


```
1 0001 0 MODULE LNK_PSCTSERINS (IDENT='V04-000'  
2 0002 0 ADDRESSING_MODE (EXTERNAL=GENERAL,  
3 0003 0 NONEXTERNAL=LONG_RELATIVE)  
4 0004 0 ) =  
5 0005 0  
6 0006 1 BEGIN  
7 0007 1  
8 0008 1  
9 0009 1 *****  
10 0010 1 *  
11 0011 1 * COPYRIGHT (c) 1978, 1980, 1982, 1984 BY *  
12 0012 1 * DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASSACHUSETTS. *  
13 0013 1 * ALL RIGHTS RESERVED. *  
14 0014 1 *  
15 0015 1 * THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND COPIED *  
16 0016 1 * ONLY IN ACCORDANCE WITH THE TERMS OF SUCH LICENSE AND WITH THE *  
17 0017 1 * INCLUSION OF THE ABOVE COPYRIGHT NOTICE. THIS SOFTWARE OR ANY OTHER *  
18 0018 1 * COPIES THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY *  
19 0019 1 * OTHER PERSON. NO TITLE TO AND OWNERSHIP OF THE SOFTWARE IS HEREBY *  
20 0020 1 * TRANSFERRED. *  
21 0021 1 *  
22 0022 1 * THE INFORMATION IN THIS SOFTWARE IS SUBJECT TO CHANGE WITHOUT NOTICE *  
23 0023 1 * AND SHOULD NOT BE CONSTRUED AS A COMMITMENT BY DIGITAL EQUIPMENT *  
24 0024 1 * CORPORATION. *  
25 0025 1 *  
26 0026 1 * DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE OR RELIABILITY OF ITS *  
27 0027 1 * SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL. *  
28 0028 1 *  
29 0029 1 *  
30 0030 1 *****  
31 0031 1  
32 0032 1  
33 0033 1  
34 0034 1  
35 0035 1  
36 0036 1 ++  
37 0037 1  
38 0038 1 MODULE: LNK_PSCTSERINS  
39 0039 1  
40 0040 1 FACILITY: LINKER  
41 0041 1  
42 0042 1 ABSTRACT: SEARCH AND INSERT ROUTINES FOR P-SECTIONS  
43 0043 1  
44 0044 1 HISTORY:  
45 0045 1  
46 0046 1 VERSION: X01.00  
47 0047 1  
48 0048 1 AUTHOR: T.J. PORTER 16-MAR-77  
49 0049 1  
50 0050 1 MODIFIED BY:  
51 0051 1  
52 0052 1 V03-005 BLS0111 Benn Schreiber 21-Nov-1981  
53 0053 1 Ignore deleted global psects  
54 0054 1  
55 0055 1 V03-004 BLS0100 Benn Schreiber 7-Nov-1981  
56 0056 1 Don't zero name field on allocation  
57 0057 1
```

LNK_PSCTSERINS
V04=000

N 9
16-Sep-1984 00:32:33 VAX-11 Bliss-32 V4.0-742
14-Sep-1984 12:40:36 [LINKER.SRC]LNKPSCTBL.B32;1

Page 2
(1)

:	58	0058	1	:	V03-003	BLS0035	Benn Schreiber	13-Jan-1981
:	59	0059	1	:		65K psects		
:	60	0060	1	:				
:	61	0061	1	:	V03-002	BLS0025	Benn Schreiber	28-Nov-1980
:	62	0062	1	:		Add pointer from psect descriptor to cluster descriptor.		
:	63	0063	1	:				
:	64	0064	1	:	V03-001	BLS0007	Benn Schreiber,	3-Jun-1980
:	65	0065	1	:		Convert to MDL data structures		
:	66	0066	1	!--				

```

68 0067 1 |
69 0068 1 | ++
70 0069 1 |
71 0070 1 | FUNCTIONAL DESCRIPTION:
72 0071 1 |
73 0072 1 | PROGRAM SECTIONS ARE REPRESENTED BY DESCRIPTORS (NAMES OF
74 0073 1 | FORM PSC$X.YY...) IN A SINGLY LINKED, LEXICALLY ORDERED
75 0074 1 | LIST. THE [CIST HEAD IS IN THE CLUSTER DESCRIPTOR.
76 0075 1 |
77 0076 1 | THIS MODULE CONTAINS THE ROUTINES TO INSERT A P-SECTION
78 0077 1 | IN SUCH A LIST AS WELL AS TO FIND P-SECTIONS. AN INSERT
79 0078 1 | OF A P-SECTION IS PRECEDED BY A SEARCH FOR A
80 0079 1 | P-SECTION OF THE SAME NAME. IF NOT FOUND, THEN THE DESCRIPTOR IS
81 0080 1 | ALLOCATED IN THE CORRECT PLACE IN THE CURRENT CLUSTER'S LIST.
82 0081 1 |
83 0082 1 | THERE ARE TWO WAYS OF SEARCHING FOR P-SECTION
84 0083 1 | (1) BY NAME (LNK$FNDPSCNAM)
85 0084 1 | (2) BY ATTRIBUTE MASK (LNK$FNDPSCMSK)
86 0085 1 |
87 0086 1 | THE CALLING SEQUENCES ARE AS FOLLOWS:
88 0087 1 |
89 0088 1 | LNK$FNDPSCNAM (PSCTNAME,GPSBITS,SCOPE,PSCTDESC)
90 0089 1 | LNK$FNDPSCMSK (MASK,MATCH,PSCTDESC)
91 0090 1 |
92 0091 1 | WHERE:
93 0092 1 | PSCTNAME = ADDRESS OF COUNTED ASCII STRING
94 0093 1 | WHICH IS THE P-SECTION NAME
95 0094 1 | PSCTDESC = ADDRESS OF LONGWORD TO RECEIVE
96 0095 1 | THE P-SECTION DESCRIPTOR ADDRESS IF
97 0096 1 | P-SECTION FOUND OR INSERTED.
98 0097 1 | GPSBITS = PSECT ATTRIBUTES
99 0098 1 | SCOPE = 0 FOR SEARCH OF CURRENT CLUSTER ONLY
100 0099 1 | ELSE ALL DOWN TO CURRENT.
101 0100 1 | MASK = MASK OF FLAG BITS TO BE EXTRACTED
102 0101 1 | (FOR PURPOSES OF COMPARISON) FROM
103 0102 1 | THE P-SECTION DESCRIPTOR FLAGS.
104 0103 1 | MATCH = THE PATTERN OF FLAGS TO BE MATCHED
105 0104 1 | AFTER ADDING THE P-SECTION
106 0105 1 | FLAGS WITH THE MASK.
107 0106 1 | I.E., SEARCH BY MASK IS SUCCESSFUL IF (P-SECTION
108 0107 1 | FLAGS) .AND. (MASK) = MATCH.
109 0108 1 | ROUTINES HAVE THE VALUE TRUE IF P-SECTION IS
110 0109 1 | FOUND TO EXIST BEFORE THE CALL, FALSE OTHERWISE.
111 0110 1 |
112 0111 1 | --
113 0112 1 |
114 0113 1 | LIBRARY 'STARLETL32'; ! SYSTEM DATA STRUCTURES
115 0114 1 | REQUIRE 'PREFIX'; ! MACROS ETC.
116 0115 1 | LIBRARY 'DATBAS'; ! DATA BASE DEFINITIONS
117 0116 1 |
118 0231 1 |
119 0232 1 | !
120 0233 1 | GLOBAL LNK$GL_PSCALLOC : VECTOR[2]; ! LENGTH AND ADDRESS OF PSECT DESC ALLOCATION POOL
121 0234 1 |
122 0235 1 |
123 0236 1 |
124 0237 1 | EXTERNAL LITERAL

```

125	0238	1	PSCSC_ALLOBLK : BYTLIT;	! NUMBER OF PAGES TO PREALLOCATE
126	0239	1		
127	0240	1	EXTERNAL ROUTINE	
128	0241	1	LIB\$INSERT_TREE,	! LOOKUP/INSERT IN BINARY TREE
129	0242	1	LIB\$LOOKUP_TREE,	! LOOKUP IN BINARY TREE
130	0243	1	LNK\$ALLOBLR;	! DYNAMIC MEMORY ALLOCATOR
131	0244	1	!	
132	0245	1	EXTERNAL	
133	0246	1	LNK\$GW_NMODULES : WORD,	! NUMBER OF MODULES
134	0247	1	LNK\$GL_CTLMSK : BLOCK[BYTE],	
135	0248	1	LNK\$GL_CLULST : VECTOR[2],	! CLUSTER DESCRIPTOR LISTHEAD
136	0249	1	LNK\$GL_CURCLU;	! CURRENT CLUSTER DESCRIPTOR
137	0250	1		
138	0251	1	EXTERNAL LITERAL	
139	0252	1	LIB\$NORMAL;	! NORMAL RETURN FROM LIB\$INSERT_TREE

```

: 141 0253 1 GLOBAL ROUTINE LNK$COMPARE_PSCNOD (PSECTNAME, CURRENTNODE) =
: 142 0254 1 |
: 143 0255 1 | LOCAL ROUTINE TO COMPARE PSECT NAME WITH CURRENT NODE. CALLED BY
: 144 0256 1 | LIB$INSERT TREE. THIS ROUTINE DIFFERES FROM COMPARE_PSECTS IN THAT
: 145 0257 1 | CURRENTNODE IS A POINTER TO A NODE WHICH POINTS TO A PSECT DESCRIPTOR.
: 146 0258 1 |
: 147 0259 2 BEGIN
: 148 0260 2
: 149 0261 2 MAP
: 150 0262 2     PSECTNAME : REF VECTOR[ BYTE],
: 151 0263 2     CURRENTNODE : REF BLOCK[,BYTE];
: 152 0264 2
: 153 0265 2 LOCAL
: 154 0266 2     THISPSECT : REF BLOCK[,BYTE];
: 155 0267 2
: 156 0268 2     THISPSECT = .CURRENTNODE[NODE$PTR];                ! PICK UP POINTER TO PSECT DESCRIPTOR
: 157 0269 2     RETURN CH$COMPARE(.PSECTNAME[0],PSECTNAME[1],
: 158 0270 2         .THISPSECT[PSC$B_NAMLNG],THISPSECT[PSC$T_NAME])
: 159 0271 2
: 160 0272 1 END;

```

```

.TITLE LNK_PSCTSERINS
.IDENT \V04-000\
.PSECT $GLOBALS,NOEXE,2

```

```

0000 LNK$GL_PSCALLOC:
.BLK8 8

```

```

.EXTRN PSC$C_ALLOBLK, LIB$INSERT_TREE
.EXTRN LIB$LOOKUP_TREE
.EXTRN LNK$ALLOBLK, LNK$GW_MODULES
.EXTRN LNK$GL_CTLMSK, LNK$GL_CLULST
.EXTRN LNK$GL_CURCLU, LIB$NORMAL

```

```

.PSECT $CODE$,NOWRT,2

```

				001C 00000	.ENTRY LNK\$COMPARE_PSCNOD, Save R2,R3,R4	: 0253
	50	08	AC	D0 00002	MOVL CURRENTNODE, R0	: 0268
	50	0A	A0	D0 00006	MOVL 10(R0), THISPSECT	
	52	04	AC	D0 0000A	MOVL PSECTNAME, R2	: 0269
	53		62	9A 0000E	MOVZBL (R2), R3	
	51	2D	A0	9A 00011	MOVZBL 45(THISPSECT), R1	: 0270
	54		01	D0 00015	MOVL #1, R4	
51		00	01	A2	CMPCS R3, 1(R2), #0, R1, 46(THISPSECT)	
			2E	A0		
			03	1A 00020	BGTRU 1\$	
	54		01	D9 00022	SBWC #1, R4	
	50		54	D0 00025 1\$:	MOVL R4, R0	
			04	00028	RET	: 0272

: Routine Size: 41 bytes, Routine Base: \$CODE\$ + 0000

```

: 162 0273 1 GLOBAL ROUTINE LNK$FNDPSCNAM(PSCNAME,GPSBITS,SCOPE,PSCTDESC)=
: 163 0274 2 BEGIN
: 164 0275 2
: 165 0276 2 FIND P-SECTION BY NAME. PSCNAME IS THE ADDRESS OF THE ASCII
: 166 0277 2 STRING P-SECTION NAME. PSCTDESC IS THE PLACE TO PUT THE
: 167 0278 2 FOUND OR ALLOCATED DESCRIPTOR. IF SCOPE IS NON ZERO, ALL CLUSTERS
: 168 0279 2 ARE SEARCHED. IF PSECT IS NOT FOUND, IT IS ENTERED IN CURRENT CLUSTER.
: 169 0280 2 GPSBITS ARE THE PSECT ATTRIBUTES FROM THE PSECT DEFINITION IN THE GSD.
: 170 0281 2
: 171 0282 2
: 172 0283 2 ROUTINE COMPARE_PSECTS (PSECTNAME, CURRENTNODE) =
: 173 0284 2
: 174 0285 2 LOCAL ROUTINE TO COMPARE PSECT NAME WITH CURRENT DESCRIPTOR. CALLED BY
: 175 0286 2 LIB$INSERT_TREE
: 176 0287 2
: 177 0288 2 BEGIN
: 178 0289 2
: 179 0290 2 MAP
: 180 0291 2 PSECTNAME : REF VECTOR[,BYTE],
: 181 0292 2 CURRENTNODE : REF BLOCK[,BYTE];
: 182 0293 2
: 183 0294 2 RETURN CH$COMPARE(.PSECTNAME[0],PSECTNAME[1],
: 184 0295 2 .CURRENTNODE[PSCT$B_NAMLANG],CURRENTNODE[PSCT$T_NAME])
: 185 0296 2
: 186 0297 2 END;

```

					001C 0000 COMPARE_PSECTS:		
					.WORD	Save R2,R3,R4	: 0283
		52	04	AC D0 00002	MOVL	PSECTNAME, R2	: 0294
		53		62 9A 00006	MOVZBL	(R2), R3	
		50	08	AC D0 00009	MOVL	CURRENTNODE, R0	: 0295
		51	2D	A0 9A 0000D	MOVZBL	45(R0), R1	
		54		01 D0 00011	MOVL	#1, R4	
51	00	01	A2	53 2D 00014	CMPC5	R3, 1(R2), #0, R1, 46(R0)	
				2E A0 0001A			
				03 1A 0001C	BGTRU	1\$	
		54		01 D9 0001E	SBWC	#1, R4	
		50		54 D0 00021	MOVL	R4, R0	: 0297
				04 00024	RET		

: Routine Size: 37 bytes, Routine Base: \$CODE\$ + 0029

```

: 187 0298 2
: 188 0299 2 ROUTINE ALLOC_PSECT (PSECTNAME,RETADR) =
: 189 0300 2 BEGIN
: 190 0301 2
: 191 0302 2 LOCAL ROUTINE TO ALLOCATE PSECT DESCRIPTOR. CALLED BY LIB$INSERT_NODE
: 192 0303 2
: 193 0304 2 MAP
: 194 0305 2 PSECTNAME : REF VECTOR[,BYTE];
: 195 0306 2
: 196 0307 2 LOCAL

```

```

: 197      0308      3      BLOCKSIZE;
: 198      0309
: 199      0310      3      BLOCKSIZE = PSC$C SIZE+.PSECTNAME[0];           ! COMPUTE SIZE OF BLOCK
: 200      0311      3      IF .LNK$GL_PSCALLOCC[0] LEQU .BLOCKSIZE           ! IF NO ROOM IN THIS PREALLOCATED BLOCK
: 201      0312      4      THEN BEGIN                                     ! THEN ALLOCATE ANOTHER ONE
: 202      0313      4      LNK$ALLOBLK(PSC$C ALLOBLK*512, LNK$GL_PSCALLOCC[1]);
: 203      0314      4      LNK$GL_PSCALLOCC[0] = PSC$C_ALLOBLK*512;
: 204      0315      3      END;
: 205      0316
: 206      0317      3      .RETADR = .LNK$GL_PSCALLOCC[1];           ! RETURN BLOCK ADDRESS
: 207      0318      3      CH$FILL(0, PSC$C_SIZE, .LNK$GL_PSCALLOCC[1]);       ! ZERO THE BLOCK
: 208      0319      3      LNK$GL_PSCALLOCC[0] = .LNK$GL_PSCALLOCC[0] - .BLOCKSIZE; ! UPDATE SIZE AND ADDRESS IN PREALLOCATED BLOCK
: 209      0320      3      LNK$GL_PSCALLOCC[1] = .LNK$GL_PSCALLOCC[1] + .BLOCKSIZE;
: 210      0321
: 211      0322      3      RETURN TRUE
: 212      0323      2      END;

```

```

                                00FC 00000 ALLOC_PSECT:
                                .WORD      Save R2,R3,R4,R5,R6,R7           : 0299
57 00000000' EF 9E 00002      MOVAB      LNK$GL_PSCALLOCC+4, R7
56          04 BC 9A 00009      MOVZBL   @PSECTNAME, BLOCKSIZE           : 0310
56          2E C0 0000D      ADDL2   #46, BLOCKSIZE
56          FC A7 D1 00010      CMPL    LNK$GL_PSCALLOCC, BLOCKSIZE     : 0311
                                17 1A 00014      BGTRU   1$
                                57 DD 00016      PUSHL   R7                               : 0313
                                8F DD 00018      PUSHL   #<PSC$C_ALLOBLK*512>
00000000G 00          02 FB 0001E      CALLS   #2, LNK$ALLOBLK
                                FC A7 00000000* 8F D0 00025      MOVL    #<PSC$C_ALLOBLK*512>, LNK$GL_PSCALLOCC : 0314
2E          08 BC          67 D0 0002D 1$: MOVL    LNK$GL_PSCALLOCC+4, @RETADR       : 0317
                                00 2C 00031      MOVCS   #0, (SP), #0, #46, @LNK$GL_PSCALLOCC+4 : 0318
                                FC A7          56 C2 00038      SUBL2   BLOCKSIZE, LNK$GL_PSCALLOCC       : 0319
                                67          56 C0 0003C      ADDL2   BLOCKSIZE, LNK$GL_PSCALLOCC+4     : 0320
                                50          01 D0 0003F      MOVL    #1, R0                            : 0322
                                04 00042      RET                                         : 0323

```

: Routine Size: 67 bytes, Routine Base: \$CODE\$ + 004E

```

: 213      0324      2      ROUTINE ALLOC_NODE (PSECTNAME, RETADR, PSECTDESC) =
: 214      0325      2      BEGIN
: 215      0326      2      !
: 216      0327      2      ! LOCAL ROUTINE TO ALLOCATE A NODE WHICH WILL POINT TO A PSECT
: 217      0328      2      ! DESCRIPTOR
: 218      0329      2      !
: 219      0330      2      MAP
: 220      0331      2      PSECTNAME : REF VECTOR[,BYTE];
: 221      0332      2      LOCAL
: 222      0333      2      BLOCKADDR : REF BLOCK[,BYTE];
: 223      0334      2      !
: 224      0335      2      IF .LNK$GL_PSCALLOCC[0] LEQU NODE$C_LONG           ! IF NOT ROOM IN CURRENT PREALLOCATED BLOCK
: 225      0336
: 226      0337

```

```

: 227 0338 4 THEN BEGIN ! THEN ALLOCATE ANOTHER BLOCK
: 228 0339 4 LNK$ALLOBLK(PSC$C_ALLOBLK*512, LNK$GL_PSCALLOC[1]);
: 229 0340 4 LNK$GL_PSCALLOC[0] = PSC$C_ALLOBLK*512;
: 230 0341 3 END;
: 231 0342 3
: 232 0343 3 BLOCKADDR = .LNK$GL_PSCALLOC[1]; ! GET ADDRESS OF NEW BLOCK
: 233 0344 3 .RETADR = .BLOCKADDR; ! RETURN ADDRESS TO CALLER
: 234 0345 3 BLOCKADDR[NODE$PTR] = .PSECTDESC; ! POINT NODE TO PSECT DESCRIPTOR
: 235 0346 3 LNK$GL_PSCALLOC[0] = .LNK$GL_PSCALLOC[0] - NODE$C_LONG; ! UPDATE POINTER AND COUNTER FOR PREALLOCATED BLOCK
: 236 0347 3 LNK$GL_PSCALLOC[1] = .LNK$GL_PSCALLOC[1] + NODE$C_LONG;
: 237 0348 3
: 238 0349 3 RETURN TRUE
: 239 0350 2 END;

```

```

                                0004 00000 ALLOC_NODE:
                                .WORD Save R2 ; 0325
                                MOVAB LNK$GL_PSCALLOC, R2 ; 0325
                                CMPL LNK$GL_PSCALLOC, #14 ; 0337
                                BGTRU 1$ ; 0339
                                PUSHAB LNK$GL_PSCALLOC+4 ; 0339
                                PUSHL #<PSC$C_ALLOBLK*512> ; 0339
                                CALLS #2, LNK$ALLOBLK ; 0339
                                MOVL #<PSC$C_ALLOBLK*512>, LNK$GL_PSCALLOC ; 0340
                                MOVL LNK$GL_PSCALLOC+4, BLOCKADDR ; 0343
                                MOVL BLOCKADDR, @RETADR ; 0344
                                MOVL PSECTDESC, 10(BLOCKADDR) ; 0345
                                SUBL2 #14, LNK$GL_PSCALLOC ; 0346
                                ADDL2 #14, LNK$GL_PSCALLOC+4 ; 0347
                                MOVL #1, R0 ; 0349
                                RET ; 0350
00000000G 00 04 A2 9F 0000E 8F DD 00011 02 FB 00017 8F D0 0001E 50 04 A2 D0 00025 1$: MOVL LNK$GL_PSCALLOC+4, BLOCKADDR
08 BC 50 D0 00029 MOVL BLOCKADDR, @RETADR
0A A0 0C AC D0 0002D MOVL PSECTDESC, 10(BLOCKADDR)
04 A2 0E C2 00032 SUBL2 #14, LNK$GL_PSCALLOC
50 01 D0 00039 ADDL2 #14, LNK$GL_PSCALLOC+4
04 0003C 01 D0 00039 MOVL #1, R0
04 0003C RET

```

; Routine Size: 61 bytes, Routine Base: \$CODE\$ + 0091

```

: 240 0351 2
: 241 0352 2 :
: 242 0353 2 : MAIN BODY OF LNK$FNDPSCNAM
: 243 0354 2 :
: 244 0355 2 MAP
: 245 0356 2 PSCTNAME : REF VECTOR[,BYTE]; ! FOR ASCII STRING
: 246 0357 2 LOCAL
: 247 0358 2 JUND,
: 248 0359 2 NDESC : REF BLOCK[,BYTE],
: 249 0360 2 PDESC : REF BLOCK[,BYTE],
: 250 0361 2 CLUSTER : REF BLOCK[,BYTE]; ! CLUSTER DESCRIPTOR
: 251 0362 2 !
: 252 0363 2 IF .SCOPE NEQ 0 ! IF GLOBAL SEARCH SCOPE
: 253 0364 2 THEN CLUSTER = LNK$GL_CLULST[0] ! START AT TOP OF CLUSTER LIST
: 254 0365 2 ELSE CLUSTER = LNK$GL_CURCLU; ! IF LOCAL START AT CURRENT CLUSTER
: 255 0366 2
: 256 0367 2 WHILE (CLUSTER = .CLUSTER[CLUSL_NXTCLU]) NEQ 0 ! MOVE ON TO NEXT CLUSTER IN LIST
: 257 0368 2 DO BEGIN
: 258 0369 2 IF .CLUSTER EQL .LNK$GL_CURCLU ! IF THIS IS CURRENT CLUSTER

```

```

259 0370 3      AND .SCOPE EQL 0                                ! AND PSECT IS NON-GLOBAL
260 0371 4      THEN BEGIN
261 0372 4      FOUND = LIB$INSERT_TREE(CLUSTER[CLUSL_LPSLST],.PSCTNAME,%REF(0),
262 0373 4      COMPARE_PSECTS,ALLOC_PSECT,PDESC);
263 0374 4      |
264 0375 4      | FILL IN THE ALLOCATED NODE
265 0376 4      |
266 0377 4      | IF .FOUND EQL LIB$NORMAL
267 0378 5      | THEN BEGIN
268 0379 5      |   CH$MOVE(.PSCTNAME[0]+1,PSCTNAME[0],PDESC[PSC$B_NAMLNG]); ! COPY IN THE NAME
269 0380 5      |   PDESC[PSC$L_LSTMPCL] = PDESC[PSC$L_MPCLST]; ! INIT LAST ENTRY POINTER
270 0381 5      |   PDESC[PSC$L_CLUDSC] = .LNK$GL_CURCLU; ! SET CLUSTER DESCRIPTOR ADDRESS
271 0382 5      |   PDESC[PSC$L_OMDNUM] = .LNK$GW_NMODULES; ! SET DEFINER MODULE NUMBER
272 0383 5      |   PDESC[PSC$B_ALIGN] = -1; ! FLAG NO ALIGNMENT SET
273 0384 5      |   IF (.GPSBITS AND GPSSM GBL) NEQ 0 ! IF THIS IS A GLOBAL PSECT
274 0385 5      |   THEN LIB$INSERT_TREE(CLUSTER[CLUSL_GPSLST], ! THEN INSERT INTO GLOBAL PSECT LIS
275 0386 5      |   .PSCTNAME,%REF(0),
276 0387 5      |   LNK$COMPARE_PSCNOD,ALLOC_NODE,NDESC,.PDESC);
277 0388 4      |
278 0389 4      |   .PSCTDESC = .PDESC; ! RETURN ADDRESS TO CALLER
279 0390 4      |   RETURN .FOUND
280 0391 4      | END
281 0392 3      | ELSE IF LIB$LOOKUP_TREE(CLUSTER[CLUSL_GPSLST],.PSCTNAME, ! GLOBAL PSECT, SO LOOK IN GLOBAL PS
282 0393 3      |   LNK$COMPARE_PSCNOD,PDESC)
283 0394 4      |   AND NOT (PDESC = .PDESC[NODE$L_PTR]; ! GET PSECT DESCRIPTOR ADDRESS
284 0395 4      |   .PDESC[PSC$V_DELETED]); ! TEST IF PSECT HAS BEEN DELETED
285 0396 4      |   THEN BEGIN
286 0397 4      |   .PSCTDESC = .PDESC; ! FOUND, SO RETURN TO CALLER
287 0398 4      |   RETURN TRUE;
288 0399 3      |   END;
289 0400 2      | END;
290 0401 2      |
291 0402 2      | PSECT WAS NOT FOUND IN ANY CLUSTER, AND IF WE GOT HERE IT MUST HAVE BEEN A GLOBAL
292 0403 2      | PSECT. CALL LNK$FNDPSCNAM AGAIN WITH LOCAL SCOPE TO ENTER IN CURRENT CLUSTER.
293 0404 2      |
294 0405 2      | RETURN LNK$FNDPSCNAM (.PSCTNAME, .GPSBITS, 0, .PSCTDESC) ! CALL WITH LOCAL SCOPE TO ENTER IN
295 0406 2      |
296 0407 1      | END; ! OF LNK$FNDPSCNAM

```

		OFFC 00000		.ENTRY	LNK\$FNDPSCNAM, Save R2,R3,R4,R5,R6,R7,R8,-	0273
					R9,R10,R11	
5B	00000000G	00	9E 00002	MOVAB	LNK\$GL_CURCLU, R11	
5A	FF25	CF	9E 00009	MOVAB	LNK\$COMPARE_PSCNOD, R10	
5E		0C	C2 0000E	SUBL2	#12, SP	
		0C	AC D5 00011	TSTL	SCOPE	0363
		09	13 00014	BEQL	1\$	
57	00000000G	00	9E 00016	MOVAB	LNK\$GL_CLULST, CLUSTER	0364
		03	11 0001D	BRB	2\$	
57		6B	9E 0001F 1\$:	MOVAB	LNK\$GL_CURCLU, CLUSTER	0365
58	04	AC	D0 00022 2\$:	MOVL	PSCTNAME, R8	0372
57		67	D0 00026 3\$:	MOVL	(CLUSTER), CLUSTER	0367
		03	12 00029	BNEQ	4\$	
		00B1	31 0002B	BRW	9\$	

		68		57	D1	0002E	4\$:	C MPL	CLUSTER, LNK\$GL_CURCLU	:	0369		
				7A	12	00031		BNEQ	6\$:			
			0C	AC	D5	00033		TSTL	SCOPE	:	0370		
				75	12	00036		BNEQ	6\$:			
			08	AE	9F	00038		PUSHAB	PDESC	:	0372		
			4E	AA	9F	0003B		PUSHAB	ALLOC_PSECT	:			
			29	AA	9F	0003E		PUSHAB	COMPARE_PSECTS	:			
			0C	AE	D4	00041		CLRL	12(SP)	:			
			0C	AE	9F	00044		PUSHAB	12(SP)	:			
				58	DD	00047		PUSHL	R8	:			
			10	A7	9F	00049		PUSHAB	16(CLUSTER)	:			
	00000000G	00		06	FB	0004C		CALLS	#6, LIB\$INSERT_TREE	:			
		59		50	D0	00053		MOVL	R0, FOUND	:			
	00000000G	8F		59	D1	00056		C MPL	FOUND, #LIB\$_NORMAL	:	0377		
				45	12	0005D		BNEQ	5\$:			
			50	68	9A	0005F		MOVZBL	(R8), R0	:	0379		
				50	D6	00062		INCL	R0	:			
			56	08	AE	D0	00064	MOVL	PDESC, R6	:			
2D	A6			68	50	28	00068	MOV3	R0, (R8), 45(R6)	:			
				10	A6	0C	0006D	MOVAB	12(R6), 16(R6)	:	0380		
		24		A6	6B	D0	00072	MOVL	LNK\$GL_CURCLU, 36(R6)	:	0381		
		28		A6	00	3C	00076	MOVZWL	LNK\$GW_NMODULES, 40(R6)	:	0382		
		2C		A6	01	8E	0007E	MNEGB	#1, 44(R6)	:	0383		
		08		AC	04	E1	00082	BBC	#4, GPSBITS, 5\$:	0384		
1D					56	DD	00087	PUSHL	R6	:	0387		
				08	AE	9F	00089	PUSHAB	NDESC	:	0385		
				0091	CA	9F	0008C	PUSHAB	ALLOC_NODE	:			
					5A	DD	00090	PUSHL	R10	:			
				10	AE	D4	00092	CLRL	16(SP)	:	0386		
				10	AE	9F	00095	PUSHAB	16(SP)	:			
					58	DD	00098	PUSHL	R8	:			
				14	A7	9F	0009A	PUSHAB	20(CLUSTER)	:	0385		
	00000000G	00		07	FB	0009D		CALLS	#7, LIB\$INSERT_TREE	:			
		10		08	AE	D0	000A4	5\$:	MOVL	PDESC, @PSCDESC	:	0389	
				50	D0	000A9		MOVL	FOUND, R0	:	0390		
					04	000AC		RET		:			
				08	AE	9F	000AD	6\$:	PUSHAB	PDESC	:	0392	
				0500	8F	BB	000B0	PUSHR	#*M<R8,R10>	:			
				14	A7	9F	000B4	PUSHAB	20(CLUSTER)	:			
	00000000G	00		04	FB	000B7		CALLS	#4, LIB\$LOOKUP_TREE	:			
		03		50	E8	000BE		BLBS	R0, 8\$:			
				FF62	31	000C1		7\$:	BRW	3\$:	0394	
				50	08	AE	D0	000C4	8\$:	MOVL	PDESC, R0	:	
		08		0A	A0	D0	000C8	MOVL	10(R0), PDESC	:			
				08	AE	D0	000CD	MOVL	PDESC, R0	:	0395		
EB		08		06	E0	000D1		BBS	#6, 11(R0), 7\$:			
		10		08	AE	D0	000D6	MOVL	PDESC, @PSCDESC	:	0397		
				50	01	D0	000DB	MOVL	#1, R0	:	0398		
					04	000DE		RET		:			
				10	AC	DD	000DF	9\$:	PUSHL	PSCDESC	:	0405	
					7E	D4	000E2	CLRL	-(SP)	:			
				08	AC	DD	000E4	PUSHL	GPSBITS	:			
					58	DD	000E7	PUSHL	R8	:			
	00CE	CA		04	FB	000E9		CALLS	#4, LNK\$FNTPSCNAM	:			
				04	000EE			RET		:	0407		

; Routine Size: 239 bytes, Routine Base: \$CODE\$ + 00CE

LNK_PSCTSERINS
V04=000

J 10
16-Sep-1984 00:32:33
14-Sep-1984 12:40:36

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

Page 11
(4)

LN
VC
61
64
6E

45
77

55
63

60
65

7E
6F

61
20
72

20
74
00

00
00
00
00

```

298 0408 1 GLOBAL ROUTINE LNK$FNDPSCMSK(MASK,MATCH,ACTION_ROUTINE) =
299 0409 2 BEGIN
300 0410 2
301 0411 2 FIND NEXT P-SECTION (IN CURRENT CLUSTER) WHOSE ATTRIBUTES
302 0412 2 HAVE A PARTICULAR VALUE (MATCH) AFTER EXTRACTING
303 0413 2 WITH MASK.
304 0414 2
305 0415 2 OWN
306 0416 2 PREVENTRY : REF BLOCK[,BYTE], ! PREVIOUS ENTRY
307 0417 2 PREVMASK, ! MASK ON LAST CALL
308 0418 2 PREVMATCH; ! MATCH ON LAST CALL
309 0419 2
310 0420 2 MAP
311 0421 2 LNK$GL_CURCLU : REF BLOCK[,BYTE]; ! CHANGE ATTRIBUTES
312 0422 2
313 0423 2 LOCAL
314 0424 2 CURENTRY : REF BLOCK[,BYTE]; ! CURRENT ENTRY BEING EXAMINED
315 0425 2
316 0426 2 IF .PREVENTRY EQL 0 ! IF FIRST TIME IN
317 0427 2 OR .MASK NEQ .PREVMASK ! OR A NEW SEARCH
318 0428 2 OR .MATCH NEQ .PREVMATCH ! PATTERN
319 0429 2 THEN BEGIN ! START AT THE BEGINNING OF THE CORRECT LIST
320 0430 2 PREVENTRY = .LNK$GL_CURCLU[CLUS$L_LPSLST];
321 0431 2 PREVMASK = .MASK; ! RESET MASK AND
322 0432 2 PREVMATCH = .MATCH; ! MATCH VALUES.
323 0433 2 END;
324 0434 2
325 0435 2 IF (CURENTRY = .PREVENTRY) EQL 0 ! IF NO PSECTS AT ALL
326 0436 2 THEN RETURN TRUE; ! THEN ALL DONE
327 0437 2 PREVENTRY = .PREVENTRY[PSC$LEFT]; ! PROCESS LEFT SUBTREE
328 0438 2 IF .PREVENTRY NEQ 0 ! IF THERE IS A LEFT SUBTREE
329 0439 2 THEN LNK$FNDPSCMSK(.MASK,.MATCH,.ACTION_ROUTINE);
330 0440 2 PREVENTRY = .CURENTRY; ! PROCESS CURRENT NODE
331 0441 2 IF (.CURENTRY[PSC$W_FLAGS] AND .PREVMASK) EQL .PREVMATCH ! IF THIS PSECT IS CORECT
332 0442 2 AND NOT .CURENTRY[PSC$V_DELETED] ! AND IT HAS NOT BEEN DELETED
333 0443 2 THEN (.ACTION_ROUTINE)(.CURENTRY); ! CALL ROUTINE FOR CURRENT ENTRY
334 0444 2 PREVENTRY = .CURENTRY[PSC$RIGHT]; ! PROCESS RIGHT SUBTREE
335 0445 2 IF .PREVENTRY NEQ 0
336 0446 2 THEN LNK$FNDPSCMSK(.MASK,.MATCH,.ACTION_ROUTINE); ! PROCESS RIGHT SUBTREE IF PRESENT
337 0447 2 PREVENTRY = .CURENTRY; ! RESTORE ENTRY
338 0448 2 RETURN TRUE;
339 0449 1 END;

```

```

.PSECT $OWNS,NOEXE,2
0000 PREVENTRY:
      .BLKB 4
0004 PREVMASK:
      .BLKB 4
0008 PREVMATCH:
      .BLKB 4

```

```
.PSECT $CODES,NOWRT,2
```

			001C 00000	.ENTRY	LNK\$FNDPSCMSK, Save R2,R3,R4	: 0408
	54	FB	AF 9E 00002	MOVAB	LNK\$FNDPSCMSK, R4	:
	53	00000000'	EF 9E 00006	MOVAB	PREVENTRY, R3	:
			63 D5 0000D	TSTL	PREVENTRY	: 0426
			0E 13 0000F	BEQL	1\$:
04	A3	04	AC D1 00011	CPL	MASK, PREVMASK	: 0427
			07 12 00016	BNEQ	1\$:
08	A3	08	AC D1 00018	CPL	MATCH, PREVMATCH	: 0428
			10 13 0001D	BEQL	2\$:
	50	00000000G	00 D0 0001F 1\$:	MOVL	LNK\$GL_CURCLU, R0	: 0430
	63	10	A0 D0 00026	MOVL	16(R0), PREVENTRY	:
04	A3	04	AC 7D 0002A	MOVQ	MASK, PREVMASK	: 0431
	52		63 D0 0002F 2\$:	MOVL	PREVENTRY, CURENTRY	: 0435
			41 13 00032	BEQL	6\$:
	73		93 D0 00034	MOVL	@PREVENTRY, PREVENTRY	: 0437
			0A 13 00037	BEQL	3\$: 0438
	7E	08	AC 7D 00039	MOVQ	MATCH, -(SP)	: 0439
		04	AC DD 0003D	PUSHL	MASK	:
	64		03 FB 00040	CALLS	#3, LNK\$FNDPSCMSK	:
	63		52 D0 00043 3\$:	MOVL	CURENTRY, PREVENTRY	: 0440
	50	0A	A2 3C 00046	MOVZWL	10(CURENTRY), R0	: 0441
	51	04	A3 D2 0004A	MCOML	PREVMASK, R1	:
	50		51 CA 0004E	BICL2	R1, R0	:
08	A3		50 D1 00051	CPL	R0, PREVMATCH	:
			0B 12 00055	BNEQ	4\$:
06	0B	A2	06 E0 00057	BBS	#6, 11(CURENTRY), 4\$: 0442
			52 DD 0005C	PUSHL	CURENTRY	: 0443
	0C	BC	01 FB 0005E	CALLS	#1, @ACTION_ROUTINE	:
	63	04	A2 D0 00062 4\$:	MOVL	4(CURENTRY), PREVENTRY	: 0444
			0A 13 00066	BEQL	5\$: 0445
	7E	08	AC 7D 00068	MOVQ	MATCH, -(SP)	: 0446
		04	AC DD 0006C	PUSHL	MASK	:
	64		03 FB 0006F	CALLS	#3, LNK\$FNDPSCMSK	:
	63		52 D0 00072 5\$:	MOVL	CURENTRY, PREVENTRY	: 0447
	50		01 D0 00075 6\$:	MOVL	#1, R0	: 0448
			04 00078	RET		: 0449

: Routine Size: 121 bytes, Routine Base: \$CODE\$ + 01BD

: 340 0450 0 END ELUDOM

PSECT SUMMARY

Name	Bytes	Attributes
\$GLOBALS	8	NOVEC, WRT, RD, NOEXE, NOSHR, LCL, REL, CON, NOPIC, ALIGN(2)
\$CODE\$	566	NOVEC, NOWRT, RD, EXE, NOSHR, LCL, REL, CON, NOPIC, ALIGN(2)
\$OWNS	12	NOVEC, WRT, RD, NOEXE, NOSHR, LCL, REL, CON, NOPIC, ALIGN(2)

Library Statistics

File	----- Total	Symbols Loaded	----- Percent	Pages Mapped	Processing Time
\$255SDUA28:[SYSLIB]STARLET.L32;1	9776	7	0	581	00:01.0
-\$255SDUA28:[LINKER.OBJ]DATBAS.L32;1	538	17	3	28	00:00.5

COMMAND QUALIFIERS

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

: Size: 566 code + 20 data bytes
: Run Time: 00:11.9
: Elapsed Time: 00:52.7
: Lines/CPU Min: 2261
: Lexemes/CPU-Min: 14798
: Memory Used: 97 pages
: Compilation Complete

LNKPROLTB
LIS

LNKSYMTBL
LIS

LNKSYMOUT
LIS

LNKUMALLO
LIS

LNKPSCTBL
LIS

LNKPROSHR
LIS

LNKSTATSD
LIS