

\_S

Ps

Yz

Zs

Zs

Zs

Zs

Zs

Zs

Zs

Zs

Zs

Zs

Zs

Zs

Zs

Zs

Zs

Zs

```
SSSSSSSSSSSSSS YYY YYY SSSSSSSSSSSSS
SSSSSSSSSSSSSS YYY YYY SSSSSSSSSSSSS
SSSSSSSSSSSSSS YYY YYY SSSSSSSSSSSSS
SSS           YYY YYY SSS
SSS           YYY YYY SSS
SSS           YYY YYY SSS
SSS           YYY YYY SSS
SSS           YYY YYY SSS
SSS           YYY YYY SSS
SSSSSSSSSSSS YYY YYY SSSSSSSSSSSSS
SSSSSSSSSSSS YYY YYY SSSSSSSSSSSSS
SSSSSSSSSSSS YYY YYY SSSSSSSSSSSSS
SSS           YYY YYY SSS
SSS           YYY YYY SSS
SSS           YYY YYY SSS
SSS           YYY YYY SSS
SSS           YYY YYY SSS
SSS           YYY YYY SSS
SSS           YYY YYY SSS
SSS           YYY YYY SSS
SSS           YYY YYY SSS
SSS           YYY YYY SSS
SSSSSSSSSSSS YYY YYY SSSSSSSSSSSSS
SSSSSSSSSSSS YYY YYY SSSSSSSSSSSSS
SSSSSSSSSSSS YYY YYY SSSSSSSSSSSSS
```



(1)	40	HISTORY	; DETAILED
(2)	66	DECLARATIONS	
(3)	127	ALLOCPFN - ALLOCATE A PAGE FROM THE FREE PAGE LIST	
(4)	177	DELCONPFN - DELETE CONTENTS OF PFN	
(6)	283	REMPFN - REMOVE A PFN FROM LIST	
(7)	391	RLPFNSAVPTE - RELEASE PFN SAVING PAGE TABLE ENTRY	
(8)	445	DELPFNLIST - DELETE PFN FROM PFN LIST	
(9)	490	RELPFN - RELEASE PFN TO FREE OR MODIFY LIST	
(10)	582	INSPFN - INSERT PFN AT HEAD OR TAIL OF PFN LIST	
(11)	736	ALLOCONTIG - Allocate N Physically Contiguous Pages	

```

0000 1 .TITLE ALLOCPFN - PFN LIST MANIPULATING ROUTINES
0000 2 .IDENT 'V04-000'
0000 3
0000 4 :*****
0000 5 :*
0000 6 :* COPYRIGHT (c) 1978, 1980, 1982, 1984 BY
0000 7 :* DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASSACHUSETTS.
0000 8 :* ALL RIGHTS RESERVED.
0000 9 :*
0000 10 :* THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND COPIED
0000 11 :* ONLY IN ACCORDANCE WITH THE TERMS OF SUCH LICENSE AND WITH THE
0000 12 :* INCLUSION OF THE ABOVE COPYRIGHT NOTICE. THIS SOFTWARE OR ANY OTHER
0000 13 :* COPIES THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY
0000 14 :* OTHER PERSON. NO TITLE TO AND OWNERSHIP OF THE SOFTWARE IS HEREBY
0000 15 :* TRANSFERRED.
0000 16 :*
0000 17 :* THE INFORMATION IN THIS SOFTWARE IS SUBJECT TO CHANGE WITHOUT NOTICE
0000 18 :* AND SHOULD NOT BE CONSTRUED AS A COMMITMENT BY DIGITAL EQUIPMENT
0000 19 :* CORPORATION.
0000 20 :*
0000 21 :* DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE OR RELIABILITY OF ITS
0000 22 :* SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL.
0000 23 :*
0000 24 :*
0000 25 :*****
0000 26
0000 27 :++
0000 28 : FACILITY: EXECUTIVE, MEMORY MANAGEMENT SUBROJTINES
0000 29
0000 30 : ABSTRACT: ALLOCPFN CONTAINS THE ROUTINES FOR MANIPULATING PFN LISTS
0000 31 : THE KNOWLEDGE OF THE FORMAT AND LOCATION OF THESE LISTS IS RESTRICTED
0000 32 : TO THIS MODULE, WRTMFYPAG, AND SWAPPER. SINCE SHRCNT AND WSLX
0000 33 : OVERLAP FLINK AND BLINK, THOSE WORD REFERENCES CAN ALSO BE THOUGH
0000 34 : OF AS WORD WIDTH DEPENDENCIES FOR THE PFN
0000 35
0000 36 : ENVIRONMENT: THESE ROUTINES MUST ALL BE CALLED AT THE SYNCH IPL!!!
0000 37
0000 38 :--
0000 39
0000 40 : .SBTTL HISTORY ; DETAILED
0000 41
0000 42 : AUTHOR: PETER H. LIPMAN , CREATION DATE: 14-SEP-76
0000 43
0000 44 : MODIFIED BY:
0000 45
0000 46 : V03-005 KPL0002 Peter Lieberwirth 22-Feb-1984
0000 47 : Fix bugs in V03-004, specifically, preserve R1, restore
0000 48 : saved registers correctly in error case, correct loop
0000 49 : counter for zero-based loop.
0000 50
0000 51 : V03-004 KPL0001 Peter Lieberwirth 6-Feb-1984
0000 52 : Add MMG$ALLOCONTIG, a routine to allocate physically-contiguous
0000 53 : memory.
0000 54
0000 55 : V03-003 WMC0002 Wayne Cardoza 13-Apr-1983
0000 56 : Leave type bit set for bad pages.
0000 57

```

ALLOCPFN  
V04-000

- PFN LIST MANIPULATING ROUTINES  
HISTORY ; DETAILED

G 14

15-SEP-1984 23:49:54 VAX/VMS Macro V04-00  
5-SEP-1984 03:39:55 [SYS.SRC]ALLOCPFN.MAR;1

Page 2  
(1)

AL  
VO

0000 58 :  
0000 59 :  
0000 60 :  
0000 61 :  
0000 62 :  
0000 63 :  
0000 64 :

V03-002 WMC0001 Wayne Cardoza 05-Apr-1983  
RELPFN will check for bad pages.  
V03-001 LJK0140 Lawrence J. Kenah 4-Mar-1982  
Change insertion and removal routines to take into account  
the fact that array elements for PFN 0 may not exist.

```

0000 66      .SBTTL  DECLARATIONS
0000 67      :
0000 68      : INCLUDE FILES:
0000 69      :
0000 70      $OPDEF          ; Define opcode equivalences
0000 71      $PFNDEF         ; PFN DATA BASE DEFINITIONS
0000 72      $PRIDEF         ; PRIORITY INCREMENT CLASS DEFINITIONS
0000 73      $PTEDEF         ; PAGE TABLE ENTRY DEFINITIONS
0000 74      $VADEF          ; VIRTUAL ADDRESS FIELD DEFINITIONS
0000 75      $WQHDEF         ; WAIT QUEUE HEADER DEFINITIONS
0000 76      :
0000 77      : EXTERNAL SYMBOLS:
0000 78      :
0000 79      :
0000 80      : MACROS:
0000 81      :
0000 82      :
0000 83      : EQUATED SYMBOLS:
0000 84      :
0000 85      :
0000 86      : OWN STORAGE:
0000 87      :
00000000 88      .PSECT $$$210, LONG
0000 89      :
0000 90      : PFN LISTS INDEXED BY LIST IDENTIFIER, FREE, MODIFIED, AND BAD PAGE LISTS
0000 91      :
0000 92      PFNSAL_HEAD::
0000 93      .LONG 0,0,0          ;HEAD OF PFN LISTS
00000004 000C 94      PFNSAL_MFYLSTHD==PFNSAL_HEAD+4 ;ADDRESS OF MODIFIED PAGE LIST HEAD
0000 95      PFNSAL_TAIL:
0000 96      .LONG 0,0,0          ;TAIL OF PFN LISTS
00000000 000C 97      PFNSAL_COUNT:
0000 98      .LONG 0,0,0          ;COUNT OF PFN LIST ENTRIES
00000000 0018 99      SCH$GL_FREECNT==PFNSAL_COUNT ;FREE PAGE COUNT
00000018 0024 100     SCH$GL_MFYCNT==PFNSAL_COUNT+4 ;MODIFIED PAGE COUNT
0000001C 0024 101     PFNSGL_PHYPGCNT:: ;AVAILABLE PHYSICAL PAGE COUNT
00000000 0024 102     .LONG 0 ;
0000 103     PFNSAL_HILIMIT::
00000000 0028 104     .LONG 1230,25,1230 ;"LIST TOO FULL" THRESHOLD
40000000 00000019 40000000 0028 105     SCH$GL_FREEREQ==PFNSAL_HILIMIT ;REQUIRED FREE PAGES BY SWAPPER
00000028 0034 106     SCH$GL_MFYLOLIM==PFNSAL_HILIMIT+4 ;MODIFIED PAGE LIST HILIMIT
0000002C 0034 107     PFNSAL_LOLIMIT::
00000000A 0000000A 0000000A 0034 108     .LONG 10,10,-1 ;"LIST NEARLY EMPTY" THRESHOLD
00000034 0040 109     SCH$GL_FREELOLIM==PFNSAL_LOLIMIT
00000038 0040 110     SCH$GL_MFYLOLIM==PFNSAL_LOLIMIT+4
00000019 0040 111     SCH$GL_MFYLIMSV:: ;THE MODIFIED PAGE WRITER RESTORES
0000 112     .LONG 25 ;THIS VALUE TO SCH$GL_MFYLOLIM AFTER
0000 113     ;WRITING ALL THE MODIFIED PAGES
0000000A 0044 114     SCH$GL_MFYLOSV:: ;THE MODIFIED PAGE WRITER RESTORES
0000 115     .LONG 10 ;THIS VALUE TO SCH$GL_MFYLOLIM AFTER
0000 116     ;WRITING ALL THE MODIFIED PAGES.
0000 117     :
0000 118     : *****
0000 119     :
0000 120     : ***** THIS ENTIRE MODULE MUST BE RESIDENT *****
0000 121     :
00000000 122     .PSECT $MMGCOD

```

ALLOCPFN  
V04-000

- PFN LIST MANIPULATING ROUTINES  
DECLARATIONS

I 14

15-SEP-1984 23:49:54  
5-SEP-1984 03:39:55

VAX/VMS Macro V04-00  
[SYS.SRC]ALLOCPFN.MAR;1

Page 4  
(2)

0000 123 :  
0000 124 : \*\*\*\*\*  
0000 125 :

AL  
V0

```

0000 127      .SBTTL  ALLOCPFN - ALLOCATE A PAGE FROM THE FREE PAGE LIST
0000 128      :++
0000 129      : FUNCTIONAL DESCRIPTION:
0000 130      :
0000 131      :     ALLOCATE A PFN FROM THE FRONT OF THE FREE LIST, RELEASING ITS
0000 132      : CONTENTS IF ANY.  IPL MUST BE AT SYNCH OR HIGHER.
0000 133      :
0000 134      : CALLING SEQUENCE:
0000 135      :
0000 136      :     BSBW  MMG$ALLOCPFN
0000 137      :
0000 138      : INPUT PARAMETERS:
0000 139      :
0000 140      :     IPL MUST BE AT SYNCH OR HIGHER
0000 141      :
0000 142      : IMPLICIT INPUTS:
0000 143      :     NONE
0000 144      :
0000 145      : OUTPUT PARAMETERS:
0000 146      :
0000 147      :     RO = PFN IF PAGE IS ALLOCATED OR NEGATIVE IF NONE AVAILABLE
0000 148      :     PFNSAL_PTE[RO] = 0
0000 149      :     PFNSAW_REFCNT[RO] = 0
0000 150      :
0000 151      : IMPLICIT OUTPUTS:
0000 152      :
0000 153      :     IF PFN HAD PREVIOUS CONTENTS (PFNSAL_PTE[RO] NEQ 0), THEN
0000 154      : THE PREVIOUS PTE IS TRANSFORMED FROM 'TRANSITION' TO ITS BACKUP ADDRESS.
0000 155      :
0000 156      : COMPLETION CODES:
0000 157      :     NONE
0000 158      :
0000 159      : SIDE EFFECTS:
0000 160      :     NONE
0000 161      :
0000 162      :--
0000 163      MMG$ALLOCPFN::
0000 164      ASSUME  PFNSC_FREPAGLST EQ 0
0000 165      CLRL   R2                ;FREE PAGE LIST ID
0000 166      BSBW  MMG$REMPFNH     ;GET A PAGE FROM THE HEAD
0000 167      BBC   #31,R0,20$     ;BRANCH IF GOT A PAGE
0000 168      RSB                ;OTHERWISE RETURN, NONE AVAILABLE
000A 169      :
000A 170      : MUST NOW RELEASE ANY PREVIOUS CONTENTS
000A 171      :
000A 172      20$:
000A 173      TSTW  @W*PFNSAW_REFCNT[RO] ;REF COUNT OK?
000F 174      BEQL  MMG$DELCONPFN     ;IT MUST BE ZERO AT THIS POINT
0011 175      BUG_CHECK FREEPAGREF,FATAL ;FREE PAGE REFERENCE COUNT NONZERO

```

52 D4 0000 165  
0092 30 0002 166  
01 50 1F E1 0005 167  
05 0009 168  
000A 169  
000A 170  
000A 171  
000A 172  
0000'DF40 B5 000A 173  
04 13 000F 174  
0011 175





```

0049 234 ASSUME PFNSC_GBLWRT EQ 3
0049 235 ASSUME PFNSC_PPGTBL EQ 4
0049 236 ASSUME PFNSC_GPGTBL EQ 5
0049 237
51 03 00 EF 0049 238 EXTZV #PFNSV PAGTYP,#PFNS$ PAGTYP,- ;GET PAGE TYPE TO DISPATCH ON
0000'DF40 004C 239 @W^PFNSAB_TYPE[RO],RT
0051 240 CASE R1 <-
0051 241 40$,- ;PROCESS PAGE
0051 242 50$,- ;SYSTEM PAGE
0051 243 40$,- ;GLOBAL READ ONLY
0051 244 40$,- ;GLOBAL WRITABLE
0051 245 30$,- ;PROCESS PAGE TABLE
0051 246 50$,- ;GLOBAL PAGE TABLE
0051 247 >
0061 248 :
0061 249 : ERROR IN DELCONPFN
0061 250 :
0061 251 20$: BUG_CHECK DELCONPFN,FATAL ;
0065 252 :
0065 253 : PROCESS PAGE TABLE PAGE
0065 254 :
51 53 0000'CF C3 0065 255 30$: SUBL3 W^SWP$GL_BALSPT,R3,R1 ;BYTE OFFSET INTO SPT FOR PROCESS HEADERS
51 51 0000'CF C6 006B 256 DIVL W^SWP$GL_BSLOTSZ,R1 ;PROCESS HEADER INDEX
51 51 FE 8F 78 0070 257 ASHL #-2,R1,RT ;FOR BYTE COUNT
FF88' 30 0075 258 BSBW MMG$DECPHDREF1 ;DECREMENT PROCESS HEADER REF CNT
03 11 0078 259 BRB 50$
007A 260 :
007A 261 : PROCESS OR GLOBAL PAGE, MUST REDUCE PAGE TABLE REFERENCE COUNT
007A 262 :
FF83' 30 007A 263 40$: BSBW MMG$DECPTRF ;COUNT ONE LESS REF FOR PAGE TABLE
007D 264 :
007D 265 : REINITIALIZE THE PFN DATA BASE NOW
007D 266 :
0000'DF40 D4 007D 267 50$: CLRL @W^PFNSAL_PTE[RO] ;ZERO THE PTE POINTER (PAGE EMPTY)
0000'DF40 D4 0082 268 60$: CLRL @W^PFNSAL_BAK[RO] ;ZERO THE BACKING STORE ADDRESS
0000'DF40 B4 0087 269 CLRW @W^PFNSAW_SWPVBN[RO] ;ZERO THE SWAP FILE VBN
0000'DF40 94 008C 270 CLRB @W^PFNSAB_STATE[RO] ;ZERO THE PFN STATE
0000'DF40 94 0091 271 CLRB @W^PFNSAB_TYPE[RO] ;ZERO THE PFN TYPE
0096 272 :
0096 273 : ***** THE FOLLOWING ASSUMPTIONS ARE MADE HERE
0096 274 :
0096 275 : PFNSAW_WSLX[RO] IS THE SAME AS PFNSAW_BLINK[RO]
0096 276 : PFNSAW_SHRCNT[RO] IS THE SAME AS PFNSAW_FLINK[RO]
0096 277 :
0096 278 : BOTH OF THE ABOVE ARE ZEROED BY MMG$REMPFN OR MMG$REMPFNH
0096 279 :
05 0096 280 RSB

```

AL  
Ps  
  
PS  
--  
SA  
SS  
SM  
ZS  
  
Ph  
--  
In  
Co  
Pa  
Sy  
Pa  
Sy  
Ps  
Cr  
As  
  
Th  
38  
Th  
82  
19  
  
Ma  
--  
-S  
-S  
TO  
  
62  
  
Th  
  
MA

```

0097 283 .SBTTL REMPFN - REMOVE A PFN FROM LIST
0097 284
0097 285 :++
0097 286 : FUNCTIONAL DESCRIPTION:
0097 287 :
0097 288 : REMOVE A PFN FROM THE SPECIFIED PFN LIST. THIS ROUTINE ACCEPTS
0097 289 : A PFN AND A PFN LIST IDENTIFIER AND REMOVES THE PFN FROM THE LIST.
0097 290 : IT MUST BE CALLED AT IPL GREATER OR EQUAL TO SYNCH.
0097 291 :
0097 292 : CALLING SEQUENCE:
0097 293 :
0097 294 : BSBW MMG$REMPFNH ;REMOVE PFN FROM HEAD OF LIST
0097 295 : BSBW MMG$REMPFN ;REMOVE SPECIFIED PFN FROM LIST
0097 296 :
0097 297 : INPUT PARAMETERS:
0097 298 :
0097 299 : R0 = PFN TO REMOVE (UNLESS REMPFNH)
0097 300 : R2 = PFN LIST IDENTIFIER
0097 301 :
0097 302 : IMPLICIT INPUTS:
0097 303 : NONE
0097 304 :
0097 305 : OUTPUT PARAMETERS:
0097 306 :
0097 307 : R0 = PFN REMOVED
0097 308 : OR NEGATIVE IF REMPFNH FROM EMPTY LIST
0097 309 :
0097 310 : IMPLICIT OUTPUTS:
0097 311 :
0097 312 : THE LIST HEAD, TAIL AND COUNT FOR THE SPECIFIED LIST IS UPDATED
0097 313 :
0097 314 : COMPLETION CODES:
0097 315 : NONE
0097 316 :
0097 317 : SIDE EFFECTS:
0097 318 : THESE ROUTINES ARE WRITTEN IN SUCH A WAY THAT THE FIRST ELEMENT
0097 319 : IN EACH ARRAY NEED NOT CORRESPOND TO PFN 0. THAT IS, THERE NEED NOT BE
0097 320 : ARRAY ELEMENTS CORRESPONDING TO PFN 0.
0097 321 :
0097 322 :--
0097 323
0097 324 MMG$REMPFNH:
50 0000'CF42 D0 0097 325 MOVL W^PFN$AL HEAD[R2],R0 ;R0 = PFN FROM HEAD OF LIST
03 12 009D 326 BNEQ MMG$REMPFN ;REMOVE THE PFN IF ANY
50 05 D7 009F 327 DECL R0 ;RETURN NEGATIVE IF EMPTY
00A1 328 RSB
00A2 329 :
00A2 330 : R0 = PFN TO REMOVE, R2 = LIST IDENTIFIER
00A2 331 :
00A2 332 MMG$REMPFN:
00A2 333 PFN REFERENCE -
00A2 334 MOVZWL <@W^PFN$Ax FLINK[R0],R1>,- ;R1 = NEXT PFN
00A2 335 LONG_OPCODE=MOVL,-
00A2 336 IMAGE=SYS NONPAGED
00A8 337 PFN REFERENCE -
00A8 338 MOVZWL <@W^PFN$Ax BLINK[R0],R3>,- ;R3 = PREVIOUS PFN
00A8 339 LONG_OPCODE=MOVL,-

```

```

10 13 00A8 340          IMAGE=SYS_NONPAGED
      00AE 341          BEQL 10$                ;BRANCH IF HEAD OF LIST
      00B0 342
      00B0 343
      00B0 344          MOVW <R1,@W^PFNSAx FLINK[R3]>,-      ;FLINK[PREV] = NEXT
      00B0 345          LONG OP CODE=MOVL,-                ;R3 NEQ 0
      00B0 346          IMAGE=SYS_NONPAGED
10 12 00B6 347          BNEQ 15$                ;BRANCH IF NOT TAIL
      00B8 348
      00B8 349          ; IF NEXT = 0 THEN PFN BEING REMOVED WAS THE TAIL OF THE LIST
      00B8 350
000C'CF42 53 D0 00B8 351 5$:  MOVW R3,W^PFNSAx_TAIL[R2]      ;TAIL = PREVIOUS
      OE 11 00BE 352          BRB 20$                ;JOIN COMMON CODE
      00C0 353
      00C0 354          ; IF PREVIOUS = 0 THEN PFN BEING REMOVED WAS HEAD OF THE LIST
      00C0 355
0000'CF42 51 D0 00C0 356 10$:  MOVW R1,W^PFNSAx_HEAD[R2]      ;HEAD = NEXT
      FO 13 00C6 357          BEQL 5$                ;BRANCH IF ALSO TAIL
      00C8 358
      00C8 359 15$:  MOVW <R3,@W^PFNSAx BLINK[R1]>,-      ;BLINK[NEXT] = PREVIOUS
      00C8 360          LONG OP CODE=MOVL,-                ;R1 NEQ 0
      00C8 361          IMAGE=SYS_NONPAGED
      00C8 362
      00CE 363 20$:  SOBGR W^PFNSAx_COUNT[R2],30$      ;ONE LESS PFN, BRANCH IF NOT EMPTY
09 0018'CF42 F5 00CE 364
      00D4 365          ; CHECK CONSISTENCY OF THE COUNT AND LIST HEAD OF EMPTY LIST
      00D4 366          ;
      00D4 367          ;
      53 51 C8 00D4 368          BISL R1,R3                ;BOTH HEAD AND TAIL 0?
      04 13 00D7 369          BEQL 30$                ;BRANCH IF OK
      00D9 370          BUG_CHECK PFNLISTCNT,FATAL      ;INCONSISTENT PFN LIST COUNT
      00DD 371 30$:
      00DD 372
      00DD 373          CLRW <@W^PFNSAx FLINK[R0]>,-      ;ZERO THE NEW PAGE'S FLINK
      00DD 374          LONG OP CODE=CLRL,-
      00DD 375          IMAGE=SYS_NONPAGED
      00E2 376          CLRW <@W^PFNSAx BLINK[R0]>,-      ;ZERO THE NEW PAGE'S BLINK
      00E2 377          LONG OP CODE=CLRL,-
      00E2 378          IMAGE=SYS_NONPAGED
0034'CF42 0018'CF42 D1 00E2 379
      06 14 00E7 380          CMLP W^PFNSAx_COUNT[R2],W^PFNSAx_LOLIMIT[R2] ;NEARLY EMPTY?
      00F0 381          BGTR 40$                ;BRANCH IF NOT
      00F2 382          ; THIS LIST IS BELOW ITS THRESHOLD SIZE, INFORM THE INTERESTED PARTY
      00F2 383          ;
      00F2 384          ;
      00F2 385          CASE R2,<-
      00F2 386          WAKESWAPPER-                ;FREE LIST - INFORM SWAPPER
      00F2 387          >
      00F8 388 40$:
      05 00F8 389          RSB

```

```

00F9 391 .SBTTL RLPFNSAVPTE - RELEASE PFN SAVING PAGE TABLE ENTRY
00F9 392 :++
00F9 393 : FUNCTIONAL DESCRIPTION:
00F9 394 :
00F9 395 : THIS ROUTINE RELEASES THE GIVEN PFN BUT LEAVES THE CONTENTS OF
00F9 396 : THE PTE ADDRESSED BY R3 ALONE. IT IS USED FOR RELEASING THE PFN
00F9 397 : FOR A GLOBAL DZRO PAGE WHEN 2 OR MORE PROCESSES TRIED TO ZERO THE
00F9 398 : PAGE. ONE MAKES IT, THE OTHERS RELEASE THE PFN AND START OVER.
00F9 399 :
00F9 400 : CALLING SEQUENCE:
00F9 401 :
00F9 402 : BSBW MMG$RLPFNSAVPTE
00F9 403 :
00F9 404 : INPUT PARAMETERS:
00F9 405 :
00F9 406 : R0 = PFN
00F9 407 : R3 = SYSTEM VIRTUAL ADDRESS OF PAGE TABLE ENTRY
00F9 408 :
00F9 409 : IMPLICIT INPUTS:
00F9 410 :
00F9 411 : NONE
00F9 412 :
00F9 413 : OUTPUT PARAMETERS:
00F9 414 :
00F9 415 : NONE
00F9 416 :
00F9 417 : IMPLICIT OUTPUTS:
00F9 418 :
00F9 419 : NONE
00F9 420 :
00F9 421 : COMPLETION CODES:
00F9 422 :
00F9 423 : NONE
00F9 424 :
00F9 425 : SIDE EFFECTS:
00F9 426 :
00F9 427 : NONE
00F9 428 :
00F9 429 : --
00F9 430 :
00F9 431 MMG$RLPFNSAVPTE::
0000'DF40 D4 00F9 432 CLR  @W^PFNSAL_PTE[R0] ;NO PTE BACK POINTER
0000'DF40 B4 00FE 433 CLR  @W^PFNSAW_REFCNT[R0] ;AND NO REFERENCES
17 90 0103 434 MOVB #<PFNSC ACTIVE ! PFNSM_DELCON>,-
0000'DF40 0105 435 @W^PFNSAB_STATE[R0] ;SHUT OFF MODIFY, TURN ON DELCON
0109 436 :
0109 437 : THE FOLLOWING ASSUMES THAT THE PTE IS A GLOBAL PTE THAT MUST HAVE
0109 438 : A REFERENCE COUNT TAKEN AWAY EVEN THOUGH ITS CONTENTS ARE NOT TO
0109 439 : BE OVERWRITTEN.
0109 440 :
FEF4' 30 0109 441 BSBW MMG$DECPTREF ;COUNT ONE LESS PAGE TABLE REFERENCE
08 11 010C 442 BRB MMG$RELPFN ;RELEASE THE PFN
010E 443 ;AND RETURN TO CALLER
  
```

```

010E 445 .SBTTL DELPFNLIST - DELETE PFN FROM PFN LIST
010E 446 :++
010E 447 : FUNCTIONAL DESCRIPTION:
010E 448 :
010E 449 : THIS ROUTINE REMOVES THE PFN FROM THE SPECIFIED LIST
010E 450 : SETS DELCON AND RELEASES THE PAGE.
010E 451 :
010E 452 : CALLING SEQUENCE:
010E 453 :
010E 454 : BSBW MMG$DELPFNLIST
010E 455 :
010E 456 : INPUT PARAMETERS:
010E 457 :
010E 458 : R0 = PFN
010E 459 : R2 = LIST IDENTIFIER
010E 460 :
010E 461 : IMPLICIT INPUTS:
010E 462 :
010E 463 : NONE
010E 464 :
010E 465 : OUTPUT PARAMETERS:
010E 466 :
010E 467 : NONE
010E 468 :
010E 469 : IMPLICIT OUTPUTS:
010E 470 :
010E 471 : NONE
010E 472 :
010E 473 : COMPLETION CODES:
010E 474 :
010E 475 : NONE
010E 476 :
010E 477 : SIDE EFFECTS:
010E 478 :
010E 479 : NONE
010E 480 :
010E 481 :--
010E 482 :
010E 483 MMG$DELPFNLIST::
010E 484 BSBW MMG$REMPFN ;REMOVE PFN FROM SPECIFIED LIST
0110 485 BISB #PFNSM_DELCON,@W^PFNSAB_STATE[R0] ;JAM 'DELETE CONTENTS' BIT
0116 486 :
0116 487 : FALL THROUGH TO MMG$RELPFN
0116 488 :

```

0000'DF40 92 10  
10 88

0116 490 .SBTTL RELPFN - RELEASE PFN TO FREE OR MODIFY LIST

0116 491 :++  
0116 492 : FUNCTIONAL DESCRIPTION:

0116 493 :  
0116 494 : THIS ROUTINE RELEASES THE SPECIFIED PFN TO THE MODIFY OR  
0116 495 : FREE PAGE LIST ACCORDING TO THE STATE BITS IN PFNSAB\_STATE[R0]  
0116 496 :  
0116 497 : CALLING SEQUENCE:

0116 498 :  
0116 499 : BSBW MMG\$RELPFN  
0116 500 :  
0116 501 : INPUT PARAMETERS:

0116 502 :  
0116 503 : R0 = PFN, REFCNT=0  
0116 504 :  
0116 505 : IMPLICIT INPUTS:

0116 506 : NONE  
0116 507 :  
0116 508 : OUTPUT PARAMETERS:

0116 509 :  
0116 510 : R0 PRESEVED  
0116 511 :  
0116 512 : IMPLICIT OUTPUTS:

0116 513 : NONE  
0116 514 :  
0116 515 : COMPLETION CODES:

0116 516 : NONE  
0116 517 :  
0116 518 : SIDE EFFECTS:

0116 519 : NONE  
0116 520 :  
0116 521 : --

0116 522 : MMG\$RELPFN::

0116 523 : MOVL @W^PFNSAL\_PTE[R0],R3 ;R3 = SVAPTE  
0116 524 : BEQL 10\$ ;BRANCH IF PAGE IS EMPTY  
0116 525 : BITW #<PTESM\_VALID ! PTESM\_TYP1 ! PTESM\_TYPO>@-16,2(R3)  
0116 526 : BNEQ 60\$ ;BRANCH IF NOT TRANSITION PAGE  
52 0000'DF40 01 07 EF 0126 527 10\$: EX?ZV #PFNSV\_MODIFY,#1,@W^PFNSAB\_STATE[R0],R2 ;R2 = MODIFY BIT  
0116 528 : BNEQ 40\$ ;BRANCH IF PAGE MODIFIED

0130 529 :  
0130 530 : PAGE NOT MODIFIED R2 = 0 = LIST ID FOR FREE PAGE LIST  
0130 531 : SEE IF SUPPOSED TO DELETE THE CONTENTS OF THE PAGE  
0130 532 :

17 0000'DF40 04 E1 0130 533 : BBC #PFNSV\_DELCON,@W^PFNSAB\_STATE[R0],50\$ ;BRANCH IF RETAIN CONTENTS  
29 0000'DF40 05 E0 0137 534 : BBS #PFNSV\_BADPAG,@W^PFNSAB\_TYPE[R0],90\$ ;GO PUT ON BAD PAGE LIST  
FED4 30 013E 535 : BSBW MMG\$DE[CONPFN ;DELETE PAGE CONTENTS  
0141 536 : ASSUME PFNSC\_FREPAGLST EQ 0  
52 D4 0141 537 : CLRL R2 ;FREE PAGE LIST ID  
59 10 0143 538 : BSBB MMG\$INSPFNH ;PUT PAGE ON FRONT OF FREE LIST  
05 0145 539 : RSB

51 0000'DF40 D0 0146 540 40\$:  
0146 541 : MOVL @W^PFNSAL\_BAK[R0],R1 ;GET BACKING STORE CELL  
28 10 014C 542 : BSBB MMG\$DALCBKSTORE ;FREE THE BACKING STORE PAGE

0088 30 014E 543 50\$:  
014E 544 : BSBW MMG\$INSPFNT ;PUT PAGE AT END OF LIST IN R2  
05 0151 545 : RSB  
0152 546 :

```

0152 547 : PAGE IS NOT A TRANSITION PAGE, IF IT'S NOT VALID, THAT'S A FATAL ERROR.
0152 548 : IF VALID, THEN FOLD MODIFY BACK TO PFN DATA, SHUT OFF VALID AND MODIFY,
0152 549 : AND CONTINUE WITH RELEASING THE PAGE.
0152 550 :
0152 551 60$: BGTR 80$ ;BRANCH IF PTE NOT VALID
0154 552 BBCC #PTESV_MODIFY,(R3),70$ ;CLEAR MODIFY, BRANCH IF WAS CLEAR
0158 553 BISB #PFNSM_MODIFY,@W^PFNSAB_STATE[R0] ;RECORD IT IN PFN DATA
015F 554 70$: BBSC #PTESV_VALID,(R3),10$ ;CLEAR VALID AND BRANCH (IT WAS SET)
0163 555 80$: BUG_CHECK PAGNTRNVAL,FATAL ;PAGE NOT IN TRANSITION OR VALID
0167 556 :
0167 557 : PAGE IS MARKED BAD SO IT WILL BE PUT ON BAD PAGE LIST
0167 558 :
0167 559 90$: BSBW MMG$DELCONPFN ;DELETE PAGE CONTENTS
016A 560 BISB #PFNSM_BADPAG,@W^PFNSAB_TYPE[R0] ;RESET THE BIT TO HELP TRACK THEM
0170 561 MOVL #PFNSC_BADPAGLST,R2
0173 562 BSBB MMG$INSPFNT ;PUT IT AT END OF LIST
0175 563 RSB
0176 564 :
0176 565 : SUBROUTINE TO FREE VBN'S TO PAGING FILES.
0176 566 : R0,P2 PRESERVED
0176 567 : R1,R3 DESTROYED
0176 568 :
0176 569 MMG$DALCBAKSTORE::
0176 570 TSTW @W^PFNSAW_SWPVBN[R0] ;CHECK FOR SWAP FILE BACKING STORE
0178 571 BNEQ 50$ ;BRANCH IF YES, PAGE MUST KEEP BAK ADDR
017D 572 EXTV #PFNSV_BAK,#PFNSS_BAK,R1,R3 ;BACKING STORE VBN VS PROCESS PTE
0182 573 BLEQ 50$ ;BRANCH IF NOT, NO BLOCK TO RELEASE
0184 574 PUSHF #M<R0,R2> ;SAVE PFN, MODIFY LIST DESTINATION
0186 575 MOVL R3,R0 ;SET UP VBN FOR MMG$DALCPAGFIL
0189 576 ASHL #-PFNSV_PGFLX,R1,R3 ;SET PAGING FILE TO RELEASE BLOCK TO
018E 577 BSBB MMG$DALCPAGFIL ;FREE THE BLOCK
0191 578 POPR #M<R0,R2> ;RESTORE PFN, MODIFY LIST INDEX
0193 579 BICL #PFNSM_BAK,@W^PFNSAL_BAK[R0] ;CLEAR OUT BACKING STORE ADDRESS
019D 580 50$: RSB ;RETURN

```

07 63 OF 14  
0000'DF40 80 8F 88  
C3 63 1F E4

FEAB 30  
0000'DF40 20 88  
52 02 D0  
67 10 0173  
05 0175

0000'DF40 B5  
53 51 17 00 EE  
19 15 0182  
05 BB 0184  
50 53 D0 0186  
53 51 E8 8F 78 0189  
FE6F' 30 018E  
05 BA 0191

0000'DF40 007FFFFFFF 8F CA 0193  
05 019D



```

019E 582 .SBTTL INSPFN - INSERT PFN AT HEAD OR TAIL OF PFN LIST
019E 583
019E 584 :++
019E 585 : FUNCTIONAL DESCRIPTION:
019E 586 :
019E 587 : INSERT PFN ON SPECIFIED PFN LIST, SETTING THE LOCATION FIELD
019E 588 : IN PFNSAB_STATE TO REFLECT THE LIST IDENTIFIER. IPL MUST BE AT SYNCH
019E 589 : OR HIGHER. IF THE LIST WAS EMPTY BEFORE THE INSERT SPECIAL ACTION
019E 590 : IS TAKEN TO INFORM ANY PROCESS THAT MIGHT BE WAITING.
019E 591
019E 592 : CALLING SEQUENCE:
019E 593 :
019E 594 : BSBW MMGSDALLOCPFN ;DEALLOCATE PFN, PUT AT TAIL OF FREE LIST
019E 595 : BSBW MMG$INSPFNT ;INSERT PFN AT TAIL OF LIST SPECIFIED IN R2
019E 596 : BSBW MMG$INSPFNH ;INSERT PFN AT HEAD OF LIST SPECIFIED IN R2
019E 597
019E 598
019E 599 : INPUT PARAMETERS:
019E 600 :
019E 601 : R0 = PFN TO INSERT IN THE LIST
019E 602 : R2 = PFN LIST IDENTIFIER (UNLESS DALLOCPFN)
019E 603
019E 604 : IMPLICIT INPUTS:
019E 605 : NONE
019E 606
019E 607 : OUTPUT PARAMETERS:
019E 608 :
019E 609 : R0 PRESERVED
019E 610
019E 611 : IMPLICIT OUTPUTS:
019E 612 :
019E 613 : PFN LIST HEAD, TAIL AND COUNT ARE UPDATED
019E 614
019E 615 : COMPLETION CODES:
019E 616 : NONE
019E 617
019E 618 : SIDE EFFECTS:
019E 619 :
019E 620 : IF THE LIST WAS EMPTY BEFORE THE INSERT, THEN IF THE LIST IS
019E 621 : THE FREE PAGE LIST, ANYONE WAITING FOR FREE PAGES IS AWAKENED. IF
019E 622 : THE LIST IS THE MODIFIED PAGE LIST, THE MODIFIED PAGE WRITER IS INFORMED.
019E 623
019E 624 : THESE ROUTINES ARE WRITTEN IN SUCH A WAY THAT THE FIRST ELEMENT
019E 625 : IN EACH ARRAY NEED NOT CORRESPOND TO PFN 0. THAT IS, THERE NEED NOT BE
019E 626 : ARRAY ELEMENTS CORRESPONDING TO PFN 0.
019E 627
019E 628 :--
019E 629
019E 630 MMG$INSPFNH::
019E 631 TSTW @W^PFNSAW_REFcnt[R0] ;INSERT PFN AT HEAD OF LIST
019E 632 BNEQ INSPFNREF- ;REFERENCE COUNT MUST BE ZERO
019E 633 INSV R2,#PFNSV_LOC,#PFNSS_LOC,@W^PFNSAB_STATE[R0] ;SET LOCATION
019E 634 MOVL W^PFNSAL_READ[R2],R1 ;R1=PFN OF OLD HEAD
019E 635 BEQL 10$ ;SKIP NEXT IF LIST PREVIOUSLY EMPTY
019E 636 PFN_REFERENCE -
019E 637 MOVW <R0,@W^PFNSAx,BLINK[R1]>,- ;BLINK(OLD) = NEW
019E 638 LONG_OPCODE=MOVL,-

```

```

0000'DF40 B5
          31 12
0000'DF40 03 00 52 FO
          51 0000'CF42 D0
          06 13

```

```

0000'CF42  50  D0  01B5  639      MOVL  IMAGE=SYS NONPAGED
                                RO,W^PFNSAL_HEAD[R2] ;HEAD = NEW
                                PFN REFERENCE -
                                <@W^PFNSAx BLINK[RO]>,- ;BLINK(NEW)=0
                                LONG OPCODE=CLRL,-
                                IMAGE=SYS NONPAGED
                                PFN REFERENCE -
                                <R1,@W^PFNSAx FLINK[RO]>,- ;FLINK(NEW)=OLD
                                LONG OPCODE=MOVL,-
                                IMAGE=SYS_NONPAGED
0000'CF42  51  12  01C  649      BNEQ  NOTEMPTY ;BRANCH IF LIST WAS NOT EMPTY
                                50  D0  01CE  650      MOVL  RO,W^PFNSAL_TAIL[R2] ;LIST NOW HAS JUST ONE ENTRY
                                3C  11  01D4  651      BRB   WASEMPTY ;INFORM OTHERS, LIST NO LONGER EMPTY
                                01D6  652      :
                                01D6  653      : PFN REFERENCE COUNT NON-ZERO
                                01D6  654      :
                                01D6  655      INSPFNREF:
                                01D6  656      BUG_CHECK PFNREFNZRO,FATAL ;PFN REFERENCE COUNT NONZERO
                                01DA  657
                                01DA  658 MMGSDALLOCPFN::
                                52  D4  01DA  659      ASSUME PFNSC_FREPAGLST EQ 0
                                01DA  660      CLRL  R2 ;FREE PAGE LIST ID
                                01DC  661 MMG$INSPFN:: ;INSERT PFN AT TAIL OF LIST
                                0000'DF40  B5  01DC  662      TSTW  @W^PFNSAx_W_REFCNT[RO] ;MAKE SURE REFERENCE COUNT IS ZERO
                                F3  12  01E1  663      BNEQ  INSPFNREF- ;BRANCH IF IT IS NOT, ERROR
                                0000'DF40  03  00  52  F0  01E3  664      INSV  R2,#PFNSV_LOC,#PFNS$-LOC,@W^PFNSAx_STATE[RO] ;SET LOCATION
                                51  0000'CF42  D0  01EB  665      MOVL  W^PFNSAL_TAIL[R2],R1 ;R1=PFN OF LAST ENTRY ON LIST
                                06  13  01F1  666      BEQL  20$ ;SKIP NEXT IF LIST PREVIOUSLY EMPTY
                                01F3  667      PFN REFERENCE -
                                01F3  668      MOVW  <RO,@W^PFNSAx FLINK[R1]>,- ;FLINK(OLD)=NEW
                                01F3  669      LONG OPCODE=MOVL,-
                                0000'CF42  50  D0  01F9  671  20$: MOVL  RO,W^PFNSAL_TAIL[R2] ;NEW ENTRY IS THE NEW TAIL
                                01FF  672      PFN REFERENCE -
                                01FF  673      CLRW  <@W^PFNSAx FLINK[RO]>,- ;NEW ENTRY FLINK IS 0
                                01FF  674      LONG OPCODE=CLRL,-
                                01FF  675      IMAGE=SYS NONPAGED
                                0204  676      PFN REFERENCE -
                                0204  677      MOVW  <R1,@W^PFNSAx BLINK[RO]>,- ;NEW ENTRY BLINK IS OLD TAIL
                                0204  678      LONG OPCODE=MOVL,-
                                0204  679      IMAGE=SYS_NONPAGED
                                0000'CF42  13  12  020A  680      BNEQ  NOTEMPTY ;BRANCH IF LIST WAS NOT EMPTY
                                50  D0  020C  681      MOVL  RO,W^PFNSAL_HEAD[R2] ;LIST WAS EMPTY, NEW ENTRY IS HEAD AND TAIL
                                0212  682      :
                                0212  683      : THIS LIST JUST WENT FROM EMPTY TO ONE ENTRY.
                                0212  684      :
                                0212  685      WASEMPTY:
                                0018'CF42  01  D0  0212  686      MOVL  #1,W^PFNSAL_COUNT[R2] ;RESET THE LIST COUNTER
                                0218  687      CASE  R2,<-
                                0218  688      WAKFREPAGEWAITQ- ;FREE PAGE LIST WAS EMPTY
                                0218  689      >
                                05  021E  690      RSB
                                021F  691
                                08  0018'CF42  0028'CF42  F2  021F  692      NOTEMPTY:
                                021F  693      AOBLSS W^PFNSAL_HILIMIT[R2],W^PFNSAL_COUNT[R2],20$
                                0229  694      :
                                0229  695      : THIS LIST IS ABOVE ITS THRESHOLD SIZE, INFORM THE INTERESTED PARTY

```

```

0229 696 ;
0229 697 ; CASE R2,<-
0229 698 WAKESWAPPER,- ;FREE PAGE LIST
0229 699 WAKE_MPW- ;MODIFIED PAGE LIST
0229 700 >
05 0231 701 20$:
0231 702 RSB
0232 703
0232 704 ;
0232 705 ; FREE PAGE LIST JUST WENT FROM EMPTY TO ONE PAGE. MUST REPORT
0232 706 ; THE "FREE PAGE AVAILABLE" EVENT FOR ALL PROCESSES WAITING.
0232 707 ;
0232 708 WAKFREPAGEWAITQ:
52 11 BB 0232 709 PUSHF #^M<R0,R4> ;SAVE THE REGISTERS TO BE RETURNED
02 02 DO 0234 710 MOVL #PRIS_RESAVL,R2 ;"RESOURCE AVAILABLE" PRIORITY INC
0008'CF B5 0237 711 10$:
10 15 023B 712 TSTW W^SCH$GQ_FPGWQ+WQHSW_WQCNT ;ANYONE WAITING?
0000'CF 01' 88 023D 713 BLEQ 20$ ;BRANCH IF NOT
54 0000'CF DO 0242 714 BISB S^#<1@SCH$V REORD>,W^SCH$GB RESCAN ;SET FLAG FOR OSWPSCHED
EA 11 0247 715 MOVL W^SCH$GQ_FPGWQ,R4 ;GET THE NEXT PCB
11 BA 024B 716 RPTFVT FPGA ;REPORT "FREE PAGE AVAILABLE"
05 024D 717 BRB 10$
024F 718 20$: POPR #^M<R0,R4> ;RESTORE REGISTERS
0250 719 RSB
0250 720 ;
0250 721 ; NEED TO START WRITING MODIFIED PAGES
0250 722 ;
06 0000'CF 00' E1 0250 723 WAKE_MPW:
0256 724 BBC S^#SCH$V_MPW,W^SCH$GB_SIP,WAKESWAPPER ;BRANCH IF MODIFIED PAGE
002F'CF 01 90 0256 725 ;WRITER IS NOT ALREADY ACTIVE
025B 726 MOVB #1,W^SCH$GL_MFYLM+3 ;DISABLE THE THRESHOLD CHECK WHILE
05 025B 727 ;MODIFIED PAGE WRITER IS ACTIVE
025C 728 RSB
025C 729 ;
025C 730 ; INFORM THE SWAPPER PROCESS THAT SOMETHING OF INTEREST HAS OCCURRED
025C 731 ;
FDA1' 30 025C 732 WAKESWAPPER:
05 025C 733 BSBW SCH$SWPWAKE ;WAKE UP THE SWAPPER PROCESS
025F 734 RSB

```

```

0260 736 .SBTTL ALLOCONTIG - Allocate N Physically Contiguous Pages
0260 737 :++
0260 738 : FUNCTIONAL DESCRIPTION:
0260 739 :
0260 740 : Find N physically-contiguous pages. Do this via a sequential
0260 741 : scan through the PFN data base looking for free pages. Pages cannot
0260 742 : be deallocated.
0260 743 :
0260 744 : CALLING SEQUENCE:
0260 745 :
0260 746 : BSBW MMGSALLOCONTIG
0260 747 :
0260 748 : INPUT PARAMETERS:
0260 749 :
0260 750 : IPL must be at SYNCH
0260 751 : R1 = number of pages that must be physically contiguous (range size)
0260 752 :
0260 753 : IMPLICIT INPUTS:
0260 754 :
0260 755 : None.
0260 756 :
0260 757 : OUTPUT PARAMETERS:
0260 758 :
0260 759 : R0 = First PFN in range if found or negative if no range found
0260 760 :
0260 761 : IMPLICIT OUTPUTS:
0260 762 :
0260 763 : MMGSDELCONPFN is called for each PFN in range, if found.
0260 764 : R2 contents destroyed.
0260 765 :
0260 766 : COMPLETION CODES, SIDE EFFECTS:
0260 767 :
0260 768 : None.
0260 769 :
0260 770 :--
0260 771 MMGSALLOCONTIG::
1A BB 0260 772 PUSH R1,R3,R4 : save work registers:
0262 773 : r2 = local maxpfn
0262 774 : r3 = PFN index for trial range
0262 775 : r4 = local maxpfn for trial range
52 00000000'GF 50 D4 0262 776 CLRL R0 : r0 = PFN index for main loop
51 C3 0264 777 SUBL3 R1,G*MMGSGL_MAXPFN,R2 : calculate local maxpfn - stop search
026C 778 : at MAXPFN-<trial range>
0000'DF40 95 026C 779 10$: TSTB @W*PFN$AB_STATE[R0] : is this PFN free?
0A 13 0271 780 BEQL 20$ : if EQL yes, go search trial range
F5 50 52 F3 0273 7 1 15$: ANBLEQ R2,R0,10$ : keep looking
50 01 CE 0277 7 2 : indicate failure
1A BA 027A 783 POPR #M<R1,R3,R4> : restore work registers
05 05 027C 784 RSB :
027D 785 :
027D 786 : Found one free PFN, now look if there are N consecutive free PFNs
027D 787 :
027D 788 20$: : search trial range
54 53 50 D0 027D 789 MOVL R0,R3 : copy beginning of range
51 53 C1 0280 790 ADDL3 R3,R1,R4 : success if PFNs from [r3] to [r4]
54 D7 0284 791 DECL R4 : are free, r4 = loop limit
07 11 0286 792 BRB 40$ : skip into middle of next loop

```

```

0288 793 :
0288 794 : This loop looks scans thx N pages from the first free PFN. If all N are
0288 795 : free we've found N physically-contiguous pages.
0288 796 :
0288 797 30$:
0000'DF43 95 0288 798 TSTB @W^PFNSAB_STATE[R3] ; scan range
27 12 0288 799 BNEQ 60$ ; is this PFN free?
028F 800 40$: PFN REFERENCE - ; if NEQ no, get out of this loop
028F 801 TSTW <@W^PFNSAx FLINK[R3]>,- ; make sure link nonzero
028F 802 LONG_OPCODE=TSTL -
028F 803 IMAGE=SYS_NONPAGED
EE 53 20 13 0294 804 BEQL 60$ ; if EQL no, get out of this loop
54 F3 0296 805 AOBLEQ R4,R3,30$ ; keep looking N times
029A 806 :
029A 807 : Fell through AOBLEQ, means all N pages were free. Now, DELCONPFN each page.
029A 808 :
7E 50 D0 029A 809 MOVL R0, -(SP) ; save beginning of range
0000'DF40 B5 029D 810 50$: TSTW @W^PFNSAx_REFCNT[R0] ; consistency check
17 12 02A2 811 BNEQ 70$ ; bug if free page has non-zero refcnt
FD6E 30 02A4 812 BSBW MMGSDELCONPFN ; delete PFN contents
52 D4 02A7 813 CLRL R2 ; indicate free list to REMPFN
FDF6 30 02A9 814 BSBW MMGSREMPFN ; remove PFN from free list
ED 50 54 F3 02AC 815 AOBLEQ R4,R0,50$ ; delconpfn N pages, limit still good
50 8E D0 02B0 816 MOVL (SP)+,R0 ; restore beginning of range
1A BA 02B3 817 POPR #^M<R1,R3,R4> ; restore work registers
OS 05 02B5 818 RSB
02B6 819 :
02B6 820 : Search through trial range failed. N consecutive pages were not free.
02B6 821 :
50 53 D0 02B6 822 60$: MOVL R3,R0 ; reset main loop index
B8 11 02B9 823 BRB 15$ ; and join main loop.
02BB 824 :
02BB 825 70$: BUG_CHECK FREEPAGREF,FATAL ; free page reference count nonzero
02BF 826 :
02BF 827 :
02BF 828 .END

```

ALLOCPFN  
Symbol table

- PFN LIST MANIPULATING ROUTINES K 15

15-SEP-1984 23:49:54 VAX/VMS Macro V04-00  
5-SEP-1984 03:39:55 [SYS.SRC]ALLOCPFN.MAR;1

```

...PFN
BUGS_DELCONPFN = 0000028F R 03
BUGS_FREEPAGREF ***** X 03
BUGS_PAGNTRNVAL ***** X 03
BUGS_PFNLISTCNT ***** X 03
BUGS_PFNREFNZRO ***** X 03
EVT$FPGA ***** X 03
INSPFNREF 000001D6 R 03
MMGSALLOCONTIG 00000260 RG 03
MMGSALLOCPFN 00000000 RG 03
MMGSDALCBKSTORE 00000176 RG 03
MMGSDALCPAGFIL ***** X 03
MMGSDALLOCPFN 000001DA RG 03
MMGSDECPHDREF1 ***** X 03
MMGSDECPTRF ***** X 03
MMGSDELCONPFN 00000015 RG 03
MMGSDELPFNLIST 0000010E RG 03
MMGSGL_MAXPFN ***** X 03
MMGSINSPFNH 0000019E RG 03
MMGSINSPFNT 000001DC RG 03
MMGSRELPFN 00000116 RG 03
MMGSREMPFN 000000A2 RG 03
MMGSREMPFNH 00000097 RG 03
MMGSRLPFNSAVPTE 000000F9 RG 03
NOTEPTY 0000021F R 03
OPS_CLRL = 000000D4
OPS_CLRW = 000000B4
OPS_MOVL = 000000D0
OPS_MOVW = 000000B0
OPS_MOVZWL = 0000003C
OPS_TSTL = 000000D5
OPS_TSTW = 000000B5
PFNSAB_STATE ***** X 03
PFNSAB_TYPE ***** X 03
PFNSAL_BAK ***** X 03
PFNSAL_COUNT 00000018 R 02
PFNSAL_HEAD 00000000 RG 02
PFNSAL_HILIMIT 00000028 RG 02
PFNSAL_LOLIMIT 00000034 RG 02
PFNSAL_MFYLISTHD = 00000004 RG 02
PFNSAL_PTE ***** X 03
PFNSAL_TAIL 0000000C R 02
PFNSAW_REFCNT ***** X 03
PFNSAW_SWPVBN ***** X 03
PFNSAX_BLINK ***** X 03
PFNSAX_FLINK ***** X 03
PFNSC_ACTIVE = 00000007
PFNSC_BADPAGLST = 00000002
PFNSC_FREPAGLST = 00000000
PFNSC_GBLWRT = 00000003
PFNSC_GLOBAL = 00000002
PFNSC_GPGTBL = 00000005
PFNSC_PPGTBL = 00000004
PFNSC_PROCESS = 00000000
PFNSC_SYSTEM = 00000001
PFNSGL_PHYPGCNT 00000024 RG 02
PFNSM_BADPAG = 00000020

```

```

PFNSM_BAK
PFNSM_DELCON
PFNSM_GBLBAK
PFNSM_MODIFY
PFNSS_BAK
PFNSS_LOC
PFNSS_PAGTYP
PFNSV_BADPAG
PFNSV_BAK
PFNSV_DELCON
PFNSV_GBLBAK
PFNSV_LOC
PFNSV_MODIFY
PFNSV_PAGTYP
PFNSV_PGFLX
PRIS_RESAVL
PTESM_OWN
PTESM_PFN
PTESM_PROT
PTESM_TYPO
PTESM_TYP1
PTESM_VALID
PTESV_MODIFY
PTESV_TYP1
PTESV_VALID
SCH$GB_RESCAN
SCH$GB_SIP
SCH$GL_FREECNT
SCH$GL_FREELIM
SCH$GL_FREEREQ
SCH$GL_MFYCNT
SCH$GL_MFYLIM
SCH$GL_MFYLIMSV
SCH$GL_MFYLOLIM
SCH$GL_MFYLOSV
SCH$GQ_FPGWQ
SCH$RSE
SCH$SWPWAKE
SCH$V_MPW
SCH$V_REORD
SWP$GL_BALSPT
SWP$GL_BSL0TSZ
WAKESWAPPER
WAKE_MPW
WAKFREPAGWAITQ
WASEMPTY
WQHSW_WQCNT

```

```

= 007FFFFFFF
= 00000010
= 00800000
= 00000080
= 00000017
= 00000003
= 00000003
= 00000005
= 00000000
= 00000004
= 00000017
= 00000000
= 00000007
= 00000000
= 00000018
= 00000002
= 01800000
= 001FFFFFFF
= 78000000
= 00400000
= 04000000
= 80000000
= 0000001A
= 0000001A
= 0000001F
***** X 03
***** X 03
= 00000018 RG 02
= 00000034 RG 02
= 00000028 RG 02
= 0000001C RG 02
= 0000002C RG 02
= 00000040 RG 02
= 00000038 RG 02
00000044 RG 02
***** X 03
***** X 03
***** X 03
***** X 03
***** X 03
***** X 03
0000025C R 03
00000250 R R 03
00000232 R R 03
00000212 R 03
= 00000008

```

-----  
! Psect synopsis !  
-----

PSECT name	Allocation	PSECT No.	Attributes
. ABS	00000000 ( 0.)	00 ( 0.)	NOPIC USR CON ABS LCL NOSHR NOEXE NORD NOWRT NOVEC BYTE
\$AB\$\$	00000000 ( 0.)	01 ( 1.)	NOPIC USR CON ABS LCL NOSHR EXE RD WRT NOVEC BYTE
\$\$\$210	00000048 ( 72.)	02 ( 2.)	NOPIC USR CON REL LCL NOSHR EXE RD WRT NOVEC LONG
\$MMGCOD	000002BF ( 703.)	03 ( 3.)	NOPIC USR CON REL LCL NOSHR EXE RD WRT NOVEC BYTE
Z\$INIT\$PFN_FIXUP_TABLE	0000004E ( 78.)	04 ( 4.)	NOPIC USR CC'V REL LCL NOSHR EXE RD WRT NOVEC BYTE

-----  
! Performance indicators !  
-----

Phase	Page faults	CPU Time	Elapsed Time
Initialization	41	00:00:00.05	00:00:02.03
Command processing	145	00:00:00.52	00:00:04.60
Pass 1	238	00:00:06.59	00:00:21.96
Symbol table sort	0	00:00:00.78	00:00:02.84
Pass 2	151	00:00:02.10	00:00:07.14
Symbol table output	13	00:00:00.09	00:00:00.34
Psect synopsis output	2	00:00:00.03	00:00:00.34
Cross-reference output	0	00:00:00.00	00:00:00.00
Assembler run totals	592	00:00:10.19	00:00:39.28

The working set limit was 1350 pages.  
38901 bytes (76 pages) of virtual memory were used to buffer the intermediate code.  
There were 30 pages of symbol table space allocated to hold 543 non-local and 42 local symbols.  
828 source lines were read in Pass 1, producing 22 object records in Pass 2.  
19 pages of virtual memory were used to define 17 macros.

-----  
! Macro library statistics !  
-----

Macro library name	Macros defined
_\$255\$DUA28:[SYS.OBJ]LIB.MLB;1	9
-\$255\$DUA28:[SYSLIB]STARLET.MLB;2	5
TOTALS (all libraries)	14

623 GETS were required to define 14 macros.

There were no errors, warnings or information messages.

MACRO/LIS=LIS\$:ALLOCPFN/OBJ=OBJ\$:ALLOCPFN MSRCS\$:ALLOCPFN/UPDATE=(ENHS\$:ALLOCPFN)+EXECMLS\$/LIB



This image displays a grid of 100 small, illegible document thumbnails arranged in 10 rows and 10 columns. The thumbnails are too small to read, but several larger, legible text elements are visible within the grid:

- CMODSDSP MAR**: Located in the middle-left section of the grid.
- ALLOCPFN LIS**: Located in the upper-right section of the grid.
- ACCOUNT LIS**: Located in the middle-right section of the grid.
- SYSMAR MAR**: Located in the lower-middle section of the grid.
- SYSPARAM MAR**: Located in the lower-middle section of the grid, to the right of SYSMAR.
- ASTDEL LIS**: Located in the lower-right section of the grid.