files in default cluster
468 0579 2      then
469 0580 2      signal_stop (lnk$_nomods) ;                          ! There were no input modules - fatal error
470 0581 2
471 0582 2
472 0583 2
473 0584 2      Now link the debugger into the input file list if DEBUG and/or TRACEBACK
474 0585 2      was requested.
475 0586 2
476 0587 2      However suppress it if this is a system image
477 0588 2
478 0589 2
479 0590 2      if .lnk$gl_ctlmsk [lnk$v_sys] or .lnk$gl_ctlmsk [lnk$v_shr]
480 0591 2      then lnk$gl_ctlmsk [lnk$v_trace] = false ;
481 0592 2
482 0593 2      if .lnk$gl_ctlmsk [lnk$v_dbg] or .lnk$gl_ctlmsk [lnk$v_trace] ! If /DEBUG specified or /TRACE was
483 0594 2      then ! present or implied
484 0595 2      begin
485 0596 2      if not .lnk$gl_ctlmsk [lnk$v_trace] ! If only dbg then
486 0597 2      then signal (lnk$_tracign) ; ! Report conflict as info.
487 0598 2
488 0599 2      infilflgs = fdb$m_debugger ; ! Set the file flag
489 0600 2      if .debug_desc [dsc$w_length] neq 0 ! If /DEBUG=FILENAME
490 0601 4      then begin
491 0602 4      inputfile (debug_desc) ; ! Then go do it
492 0603 4      lib$find_file_end (rlf_context) ;
493 0604 4      end
494 0605 4      else begin ! Else default debugger
495 0606 4      local symnam : ref block [,byte], ! Pointer to name part of symbol block
496 0607 4      symval : ref block [,byte] ; ! Pointer to value part of symbol block
497 0608 4
498 0609 4      lnk$search (lnk$gt_imgsta,symval,symnam) ; ! Look up the symbol
499 0610 4      lnk$insert (lnk$gt_imgsta,symval,symnam) ; ! And insert it
500 0611 4      lnk$insudfsym (.symval) ; ! Insert symbol in undefined list
501 0612 4      lnk$gw_nsymbols = .lnk$gw_nsymbols + 1 ; ! Count the symbol
502 0613 3      end ;
503 0614 2      end ;
504 0615 2
505 0616 2      lnk$gl_ctlmsk [lnk$v_verify] = false ; ! Turn off verify status
506 0617 2
507 0618 2      lnk$collectpsct () ; ! Perform psect collection and definition
508 0619 2
509 0620 2      if .lnk$gl_defclu [clu$l_fstfdb] neq 0 ! If default cluster has some files
510 0621 2      then begin
511 0622 2      lnk$gl_lastclu [clu$l_nextclu] = lnk$gl_defclu ; ! Link it on to end of
512 0623 2      if (lnk$gl_defclu [clu$l_prevclu] = .lnk$gl_lastclu) eql 0 ! Cluster list
513 0624 2      then lnk$gl_defclu [clu$l_prevclu] = lnk$gl_clulst ;

```

```

: 514      0625      3      lnk$gl_lastclu      =      lnk$gl_defclu ;      ! And make it the last cluster then
: 515      0626      3      end ;
: 516      0627      3
: 517      0628      3      if .lnk$gl_ctlmsk [lnk$u_alluniv]      ! if univ=* requested, allow it only
: 518      0629      3      then if not .univstar_valid      ! if /NOEXE specified for .STB
: 519      0630      3      then begin      ! generation
: 520      0631      3          signal (lnk$univstar) ;
: 521      0632      3          lnk$gl_ctlmsk [lnk$u_alluniv] = false ;      ! otherwise disable it
: 522      0633      3      end ;
: 523      0634      3      return ;
: 524      0635      3      end ;

```

|    |    |    |    |    |    |           |             |                                  |       |            |            |            |  |
|----|----|----|----|----|----|-----------|-------------|----------------------------------|-------|------------|------------|------------|--|
|    |    |    |    |    |    |           | .TITLE      | LNK_GETCMDINP                    |       |            |            |            |  |
|    |    |    |    |    |    |           | .IDENT      | \V04-000\                        |       |            |            |            |  |
|    |    |    |    |    |    |           | .PSECT      | _LIB\$KEY1\$,NOWRT, SHR, PIC,1   |       |            |            |            |  |
|    |    |    |    |    |    |           | 00000       | ;TPASKEYSTO                      |       |            |            |            |  |
|    |    |    |    |    |    | U.2:      | .BLKB       | 0                                |       |            |            |            |  |
| 4D | 45 | 54 | 53 | 59 | 53 | 00000     | ;TPASKEYST  |                                  |       |            |            |            |  |
|    |    |    |    |    |    | U.4:      | .ASCII      | \SYSTEM\                         |       |            |            |            |  |
|    |    |    |    |    |    | FF        | 00006       | .BYTE                            | -1    |            |            |            |  |
|    |    |    |    |    |    | 00007     | ;TPASKEYSTO |                                  |       |            |            |            |  |
|    |    |    |    |    |    | U.9:      | .BLKB       | 0                                |       |            |            |            |  |
|    |    |    |    |    |    | 50        | 55          | 4F                               | 52    | 47         | 00007      | ;TPASKEYST |  |
|    |    |    |    |    |    | U.11:     | .ASCII      | \GROUP\                          |       |            |            |            |  |
|    |    |    |    |    |    | FF        | 0000C       | .BYTE                            | -1    |            |            |            |  |
|    |    |    |    |    |    | 0000D     | ;TPASKEYSTO |                                  |       |            |            |            |  |
|    |    |    |    |    |    | U.16:     | .BLKB       | 0                                |       |            |            |            |  |
| 53 | 53 | 45 | 43 | 4F | 52 | 50        | 0000D       | ;TPASKEYST                       |       |            |            |            |  |
|    |    |    |    |    |    | U.18:     | .ASCII      | \PROCESS\                        |       |            |            |            |  |
|    |    |    |    |    |    | FF        | 00014       | .BYTE                            | -1    |            |            |            |  |
|    |    |    |    |    |    | 00015     | ;TPASKEYSTO |                                  |       |            |            |            |  |
|    |    |    |    |    |    | U.23:     | .BLKB       | 0                                |       |            |            |            |  |
|    |    |    |    |    |    | 4C        | 4C          | 41                               | 00015 | ;TPASKEYST |            |            |  |
|    |    |    |    |    |    | U.25:     | .ASCII      | \ALL\                            |       |            |            |            |  |
|    |    |    |    |    |    | FF        | 00018       | .BYTE                            | -1    |            |            |            |  |
|    |    |    |    |    |    | 00019     | ;TPASKEYSTO |                                  |       |            |            |            |  |
|    |    |    |    |    |    | U.30:     | .BLKB       | 0                                |       |            |            |            |  |
|    |    |    |    |    |    | 45        | 4E          | 4F                               | 4E    | 00019      | ;TPASKEYST |            |  |
|    |    |    |    |    |    | U.32:     | .ASCII      | \NONE\                           |       |            |            |            |  |
|    |    |    |    |    |    | FF        | 0001D       | .BYTE                            | -1    |            |            |            |  |
|    |    |    |    |    |    | FF        | 0001E       | ;TPASKEYFILL                     |       |            |            |            |  |
|    |    |    |    |    |    | U.37:     | .BYTE       | -1                               |       |            |            |            |  |
|    |    |    |    |    |    |           | .PSECT      | _LIB\$STATES\$,NOWRT, SHR, PIC,1 |       |            |            |            |  |
|    |    |    |    |    |    |           | 00000       | USERLIB_STATES::                 |       |            |            |            |  |
|    |    |    |    |    |    |           | .BLKB       | 0                                |       |            |            |            |  |
|    |    |    |    |    |    | 7100      | 00000       | ;TPASTYPE                        |       |            |            |            |  |
|    |    |    |    |    |    | U.5:      | .WORD       | 28928                            |       |            |            |            |  |
|    |    |    |    |    |    | 00000000* | 00002       | ;TPASADDR                        |       |            |            |            |  |
|    |    |    |    |    |    | U.6:      | .LONG       | <<LNK\$GL_ULIBMASK-U.6>-4>       |       |            |            |            |  |
|    |    |    |    |    |    | 00000001  | 00006       | ;TPASMASK                        |       |            |            |            |  |
|    |    |    |    |    |    | U.7:      | .LONG       | 1                                |       |            |            |            |  |
|    |    |    |    |    |    | FFFF      | 0000A       | ;TPASTARGET                      |       |            |            |            |  |

```

7101 0000C U.8: .WORD -1 ;
:TPASTYPE ;
00000000* 0000E U.12: .WORD 28929 ;
:TPASACDR ;
00000002 00012 U.13: .LONG <<LNK$GL_ULIBMASK-U.13>-4> ;
:TPASMASK ;
FFFF 00016 U.14: .LONG 2 ;
:TPASTARGET ;
7102 00018 U.15: .WORD -1 ;
:TPASTYPE ;
00000000* 0001A U.19: .WORD 28930 ;
:TPASADDR ;
00000004 0001E U.20: .LONG <<LNK$GL_ULIBMASK-U.20>-4> ;
:TPASMASK ;
FFFF 00022 U.21: .LONG 4 ;
:TPASTARGET ;
7103 00024 U.22: .WORD -1 ;
:TPASTYPE ;
00000000* 00026 U.26: .WORD 28931 ;
:TPASADDR ;
00000007 0002A U.27: .LONG <<LNK$GL_ULIBMASK-U.27>-4> ;
:TPASMASK ;
FFFF 0002E U.28: .LONG 7 ;
:TPASTARGET ;
7504 00030 U.29: .WORD -1 ;
:TPASTYPE ;
00000000* 00032 U.33: .WORD 29956 ;
:TPASADDR ;
80000000 00036 U.34: .LONG <<LNK$GL_ULIBMASK-U.34>-4> ;
:TPASMASK ;
FFFF 0003A U.35: .LONG -2147483648 ;
:TPASTARGET ;
U.36: .WORD -1 ;

```

.PSECT \_LIB\$KEYOS,NOWRT, SHR, PIC,1

```

00000 USERLIB_KEYS:
00000 .BLKB 0
00000 :TPASKEY0
0000* 00000 U.1: .BLKB 0
:TPASKEY
0000* 00002 U.3: .WORD <U.2-U.1> ;
:TPASKEY
0000* 00004 U.10: .WORD <U.9-U.1> ;
:TPASKEY
0000* 00006 U.17: .WORD <U.16-U.1> ;
:TPASKEY
0000* 00008 U.24: .WORD <U.23-U.1> ;
:TPASKEY
U.31: .WORD <U.30-U.1> ;

```

.PSECT \$SPLITS,NOWRT,NOEXE,2

```

46 45 49 52 42 00000 P.AAB: .ASCII \BRIEF\ ;
00005 .BLKB 3 ;
00000005 00008 P.AAA: .LONG 5 ;
00000000' 0000C .ADDRESS P.AAB ;

```

|    |    |    |    |    |    |    |    |    |    |          |       |        |        |              |       |        |          |                   |             |   |
|----|----|----|----|----|----|----|----|----|----|----------|-------|--------|--------|--------------|-------|--------|----------|-------------------|-------------|---|
|    | 53 | 55 | 4F | 55 | 47 | 49 | 54 | 4E | 4F | 43       | 00010 | P.AAD: | .ASCII | \CONTIGUOUS\ | :     |        |          |                   |             |   |
|    |    |    |    |    |    |    |    |    |    |          | 0001A |        | .BLKB  | 2            | :     |        |          |                   |             |   |
|    |    |    |    |    |    |    |    |    |    | 0000000A | 0001C | P.AAC: | .LONG  | 10           | :     |        |          |                   |             |   |
| 45 | 43 | 4E | 45 | 52 | 45 | 46 | 45 | 52 | 5F | 53       | 53    | 4F     | 52     | 43           | 00020 | P.AAF: | .ADDRESS | P.AAD             | :           |   |
|    |    |    |    |    |    |    |    |    |    |          |       |        |        |              | 00024 | P.AAF: | .ASCII   | \CROSS_REFERENCE\ | :           |   |
|    |    |    |    |    |    |    |    |    |    |          |       |        |        |              | 00033 | P.AAE: | .BLKB    | 1                 | :           |   |
|    |    |    |    |    |    |    |    |    |    |          |       |        |        |              | 00034 | P.AAE: | .LONG    | 15                | :           |   |
|    |    |    |    |    |    |    |    |    |    |          |       |        |        |              | 00038 | P.AAF: | .ADDRESS | P.AAF             | :           |   |
|    |    |    |    |    |    |    | 47 | 55 | 42 | 45       | 44    |        |        |              | 0003C | P.AAH: | .ASCII   | \DEBUG\           | :           |   |
|    |    |    |    |    |    |    |    |    |    |          |       |        |        |              | 00041 | P.AAG: | .BLKB    | 3                 | :           |   |
|    |    |    |    |    |    |    |    |    |    |          |       |        |        |              | 00044 | P.AAG: | .LONG    | 5                 | :           |   |
|    |    |    |    |    |    |    |    |    |    |          |       |        |        |              | 00048 | P.AAH: | .ADDRESS | P.AAH             | :           |   |
| 45 | 4C | 42 | 41 | 54 | 55 | 43 | 45 | 58 | 45 |          |       |        |        |              | 0004C | P.AAJ: | .ASCII   | \EXECUTABLE\      | :           |   |
|    |    |    |    |    |    |    |    |    |    |          |       |        |        |              | 00056 | P.AAI: | .BLKB    | 2                 | :           |   |
|    |    |    |    |    |    |    |    |    |    |          |       |        |        |              | 00058 | P.AAI: | .LONG    | 10                | :           |   |
|    |    |    |    |    |    |    |    |    |    |          |       |        |        |              | 0005C | P.AAJ: | .ADDRESS | P.AAJ             | :           |   |
|    |    |    |    |    |    |    | 4C | 4C | 55 | 46       |       |        |        |              | 00060 | P.AAL: | .ASCII   | \FULL\            | :           |   |
|    |    |    |    |    |    |    |    |    |    |          |       |        |        |              | 00064 | P.AAK: | .LONG    | 4                 | :           |   |
|    |    |    |    |    |    |    |    |    |    |          |       |        |        |              | 00068 | P.AAL: | .ADDRESS | P.AAL             | :           |   |
|    |    |    |    |    |    |    | 52 | 45 | 44 | 41       | 45    | 48     |        |              | 0006C | P.AAN: | .ASCII   | \HEADER\          | :           |   |
|    |    |    |    |    |    |    |    |    |    |          |       |        |        |              | 00072 | P.AAM: | .BLKB    | 2                 | :           |   |
|    |    |    |    |    |    |    |    |    |    |          |       |        |        |              | 00074 | P.AAM: | .LONG    | 6                 | :           |   |
|    |    |    |    |    |    |    |    |    |    |          |       |        |        |              | 00078 | P.AAN: | .ADDRESS | P.AAN             | :           |   |
|    |    |    |    |    |    |    |    |    |    |          |       |        |        |              | 0007C | P.AAP: | .ASCII   | \MAP\             | :           |   |
|    |    |    |    |    |    |    |    |    |    |          |       |        |        |              | 0007F | P.AAD: | .BLKB    | 1                 | :           |   |
|    |    |    |    |    |    |    |    |    |    |          |       |        |        |              | 00080 | P.AAD: | .LONG    | 3                 | :           |   |
|    |    |    |    |    |    |    |    |    |    |          |       |        |        |              | 00084 | P.AAP: | .ADDRESS | P.AAP             | :           |   |
|    |    |    |    |    |    |    | 45 | 47 | 41 | 4D       | 49    | 30     | 50     |              | 00088 | P.AAR: | .ASCII   | \POIMAGE\         | :           |   |
|    |    |    |    |    |    |    |    |    |    |          |       |        |        |              | 0008F | P.AAQ: | .BLKB    | 1                 | :           |   |
|    |    |    |    |    |    |    |    |    |    |          |       |        |        |              | 00090 | P.AAQ: | .LONG    | 7                 | :           |   |
|    |    |    |    |    |    |    |    |    |    |          |       |        |        |              | 00094 | P.AAR: | .ADDRESS | P.AAR             | :           |   |
|    |    |    |    |    |    |    | 54 | 43 | 45 | 54       | 4F    | 52     | 50     |              | 00098 | P.AAT: | .ASCII   | \PROTECT\         | :           |   |
|    |    |    |    |    |    |    |    |    |    |          |       |        |        |              | 0009F | P.AAS: | .BLKB    | 1                 | :           |   |
|    |    |    |    |    |    |    |    |    |    |          |       |        |        |              | 000A0 | P.AAS: | .LONG    | 7                 | :           |   |
|    |    |    |    |    |    |    |    |    |    |          |       |        |        |              | 000A4 | P.AAT: | .ADDRESS | P.AAT             | :           |   |
|    |    |    |    |    |    |    | 45 | 4C | 42 | 41       | 45    | 52     | 41     | 48           | 53    | 000A8  | P.AAV:   | .ASCII            | \SHAREABLE\ | : |
|    |    |    |    |    |    |    |    |    |    |          |       |        |        |              | 000B1 | P.AAU: | .BLKB    | 3                 | :           |   |
|    |    |    |    |    |    |    |    |    |    |          |       |        |        |              | 000B4 | P.AAU: | .LONG    | 9                 | :           |   |
|    |    |    |    |    |    |    |    |    |    |          |       |        |        |              | 000B8 | P.AAV: | .ADDRESS | P.AAV             | :           |   |
| 45 | 4C | 42 | 41 | 54 | 5F | 4C | 4F | 42 | 4D | 59       | 53    |        |        |              | 000BC | P.AAX: | .ASCII   | \SYMBOL_TABLE\    | :           |   |
|    |    |    |    |    |    |    |    |    |    |          |       |        |        |              | 000C8 | P.AAW: | .LONG    | 12                | :           |   |
|    |    |    |    |    |    |    |    |    |    |          |       |        |        |              | 000CC | P.AAX: | .ADDRESS | P.AAX             | :           |   |
|    |    |    |    |    |    |    | 42 | 49 | 4C | 53       | 59    | 53     |        |              | 000D0 | P.AAZ: | .ASCII   | \SYSLIB\          | :           |   |
|    |    |    |    |    |    |    |    |    |    |          |       |        |        |              | 000D6 | P.AAY: | .BLKB    | 2                 | :           |   |
|    |    |    |    |    |    |    |    |    |    |          |       |        |        |              | 000D8 | P.AAY: | .LONG    | 6                 | :           |   |
|    |    |    |    |    |    |    |    |    |    |          |       |        |        |              | 000DC | P.AAZ: | .ADDRESS | P.AAZ             | :           |   |
|    |    |    |    |    |    |    | 52 | 48 | 53 | 53       | 59    | 53     |        |              | 000E0 | P.ABB: | .ASCII   | \SYSSHR\          | :           |   |
|    |    |    |    |    |    |    |    |    |    |          |       |        |        |              | 000E6 | P.ABA: | .BLKB    | 2                 | :           |   |
|    |    |    |    |    |    |    |    |    |    |          |       |        |        |              | 000E8 | P.ABA: | .LONG    | 6                 | :           |   |
|    |    |    |    |    |    |    |    |    |    |          |       |        |        |              | 000EC | P.ABB: | .ADDRESS | P.ABB             | :           |   |
|    |    |    |    |    |    |    | 4D | 45 | 54 | 53       | 59    | 53     |        |              | 000F0 | P.ABD: | .ASCII   | \SYSTEM\          | :           |   |
|    |    |    |    |    |    |    |    |    |    |          |       |        |        |              | 000F6 | P.ABC: | .BLKB    | 2                 | :           |   |
|    |    |    |    |    |    |    |    |    |    |          |       |        |        |              | 000F8 | P.ABC: | .LONG    | 6                 | :           |   |
|    |    |    |    |    |    |    |    |    |    |          |       |        |        |              | 000FC | P.ABD: | .ADDRESS | P.ABD             | :           |   |
|    |    |    |    |    |    |    | 45 | 43 | 41 | 52       | 54    |        |        |              | 00100 | P.ABF: | .ASCII   | \TRACE\           | :           |   |
|    |    |    |    |    |    |    |    |    |    |          |       |        |        |              | 00105 | P.ABE: | .BLKB    | 3                 | :           |   |
|    |    |    |    |    |    |    |    |    |    |          |       |        |        |              | 00108 | P.ABE: | .LONG    | 5                 | :           |   |



```

00000000' 0010C .ADDRESS P.ABF
59 52 41 52 42 49 4C 52 45 53 55 00110 P.ABH: .ASCII \USERLIBRARY\
0011B .BLKB 1
0000000B 0011C P.ABG: .LONG 11
00000000' 00120 .ADDRESS P.ABH
45 44 55 4C 43 4E 49 00124 P.ABJ: .ASCII \INCLUDE\
0012B .BLKB 1
00000007 0012C P.ABI: .LONG 7
00000000' 00130 .ADDRESS P.ABJ
59 52 41 52 42 49 4C 00134 P.ABL: .ASCII \LIBRARY\
0013B .BLKB 1
00000007 0013C P.ABK: .LONG 7
00000000' 00140 .ADDRESS P.ABL
53 4E 4F 49 54 50 4F 00144 P.ABN: .ASCII \OPTIONS\
0014B .BLKB 1
00000007 0014C P.ABM: .LONG 7
00000000' 00150 .ADDRESS P.ABN
43 52 41 45 53 5F 45 56 49 54 43 45 4C 45 53 00154 P.ABP: .ASCII \SELECTIVE_SEARCH\
48 00163
00000010 00164 P.ABO: .LONG 16
00000000' 00168 .ADDRESS P.ABP
31 50 0016C P.ABR: .ASCII \P1\
0016E .BLKB 2
00000002 00170 P.ABQ: .LONG 2
00000000' 00174 .ADDRESS P.ABR
3A 4C 4E 00178 P.ABT: .ASCII \NL:\
0017B .BLKB 1
00000003 0017C P.ABS: .LONG 3
00000000' 00180 .ADDRESS P.ABT
0314 0004 00184 P.ABU: .WORD 4, 788
00000000' 00188 .ADDRESS LNK$GL_FILESLEFT
00000000 0018C .LONG 0
00000000 00190 .LONG 0
0A 00194 P.ABV: .BYTE 10
41 54 53 47 4D 49 24 53 59 53 00195 .ASCII \SYSS$IMGSTA\
0019F .BLKB 1
42 4C 4F 2E 001A0 P.ABW: .ASCII \.OLB\
4A 42 4F 2E 001A4 P.ABX: .ASCII \.OBJ\
45 58 45 2E 001A8 P.ABY: .ASCII \.EXE\

.PSECT $OWNS,NOEXE,2

00000000 00000 UNIVSTAR_VALID:
.LONG 0
00000000 00004 LOCAL_QUALIFIER:
.LONG 0
00008 LISTPTR: .BLKB 4
0000C DFLT_NAME:
.BLKB 255
0010B .BLKB 1
0010C DFLT_INREL:
.BLKB 96
00000000 0016C LAST_FNM_DESC:
.LONG 0
00000000 00170 RLF_CONTEXT:
.LONG 0
00000000 00174 AN_OPEN_FDB:

```

```

0000 00178 DEBUG_DESC: .LONG 0
00 0017A .WORD 0
02 0017B .BYTE 0
00000000 0017C .BYTE 2
0000 00180 MODULE_DESC: .LONG 0
00 00182 .WORD 0
02 00183 .BYTE 0
00000000 00184 .BYTE 2
0000 00188 SHARE_DESC: .LONG 0
00 0018A .WORD 0
02 0018B .BYTE 0
00000000 0018C .BYTE 2
0000 00190 TOKEN_DESC: .LONG 0
00 00192 .WORD 0
02 00193 .BYTE 0
00000000 00194 .BYTE 2
00198 SYSTEMBASE: .LONG 0
0019C BASEDFLAG: .BLKB 4
001A0 OUTFILE: .BLKB 4
001A4 NEXTRSA: .BLKB 4
00000000' 001A8 TYPEADR: .ADDRESS DEFILETYPE
001AC INFILFLGS:
001B0 TPDATA: .BLKB 4
001B4 RSAREMSIZE:
001B5 .BLKB 1
001B8 IMAGE_QUAL TABLE: .BLKB 3
00000000' 001D0 .ADDRESS SD_TRACE, SD_CONTIGUOUS, SD_POIMAGE, -
SD_PROTECT, SD_SYSLIB, SD_SYSSHR, -
SD_HEADER
19 14 0A 17 18 13 12 001D4 BIT_TABLE: .BYTE 18, 19, 24, 23, 10, 20, 25
.PSECT $GLOBALS,NOEXE,2

00000 LNK$GL_FILESLEFT:: .BLKB 4
00004 LNK$GL_ULIBMASK:: .BLKB 4
00008 LNK$GL_INRELNAM:: .BLKB 4
0000C LNK$GL_RELNAM_EXE:: .BLKB 4
00010 LNK$GL_RELNAM_MAP:: .BLKB 4
00014 LNK$GL_RELNAM_SYM:: .BLKB 4
00018 LNK$GB_LOCNOV_EXE:: .BLKB 1

```

: R  
:

00019 LNK\$GB\_LOCNV\_MAP::  
      .BLKB 1  
0001A LNK\$GB\_LOCNV\_SYM::  
      .BLKB 1  
0001B LNK\_GB\_GETCMD\_SP1::  
      .BLKB 1  
0001C LNK\$GL\_SYSINPUT\_FLAG::  
      .BLKB 4  
00020 LNK\$GL\_CLIWORK::  
      .BLKB 4

SD\_BRIEF= P.AAA  
SD\_CONTIGUOUS= P.AAC  
SD\_CROSS\_REFERENCE= P.AAE  
SD\_DEBUG= P.AAG  
SD\_EXECUTABLE= P.AAI  
SD\_FULL= P.AAK  
SD\_HEADER= P.AAM  
SD\_MAP= P.AAO  
SD\_POIMAGE= P.AAQ  
SD\_PROTECT= P.AAS  
SD\_SHAREABLE= P.AAU  
SD\_SYMBOL\_TABLE= P.AAW  
SD\_SYSLIB= P.AAY  
SD\_SYSSHR= P.ABA  
SD\_SYSTEM= P.ABC  
SD\_TRACE= P.ABE  
SD\_USERLIBRARY= P.ABG  
SD\_INCLUDE= P.ABI  
SD\_LIBRARY= P.ABK  
SD\_OPTIONS= P.ABM  
SD\_SELECTIVE\_SEARCH= P.ABO  
SD\_P1= P.ABQ  
NUCL\_DEVICE= P.ABS  
LNK\$GT\_JPILST== P.ABU  
LNK\$GT\_IMGSTA== P.ABV  
LIBFILTYPE== P.ABW  
DEFILTYPE== P.ABX  
SHRFILTYPE== P.ABY  
.EXTRN CLIS\_NEGATED, CLIS\_LOCNEG  
.EXTRN CLIS\_DEFAULTED, CLIS\_LOCPRES  
.EXTRN CLIS\_PRESENT, LINS\_UNIVSTAR  
.EXTRN LINS\_CLOSEIN, LINS\_CMDSYNTAX  
.EXTRN LINS\_FIRSTMOD, LINS\_ILLOQUALVAL  
.EXTRN LINS\_LINERR, LINS\_NOMODS  
.EXTRN LINS\_OPENIN, LINS\_TRACIGN  
.EXTRN STRSCOPY\_R, LIBSCVT\_OTB  
.EXTRN LIBSCVT\_DTb, LIBSCVT\_HTB  
.EXTRN CLISPRESENT, CLISGET-VALUE  
.EXTRN LIBSFIND\_FILE, LIBSFIND\_FILE\_END  
.EXTRN LIBSINSERT\_TREE  
.EXTRN LIBSLOOKUP\_TREE  
.EXTRN LIB\$PARSE, LNK\$FNTPSCNAM  
.EXTRN LNK\$ALLOBLK, LNK\$DEALBLK  
.EXTRN LNK\$PROCOPTNS, LNK\$FILNAMDSC  
.EXTRN LNK\$INSUDFSYM, LNK\$INSERT  
.EXTRN LNK\$CLOSEFILE, LNK\$OPENLIB

| Address | OpCode | OpType | OpData     | OpLabel | OpComment  | OpValue |
|---------|--------|--------|------------|---------|--|---------|
|         |        |        |            |         | .EXTRN LNK\$SEARCH, LNK\$GL_INCLST                 |         |
|         |        |        |            |         | .EXTRN LNK\$GW_STACK, LNK\$GL_CURFIL               |         |
|         |        |        |            |         | .EXTRN LNK\$GL_CTLMSK, LNK\$GL_LASTCLU             |         |
|         |        |        |            |         | .EXTRN LNK\$GL_CLULST, LNK\$GL_DEFCLU              |         |
|         |        |        |            |         | .EXTRN LNK\$GL_CURCLU, LNK\$GL_CLUTREE             |         |
|         |        |        |            |         | .EXTRN LNK\$GL_FVMLST, LNK\$GL_IMGFIL              |         |
|         |        |        |            |         | .EXTRN LNK\$GL_MAPFIL, LNK\$GW_NCLUSTERS           |         |
|         |        |        |            |         | .EXTRN LNK\$GW_NSymbols                            |         |
|         |        |        |            |         | .EXTRN LNK\$GL_SYMFIL, LNK\$GL_CCLULST             |         |
|         |        |        |            |         | .EXTRN LNK\$GL_PSCDFLST                            |         |
|         |        |        |            |         | .PSECT \$CODE\$,NOWRT,2                            |         |
|         |        |        | OFFC 00000 |         | .ENTRY LNK\$GETCMD, Save R2,R3,R4,R5,R6,R7,R8,R9,- | 0454    |
|         |        |        |            |         | R10,R11  |         |
|         |        |        |            |         | MOVAB GET_SYM, R11                                 |         |
|         |        |        |            |         | MOVAB GET_MAP, R10                                 |         |
|         |        |        |            |         | MOVAB GET_IMAGE, R9                                |         |
|         |        |        |            |         | MOVAB CLISPRESNT, R8                               |         |
|         |        |        |            |         | MOVAB LNK\$GL_DEFCLU+76, R7                        |         |
|         |        |        |            |         | MOVAB SD_DEBUG, R6                                 |         |
|         |        |        |            |         | MOVAB DEBUG_DESC, R5                               |         |
|         |        |        |            |         | MOVAB LNK\$GL_CTLMSK, R4                           |         |
|         |        |        |            |         | SUBL2 #R, SP                                       |         |
| 01      | 51     | 18     | A0         |         | MOVL ARGLIST, R0                                   | 0476    |
|         | A4     |        | 01         |         | EXTZV #2, #1, 24(R0), R1                           |         |
|         |        |        | 04         |         | INSV R1, #4, #1, LNK\$GL_CTLMSK+1                  |         |
|         |        |        | 3C         |         | MOVAB #122, RSAREMSIZE                             | 0478    |
|         |        |        | A5         |         | PUSHAB NEXTRSA                                     | 0479    |
|         |        |        | 7E         |         | MOVZBL RSAREMSIZE, -(SP)                           |         |
|         |        |        | 00         |         | CALLS #2, LNK\$ALLOBLK                             |         |
|         |        |        |            |         | CLRL I   | 0486    |
|         |        |        |            |         | PUSHL IMAGE_QUAL_TABLE[I]                          | 0487    |
|         |        |        | 68         |         | CALLS #1, CLISPRESNT                               |         |
|         |        |        | 53         |         | MOVZBL BIT_TABLE[I], R3                            |         |
| 64      |        |        | 01         |         | INSV R0, R3, #1, LNK\$GL_CTLMSK                    |         |
|         |        |        | EB         |         | AOBLEQ #6, I, 1\$                                  |         |
|         |        |        | 52         |         | PUSHAB SD_USERLIBRARY                              | 0489    |
|         |        |        | 00D8       |         | CALLS #1, CLISPRESNT                               |         |
|         |        |        | 68         |         | BLBC R0, 2\$                                       |         |
|         |        |        | 07         |         | CALLS #0, SETUSERLIB                               | 0490    |
|         |        |        | 00000000V  |         | PUSHAB SD_SYSTEM                                   | 0492    |
|         |        |        | 00B4       |         | CALLS #1, CLISPRESNT                               |         |
|         |        |        | 68         |         | BLBC R0, 3\$                                       |         |
|         |        |        | 05         |         | BISB2 #8, LNK\$GL_CTLMSK                           | 0494    |
|         |        |        | 64         |         | BRB 4\$  | 0495    |
|         |        |        |            |         | PUSHAB SD_SHAREABLE                                | 0499    |
|         |        |        | 70         |         | CALLS #1, CLISPRESNT                               |         |
|         |        |        | 68         |         | BLBC R0, 5\$                                       |         |
|         |        |        | 11         |         | BISB2 #4, LNK\$GL_CTLMSK                           | 0501    |
|         |        |        | 64         |         | BICL2 #262146, LNK\$GL_CTLMSK                      | 0502    |
|         |        |        | 00040002   |         | CALLS #0, SPECIALIMAGE                             | 0504    |
|         |        |        | 00000000V  |         | PUSHL R6   | 0507    |
|         |        |        | EF         |         | CALLS #1, CLISPRESNT                               |         |
|         |        |        | 68         |         | BLBC R0, 6\$                                       |         |
|         |        |        | 0F         |         | PUSHL R5   | 0509    |
|         |        |        |            |         | PUSHL R6   |         |

|    |    |           |    |           |    |             |             |                               |                             |      |
|----|----|-----------|----|-----------|----|-------------|-------------|-------------------------------|-----------------------------|------|
|    |    | 00000000G | 00 | 02        | FB | 000BD       | CALLS       | #2, CLISGET_VALUE             | 0510                        |      |
|    |    |           | 64 | 40        | 8F | 88 000C4    | BISB2       | #64, LNK\$GL_CTLMSK           | 0513                        |      |
|    |    |           | 69 |           | 00 | FB 000C8    | 68: CALLS   | #0, GET_IMAGE                 | 0514                        |      |
|    |    |           | 6A |           | 00 | FB 000CB    | CALLS       | #0, GET_MAP                   | 0515                        |      |
|    |    |           | 6B |           | 00 | FB 000CE    | CALLS       | #0, GET_SYM                   | 0517                        |      |
|    |    | FEBC      | C5 | 01        | D0 | 000D1       | MOVL        | #1, LOCAL_QUALIFIER           | 0518                        |      |
|    |    | 00000000V | EF | 00        | FB | 000D6       | 79: CALLS   | #0, GET_INPUT                 | 0520                        |      |
|    |    |           | 0B | 50        | E9 | 000DD       | BLBC        | R0, 89                        | 0521                        |      |
|    |    |           | 69 | 00        | FB | 000E0       | CALLS       | #0, GET_IMAGE                 | 0522                        |      |
|    |    |           | 6A | 00        | FB | 000E3       | CALLS       | #0, GET_MAP                   | 0518                        |      |
|    |    |           | 6B | 00        | FB | 000E6       | CALLS       | #0, GET_SYM                   | 0526                        |      |
|    |    |           |    | EB        | 11 | 000E9       | BRB         | 79                            | 0529                        |      |
|    |    |           |    | A5        | 9F | 000EB       | 89: PUSHAB  | RLF_CONTEXT                   | 0530                        |      |
|    |    | 00000000G | 00 | 01        | FB | 000EE       | CALLS       | #1, LIB\$FIND_FILE_END        | 0538                        |      |
| 50 | 64 |           | 01 | 03        | EF | 000F5       | EXTZV       | #3, #1, LNK\$GL_CTLMSK, R0    | 0540                        |      |
| 51 | 64 |           | 01 | 02        | EF | 000FA       | EXTZV       | #2, #1, LNK\$GL_CTLMSK, R1    | 0542                        |      |
|    |    |           | 50 | 51        | CB | 000FF       | BISL2       | R1, R0                        | 0543                        |      |
| 52 | 64 |           | 01 | 01        | EF | 00102       | EXTZV       | #1, #1, LNK\$GL_CTLMSK, R2    | 0546                        |      |
| 64 |    |           | 52 | 50        | 88 | 00107       | BISB2       | R0, R2                        | 0549                        |      |
|    |    |           | 00 | 52        | F0 | 0010A       | INSV        | R2, #0, #1, LNK\$GL_CTLMSK    | 0550                        |      |
|    |    |           | 64 | 04        | E1 | 0010F       | BBC         | #4, LNK\$GL_CTLMSK, -119      | 0558                        |      |
|    |    |           |    | A6        | 9F | 00113       | PUSHAB      | SD_FULL                       | 0561                        |      |
|    |    |           | 68 | 01        | FB | 00116       | CALLS       | #1, CLISPRESENT               | 0562                        |      |
|    |    |           | 09 | 50        | E9 | 00119       | BLBC        | R0, 99                        | 0563                        |      |
|    |    | 01        | A4 | 01        | 88 | 0011C       | BISB2       | #1, LNK\$GL_CTLMSK+1          | 0567                        |      |
|    |    | 01        | A4 | 40        | 8F | 8A 00120    | BICB2       | #64, LNK\$GL_CTLMSK+1         | 0568                        |      |
|    |    |           | 68 | C4        | A6 | 9F 00125    | 99: PUSHAB  | SD_BRIEF                      | 0569                        |      |
|    |    |           | 04 | 01        | FB | 00128       | CALLS       | #1, CLISPRESENT               | 0576                        |      |
|    |    |           | 01 | A4        | 50 | E9 0012B    | BLBC        | R0, 109                       | 0578                        |      |
|    |    |           |    | 02        | 88 | 0012E       | BISB2       | #2, LNK\$GL_CTLMSK+1          | 0579                        |      |
|    |    |           |    | A6        | 9F | 00132       | 109: PUSHAB | SD_CROSS_REFERENCE            | 0580                        |      |
|    |    |           | 68 | 01        | FB | 00135       | CALLS       | #1, CLISPRESENT               | 0583                        |      |
|    |    |           | 04 | 50        | E9 | 00138       | BLBC        | R0, 119                       | 0584                        |      |
|    |    |           | 64 | 80        | 8F | 88 0013B    | BISB2       | #128, LNK\$GL_CTLMSK          | 0585                        |      |
|    |    |           | 4F | 24        | A5 | E9 0013F    | 119: BLBC   | BASEDFLAG, 189                | 0586                        |      |
|    |    |           | 13 | 03        | A4 | 01 E1 00143 | BBC         | #1, LNK\$GL_CTLMSK+3, 129     | 0587                        |      |
|    |    |           | 50 | 20        | A5 | 000001FF    | 8F C1 00148 | ADDL3                         | #511, SYSTEMBASE, R0        | 0588 |
|    |    |           | 67 |           | 50 | 000001FF    | 8F CB 00151 | BICL3                         | #511, R0, LNK\$GL_DEFCLU+76 | 0589 |
|    |    |           |    | 04        | 11 | 00159       | BRB         | 139                           | 0590                        |      |
|    |    |           | 67 | 20        | A5 | D0 0015B    | 129: MOVL   | SYSTEMBASE, LNK\$GL_DEFCLU+76 | 0591                        |      |
|    |    |           | 50 | 00000000G | 00 | D0 0015F    | 139: MOVL   | LNK\$GL_FVMLST, R0            | 0592                        |      |
|    |    |           | 51 | 67        | D0 | 00166       | MOVL        | LNK\$GL_DEFCLU+76, R1         | 0593                        |      |
|    |    | 40000000  | 8F | 51        | D1 | 00169       | CMPL        | R1, #1073741824               | 0594                        |      |
|    |    |           |    | 1D        | 1F | 00170       | BLSSU       | 159                           | 0595                        |      |
|    |    | 80000000  | 8F | 51        | D1 | 00172       | CMPL        | R1, #-2147483648              | 0596                        |      |
|    |    |           |    | 0A        | 1F | 00179       | BLSSU       | 149                           | 0597                        |      |
|    |    | 04        | A0 | 80000000  | 8F | D0 0017B    | MOVL        | #-2147483648, 4(R0)           | 0598                        |      |
|    |    |           |    | 0D        | 11 | 00183       | BRB         | 169                           | 0599                        |      |
|    |    | 04        | A0 | 40000000  | 8F | D0 00185    | 149: MOVL   | #1073741824, 4(R0)            | 0600                        |      |
|    |    |           |    | 03        | 11 | 0018D       | BRB         | 169                           | 0601                        |      |
|    |    |           |    | A0        | D4 | 0018F       | 159: CLRL   | 4(R0)                         | 0602                        |      |
|    |    |           |    | 00        | D5 | 00192       | 169: TSTL   | FIRSTCLUSTER                  | 0603                        |      |
|    |    |           |    | 12        | 12 | 00198       | BNEQ        | 179                           | 0604                        |      |
|    |    |           |    | BC        | A7 | D5 0019A    | TSTL        | LNK\$GL_DEFCLU+8              | 0605                        |      |
|    |    |           |    | 0D        | 12 | 0019D       | BNEQ        | 179                           | 0606                        |      |
|    |    |           |    | 8F        | DD | 0019F       | PUSHL       | #LINS NOMODS                  | 0607                        |      |
|    |    | 00000000G | 00 | 01        | FB | 001A5       | CALLS       | #1, LIB\$STOP                 | 0608                        |      |

|    |           |    |    |       |       |       |        |                                  |        |
|----|-----------|----|----|-------|-------|-------|--------|----------------------------------|--------|
| 04 |           | 64 | 03 | E0    | 001AC | 17\$: | BBS    | #3, LNK\$GL_CTLMSK, 18\$         | : 0590 |
| 04 |           | 64 | 02 | E1    | 001B0 |       | BBC    | #2, LNK\$GL_CTLMSK, 19\$         |        |
|    | 02        | A4 | 04 | 8A    | 001B4 | 18\$: | BICB2  | #4, LNK\$GL_CTLMSK+2             | : 0591 |
| 05 |           | 64 | 06 | E0    | 001B8 | 19\$: | BBS    | #6, LNK\$GL_CTLMSK, 20\$         | : 0593 |
| 5F |           | A4 | 02 | E1    | 001BC |       | BBC    | #2, LNK\$GL_CTLMSK+2, 23\$       |        |
| 0D |           | A4 | 02 | E0    | 001C1 | 20\$: | BBS    | #2, LNK\$GL_CTLMSK+2, 21\$       | : 0596 |
|    | 00000000G | 00 | 8F | DD    | 001C6 |       | PUSHL  | #LINS TRACTGN                    | : 0597 |
|    | 34        | A5 | 01 | FB    | 001CC |       | CALLS  | #1, LTBSSIGNAL                   |        |
|    |           |    | 20 | D0    | 001D3 | 21\$: | MOVL   | #32, INFILFLGS                   | : 0599 |
|    |           |    | 65 | B5    | 001D7 |       | TSTW   | DEBUG_DESC                       | : 0600 |
|    |           |    | 15 | 13    | 001D9 |       | BEQL   | 22\$                             |        |
|    | 00000000V | EF | 55 | DD    | 001DB |       | PIJSHL | R5                               | : 0602 |
|    |           |    | 01 | FB    | 001DD |       | CALLS  | #1, INPUTFILE                    |        |
|    | 00000000G | 00 | A5 | 9F    | 001E4 |       | PUSHAB | RLF_CONTEXT                      | : 0603 |
|    |           |    | 01 | FB    | 001E7 |       | CALLS  | #1, LIB\$FIND_FILE_END           |        |
|    |           |    | 30 | 11    | 001EE |       | BRB    | 23\$                             | : 0600 |
|    |           |    | 5E | DD    | 001F0 | 22\$: | PUSHL  | SP                               | : 0609 |
|    |           |    | 08 | AE    | 9F    | 001F2 | PUSHAB | SYMVAL                           |        |
|    |           |    | C6 | 9F    | 001F5 |       | PUSHAB | LNK\$GT_IMGSTA                   |        |
|    | 00000000G | 00 | 03 | FB    | 001F9 |       | CALLS  | #3, LNR\$SEARCH                  |        |
|    |           |    | 5E | DD    | 00200 |       | PUSHL  | SP                               | : 0610 |
|    |           |    | 08 | AE    | 9F    | 00202 | PUSHAB | SYMVAL                           |        |
|    |           |    | C6 | 9F    | 00205 |       | PUSHAB | LNK\$GT_IMGSTA                   |        |
|    | 00000000G | 00 | 03 | FB    | 00209 |       | CALLS  | #3, LNR\$INSERT                  |        |
|    |           |    | 04 | AE    | DD    | 00210 | PUSHL  | SYMVAL                           | : 0611 |
|    | 00000000G | 00 | 01 | FB    | 00213 |       | CALLS  | #1, LNK\$INSUDFSYM               |        |
|    |           |    | 00 | B6    | 0021A |       | INCW   | LNK\$GW_NSMBOLS                  | : 0612 |
|    | 01        | A4 | 10 | 8A    | 00220 | 23\$: | BICB2  | #16, LNK\$GL_CTLMSK+1            | : 0616 |
|    | 00000000V | EF | 00 | FB    | 00224 |       | CALLS  | #0, LNK\$COLLECTPSCT             | : 0618 |
|    |           |    | A7 | D5    | 0022B |       | TSTL   | LNK\$GL_DEFCLU+8                 | : 0620 |
|    |           |    | 21 | 13    | 0022E |       | BEQL   | 25\$                             |        |
|    |           |    | 00 | D0    | 00230 |       | MOVL   | LNK\$GL_LASTCLU, R0              | : 0622 |
|    |           |    | A7 | 9E    | 00237 |       | MOVAB  | LNK\$GL_DEFCLU, (R0)             |        |
|    | B8        | A7 | 50 | D0    | 0023B |       | MOVL   | R0, LNR\$GL_DEFCLU+4             | : 0623 |
|    |           |    | 08 | 12    | 0023F |       | BNEQ   | 24\$                             |        |
|    | B8        | A7 | 00 | 9E    | 00241 |       | MOVAB  | LNK\$GL_CLULST, LNK\$GL_DEFCLU+4 | : 0624 |
|    | 00000000G | 00 | A7 | 9E    | 00249 | 24\$: | MOVAB  | LNK\$GL_DEFCLU, LNK\$GL_LASTCLU  | : 0625 |
| 16 |           |    | 02 | E1    | 00251 | 25\$: | BBC    | #2, LNR\$GL_CTLMSK+3, 26\$       | : 0628 |
|    |           |    | C5 | E8    | 00256 |       | BLBS   | UNIVSTAR_VALID, 26\$             | : 0629 |
|    |           |    | 8F | DD    | 0025B |       | PUSHL  | #LINS UNIVSTAR                   | : 0631 |
|    | 00000000G | 00 | 01 | FB    | 00261 |       | CALLS  | #1, LTBSSIGNAL                   |        |
|    |           |    | 04 | 8A    | 00268 |       | BICB2  | #4, LNK\$GL_CTLMSK+3             | : 0632 |
|    | 03        | A4 | 04 | 0026C | 26\$: | RET   |        | : 0635                           |        |

; Routine Size: 621 bytes, Routine Base: \$CODE\$ + 0000

; 525 0636 1

```

: 527      0637 1 routine clierror (desc,errorcode) : novalue =
: 528      0638 1
: 529      0639 1 !+
: 530      0640 1
: 531      0641 1 | This routine is called by result parser if it detects any error in this
: 532      0642 1 | command. requestdesc is the address of the current parameter descriptor and
: 533      0643 1 | errorcode is the encoded reason for the error.
: 534      0644 1 |
: 535      0645 1 | -
: 536      0646 2 begin
: 537      0647 2 map
: 538      0648 2     desc : ref block [,byte] ;           ! Pointer to a block
: 539      0649 2
: 540      0650 2 signal_stop (lin$_cmdsyntax, 0,lin$_linerr, 1,desc [dsc$_length], .errorcode) ;
: 541      0651 2
: 542      0652 1 end ;

```

```

                                0000 00000 CLIERROR:
                                .WORD  Save nothing
                                MOVQ   DESC, -(SP)
                                PUSHL  #1
                                PUSHL  #LINS_LINERR
                                CLRL   -(SP)
                                PUSHL  #LINS_CMDSYNTAX
                                CALLS  #6, LIB$STOP
                                RET

```

```

: 0637
: 0650
:
: 0652

```

: Routine Size: 30 bytes, Routine Base: \$CODE\$ + 026D

: 543 0653 1

```

: 545 0654 1 routine call_tparse (string,states,keys) =
: 546 0655 1
: 547 0656 1 :-
: 548 0657 1
: 549 0658 1 Thi routine calls TPARSE given the string, states and keys
: 550 0659 1
: 551 0660 1 Inputs:
: 552 0661 1
: 553 0662 1 STRING address of string descriptor for string to parse
: 554 0663 1 STATES address of tparse states table
: 555 0664 1 KEYS address of tparse keys table
: 556 0665 1
: 557 0666 1 :-
: 558 0667 2 begin
: 559 0668 2 map
: 560 0669 2 string : ref block [,byte] ;
: 561 0670 2 local
: 562 0671 2 status,
: 563 0672 2 tpablock : block [tpa$k_length0,byte] ;
: 564 0673 2
: 565 0674 2
: 566 0675 2 Initialize the tparse block
: 567 0676 2
: 568 0677 2 ch$fill (0, tpa$k_length0, tpablock) ;
: 569 0678 2
: 570 0679 2 tpablock [tpa$l_count] = tpa$k_count0 ;
: 571 0680 2 tpablock [tpa$l_options] = tpa$m_abbrev ;
: 572 0681 2 tpablock [tpa$l_stringcnt] = .string [dsc$w_length] ;
: 573 0682 2 tpablock [tpa$l_stringptr] = .string [dsc$a_pointer] ;
: 574 0683 2
: 575 0684 2
: 576 0685 2 Initialize own storage used
: 577 0686 2
: 578 0687 2 tpdata = 0 ;
: 579 0688 2
: 580 0689 2 return lib$tparse (tpablock, .states, .keys)
: 581 0690 1 end ;

```

|    |           | 003C 0000 CALL_TPARSE: |                |        |                             |        |
|----|-----------|------------------------|----------------|--------|-----------------------------|--------|
|    |           | 5E                     | 24 C2 00002    | .WORD  | Save R2,R3,R4,R5            | : 0654 |
| 24 | 00        | 6E                     | 00 2C 00005    | SUBL2  | #36, SP                     | : 0677 |
|    |           |                        | 6E 0000A       | MOVCS  | #0, (SP), #0, #36, TPABLOCK | : 0679 |
|    |           | 04 6E                  | 08 D0 J000B    | MOVL   | #8, TPABLOCK                | : 0680 |
|    |           | 50 AE                  | 02 D0 0000E    | MOVL   | #2, TPABLOCK+4              | : 0681 |
|    |           | 08 AE                  | 04 AC D0 00012 | MOVL   | STRING, R0                  | : 0682 |
|    |           | 0C AE                  | 60 3C 00016    | MOVZWL | (R0), TPABLOCK+8            | : 0687 |
|    |           |                        | 04 A0 D0 0001A | MOVL   | 4(R0), TPABLOCK+12          | : 0689 |
|    |           | 7E                     | 04 EF D4 0001F | CLRL   | TPDATA                      | : 0689 |
|    |           |                        | 08 AC 7D 00025 | MOVQ   | STATES, -(SP)               | : 0690 |
|    |           |                        | 08 AE 9F 00029 | PUSHAB | TPABLOCK                    |        |
|    | 00000000G | 00                     | 03 FB 0002C    | CALLS  | #3, LIB\$TPARSE             |        |
|    |           |                        | 04 00033       | RET    |                             | : 0690 |



LNK\_GETCMDINP  
V04=000

<sup>6</sup>  
12-Sep-1984 23:57:45  
14-Sep-1984 12:40:30

YAX-11 BLISS-32 V4.0-742  
[LINKER.SRC]LNKGETCMD.B32;1

Page 23  
(8)

: Routine Size: 52 bytes,      Routine Base: \$CODES + 028B

: 582                    0691 1

LN  
VO



```

584 0692 1 routine specialimage : novalue =
585 0693 1
586 0694 1 |*
587 0695 1
588 0696 1 This routine is called by the CLI if the system or shareable image qualifier
589 0697 1 is seen on the primary output parameter.
590 0698 1
591 0699 1 If a value is present on the qualifier, the CLI is re-called through CLIADDR
592 0700 1 to convert the value which then sets the base address of the default cluster
593 0701 1 of the image being linked. If no value is present the base address is set
594 0702 1 to the lower end of system or process space. In both cases the stack size is
595 0703 1 set to zero. A system image will not but a shareable image will have a
596 0704 1 header.
597 0705 1
598 0706 1 -
599 0707 2 begin
600 0708 2 local
601 0709 2     symbolstring      : vector [sym$c_maxlng+1, byte],      ! Base address symbol storage
602 0710 2     nament           : ref block [,byte],                    ! Pointer to symbol name block
603 0711 2     symtabent        : ref block [,byte];                     ! Pointer to table entry for it
604 0712 2
605 0713 2
606 0714 2 lnk$gw_stack = 0 ;                                          ! Init stack
607 0715 2
608 0716 2 if .lnk$gl_ctlmsk [lnk$v_snr]                               ! If a shareable image
609 0717 2 then
610 0718 2     begin
611 0719 2     cli$get_value (sd_shareable, share_desc) ;             ! then get name now
612 0720 2     return ;                                               ! All done with /SHAREABLE...
613 0721 2     end ;
614 0722 2
615 0723 2 if .lnk$gl_ctlmsk [lnk$v_sys]                               ! If a system image
616 0724 2 then lnk$gl_fvmlst [fvmlst_address] = system_space ;      ! Set first available virtual memory
617 0725 2
618 0726 2 if not cli$get_value (sd_system, token_desc)               ! If there was no qualifier value
619 0727 2 then return ;                                             ! Then all done
620 0728 2
621 0729 2 lnk$gl_defclu [clu$v_based] = true ;                        ! Set the bit for fixed in address space
622 0730 2 lnk$gl_ctlmsk [lnk$v_picing] = false ;                     ! And image is not pic
623 0731 2 basedf[ag] = true ;
624 0732 2
625 0733 2 if .(.token_desc [dsc$a_pointer])<0,8> eql %c%'          ! If value for the base was prefixed with
626 0734 2 then                                                       ! a %, then we definitely have a numeric
627 0735 2     begin                                                 ! base -- find out the radix
628 0736 2     local          status ;
629 0737 2
630 0738 2     selectone .(.token_desc [dsc$a_pointer]+1)<0,8> of   ! Base specified as %?
631 0739 2     set
632 0740 2     [%c'X'] : status = lib$cvt_htb (.token_desc [dsc$w_length] - 2      ! Value is in hex
633 0741 2     ,.token_desc [dsc$a_pointer] + 2
634 0742 2     , systembase
635 0743 2     ;
636 0744 2     [%c'D'] : status = lib$cvt_dtb (.token_desc [dsc$w_length] - 2      ! Value is in decimal
637 0745 2     ,.token_desc [dsc$a_pointer] + 2
638 0746 2     , systembase
639 0747 2     ;
640 0748 2     [%c'O'] : status = lib$cvt_otb (.token_desc [dsc$w_length] - 2      ! Value is in octal

```

```

641 0749
642 0750
643 0751
644 0752
645 0753
646 0754
647 0755
648 0756
649 0757
650 0758
651 0759
652 0760
653 0761
654 0762
655 0763
656 0764
657 0765
658 0766
659 0767
660 0768
661 0769
662 0770
663 0771
664 0772
665 0773
666 0774
667 0775
668 0776
669 0777
670 0778
671 0779
672 0780
673 0781
674 0782
675 0783
676 0784

        ..token_desc [dsc$a_pointer] + 2
        }
        systembase
    [otherwise] : clierror (token_desc, lin$_illqualval) ;      ! Anything else is illegal
tes ;

if not .status      ! If error occurred in value conversion,
then clierror (token_desc, .status) ;      ! report it here
end

else
    Didn't explicitly specify a radix -- assume decimal
if not lib$cvd_dtb (.token_desc [dsc$a_pointer], systembase)
then
    begin
    if (symbolstring [0] = minu (.token_desc [dsc$a_pointer], sym$c_maxlng)
        ) eql 0
    then
        return ;
    ch$move (.symbolstring [0]
        ..token_desc [dsc$a_pointer]
        symbolstring [1]
        ) ;
    lnk$search (symbolstring [0], symtabent, nament) ;      ! Then search for place to insert it
    lnk$insert (symbolstring [0], symtabent, nament) ;      ! and do it
    lnk$insudfsym (.symtabent) ;      ! Insert undefined symbol
    lnk$gw_nsymbols = .lnk$gw_nsymbols + 1 ;      ! Count as one more symbol
    lnk$gl_defclu [clu$l_base] = .symtabent ;      ! Save the symbol table address
    lnk$gl_defclu [clu$v_sympas] = true ;      ! Flag as symbolic
    end ;      ! And all done till it is defined
return true
end ;

```

OFFC 0000 SPECIALIMAGE:

|    |    |           |    |    |       |                                      |                         |
|----|----|-----------|----|----|-------|--------------------------------------|-------------------------|
|    |    |           |    |    | .WORD | Save R2,R3,R4,R5,R6,R7,R8,R9,R10,R11 | : 0692                  |
|    | 5B | A9        | AF | 9E | 00002 | MOVAB                                | CLIERROR, R11           |
|    | 5A | 00000000G | 00 | 9E | 00006 | MOVAB                                | LIB\$CVT_DTB, R10       |
|    | 59 | 00000000G | 00 | 9E | 0000D | MOVAB                                | CLISGET_VALUE, R9       |
|    | 58 | 00000000G | 00 | 9E | 00014 | MOVAB                                | LNK\$GL_DEFCLU+88, R8   |
|    | 57 | 00000000G | 00 | 9E | 0001B | MOVAB                                | LNK\$GL_CTLMSK, R7      |
|    | 56 | 00000000' | EF | 9E | 00022 | MOVAB                                | TOKEN_DESC, R6          |
|    | 5E |           | 28 | C2 | 00029 | SUBL2                                | #40, SP                 |
|    |    | 00000000G | 00 | B4 | 0002C | CLRW                                 | LNK\$GW_STACK           |
| OD | 67 |           | 02 | E1 | 00032 | BBC                                  | #2, LNK\$GL_CTLMSK, 1\$ |
|    |    | F8        | A6 | 9F | 00036 | PUSHAB                               | SHARE_DESC              |
|    |    | 00000000' | EF | 9F | 00039 | PUSHAB                               | SD_SHAREABLE            |
|    | 69 |           | 02 | FB | 0003F | CALLS                                | #2, CLISGET_VALUE       |
|    |    |           |    | 04 | 00042 | RET                                  |                         |
| OF | 67 |           | 03 | E1 | 00043 | BBC                                  | #3, LNK\$GL_CTLMSK, 2\$ |
|    |    |           |    |    |       |                                      | : 0714                  |
|    |    |           |    |    |       |                                      | : 0716                  |
|    |    |           |    |    |       |                                      | : 0719                  |
|    |    |           |    |    |       |                                      | : 0718                  |
|    |    |           |    |    |       |                                      | : 0723                  |

|           |    |           |    |    |       |       |                     |                       |            |      |
|-----------|----|-----------|----|----|-------|-------|---------------------|-----------------------|------------|------|
| 04        | 50 | 00000000G | 00 | D0 | 00047 | MOVL  | LNK\$GL_FVMLST, R0  | 0724                  |            |      |
|           | A0 | 80000000  | 8F | D0 | 0004E | MOVL  | #-2147483648, 4(R0) |                       |            |      |
|           |    |           | 5F | DD | 00056 | 2\$:  | PUSHL               | R6                    | 0726       |      |
|           |    | 00000000' | EF | 9F | 00058 |       | PUSHAB              | SD_SYSTEM             |            |      |
|           | 69 |           | 02 | FB | 0005E |       | CALLS               | #2, CLISGET_VALUE     |            |      |
|           | 01 |           | 50 | EB | 00061 |       | BLBS                | R0, 3\$               |            |      |
|           |    |           |    | 04 | 00064 |       | RET                 |                       |            |      |
|           | 68 |           | 01 | 88 | 00065 | 3\$:  | BISB2               | #1, LNK\$GL_DEFCLU+88 | 0729       |      |
| 02        | A7 |           | 02 | 8A | 00068 |       | BICB2               | #2, LNK\$GL_CTLMSK+2  | 0730       |      |
| 0C        | A6 |           | 01 | D0 | 0006C |       | MOVL                | #1, BASEDFLAG         | 0731       |      |
|           | 52 | 04        | A6 | D0 | 00070 |       | MOVL                | TOKEN_DESC+4, R2      | 0733       |      |
|           | 25 |           | 62 | 91 | 00074 |       | CMPB                | (R2), -#37            |            |      |
|           |    |           | 67 | 12 | 00077 |       | BNEQ                | 8\$                   |            |      |
|           | 50 | 01        | A2 | 9A | 00079 |       | MOVZBL              | 1(R2), R0             | 0738       |      |
| 58        | 8F |           | 50 | 91 | 0007D |       | CMPB                | R0, #88               | 0740       |      |
|           |    |           | 15 | 12 | 00081 |       | BNEQ                | 4\$                   |            |      |
|           |    |           | 08 | A6 | 9F    | 00083 | PUSHAB              | SYSTEMBASE            |            |      |
|           |    |           | 02 | A2 | 9F    | 00086 | PUSHAB              | 2(R2)                 | 0741       |      |
|           | 7E |           | 66 | 3C | 00089 |       | MOVZWL              | TOKEN_DESC, -(SP)     | 0740       |      |
|           | 6E |           | 02 | C2 | 0008C |       | SUBL2               | #2, (SP)              |            |      |
| 00000000G | 00 |           | 03 | FB | 0008F |       | CALLS               | #3, LIB\$CVT_HTB      |            |      |
|           |    |           | 3D | 11 | 00096 |       | BRB                 | 7\$                   |            |      |
|           | 44 | 8F        | 50 | 91 | 00098 | 4\$:  | CMPB                | R0, #68               | 0744       |      |
|           |    |           | 11 | 12 | 0009C |       | BNEQ                | 5\$                   |            |      |
|           |    |           | 08 | A6 | 9F    | 0009E | PUSHAB              | SYSTEMBASE            |            |      |
|           |    |           | 02 | A2 | 9F    | 000A1 | PUSHAB              | 2(R2)                 | 0745       |      |
|           | 7E |           | 66 | 3C | 000A4 |       | MOVZWL              | TOKEN_DESC, -(SP)     | 0744       |      |
|           | 6E |           | 02 | C2 | 000A7 |       | SUBL2               | #2, (SP)              |            |      |
|           | 6A |           | 03 | FB | 000AA |       | CALLS               | #3, LIB\$CVT_DTB      |            |      |
|           |    |           | 26 | 11 | 000AD |       | BRB                 | 7\$                   |            |      |
|           | 4F | 8F        | 50 | 91 | 000AF | 5\$:  | CMPB                | R0, #79               | 0748       |      |
|           |    |           | 15 | 12 | 000B3 |       | BNEQ                | 6\$                   |            |      |
|           |    |           | 08 | A6 | 9F    | 000B5 | PUSHAB              | SYSTEMBASE            |            |      |
|           |    |           | 02 | A2 | 9F    | 000B8 | PUSHAB              | 2(R2)                 | 0749       |      |
|           | 7E |           | 66 | 3C | 000BB |       | MOVZWL              | TOKEN_DESC, -(SP)     | 0748       |      |
|           | 6E |           | 02 | C2 | 000BE |       | SUBL2               | #2, (SP)              |            |      |
| 00000000G | 00 |           | 03 | FB | 000C1 |       | CALLS               | #3, LIB\$CVT_OTB      |            |      |
|           |    |           | 0B | 11 | 000C8 |       | BRB                 | 7\$                   |            |      |
|           |    | 00000000G | 8F | DD | 000CA | 6\$:  | PUSHL               | #LINS_ILLQUALVAL      | 0752       |      |
|           |    |           | 56 | DD | 000D0 |       | PUSHL               | R6                    |            |      |
|           | 68 |           | 02 | FB | 000D2 |       | CALLS               | #2, CLIERROR          |            |      |
|           | 69 |           | 50 | EB | 000D5 | 7\$:  | BLBS                | STATUS, 10\$          | 0755       |      |
|           |    |           | 50 | DD | 000D8 |       | PUSHL               | STATUS                | 0756       |      |
|           |    |           | 56 | DD | 000DA |       | PUSHL               | R6                    |            |      |
|           | 6B |           | 02 | FB | 000DC |       | CALLS               | #2, CLIERROR          |            |      |
|           |    |           |    | 04 | 000DF |       | RET                 |                       | 0733       |      |
|           |    |           | 08 | A6 | 9F    | 000E0 | 8\$:                | PUSHAB                | SYSTEMBASE | 0762 |
|           |    |           |    | 52 | DD    | 000E3 |                     | PUSHL                 | R2         |      |
|           | 7E |           | 66 | 3C | 000E5 |       | MOVZWL              | TOKEN_DESC, -(SP)     |            |      |
|           | 6A |           | 03 | FB | 000E8 |       | CALLS               | #3, LIB\$CVT_DTB      |            |      |
|           | 53 |           | 50 | EB | 000EB |       | BLBS                | R0, 10\$              |            |      |
|           | 50 |           | 66 | 3C | 000EE |       | MOVZWL              | TOKEN_DESC, R0        | 0765       |      |
|           | 1F |           | 50 | B1 | 000F1 |       | CMPW                | R0, #31               |            |      |
|           |    |           | 03 | 1B | 000F4 |       | BLEQU               | 9\$                   |            |      |
|           | 50 |           | 1F | D0 | 000F6 |       | MOVL                | #31, R0               |            |      |
| 08        | AE |           | 50 | 90 | 000F9 | 9\$:  | MOVB                | R0, SYMBOLSTRING      |            |      |
|           |    |           | 50 | D5 | 000FD |       | TSTL                | R0                    | 0768       |      |

|    |    |           |           |    |             |        |                                   |      |
|----|----|-----------|-----------|----|-------------|--------|-----------------------------------|------|
| 09 | AE | 04        | 50<br>B6  | 08 | 40 13 000FF | BEQL   | 10\$                              | 0772 |
|    |    |           |           |    | AE 9A 00101 | MOVZBL | SYMBOLSTRING, R0                  | 0774 |
|    |    |           |           |    | 50 28 00105 | MOVCS  | R0, @TOKEN_DESC+4, SYMBOLSTRING+1 | 0776 |
|    |    |           |           | 08 | 5E DD 0010B | PUSHL  | SP                                |      |
|    |    |           |           | 10 | AE 9F 0010D | PUSHAB | SYMTABENT                         |      |
|    |    | 00000000G | 00        |    | AE 9F 00110 | PUSHAB | SYMBOLSTRING                      |      |
|    |    |           |           |    | 03 FB 00113 | CALLS  | #3, LNK\$SEARCH                   |      |
|    |    |           |           | 08 | 5E DD 0011A | PUSHL  | SP                                | 0777 |
|    |    |           |           | 10 | AE 9F 0011C | PUSHAB | SYMTABENT                         |      |
|    |    | 00000000G | 00        |    | AE 9F 0011F | PUSHAB | SYMBOLSTRING                      |      |
|    |    |           |           | 04 | 03 FB 00122 | CALLS  | #3, LNK\$INSERT                   |      |
|    |    | 00000000G | 00        |    | AE DD 00129 | PUSHL  | SYMTABENT                         | 0778 |
|    |    |           |           |    | 01 FB 0012C | CALLS  | #1, LNK\$INSUDFSYM                |      |
|    |    |           | 00000000G | 00 | B6 00133    | INCL   | LNK\$GW NSYMBOLS                  | 0779 |
|    | F4 | A8        | 04        | AE | DD 00139    | MOVL   | SYMTABENT, LNK\$GL_DEFCLU+76      | 0780 |
|    |    | 68        |           | 02 | 88 0013E    | BISB2  | #2, LNK\$GL_DEFCLU+88             | 0781 |
|    |    |           |           | 04 | 00141 10\$: | RET    |                                   | 0784 |

: Routine Size: 322 bytes, Routine Base: \$CODE\$ + 02BF

: 677 0785 1

```

: 679 0786 1 global routine lnk$setlibrin (option_flag) : novalue =
: 680 0787 1
: 681 0788 1 !+
: 682 0789 1 !-
: 683 0790 1 This routine is called by the CLI if the LIBRARY input file qualifier is
: 684 0791 1 seen on an input parameter.
: 685 0792 1 !-
: 686 0793 1 !-
: 687 0794 2 begin
: 688 0795 2 builtin
: 689 0796 2 nullparameter ;
: 690 0797 2 local
: 691 0798 2 called_by_option ;
: 692 0799 2
: 693 0800 2 called_by_option = not nullparameter (1) ;
: 694 0801 2
: 695 0802 2 if not .called_by_option
: 696 0803 2 then if cli$present (sd include)
: 697 0804 2 then infilflgs = .infilflgs or fdb$m_libextr ;
: 698 0805 2
: 699 0806 2 if .lnk$gl_clulst [0] eql 0
: 700 0807 2 and .lnk$gl_curclu [clu$l_fstfdb] eql 0
: 701 0808 2 and (.infilflgs and fdb$m_libextr) eql 0
: 702 0809 2 then
: 703 0810 2 signal_stop (lin$_firstmod) ;
: 704 0811 2
: 705 0812 2 typeadr = ch$ptr (libfiltype) ;
: 706 0813 2 return ;
: 707 0814 1 end ;

```

```

! It is important to know
! whether or not we were called
! by lnk$procopts in order to
! know who should do the
! parsing -- the cli or me

! If no preceding clusters
! And no files in current cluster
! And not explicit inclusion from the

! File give up with an error

! Set address of default file type

```

|  |    |           |             |        |                         |        |
|--|----|-----------|-------------|--------|-------------------------|--------|
|  |    |           | 0004 00000  | .ENTRY | LNK\$SETLIBRIN, Save R2 | : 0786 |
|  | 52 | 00000000' | EF 9E 00002 | MOVAB  | INFILFLGS, R2           | : 0800 |
|  |    |           | 6C 95 00009 | TSTB   | (AP)                    |        |
|  | 50 |           | 05 12 0000B | BNEQ   | 1\$                     |        |
|  |    |           | 01 D0 0000D | MOVL   | #1, R0                  |        |
|  |    |           | 09 11 00010 | BRB    | 2\$                     |        |
|  |    |           | 50 D4 00012 | CLRL   | R0                      |        |
|  |    | 04        | AC D5 00014 | TSTL   | 4(AP)                   |        |
|  |    |           | 02 12 00017 | BNEQ   | 2\$                     |        |
|  |    |           | 50 C6 00019 | INCL   | R0                      |        |
|  | 50 |           | 50 D2 0001B | MCOML  | R0, CALLED BY OPTION    |        |
|  | 14 |           | 50 E8 0001E | BLBS   | CALLED BY OPTION, 3\$   | : 0802 |
|  |    | 00000000' | EF 9F 00021 | PUSHAB | SD_INCLUDE              | : 0803 |
|  | 00 |           | 01 FB 00027 | CALLS  | #1, CLIPRESENT          |        |
|  | 04 |           | 50 E9 0002E | BLBC   | R0, 3\$                 |        |
|  | 62 | 40        | 8F 88 00031 | BISB2  | #64, INFILFLGS          | : 0804 |
|  |    | 00000000G | 00 D5 00035 | TSTL   | LNK\$GL_CLULST          | : 0806 |
|  |    |           | 1D 12 0003B | BNEQ   | 4\$                     |        |
|  | 50 | 00000000G | 00 D0 0003D | MOVL   | LNK\$GL_CURCLU, R0      | : 0807 |
|  |    | 08        | A0 D5 00044 | TSTL   | 8(R0)                   |        |
|  |    |           | 11 12 00047 | BNEQ   | 4\$                     |        |
|  | 0D | 62        | 06 E0 00049 | BBS    | #6, INFILFLGS, 4\$      | : 0808 |
|  |    | 00000000G | 8F DD 0004D | PUSHL  | #LINS_FIRSTMOD          | : 0810 |

LNK\_GETCMDINP  
V04=000

N 6  
15-Sep-1984 23:57:45 VAX-11 Bliss-32 V4.0-742  
14-Sep-1984 12:40:30 [LINKER.SRC]LNKGETCMD.B32;1

Page 29  
(8)

LN  
VO

|           |    |           |    |       |       |               |       |                     |   |      |
|-----------|----|-----------|----|-------|-------|---------------|-------|---------------------|---|------|
| 00000000G | 00 | 01        | FB | 00053 | CALLS | #1, LIB\$STOP | :     |                     |   |      |
| FC        | A2 | 00000000' | EF | 9E    | 0005A | 4\$:          | MOVAB | LIBFILTYPE, TYPEADR | : | 0812 |
|           |    |           |    | 04    | 00062 |               | RET   |                     | : | 0814 |

: Routine Size: 99 bytes, Routine Base: \$CODE\$ + 0401

: 708 0815 1

```

: 710      0816 1 routine optionfile : novalue =
: 711      0817 1
: 712      0818 1 !+
: 713      0819 1 !
: 714      0820 1 ! Set the default file type for link time options
: 715      0821 1 !
: 716      0822 1 !-
: 717      0823 2 begin
: 718      0824 2 bind   optfiltype = uplit (byte ('.OPT')) ;
: 719      0825 2
: 720      0826 2 typeadr = ch$ptr (optfiltype) ;
: 721      0827 2 return ;
: 722      0828 1 end ;

```

.PSECT \$SPLITS,NOWRT,NOEXE,2

54 50 4F 2E 001AC P.ABZ: .ASCII \.OPT\ ;

OPTFILTYPE= P.ABZ

.PSECT \$CODE\$,NOWRT,2

0000 00000 OPTIONFILE:

00000000' EF 00000000' EF 9E 00002  
04 0000D

.WORD Save nothing  
MOVAB OPTFILTYPE, TYPEADR  
RET

: 0816  
: 0826  
: 0828

: Routine Size: 14 bytes. Routine Base: \$CODE\$ + 0464

: 723 0829 1



```

: 725      0830 1 glcbl routine lnk$setshrblin : novalue =
: 726      0831 1
: 727      0832 1 !+
: 728      0833 1 !-
: 729      0834 1 ! Set the default file type for shareable images.
: 730      0835 1 !-
: 731      0836 1 !-
: 732      0837 2 begin
: 733      0838 2 typeadr = ch$ptr (shrfiltype) ;           ! Set default file type pointer
: 734      0839 2
: 735      0840 2 return ;
: 736      0841 1 end ;

```

```

00000000' EF 00000000' EF 0000 0000
04 0000D

```

```

.ENTRY LNK$SETSHRBLIN, Save nothing
MOVAB SHRFILTYPE, TYPEADR
RET

```

```

: 0830
: 0838
: 0841

```

: Routine Size: 14 bytes, Routine Base: \$CODE\$ + 0472

: 737 0842 1

```

: 739 0843 1 routine setuserlib =
: 740 0844 1
: 741 0845 1 !+
: 742 0846 1
: 743 0847 1 This routine is called by the CLI for the /USERLIBRARY qualifier.
: 744 0848 1 The default is /USERLIBRARY= (PROCESS, GROUP, SYSTEM). If there
: 745 0849 1 is a list of logical name tables, only those tables specified will
: 746 0850 1 be searched.
: 747 0851 1
: 748 0852 1 -
: 749 0853 2 begin
: 750 0854 2
: 751 0855 2 lnk$gl_ctlmsk [lnk$u_usrlib] = true ;
: 752 0856 2 while cli$get_value (sd_userlibrary, token_desc) ! Loop for all specified
: 753 0857 3 do begin
: 754 0858 3 if .token_desc [dsc$w_length] neq 0 ! If a string was there
: 755 0859 3 then
: 756 0860 3 if not call_tparse (token_desc [dsc$w_length], userlib_states, userlib_keys)
: 757 0861 3 then
: 758 0862 3 clierror (.token_desc, lnk$_illqualval) ; ! No return from this
: 759 0863 2 end ;
: 760 0864 2
: 761 0865 2 if .lnk$gl_ulibmask eql 0 ! If nothing found
: 762 0866 2 then lnk$gl_ulibmask = 7 ! Then enable all logical name tables
: 763 0867 2 else if .lnk$gl_ulibmask lss 0 ! will be lss 0 if /USERLIB=NONE
: 764 0868 2 then lnk$gl_ulibmask = 0 ;
: 765 0869 2
: 766 0870 2 return true ;
: 767 0871 1 end ; ! of setuserlib

```

000C 00000 SETUSERLIB:

|           |    |           |    |          |        |                        |        |
|-----------|----|-----------|----|----------|--------|------------------------|--------|
|           |    |           |    |          | .WORD  | Save R2,R3             | : 0843 |
|           | 53 | 00000000' | EF | 9E 00002 | MOVAB  | LNK\$GL_ULIBMASK, R3   |        |
|           | 52 | 00000000' | EF | 9E 00009 | MOVAB  | TOKEN_DESC, R2         |        |
| 00000000G | 00 | 40        | 8F | 88 00010 | BISB2  | #64, [LNK\$GL_CTLMSK+2 | : 0855 |
|           |    |           | 52 | DD 00018 | PUSHL  | R2                     | : 0856 |
|           |    | 00000000' | EF | 9F 0001A | PUSHAB | SD_USERLIBRARY         |        |
| 00000000G | 00 |           | 02 | FB 00020 | CALLS  | #2, CLI\$GET_VALUE     |        |
|           | 29 |           | 50 | E9 00027 | BLBC   | R0, 2\$                |        |
|           |    |           | 62 | B5 0002A | TSTW   | TOKEN_DESC             | : 0858 |
|           |    |           | EA | 13 0002C | BEQL   | 1\$                    |        |
|           |    | 00000000' | EF | 9F 0002E | PUSHAB | USERLIB_KEYS           | : 0860 |
|           |    | 00000000' | EF | 9F 00034 | PUSHAB | USERLIB_STATES         |        |
|           |    |           | 52 | DD 0003A | PUSHL  | R2                     |        |
| FDCA      | CF |           | 03 | FB 0003C | CALLS  | #3, CALL_TPARSE        |        |
|           | D4 |           | 50 | E8 00041 | BLBS   | R0, 1\$                |        |
|           |    | 00000000G | 8F | DD 00044 | PUSHL  | #LNK\$_ILLQUALVAL      | : 0862 |
|           |    |           | 62 | DD 0004A | PUSHL  | TOKEN_DESC             |        |
| FD9C      | CF |           | 02 | FB 0004C | CALLS  | #2, CLI\$ERROR         |        |
|           |    |           | C5 | 11 00051 | BRB    | 1\$                    | : 0856 |
|           | 50 |           | 63 | D0 00053 | MOVL   | LNK\$GL_ULIBMASK, R0   | : 0865 |
|           |    |           | 05 | 12 00056 | BNEQ   | 3\$                    |        |
|           | 63 |           | 07 | D0 00058 | MOVL   | #7, LNK\$GL_ULIBMASK   | : 0866 |

LNK\_GETCMDINP  
V04=000

E 7  
13-Sep-1984 23:57:45 YAX-11 Bliss-32 V4.0-742  
14-Sep-1984 12:40:30 [LINKER.SRC]LNKGETCMD.B32;1

Page 33  
(11)

LNK  
V04

50

|    |    |       |      |      |                  |
|----|----|-------|------|------|------------------|
| 04 | 11 | 0005B |      | BRB  | 4\$              |
| 02 | 18 | 0005D | 3\$: | BGEQ | 4\$              |
| 63 | D4 | 0005F |      | CLRL | LNK\$GL_ULIBMASK |
| 01 | D0 | 00061 | 4\$: | MOVL | #1, R0           |
|    | 04 | 00064 |      | RET  |                  |

:  
: 0867  
: 0868  
: 0870  
: 0871

: Routine Size: 101 bytes, Routine Base: \$CODE\$ + 0480

: 768 0872 1

: 1  
: 1  
: 1  
: 1  
: 1  
: 1  
: 1  
: 1  
: 1  
: 1

```
0873 1 global routine inputfile (desc, option_flag) : novalue =
0874 1
0875 1
0876 1
0877 1 This routine is called by the CLI for each file in the command input
0878 1 parameter list.
0879 1
0880 1 The routine allocates an internal file descriptor block and then, using the
0881 1 input file fab allocated in the command processing driver routine attempts
0882 1 to open the specified file.
0883 1
0884 1 If this is successful, the auxiliary filename block part of our FDB is
0885 1 initialized by RMS. Other useful fields of the FDB (such as the file flags)
0886 1 are initialized and then the FDB is added to the FDB list. The file is then
0887 1 closed, so that an open by file id may be performed on the file hereafter.
0888 1
0889 1 Failure to open the file at this point is fatal and the link aborts.
0890 1
0891 1
0892 1 REQSTDESC is the address of the input parameter descriptor being
0893 1 used by the CLI.
0894 1
0895 1 INFILFLGS contains the settings of the input file control flags
0896 1 as determined by the CLI and qualifiers used.
0897 1
0898 1
0899 2 begin
0900 2 routine get_module ( module_dsc ) =
0901 2
0902 2 This routine plucks the modules out of the include list passed here from
0903 2 LNK$PROCOPTNS and returns the individual module name descriptors to
0904 2 inputfile
0905 2
0906 2 begin
0907 2 local length,
0908 2 address,
0909 2 remaining_length ;
0910 2
0911 2 map module_dsc : ref block [,byte] ;
0912 2
0913 2 bind include_list = .lnk$gl_inclst : block [,byte] ; ! Here is the list...
0914 2
0915 2 if .listptr lss .include_list [dsc$a_pointer] ! If failure to find a module
0916 2 then ! zeroed the pointer, then
0917 2 return false ; ! quit here
0918 2
0919 2 address = .listptr ; ! Current module is already pointed to
0920 2 if ch$fail (listptr = ch$find_ch ! If we fail to find a delimiter
0921 2 (remaining_length =
0922 2 (.include_list [dsc$a_length] -
0923 2 (.listptr - .include_list [dsc$a_pointer]))
0924 2 )
0925 2 .listptr
0926 2 )
0927 2 )
0928 2 then length = .remaining_length ! then the length is what was left
0929 2 else length = .listptr - .address ; ! otherwise the length is the difference
```

```

: 827      0930      2
: 828      0931      2
: 829      0932      2
: 830      0933      2
: 831      0934      2
: 832      0935      2
: 833      0936      2
: 834      0937      2
: 835      0938      2
: 836      0939      2

```

```

if .length lss 0
then length = 0 ;

listptr = .listptr + 1 ;

str$copy_r ( .module_dsc, length, .address) ;

return true ;
end ;

```

```

: in the pointers
: Make sure module name length was > 0

: Move ptr 1 char past delimiter
: (tparse cleans out blanks and tabs)
: Do this because inputfile expects a
: dynamic descriptor

```

|    |    | 003C 00000 GET_MODULE: |           |    |       |       |        |                             |  |      |
|----|----|------------------------|-----------|----|-------|-------|--------|-----------------------------|--|------|
|    |    | 55                     | 00000000' | EF | 9E    | 00002 | .WORD  | Save R2,R3,R4,R5            |  | 0900 |
|    |    | 5E                     |           | 04 | C2    | 00009 | MOVAB  | LISTPTR, R5                 |  |      |
|    |    | 50                     | 00000000G | 00 | D0    | 0000C | SUBL2  | #4, SP                      |  | 0913 |
|    |    | 52                     |           | 65 | D0    | 00013 | MOVL   | LNK\$GL INCLST, R0          |  | 0915 |
|    | 04 | A0                     |           | 52 | D1    | 00016 | MOVL   | LISTPTR, R2                 |  |      |
|    |    |                        |           | 3D | 19    | 0001A | CPL    | R2, 4(R0)                   |  |      |
|    |    | 54                     |           | 52 | D0    | 0001C | BLSS   | 5\$                         |  | 0919 |
| 51 | 04 | A0                     |           | 52 | C3    | 0001F | MOVL   | R2, ADDRESS                 |  | 0923 |
|    |    | 53                     |           | 60 | 3C    | 00024 | SUBL3  | R2, 4(R0), R1               |  |      |
|    |    | 53                     |           | 51 | C0    | 00027 | MOVZWL | (R0), REMAINING_LENGTH      |  |      |
| 62 |    | 53                     |           | 2C | 3A    | 0002A | ADDL2  | R1, REMAINING_LENGTH        |  |      |
|    |    |                        |           | 02 | 12    | 0002E | LOCC   | #44, REMAINING_LENGTH, (R2) |  | 0921 |
|    |    |                        |           | 51 | D4    | 00030 | BNEQ   | 1\$                         |  |      |
|    |    | 65                     |           | 51 | D0    | 00032 | CLRL   | R1                          |  |      |
|    |    |                        |           | 05 | 12    | 00035 | MOVL   | R1, LISTPTR                 |  | 0927 |
|    |    | 6E                     |           | 53 | D0    | 00037 | BNEQ   | 2\$                         |  | 0928 |
|    |    |                        |           | 04 | 11    | 0003A | MOVL   | REMAINING_LENGTH, LENGTH    |  |      |
| 6E |    | 65                     |           | 54 | C3    | 0003C | BRB    | 3\$                         |  | 0929 |
|    |    |                        |           | 02 | 18    | 00040 | SUBL3  | ADDRESS, LISTPTR, LENGTH    |  | 0931 |
|    |    |                        |           | 6E | D4    | 00042 | BGEQ   | 4\$                         |  | 0932 |
|    |    |                        |           | 65 | D6    | 00044 | CLRL   | LENGTH                      |  | 0934 |
|    |    |                        |           | 54 | DD    | 00046 | INCL   | LISTPTR                     |  | 0936 |
|    |    |                        | 04        | AE | 9F    | 00048 | PUSHL  | ADDRESS                     |  |      |
|    |    |                        | 04        | AC | DD    | 0004B | PUSHAB | LENGTH                      |  |      |
|    |    | 00000000G              | 00        | 03 | FB    | 0004E | PUSHL  | MODULE DSC                  |  |      |
|    |    |                        | 50        | 01 | D0    | 00055 | CALLS  | #3, STR\$COPY_R             |  | 0938 |
|    |    |                        |           |    | 04    | 00058 | MOVL   | #1, R0                      |  |      |
|    |    |                        |           | 50 | D4    | 00059 | RET    |                             |  | 0939 |
|    |    |                        |           | 04 | 0005B |       | CLRL   | R0                          |  |      |
|    |    |                        |           |    |       |       | RET    |                             |  |      |

: Routine Size: 92 bytes, Routine Base: \$CODES + 04E5

: 837 0940 2

```

: 839 0941 ~ bind
: 840 0942 ~     sysinput_desc = $descriptor ('SYSSINPUT') : $bblock,
: 841 0943 ~     include_list  = .lnk$gl_inclst           : block [,byte] ;
: 842 0944 ~ literal
: 843 0945 ~     maxoptreclen = 132 ;                       ! Maximum length of option file records
: 844 0946 ~ map
: 845 0947 ~     desc          : ref block [,byte] ;       ! Really a pointer
: 846 0948 ~ builtin
: 847 0949 ~     nullparameter ;
: 848 0950 ~ local
: 849 0951 ~     dnm_desc      : block [8, byte],       ! Default filename descriptor
: 850 0952 ~     enm_desc   : block [8, byte],       ! LIB$FIND_FILE expanded file name
: 851 0953 ~     fnm_desc   : ref block [,byte],     ! File name descriptor in fdb
: 852 0954 ~     fildesblk  : ref block [,byte],     ! Pointer to fdb
: 853 0955 ~     auxfnb     : ref block [,byte],     ! Auxiliary file name block pointer
: 854 0956 ~     input_fab  : block [fab$c_bln, byte], ! Fab for the file opens.
: 855 0957 ~     optionrab  : block [rab$c_bln, byte], ! Rab for option file input stream
: 856 0958 ~     stvcode    :                       ! Stv value for error output
: 857 0959 ~     errorcode  :                       ! Rms error code
: 858 0960 ~     called_by_option ;                 ! Set if called by lnk$_procoptions
: 859 0961 ~
: 860 0962 ~ called_by_option = not nullparameter (2) ;
: 861 0963 ~
: 862 0964 ~     Allocate an FDB
: 863 0965 ~
: 864 0966 ~
: 865 0967 ~
: 866 0968 ~
: 867 0969 ~ lnk$allofdb ( fildesblk ) ;             ! Go allocate new file descriptor
: 868 0970 ~ auxfnb     = fildesblk [fdb$t_auxfnb] ; ! Point to filename block part
: 869 0971 ~ fnm_desc   = fildesblk [fdb$q_filename] ; ! Point to file name descriptor
: 870 0972 ~ copy_desc  (desc, fnm_desc) ;         ! Copy initial file name descriptor
: 871 0973 ~
: 872 0974 ~ fildesblk [fdb$b_filflgs] = .infilflgs and not fdb$m_newudf ; ! Copy the flags turning off bit 0
: 873 0975 ~
: 874 0976 ~
: 875 0977 ~
: 876 0978 ~     If this is a library with explicit module extraction, build the FDB's module list.
: 877 0979 ~     Call CLIS$GET_VALUE or GET_MODULE, based on whether or not the file is in an options file.
: 878 0980 ~
: 879 0981 ~
: 880 0982 ~ if (.infilflgs and fdb$m_libextr) neq 0 ! If explicit module extraction
: 881 0983 ~ then
: 882 0984 ~     begin
: 883 0985 ~     local      prevmodule : initial (0) ; ! Pointer to previous module name block
: 884 0986 ~
: 885 0987 ~     if .called_by_option ! If called by LNK$PROCOPTNS, we must
: 886 0988 ~     then listptr = .include_list [dsc$a_pointer] ; ! set up listptr
: 887 0989 ~
: 888 0990 ~     while (if not .called_by_option then cli$get_value (sd_include,module_desc)
: 889 0991 ~     else get_module (module_desc)
: 890 0992 ~     )
: 891 0993 ~     do begin
: 892 0994 ~     local      currentmodule : ref block [,byte] ; ! Ptr to current module name block
: 893 0995 ~
: 894 0996 ~     lnk$alloblk (.module_desc [dsc$w_length]+5,currentmodule) ; ! Allocate space to save name
: 895 0997 ~     if .prevmodule eql 0 ! If this is the first module,
```

```

: 896 0998 4 then fildesblk [fdb$l_omdlst] = currentmodule [0,0,0,0] ; ! then set the pointer in the fdb
: 897 0999 4 else .prevmodule = currentmodule [0,0,0,0] ; ! Otherwise, point prev to current
: 898 1000 4
: 899 1001 4 prevmodule = currentmodule [0,0,0,0] ; ! Reset for next pass
: 900 1002 4 currentmodule [0,0,32,0] = 0 ; ! Zero forward pointer
: 901 1003 4 currentmodule [4,0, 8,0] = .module_desc [dsc$w_length] ; ! Move in module length and name
: 902 1004 4
: 903 1005 4 ch$move (.module_desc [dsc$w_length], .module_desc [dsc$a_pointer], currentmodule [5,0,8,0]) ;
: 904 1006 4 end ;
: 905 1007 4 end ;
: 906 1008 4
: 907 1009 4
: 908 1010 4
: 909 1011 4 !
: 910 1012 4 ! Setup up buffer descriptors and call LIB$FIND_FILE to setup an expanded filename
: 911 1013 4 ! consistent with the stickiness rules for related file names and search lists.
: 912 1014 4 !
: 913 1015 4 infilflgs = 0 ; ! Reset file flags
: 914 1016 4
: 915 1017 4 dnm_desc [dsc$w_length] = typelen ; ! Setup default filename descriptor
: 916 1018 4 dnm_desc [dsc$b_class] = dsc$k_class_z ; ! for call to LIB$FIND_FILE
: 917 1019 4 dnm_desc [dsc$b_dtype] = dsc$k_dtype_z ;
: 918 1020 4 dnm_desc [dsc$a_pointer] = .typeadr ;
: 919 1021 4
: 920 1022 4 enm_desc [dsc$w_length] = 0 ; ! Init result filename descriptor
: 921 1023 4 enm_desc [dsc$b_class] = dsc$k_class_d ; ! for call to LIB$FIND_FILE
: 922 1024 4 enm_desc [dsc$b_dtype] = dsc$k_dtype_t ;
: 923 1025 4 enm_desc [dsc$a_pointer] = 0 ;
: 924 1026 4
: 925 1027 4 errorcode = lib$find_file (.desc, enm_desc ! Determine resultant name
: 926 1028 4 , rlf_context, dnm_desc
: 927 1029 4 , 0, stvcode, %ref(3)
: 928 1030 4 , 0, input_fab [fab$l_stv], %ref(3)
: 929 1031 4 ;
: 930 1032 4 if .rlf_context neq 0
: 931 1033 4 then
: 932 1034 4 begin
: 933 1035 4 local rss : initial (.auxfnb [nam$b_rss]), ! Original NAMS$RSS
: 934 1036 4 rsa : initial (.auxfnb [nam$l_rsa]), ! Original NAMS$L_RSA
: 935 1037 4 ess : initial (.auxfnb [nam$b_ess]), ! Original NAMS$ESS
: 936 1038 4 esa : initial (.auxfnb [nam$l_esa]); ! Original NAMS$L_ESS
: 937 1039 4
: 938 1040 4 ch$move (nam$c_bln, .rlf_context [fab$l_nam], .auxfnb) ; ! Copy NAM block
: 939 1041 4 ch$move (.enm_desc [dsc$w_length], .enm_desc [dsc$a_pointer], .esa) ; ! Copy the name
: 940 1042 4
: 941 1043 4 auxfnb [nam$l_wcc] = 0 ; ! Init ptr's we won't use
: 942 1044 4 auxfnb [nam$l_rlf] = 0 ;
: 943 1045 4 auxfnb [nam$l_rsa] = .rsa ; ! Re-setup resultant name area
: 944 1046 4 auxfnb [nam$b_rss] = .rss ;
: 945 1047 4 auxfnb [nam$b_rsl] = 0 ;
: 946 1048 4 auxfnb [nam$l_esa] = .esa ; ! Re-setup expanded name area
: 947 1049 4 auxfnb [nam$b_ess] = .ess ;
: 948 1050 4 auxfnb [nam$b_esl] = .enm_desc [dsc$w_length] ;
: 949 1051 4
: 950 1052 4 fildesblk [fdb$w_usrnamlen] = .auxfnb [nam$b_esl] ; ! Copy the user's filename
: 951 1053 4 fildesblk [fdb$l_usrnamadr] = .esa ; ! Into the fdb also
: 952 1054 4 copy_desc (enm_desc, fnm_desc) ; ! Get a better file name descriptor

```

```

953      1055      |
954      1056      | $fab_init (fab = input_fab      | for error message reporting
955      1057      |           ,fac = get           | Initialize the fab
956      1058      |           ,fop = sqo          | For input
957      1059      |           ,dna = .typeadr     |
958      1060      |           ,dns = typelen     |
959      1061      |           ,fna = .esa         |
960      1062      |           ,fns = .auxfnb [nam$ |
961      1063      |           ,nam = .auxfnb      |
962      1064      |           ,shr = (upi,get,put) |
963      1065      |           )                   |
964      1066      | input_fab [fab$l_stv] = .stvc | ! Restore LIB$FIND_FILE stv value
965      1067      | end ;                          |
966      1068      |
967      1069      |
968      1070      |
969      1071      | If the parse/setup was success |
970      1072      | ful, open the file to obtain  |
971      1073      | the file i.d., etc.          |
972      1074      |
973      1075      | if .errorcode                 |
974      1076      | then begin                    |
975      1077      |   if .fildesblk [fdb$v_libr]  | ! Library file
976      1078      |   then begin                 | ! Library open by name block
977      1079      |     auxfnb [nam$b_rsl] = .aux | ! Setup resultant name for map
978      1080      |     auxfnb [nam$b_rss] = .aux | ! contents
979      1081      |     auxfnb [nam$l_rsa] = .aux |
980      1082      |     lnk$gl_curfil = .fildesb |
981      1083      |     errorcode = lnk$openlib (. | ! Set ptr to current fdb for the temporary
982      1084      |     auxfnb, .an_open_fdb) ;  | ! benefit of LNK$OPENLIB
983      1085      |     lnk$gl_curfil = 0 ;      | ! Must now clear it -- causes problems
984      1086      |   end                       | ! later if non-zero
985      1087      |   else begin                 | ! Not a library file
986      1088      |     $getjpi (itmlst = lnk$gt_ | ! Close a file if we're near the limit
987      1089      |     jpilst) ;                |
988      1090      |     if .lnk$gl_filesleft [eq |
989      1091      |     3 then if .an_open_fdb ne |
990      1092      |     g 0 then lnk$closefile (. |
991      1093      |     an_open_fdb) ;          |
992      1094      |   if .fildesblk [fdb$v_shr]  | ! If shareable image, gotta do:
993      1095      |   then begin                 |
994      1096      |     input_fab [fab$v_sqo]    | ! Not sequential only
995      1097      |     = false ;                | ! Record and block operations
996      1098      |     input_fab [fab$v_bro]    | ! Shared access
997      1099      |     = true ;                 | ! Set for shared gets
998      1100      |     input_fab [fab$v_upi]    | ! and puts (linker doesn't write to
999      1101      |     = true ;                 | ! it, but must set so we can read it
1000     1102      |     input_fab [fab$v_shrget] | ! in case installed /write)
1001     1103      |     = true ;                 | ! Do regular $open
1002     1104      |     input_fab [fab$v_shrput] | ! Remember IFI
1003     1105      |     = true ;                 |
1004     1106      |   end ;                       |
1005     1107      |   errorcode = $open (fab=input |
1006     1108      |   fab) ;                     |
1007     1109      |   fildesblk [fdb$w_ifi] = .in |
1008     1110      |   put_fab [fab$w_ifi] ;      |
1009     1111      | end ;                          |

```

If ERRORCODE has its low bit set at this point then the file open was successful.



```

: 1010      1112 2 if not .errorcode
: 1011      1113 2 then stvcode = .input_fab [fab$l_stv]
: 1012      1114 2 else begin
: 1013      1115 2   if .lnk$gl_inrelnam eql 0
: 1014      1116 2   then if setup_inrelnam (1,.fnm_desc, dflt_inrel, dflt_name)
: 1015      1117 2     then lnk$gl_inrelnam = dflt_inrel ;
: 1016      1118 2
: 1017      1119 2   if not .called_by_option
: 1018      1120 2   then last_fnm_desc = .fnm_desc ;
: 1019      1121 2
: 1020      1122 2   if not .fildesblk [fdb$sv_option]
: 1021      1123 2   then begin
: 1022      1124 2     .....
: 1023      1125 2     Not an OPTIONS file. Simply stick the FDB at the end
: 1024      1126 2     of the current cluster FDB list.
: 1025      1127 2
: 1026      1128 2     local lastfdb : ref block [,byte] initial (.lnk$gl_curclu [clu$l_lstfdb]) ;
: 1027      1129 2
: 1028      1130 2     lastfdb [fdb$l_nxtfdb] = .fildesblk ;           ! Put FDB at end of current cluster
: 1029      1131 2     lnk$gl_curclu [clu$l_lstfdb] = .fildesblk ;       ! Remember end of list.
: 1030      1132 2     end
: 1031      1133 2   else begin
: 1032      1134 2     .....
: 1033      1135 2     OPTIONS file. Connect to it and process the records inside it.
: 1034      1136 2     The entire contents of the OPTIONS file begins a new context for
: 1035      1137 2     related filename processing - beginning with 'SYSSDISK:[]'.
: 1036      1138 2
: 1037      1139 2     bind namdvi = auxfnb [nam$st_dvi] : vector [,byte] ; ! Addr of nam block dvi
: 1038      1140 2
: 1039      1141 2     lnk$gl_sysinput_flag =                               ! Setup 'sysinput' flag
: 1040      1142 2     begin
: 1041      1143 2       if .bblock [input_fab [fab$l_dev], dev$sv_trm] ! Is device a terminal?
: 1042      1144 2       or (.auxfnb [nam$sv_ppf]                          ! A process permanent file?
: 1043      1145 2         and
: 1044      1146 2         ch$eql (.sysinput_desc [dsc$w_length]        ! SYSSINPUT ?
: 1045      1147 2         ..sysinput_desc [dsc$a_pointer]
: 1046      1148 2         ;.namdvi [0], namdvi [1], 0
: 1047      1149 2       )
: 1048      1150 2       then 1
: 1049      1151 2       else 0
: 1050      1152 2     end ;
: 1051      1153 2
: 1052      1154 2     $rab_init      (rab = optionrab
: 1053      1155 2     .fab = input_fab
: 1054      1156 2     .usz = maxopfreclen
: 1055      1157 2     .rac = seq
: 1056      1158 2     .rop = loc
: 1057      1159 2     ) ;
: 1058      1160 2
: 1059      1161 2     if not (errorcode = $connect (rab=optionrab)) ! Connect the options file for reading
: 1060      1162 2     then stvcode = .optionrab [rab$l_stv]
: 1061      1163 2     else begin
: 1062      1164 2       ! And if connected ok
: 1063      1165 2       local save_context : initial (.rlf_context) ; ! Save current context
: 1064      1166 2
: 1065      1167 2
: 1066      1168 2

```

P  
P  
P  
P

```

: 1067      1169 5      lnk$alloblk (maxoptreclen, optionrab [rab$l_ubf]) ; ! Allocate buffer for records
: 1068      1170 5      crossing block boundaries
: 1069      1171 5      typeadr      = ch$ptr (defiletype) ;      Reset the default file type
: 1070      1172 5      rlf context  = 0 ;      Start new related name context
: 1071      1173 5      lnk$procopts (.fnm_desc, infilflgs, optionrab) ; Go process it
: 1072      1174 5      rlf context  = .save_context ; Restore related name context
: 1073      1175 5      lnk$gl_curclu = lnk$gl_defclu ; And reset default as current
: 1074      1176 5      lnk$dealkblk (maxoptreclen, .optionrab [rab$l_ubf]) ; ! And deallocate
: 1075      1177 5      end ;
: 1076      1178 3      end ;
: 1077      1179 2      end ;
: 1078      1180 2
: 1079      1181 2      if not .errorcode      ! Report any errors
: 1080      1182 2      then signal_stop (lin$_openin, 1, .fnm_desc, .errorcode, .stvcde) ;
: 1081      1183 2
: 1082      1184 2
: 1083      1185 2
: 1084      1186 2      For performance reasons, only close the file if we have to -- if its an Options file
: 1085      1187 2      or a shareable image
: 1086      1188 2
: 1087      1189 2
: 1088      1190 2      if .fildesblk [fdb$_v_option] or .fildesblk [fdb$_v_imglib]
: 1089      1191 2      then lnk$closefile (.fildesblk)
: 1090      1192 2      else an_open_fdb = .fildesblk ;
: 1091      1193 2
: 1092      1194 2      typeadr = ch$ptr (defiletype) ;      ! Reset the default file type
: 1093      1195 2      return ;
: 1094      1196 1      end ;

```

```

.PSECT $PLITS,NOWRT,NOEXE,2
54 55 50 4E 49 24 53 59 53 001B0 P.ACB: .ASCII \SYSS$INPUT\ ;
001B9 .BLKB 3 ;
00000009 001BC P.ACA: .LONG 9 ;
00000000' 001C0 .ADDRESS P.ACB ;
SYSINPUT_DESC= P.ACA
.EXTRN SYSS$GETJPI, SYSS$OPEN
.EXTRN SYSS$CONNECT
.PSECT $CODE$,NOWRT,2
OFFC 00000 .ENTRY INPUTFILE, Save R2,R3,R4,R5,R6,R7,R8,R9,- ; 0873
R10,R11
5E FF40 CE 9E 00002 MOVAB -192(SP), SP ;
52 00000000G 00 D0 00007 MOVL LNK$GL_INCLST, R2 ; 0943
02 6C 91 0000E CMPB (AP), #2 ; 0962
58 05 1E 00011 BGEQU 1$ ;
01 D0 00013 MOVL #1, R8 ;
09 11 00016 BRB 2$ ;
58 D4 00018 1$: CLRL R8 ;
08 AC D5 0001A TSTL 8(AP) ;
02 12 0001D BNEQ 2$ ;
58 D6 0001F INCL R8 ;
58 58 D2 00021 2$: MCOML R8, CALLED_BY_OPTION ;

```

|    |                 |    |           |           |    |       |        |                                    |                     |
|----|-----------------|----|-----------|-----------|----|-------|--------|------------------------------------|---------------------|
|    | 00000000V       | EF | 10        | AE        | 9F | 00024 | PUSHAB | FILDESBLK                          | 0969                |
|    |                 | 5A | 10        | AE        | FB | 00027 | CALLS  | #1, LNK\$ALLOFDB                   | 0970                |
|    |                 | 56 | 26        | AA        | 9E | 00032 | MOVL   | FILDESBLK, R10                     | 0971                |
|    |                 | 57 | 14        | AA        | 9E | 00036 | MOVAB  | 38(R10), AUXFNB                    | 0972                |
|    |                 | 59 | 04        | AC        | 9E | 0003A | MOVAB  | 20(R10), FNM_DESC                  | 0974                |
|    |                 | 67 |           | AC        | 9E | 0003E | MOVL   | DESC, R9                           | 0982                |
|    |                 | 6E | 0A        | 69        | 7D | 0003E | MOVQ   | (R9), (FNM_DESC)                   | 0984                |
| 00 | BE 00000000'    | EF |           | AA        | 9E | 00041 | MOVAB  | 10(R10), (SP)                      | 0987                |
|    | 79 00000000'    | EF |           | 01        | 8B | 00045 | BICB3  | #1, INFILFLGS, @0(SP)              | 0988                |
|    |                 |    |           | 06        | E1 | 0004E | BBC    | #6, INFILFLGS, 9\$                 | 0990                |
|    |                 |    |           | 5B        | D4 | 00056 | CLRL   | PREVMODULE                         |                     |
|    |                 | 0B |           | 58        | E9 | 00058 | BLBC   | CALLED_BY_OPTION, 3\$              |                     |
|    | 00000000'       | EF | 04        | A2        | D0 | 0005B | MOVL   | 4(R2), -LISTPTR                    |                     |
|    | OC              | AE |           | 58        | D2 | 00063 | MCOML  | CALLED_BY_OPTION, 12(SP)           |                     |
|    |                 | 15 | OC        | AE        | E9 | 00067 | BLBC   | 12(SP), 5\$                        |                     |
|    |                 |    | 00000000' | EF        | 9F | 0006B | PUSHAB | MODULE_DESC                        |                     |
|    |                 |    | 00000000' | EF        | 9F | 00071 | PUSHAB | SD_INCLUDE                         |                     |
|    | 00000000G       | 00 |           | 02        | FB | 00077 | CALLS  | #2, CLISGET_VALUE                  |                     |
|    |                 |    |           | 0B        | 11 | 0007E | BRB    | 6\$                                |                     |
|    |                 |    | 00000000' | EF        | 9F | 00080 | PUSHAB | MODULE_DESC                        | 0991                |
|    | FF19            | CF |           | 01        | FB | 00086 | CALLS  | #1, GET_MODULE                     |                     |
|    |                 | 41 |           | 50        | E9 | 0008B | BLBC   | R0, 9\$                            |                     |
|    |                 |    | 14        | AE        | 9F | 0008E | PUSHAB | CURRENTMODULE                      | 0996                |
|    |                 | 7E | 00000000' | EF        | 3C | 00091 | MOVZWL | MODULE_DESC, -(SP)                 |                     |
|    |                 | 6E |           | 05        | C0 | 00098 | ADDL2  | #5, (SP)                           |                     |
|    | 00000000G       | 00 |           | 02        | FB | 0009B | CALLS  | #2, LNK\$ALLOBLK                   |                     |
|    |                 | 50 | 14        | AE        | D0 | 000A2 | MOVL   | CURRENTMODULE, R0                  | 0998                |
|    |                 |    |           | 5B        | D5 | 000A6 | TSTL   | PREVMODULE                         | 0997                |
|    |                 |    |           | 06        | 12 | 000A8 | BNEQ   | 7\$                                |                     |
|    | 04              | AA |           | 50        | D0 | 000AA | MOVL   | R0, 4(R10)                         | 0998                |
|    |                 |    |           | 03        | 11 | 000AE | BRB    | 8\$                                |                     |
|    |                 | 6B |           | 50        | D0 | 000B0 | MOVL   | R0, (PREVMODULE)                   | 0999                |
|    |                 | 5B |           | 50        | D0 | 000B3 | MOVL   | R0, PREVMODULE                     | 1001                |
|    |                 |    |           | 60        | D4 | 000B6 | CLRL   | (R0)                               | 1002                |
| 05 | 04 A0 00000000' | A0 | 00000000' | EF        | 90 | 000B8 | MOVQ   | MODULE_DESC, 4(R0)                 | 1003                |
|    |                 | FF | 00000000' | EF        | 28 | 000C0 | MOVC3  | MODULE_DESC, @MODULE_DESC+4, 5(R0) | 1005                |
|    |                 |    |           | 98        | 11 | 000CD | BRB    | 4\$                                | 0990                |
|    |                 |    | 00000000' | EF        | D4 | 000CF | CLRL   | INFILFLGS                          | 1015                |
|    |                 | F8 | AD        | 04        | D0 | 000D5 | MOVL   | #4, DNM_DESC                       | 1017                |
|    |                 | FC | AD        | 00000000' | EF | D0    | 000D9  | MOVL                               | TYPEADR, DNM_DESC+4 |
|    |                 | F0 | AD        | 020E0000  | 8F | D0    | 000E1  | MOVL                               | #34471936, ENM_DESC |
|    |                 |    | F4        | AD        | D4 | 000E9 | CLRL   | ENM_DESC+4                         | 1022                |
|    |                 | OC | AE        | 03        | D0 | 000EC | MOVL   | #3, -12(SP)                        | 1025                |
|    |                 |    | OC        | AE        | 9F | 000F0 | PUSHAB | 12(SP)                             | 1029                |
|    |                 |    | 1C        | AE        | 9F | 000F3 | PUSHAB | STVCODE                            |                     |
|    |                 |    |           | 7E        | D4 | 000F6 | CLRL   | -(SP)                              | 1027                |
|    |                 |    | F8        | AD        | 9F | 000F8 | PUSHAB | DNM_DESC                           |                     |
|    |                 |    | 00000000' | EF        | 9F | 000FB | PUSHAB | RLF_CONTEXT                        |                     |
|    |                 |    | F0        | AD        | 9F | 00101 | PUSHAB | ENM_DESC                           |                     |
|    |                 |    |           | 59        | DD | 00104 | PUSHL  | R9                                 |                     |
|    | 00000000G       | 00 |           | 07        | FB | 00106 | CALLS  | #7, LIB\$FIND FILE                 |                     |
|    |                 | OC |           | 50        | D0 | 0010D | MOVL   | R0, ERRORCODE                      |                     |
|    |                 |    | 00000000' | EF        | D0 | 00111 | MOVL   | RLF_CONTEXT, R0                    | 1032                |
|    |                 |    |           | 03        | 12 | 00118 | BNEQ   | 10\$                               |                     |
|    |                 |    |           | 0085      | 31 | 0011A | BRW    | 11\$                               |                     |
|    |                 | 08 | AE        | 02        | A6 | 9A    | 0011D  | MOVZBL                             | 2(AUXFNB), RSS      |
|    |                 |    | 5B        | 04        | A6 | D0    | 00122  | MOVL                               | 4(AUXFNB), RSA      |

|      |    |           |    |              |      |       |       |                 |                               |                  |      |
|------|----|-----------|----|--------------|------|-------|-------|-----------------|-------------------------------|------------------|------|
|      |    | 04        | AE | 0A           | A6   | 9A    | 00126 | MOVZBL          | 10(AUXFNB), ESS               | 1037             |      |
|      |    | 59        | OC | OC           | A6   | DO    | 0012B | MOVL            | 12(AUXFNB), ESA               | 1038             |      |
| 66   |    | 28        | B0 | 0060         | 8F   | 28    | 0012F | MOVCS           | #96, @40(R0), (AUXFNB)        | 1040             |      |
| 69   |    | F4        | BD | F0           | AD   | 28    | 00136 | MOVCS           | ENM_DESC, @ENM_DESC+4, (ESA)  | 1041             |      |
|      |    |           |    | 30           | A6   | D4    | 0013C | CLRL            | 48(AUXFNB)                    | 1043             |      |
|      |    |           |    | 10           | A6   | D4    | 0013F | CLRL            | 16(AUXFNB)                    | 1044             |      |
|      |    | 04        | A6 | 5B           | DO   | 00142 | MOVL  | RSA, 4(AUXFNB)  | 1045                          |                  |      |
|      |    | 02        | A6 | 08           | AE   | 9B    | 00146 | MOVZBW          | RSS, 2(AUXFNB)                | 1046             |      |
|      |    | 0C        | A6 | 59           | DO   | 0014B | MOVL  | ESA, 12(AUXFNB) | 1048                          |                  |      |
|      |    | 0A        | A6 | 04           | AE   | 90    | 0014F | MOVBS           | ESS, 10(AUXFNB)               | 1049             |      |
|      |    | 0B        | A6 | F0           | AD   | 90    | 00154 | MOVBS           | ENM_DESC, 11(AUXFNB)          | 1050             |      |
|      |    | 0C        | AA | 0B           | A6   | 9B    | 00159 | MOVZBW          | 11(AUXFNB), 12(R10)           | 1052             |      |
|      |    | 10        | AA | 59           | DO   | 0015E | MOVL  | ESA, 16(R10)    | 1053                          |                  |      |
| 0050 | 8F |           |    | F0           | AD   | 7D    | 00162 | MOVQ            | ENM_DESC, (FNM_DESC)          | 1054             |      |
|      |    | 00        |    | 6E           | 00   | 2C    | 00166 | MOVCS           | #0, -(SP), #0, #80, \$RMS_PTR | 1065             |      |
|      |    |           |    | 60           | AE   |       | 0016D |                 |                               |                  |      |
|      |    | 60        | AE | 5003         | 8F   | B0    | 0016F | MOVW            | #20483, \$RMS_PTR             |                  |      |
|      |    | 64        | AE | 40           | 8F   | 9A    | 00175 | MOVZBL          | #64, \$RMS_PTR+4              |                  |      |
|      |    | 76        | AE | 4302         | 8F   | B0    | 0017A | MOVW            | #17154, \$RMS_PTR+22          |                  |      |
|      |    | 7F        | AE |              | 02   | 90    | 00180 | MOVBS           | #2, \$RMS_PTR+31              |                  |      |
|      |    | C8        | AD |              | 56   | DO    | 00184 | MOVL            | AUXFNB, \$RMS_PTR+40          |                  |      |
|      |    | CC        | AD |              | 59   | DO    | 00188 | MOVL            | ESA, \$RMS_PTR+44             |                  |      |
|      |    | DO        | AD | 00000000'    | EF   | DO    | 0018C | MOVL            | TYPEADR, \$RMS_PTR+48         |                  |      |
|      |    | D4        | AD | 0B           | A6   | 90    | 00194 | MOVBS           | 11(AUXFNB), \$RMS_PTR+52      |                  |      |
|      |    | D5        | AD |              | 04   | 90    | 00199 | MOVBS           | #4, \$RMS_PTR+53              |                  |      |
|      |    | 6C        | AE | 18           | AE   | DO    | 0019D | MOVL            | STVCODE, -INPUT_FAB+12        | 1066             |      |
|      |    |           | 03 | 0C           | AE   | E8    | 001A2 | 11\$:           | BLBS                          | ERRORCODE, 12\$  | 1075 |
|      |    |           |    |              | 0096 | 31    | 001A6 | BRW             | 17\$                          |                  |      |
| 2C   |    | 00        | BE |              | 01   | E1    | 001A9 | 12\$:           | BBC                           | #1, @0(SP), 13\$ | 1077 |
|      |    | 02        | A6 | 0A           | A6   | B0    | 001AE | MOVW            | 10(AUXFNB), 2(AUXFNB)         | 1080             |      |
|      |    | 04        | A6 | 0C           | A6   | DO    | 001B3 | MOVL            | 12(AUXFNB), 4(AUXFNB)         | 1081             |      |
|      |    | 00000000G | 00 |              | 5A   | DO    | 001B8 | MOVL            | R10, LNK\$GL_CURFIL           | 1082             |      |
|      |    |           |    | 00000000'    | EF   | DD    | 001BF | PUSHL           | AN_OPEN_FDB                   | 1083             |      |
|      |    | 00000000G | 00 |              | 56   | DD    | 001C5 | PUSHL           | AUXFNB                        |                  |      |
|      |    | 0C        | AE |              | 02   | FB    | 001C7 | CALLS           | #2, LNK\$OPENLIB              |                  |      |
|      |    |           |    | 00000000G    | 50   | DO    | 001CE | MOVL            | R0, ERRORCODE                 |                  |      |
|      |    |           |    |              | 00   | D4    | 001D2 | CLRL            | LNK\$GL_CURFIL                | 1084             |      |
|      |    |           |    |              | 53   | 11    | 001D8 | BRB             | 16\$                          | 1077             |      |
|      |    |           |    |              | 7E   | 7C    | 001DA | 13\$:           | CLRQ                          | -(SP)            | 1087 |
|      |    |           |    |              | 7E   | D4    | 001DC | CLRL            | -(SP)                         |                  |      |
|      |    |           |    | 00000000'    | EF   | 9F    | 001DE | PUSHAB          | LNK\$GT_JPILST                |                  |      |
|      |    |           |    |              | 7E   | 7C    | 001E4 | CLRQ            | -(SP)                         |                  |      |
|      |    |           |    |              | 7E   | D4    | 001E6 | CLRL            | -(SP)                         |                  |      |
|      |    | 00000000G | 00 |              | 07   | FB    | 001E8 | CALLS           | #7, SYSS\$GETJPI              |                  |      |
|      |    |           | 03 | 00000000'    | EF   | D1    | 001EF | CMPL            | LNK\$GL_FILESLEFT, #3         | 1088             |      |
|      |    |           |    |              | 12   | 14    | 001F6 | BGTR            | 14\$                          |                  |      |
|      |    |           |    | 50 00000000' | EF   | DO    | 001F8 | MOVL            | AN_OPEN_FDB, R0               | 1089             |      |
|      |    |           |    |              | 09   | 13    | 001FF | BEQL            | 14\$                          |                  |      |
|      |    | 00000000G | 00 |              | 50   | DD    | 00201 | PUSHL           | R0                            | 1090             |      |
| 0B   |    | 00        | BE |              | 01   | FB    | 00203 | CALLS           | #1, LNK\$CLOSEFILE            |                  |      |
|      |    | 64        | AE | 40           | 02   | E1    | 0020A | 14\$:           | BBC                           | #2, @0(SP), 15\$ | 1092 |
|      |    | 76        | AE | 4340         | 8F   | 8A    | 0020F | BICB2           | #64, INPUT_FAB+4              | 1094             |      |
|      |    |           |    | 60           | 8F   | AB    | 00214 | BISW2           | #17216, INPUT_FAB+23          | 1098             |      |
|      |    | 00000000G | 00 |              | AE   | 9F    | 0021A | 15\$:           | PUSHAB                        | INPUT_FAB        | 1101 |
|      |    | 0C        | AF |              | 01   | FB    | 0021D | CALLS           | #1, SYSS\$OPEN                |                  |      |
|      |    | 24        | AA | 62           | 50   | DO    | 00224 | MOVL            | R0, ERRORCODE                 |                  |      |
|      |    |           |    |              | AE   | B0    | 00228 | MOVW            | INPUT_FAB+2, 36(R10)          | 1102             |      |

|      |           |           |           |           |    |       |       |        |  |                           |      |
|------|-----------|-----------|-----------|-----------|----|-------|-------|--------|--|---------------------------|------|
| 67   | 00000000G | 00        | 60        | AE        | 9F | 0022D | 16\$: | PUSHAB | INPUT FAB                                  | 1104                      |      |
|      |           | 60        |           | 01        | FB | 00230 |       | CALLS  | #1, LNK\$FILNAMDSC                         |                           |      |
|      |           | 07        |           | 08        | 28 | 00237 |       | MOVCS  | #8, (R0), (FNM_DESC)                       |                           |      |
|      | 18        | AE        | 0C        | AE        | EB | 0023B |       | BLBS   | ERRORCODE, 18\$                            | 1112                      |      |
|      |           |           | 6C        | AE        | DO | 0023F | 17\$: | MOVL   | INPUT_FAB+12, STVCODE                      | 1113                      |      |
|      |           |           |           | 4E        | 11 | 00244 |       | BRB    | 21\$                                       |                           |      |
|      |           | 00000000' |           | EF        | D5 | 00246 | 18\$: | TSTL   | LNK\$GL_INRELNAM                           | 1115                      |      |
|      |           |           |           | 25        | 12 | 0024C |       | BNEQ   | 19\$                                       |                           |      |
|      |           | 00000000' |           | EF        | 9F | 0024E |       | PUSHAB | DFLT_NAME                                  | 1116                      |      |
|      |           | 00000000' |           | EF        | 9F | 00254 |       | PUSHAB | DFLT_INREL                                 |                           |      |
|      |           |           |           | 57        | DD | 0025A |       | PUSHL  | FNM_DESC                                   |                           |      |
|      | 00000000V | EF        |           | 01        | DD | 0025C |       | PUSHL  | #1   |                           |      |
|      |           | 0B        |           | 04        | FB | 0025E |       | CALLS  | #4, SETUP_INRELNAM                         |                           |      |
|      | 00000000' | EF        | 00000000' | 50        | E9 | 00265 |       | BLBC   | R0, 19\$                                   |                           |      |
|      |           | 07        |           | EF        | 9E | 00268 |       | MOVAB  | DFLT_INREL, LNK\$GL_INRELNAM               | 1117                      |      |
|      | 00000000' | EF        |           | 58        | E8 | 00273 | 19\$: | BLBS   | CALLED BY_OPTION, 20\$                     | 1119                      |      |
|      |           | 07        |           | 57        | DO | 00276 |       | MOVL   | FNM_DESC, LAST_FNM_DESC                    | 1120                      |      |
| 14   | 00000000' | EF        |           | 04        | EO | 0027D | 20\$: | BBS    | #4, @0(SP), 22\$                           | 1122                      |      |
|      | 00        | BE        |           | 00        | DO | 00282 |       | MOVL   | LNK\$GL_CURCLU, R0                         | 1130                      |      |
|      |           | 50        | 00000000G | A0        | DO | 00289 |       | MOVL   | 12(R0), LASTFDB                            |                           |      |
|      |           | 51        | 0C        | 5A        | DO | 0028D |       | MOVL   | R10, (LASTFDB)                             | 1132                      |      |
|      |           | 61        |           | 5A        | DO | 00290 |       | MOVL   | R10, 12(R0)                                | 1133                      |      |
|      |           | 0C        |           | 6A        | 11 | 00294 | 21\$: | BRB    | 26\$                                       | 1122                      |      |
|      |           | 50        | 14        | A6        | 9E | 00296 | 22\$: | MOVAB  | 20(AUXFNB), R0                             | 1143                      |      |
| 18   | E0        | AD        |           | 02        | E0 | 0029A |       | BBS    | #2, INPUT_FAB+64, 23\$                     | 1147                      |      |
|      |           | 19        | 36        | A6        | E9 | 0029F |       | BLBC   | 54(AUXFNB), 24\$                           | 1148                      |      |
|      |           | 51        |           | 60        | 9A | 002A3 |       | MOVZBL | (R0), R1                                   | 1152                      |      |
| 51   | 00        | 00000000' | FF        | 00000000' | EF | 2D    | 002A6 | CMPCS  | SYSINPUT_DESC, @SYSINPUT_DESC+4, #0, R1, - |                           |      |
|      |           |           | 01        | A0        |    | 002B3 |       |        | 1(R0)                                      |                           |      |
|      |           |           |           | 05        | 12 | 002B5 |       | BNEQ   | 24\$                                       |                           |      |
|      |           |           | 50        | 01        | DO | 002B7 | 23\$: | MOVL   | #1, R0                                     | 1147                      |      |
|      |           |           |           | 02        | 11 | 002BA |       | BRB    | 25\$                                       |                           |      |
|      | 00000000' | EF        |           | 50        | D4 | 002BC | 24\$: | CLRL   | R0   |                           |      |
| 0044 | 8F        | 00        | 00000000' | 50        | DO | 002BE | 25\$: | MOVL   | R0, LNK\$GL_SYSINPUT_FLAG                  | 1146                      |      |
|      |           | 6E        |           | 00        | 2C | 002C5 |       | MOVCS  | #0, (SP), #0, #68, \$RMS_PTR               | 1163                      |      |
|      |           |           | 1C        | AE        |    | 002CC |       |        |  |                           |      |
|      | 1C        | AE        | 4401      | 8F        | B0 | 002CE |       | MOVW   | #17409, \$RMS_PTR                          |                           |      |
|      | 20        | AE        | 00010000  | 8F        | DO | 002D4 |       | MOVL   | #65536, \$RMS_PTR+4                        |                           |      |
|      |           |           | 3A        | AE        | 94 | 002DC |       | CLRB   | \$RMS_PTR+30                               |                           |      |
|      | 3C        | AE        | 84        | 8F        | 9B | 002DF |       | MOVZBW | #132, \$RMS_PTR+32                         |                           |      |
|      | 58        | AE        | 60        | AE        | 9E | 002E4 |       | MOVAB  | INPUT_FAB, \$RMS_PTR+60                    |                           |      |
|      |           |           | 1C        | AE        | 9F | 002E9 |       | PUSHAB | OPTIONRAB                                  | 1164                      |      |
|      | 00000000G | 00        |           | 01        | FB | 002EC |       | CALLS  | #1, SYSSCONNECT                            |                           |      |
|      |           | 0C        |           | 50        | DO | 002F3 |       | MOVL   | P0, ERRORCODE                              |                           |      |
|      |           | 07        | 0C        | AE        | EB | 002F7 |       | BLBS   | ERRORCODE, 27\$                            |                           |      |
|      |           | 18        |           | AE        | DO | 002FB |       | MOVL   | OPTIONRAB+12, STVCODE                      | 1165                      |      |
|      |           |           |           | 58        | 11 | 00300 | 26\$: | BRB    | 28\$                                       |                           |      |
|      |           |           | 52        | 00000000' | EF | DO    | 00302 | 27\$:  | MOVL                                       | RLF_CONTEXT, SAVE_CONTEXT | 1167 |
|      |           |           |           | 40        | AE | 9F    | 00309 |        | PUSHAB                                     | OPTIONRAB+36              | 1169 |
|      |           |           | 7E        | 84        | 8F | 9A    | 0030C |        | MOVZBL                                     | #132, -(SP)               |      |
|      | 00000000G | 00        |           | 02        | FB | 00310 |       | CALLS  | #2, LNK\$ALLOBLK                           |                           |      |
|      | 00000000' | EF        | 00000000' | EF        | 9E | 00317 |       | MOVAB  | DEFILETYPE, TYPEADR                        | 1171                      |      |
|      |           |           | 00000000' | EF        | D4 | 00322 |       | CLRL   | RLF_CONTEXT                                | 1172                      |      |
|      |           |           |           | 1C        | AE | 9F    | 00328 |        | PUSHAB                                     | OPTIONRAB                 | 1173 |
|      |           |           | 00000000' | EF        | 9F | 0032B |       | PUSHAB | INFILFLGS                                  |                           |      |
|      |           |           |           | 57        | DD | 00331 |       | PUSHL  | FNM_DESC                                   |                           |      |
|      | 00000000G | 00        |           | 03        | FB | 00333 |       | CALLS  | #3, LNK\$PROCOPTS                          |                           |      |

|           |    |           |    |       |       |        |                                |                     |      |      |
|-----------|----|-----------|----|-------|-------|--------|--------------------------------|---------------------|------|------|
| 00000000' | EF |           | 52 | DD    | 0033A | MOVL   | SAVE CONTEXT, RLF CONTEXT      | :                   | 1174 |      |
| 00000000G | 00 | 00000000G | 00 | 9E    | 00341 | MOVAB  | LNK\$GL_DEFCLU, LNK\$GL_CURCLU | :                   | 1175 |      |
|           |    | 40        | AE | DD    | 0034C | PUSHL  | OPTIONRAB+36                   | :                   | 1176 |      |
|           | 7E | 84        | 8F | 9A    | 0034F | MOVZBL | #132, -(SP)                    | :                   |      |      |
| 00000000G | 00 |           | 02 | FB    | 00353 | CALLS  | #2, LNK\$DEALBLK               | :                   |      |      |
|           | 17 | 0C        | AE | EB    | 0035A | 28\$:  | BLBS                           | ERRORCODE, 29\$     | :    | 1181 |
|           |    | 18        | AE | DD    | 0035E |        | PUSHL                          | STVCODE             | :    | 1182 |
|           |    | 10        | AE | DD    | 00361 |        | PUSHL                          | ERRORCODE           | :    |      |
|           |    |           | 57 | DD    | 00364 |        | PUSHL                          | FNM_DESC            | :    |      |
|           |    |           | 01 | DD    | 00366 |        | PUSHL                          | #1                  | :    |      |
|           |    | 00000000G | 8F | DD    | 00368 |        | PUSHL                          | #LINS_OPENIN        | :    |      |
| 00000000G | 00 |           | 05 | FB    | 0036E |        | CALLS                          | #5, LIB\$STOP       | :    |      |
| 04        | 00 |           | 04 | E0    | 00375 | 29\$:  | BBS                            | #4, @0(SP), 30\$    | :    | 1190 |
|           | BE |           | AA | E9    | 0037A |        | BLBC                           | 11(R10), 31\$       | :    |      |
|           | 0B | 0B        | 5A | DD    | 0037E | 30\$:  | PUSHL                          | R10                 | :    | 1191 |
| 00000000G | 00 |           | 01 | FB    | 00380 |        | CALLS                          | #1, LNK\$CLOSEFILE  | :    |      |
|           |    |           | 07 | 11    | 00387 |        | BRB                            | 32\$                | :    |      |
| 00000000' | EF |           | 5A | DD    | 00389 | 31\$:  | MOVL                           | R10, AN_OPEN_FDB    | :    | 1192 |
| 00000000' | EF | 00000000' | EF | 9E    | 00390 | 32\$:  | MOVAB                          | DEFILETYPE, TYPEADR | :    | 1194 |
|           |    |           | 04 | 0039B |       | RET    |                                | :                   | 1196 |      |

: Routine Size: 924 bytes, Routine Base: \$CODE\$ + 0541

: 1095 1197 1

```

1097 1198 1 global routine lnk$allofdb (fdbpointer) : novalue =
1098 1199 1
1099 1200 1 !+
1100 1201 1
1101 1202 1 This routine allocates some more dynmaic memory for file descriptor blocks
1102 1203 1 and RMS resultant strings and inserts into an auxiliary file name block the
1103 1204 1 length and address for resultant and expanded strings returned on opens on
1104 1205 1 any call.
1105 1206 1
1106 1207 1 If there is not sufficient resultant string area for the maximum, more is
1107 1208 1 allocated.
1108 1209 1
1109 1210 1 The routine checks rsl of the previously allocated auxiliary file name block
1110 1211 1 and if non zero, reduces the remaining string area by that actually used.
1111 1212 1 If still zero the area is reduced by the maximum size before the decision
1112 1213 1 to allocate more is made.
1113 1214 1
1114 1215 1 -
1115 1216 2 begin
1116 1217 2 own
1117 1218 2     prevfnb      : ref block [,byte] ;
1118 1219 2                                     ! Previous auxiliary filename block
1119 1220 2 local
1120 1221 2     fdblock      : ref block [,byte],
1121 1222 2     rsareduction,
1122 1223 2     temprsapointer ;
1123 1224 2                                     ! File descriptor block allocated
1124 1225 2                                     ! For allocating resultant
1125 1226 2 map
1126 1227 2     fdbpointer   : ref vector ;
1127 1228 2                                     ! String buffers
1128 1229 2                                     ! Passed an address at wich to store
1129 1230 2 if .prevfnb neq 0
1130 1231 2 then
1131 1232 2     begin
1132 1233 2     if (rsareduction = .prevfnb [nam$b_rsl]) eql 0
1133 1234 2     then rsareduction = .prevfnb [nam$b_rss] ;
1134 1235 2     rsaremsize = .rsaremsize - .rsareduction ;
1135 1236 2     nextrsa    = .nextrsa + .rsareduction ;
1136 1237 2     end ;
1137 1238 2 if .rsaremsize lssu nam$c_maxrss
1138 1239 2 then
1139 1240 2     begin
1140 1241 2     lnk$alloblk (rsa_allosize, temprsapointer) ;
1141 1242 2     if .temprsapointer eql .nextrsa + .rsaremsize
1142 1243 2     then
1143 1244 2         rsaremsize = .rsaremsize + rsa_allosize
1144 1245 2         ! Not enough space in resltant string
1145 1246 2         ! buffer for maximum one
1146 1247 2         ! Go allocate some more
1147 1248 2         ! If it was allocated continguously
1148 1249 2         ! with previous one
1149 1250 2         ! then merge
1150 1251 2     else
1151 1252 2         begin
1152 1253 2         rsaremsize = rsa_allosize ;
1153 1254 2         nextrsa    = .temprsapointer ;
1154 1255 2         end ;
1155 1256 2     end ;
1156 1257 2 lnk$alloblk (fdb$c_size+nam$c_bln, fdblock) ;
1157 1258 2 fdbpointer [0] = .fdblock ;
1158 1259 2 prevfnb      = fdblock [fdb$t_auxfnb] ;
1159 1260 2                                     ! Allocate file descriptor
1160 1261 2                                     ! Return its address
1161 1262 2                                     ! Get address of the file name block part

```

```

1154      1255      2 ch$fill (0,fdb$c_size+nam$c_bln, .fdblock) ;           ! Zero the fdb
1155      1256      2
1156      P 1257      2 $nam_init (nam = .prevfnb                          ! Initialize the nam block
1157      P 1258      2      ,rsa = .nextrsa
1158      P 1259      2      ,rss = nam$c_maxrss
1159      P 1260      2      ,esa = .nextrsa
1160      P 1261      2      ,ess = nam$c_maxrss
1161      1262      2      ) ;
1162      1263      2 return ;
1163      1264      2 end ;

```

.PSECT \$OWNS,NOEXE,2

001DB .BLKB 1  
001DC PREVFN: .BLKB 4

.PSECT \$CODE\$,NOWRT,2

|      |    |  |  |    |           |        |   |        |                              |                          |      |
|------|----|--|--|----|-----------|--------|---|--------|------------------------------|--------------------------|------|
|      |    |  |  |    | 01FC 0000 | .ENTRY | LNK\$ALLOFDB, Save R2,R3,R4,R5,R6,R7,R8 | 1198   |                              |                          |      |
|      |    |  |  | 58 | 00000000G | 00     | 9E 00002                                | MOVAB  | LNK\$ALLOBLK, R8             |                          |      |
|      |    |  |  | 57 | 00000000' | EF     | 9E 00009                                | MOVAB  | RSAREMSIZE, R7               |                          |      |
|      |    |  |  | 5E |           | 08     | C2 00010                                | SUBL2  | #8, SP                       |                          |      |
|      |    |  |  | 50 | 28        | A7     | D0 00013                                | MOVL   | PREVFN, R0                   | 1228                     |      |
|      |    |  |  |    |           | 11     | 13 00017                                | BEQL   | 2\$                          |                          |      |
|      |    |  |  | 51 | 03        | A0     | 9A 00019                                | MOVZBL | 3(R0), RSAREDUCTION          | 1231                     |      |
|      |    |  |  |    |           | 04     | 12 0001D                                | BNEQ   | 1\$                          |                          |      |
|      |    |  |  | 51 | 02        | A0     | 9A 0001F                                | MOVZBL | 2(R0), RSAREDUCTION          | 1232                     |      |
|      |    |  |  | 67 |           | 51     | 82 00023                                | 1\$:   | SUBB2                        | RSAREDUCTION, RSAREMSIZE | 1234 |
|      |    |  |  | FO |           | A7     | C0 00026                                | ADDL2  | RSAREDUCTION, NEXTRSA        | 1235                     |      |
|      |    |  |  | FF |           | 8F     | 91 0002A                                | 2\$:   | CMPB                         | RSAREMSIZE, #255         | 1238 |
|      |    |  |  |    |           | 24     | 1E 0002E                                | BGEQU  | 4\$                          |                          |      |
|      |    |  |  |    |           | 5E     | DD 00030                                | PUSHL  | SP                           | 1241                     |      |
|      |    |  |  | 7E | 017E      | 8F     | 3C 00032                                | MOVZWL | #382, -(SP)                  |                          |      |
|      |    |  |  | 68 |           | 02     | FB 00037                                | CALLS  | #2, LNK\$ALLOBLK             |                          |      |
|      |    |  |  | 50 |           | 67     | 9A 0003A                                | MOVZBL | RSAREMSIZE, R0               | 1242                     |      |
|      |    |  |  | 50 | FO        | A7     | C0 0003D                                | ADDL2  | NEXTRSA, R0                  |                          |      |
|      |    |  |  | 50 |           | 6E     | D1 00041                                | CMP    | TEMPRSAPOINTER, R0           |                          |      |
|      |    |  |  |    |           | 06     | 12 00044                                | BNEQ   | 3\$                          |                          |      |
|      |    |  |  | 67 | 7E        | 8F     | 80 00046                                | ADDB2  | #126, RSAREMSIZE             | 1244                     |      |
|      |    |  |  |    |           | 08     | 11 0004A                                | BRB    | 4\$                          |                          |      |
|      |    |  |  | 67 | 7E        | 8F     | 90 0004C                                | 3\$:   | MOVB                         | #126, RSAREMSIZE         | 1247 |
|      |    |  |  | FO |           | A7     | D0 00050                                | MOVL   | TEMPRSAPOINTER, NEXTRSA      | 1248                     |      |
|      |    |  |  |    |           | 04     | AE 9F 00054                             | 4\$:   | PUSHAB                       | FDBLOCK                  | 1252 |
|      |    |  |  | 7E | 86        | 8F     | 9A 00057                                | MOVZBL | #134, -(SP)                  |                          |      |
|      |    |  |  | 68 |           | 02     | FB 0005B                                | CALLS  | #2, LNK\$ALLOBLK             |                          |      |
|      |    |  |  | 04 | BC        | 04     | AE D0 0005E                             | MOVL   | FDBLOCK, @FDBPOINTER         | 1253                     |      |
|      |    |  |  | 04 | AE        | 04     | 26 C1 00063                             | ADDL3  | #38, FDBLOCK, PREVFN         | 1254                     |      |
| 0086 | 8F |  |  |    |           | 00     | 6E 00 2C 00069                          | MOVCS  | #0, (SP), #0, #134, @FDBLOCK | 1255                     |      |
|      |    |  |  |    |           | 04     | BE 00070                                |        |                              |                          |      |
|      |    |  |  | 56 | 28        | A7     | D0 00072                                | MOVL   | PREVFN, R6                   | 1262                     |      |
| 0060 | 8F |  |  |    |           | 00     | 6E 00 2C 00076                          | MOVCS  | #0, (SP), #0, #96, (R6)      |                          |      |
|      |    |  |  |    |           | 66     | 66 0007D                                |        |                              |                          |      |
|      |    |  |  | 66 | 6002      | 8F     | B0 0007E                                | MOVW   | #24578, (R6)                 |                          |      |



LNK\_GETCMDINP  
V04=000

F 8  
15-Sep-1984 23:57:45  
14-Sep-1984 12:40:30

VAX-11 Bliss-32 V4.0-742  
[LINKER.SRC]LNKGETCMD.B32;1

Page 47  
(14)

|    |    |    |    |       |       |       |                 |
|----|----|----|----|-------|-------|-------|-----------------|
| 02 | A6 |    | 01 | 8E    | 00083 | MNEGB | #1, 2(R6)       |
| 04 | A6 | FO | A7 | D0    | 00087 | MOVL  | NEXTRSA, 4(R6)  |
| 0A | A6 |    | 01 | 8E    | 0008C | MNEGB | #1, 10(R6)      |
| 0C | A6 | FO | A7 | D0    | 00090 | MOVL  | NEXTRSA, 12(R6) |
|    |    |    | 04 | 00095 |       | RET   |                 |

:  
:  
:  
:  
: 1264

. Routine Size: 150 bytes, Routine Base: \$CODE\$ + 08DD

; 1164 126<sup>c</sup> 1

```

: 1166      1266 1 global routine lnk$clunamcmp (keyname, node) =
: 1167      1267 1
: 1168      1268 1
: 1169      1269 1
: 1170      1270 1 This routine compares an ascic keyname with the current entry
: 1171      1271 1
: 1172      1272 1
: 1173      1273 2 begin
: 1174      1274 2 map
: 1175      1275 2     keyname : ref vector [,byte],
: 1176      1276 2     node   : ref block [,byte] ;
: 1177      1277 2 local
: 1178      1278 2     clu    : ref block [,byte] ;
: 1179      1279 2
: 1180      1280 2 clu = .node [node$l_ptr] ;           ! Point to cluster descriptor
: 1181      1281 2 return ch$compare (.keyname [0], keyname [1]
: 1182      1282 2     ;.clu [clu$b_naming], clu [clu$t_name]
: 1183      1283 2     )
: 1184      1284 1 end ;

```

|    |    |    |    |             |        |                               |        |
|----|----|----|----|-------------|--------|-------------------------------|--------|
|    |    |    |    | 001C 00000  | .ENTRY | LNK\$CLUNAMCMP, Save R2,R3,R4 | : 1266 |
|    | 50 | 08 | AC | D0 00002    | MOVL   | NODE, R0                      | : 1280 |
|    | 50 | 0A | A0 | D0 00006    | MOVL   | 10(R0), CLU                   |        |
|    | 52 | 04 | AC | D0 0000A    | MOVL   | KEYNAME, R2                   | : 1281 |
|    | 53 |    | 62 | 9A 0000E    | MOVZBL | (R2), R3                      |        |
|    | 51 | 5C | A0 | 9A 00011    | MOVZBL | 92(CLU), R1                   | : 1282 |
|    | 54 |    | 01 | D0 00015    | MOVL   | #1, R4                        |        |
| 51 | 00 | 01 | A2 | 53 2D 00018 | CMPC5  | R3, 1(R2), #0, R1, 93(CLU)    |        |
|    |    |    | 5D | A0 0001E    |        |                               |        |
|    | 54 |    | 03 | 1A 00020    | BGTRU  | 1\$                           |        |
|    | 50 |    | 01 | D9 00022    | SBWC   | #1, R4                        |        |
|    |    |    | 54 | D0 00025    | MOVL   | R4, R0                        |        |
|    |    |    | 04 | 00028       | RET    |                               | : 1284 |

: Routine Size: 41 bytes, Routine Base: \$CODE\$ + 0973

: 1185 1285 1

```

: 1187      1286 1 global routine lnk$compare_pdd (keyname,pdd) =
: 1188      1287 1
: 1189      1288 1 | +
: 1190      1289 1 |
: 1191      1290 1 | Routine to compare ascii psect name with current entry
: 1192      1291 1 |
: 1193      1292 1 | -
: 1194      1293 2 begin
: 1195      1294 2 map
: 1196      1295 2     keyname : ref vector [ ,byte],
: 1197      1296 2     pdd      : ref block [ ,byte] ;
: 1198      1297 2
: 1199      1298 2 return ch$compare (.keyname [0], keyname [1]
: 1200      1299 2     ; (.pdd [pdd$b_nam[ng] and not %x'80'), pdd [pdd$t_name]
: 1201      1300 2     )
: 1202      1301 1 end ;

```

|    |    |    |    |    |    |            |        |                                 |        |
|----|----|----|----|----|----|------------|--------|---------------------------------|--------|
|    |    |    |    |    |    | 001C 00000 | .ENTRY | LNK\$COMPARE_PDD, Save R2,R3,R4 | : 1286 |
|    |    |    | 52 | 04 | AC | D0 00002   | MOVL   | KEYNAME, R2                     | : 1298 |
|    |    |    | 53 |    | 62 | 9A 00006   | MOVZBL | (R2), R3                        |        |
|    |    |    | 50 | 08 | AC | D0 00009   | MOVL   | PDD, R0                         | : 1299 |
| 51 | OF | A0 | 07 |    | 00 | EF 0000D   | EXTZV  | #0, #7, 15(R0), R1              |        |
|    |    |    | 54 |    | 01 | D0 00013   | MOVL   | #1, R4                          |        |
| 51 |    | 00 | A2 |    | 53 | 2D 00016   | CMPCS  | R3, 1(R2), #0, R1, 16(R0)       |        |
|    |    |    |    | 10 | A0 | 0001C      |        |                                 |        |
|    |    |    | 54 |    | 03 | 1A 0001E   | BGTRU  | 1\$                             |        |
|    |    |    | 50 |    | 01 | D9 00020   | SBWC   | #1, R4                          |        |
|    |    |    |    |    | 54 | D0 00023   | MOVL   | R4, R0                          |        |
|    |    |    |    |    | 04 | 00026      | RET    |                                 | : 1301 |

; Routine Size: 39 bytes, Routine Base: \$CODE\$ + 099C

: 1203 1302 1

```

: 1205      1303 1 global routine lnk$alloc_pdd (keyname,retadr) =
: 1206      1304 1
: 1207      1305 1 !+
: 1208      1306 1 !:-
: 1209      1307 1 Allocate a new pdd block
: 1210      1308 1 !:-
: 1211      1309 1 !:-
: 1212      1310 2 begin
: 1213      1311 2 map
: 1214      1312 2     keyname : ref vector [ ,byte],
: 1215      1313 2     retadr  : ref vector [ ,long] ;
: 1216      1314 2 local
: 1217      1315 2     pdd      : ref block [ ,byte] ;
: 1218      1316 2
: 1219      1317 2 lnk$alloblk (pdd$c_size+.keyname [0], pdd) ;           ! Allocate the pdd
: 1220      1318 2
: 1221      1319 2 ch$fill (0, pdd$c_size, .pdd) ;                       ! Zero it
: 1222      1320 2
: 1223      1321 2 pdd [pdd$b_flags] = psc$m_optpsc ;                             ! Set defined by option flag
: 1224      1322 2 pdd [pdd$b_align] = -1 ;                                       ! Set no alignment specified
: 1225      1323 2
: 1226      1324 2 ch$move (.keyname [0]+1, keyname [0], pdd [pdd$b_namlng]) ; ! Insert the name
: 1227      1325 2
: 1228      1326 2 retadr [0] = .pdd ;
: 1229      1327 2 return true
: 1230      1328 1 end ;

```

|    |    |           |    |                  |        |                                     |        |
|----|----|-----------|----|------------------|--------|-------------------------------------|--------|
|    |    |           |    | 007C 00000       | .ENTRY | LNK\$ALLOC_PDD, Save R2,R3,R4,R5,R6 | : 1303 |
|    |    | 5E        |    | 04 C2 00002      | SUBL2  | #4, SP                              |        |
|    |    |           |    | 5E DD 00005      | PUSHL  | SP                                  | : 1317 |
|    |    | 7E        | 04 | BC 9A 00007      | MOVZBL | @KEYNAME, -(SP)                     |        |
|    |    | 6E        |    | 10 C0 0000B      | ADDL2  | #16, (SP)                           |        |
|    |    | 00000000G | 00 | 02 FB 0000E      | CALLS  | #2, LNK\$ALLOBLK                    |        |
|    |    |           | 56 | 6E D0 00015      | MOVL   | PDD, R6                             | : 1319 |
| 10 | 00 |           | 6E | 00 2C 00018      | MOVCS  | #0, (SP), #0, #16, (R6)             |        |
|    |    |           |    | 66 0001D         |        |                                     |        |
|    |    | 0A        | A6 | 0400 8F B0 0001E | MOVW   | #1024, 10(R6)                       | : 1321 |
|    |    | 0E        | A6 | 01 8E 00024      | MNEGB  | #1, 14(R6)                          | : 1322 |
|    |    |           | 50 | 04 BC 9A 00028   | MOVZBL | @KEYNAME, R0                        | : 1324 |
|    |    |           |    | 50 D6 0002C      | INCL   | R0                                  |        |
|    | OF | A6        | 04 | 50 28 0002E      | MOVCS  | R0, @KEYNAME, 15(R6)                |        |
|    |    |           | 08 | BC 56 D0 00034   | MOVL   | R6, @RETADR                         | : 1326 |
|    |    |           | 50 | 01 D0 00038      | MOVL   | #1, R0                              | : 1327 |
|    |    |           |    | 04 0003B         | RET    |                                     | : 1328 |

: Routine Size: 60 bytes, Routine Base: \$CODE\$ + 09C3

: 1231 1329 1

```
1233 1330 1 global routine lnk$collectpsct : novalue =
1234 1331 1
1235 1332 1 +
1236 1333 1
1237 1334 1 This routine is called at the end of command processing to process the data
1238 1335 1 base set up by the PSECT definition and collection options. A pass is made
1239 1336 1 over the list of PSECTS hanging off each cluster mentioned in a collect
1240 1337 1 option. For each PSECT in that list, define it.
1241 1338 1
1242 1339 1 -
1243 1340 2 begin
1244 1341 2 local
1245 1342 2 curcluster : ref block [ ,byte], ! Pointer to current cluster descriptor
1246 1343 2 lstcluster : ref block [ ,byte], ! Pointer to last cluster descriptor
1247 1344 2 curpsct : ref block [ ,byte], ! Pointer to current psect name block
1248 1345 2 cludesc : ref block [ ,byte], ! Pointer to cluster descriptor
1249 1346 2 psctdesc : ref block [ ,byte], ! Pointer to psect descriptor
1250 1347 2 psctdef : ref block [ ,byte] ; ! Pointer to psect definition block
1251 1348 2
1252 1349 2 curcluster = .lnk$gl_cclulst ; ! Point to first cluster name block
1253 1350 2
1254 1351 2 loop through all named clusters
1255 1352 2
1256 1353 2 while .curcluster neq 0 do
1257 1354 3 begin
1258 1355 3 if not lib$lookup_tree (lnk$gl_clutree ! Lookup up cluster
1259 1356 3 .curcluster [ccd$b_namng]
1260 1357 3 .lnk$clunamcmp
1261 1358 3 ; cludesc
1262 1359 3 )
1263 1360 3 then
1264 1361 4 begin ! And if not defined yet
1265 1362 4 lnk$a'allocluster (cludesc) ; ! Then do it now
1266 1363 4 ch$move (.curcluster [ccd$b_namng]+1 ! Copy in the name
1267 1364 4 , curcluster [ccd$b_namng]
1268 1365 4 ; cludesc [clu$b_namng]
1269 1366 4 ) ;
1270 1367 4 lnk$insert_clu (.cludesc) ; ! Insert into cluster tree
1271 1368 4 if .curcluster [ccd$b_protect] ! If the cluster should be protected
1272 1369 4 then
1273 1370 4 cludesc [clu$v_protect] = true ; ! Then flag it now
1274 1371 4 lnk$gl_curclu = .cludesc ; ! Make it the current cluster
1275 1372 4 end
1276 1373 3 else
1277 1374 3 lnk$gl_curclu = .cludesc [node$l_ptr] ; ! Found--make it the current cluster
1278 1375 3
1279 1376 3 curpsct = .curcluster [ccd$l_psclst] ; ! Point to first psect block
1280 1377 3 lstcluster = .curcluster ; ! Remember this one
1281 1378 3 curcluster = .curcluster [ccd$l_nxtclu] ; ! And point to next one
1282 1379 3 lnk$dealblk (ccd$c_size, .lstcluster) ; ! Deallocate it
1283 1380 3
1284 1381 3 Now get all the psects collected into that cluster
1285 1382 3
1286 1383 3 while .curpsct neq 0
1287 1384 4 do begin
1288 1385 4 local defbyopt ;
1289 1386 4
```

```

: 1290 1387 5 if not (defbyopt = lnk$srcpsctdef (curpsct [cpd$b_namlng], psctdef))
: 1291 1388 4 then
: 1292 1389 4 lib$insert_tree (lnk$gl_pscdflst ! Insert into defined by option list
: 1293 1390 4 ,curpsct [cpd$b_namlng]
: 1294 1391 4 ,%ref (0)
: 1295 1392 4 ,lnk$compare_pdd
: 1296 1393 4 ,lnk$alloc_pdd,psctdef
: 1297 1394 4 )
: 1298 1395 4 psctdef [pdd$w_flags] = .psctdef [pdd$w_flags] or gps$m_gbl ; ! Mark psect as global
: 1299 1396 4 lnk$findpsctnam (curpsct [cpd$b_namlng] ! Enter psect in current cluster
: 1300 1397 4 ,.psctdef [pdd$w_flags]
: 1301 1398 4 ,0
: 1302 1399 4 , psctdesc
: 1303 1400 4 ) ;
: 1304 1401 4 if .defbyopt ! If defined by option
: 1305 1402 4 then ! then
: 1306 1403 5 begin
: 1307 1404 5 psctdef [pdd$b_namlng] = .psctdef [pdd$b_namlng] or %x'80' ; ! flag psect already defined
: 1308 1405 5 psctdesc [psc$w_flags] = .psctdef [pdd$w_flags] ; ! copy the defined flags
: 1309 1406 5 psctdesc [psc$b_align] = .psctdef [pdd$b_align] ; ! and the alignment
: 1310 1407 5
: 1311 1408 5 if .psctdef [pdd$b_align] eql %x'FF' ! if defaulted
: 1312 1409 5 then psctdesc [psc$b_align] = 0 ; ! then use byte alignment
: 1313 1410 4 end ;
: 1314 1411 4
: 1315 1412 4 curpsct = .curpsct [cpd$l_nxtpsc] ; ! Point to next name block
: 1316 1413 4 end ! Curpsct neq 0
: 1317 1414 2 end ; ! Curcluter neq 0
: 1318 1415 2
: 1319 1416 2 return ;
: 1320 1417 1 end ;

```

|    |           |           |    |                |        |  |      |
|----|-----------|-----------|----|----------------|--------|--|------|
|    |           |           |    | 0FFC 0000      | .ENTRY | LNK\$COLLECTPSCT, Save R2,R3,R4,R5,R6,R7,R8,-; | 1330 |
|    |           |           |    |                |        | R9,R10,R11                                     |      |
|    | 5B        | 00000000G | 00 | 9E 00002       | MOVAB  | LNK\$GL_CURCLU, R11                            |      |
|    | 5A        | FF67      | CF | 9E 00009       | MOVAB  | LNK\$CLONAMCMP, R10                            |      |
|    | 5E        |           | 10 | C2 0000E       | SUBL2  | #16, SP  |      |
|    | 57        | 00000000G | 00 | D0 00011       | MOVL   | LNK\$GL_CCLULST, CURCLUSTER                    | 1349 |
|    |           |           | 57 | D5 00018       | TSTL   | CURCLUSTER                                     | 1353 |
|    |           |           | 01 | 12 0001A       | BNEQ   | 2\$  |      |
|    |           |           |    | 04 0001C       | RET    |  |      |
|    |           |           | 04 | AE 9F 0001D    | PUSHAB | CLUDESC  | 1356 |
|    |           |           |    | 5A DD 00020    | PUSHL  | R10  |      |
|    |           |           | 04 | A7 9F 00022    | PUSHAB | 4(CURCLUSTER)                                  |      |
|    |           |           |    | 00 9F 00025    | PUSHAB | LNK\$GL_CLUTREE                                | 1355 |
|    | 00000000G |           | 00 | 04 FB 0002B    | CALLS  | #4, LIB\$LOOKUP_TREE                           | 1356 |
|    |           |           | 31 | 50 E8 00032    | BLBS   | R0, 4\$  |      |
|    |           |           | 04 | AE 9F 00035    | PUSHAB | CLUDESC  | 1362 |
|    | 00000000V |           | EF | 01 FB 00038    | CALLS  | #1, LNK\$ALLOCLUSTER                           |      |
|    |           |           | 50 | 04 A7 9A 0003F | MOVZBL | 4(CURCLUSTER), R0                              | 1363 |
|    |           |           |    | 50 D6 00043    | INCL   | R0   |      |
|    |           |           | 56 | 04 AE D0 00045 | MOVL   | CLUDESC, R6                                    | 1365 |
| 5C | A6        | 04        | A7 | 50 28 00049    | MOV3   | R0, 4(CURCLUSTER), 92(R6)                      |      |

|           |    |           |    |       |       |        |                          |                         |      |      |
|-----------|----|-----------|----|-------|-------|--------|--------------------------|-------------------------|------|------|
| 00000000V | EF |           | 56 | DD    | 0004F | PUSHL  | R6                       | :                       | 1367 |      |
|           | 05 | 30        | 01 | FB    | 00051 | CALLS  | #1, LNK\$INSERT_CLU      | :                       |      |      |
| 58        | A6 | 80        | A7 | E9    | 00058 | BLBC   | 48(CURCLUSTER), 3\$      | :                       | 1368 |      |
|           | 68 |           | 8F | 88    | 0005C | BISB2  | #128, 88(R6)             | :                       | 1370 |      |
|           |    |           | 56 | DO    | 00061 | 3\$:   | MOVL                     | R6, LNK\$GL_CURCLU      | 1371 |      |
|           |    |           | 08 | 11    | 00064 | BRB    | 5\$                      | :                       | 1355 |      |
|           | 50 | 04        | AE | DO    | 00066 | 4\$:   | MOVL                     | CLUDESC, R0             | 1374 |      |
|           | 6B | 0A        | A0 | DO    | 0006A | MOVL   | 10(R0), LNK\$GL_CURCLU   | :                       |      |      |
|           | 58 | 2C        | A7 | DO    | 0006E | 5\$:   | MOVL                     | 44(CURCLUSTER), CURPSCT | 1376 |      |
|           | 59 |           | 57 | DO    | 00072 | MOVL   | CURCLUSTER, LSTCLUSTER   | :                       | 1377 |      |
|           | 57 |           | 67 | DO    | 00075 | MOVL   | (CURCLUSTER), CURCLUSTER | :                       | 1378 |      |
|           |    |           | 59 | DD    | 00078 | PUSHL  | LSTCLUSTER               | :                       | 1379 |      |
| 00000000G | 00 |           | 31 | DD    | 0007A | PUSHL  | #49                      | :                       |      |      |
|           |    |           | 02 | FB    | 0007C | CALLS  | #2, LNK\$DEALBLK         | :                       |      |      |
|           |    |           | 58 | D5    | 00083 | 6\$:   | TSTL                     | CURPSCT                 | 1383 |      |
|           |    |           | 91 | 13    | 00085 | BEQL   | 1\$                      | :                       |      |      |
|           |    | 08        | AE | 9F    | 00087 | PUSHAB | PSCTDEF                  | :                       | 1387 |      |
|           |    | 04        | A8 | 9F    | 0008A | PUSHAB | 4(CURPSCT)               | :                       |      |      |
| 00000000V | EF |           | 02 | FB    | 0008D | CALLS  | #2, LNK\$SRCPCDEF        | :                       |      |      |
|           | 53 |           | 50 | DO    | 00094 | MOVL   | R0, DEFBYOPT             | :                       |      |      |
|           | 1F |           | 53 | E8    | 00097 | BLBS   | DEFBYOPT, 7\$            | :                       |      |      |
|           |    | 08        | AE | 9F    | 0009A | PUSHAB | PSCTDEF                  | :                       | 1390 |      |
|           |    | 50        | AA | 9F    | 0009D | PUSHAB | LNK\$ALLOC_PDD           | :                       |      |      |
|           |    | 29        | AA | 9F    | 000A0 | PUSHAB | LNK\$COMPARE_PDD         | :                       |      |      |
|           |    | 0C        | AE | D4    | 000A3 | CLRL   | 12(SP)                   | :                       | 1391 |      |
|           |    | 0C        | AE | 9F    | 000A6 | PUSHAB | 12(SP)                   | :                       |      |      |
|           |    | 04        | A8 | 9F    | 000A9 | PUSHAB | 4(CURPSCT)               | :                       | 1390 |      |
| 00000000G | 00 | 00000000G | 00 | 9F    | 000AC | PUSHAB | LNK\$GL_PSCDFLST         | :                       | 1389 |      |
|           | 52 |           | 06 | FB    | 000B2 | CALLS  | #6, LIB\$INSERT_TREE     | :                       | 1390 |      |
| 0A        | A2 |           | 08 | AE    | DO    | 7\$:   | MOVL                     | PSCTDEF, R2             | 1395 |      |
|           |    |           | 10 | 88    | 000BD | BISB2  | #16, 10(R2)              | :                       |      |      |
|           |    |           | 0C | AE    | 9F    | 000C1  | PUSHAB                   | PSCTDESC                | :    | 1396 |
|           |    |           | 7E | D4    | 000C4 | CLRL   | -(SP)                    | :                       |      |      |
|           | 7E | 0A        | A2 | 3C    | 000C6 | MOVZWL | 10(R2), -(SP)            | :                       | 1397 |      |
|           |    | 04        | A8 | 9F    | 000CA | PUSHAB | 4(CURPSCT)               | :                       | 1396 |      |
| 00000000G | 00 |           | 04 | FB    | 000CD | CALLS  | #4, LNK\$FNDPSCNAM       | :                       |      |      |
|           | 1D |           | 53 | E9    | 000D4 | BLBC   | DEFBYOPT, 8\$            | :                       | 1401 |      |
|           | 0F | 80        | 8F | 88    | 000D7 | BISB2  | #128, 15(R2)             | :                       | 1404 |      |
|           | 50 | 0C        | AE | DO    | 000DC | MOVL   | PSCTDESC, R0             | :                       | 1405 |      |
|           | 0A | 0A        | A2 | B0    | 000E0 | MOVW   | 10(R2), 10(R0)           | :                       |      |      |
|           | 2C | 0E        | A2 | 90    | 000E5 | MOVW   | 14(R2), 44(R0)           | :                       | 1406 |      |
|           | FF | 0E        | A2 | 91    | 000EA | CMPB   | 14(R2), #255             | :                       | 1408 |      |
|           |    |           | 03 | 12    | 000EF | BNEQ   | 8\$                      | :                       |      |      |
|           |    | 2C        | A0 | 94    | 000F1 | CLRB   | 44(R0)                   | :                       | 1409 |      |
|           | 58 |           | 68 | DO    | 000F4 | 8\$:   | MOVL                     | (CURPSCT), CURPSCT      | 1412 |      |
|           |    |           | 8A | 11    | 000F7 | BRB    | 6\$                      | :                       | 1383 |      |
|           |    |           | 04 | 000F9 | RET   |        |                          | :                       | 1417 |      |

: Routine Size: 250 bytes, Routine Base: \$CODE\$ + 09FF

: 1321 1418 1

```

: 1323      1419 1 global routine lnk$srcpscdef (pscnam,psctdesc) =
: 1324      1420 1
: 1325      1421 1 !+
: 1326      1422 1 !:
: 1327      1423 1 !: Look up PSECT name in the psechts defined by option.
: 1328      1424 1 !:
: 1329      1425 1 !:-
: 1330      1426 2 begin
: 1331      1427 2 map
: 1332      1428 2      psctdesc      : ref block [ ,byte] ;
: 1333      1429 2      psctdesc      : ref block [ ,byte] ;
: 1334      1430 2
: 1335      1431 2 return lib$lookup_tree (lnk$gl_pscdfilst, .pscnam, lnk$compare_pdd, .psctdesc)
: 1336      1432 2
: 1337      1433 1 end ;

```

```

                                0000 0000      .ENTRY LNK$SRCPCDEF, Save nothing      : 1419
                                08 AC DD 00002    PUSHL PSCTDESC                          : 1431
                                FE9A CF 9F 00005    PUSHAB LNK$COMPARE_PDD
                                04 AC DD 00009    PUSHL PSCNAM
                                00000000G 00 00 9F 00C0C    PUSHAB LNK$GL_PSCDFLST
                                04 FB 00012    CALLS #4, LIB$LOOKUP_TREE
                                04 00019    RET

```

: Routine Size: 26 bytes, Routine Base: \$CODE\$ + 0AF9

: 1338 1434 1



```

: 1340      1435 1 global routine lnk$allocluster (clusterdesc,dummyarg) : novalue =
: 1341      1436 1
: 1342      1437 1
: 1343      1438 1
: 1344      1439 1 Allocate a cluster descriptor and link on end of the list and initialize
: 1345      1440 1 some fields of it now.
: 1346      1441 1
: 1347      1442 1 If the second argument is present, the cluster descriptor is not linked
: 1348      1443 1 into the list.
: 1349      1444 1
: 1350      1445 1
: 1351      1446 2 begin
: 1352      1447 2 map
: 1353      1448 2     clusterdesc      : ref vector [,long] ;
: 1354      1449 2 local
: 1355      1450 2     newcluster       : ref block [,byte] ;
: 1356      1451 2 builtin
: 1357      1452 2     nullparameter ;
: 1358      1453 2
: 1359      1454 2 lnk$alloblk (clu$c_size,clusterdesc [0]) ;           ! Allocate the block
: 1360      1455 2 newcluster = .clusterdesc [0] ;                 ! Retrieve its address
: 1361      1456 2 ch$fill (0,clu$c_size,.newcluster) ;           ! Zero the descriptor
: 1362      1457 2
: 1363      1458 2 newcluster [clu$l_lstfdb] = newcluster [clu$l_fstfdb] ; ! So empty list
: 1364      1459 2 newcluster [clu$l_lstisd] = newcluster [clu$l_fstisd] ; !
: 1365      1460 2 newcluster [clu$l_lstadrl] = newcluster [clu$l_fstadrl] ; !
: 1366      1461 2 lnk$gw_nclusters = .lnk$gw_nclusters + 1 ; ! Count the cluster
: 1367      1462 2
: 1368      1463 2 if nullparameter (2)                               ! If second argument not supplied
: 1369      1464 2 then
: 1370      1465 3     begin
: 1371      1466 3     lnk$gl_lastclu [clu$l_nxtclu] = .newcluster ; ! then link into end of cluster list
: 1372      1467 3     newcluster [clu$l_prevclu] = .lnk$gl_lastclu ; ! Set pointer
: 1373      1468 3     lnk$gl_lastclu = .newcluster ;           ! For last in list
: 1374      1469 3     end ;
: 1375      1470 2 return ;
: 1376      1471 1 end ;

```

|      |    |                      |                |            |        |  |        |
|------|----|----------------------|----------------|------------|--------|--|--------|
|      |    |                      |                | 00FC 00000 | .ENTRY | LNK\$ALLOCLUSTER, Save R2,R3,R4,R5,R6,R7 | : 1435 |
|      |    | 57 00000000G         | 00 9E 00002    |            | MOVAB  | LNK\$GL_LASTCLU, R7                      |        |
|      |    |                      | 04 AC DD 00009 |            | PUSHL  | CLUSTERDESC                              | : 1454 |
|      |    | 7E 88 8F 9A 0000C    |                |            | MOVZBL | #136, -(SP)                              |        |
|      |    | 00000000G            | 00 02 FB 00010 |            | CALLS  | #2, LNK\$ALLOBLK                         |        |
|      |    | 56 04 BC D0 00017    |                |            | MOVL   | @CLUSTERDESC, NEWCLUSTER                 | : 1455 |
| 0088 | 8F | 00                   | 6E 00 2C 0001B |            | MOVCS  | #0, (SP), #0, #136, (NEWCLUSTER)         | : 1456 |
|      |    |                      | 66 00022       |            |        |  |        |
|      |    | 0C A6 08 A6 9E 00023 |                |            | MOVAB  | 8(R6), 12(NEWCLUSTER)                    | : 1458 |
|      |    | 1C A6 18 A6 9E 00028 |                |            | MOVAB  | 24(R6), 28(NEWCLUSTER)                   | : 1459 |
|      |    | 40 A6 3C A6 9E 0002D |                |            | MOVAB  | 60(R6), 64(NEWCLUSTER)                   | : 1460 |
|      |    |                      | 00 B6 00032    |            | INCW   | LNK\$GW_NCLUSTERS                        | : 1461 |
|      |    | 02 00000000G         | 00 B6 00032    |            | CMPB   | (AP), #2                                 | : 1463 |
|      |    |                      | 6C 91 00038    |            | BLSSU  | 1\$                                      |        |
|      |    |                      | 05 1F 0003B    |            | TSTL   | 8(AP)                                    |        |
|      |    |                      | 08 AC D5 0003D |            |        |  |        |



```

: 1379      1473 1 global routine lnk$insert_clu (cludesc) =
: 1380      1474 1
: 1381      1475 1 | +
: 1382      1476 1 |
: 1383      1477 1 | This routine is called to insert a cluster descriptor into the balanced
: 1384      1478 1 | binary tree of cluster descriptors.
: 1385      1479 1 |
: 1386      1480 1 | Inputs:
: 1387      1481 1 |
: 1388      1482 1 |     CLUDESC          address of cluster descriptor
: 1389      1483 1 |
: 1390      1484 1 |
: 1391      1485 1 | -
: 1392      1486 2 begin
: 1393      1487 2 routine clunodealloc (keyname,retadr,cludesc) =
: 1394      1488 2 begin
: 1395      1489 2 |
: 1396      1490 2 | routine to allocate node for new cluster
: 1397      1491 2 |
: 1398      1492 2 map
: 1399      1493 2     keyname          : ref vector [,byte],
: 1400      1494 2     retadr           : ref vector [,long],
: 1401      1495 2     cludesc          : ref block [,byte] ;
: 1402      1496 2 local
: 1403      1497 2     nodeptr          : ref block [,byte] ;
: 1404      1498 2
: 1405      1499 2 lnk$alloblk (node$c_long+.keyname [0],nodeptr) ;
: 1406      1500 2 nodeptr [node$l_ptr] = .cludesc ;
: 1407      1501 2 retadr [0] = .nodeptr ;
: 1408      1502 2
: 1409      1503 2 return true
: 1410      1504 2 end ;

```

|           |       |    |             |               |                  |        |
|-----------|-------|----|-------------|---------------|------------------|--------|
|           |       |    | 0000 0000   | CLUNODEALLOC: |                  |        |
|           | 5E    |    | 04 C2 00002 | .WORD         | Save nothing     | : 1487 |
|           |       |    | 5E DD 00005 | SUBL2         | #4, SP           |        |
|           | 7E    | 04 | BC 9A 00007 | PUSHL         | SP               | : 1499 |
|           | 6E    |    | 0E C0 0000B | MOVZBL        | @KEYNAME, -(SP)  |        |
| 00000000G | 00    |    | 02 FB 0000E | ADDL2         | #14, (SP)        |        |
|           | 50    |    | 6E D0 00015 | CALLS         | #2, LNK\$ALLOBLK |        |
|           | 0A A0 | 0C | AC D0 00018 | MOVL          | NODEPTR, R0      | : 1500 |
|           | 08 BC |    | 50 D0 0001D | MOVL          | CLUDESC, 10(R0)  | : 1501 |
|           | 50    |    | 01 D0 00021 | MOVL          | R0, @RETADR      | : 1503 |
|           |       |    | 04 00024    | RET           | #1, R0           | : 1504 |

: Routine Size: 37 bytes, Routine Base: \$CODE\$ + 0B63

```

: 1411      1505 2 |
: 1412      1506 2 | Main body of lnk$insert_clu
: 1413      1507 2 |
: 1414      1508 2 map

```



```

1424 1517 1 routine get_image : novalue =
1425 1518 1
1426 1519 1 !++
1427 1520 1
1428 1521 1 This routine processes the /EXE switch and determines the name and
1429 1522 1 the associated related name of the .EXE file.
1430 1523 1
1431 1524 1 !--
1432 1525 2 begin
1433 1526 2 local
1434 1527 2     exe_desc      : dynamic_descriptor,      ! Descriptor of .EXE file
1435 1528 2     new_file      : initial (0),             ! Set if new .EXE request is made
1436 1529 2     status        :
1437 1530 2 own
1438 1531 2     exe_name      : vector [nam% maxrss, byte], ! Related file for .EXE
1439 1532 2     exe_inrel     : block [nam%_bln, byte] ;    ! Related NAMS for .EXE
1440 1533 2
1441 1534 3 if (status = cli$present (sd_executable)) ! /EXECUTABLE
1442 1535 3 then begin
1443 1536 3     if .local_qualifier
1444 1537 4     then begin
1445 1538 4         if .status neq cli$_locpres
1446 1539 4         then return
1447 1540 5         else begin
1448 1541 5             setup_inrelnam (0, .last_fnm_desc, exe_inrel, exe_name) ; ! Setup related name
1449 1542 5             lnk$gl_relnam_exe = exe_inrel ; ! using last input file
1450 1543 5             end
1451 1544 4         end
1452 1545 3     else lnk$gl_relnam_exe = dflt_inrel ; ! Use default related name
1453 1546 3
1454 1547 3     new_file      = true ; ! Setup new .EXE file spec
1455 1548 3     univstar_valid = false ; ! Disallow UNIV=* for STB generation
1456 1549 4     lnk$gb_locnov_exe = (if cli$get_value (sd_executable, exe_desc) ! Get image name (no name is okay)
1457 1550 4         then false ! and setup 'local no-value' flag
1458 1551 4         else .local_qualifier
1459 1552 3     ) ;
1460 1553 3     if .lnk$gl_ctlmsk [lnk$sv_shr] ! If a shareable image
1461 1554 3     then if .exe_desc [dsc$w_length] eql 0 ! and /EXE didn't specify a name
1462 1555 3     then copy_desc (share_desc, exe_desc) ; ! then use /SHAREABLE name
1463 1556 3     end
1464 1557 3 else begin
1465 1558 4     if ((not .local_qualifier) and (.status eql cli$_negated)) ! But, if explicitly negated,
1466 1559 4     or (( .local_qualifier) and (.status eql cli$_locneg)) !
1467 1560 4     then begin ! disable image production
1468 1561 4         new_file      = true ; ! Setup new .EXE file spec
1469 1562 4         univstar_valid = true ; ! Will now allow UNIV=* for STB gene
1470 1563 4         copy_desc (null_device, exe_desc) ;
1471 1564 3         end ;
1472 1565 2     end ;
1473 1566 2
1474 1567 2 if .new_file
1475 1568 2 then begin
1476 1569 3     outfile = lnk$gl_imgfil ; ! Point to the pointer
1477 1570 3     outputfile (exe_desc) ; ! Set up image output file
1478 1571 2     end ;
1479 1572 1 end ;

```

.PSECT \$OWNS,NOEXE,2

001E0 EXE\_NAME:  
          .BLKB 255  
002DF       .BLKB 1  
002E0 EXE\_INREL:  
          .BLKB 96

.PSECT \$CODE\$,NOWRT,2

|              |    |            |                  |        |                                |  |  |  |      |
|--------------|----|------------|------------------|--------|--------------------------------|--|--|--|------|
|              |    | 003C 00000 | GET_IMAGE:       |        |                                |  |  |  |      |
|              |    |            |                  | .WORD  | Save R2,R3,R4,R5               |  |  |  | 1517 |
|              | 55 | 00000000'  | EF 9E 00002      | MOVAB  | SD EXECUTABLE, R5              |  |  |  |      |
|              | 54 | 00000000'  | EF 9E 00009      | MOVAB  | LNK\$GL_RELNAM_EXE, R4         |  |  |  |      |
|              | 53 | 00000000'  | EF 9E 00010      | MOVAB  | LOCAL_QUALIFIER, R3            |  |  |  |      |
|              | 5E |            | 04 C2 00017      | SUBL2  | #4, SP                         |  |  |  |      |
|              |    | 02000000   | 8F DD 0001A      | PUSHL  | #33554432                      |  |  |  | 1527 |
|              |    | 04         | AE D4 00020      | CLRL   | EXE_DESC+4                     |  |  |  |      |
|              |    |            | 52 D4 00023      | CLRL   | NEW_FILE                       |  |  |  |      |
|              |    |            | 55 DD 00025      | PUSHL  | R5                             |  |  |  | 1534 |
| 00000000G    | 00 |            | 01 FB 00027      | CALLS  | #1, CLISPRESENT                |  |  |  |      |
|              | 60 |            | 50 E9 0002E      | BLBC   | STATUS, 6\$                    |  |  |  |      |
|              | 26 |            | 63 E9 00031      | BLBC   | LOCAL_QUALIFIER, 2\$           |  |  |  | 1536 |
| 00000000G    | 8F |            | 50 D1 00034      | CMPL   | STATUS, #CLIS_LOCPRES          |  |  |  | 1538 |
|              |    |            | 01 13 0003B      | BEQL   | 1\$                            |  |  |  |      |
|              |    |            | 04 0003D         | RET    |                                |  |  |  |      |
|              |    | 01DC       | C3 9F 0003E 1\$: | PUSHAB | EXE_NAME                       |  |  |  | 1541 |
|              |    | 02DC       | C3 9F 00042      | PUSHAB | EXE_INREL                      |  |  |  |      |
|              |    | 0168       | C3 DD 00046      | PUSHL  | LAST_FNM_DESC                  |  |  |  |      |
|              |    |            | 7E D4 0004A      | CLRL   | -(SP)                          |  |  |  |      |
| 00000000V    | EF |            | 04 FB 0004C      | CALLS  | #4, SETUP_INRELNAM             |  |  |  |      |
|              | 64 | 02DC       | C3 9E 00053      | MOVAB  | EXE_INREL, LNK\$GL_RELNAM_EXE  |  |  |  | 1542 |
|              |    |            | 05 11 00058      | BRB    | 3\$                            |  |  |  | 1537 |
|              | 64 | 0108       | C3 9E 0005A 2\$: | MOVAB  | DFLT_INREL, LNK\$GL_RELNAM_EXE |  |  |  | 1545 |
|              | 52 |            | 01 D0 0005F 3\$: | MOVL   | #1, NEW_FILE                   |  |  |  | 1547 |
|              |    | FC         | A3 D4 00062      | CLRL   | UNIVSTAR_VALID                 |  |  |  | 1548 |
|              |    | 4020       | 8F BB 00065      | PUSHR  | #*M<R5,SP>                     |  |  |  | 1549 |
| 00000000G    | 00 |            | 02 FB 00069      | CALLS  | #2, CLISGET_VALUE              |  |  |  |      |
|              | 04 |            | 50 E9 00070      | BLBC   | RO, 4\$                        |  |  |  |      |
|              |    |            | 50 D4 00073      | CLRL   | RO                             |  |  |  |      |
|              |    |            | 03 11 00075      | BRB    | 5\$                            |  |  |  |      |
|              | 50 |            | 63 D0 00077 4\$: | MOVL   | LOCAL_QUALIFIER, RO            |  |  |  | 1551 |
|              | 0C | A4         | 50 90 0007A 5\$: | MOVB   | RO, LNK\$GB_LOCNV_EXE          |  |  |  | 1549 |
| 2F 00000000G | 00 |            | 02 E1 0007E      | BBC    | #2, LNK\$GL_CTLMSK, 9\$        |  |  |  | 1553 |
|              |    |            | 6E B5 00086      | TSTW   | EXE_DESC                       |  |  |  | 1554 |
|              |    |            | 2B 12 00088      | BNEQ   | 9\$                            |  |  |  |      |
|              | 6E | 0184       | C3 7D 0008A      | MOVQ   | SHARE_DESC, EXE_DESC           |  |  |  | 1555 |
|              |    |            | 24 11 0008F      | BRB    | 9\$                            |  |  |  | 1534 |
|              | 0C |            | 63 E8 00091 6\$: | BLBS   | LOCAL_QUALIFIER, 7\$           |  |  |  | 1558 |
| 00000000G    | 8F |            | 50 D1 00094      | CMPL   | STATUS, #CLIS_NEGATED          |  |  |  |      |
|              |    |            | 0C 13 0009B      | BEQL   | 8\$                            |  |  |  |      |
|              | 15 |            | 63 E9 0009D      | BLBC   | LOCAL_QUALIFIER, 9\$           |  |  |  | 1559 |
| 00000000G    | 8F |            | 50 D1 000A0 7\$: | CMPL   | STATUS, #CLIS_LOCNEG           |  |  |  |      |

|           |    |           |                  |       |                         |   |      |
|-----------|----|-----------|------------------|-------|-------------------------|---|------|
|           |    |           | 0C 12 000A7      | BNEQ  | 9\$                     | : |      |
|           |    |           | 01 D0 000A9 8\$: | MOVL  | #1, NEW FILE            | : | 1561 |
| FC        | 52 |           | 01 D0 000AC      | MOVL  | #1, UNIDSTAR_VALID      | : | 1562 |
|           | A3 |           | C5 7D 000B0      | MOVQ  | NULL_DEVICE, -EXE_DESC  | : | 1563 |
|           | 6E | 0124      | 52 E9 000B5 9\$: | BLBC  | NEW FILE, 10\$          | : | 1567 |
|           | 12 |           | 00 9E 000B8      | MOVAB | LNK\$GL_IMGFIL, OUTFILE | : | 1569 |
| 019C      | C3 | 00000000G | 5E DD 000C1      | PUSHL | SP                      | : | 1570 |
|           |    |           | 01 FB 000C3      | CALLS | #1, OUTPUTFILE          | : |      |
| 00000000V | EF |           | 04 000CA 10\$:   | RET   |                         | : | 1572 |

; Routine Size: 203 bytes, Routine Base: \$CODE\$ + 0BB7

```

: 1481      1573 1 routine get_sym : novalue =
: 1482      1574 1
: 1483      1575 1 !++
: 1484      1576 1
: 1485      1577 1 This routine retrieves the symbol table file name and qualifiers
: 1486      1578 1
: 1487      1579 1 !--
: 1488      1580 2 begin
: 1489      1581 2 local
: 1490      1582 2     sym_desc      : dynamic_descriptor,      ! Descriptor of .SYM file
: 1491      1583 2     status ;
: 1492      1584 2 own
: 1493      1585 2     sym_name      : vector [nam$c_maxrss, byte], ! Related file for .SYM
: 1494      1586 2     sym_inrel    : block [nam$c_bln, byte] ; ! Related NAMS for .SYM
: 1495      1587 2
: 1496      1588 3 if (status = cli$present (sd_symbol_table)) ! /SYMBOL_TABLE
: 1497      1589 3 then begin
: 1498      1590 3     if .local_qualifier
: 1499      1591 4     then begin
: 1500      1592 4         if .status neq cli$_locpres
: 1501      1593 4         then return
: 1502      1594 5         else begin
: 1503      1595 5             setup_inrelnam (0,.last_fnm_desc, sym_inrel, sym_name) ; ! Setup related name
: 1504      1596 5             lnk$gl_relnam_sym = sym_inrel ; ! using last input file
: 1505      1597 5             end
: 1506      1598 4         end
: 1507      1599 3     else lnk$gl_relnam_sym = dflt_inrel ; ! Use default related name
: 1508      1600 3
: 1509      1601 3     lnk$gl_ctlmsk [lnk$v_symtbl] = true ; ! Ask for symbol table
: 1510      1602 4     lnk$gb_locnov_sym = (if cli$get_value (sd_symbol_table, sym_desc) ! Get .STB name (no name is okay)
: 1511      1603 4         then false ! and setup 'local value' flag
: 1512      1604 4         else .local_qualifier
: 1513      1605 3         ) ;
: 1514      1606 3     outfile = lnk$gl_symfil ; ! Point to the pointer
: 1515      1607 3     outputfile (sym_desc) ; ! And set up output file
: 1516      1608 3     end
: 1517      1609 3 else begin
: 1518      1610 4     if ((not .local_qualifier) and (.status eql cli$_negated)) ! But, if explicitly negated,
: 1519      1611 4     or (( .local_qualifier) and (.status eql cli$_locneg)) !
: 1520      1612 3     then lnk$gl_ctlmsk [lnk$v_symtbl] = false ; ! disable production
: 1521      1613 2     end ;
: 1522      1614 1 end ;

```

.PSECT \$OWNS,NOEXE,2

```

00340 SYM_NAME:
      .BLKB 255
0043F      .BLKB 1
00440 SYM_INREL:
      .BLKB 96

```

.PSECT \$CODE\$,NOWRT,2





```

1525 1616 1 routine get_map : novalue =
1526 1617 1
1527 1618 1 !++
1528 1619 1
1529 1620 1 This routine retrieves the map file name and qualifiers
1530 1621 1
1531 1622 1 !--
1532 1623 2 begin
1533 1624 2 local
1534 1625 2     map_desc      : dynamic_descriptor,      ! Descriptor of .MAP file
1535 1626 2     status       ;
1536 1627 2 own
1537 1628 2     map_name      : vector [nam$c_maxrss, byte], ! Related file for .MAP
1538 1629 2     map_inrel     : block [nam$c_bln, byte] ;    ! Related NAMS for .MAP
1539 1630 2
1540 1631 3 if (status = cli$present (sd_map))           ! /MAP
1541 1632 3 then begin
1542 1633 3     if .local_qualifier
1543 1634 4     then begin
1544 1635 4         if .status neq cli$_locpres
1545 1636 4         then return
1546 1637 5         else begin
1547 1638 5             setup_inrelnam (0, .last_fnm_desc, map_inrel, map_name) ; ! Setup related name
1548 1639 5             lnk$gl_relnam_map = map_inrel ; ! using last input file
1549 1640 5         end
1550 1641 4         end
1551 1642 3     else lnk$gl_relnam_map = dflt_inrel ; ! Use default related name
1552 1643 3
1553 1644 3     lnk$gl_ctlmsk [lnk$v_map] = true ; ! Ask for map
1554 1645 4     lnk$gb_locnov_map = (if cli$get_value (sd_map, map_desc) ! Get map name (no name is okay)
1555 1646 4         then false ! and setup 'local value' flag
1556 1647 4         else .local_qualifier
1557 1648 3     ) ;
1558 1649 3     outfile = lnk$gl_mapfil ; ! Point to the pointer
1559 1650 3     outputfile (map_desc) ; ! And set up output file
1560 1651 3     end
1561 1652 3 else begin
1562 1653 4     if ((not .local_qualifier) and (.status eql cli$_negated)) ! But, if explicitly negated,
1563 1654 4     or (( .local_qualifier) and (.status eql cli$_locneg)) !
1564 1655 3     then lnk$gl_ctlmsk [lnk$v_map] = false ; ! disable production
1565 1656 2     end ;
1566 1657 1 end ;

```

.PSECT \$OWNS,NOEXE,2

```

004A0 MAP_NAME:
          .BLKB 255
0059F     .BLKB 1
005A0 MAP_INREL:
          .BLKB 96

```

.PSECT \$CODE\$,NOWRT,2



```

1569 1659 1 routine get_input =
1570 1660 1
1571 1661 1 !++
1572 1662 1
1573 1663 1 This routine retrieves the input files
1574 1664 1
1575 1665 1 --
1576 1666 2 begin
1577 1667 2 local status ;
1578 1668 2
1579 1669 2 if status = cli$get_value (sd_p1, token_desc) ! Get input file
1580 1670 2 then
1581 1671 3 begin
1582 1672 3 if cli$present (sd_library) ! /LIBRARY
1583 1673 4 then begin
1584 1674 4 infilflgs = .infilflgs or (fdb$m_libr or fdb$m_libsrch) ;
1585 1675 4 lnk$setlibrin () ;
1586 1676 3 end ;
1587 1677 3
1588 1678 3 if cli$present (sd_selective_search) ! /SELECTIVE_SEARCH
1589 1679 3 then infilflgs = .infilflgs or fdb$m_selser ;
1590 1680 3
1591 1681 3 if (.infilflgs and fdb$m_libextr) eql 0
1592 1682 3 then if cli$present (sd_include) ! /INCLUDE LIST
1593 1683 4 then begin
1594 1684 4 infilflgs = .infilflgs or (fdb$m_libextr or fdb$m_libr) ;
1595 1685 4 lnk$setlibrin () ;
1596 1686 3 end ;
1597 1687 3
1598 1688 3 if cli$present (sd_options) ! /OPTIONS FILE
1599 1689 4 then begin
1600 1690 4 infilflgs = .infilflgs or fdb$m_option ;
1601 1691 4 optionfile () ;
1602 1692 3 end ;
1603 1693 3
1604 1694 3 inputfile (token_desc) ; ! Set up input file FDB
1605 1695 2 end ;
1606 1696 2
1607 1697 2 return .status ;
1608 1698 1 end ;

```

|  |           |           |       |                      |        |                     |        |
|--|-----------|-----------|-------|----------------------|--------|---------------------|--------|
|  |           |           |       | 007C 0000 GET_INPUT: |        |                     |        |
|  |           |           |       |                      | .WORD  | Save R2,R3,R4,R5,R6 | : 1659 |
|  | 56        | F61C      | CF 9E | 00002                | MOVAB  | LNK\$SETLIBRIN, R6  |        |
|  | 55        | 00000000G | 00 9E | 00007                | MOVAB  | CLIPRESENT, R5      |        |
|  | 54        | 0C000000' | EF 9E | 0000E                | MOVAB  | SD P1, R4           |        |
|  | 53        | 00000000' | EF 9E | 00015                | MOVAB  | INFILFLGS, R3       |        |
|  |           | E4        | A3 9F | 0001C                | PUSHAB | TOKEN_DESC          | : 1669 |
|  |           |           | 54 DD | 0001F                | PUSHL  | R4                  |        |
|  | 00000000G | 00        | 02 FB | 00021                | CALLS  | #2, CLIPGET_VALUE   |        |
|  | 52        |           | 50 D0 | 00028                | MOVL   | R0, STATUS          |        |
|  | 48        |           | 52 E9 | 0002B                | BLBC   | STATUS, 5\$         |        |
|  |           | CC        | A4 9F | 0002E                | PUSHAB | SD_LIBRARY          | : 1672 |

|      |    |    |       |       |        |                     |      |
|------|----|----|-------|-------|--------|---------------------|------|
| 65   |    | 01 | FB    | 00031 | CALLS  | #1, CLIPRESENT      |      |
| 07   |    | 50 | E9    | 00034 | BLBC   | RO, 1\$             |      |
| 63   | 82 | 8F | 88    | 00037 | BISB2  | #130, INFILFLGS     | 1674 |
| 66   |    | 00 | FB    | 0003B | CALLS  | #0, LNK\$SETLIBRIN  | 1675 |
|      | F4 | A4 | 9F    | 0003E | PUSHAB | SD_SELECTIVE_SEARCH | 1678 |
| 65   |    | 01 | FB    | 00041 | CALLS  | #1, CLIPRESENT      |      |
| 03   |    | 50 | E9    | 00044 | BLBC   | RO, 2\$             |      |
| 63   |    | 08 | 88    | 00047 | BISB2  | #8, INFILFLGS       | 1679 |
| 63   | 10 | 06 | E0    | 0004A | BBS    | #6, INFILFLGS, 3\$  | 1681 |
|      |    | BC | A4    | 9F    | PUSHAB | SD_INCLUDE          | 1682 |
| 65   |    | 01 | FB    | 00051 | CALLS  | #1, CLIPRESENT      |      |
| 07   |    | 50 | E9    | 00054 | BLBC   | RO, 3\$             |      |
| 63   |    | 42 | 8F    | 88    | BISB2  | #66, INFILFLGS      | 1684 |
| 66   |    | 00 | FB    | 0005B | CALLS  | #0, LNK\$SETLIBRIN  | 1685 |
|      |    | DC | A4    | 9F    | PUSHAB | SD_OPTIONS          | 1688 |
| 65   |    | 01 | FB    | 00061 | CALLS  | #1, CLIPRESENT      |      |
| 07   |    | 50 | E9    | 00064 | BLBC   | RO, 4\$             |      |
| 63   |    | 10 | 88    | 00067 | BISB2  | #16, INFILFLGS      | 1690 |
| 63   | 63 | A6 | 00    | FB    | CALLS  | #0, OPTIONFILE      | 1691 |
|      |    | E4 | A3    | 9F    | PUSHAB | TOKEN_DESC          | 1694 |
| 0140 | C6 | 01 | FB    | 00071 | CALLS  | #1, INPUTFILE       |      |
|      | 50 | 52 | D0    | 00076 | MOVL   | STATUS, RO          | 1697 |
|      |    | 04 | 00079 | RET   |        |                     | 1698 |

; Routine Size: 122 bytes, Routine Base: \$CODE\$ + ODDF

; 1609 1699 1



```

: 1638      1726 1 routine setup_inrelnam
: 1639      1727 1      (ofp_flag
: 1640      1728 1      .fnm_desc : ref block [,byte]
: 1641      1729 1      .nam      : ref block [,byte]
: 1642      1730 1      .file_name : ref vector [,byte]
: 1643      1731 1      } : =
: 1644      1732 1
: 1645      1733 2 begin
: 1646      1734 2 local
: 1647      1735 2     status,
: 1648      1736 2     fab      : block [fab$c_bln, byte] ;      ! Fab $PARSE
: 1649      1737 2
: 1650      P 1738 2 $fab_init      (fab = fab      ! Intialize the fab
: 1651      P 1739 2     .dna = .typeadr
: 1652      P 1740 2     .dns = typelen
: 1653      P 1741 2     .fns = .fnm_desc [dsc$w_length]
: 1654      P 1742 2     .fna = .fnm_desc [dsc$a_pointer]
: 1655      P 1743 2     .nam = .nam
: 1656      1744 2     ) ;
: 1657      1745 2 fab [fab$v_ofp] = .ofp_flag ;      ! Decide on 'output file parse' setting
: 1658      1746 2
: 1659      P 1747 2 $nam_init      (nam = .nam      ! Initialize the nam block
: 1660      P 1748 2     .rsa = .file_name
: 1661      P 1749 2     .rss = nam$c_maxrss
: 1662      P 1750 2     .esa = .file_name
: 1663      P 1751 2     .ess = nam$c_maxrss
: 1664      1752 2     ) ;
: 1665      1753 2
: 1666      1754 2 if (status = $parse (fab=fab))
: 1667      1755 2 then
: 1668      1756 2     nam [nam$b_rsl] = .nam [nam$b_esl] ;
: 1669      1757 2
: 1670      1758 2 return .status .
: 1671      1759 1 end ;

```

.EXTRN SYSSPARSE

007C 0000 SETUP\_INRELNAM:

|      |    |    |    |      |    |    |       |  |       |                              |   |      |
|------|----|----|----|------|----|----|-------|--|-------|------------------------------|---|------|
|      |    |    |    |      |    |    |       |  | WORD  | Save R2,R3,R4,R5,R6          | : | 1726 |
| 0050 | 8F | 00 | 5E | B0   | AE | 9E | 00002 |  | MOVAB | -80(SP), SP                  | : |      |
|      |    |    | 6E |      | 00 | 2C | 00006 |  | MOVCS | #0, (SP), #0, #80, \$RMS_PTR | : | 1744 |
|      |    |    |    |      | 6E |    | 00000 |  |       |                              | : |      |
|      |    |    | 6E | 5003 | 8F | B0 | 0000E |  | MOVW  | #20483, \$RMS_PTR            | : |      |
|      |    | 16 | AE |      | 02 | 90 | 00013 |  | MOVB  | #2, \$RMS_PTR+22             | : |      |
|      |    | 1F | AE |      | 02 | 90 | 00017 |  | MOVB  | #2, \$RMS_PTR+31             | : |      |
|      |    |    | 56 | 0C   | AC | D0 | 0001B |  | MOVL  | NAM, R6                      | : |      |
|      |    | 28 | AE |      | 56 | D0 | 0001F |  | MOVL  | R6, \$RMS_PTR+40             | : |      |
|      |    |    | 50 | 08   | AC | D0 | 00023 |  | MOVL  | FNМ DESC, R0                 | : |      |
|      |    | 2C | AE | 04   | A0 | D0 | 00027 |  | MOVL  | 4(R0), \$RMS_PTR+44          | : |      |
|      |    | 34 | AE |      | 60 | 90 | 0002C |  | MOVB  | (R0), \$RMS_PTR+52           | : |      |
| 07   | AE | 01 | 05 | 04   | AC | F0 | 00030 |  | INSV  | OFF_FLAG, #5, #1, FAB+7      | : | 1745 |
| 0060 | 8F | 00 | 6E |      | 00 | 2C | 00037 |  | MOVCS | #0, -(SP), #0, #96, (R6)     | : | 1752 |
|      |    |    |    |      | 66 |    | 0003E |  |       |                              | : |      |
|      |    |    | 66 | 6002 | 8F | B0 | 0003F |  | MOVW  | #24578, (R6)                 | : |      |
|      |    | 02 | A6 |      | 01 | 8E | 00044 |  | MNEGB | #1, 2(R6)                    | : |      |

```

04 A6 10 AC D0 00048      MOVL FILE_NAME, 4(R6)
0A A6      01 8E 0004D      MNEGB #1, T0(R6)
0C A6 10 AC D0 00051      MOVL FILE_NAME, 12(R6)
00000000G 00      5E DD 00056      PUSHL SP
03 A6      01 FB 00058      CALLS #1, SYSSPARSE
05      50 E9 0005F      BLBC STATUS, 1$
03 A6 0B A6 90 00062      MOVB 11(R6), 3(R6)
04 00067 1$:      RET

```

; Routine Size: 104 bytes, Routine Base: \$CODE\$ + 0E90

```

: 1672      1760 1
: 1673      1761 1 end
: 1674      1762 0 eludom

```

.EXTRN LIB\$SIGNAL, LIB\$STOP

PSECT SUMMARY

| Name          | Bytes | Attributes   |
|---------------|-------|--|
| \$SPLITS      | 452   | NOVEC, NOWRT, RD, NOEXE, NOSHR, LCL, REL, CON, NOPIC, ALIGN(2) |
| \$GLOBALS     | 36    | NOVEC, WRT, RD, NOEXE, NOSHR, LCL, REL, CON, NOPIC, ALIGN(2)   |
| \$OWNS        | 1536  | NOVEC, WRT, RD, NOEXE, NOSHR, LCL, REL, CON, NOPIC, ALIGN(2)   |
| _LIB\$KEY0\$  | 10    | NOVEC, NOWRT, RD, EXE, SHR, LCL, REL, CON, PIC, ALIGN(1)       |
| _LIB\$STATE\$ | 60    | NOVEC, NOWRT, RD, EXE, SHR, LCL, REL, CON, PIC, ALIGN(1)       |
| _LIB\$KEY1\$  | 31    | NOVEC, NOWRT, RD, EXE, SHR, LCL, REL, CON, PIC, ALIGN(1)       |
| \$CODE\$      | 3832  | NOVEC, NOWRT, RD, EXE, NOSHR, LCL, REL, CON, NOPIC, ALIGN(2)   |

Library Statistics

| File                                   | Total | Symbols Loaded | Percent | Pages Mapped | Processing Time |
|--|-------|----------------|---------|--------------|-----------------|
| -\$255\$DUA28:[SYSLIB]STARLET.L32;1    | 9776  | 123            | 1       | 581          | 00:01.1         |
| -\$255\$DUA28:[LINKER.OBJ]DATBAS.L32;1 | 538   | 76             | 14      | 28           | 00:00.4         |
| -\$255\$DUA28:[SYSLIB]CLIMAC.L32;1     | 14    | 0              | 0       | 9            | 00:00.1         |
| -\$255\$DUA28:[SYSLIB]TPAMAC.L32;1     | 42    | 21             | 50      | 14           | 00:00.1         |

COMMAND QUALIFIERS

; BLISS/CHECK=(FIELD, INITIAL, OPTIMIZE)/LIS=LISS:LNKGETCMD/OBJ=OBJ\$:LNKGETCMD MSRC\$:LNKGETCMD/UPDATE=(ENHS:LNKGETCMD)

```

: 1675      1763 0

```



LNK\_GETCMDINP  
V04=000

D 10  
15-Sep-1984 23:57:45

VAX-11 Bliss-32 v4.0-742

Page 71

LNK  
V04

: Size: 3832 code + 2125 data bytes  
: Run Time: 01:12.1  
: Elapsed Time: 02:27.4  
: Lines/CPU Min: 1466  
: Lexemes/CPU-Min: 25965  
: Memory Used: 386 pages  
: Compilation Complete



0216 AH-BT13A-SE  
VAX/VMS V4.0

DIGITAL EQUIPMENT CORPORATION  
CONFIDENTIAL AND PROPRIETARY

