H 5

```
FFFFFFFFFFFFFFF         111           111         XXX         XXX
FFFFFFFFFFFFFFF         111           111         XXX         XXX
FFFFFFFFFFFFFFF         111           111         XXX         XXX
FFF                  111111        111111         XXX         XXX
FFF                  111111        111111         XXX         XXX
FFF                  111111        111111         XXX         XXX
FFF                     111           111            XXX   XXX
FFF                     111           111            XXX   XXX
FFF                     111           111            XXX   XXX
FFFFFFFFFFFF            111           111               XXX
FFFFFFFFFFFF            111           111               XXX
FFFFFFFFFFFF            111           111               XXX
FFF                     111           111            XXX   XXX
FFF                     111           111            XXX   XXX
FFF                     111           111            XXX   XXX
FFF                     111           111         XXX         XXX
FFF                     111           111         XXX         XXX
FFF                     111           111         XXX         XXX
FFF                  111111111     111111111      XXX         XXX
FFF                  111111111     111111111      XXX         XXX
FFF                  111111111     111111111      XXX         XXX
```

_$25

Symb

IOC$
IO_C
IO_C
IO_D
IO_F
IO_S
KILL

KILL
KILL
LB_E
LB_D
LB_F
LB_M
LB_L
LOCA
LOCA
LOCK

LOCK
LOCK
LOCK
LOC_
LOC_
L_CO
L_CO
L_DA
L_DA
MAIN
MAKE
MAKE
MAKE
MAKE
MAKE
MAKE

MAKE
MAKE
MAP_
MAP_

MAP
MARK
MARK
MARK
MARK
MARK

FILEID**RDBLOK


```
RRRRRRRR   DDDDDDDD   BBBBBBBB   LL          OOOOOO    KK      KK
RRRRRRRR   DDDDDDDD   BBBBBBBB   LL          OOOOOO    KK      KK
RR      RR DD      DD BB      BB LL        OO      OO  KK      KK
RR      RR DD      DD BB      BB LL        OO      OO  KK      KK
RR      RR DD      DD BB      BB LL        OO      OO  KK    KK
RR      RR DD      DD BB      BB LL        OO      OO  KK    KK
RRRRRRRR   DD      DD BBBBBBBB   LL        OO      OO  KKKKKK
RRRRRRRR   DD      DD BBBBBBBB   LL        OO      OO  KKKKKK
RR  RR     DD      DD BB      BB LL        OO      OO  KK    KK
RR  RR     DD      DD BB      BB LL        OO      OO  KK    KK
RR    RR   DD      DD BB      BB LL        OO      OO  KK      KK    ....
RR    RR   DD      DD BB      BB LL        OO      OO  KK      KK    ....
RR      RR DDDDDDDD   BBBBBBBB   LLLLLLLLLL  OOOOOO    KK      KK    ....
RR      RR DDDDDDDD   BBBBBBBB   LLLLLLLLLL  OOOOOO    KK      KK    ....

LL          IIIIII    SSSSSSSS
LL          IIIIII    SSSSSSSS
LL            II    SS
LL            II    SS
LL            II    SS
LL            II    SS
LL            II      SSSSSS
LL            II      SSSSSS
LL            II            SS
LL            II            SS
LL            II            SS
LL            II            SS
LLLLLLLLLL  IIIIII  SSSSSSSS
LLLLLLLLLL  IIIIII  SSSSSSSS
```

```
0 MODULE RDBLOK (
0                 LANGUAGE (BLISS32),
0                 IDENT = 'V04-000'
0                 ) =
1 BEGIN
1
1 !
1 !****************************************************************************
1 !*                                                                          *
1 !*  COPYRIGHT (c) 1978, 1980, 1982, 1984 BY                                 *
1 !*  DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASSACHUSETTS.                  *
1 !*  ALL RIGHTS RESERVED.                                                    *
1 !*                                                                          *
1 !*  THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND COPIED   *
1 !*  ONLY IN  ACCORDANCE WITH  THE  TERMS  OF  SUCH  LICENSE  AND WITH THE   *
1 !*  INCLUSION OF THE ABOVE COPYRIGHT NOTICE. THIS SOFTWARE OR  ANY  OTHER   *
1 !*  COPIES THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY   *
1 !*  OTHER PERSON.  NO TITLE TO AND OWNERSHIP OF  THE  SOFTWARE IS  HEREBY   *
1 !*  TRANSFERRED.                                                            *
1 !*                                                                          *
1 !*  THE INFORMATION IN THIS SOFTWARE IS  SUBJECT TO CHANGE WITHOUT NOTICE   *
1 !*  AND  SHOULD  NOT  BE  CONSTRUED AS  A COMMITMENT BY DIGITAL EQUIPMENT   *
1 !*  CORPORATION.                                                            *
1 !*                                                                          *
1 !*  DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE  OR  RELIABILITY OF ITS   *
1 !*  SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL.                 *
1 !*                                                                          *
1 !*                                                                          *
1 !****************************************************************************
1
1 !++
1 !
1 ! FACILITY:  F11ACP Structure Level 2
1 !
1 ! ABSTRACT:
1 !
1 !       This module contains routines for basic block I/O, as well
1 !       as the buffer management mechanism.
1 !
1 ! ENVIRONMENT:
1 !
1 !       STARLET operating system, including privileged system services
1 !       and internal exec routines.
1 !
1 !--
1 !
1 !
1 ! AUTHOR:  Andrew C. Goldstein,  CREATION DATE:  13-Dec-1976  22:48
1 !
1 ! MODIFIED BY:
1 !
1 !       V03-032 CDS0015         Christian D. Saether    30-Aug-1984
1 !               Allow for multi-header directory files.
1 !
1 !       V03-031 CDS0014         Christian D. Saether    24-Aug-1984
1 !               Removing an entry from the directory index cache
1 !               free list had one too many levels of indirection,
```

```
                 occasionally messing up the queue.

         V03-030 CDS0013             Christian D. Saether     7-Aug-1984
                 Add KILL_DINDX routine.  Remove tests for dir_fcb
                 and primary_fcb in unhook_bfrd - the fcb$v_dir flag
                 handles those conditions correctly.

         V03-029 CDS0012             Christian D. Saether     6-Aug-1984
                 Fix bug in CDS0011 causing infinite loop when
                 mounting disk /nocache, or while marked for dismount.
                 Use L_NORM linkage for UNHOOK_BFRD.

         V03-028 CDS0011             Christian D. Saether    15-Jul-1984
                 Add support for maintaining directory index blocks.
                 Add MAKE_DIRINDX routine.
                 Remove intermediate blocks from CDS0010.  The
                 problem was in the form of the bind_common
                 declarations and was resolved there.

         V03-027 CDS0010             Christian D. Saether     7-Jul-1984
                 Break up READ_BLOCK and FIND_BUFFER into smaller
                 blocks to stop the absurd amount of cse that the
                 compiler generates.

         V03-026 CDS0009             Christian D. Saether     2-Jul-1984
                 Remove CACHELOCK consistency check so that cell
                 can be used for STS_DISKREAD flag.
                 Raise minimum file header requirements to 3 buffers
                 from 2 buffers.

         V03-025 CDS0008             Christian D. Saether    24-Jun-1984
                 Restore multi-block read capability.

         V03-024 CDS0007             Christian D. Saether    20-Jun-1984
                 Raise/lower process diocnt around qio so that file
                 system i/o is not blocked for lack of quota.  Do
                 same for ASTCNT so it does not fail for lack of
                 ast quota either.

         V03-023 CDS0006             Christian D. Saether    25-May-1984
                 When not cluster accessible, do not bump
                 sequence number on modifies.
                 Add routine to scan appropriate pool and kick
                 out buffers of a given type.

         V03-022 Expand test for clusterness to test whether we are
                 are a cluster at all.

         V03-021 CDS0004             Christian D. Saether     8-May-1984
                 Do not leave lock on buffer if not cluster device.
                 Modify buffer validation for bitmap, index, and quota
                 type buffers.

         V03-020 CDS0003             Christian D. Saether    10-Apr-1984
                 Bump appropriate pool hit/miss counters.
                 Don't set VALID, DIRTY in bfrd in CREATE_BLOCK if
                 for special lbn -1.  Set VALID, DIRTY in bfrd in
```

```
    1 !              RESET_LBN routine.
    1 !
    1 !      V03-019 CDS0002          Christian D. Saether      4-Apr-1984
    1 !              On lockbasis mismatch, when tolerated at all (for
    1 !              blocks in the data or headers), toss the buffer out
    1 !              to get the lock dissociated because it's the wrong
    1 !              one.
    1 !
    1 !      V03-018 CDS0001          Christian D. Saether     20-Mar-1984
    1 !              Establish BASE as AST parameter in SERIAL_CACHE routine.
    1 !
    1 !      V03-017 ACG0408          Andrew C. Goldstein,     23-Mar-1984  14:47
    1 !              Add AST parameter so that impure storage is fully based
    1 !
    1 !      V03-016 ACG0403          Andrew C. Goldstein,     15-Mar-1984  17:35
    1 !              Correct test for new buffer in WRONG_LOCKBASIS;
    1 !              fix buffer address computation in TOSS_CACHE_DATA.
    1 !
    1 !      V03-015 CDS     Christian D. Saether      9-Mar-1984
    1 !              Rewrite cache routines for shared paged pool caching.
    1 !
    1 !**
    1
    1
    1 LIBRARY 'SYS$LIBRARY:LIB.L32';
    1 REQUIRE 'SRC$:FCPDEF.B32';
    1
    1 FORWARD ROUTINE
    1        GET_REQD_BFR_CREDITS : L_NORM NOVALUE,
    1        RETURN_BFRD          : L_JSB_TARG NOVALUE,
    1        FIND_BUFFER          : L_NORM,
    1        INVALIDATE           : L_NORM NOVALUE,
    1        UNHOOK_BFRD          : L_NORM NOVALUE,
    1        WRITE_BLOCK          : L_NORM NOVALUE,
    1        FREE_ONE             : L_JSB NOVALUE,
    1        WAKE_WAITER          : NOVALUE,
    1        RELEASE_CACHE        : L_JSB NOVALUE,
    1        SERIAL_CACHE         : L_JSB NOVALUE;
    1
    1 EXTERNAL ROUTINE
    1        WAIT_FOR_AST    : NOVALUE,        ! exit thread until completion ast
    1        CONTINUE_THREAD : NOVALUE;        ! completion AST to resume thread
    1
    1 EXTERNAL
    1        CTL$GL_PHD      : REF BBLOCK ADDRESSING_MODE (GENERAL),
    1        CTL$GL_PCB      : REF BBLOCK ADDRESSING_MODE (GENERAL);
    1
    1 MACRO
    1        QFLNK = 0,0,32,0 %,
    1        QBLNK = 4,0,32,0 %,
    1
M   1        LOOKUP_LBN (LBN) =
M   1
M   1               BEGIN
M   1               BIND
M   1                   HSHTBL = .CACHE_HDR [F11BC$L_LBNHSHBAS] : VECTOR [,WORD];
M   1
```

```
                  HSHTBL [ABS (LBN MOD .CACHE_HDR [F11BC$W_LBNHSHCNT])]
                  END %,

          LOOKUP_LOCK (LOCKBASIS, PARLKID) =

                  BEGIN
                  BIND
                      HSHTBL = .CACHE_HDR [F11BC$L_BLHSHBAS] : VECTOR [,WORD];

                  HSHTBL [ABS ((LOCKBASIS + PARLKID) MOD .CACHE_HDR [F11BC$W_BLHSHCNT])]
                  END %,

! Calculate address of given descriptor within 1-based descriptor blockvectors.
!

          BFRD_ADDR (BAS1INDX) =
                  .CACHE_HDR [F11BC$L_BFRDBAS] + (BAS1INDX - 1)*BFRD$S_BFRDDEF %,

          BFRLD_ADDR (BAS1INDX) =
                  .CACHE_HDR [F11BC$L_BFRLDBAS] + (BAS1INDX - 1)*BFRL$S_BFRLDEF %;

BIND
          POOL_TABLE      = UPLIT BYTE (  2,        ! file headers
                                          0,        ! storage map
                                          1,        ! directories
                                          2,        ! index file blocks
                                          1,        ! random data blocks
                                          1,        ! quota file blocks
                                          3         ! directory index blocks
                          ) : VECTOR [,BYTE];
```

```
1 GLOBAL ROUTINE MAKE_DIRINDX (FCB) : L_NORM =
1
1 !++
1 !
1 ! FUNCTIONAL DESCRIPTION:
1 !
1 ! This routine validates the directory index pointed to from the
1 ! given FCB, under the cache interlock.
1 !
1 ! If the given FCB does not have a directory index, attempt to
1 ! attach one to it.
1 !
1 ! ROUTINE VALUE:
1 !
1 !       0 - if the given FCB is invalid
1 !       1 - if the given FCB is valid
1 !
1 !--
1
2 BEGIN
2
2 MAP
2       FCB     : REF BBLOCK;
2
2 BIND_COMMON;
2
2 EXTERNAL
2       PMS$GL_DIRHIT   : ADDRESSING_MODE (GENERAL),
2       PMS$GL_DIRMISS  : ADDRESSING_MODE (GENERAL);
2 LOCAL
2       DIRINDX         : REF BBLOCK FIELD (DIRC),
2       DNDX_BFRD       : REF BBLOCK,
2       FCBVALID        : INITIAL (0);
2
2 IF .LB_HDRSEQ [.DIR_LCKINDX] EQL 0
2 THEN
2     LB_HDRSEQ [.DIR_LCKINDX] = .LB_HDRSEQ [.DIR_LCKINDX] + 1;
2
2 IF .LB_DATASEQ [.DIR_LCKINDX] EQL 0
2 THEN
2     LB_DATASEQ [.DIR_LCKINDX] = .LB_DATASEQ [.DIR_LCKINDX] + 1;
2
2 SERIAL_CACHE ();
2
2 IF (DIRINDX = .FCB [FCB$L_DIRINDX]) EQL 0
2 THEN
3     BEGIN
3     LOCAL
3         INDX0,
3         POOL_LRU        : REF BBLOCK;
3
3     POOL_LRU = CACHE_HDR [F11BC$Q_POOL_LRU] + 3*8;
3
3     REMQUE (.POOL_LRU [QFLNK], DNDX_BFRD);
3
3     INDX0 = ((.DNDX_BFRD - .CACHE_HDR [F11BC$L_BFRDBAS])/BFRD$S_BFRDDEF);
```

```
3      UNHOOK_BFRD (.INDX0 + 1, .DNDX_BFRD);
3
3      DIRINDX = (.INDX0*512) + .CACHE_HDR [F11BC$L_BUFBASE];
3
3      DNDX_BFRD [BFRD$L_LBN] = .FCB;
3      DNDX_BFRD [BFRD$B_BTYPE] = DIRINDX_TYPE;
3      DNDX_BFRD [BFRD$L_LOCKBASIS] = .LB_BASIS [.DIR_LCKINDX];
3      DNDX_BFRD [BFRD$L_UCB] = .CURRENT_UCB;
3      FCB [FCB$L_DIRINDX] = .DIRINDX;
3      DNDX_BFRD [BFRD$V_VALID] = 1;
3      DIRINDX [DIRC$W_INUSE] = 0;
3      PMS$GL_DIRMISS = .PMS$GL_DIRMISS + 1;
3      END
2  ELSE
3      BEGIN
3
4      DNDX_BFRD = (((.DIRINDX - .CACHE_HDR [F11BC$L_BUFBASE])/512)*BFRD$S_BFRDDEF)
3                 + .CACHE_HDR [F11BC$L_BFRDBAS];
3
3      REMQUE (.DNDX_BFRD, DNDX_BFRD);
3
3      IF .DNDX_BFRD [BFRD$L_SEQNUM] EQL .LB_HDRSEQ [.DIR_LCKINDX]
3      THEN
4          BEGIN
4          FCBVALID = 1;
4
4          IF .DIRINDX [DIRC$L_DATASEQ] NEQ .LB_DATASEQ [.DIR_LCKINDX]
4          THEN
5              BEGIN
5              DIRINDX [DIRC$W_INUSE] = 0;
5              PMS$GL_DIRMISS = .PMS$GL_DIRMISS + 1;
5              END
4          ELSE
4              PMS$GL_DIRHIT = .PMS$GL_DIRHIT + 1;
4
4          END
3      ELSE
4          BEGIN
4          DIRINDX [DIRC$W_INUSE] = 0;
4          PMS$GL_DIRMISS = .PMS$GL_DIRMISS + 1;
3          END;
3
2      END;
2
2  IF .BFRS_USED [3] NEQ 0
2  THEN
3      BUG_CHECK (XQPERR, 'should not have any in use')
2  ELSE
2      BFRS_USED [3] = 1;
2
2  IF .DNDX_BFRD [BFRD$W_CURPID] NEQ 0
2  THEN
3      BUG_CHECK (XQPERR, 'directory index buffer should not be in use')
2  ELSE
2      DNDX_BFRD [BFRD$W_CURPID] = .CTL$GL_PCB [PCB$L_PID];
2
2  INSQUE (.DNDX_BFRD, .BFR_LIST [3, QBLNK]);
```

```
;  317      1306 2
;  318      1307 2 RELEASE_CACHE ();
;  319      1308 2
;  320      1309 2 .FCBVALID
;  321      1310 2
;  322      1311 1 END;                 ! of routine MAKE_DIRINDX

                                                       .TITLE  RDBLOK
                                                       .IDENT  \V04-000\

                                                       .PSECT  $CODE$,NOWRT,2

        03  01  01  02  01  00  02  00000 P.AAA:       .BYTE   2, 0, 1, 2, 1, 1, 3                        ;

                                          POOL_TABLE=          P.AAA
                                                       .EXTRN  WAIT_FOR_AST, CONTINUE_THREAD
                                                       .EXTRN  CTL$GL_PHD, CTL$GL_PCB
                                                       .EXTRN  PMS$GL_DIRHIT, PMS$GL_DIRMISS
                                                       .EXTRN  BUG$_XQPERR

                                01FC 00000             .ENTRY  MAKE_DIRINDX, Save R2,R3,R4,R5,R6,R7,R8    ; 1192
            58        CC   AA   9E 00002             MOVAB   -52(BASE), R8                              ; 1214
            55        F4   AA   9E 00006             MOVAB   -12(BASE), R5                              ;
            56      00D4   CA   9E 0000A             MOVAB   212(BASE), R6                              ;
                           57   D4 0000F             CLRL    FCBVALID                                   ;
            50             66   D0 00011             MOVL    (R6), R0                                   ; 1226
            50      0094 CA40   DE 00014             MOVAL   148(BASE)[R0], R0                          ;
                           60   D5 0001A             TSTL    (R0)                                       ;
                           02   12 0001C             BNEQ    1$                                         ;
                           60   D6 0001E             INCL    (R0)                                       ; 1228
            50             66   D0 00020 1$:         MOVL    (R6), R0                                   ; 1230
            50      00A8 CA40   DE 00023             MOVAL   168(BASE)[R0], R0                          ;
                           60   D5 00029             TSTL    (R0)                                       ;
                           02   12 0002B             BNEQ    2$                                         ;
                           60   D6 0002D             INCL    (R0)                                       ; 1232
                        0000V   30 0002F 2$:         BSBW    SERIAL_CACHE                               ; 1234
            50        04   AC   D0 00032             MOVL    FCB, R0                                    ; 1236
            54      00B0   C0   D0 00036             MOVL    176(R0), DIRINDX                           ;
                           54   12 0003B             BNEQ    3$                                         ;
    50       FC   AA 00000040   8F   C1 0003D         ADDL3   #64, -4(BASE), POOL_LRU                    ; 1243
            52        00        B0   0F 00046         REMQUE  @0(POOL_LRU), DNDX_BFRD                    ; 1245
            50        FC        AA   D0 0004A         MOVL    -4(BASE), R0                               ; 1247
    50      52        18        A0   C3 0004E         SUBL3   24(R0), DNDX_BFRD, R0                      ;
    53      50                  20   C7 00053         DIVL3   #32, R0, INDX0                             ;
                                52   DD 00057         PUSHL   DNDX_BFRD                                  ; 1249
                      01        A3   9F 00059         PUSHAB  1(INDX0)                                   ;
       0000V  CF                02   FB 0005C         CALLS   #2, UNHOOK_BFRD                            ;
    53      53                  09   78 00061         ASHL    #9, R3, R3                                 ; 1251
    54      53        FC        BA   C1 00065         ADDL3   @-4(BASE), R3, DIRINDX                     ;
        08  A2        04        AC   D0 0006A         MOVL    FCB, 8(DNDX_BFRD)                          ; 1253
        19  A2                  06   90 0006F         MOVB    #6, 25(DNDX_BFRD)                          ; 1254
            50                  66   D0 00073         MOVL    (R6), R0                                   ; 1255
        10  A2      0080 CA40        D0 00076         MOVL    128(BASE)[R0], 16(DNDX_BFRD)               ;
        0C  A2        94        AA   D0 0007D         MOVL    -108(BASE), 12(DNDX_BFRD)                  ; 1256
            50        04        AC   D0 00082         MOVL    FCB, R0                                    ; 1257
       00B0 C0                  54   D0 00086         MOVL    DIRINDX, 176(R0)                           ;
```

```
            18  A2                08  88 0008B           BISB2   #8, 24(DNDX_BFRD)                          ; 1258
                                  3D  11 0008F           BRB     4$                                         ; 1259
                50          FC    AA  D0 00091 3$:       MOVL    -4(BASE), R0                               ; 1265
       51       54                60  C3 00095           SUBL3   (R0), DIRINDX, R1                          ;
                51    00000200    8F  C6 00099           DIVL2   #512, R1                                   ;
                51                20  C4 000A0           MULL2   #32, R1                                    ;
       52       51          18    A0  C1 000A3           ADDL3   24(R0), R1, DNDX_BFRD                      ; 1266
                52                62  0F 000A8           REMQUE  (DNDX_BFRD), DNDX_BFRD                     ; 1268
                50                66  D0 000AB           MOVL    (R6), R0                                   ; 1270
          0094 CA40         14    A2  D1 000AE           CMPL    20(DNDX_BFRD), 148(BASE)[R0]               ;
                                  17  12 000B5           BNEQ    4$                                         ;
                57                01  D0 000B7           MOVL    #1, FCBVALID                               ; 1273
                50                66  D0 000BA           MOVL    (R6), R0                                   ; 1275
          00A8 CA40         08    A4  D1 000BD           CMPL    8(DIRINDX), 168(BASE)[R0]                  ;
                                  08  12 000C4           BNEQ    4$                                         ;
                         00000000G  00  D6 000C6         INCL    PMS$GL_DIRHIT                              ; 1282
                                  08  11 000CC           BRB     5$                                         ; 1270
                                  64  B4 000CE 4$:       CLRW    (DIRINDX)                                  ; 1287
                         00000000G  00  D6 000D0         INCL    PMS$GL_DIRMISS                             ; 1288
                            06    A5  B5 000D6 5$:       TSTW    6(R5)                                      ; 1293
                                  06  13 000D9           BEQL    6$                                         ;
                                FEFF 000DB               BUGW                                               ; 1295
                               0000* 000DD               .WORD   <BUG$_XQPERR!4>                            ;
                                  04  11 000DF           BRB     7$                                         ;
            06  A5                01  B0 000E1 6$:       MOVW    #1, 6(R5)                                  ; 1297
                            1C    A2  B5 000E5 7$:       TSTW    28(DNDX_BFRD)                              ; 1299
                                  06  13 000E8           BEQL    8$                                         ;
                                FEFF 000EA               BUGW                                               ; 1301
                               0000* 000EC               .WORD   <BUG$_XQPERR!4>                            ;
                                  0C  11 000EE           BRB     9$                                         ;
                50   00000000G    00  D0 000F0 8$:       MOVL    CTL$GL_PCB, R0                             ; 1303
            1C  A2          60    A0  B0 000F7           MOVW    96(R0), 28(DNDX_BFRD)                      ;
            1C  B8                62  0E 000FC 9$:       INSQUE  (DNDX_BFRD), @28(R8)                       ; 1305
                               0000V  30 00100           BSBW    RELEASE_CACHE                              ; 1307
                50                57  D0 00103           MOVL    FCBVALID, R0                               ; 1311
                                      04 00106           RET                                                ;

; Routine Size:  263 bytes,    Routine Base:  $CODE$ + 0007
```

```
GLOBAL ROUTINE KILL_DINDX (FCB) : L_NORM NOVALUE =

!++
!
! FUNCTIONAL DESCRIPTION:
!
!       Invalidate the directory index block pointed to by
!       the given fcb, if any.  This is done under the cache
!       serialization lock.
!
!--

BEGIN

MAP
        FCB     : REF BBLOCK;

BIND_COMMON;

LOCAL
        INDX0,
        BFRD    : REF BBLOCK,
        DIRINDX : REF BBLOCK,
        PIDINDX : WORD;

SERIAL_CACHE ();

IF (DIRINDX = .FCB [FCB$L_DIRINDX]) NEQ 0
THEN
    BEGIN
    INDX0 = (.DIRINDX - .CACHE_HDR [F11BC$L_BUFBASE])/512;
    BFRD = .CACHE_HDR [F11BC$L_BFRDBAS] + BFRD$S_BFRDDEF*(.INDX0);

    PIDINDX = .CTL$GL_PCB [PCB$L_PID];

    IF .BFRD [BFRD$W_CURPID] NEQ 0
    THEN
        BEGIN
        IF .BFRD [BFRD$W_CURPID] NEQ .PIDINDX
        THEN
            BUG_CHECK (XQPERR, 'should not belong to anyone else')
        ELSE
            BEGIN

! If this one is in-process, simply clear valid to cause it to
! be cleaned up when locks are run down.  Also clear the pointer
! to the fcb in the bfrd.
!

            BFRD [BFRD$V_VALID] = 0;
            BFRD [BFRD$L_LBN] = 0;
            END;
        END
    ELSE
        BEGIN
        UNHOOK_BFRD (.INDX0+1, .BFRD);
        RETURN_BFRD (.BFRD);
```

```
;  381         1369 3            END;
;  382         1370 3
;  383         1371 3        FCB [FCB$L_DIRINDX] = 0;
;  384         1372 3
;  385         1373 2        END;
;  386         1374 2
;  387         1375 2 RELEASE_CACHE ();
;  388         1376 2
;  389         1377 1 END;                 ! of routine KILL_DINDX


                                        000C 00000       .ENTRY  KILL_DINDX, Save R2,R3                    ; 1312
                                 0000V    30 00002       BSBW    SERIAL_CACHE                              ; 1337
                 50        04       AC    D0 00005       MOVL    FCB, R0                                   ; 1339
                 50      00B0       C0    D0 00009       MOVL    176(R0), DIRINDX                          ;
                                    54    13 0000E       BEQL    4$                                        ;
                 50        FC       BA    C2 00010       SUBL2   @-4(BASE), R0                             ; 1342
                 50  00000200       8F    C6 00014       DIVL2   #512, INDX0                               ;
                 51        FC       AA    D0 0001B       MOVL    -4(BASE), R1                              ; 1343
         52      50                 05    78 0001F       ASHL    #5, INDX0, R2                             ;
                 52        18       A1    C0 00023       ADDL2   24(R1), BFRD                              ;
                 51  00000000G      00    D0 00027       MOVL    CTL$GL_PCB, R1                            ; 1345
                 53        60       A1    B0 0002E       MOVW    96(R1), PIDINDX                           ;
                           1C       A2    B5 00032       TSTW    28(BFRD)                                  ; 1347
                                    15    13 00035       BEQL    2$                                        ;
                 53        1C       A2    B1 00037       CMPW    28(BFRD), PIDINDX                         ; 1350
                                    06    13 0003B       BEQL    1$                                        ;
                                        FEFF 0003D       BUGW                                              ; 1352
                                       0000* 0003F       .WORD   <BUG$_XQPERR!4>                           ;
                                    19    11 00041       BRB     3$                                        ;
            18   A2                 08    8A 00043 1$:   BICB2   #8, 24(BFRD)                              ; 1361
                           08       A2    D4 00047       CLRL    8(BFRD)                                   ; 1362
                                    10    11 0004A       BRB     3$                                        ; 1347
                                    52    DD 0004C 2$:   PUSHL   BFRD                                      ; 1367
                           01       A0    9F 0004E       PUSHAB  1(INDX0)                                  ;
         0000V   CF                 02    FB 00051       CALLS   #2, UNHOOK_BFRD                           ;
                 50                 52    D0 00056       MOVL    BFRD, R0                                  ; 1368
                                 0000V    30 00059       BSBW    RETURN_BFRD                               ;
                 50        04       AC    D0 0005C 3$:   MOVL    FCB, R0                                   ; 1371
                         00B0       C0    D4 00060       CLRL    176(R0)                                   ;
                                 0000V    30 00064 4$:   BSBW    RELEASE_CACHE                             ; 1375
                                          04 00067       RET                                               ; 1377

; Routine Size:  104 bytes,    Routine Base:  $CODE$ + 010E
```

```
1 GLOBAL ROUTINE READ_BLOCK (LBN, COUNT, TYPE) : L_NORM =
1
1 !++
1 !
1 ! FUNCTIONAL DESCRIPTION:
1 !
1 !       This routine reads the desired block(s) from the disk.
1 !       Blocks are categorized by type to aid buffer management.
1 !       Note that the caller assumes only one block is ever read; multiple
1 !       blocks read ahead are acquired through cache hits on subsequent calls.
1 ! CALLING SEQUENCE:
1 !       READ_BLOCK (ARG1, ARG2, ARG3)
1 !
1 ! INPUT PARAMETERS:
1 !       ARG1: LBN of block(s)
1 !       ARG2: number of blocks to read
1 !       ARG3: block type code
1 !
1 ! IMPLICIT INPUTS:
1 !
1 ! OUTPUT PARAMETERS:
1 !       NONE
1 !
1 ! IMPLICIT OUTPUTS:
1 !       IO_STATUS receives status of I/O transfer
1 !
1 ! ROUTINE VALUE:
1 !       address of buffer containing block
1 !
1 ! SIDE EFFECTS:
1 !       BLOCK READ
1 !
1 !--
1
2 BEGIN
2
2
2 LOCAL
2       I,                              ! index of buffer used
2       BFRD            : REF BBLOCK,   ! pointer to buffer descriptor
2       STATUS,                         ! QIO service status
2       FOUND_COUNT;                    ! count of buffers gotten
2
2 EXTERNAL
2       ACP$GB_DATACHK  : BITVECTOR ADDRESSING_MODE (ABSOLUTE);
2                                       ! ACP datacheck enable flags
2
2 BIND_COMMON;
2
2 EXTERNAL LITERAL
2       ACP$V_READCHK   : UNSIGNED (6); ! read check enable flag
2
2 ! Find a suitable block buffer. If it does not already contain the block,
2 ! read it.
2 !
2
2 STSFLGS [STS_DISKREAD] = 0;
```

```
I = FIND_BUFFER (.LBN, .TYPE, .COUNT, FOUND_COUNT);

BFRD = .CACHE_HDR [F11BC$L_BFRDBAS] + (.I*BFRD$S_BFRDDEF);

IF .BFRD [BFRD$V_VALID]
THEN
    RETURN .CACHE_HDR [F11BC$L_BUFBASE] + (.I*512);

    BEGIN
    LOCAL
        PTR        : REF BBLOCK,
        SAVE_PRIV  : VECTOR [4];

    STSFLGS [STS_DISKREAD] = 1;

    PTR = .CTL$GL_PCB;
    PTR [PCB$W_DIOCNT] = .PTR [PCB$W_DIOCNT] + 1;
    PTR [PCB$W_ASTCNT] = .PTR [PCB$W_ASTCNT] + 1;
    SAVE_PRIV [0] = .(PTR [PCB$Q_PRIV]);
    SAVE_PRIV [1] = .(PTR [PCB$Q_PRIV]+4);
    BBLOCK [PTR [PCB$Q_PRIV], PRV$V_LOG_IO] = 1;
    BBLOCK [PTR [PCB$Q_PRIV], PRV$V_BYPASS] = 1;
    PTR = .CTL$GL_PHD;
    SAVE_PRIV [2] = .(PTR [PHD$Q_PRIVMSK]);
    SAVE_PRIV [3] = .(PTR [PHD$Q_PRIVMSK]+4);
    BBLOCK [PTR [PHD$Q_PRIVMSK], PRV$V_LOG_IO] = 1;
    BBLOCK [PTR [PHD$Q_PRIVMSK], PRV$V_BYPASS] = 1;

    PMS_TOT_READ = .PMS_TOT_READ + 1;
    STATUS = $QIO (
                  EFN    = EFN,
                  ASTADR = CONTINUE_THREAD,
                  ASTPRM = .BASE,
                  CHAN   = .IO_CHANNEL,
                  FUNC   = (IO$_READLBLK
                            OR .ACP$GB_DATACHK[ACP$V_READCHK]
                            ^ $BITPOSITION (IO$V_DATACHECK)),
                  IOSB   = IO_STATUS,
                  P1     = .CACHE_HDR [F11BC$L_BUFBASE] + (.I*512),
                  P2     = .FOUND_COUNT*512,
                  P3     = .LBN
                  );

    (PTR [PHD$Q_PRIVMSK]) = .SAVE_PRIV [2];
    (PTR [PHD$Q_PRIVMSK]+4) = .SAVE_PRIV [3];
    PTR = .CTL$GL_PCB;
    PTR [PCB$W_DIOCNT] = .PTR [PCB$W_DIOCNT] - 1;
    PTR [PCB$W_ASTCNT] = .PTR [PCB$W_ASTCNT] - 1;
    (PTR [PCB$Q_PRIV]) = .SAVE_PRIV [0];
    (PTR [PCB$Q_PRIV]+4) = .SAVE_PRIV [1];

    IF NOT .STATUS
    THEN IO_STATUS = .STATUS
    ELSE WAIT_FOR_AST();

    IF NOT .IO_STATUS
```

```
3          THEN
4              BEGIN
4              INCR J FROM 0 TO .FOUND_COUNT-1
4              DO
4                  INVALIDATE (.CACHE_HDR [F11BC$L_BUFBASE] + (.I+.J)*512);
4              ERR_EXIT (.IO_STATUS[0]);
3              END;
3
3          INCR J FROM 0 TO .FOUND_COUNT - 1
3          DO
4              BEGIN
4              BFRD [BFRD$V_VALID] = 1;
4              BFRD = .BFRD + BFRD$S_BFRDDEF;
3              END;
3
2          END;
2
3      RETURN .CACHE_HDR [F11BC$L_BUFBASE] + (.I*512)
3
1      END;                                        ! end of routine READ_BLOCK


                                                       .EXTRN  ACP$GB_DATACHK, ACP$V_READCHK
                                                       .EXTRN  SYS$QIO

                                  00FC 00000           .ENTRY  READ_BLOCK, Save R2,R3,R4,R5,R6,R7          ; 1378
              57 00000000G   00     9E 00002           MOVAB   CTL$GL_PCB, R7                              ;
              5E             14     C2 00009           SUBL2   #20, SP                                     ;
              54         FC  AA     9E 0000C           MOVAB   -4(BASE), R4                                ; 1422
       A6     AA             01     8A 00010           BICB2   #1, -90(BASE)                               ; 1434
                             5E     DD 00014           PUSHL   SP                                          ; 1436
                         08  AC     DD 00016           PUSHL   COUNT                                       ;
                         0C  AC     DD 00019           PUSHL   TYPE                                        ;
                         04  AC     DD 0001C           PUSHL   LBN                                         ;
    0000V     CF             04     FB 0001F           CALLS   #4, FIND_BUFFER                             ;
              56             50     D0 00024           MOVL    R0, I                                       ;
              51             64     D0 00027           MOVL    (R4), R1                                    ; 1438
50            56             05     78 0002A           ASHL    #5, I, R0                                   ;
53            50         18  A1     C1 0002E           ADDL3   24(R1), R0, BFRD                            ;
03       18   A3             03     E1 00033           BBC     #3, 24(BFRD), 1$                            ; 1440
                       00CC         31 00038           BRW     9$                                          ;
       A6     AA             01     88 0003B  1$:      BISB2   #1, -90(BASE)                               ; 1449
              52             67     D0 0003F           MOVL    CTL$GL_PCB, PTR                             ; 1451
                         3E  A2     B6 00042           INCW    62(PTR)                                     ; 1452
                         38  A2     B6 00045           INCW    56(PTR)                                     ; 1453
       04     AE       0084  C2     7D 00048           MOVQ    132(PTR), SAVE_PRIV                         ; 1454
     0084     C2 20000080    8F     C8 0004E           BISL2   #536871040, 132(PTR)                        ; 1457
              52 00000000G   00     D0 00057           MOVL    CTL$GL_PHD, PTR                             ; 1458
       0C     AE             62     7D 0005E           MOVQ    (PTR), SAVE_PRIV+8                          ; 1459
              62 20000080    8F     C8 00062           BISL2   #536871040, (PTR)                           ; 1462
                       08E8  CA     D6 00069           INCL    2280(BASE)                                  ; 1464
                         7E         7C 0006D           CLRQ    -(SP)                                       ; 1477
                         7E         D4 0006F           CLRL    -(SP)                                       ;
                         04  AC     DD 00071           PUSHL   LBN                                         ;
7E       10   AE             09     78 00074           ASHL    #9, FOUND_COUNT, -(SP)                      ;
              51         00  B4     D0 00079           MOVL    @0(R4), R1                                  ;
```

```
            50          56            09  78 0007D         ASHL     #9, I, R0                                   ;
                                    6041  9F 00081         PUSHAB   (R0)[R1]                                    ;
                                      5A  DD 00084         PUSHL    BASE                                        ;
                               0000G  CF  9F 00086         PUSHAB   CONTINUE_THREAD                             ;
                                  88  AA  9F 0008A         PUSHAB   -120(BASE)                                  ;
   50 00000000G 9F              01       00G  EF 0008D     EXTZV    S^ACP$V_READCHK, #1, @#ACP$GB_DATACHK, R0   ;
            50          50            0E  78 00096         ASHL     #14, R0, R0                                 ;
            7E          50            21  C9 0009A         BISL3    #33, R0, -(SP)                              ;
                                FF78  CA  DD 0009E         PUSHL    -136(BASE)                                  ;
                                      1E  DD 000A2         PUSHL    #30                                         ;
                 00000000G  00        0C  FB 000A4         CALLS    #12, SYS$QIO                                ;
                            62    0C  AE  7D 000AB         MOVQ     SAVE_PRIV+8, (PTR)                          ; 1479
                            52        67  D0 000AF         MOVL     CTL$GL_PCB, PTR                             ; 1481
                                  3E  A2  B7 000B2         DECW     62(PTR)                                     ; 1482
                                  38  A2  B7 000B5         DECW     56(PTR)                                     ; 1483
                     0084   C2    04  AE  7D 000B8         MOVQ     SAVE_PRIV, 132(PTR)                         ; 1484
                            06        50  E8 000BE         BLBS     STATUS, 2$                                  ; 1487
                       88   AA        50  D0 000C1         MOVL     STATUS, -120(BASE)                          ; 1488
                                      05  11 000C5         BRB      3$                                          ;
                    0000G   CF        00  FB 000C7  2$:    CALLS    #0, WAIT_FOR_AST                            ; 1489
                            24    88  AA  E8 000CC  3$:    BLBS     -120(BASE), 6$                              ; 1491
                            55        6E  D0 000D0         MOVL     FOUND_COUNT, R5                             ; 1494
                            52        01  CE 000D3         MNEGL    #1, J                                       ;
                                      14  11 000D6         BRB      5$                                          ;
                            51    00  B4  D0 000D8  4$:    MOVL     @0(R4), R1                                  ; 1496
            50              56        52  C1 000DC         ADDL3    J, I, R0                                    ;
            50              50        09  78 000E0         ASHL     #9, R0, R0                                  ;
                                    6041  9F 000E4         PUSHAB   (R0)[R1]                                    ;
                    0000V   CF        01  FB 000E7         CALLS    #1, INVALIDATE                              ;
            E8              52        55  F2 000EC  5$:    AOBLSS   R5, J, 4$                                   ;
                                  88  AA  BF 000F0         CHMU     -120(BASE)                                  ; 1497
                                          04 000F3         RET                                                  ;
                            51        6E  D0 000F4  6$:    MOVL     FOUND_COUNT, R1                             ; 1500
                            50        01  CE 000F7         MNEGL    #1, J                                       ;
                                      07  11 000FA         BRB      8$                                          ;
                       18   A3        08  88 000FC  7$:    BISB2    #8, 24(BFRD)                                ; 1503
                            53        20  C0 00100         ADDL2    #32, BFRD                                   ; 1504
            F5              50        51  F2 00103  8$:    AOBLSS   R1, J, 7$                                   ; 1500
            50              56        09  78 00107  9$:    ASHL     #9, I, R0                                   ; 1509
                            50    00  B4  C0 0010B         ADDL2    @0(R4), R0                                  ;
                                          04 0010F         RET                                                  ; 1511

; Routine Size:  272 bytes,    Routine Base:  $CODE$ + 0176
```

```
1 ROUTINE SERIAL_CACHE : L_JSB NOVALUE =
1
1 !++
1 !
1 ! FUNCTIONAL DESCRIPTION:
1 !
1 ! Serialize cache processing '  queuing the CDRP part of our
1 ! IO packet onto the AQB queue.  Go to sleep if someone else
1 ! is already there.
1 !
1 !--
1
2 BEGIN
2
2 BIND_COMMON;
2
2 EXTERNAL
2         PMS$GL_XQPCACHEWAIT      : ADDRESSING_MODE (GENERAL);
2
2 BUILTIN
2         TESTBITSS,
2         INSQUE;
2
2 LOCAL
2         AQB     : REF BBLOCK,
2         ACB     : REF BBLOCK;
2
2 AQB = .CURRENT_VCB [VCB$L_AQB];
2
2 IF (ACB = .ACB_ADDR) EQL 0
2 THEN
3     BEGIN
3     ACB_ADDR = (ACB = .IO_PACKET + IRP$C_CDRP);
3     ACB [ACB$L_PID] = .CTL$GL_PCB [PCB$L_PID];
3     ACB [ACB$L_AST] = CONTINUE_THREAD;
3     ACB [ACB$L_ASTPRM] = .BASE;
3     ACB [ACB$B_RMOD] = PSL$C_KERNEL + ACB$M_NODELETE;
3     ACB [ACB$B_TYPE] = DYN$C_ACB;
3     ACB [ACB$W_SIZE] = 0;
2     END;
2
2 IF INSQUE (.IO_PACKET, .AQB [AQB$L_ACPQBL])
2 THEN
2     RETURN
2 ELSE
3     BEGIN
3     PMS$GL_XQPCACHEWAIT = .PMS$GL_XQPCACHEWAIT + 1;
3     WAIT_FOR_AST ();
2     END;
2
1 END;


                                        .EXTRN  PMS$GL_XQPCACHEWAIT

                      52  DD 00000 SERIAL_CACHE:
```

```
                                                 PUSHL   R2                                      ; 1512
            50        98  AA  D0 00002           MOVL    -104(BASE), R0                          ; 1539
            52        10  A0  D0 00006           MOVL    16(R0), AQB
            50        C8  AA  D0 0000A           MOVL    -56(BASE), ACB                          ; 1541
                          2B  12 0000E           BNEQ    1$
      50      90  AA 00000060  8F  C1 00010      ADDL3   #96, -112(BASE), ACB                    ; 1544
              C8  AA          50  D0 00019       MOVL    ACB, -56(BASE)
                  51 00000000G 00  D0 0001D      MOVL    CTL$GL_PCB, R1                          ; 1545
              0C  A0      60  A1  D0 00024       MOVL    96(R1), 12(ACB)
              10  A0   0000G  CF  9E 00029       MOVAB   CONTINUE_THREAD, 16(ACB)                ; 1546
              14  A0          5A  D0 0002F       MOVL    BASE, 20(ACB)                           ; 1547
              08  A0 20020000 8F  D0 00033       MOVL    #537001984, 8(ACB)                      ; 1550
              04  B2      90  BA  0E 0003B 1$:   INSQUE  @-112(BASE), @4(AQB)                    ; 1553
                              0B  13 00040       BEQL    2$
                 00000000G    00  D6 00042       INCL    PMS$GL_XQPCACHEWAIT                     ; 1558
          0000G   CF          00  FB 00048       CALLS   #0, WAIT_FOR_AST                        ; 1559
                              04  BA 0004D 2$:   POPR    #^M<R2>                                 ; 1562
                                  05 0004F       RSB                                             ;

; Routine Size:  80 bytes,    Routine Base:  $CODE$ + 0286
```

```
GLOBAL ROUTINE FIND_BUFFER (LBN, TYPE, COUNT, FOUND_COUNT) : L_NORM =

!++
!
! FUNCTIONAL DESCRIPTION:
!
!
! CALLING SEQUENCE:
!
! INPUT PARAMETERS:
!
! IMPLICIT INPUTS:
!
! OUTPUT PARAMETERS:
!
! IMPLICIT OUTPUTS:
!
! ROUTINE VALUE:
!       index of first buffer found
!
! SIDE EFFECTS:
!       LRU list relinked, buffers may be written
!
!--

BEGIN

MAP
        TYPE    : BYTE;

BIND_COMMON;

EXTERNAL
        ACP$GB_MAXREAD          : BYTE ADDRESSING_MODE (ABSOLUTE),
        PMS$GL_FILHDR_HIT       : ADDRESSING_MODE (ABSOLUTE),
        PMS$GL_FILHDR_MISS      : ADDRESSING_MODE (ABSOLUTE),
        PMS$GL_DIRDATA_HIT      : ADDRESSING_MODE (ABSOLUTE),
        PMS$GL_DIRDATA_MISS     : ADDRESSING_MODE (ABSOLUTE),
        PMS$GL_STORAGMAP_HIT    : ADDRESSING_MODE (ABSOLUTE),
        PMS$GL_STORAGMAP_MISS   : ADDRESSING_MODE (ABSOLUTE);

LOCAL
        FNDCNT,
        INDX,
        POOL,
        INDX_ADDR,
        LOCKBASIS,
        SEQNUM,
        PIDINDX         : WORD,
        BFRD            : REF BBLOCK,
        BFRL            : REF BBLOCK,
        POOL_LRU        : REF BBLOCK;

LABEL
        GOT_ONE;

.FOUND_COUNT = 1;
```

```
SERIAL_CACHE ();

! If this is for LBN = -1, we just want a buffer.  Don't look for it,
! because other processes may be doing this also, and we don't want
! to find theirs.
!

IF (.LBN+1) EQL 0
THEN
    INDX = 0
ELSE
    BEGIN

! Get initial index by hashing.  Follow links, if any, until we
! get match on both LBN and UCB, or we run out of links to follow.
!

    INDX = .LOOKUP_LBN (.LBN)<0,16>;

    WHILE .INDX NEQ 0
    DO
        BEGIN

        BFRD = BFRD_ADDR (.INDX);

! Determine if this in fact matches on both LBN and UCB.
!

        IF .BFRD [BFRD$L_LBN] EQL .LBN
            AND .BFRD [BFRD$L_UCB] EQL .CURRENT_UCB
        THEN
            EXITLOOP
        ELSE
            INDX = .BFRD [BFRD$W_NXTBFRD];
        END;
    END;

POOL = .POOL_TABLE [.TYPE];

PIDINDX = .CTL$GL_PCB [PCB$L_PID];

! Validate that we have a lockbasis for the buffer type requested.
! We deliberately do not include DIRINDX_TYPE buffers here, because
! they are not hashed at all and are not handled with this routine.
!

CASE .TYPE FROM 0 TO 5 OF
    SET
    [HEADER_TYPE, DATA_TYPE]:
        BEGIN

        IF .CURR_LCKINDX EQL 0
        THEN
            BUG_CHECK (XQPERR, 'no current lock index lock basis');

        LOCKBASIS = .LB_BASIS [.CURR_LCKINDX];
```

```
        END;

    [DIRECTORY_TYPE]:
        BEGIN
        IF .DIR_LCKINDX EQL 0
        THEN
            BUG_CHECK (XQPERR, 'No dir lock basis');

        LOCKBASIS = .LB_BASIS [.DIR_LCKINDX];
        END;

    [INDEX_TYPE, BITMAP_TYPE, QUOTA_TYPE]:
        IF (LOCKBASIS = .LB_BASIS [0]) EQL 0
        THEN
            BUG_CHECK (XQPERR, 'no allocation lock for lock basis');

    [OUTRANGE]:
        BUG_CHECK (BADBUFTYP, 'Bad buffer type code');

    TES;

! Retrieve the lock basis and current buffer sequence number.  The
! lock basis is the same as used for the synchronization lock, and
! the sequence number was retrieved from the value block of that lock
! when it was raised to synchronize this operation.
! The appropriate lock depends on the type of buffer being requested.
!

CASE .TYPE FROM 0 TO 5 OF
    SET
    [HEADER_TYPE]:
        BEGIN
        IF .LB_HDRSEQ [.CURR_LCKINDX] EQL 0
        THEN
            LB_HDRSEQ [.CURR_LCKINDX] = 1;

        SEQNUM = .LB_HDRSEQ [.CURR_LCKINDX];
        END;

    [DIRECTORY_TYPE]:
        BEGIN
        IF .LB_DATASEQ [.DIR_LCKINDX] EQL 0
        THEN
            LB_DATASEQ [.DIR_LCKINDX] = 1;

        SEQNUM = .LB_DATASEQ [.DIR_LCKINDX];
        END;

    [DATA_TYPE]:
        BEGIN
        IF .LB_DATASEQ [.CURR_LCKINDX] EQL 0
        THEN
            LB_DATASEQ [.CURR_LCKINDX] = 1;

        SEQNUM = .LB_DATASEQ [.CURR_LCKINDX];
        END;
```

```
2 ! The storage bitmap, index file bitmap, and quota file data blocks
2 ! get their sequence numbers from the volume lock value block.
2 ! Check with the code in allocation_lock and allocation_unlock before
2 ! changing any of this.
2 !
2
2     [BITMAP_TYPE]:
3         BEGIN
3         IF .(LB_DATASEQ [0])<0,16,0> EQL 0
3         THEN
3             (LB_DATASEQ [0])<0,16,0> = 1;
3
3         SEQNUM = .(LB_DATASEQ [0])<0,16,0>;
2         END;
2
2     [INDEX_TYPE]:
3         BEGIN
3         IF .(LB_DATASEQ [0])<16,16,0> EQL 0
3         THEN
3             (LB_DATASEQ [0])<16,16,0> = 1;
3
3         SEQNUM = .(LB_DATASEQ [0])<16,16,0>;
2         END;
2
2     [QUOTA_TYPE]:
3         BEGIN
3         SEQNUM = .SAVE_VC_FLAGS<1,15,0>;
3
3         IF .SEQNUM EQL 0
3         THEN
4             BEGIN
4             SEQNUM = .SEQNUM + 1;
4             SAVE_VC_FLAGS<1,15,0> = .SEQNUM;
3             END;
3
2         END;
2
2     TES;
2
2 IF .INDX NEQ 0
2 THEN
2
2 ! We found a buffer that matches the LBN and UCB desired.
2 ! It may either be on our in-process queue (curpid will be us),
2 ! or in the general buffer cache.
2 !
2
2 GOT_ONE:
3     BEGIN
3
3     IF .BFRD [BFRD$W_CURPID] NEQ 0
3     THEN
3         IF .BFRD [BFRD$W_CURPID] EQL .PIDINDX
3         THEN
3
3 ! This is a buffer we've already put onto our in-process queue.
3 ! Move to the head of the LRU list and return.
```

```
3 !
3
4               BEGIN
4               REMQUE (.BFRD, BFRD);
4               INSQUE (.BFRD, .BFR_LIST [.POOL, QBLNK]);
4               RELEASE_CACHE ();
4               RETURN .INDX - 1
4               END
4
3           ELSE
3               bug_check (xqperr,'');
3 !             RESOLVE_AMBIGUITY ();
3
3
3 ! Verify lock basis and that things are in the right pool.
3 !
3 ! Blocks in the data block pool may legitimately have the wrong
3 ! lockbasis as a result of having been deallocated to free storage
3 ! and reallocated to another file.
3 !
3 ! File headers may legitimately have the wrong lockbasis if they
3 ! have been deleted and are being treated as primary headers when
3 ! they used to be extension headers or vice-versa.  This will
3 ! force a read from disk.  Further checks will be made in read_header
3 ! against the actual header to see if things really make sense.
3 !
3 ! Blocks can also cross pools (rare in practice) if directory or quota
3 ! file data blocks, for example, are deallocated to free storage, and
3 ! the index file is then extended, causing them to become file headers.
3 !
3 ! In all cases, unhook and return the bfrd to destroy the association
3 ! of this buffer with the lock basis and lock that is backing it.
3 !
3
3       IF .BFRD [BFRD$L_LOCKBASIS] NEQ .LOCKBASIS
3           OR .BFRD [BFRD$V_POOL] NEQ .POOL
3       THEN
3           IF .POOL EQL 1
3               OR .TYPE EQL HEADER_TYPE
3               OR .BFRD [BFRD$V_POOL] EQL 1
3           THEN
4               BEGIN
4
4               UNHOOK_BFRD (
4                   ((.BFRD - .CACHE_HDR [F11BC$L_BFRDBAS])/BFRD$S_BFRDDEF) + 1,
4                   .BFRD);
4
4               RETURN_BFRD (.BFRD);
4
4               LEAVE GOT_ONE;
4               END
3           ELSE
3               BUG_CHECK (XQPERR, 'invalid lock basis');
3
3 ! This was not on our in-process queue.  We'll need to account for the
3 ! fact that we are taking another buffer out of general circulation for
3 ! this operation.
```

```
3 !
3
3       IF .BFRS_USED [.POOL] EQL .BFR_CREDITS [.POOL]
3       THEN
4           BEGIN
4           BIND POOLAVAIL = CACHE_HDR [F11BC$L_POOLAVAIL] + .POOL*4;
4
4           IF .POOLAVAIL GTR 4
4           THEN
5               BEGIN
5               POOLAVAIL = .POOLAVAIL - 1;
5               BFR_CREDITS [.POOL] = .BFR_CREDITS [.POOL] + 1;
5               END
4           ELSE
4               FREE_ONE (.POOL);
3           END;
3
3       BFRS_USED [.POOL] = .BFRS_USED [.POOL] + 1;
3
3       IF .BFRD [BFRD$L_SEQNUM] NEQ .SEQNUM
3       THEN
3           BFRD [BFRD$V_VALID] = 0;
3
3 ! The sequence number will be stored in the BFRD when the buffer is
3 ! released back to the cache, otherwise it would be stored here.
3 !
3
3       BFRD [BFRD$W_CURPID] = .PIDINDX;
3
3 ! Pull this buffer out of the cache list and insert into our in-process queue.
3 !
3
3       REMQUE (.BFRD, BFRD);
3       INSQUE (.BFRD, .BFR_LIST [.POOL, QBLNK]);
3
3       RELEASE_CACHE ();
3
3 ! Count finding a valid buffer as a hit, an invalid one as a miss.
3 ! We are deliberately not counting hits on buffers already on our
3 ! in-process list to get a more meaningful hit ratio.
3 !
3
3       IF .BFRD [BFRD$V_VALID]
3       THEN
4           BEGIN
4           PMS_TOT_CACHE = .PMS_TOT_CACHE + 1;
4           CASE .POOL FROM 0 TO 2 OF
4           SET
4           [0]:    PMS$GL_STORAGMAP_HIT = .PMS$GL_STORAGMAP_HIT + 1;
4           [1]:    PMS$GL_DIRDATA_HIT = .PMS$GL_DIRDATA_HIT + 1;
4           [2]:    PMS$GL_FILHDR_HIT = .PMS$GL_FILHDR_HIT + 1;
4           TES;
4           END
3       ELSE
3           CASE .POOL FROM 0 TO 2 OF
3           SET
3           [0]:    PMS$GL_STORAGMAP_MISS = .PMS$GL_STORAGMAP_MISS + 1;
```

```
3                [1]:    PMS$GL_DIRDATA_MISS = .PMS$GL_DIRDATA_MISS + 1;
3                [2]:    PMS$GL_FILHDR_MISS = .PMS$GL_FILHDR_MISS + 1;
3                TES;
3
3        RETURN .INDX - 1;
2        END;                            ! of block GOT_ONE
2
2    ! Failed to find a buffer matching desired LBN and UCB in the cache.
2    ! Account for us using another buffer from the cache.
2    !
2
2    FNDCNT = 1;
2
2    IF .BFRS_USED [.POOL] EQL .BFR_CREDITS [.POOL]
2    THEN
3        BEGIN
3        BIND POOLAVAIL = CACHE_HDR [F11BC$L_POOLAVAIL] + .POOL*4;
3
3        IF .POOLAVAIL GTR 4
3        THEN
4            BEGIN
4            POOLAVAIL = .POOLAVAIL - 1;
4            BFR_CREDITS [.POOL] = .BFR_CREDITS [.POOL] + 1;
4            END
3        ELSE
3            FREE_ONE (.POOL);
2        END;
2
2    POOL_LRU = CACHE_HDR [F11BC$Q_POOL_LRU] + .POOL*8;
2
2    REMQUE (.POOL_LRU [QFLNK], BFRD);
2
2    IF .COUNT GTRU 1
2        AND .POOL EQL 1
2    THEN
3        BEGIN
3        LABEL CHK_BFRDS;
3
3        BIND POOLAVAIL = CACHE_HDR [F11BC$L_POOLAVAIL] + 4;         ! Pool 1
3        BIND POOLCNT = CACHE_HDR [F11BC$W_POOLCNT] : VECTOR [,WORD];
3
3        LOCAL
3            DOWN,
3            TRY_COUNT,
3            LO_BFRD,
3            HI_BFRD,
3            CUR_BFRD : REF BBLOCK;
3
3        TRY_COUNT = .ACP$GB_MAXREAD;
3
3        IF .COUNT LSSU .TRY_COUNT
3        THEN
3            TRY_COUNT = .COUNT;
3
3        LO_BFRD = .CACHE_HDR [F11BC$L_BFRDBAS] + (.POOLCNT [0])*BFRD$S_BFRDDEF;
3        HI_BFRD = .LO_BFRD + (.POOLCNT [1])*BFRD$S_BFRDDEF;
```

```

	DOWN = 0;
	CUR_BFRD = .BFRD;

	WHILE .FNDCNT LSSU .TRY_COUNT
	DO
	    CHK_BFRDS:
	    BEGIN

	    IF NOT .DOWN
	    THEN
		BEGIN

		CUR_BFRD = .CUR_BFRD + BFRD$S_BFRDDEF;

		IF .CUR_BFRD GEQA .HI_BFRD
		THEN
		    BEGIN
		    CUR_BFRD = .BFRD;
		    DOWN = 1;
		    LEAVE CHK_BFRDS;
		    END;

		END
	    ELSE
		BEGIN

		CUR_BFRD = .CUR_BFRD - BFRD$S_BFRDDEF;

		IF .CUR_BFRD LSSA .LO_BFRD
		THEN
		    EXITLOOP;

		END;

	    IF .CUR_BFRD [BFRD$W_CURPID] NEQ 0
	    THEN
		EXITLOOP;

! NOTE: FNDCNT has not been bumped yet.

	    INDX = .LOOKUP_LBN ((.LBN + .FNDCNT))<0,16>;

	    WHILE .INDX NEQ 0
	    DO
		BEGIN

		LOCAL TMPBFRD : REF BBLOCK;

		TMPBFRD = BFRD_ADDR (.INDX);

		IF .TMPBFRD [BFRD$L_LBN] EQL (.LBN + .FNDCNT)
		    AND .TMPBFRD [BFRD$L_UCB] EQL .CURRENT_UCB
		THEN
		    EXITLOOP
		ELSE
		    INDX = .TMPBFRD [BFRD$W_NXTBFRD];
```

```
4                       END;
4
4               IF .INDX NEQ 0
4               THEN
4                   EXITLOOP;
4
4               IF .POOLAVAIL GTR 4
4               THEN
5                   BEGIN
5                   POOLAVAIL = .POOLAVAIL - 1;
5                   BFR_CREDITS [1] = .BFR_CREDITS [1] + 1;
5                   END
4               ELSE
4                   EXITLOOP;
4
4       ! This one is ok.  Yank off LRU and count it.
4       !
4
4               REMQUE (.CUR_BFRD, CUR_BFRD);
4               FNDCNT = .FNDCNT + 1;
4
4               IF .CUR_BFRD LSSA .BFRD
4               THEN
4                   BFRD = .CUR_BFRD;
4
3               END;                    ! of block CHK_BFRDS
3
3           .FOUND_COUNT = .FNDCNT;
3
3       ! Set BFRD to the highest one in the range found.
3       !
3
3           BFRD = .BFRD + (.FNDCNT - 1)*BFRD$S_BFRDDEF;
3
2           END;            ! of consider multi-block read
2
2       BFRS_USED [.POOL] = .BFRS_USED [.POOL] + .FNDCNT;
2
2       DECR I FROM (.FNDCNT - 1) TO 0
2       DO
3           BEGIN
3
3           INDX = ((.BFRD - .CACHE_HDR [F11BC$L_BFRDBAS])/BFRD$S_BFRDDEF) + 1;
3
3           UNHOOK_BFRD (.INDX, .BFRD);
3
3       ! Insert the new buffer into the hash list using the new LBN.
3       !
3
3           INDX_ADDR = LOOKUP_LBN ((.LBN + .I));
3           BFRD [BFRD$W_NXTBFRD] = .(.INDX_ADDR)<0,16>;
3           (.INDX_ADDR)<0,16> = .INDX;
3
3       ! Fill in our new or recycled BFRD, as the case may be, and
3       ! insert it onto the appropriate in-process queue.
3       !
3
```

```
     3        BFRD [BFRD$L_LBN] = .LBN + .I;
     3        BFRD [BFRD$L_UCB] = .CURRENT_UCB;
     3        BFRD [BFRD$W_CURPID] = .PIDINDX;
     3
     3        BFRD [BFRD$L_LOCKBASIS] = .LOCKBASIS;
     3        BFRD [BFRD$L_SEQNUM] = .SEQNUM;
     3        BFRD [BFRD$B_BTYPE] = .TYPE;
     3
     3        INSQUE (.BFRD, .BFR_LIST [.POOL, QBLNK]);
     3
     3        BFRD = .BFRD - BFRD$S_BFRDDEF;
     3
     2        END;
     2
     2    RELEASE_CACHE ();
     2
     2    ! We didn't find the buffer in the cache so count a miss.
     2    !
     2
     2    CASE .POOL FROM 0 TO 2 OF
     2    SET
     2    [0]:     PMS$GL_STORAGMAP_MISS = .PMS$GL_STORAGMAP_MISS + 1;
     2    [1]:     PMS$GL_DIRDATA_MISS = .PMS$GL_DIRDATA_MISS + 1;
     2    [2]:     PMS$GL_FILHDR_MISS = .PMS$GL_FILHDR_MISS + 1;
     2    TES;
     2
     2    RETURN .INDX - 1;
     2
     1    END;


                                                .EXTRN   ACP$GB_MAXREAD, PMS$GL_FILHDR_HIT
                                                .EXTRN   PMS$GL_FILHDR_MISS
                                                .EXTRN   PMS$GL_DIRDATA_HIT
                                                .EXTRN   PMS$GL_DIRDATA_MISS
                                                .EXTRN   PMS$GL_STORAGMAP_HIT
                                                .EXTRN   PMS$GL_STORAGMAP_MISS
                                                .EXTRN   BUG$_BADBUFTYP

                           0BFC 00000           .ENTRY   FIND_BUFFER, Save R2,R3,R4,R5,R6,R7,R8,R9,-  ; 1563
                                                         R11                                          ;
            5E          1C  C2 00002            SUBL2    #28, SP                                      ;
            56      EC  AA  9E 00005            MOVAB    -20(BASE), R6                                ; 1591
            58      FC  AA  9E 00009            MOVAB    -4(BASE), R8                                 ;
            55      14  AA  9E 0000D            MOVAB    20(BASE), R5                                 ;
            57    0080  CA  9E 00011            MOVAB    128(BASE), R7                                ;
            52    00A8  CA  9E 00016            MOVAB    168(BASE), R2                                ;
            59    00D4  CA  9E 0001B            MOVAB    212(BASE), R9                                ;
       10   BC          01  D0 00020            MOVL     #1, @FOUND_COUNT                             ; 1619
                        8A  10 00024            BSBB     SERIAL_CACHE                                 ; 1621
   50       04  AC      01  C1 00026            ADDL3    #1, LBN, R0                                  ; 1628
                        04  12 0002B            BNEQ     1$                                           ;
                        5B  D4 0002D            CLRL     INDX                                         ; 1630
                        43  11 0002F            BRB      5$                                           ;
            50          68  D0 00031  1$:       MOVL     (R8), R0                                     ; 1638
            51      14  A0  3C 00034            MOVZWL   20(R0), R1                                   ;
```

```
  7E      00      04  AC       01  7A 00038          EMUL    #1, LBN, #0, -(SP)                 ;
  51      51          8E       51  7B 0003E          EDIV    R1, (SP)+, R1, R1                  ;
                               51  D5 00043          TSTL    R1                                 ;
                               03  18 00045          BGEQ    2$                                 ;
                      51       51  CE 00047          MNEGL   R1, R1                             ;
                      5B   10 B041 3C 0004A 2$:      MOVZWL  @16(R0)[R1], INDX                  ;
                               23  13 0004F 3$:      BEQL    5$                                 ; 1640
                      51       68  D0 00051          MOVL    (R8), R1                           ; 1644
          50          5B       05  78 00054          ASHL    #5, INDX, R0                       ;
                      50   18  A1  C0 00058          ADDL2   24(R1), R0                         ;
                      54   E0  A0  9E 0005C          MOVAB   -32(R0), BFRD                      ;
              04  AC       08  A4  D1 00060          CMPL    8(BFRD), LBN                       ; 1649
                               07  12 00065          BNEQ    4$                                 ;
              94  AA   0C  A4      D1 00067          CMPL    12(BFRD), -108(BASE)               ; 1650
                               06  13 0006C          BEQL    5$                                 ;
                      5B   1E  A4  3C 0006E 4$:      MOVZWL  30(BFRD), INDX                     ; 1654
                               DB  11 00072          BRB     3$                                 ; 1640
                      50   08  AC  9A 00074 5$:      MOVZBL  TYPE, R0                           ; 1658
                      53 FCAD CF40 9A 00078          MOVZBL  POOL_TABLE[R0], POOL               ;
                      51 00000000G 00 D0 0007E       MOVL    CTL$GL_PCB, R1                     ; 1660
              18  AE   60  A1      B0 00085          MOVW    96(R1), PIDINDX                    ;
          05  00           50      8F 0008A          CASEB   R0, #0, #5                         ; 1667
0031     001F     0031   0012         0008E 6$:      .WORD   7$-6$,-                            ;
                  0031   0012         00096                  12$-6$,-                           ;
                                                             9$-6$,-                            ;
                                                             12$-6$,-                           ;
                                                             7$-6$,-                            ;
                                                             12$-6$                             ;
                               FEFF 0009A            BUGW                                       ; 1694
                              0000* 0009C            .WORD   <BUG$_BADBUFTYP!4>                 ;
                               29  11 0009E          BRB     13$                                ;
                               65  D5 000A0 7$:      TSTL    (R5)                               ; 1672
                               04  12 000A2          BNEQ    8$                                 ;
                               FEFF 000A4            BUGW                                       ; 1674
                              0000* 000A6            .WORD   <BUG$_XQPERR!4>                    ;
                      50       65  D0 000A8 8$:      MOVL    (R5), R0                           ; 1676
                               0B  11 000AB          BRB     11$                                ;
                               69  D5 000AD 9$:      TSTL    (R9)                               ; 1681
                               04  12 000AF          BNEQ    10$                                ;
                               FEFF 000B1            BUGW                                       ; 1683
                              0000* 000B3            .WORD   <BUG$_XQPERR!4>                    ;
                      50       69  D0 000B5 10$:     MOVL    (R9), R0                           ; 1685
              10  AE         6740  D0 000B8 11$:     MOVL    (R7)[R0], LOCKBASIS                ;
                               0A  11 000BD          BRB     13$                                ; 1667
              10  AE           67  D0 000BF 12$:     MOVL    (R7), LOCKBASIS                    ; 1689
                               04  12 000C3          BNEQ    13$                                ;
                               FEFF 000C5            BUGW                                       ; 1691
                              0000* 000C7            .WORD   <BUG$_XQPERR!4>                    ;
          05  00       08  AC      8F 000C9 13$:     CASEB   TYPE, #0, #5                       ; 1705
0059     0027     004D   000C         000CE 14$:     .WORD   15$-14$,-                          ;
                  0068   0038         000D6                  22$-14$,-                          ;
                                                             17$-14$,-                          ;
                                                             24$-14$,-                          ;
                                                             19$-14$,-                          ;
                                                             26$-14$                            ;
                      50       65  D0 000DA 15$:     MOVL    (R5), R0                           ; 1709
                    0094 CA40  D5 000DD              TSTL    148(BASE)[R0]                      ;
```

```
                                            06  12 000E2        BNEQ    16$
                          0094 CA40         01  D0 000E4        MOVL    #1, 148(BASE)[R0]                    ; 1711
                                 50         65  D0 000EA 16$:   MOVL    (R5), R0                             ; 1713
                                 57  0094 CA40  D0 000ED        MOVL    148(BASE)[R0], SEQNUM
                                            51  11 000F3        BRB     27$                                  ; 1705
                                 50         69  D0 000F5 17$:   MOVL    (R9), R0                             ; 1718
                                          6240  D5 000F8        TSTL    (R2)[R0]
                                            04  12 000FB        BNEQ    18$
                               6240         01  D0 000FD        MOVL    #1, (R2)[R0]                         ; 1720
                                 50         69  D0 00101 18$:   MOVL    (R9), R0                             ; 1722
                                            0F  11 00104        BRB     21$
                                 50         65  D0 00106 19$:   MOVL    (R5), R0                             ; 1727
                                          6240  D5 00109        TSTL    (R2)[R0]
                                            04  12 0010C        BNEQ    20$
                               6240         01  D0 0010E        MOVL    #1, (R2)[R0]                         ; 1729
                                 50         65  D0 00112 20$:   MOVL    (R5), R0                             ; 1731
                                 57       6240  D0 00115 21$:   MOVL    (R2)[R0], SEQNUM
                                            2B  11 00119        BRB     27$                                  ; 1705
                                            62  B5 0011B 22$:   TSTW    (R2)                                 ; 1742
                                            03  12 0011D        BNEQ    23$
                                 62         01  B0 0011F        MOVW    #1, (R2)                             ; 1744
                                 57         62  3C 00122 23$:   MOVZWL  (R2), SEQNUM                         ; 1746
                                            1F  11 00125        BRB     27$                                  ; 1705
                                      02    A2  B5 00127 24$:   TSTW    2(R2)                                ; 1751
                                            04  12 0012A        BNEQ    25$
                          02     A2         01  B0 0012C        MOVW    #1, 2(R2)                            ; 1753
                                 57   02    A2  3C 00130 25$:   MOVZWL  2(R2), SEQNUM                        ; 1755
                                            10  11 00134        BRB     27$                                  ; 1705
        57      A4    AA         0F         01  EF 00136 26$:   EXTZV   #1, #15, -92(BASE), SEQNUM           ; 1760
                                            08  12 0013C        BNEQ    27$                                  ; 1762
                                            57  D6 0013E        INCL    SEQNUM                               ; 1765
  A4    AA            0F         01         57  F0 00140        INSV    SEQNUM, #1, #15, -92(BASE)           ; 1766
                                            5B  D5 00146 27$:   TSTL    INDX                                 ; 1773
                                            5E  13 00148        BEQL    32$
                                      1C    A4  B5 0014A        TSTW    28(BFRD)                             ; 1784
                                            1D  13 0014D        BEQL    29$
                          18     AE   1C    A4  B1 0014F        CMPW    28(BFRD), PIDINDX                    ; 1786
                                            12  12 00154        BNEQ    28$
                                 54         64  0F 00156        REMQUE  (BFRD), BFRD                         ; 1794
                                 50   D0 AA43   7E 00159        MOVAQ   -48(BASE)[POOL], R0                  ; 1795
                          00     B0         64  0E 0015E        INSQUE  (BFRD), @0(R0)
                                         0000V  30 00162        BSBW    RELEASE_CACHE                        ; 1796
                                          0282  31 00165        BRW     69$                                  ; 1797
                                           FEFF    00168 28$:   BUGW                                         ; 1801
                                          0000*    0016A        .WORD   <BUG$_XQPERR!4>
                          10     AE   10    A4  D1 0016C 29$:   CMPL    16(BFRD), LOCKBASIS                  ; 1825
                                            08  12 00171        BNEQ    30$
        53      18    A4         02         00  ED 00173        CMPZV   #0, #2, 24(BFRD), POOL               ; 1826
                                            34  13 00179        BEQL    34$
                                 01         53  D1 0017B 30$:   CMPL    POOL, #1                             ; 1828
                                            0D  13 0017E        BEQL    31$
                                      08    AC  95 00180        TSTB    TYPE                                 ; 1829
                                            08  13 00183        BEQL    31$
        01      18    A4         02         00  ED 00185        CMPZV   #0, #2, 24(BFRD), #1                 ; 1830
                                            1E  12 0018B        BNEQ    33$
                                            54  DD 0018D 31$:   PUSHL   BFRD                                 ; 1836
                                 50         68  D0 0018F        MOVL    (R8), R0                             ; 1835
```

```
     50          54       18   A0  C3 00192        SUBL3   24(R0), BFRD, R0                        ;
                 50            20  C6 00197        DIVL2   #32, R0                                 ;
                          01   A0  9F 0019A        PUSHAB  1(R0)                                   ;
           0000V CF            02  FB 0019D        CALLS   #2, UNHOOK_BFRD                         ;
                 50            54  D0 001A2        MOVL    BFRD, R0                                ; 1838
                            0000V  30 001A5        BSBW    RETURN_BFRD                             ;
                             0084  31 001A8 32$:   BRW     45$                                     ; 1840
                               FEFF   001AB 33$:   BUGW                                            ; 1843
                              0000*   001AD        .WORD   <BUG$_XQPERR!4>                         ;
               6643       F4 AA43  B1 001AF 34$:   CMPW    -12(BASE)[POOL], (R6)[POOL]             ; 1850
                               1D  12 001B5        BNEQ    36$                                     ;
                 50       00 B843  DE 001B7        MOVAL   @0(R8)[POOL], R0                        ; 1853
                 50       68   A0  9E 001BC        MOVAB   104(R0), R0                             ;
                 04            60  D1 001C0        CMPL    (R0), #4                                ; 1855
                               07  15 001C3        BLEQ    35$                                     ;
                               60  D7 001C5        DECL    (R0)                                    ; 1858
                             6643  B6 001C7        INCW    (R6)[POOL]                              ; 1859
                               08  11 001CA        BRB     36$                                     ; 1855
                               53  DD 001CC 35$:   PUSHL   POOL                                    ; 1862
                            0000V  30 001CE        BSBW    FREE_ONE                                ;
                 5E            04  C0 001D1        ADDL2   #4, SP                                  ;
                          F4 AA43  B6 001D4 36$:   INCW    -12(BASE)[POOL]                         ; 1865
                 57       14   A4  D1 001D8        CMPL    20(BFRD), SEQNUM                        ; 1867
                               04  13 001DC        BEQL    37$                                     ;
            18   A4            08  8A 001DE        BICB2   #8, 24(BFRD)                            ; 1869
            1C   A4       18   AE  B0 001E2 37$:   MOVW    PIDINDX, 28(BFRD)                       ; 1875
                 54            64  0F 001E7        REMQUE  (BFRD), BFRD                            ; 1880
                 50       D0 AA43  7E 001EA        MOVAQ   -48(BASE)[POOL], R0                     ; 1881
            00   B0            64  0E 001EF        INSQUE  (BFRD), @0(R0)                          ;
                            0000V  30 001F3        BSBW    RELEASE_CACHE                           ; 1883
     27          18   A4       03  E1 001F6        BBC     #3, 24(BFRD), 43$                       ; 1890
                       08F0    CA  D6 001FB        INCL    2288(BASE)                              ; 1893
     02          00            53  CF 001FF        CASEL   POOL, #0, #2                            ; 1894
   0016        000E          0006     00203 38$:   .WORD   39$-38$,-                               ;
                                                           40$-38$,-                               ;
                                                           41$-38$                                 ;
                    00000000G  9F  D6 00209 39$:   INCL    @#PMS$GL_STORAGMAP_HIT                  ; 1896
                               0E  11 0020F        BRB     42$                                     ;
                    00000000G  9F  D6 00211 40$:   INCL    @#PMS$GL_DIRDATA_HIT                    ; 1897
                               06  11 00217        BRB     42$                                     ;
                    00000000G  9F  D6 00219 41$:   INCL    @#PMS$GL_FILHDR_HIT                     ; 1898
                             01C8  31 0021F 42$:   BRW     69$                                     ; 1890
     02          00            53  CF 00222 43$:   CASEL   POOL, #0, #2                            ; 1902
   01BE        01B6          01AE     00226 44$:   .WORD   66$-44$,-                               ;
                                                           67$-44$,-                               ;
                                                           68$-44$                                 ;
                             01A5  31 0022C        BRW     66$                                     ; 1904
                 52            01  D0 0022F 45$:   MOVL    #1, FNDCNT                              ; 1917
               6643       F4 AA43  B1 00232        CMPW    -12(BASE)[POOL], (R6)[POOL]             ; 1919
                               1D  12 00238        BNEQ    47$                                     ;
                 50       00 B843  DE 0023A        MOVAL   @0(R8)[POOL], R0                        ; 1922
                 50       68   A0  9E 0023F        MOVAB   104(R0), R0                             ;
                 04            60  D1 00243        CMPL    (R0), #4                                ; 1924
                               07  15 00246        BLEQ    46$                                     ;
                               60  D7 00248        DECL    (R0)                                    ; 1927
                             6643  B6 0024A        INCW    (R6)[POOL]                              ; 1928
                               08  11 0024D        BRB     47$                                     ; 1924
```

```
                            53  DD 0024F 46$:   PUSHL   POOL                                          ; 1931
                         0000V  30 00251        BSBW    FREE_ONE                                      ;
              5E            04  C0 00254        ADDL2   #4, SP                                        ;
              50        00 B843 7E 00257 47$:   MOVAQ   @0(R8)[POOL], POOL_LRU                        ; 1934
              50            28  C0 0025C        ADDL2   #40, POOL_LRU                                 ;
              54       00   B0  0F 0025F        REMQUE  @0(POOL_LRU), BFRD                            ; 1936
              01       0C   AC  D1 00263        CMPL    COUNT, #1                                     ; 1938
                            03  1A 00267        BGTRU   49$                                           ;
                          00E6  31 00269 48$:   BRW     61$                                           ;
              01            53  D1 0026C 49$:   CMPL    POOL, #1                                      ; 1939
                            F8  12 0026F        BNEQ    48$                                           ;
      59      68 0000006C   8F  C1 00271        ADDL3   #108, (R8), R9                                ; 1944
      51      68 00000078   8F  C1 00279        ADDL3   #120, (R8), R1                                ; 1945
              6E 00000000G  9F  9A 00281        MOVZBL  @#ACP$GB_MAXREAD, TRY_COUNT                   ; 1954
              6E       0C   AC  D1 00288        CMPL    COUNT, TRY_COUNT                              ; 1956
                            04  1E 0028C        BGEQU   50$                                           ;
              6E       0C   AC  D0 0028E        MOVL    COUNT, TRY_COUNT                              ; 1958
              50            68  D0 00292 50$:   MOVL    (R8), R0                                      ; 1960
              55            61  3C 00295        MOVZWL  (R1), R5                                      ;
              55            20  C4 00298        MULL2   #32, R5                                       ;
         08   AE        18 B045 9E 0029B        MOVAB   @24(R0)[R5], LO_BFRD                          ;
              51       02   A1  3C 002A1        MOVZWL  2(R1), R1                                     ; 1961
              51            20  C4 002A5        MULL2   #32, R1                                       ;
         14   AE        08 BE41 9E 002A8        MOVAB   @LO_BFRD[R1], HI_BFRD                         ;
                       0C   AE  D4 002AE        CLRL    DOWN                                          ; 1963
              55            54  D0 002B1        MOVL    BFRD, CUR_BFRD                                ; 1964
              6E            52  D1 002B4 51$:   CMPL    FNDCNT, TRY_COUNT                             ; 1966
                            03  1F 002B7        BLSSU   52$                                           ;
                          0089  31 002B9        BRW     60$                                           ;
              12       0C   AE  E8 002BC 52$:   BLBS    DOWN, 54$                                     ; 1971
              55            20  C0 002C0        ADDL2   #32, CUR_BFRD                                 ; 1975
         14   AE            55  D1 002C3        CMPL    CUR_BFRD, HI_BFRD                             ; 1977
                            12  1F 002C7        BLSSU   55$                                           ;
              55            54  D0 002C9        MOVL    BFRD, CUR_BFRD                                ; 1980
         0C   AE            01  D0 002CC        MOVL    #1, DOWN                                      ; 1981
                            E2  11 002D0 53$:   BRB     51$                                           ; 1982
              55            20  C2 002D2 54$:   SUBL2   #32, CUR_BFRD                                 ; 1989
         08   AE            55  D1 002D5        CMPL    CUR_BFRD, LO_BFRD                             ; 1991
                            6A  1F 002D9        BLSSU   60$                                           ;
                       1C   A5  B5 002DB 55$:   TSTW    28(CUR_BFRD)                                  ; 1997
                            65  12 002DE        BNEQ    60$                                           ;
              50            68  D0 002E0        MOVL    (R8), R0                                      ; 2003
         04   AE        04 BC42 9E 002E3        MOVAB   @LBN[FNDCNT], 4(SP)                           ;
              51       14   A0  3C 002E9        MOVZWL  20(R0), R1                                    ;
7E       00   04   AE       01  7A 002ED        EMUL    #1, 4(SP), #0, -(SP)                          ;
51       51        8E       51  7B 002F3        EDIV    R1, (SP)+, R1, R1                             ;
                            51  D5 002F8        TSTL    R1                                            ;
                            03  18 002FA        BGEQ    56$                                           ;
              51            51  CE 002FC        MNEGL   R1, R1                                        ;
              5B        10 B041 3C 002FF 56$:   MOVZWL  @16(R0)[R1], INDX                             ;
                            22  13 00304 57$:   BEQL    59$                                           ; 2005
              51            68  D0 00306        MOVL    (R8), R1                                      ; 2011
      50      5B            05  78 00309        ASHL    #5, INDX, R0                                  ;
              50       18   A1  C0 0030D        ADDL2   24(R1), R0                                    ;
              50            20  C2 00311        SUBL2   #32, TMPBFRD                                  ;
         04   AE       08   A0  D1 00314        CMPL    8(TMPBFRD), 4(SP)                             ; 2013
                            07  12 00319        BNEQ    58$                                           ;
```

```
             94   AA        0C   A0   D1 0031B         CMPL    12(TMPBFRD), -108(BASE)              ; 2014
                                 06   13 00320         BEQL    59$                                  ;
                  5B        1E   A0   3C 00322 58$:    MOVZWL  30(TMPBFRD), INDX                    ; 2018
                                 DC   11 00326         BRB     57$                                  ; 2005
                                 5B   D5 00328 59$:    TSTL    INDX                                 ; 2021
                                 19   12 0032A         BNEQ    60$                                  ;
                  04             69   D1 0032C         CMPL    (R9), #4                             ; 2025
                                 14   15 0032F         BLEQ    60$                                  ;
                                 69   D7 00331         DECL    (R9)                                 ; 2028
                            02   A6   B6 00333         INCW    2(R6)                                ; 2029
                  55             65   0F 00336         REMQUE  (CUR_BFRD), CUR_BFRD                 ; 2037
                                 52   D6 00339         INCL    FNDCNT                               ; 2038
                  54             55   D1 0033B         CMPL    CUR_BFRD, BFRD                       ; 2040
                                 90   1E 0033E         BGEQU   53$                                  ;
                  54             55   D0 00340         MOVL    CUR_BFRD, BFRD                       ; 2042
                                 8B   11 00343         BRB     53$                                  ; 1966
             10   BC             52   D0 00345 60$:    MOVL    FNDCNT, @FOUND_COUNT                 ; 2046
        50        52             05   78 00349         ASHL    #5, FNDCNT, R0                       ; 2051
                  54       E0 A044    9E 0034D         MOVAB   -32(R0)[BFRD], BFRD                  ;
          F4 AA43                52   A0 00352 61$:    ADDW2   FNDCNT, -12(BASE)[POOL]              ; 2055
                                 6B   11 00357         BRB     64$                                  ; 2057
                  50             68   D0 00359 62$:    MOVL    (R8), R0                             ; 2061
        50        54        18   A0   C3 0035C         SUBL3   24(R0), BFRD, R0                     ;
                  50             20   C6 00361         DIVL2   #32, R0                              ;
                  5B        01   A0   9E 00364         MOVAB   1(R0), INDX                          ;
                                 54   DD 00368         PUSHL   BFRD                                 ; 2063
                                 5B   DD 0036A         PUSHL   INDX                                 ;
          0000V   CF             02   FB 0036C         CALLS   #2, UNHOOK_BFRD                      ;
                  51             68   D0 00371         MOVL    (R8), R1                             ; 2068
        50        52        04   AC   C1 00374         ADDL3   LBN, I, R0                           ;
                  55        14   A1   3C 00379         MOVZWL  20(R1), R5                           ;
  7E              00        50   01   7A 0037D         EMUL    #1, R0, #0, -(SP)                    ;
  50              50        8E   55   7B 00382         EDIV    R5, (SP)+, R0, R0                    ;
                                 50   D5 00387         TSTL    R0                                   ;
                                 03   18 00389         BGEQ    63$                                  ;
                  50             50   CE 0038B         MNEGL   R0, R0                               ;
                  56     10 B140      3E 0038E 63$:    MOVAW   @16(R1)[R0], INDX_ADDR               ;
             1E   A4             66   B0 00393         MOVW    (INDX_ADDR), 30(BFRD)                ; 2069
                  66             5B   B0 00397         MOVW    INDX, (INDX_ADDR)                    ; 2070
             08   A4     04 BC42      9E 0039A         MOVAB   @LBN[I], 8(BFRD)                     ; 2076
             0C   A4        94   AA   D0 003A0         MOVL    -108(BASE), 12(BFRD)                 ; 2077
             1C   A4        18   AE   B0 003A5         MOVW    PIPINDX, 28(BFRD)                    ; 2078
             10   A4        10   AE   D0 003AA         MOVL    LOCKBASIS, 16(BFRD)                  ; 2080
             14   A4             57   D0 003AF         MOVL    SEQNUM, 20(BFRD)                     ; 2081
             19   A4        08   AC   90 003B3         MOVB    TYPE, 25(BFRD)                       ; 2082
                  50     D0 AA43      7E 003B8         MOVAQ   -48(BASE)[POOL], R0                  ; 2084
             00   B0             64   0E 003BD         INSQUE  (BFRD), @0(R0)                       ;
                  54             20   C2 003C1         SUBL2   #32, BFRD                            ; 2086
                  92             52   F4 003C4 64$:    SOBGEQ  I, 62$                               ; 2057
                               0000V  30 003C7         BSBW    RELEASE_CACHE                        ; 2090
      02          00             53   CF 003CA         CASEL   POOL, #0, #2                         ; 2095
    0016        000E        0006         003CE 65$:    .WORD   66$-65$,-                            ;
                                                               67$-65$,-                            ;
                                                               68$-65$                              ;
                  00000000G      9F   D6 003D4 66$:    INCL    @#PMS$GL_STORAGMAP_MISS              ; 2097
                                 0E   11 003DA         BRB     69$                                  ;
                  00000000G      9F   D6 003DC 67$:    INCL    @#PMS$GL_DIRDATA_MISS                ; 2098
```

```
                  06  11 003E2        BRB     69$                                ;
       00000000G  9F  D6 003E4 68$:   INCL    @#PMS$GL_FILHDR_MISS               ; 2099
   50         FF  AB  9E 003EA 69$:   MOVAB   -1(R11), R0                        ; 2102
                      04 003EE        RET                                        ; 2104

; Routine Size:  1007 bytes,    Routine Base:  $CODE$ + 02D6
```

```
; 1121        2105  1 ROUTINE WAKE_WAITER (AQB) : NOVALUE =
; 1122        2106  1
; 1123        2107  1 !++
; 1124        2108  1 !
; 1125        2109  1 ! FUNCTIONAL DESCRIPTION:
; 1126        2110  1 !
; 1127        2111  1 ! Remove ourself from cache processing queue.  Wake up
; 1128        2112  1 ! the next process if queue is not empty.
; 1129        2113  1 !
; 1130        2114  1 !--
; 1131        2115  1
; 1132        2116  2 BEGIN
; 1133        2117  2
; 1134        2118  2 MAP
; 1135        2119  2         AQB     : REF BBLOCK;
; 1136        2120  2
; 1137        2121  2 LINKAGE
; 1138        2122  2         L_SCH$QAST = JSB : GLOBAL (PINCL=2, ACB=5) NOPRESERVE (3,4)
; 1139        2123  2                         NOTUSED (6,7,8,9,10,11);
; 1140        2124  2
; 1141        2125  2 GLOBAL REGISTER
; 1142        2126  2         PINCL = 2,
; 1143        2127  2         ACB = 5;
; 1144        2128  2
; 1145        2129  2 EXTERNAL ROUTINE
; 1146        2130  2         SCH$QAST        : L_SCH$QAST ADDRESSING_MODE (ABSOLUTE);
; 1147        2131  2
; 1148        2132  2 ACB = .AQB [AQB$L_ACPQFL] + IRP$C_CDRP; ! first waiter's ACB
; 1149        2133  2 PINCL = PRI$_RESAVL;
; 1150        2134  2 IF NOT SCH$QAST ()
; 1151        2135  2 THEN
; 1152        2136  2     BUG_CHECK (XQPERR, 'Failed to queue ast');
; 1153        2137  2
; 1154        2138  1 END;


                                                                .EXTRN  SCH$QAST

                                          003C 00000 WAKE_WAITER:
                                                                .WORD   Save R2,R3,R4,R5                     ; 2105
               55      04   BC 00000060   8F   C1 00002         ADDL3   #96, @AQB, ACB                       ; 2132
                                 52        02   D0 0000B         MOVL    #2, PINCL                            ; 2133
                            00000000G      9F   16 0000E         JSB     @#SCH$QAST                           ; 2134
                                 04        50   E8 00014         BLBS    R0, 1$
                                             FEFF 00017         BUGW                                         ; 2136
                                            0000* 00019         .WORD   <BUG$_XQPERR!4>
                                               04 0001B 1$:     RET                                          ; 2138

; Routine Size:  28 bytes,    Routine Base:  $CODE$ + 06C5
```

```
1 GLOBAL ROUTINE RELEASE_CACHE : L_JSB NOVALUE =
1
1 !++
1 !
1 ! FUNCTIONAL DESCRIPTION:
1 !
1 ! Remove ourself from cache processing queue.  Wake up
1 ! the next process if queue is not empty.
1 !
1 !--
1
2 BEGIN
2
2 BUILTIN
2         REMQUE;
2
2 BIND_COMMON;
2
2 LOCAL
2         AQB     : REF BBLOCK,
2         DUMMY;
2
2 AQB = .CURRENT_VCB [VCB$L_AQB];
2
2 IF REMQUE (.AQB [QFLNK], DUMMY) EQL 0   ! queue not empty
2 THEN
2     WAKE_WAITER (.AQB);
2
1 END;
```

```
                                   52  DD 00000 RELEASE_CACHE::
                                                        PUSHL   R2                              ; 2139
                      50     98    AA  D0 00002         MOVL    -104(BASE), R0                  ; 2161
                      52     10    A0  D0 00006         MOVL    16(R0), AQB
                      51     00    B2  0F 0000A         REMQUE  @0(AQB), DUMMY                  ; 2163
                                   50  DC 0000E         MOVPSL  R0                              ;
     50           50  02           01  EF 00010         EXTZV   #1, #2, R0, R0                  ;
                                   06  12 00015         BNEQ    1$                              ;
                                   52  DD 00017         PUSHL   AQB                             ; 2165
                  C7  AF           01  FB 00019         CALLS   #1, WAKE_WAITER                 ;
                                   04  BA 0001D 1$:     POPR    #^M<R2>                         ; 2167
                                       05 0001F         RSB                                     ;

; Routine Size:  32 bytes,    Routine Base:  $CODE$ + 06E1
```

```
1 GLOBAL ROUTINE GET_REQD_BFR_CREDITS : L_NORM NOVALUE =
1
1 !++
1 !
1 ! FUNCTIONAL DESCRIPTION:
1 !
1 ! Acquire minimum buffer credits so that this operation may proceed.
1 ! Wait for their availability if necessary.
1 ! Initialize the global cell CACHE_HDR, primarily used by the buffer
1 ! management routines.
1 !
1 !--
1
2 BEGIN
2
2 BIND_COMMON;
2
2 LOCAL
2     AQB     : REF BBLOCK,
2     REQD;
2
2 AQB = .CURRENT_VCB [VCB$L_AQB];
2
2 CACHE_HDR = .AQB [AQB$L_BUFCACHE];
2
2 SERIAL_CACHE ();
2
3 BEGIN
3 BIND
3           POOLAVAIL = CACHE_HDR [F11BC$L_POOLAVAIL] : VECTOR,
3           POOL_WAITQ = CACHE_HDR [F11BC$Q_POOL_WAITQ] : BLOCKVECTOR [,8,BYTE];
3
3 DECR POOL FROM 3 TO 0
3 DO
4     BEGIN
4
4 ! The minimal buffer requirements are:
4 ! 1 for storage bitmap blocks (pool 0)
4 ! 2 for directory data blocks (pool 1)
4 ! 3 for file headers (pool 2)
4 ! 1 for directory index (pool 3)
4 !
4
4
4     REQD = .POOL + 1;
4
4     IF .POOL EQL 3
4     THEN
4         REQD = 1;
4
4     IF (POOLAVAIL [.POOL] = .POOLAVAIL [.POOL] - .REQD) LSS 0
4     THEN
5         BEGIN
5
5 ! Insert our ACB temporarily at the head of the queue.  Then when
5 ! we pull our irp off of it to put on the wait queue, the cache interlock
5 ! queue will not be left empty until we do our release_cache and
```

```
5 ! at that point pull off our temp acb entry.
5 !
5
5        INSQUE (.ACB_ADDR, AQB [AQB$L_ACPQFL]);
5        REMQUE (.IO_PACKET, IO_PACKET);          ! take us out of cache queue
5        INSQUE (.IO_PACKET, .POOL_WAITQ [.POOL, QBLNK]); ! into pool queue
5        RELEASE_CACHE ();
5        WAIT_FOR_AST ();
5
5        IF .AQB [AQB$L_ACPQFL] NEQ .IO_PACKET
5        THEN
5            BUG_CHECK (XQPERR, 'messed up cache interlock queues');
5
4        END;
4
4    BFR_CREDITS [.POOL] = .REQD;
4
3    END;
2 END;          ! of BIND
2
2 RELEASE_CACHE ();
2
1 END;
```

```
                                  007C 00000       .ENTRY  GET_REQD_BFR_CREDITS, Save R2,R3,R4,R5,R6    ; 2168
              50        98   AA   D0 00002         MOVL    -104(BASE), R0                               ; 2189
              54        10   A0   D0 00006         MOVL    16(R0), AQB
         FC   AA        18   A4   D0 0000A         MOVL    24(AQB), -4(BASE)                            ; 2191
                           FB73   30 0000F         BSBW    SERIAL_CACHE                                 ; 2193
    55   FC   AA 00000068    8F   C1 00012         ADDL3   #104, -4(BASE), R5                           ; 2197
    56   FC   AA 00000048    8F   C1 0001B         ADDL3   #72, -4(BASE), R6                            ; 2198
              52             03   D0 00024         MOVL    #3, POOL                                     ; 2200
              53        01   A2   9E 00027 1$:     MOVAB   1(R2), REQD                                  ; 2212
              03             52   D1 0002B         CMPL    POOL, #3                                     ; 2214
                             03   12 0002E         BNEQ    2$
              53             01   D0 00030         MOVL    #1, REQD                                     ; 2216
            6542             53   C2 00033 2$:     SUBL2   REQD, (R5)[POOL]                             ; 2218
                             24   18 00037         BGEQ    3$
              64        C8   BA   0E 00039         INSQUE  @-56(BASE), (AQB)                            ; 2228
         90   AA        90   BA   0F 0003D         REMQUE  @-112(BASE), -112(BASE)                      ; 2229
              50        04 A642   7E 00042         MOVAQ   4(R6)[POOL], R0                              ; 2230
         00   B0        90   BA   0E 00047         INSQUE  @-112(BASE), @0(R0)
                             92   10 0004C         BSBB    RELEASE_CACHE                                ; 2231
      0000G   CF             00   FB 0004E         CALLS   #0, WAIT_FOR_AST                             ; 2232
         90   AA             64   D1 00053         CMPL    (AQB), -112(BASE)                            ; 2234
                             04   13 00057         BEQL    3$
                                FEFF 00059         BUGW                                                 ; 2236
                               0000* 0005B         .WORD   <BUG$_XQPERR!4>
       EC AA42               53   B0 0005D 3$:     MOVW    REQD, -20(BASE)[POOL]                        ; 2240
              C2             52   F4 00062         SOBGEQ  POOL, 1$                                     ; 2200
                           FF78   30 00065         BSBW    RELEASE_CACHE                                ; 2245
                                  04 00068         RET                                                  ; 2247
```

```
; Routine Size:  105 bytes,    Routine Base:  $CODE$ + 0701
```

```
1 GLOBAL ROUTINE RETURN_CREDITS : L_NORM NOVALUE =
1
1 !++
1 !
1 ! FUNCTIONAL DESCRIPTION:
1 !
1 ! This routine returns the in-process buffer credits to the
1 ! buffer pool.  It moves the next waiter in line to the head
1 ! of the cache processing queue, which will cause it to become
1 ! awakened when we release the cache interlock.
1 !
1 !--
1
2 BEGIN
2
2 BIND_COMMON;
2
2 LOCAL
2     POOLAVAIL     : REF VECTOR,
2     AQB           : REF BBLOCK;
2
2 ! This may be called multiple times.   Check if already returned credits.
2 !
2
2 IF .BFR_CREDITS [3] EQL 0
2 THEN RETURN;
2
2 AQB = .CURRENT_VCB [VCB$L_AQB];
2 POOLAVAIL = CACHE_HDR [F11BC$L_POOLAVAIL];
2
2 SERIAL_CACHE ();
2
2 DECR POOL FROM 3 TO 0
2 DO
3     BEGIN
3
3     BIND
3         POOL_WAITQ = CACHE_HDR [F11BC$Q_POOL_WAITQ] + .POOL*8 : BBLOCK;
3
3     IF .BFRS_USED [.POOL] NEQ 0
3     THEN
3         BUG_CHECK (XQPERR, 'buffer use not to zero yet');
3
3     POOLAVAIL [.POOL] = .POOLAVAIL [.POOL] + .BFR_CREDITS [.POOL];
3     BFR_CREDITS [.POOL] = 0;
3
3     IF .POOL_WAITQ [QFLNK] NEQ POOL_WAITQ [QFLNK]
3     THEN
4         BEGIN
4         LOCAL
4             WAITER;
4
4         REMQUE (.POOL_WAITQ [QFLNK], WAITER);
4         INSQUE (.WAITER, .AQB [AQB$L_ACPQFL]);  ! insert AFTER us.
3         END;
3
2     END;
```

```
; 1324          2305  2
; 1325          2306  2  RELEASE_CACHE ();
; 1326          2307  2
; 1327          2308  1  END;



                                    003C 00000          .ENTRY   RETURN_CREDITS, Save R2,R3,R4,R5                   ; 2248
            54          EC     AA     9E 00002          MOVAB    -20(BASE), R4                                      ; 2261
                        06     A4     B5 00006          TSTW     6(R4)                                              ; 2272
                               48     13 00009          BEQL     4$                                                 ;
            50          98     AA     D0 0000B          MOVL     -104(BASE), R0                                     ; 2275
            55          10     A0     D0 0000F          MOVL     16(R0), AQB                                        ;
   52        FC    AA 00000068 8F     C1 00013          ADDL3    #104, -4(BASE), POOLAVAIL                          ; 2276
                            FAFD      30 0001C          BSBW     SERIAL_CACHE                                       ; 2278
            50                 03     D0 0001F          MOVL     #3, POOL                                           ; 2280
            51          FC   BA40     7E 00022 1$:      MOVAQ    @-4(BASE)[POOL], R1                                ; 2285
            51          48     A1     9E 00027          MOVAB    72(R1), R1                                         ;
                        F4   AA40     B5 0002B          TSTW     -12(BASE)[POOL]                                    ; 2287
                               04     13 0002F          BEQL     2$                                                 ;
                             FEFF        00031          BUGW                                                        ; 2289
                            0000*        00033          .WORD    <BUG$_XQPERR!4>                                    ;
            53               6440     3C 00035 2$:      MOVZWL   (R4)[POOL], R3                                     ; 2291
          6240                 53     C0 00039          ADDL2    R3, (POOLAVAIL)[POOL]                              ;
                             6440     B4 0003D          CLRW     (R4)[POOL]                                         ; 2292
            51                 61     D1 00040          CMPL     (R1), R1                                           ; 2294
                               08     13 00043          BEQL     3$                                                 ;
            53          00     B1     0F 00045          REMQUE   @0(R1), WAITER                                     ; 2300
     00     B5                 63     0E 00049          INSQUE   (WAITER), @0(AQB)                                  ; 2301
            D2                 50     F4 0004D 3$:      SOBGEQ   POOL, 1$                                           ; 2280
                             FF24     30 00050          BSBW     RELEASE_CACHE                                      ; 2306
                                      04 00053 4$:      RET                                                         ; 2308

; Routine Size:  84 bytes,    Routine Base:  $CODE$ + 076A
```

```
GLOBAL ROUTINE WRITE_BLOCK (BUFFER) : L_NORM NOVALUE =

!++
!
! FUNCTIONAL DESCRIPTION:
!
!       This routine writes the indicated block back to the disk.
!
! CALLING SEQUENCE:
!       WRITE_BLOCK (ARG1)
!
! INPUT PARAMETERS:
!       ARG1: address of block buffer
!
! IMPLICIT INPUTS:
!       BUFFER DESCRIPTOR ARRAYS
!
! OUTPUT PARAMETERS:
!       NONE
!
! IMPLICIT OUTPUTS:
!       NONE
!
! ROUTINE VALUE:
!       NONE
!
! SIDE EFFECTS:
!       block written
!
!--

BEGIN

LOCAL
        BFRD            : REF BBLOCK,
        STATUS,                                 ! service status of QIO call
        INDX;                                   ! index of buffer

EXTERNAL
        CLU$GL_CLUB     : ADDRESSING_MODE (ABSOLUTE),
        ACP$GB_DATACHK  : BITVECTOR ADDRESSING_MODE (ABSOLUTE);
                                                ! ACP datacheck enable flags

BIND_COMMON;

EXTERNAL LITERAL
        ACP$V_WRITECHK  : UNSIGNED (6); ! write check enable flag

INDX = (.BUFFER - .CACHE_HDR [F11BC$L_BUFBASE])/512 + 1;

IF .INDX<0,16> GTRU .CACHE_HDR [F11BC$W_BFRCNT]
THEN
    BUG_CHECK (BADBUFADR, 'out of this cache, man');

BFRD = BFRD_ADDR (.INDX);

! Stuff the UCB of IO_CHANNEL to go where we want.
```

```
  2 !
  2
  2 IO_CCB [CCB$L_UCB] = .BFRD [BFRD$L_UCB];
  2
  3 BEGIN
  3 LOCAL
  3         PTR             : REF BBLOCK,
  3         SAVE_PRIV       : VECTOR [4];
  3
  3 PTR = .CTL$GL_PCB;
  3 PTR [PCB$W_DIOCNT] = .PTR [PCB$W_DIOCNT] + 1;
  3 PTR [PCB$W_ASTCNT] = .PTR [PCB$W_ASTCNT] + 1;
  3 SAVE_PRIV [0] = .(PTR [PCB$Q_PRIV]);
  3 SAVE_PRIV [1] = .(PTR [PCB$Q_PRIV]+4);
  3 BBLOCK [PTR [PCB$Q_PRIV], PRV$V_LOG_IO] = 1;
  3 BBLOCK [PTR [PCB$Q_PRIV], PRV$V_BYPASS] = 1;
  3 PTR = .CTL$GL_PHD;
  3 SAVE_PRIV [2] = .(PTR [PHD$Q_PRIVMSK]);
  3 SAVE_PRIV [3] = .(PTR [PHD$Q_PRIVMSK]+4);
  3 BBLOCK [PTR [PHD$Q_PRIVMSK], PRV$V_LOG_IO] = 1;
  3 BBLOCK [PTR [PHD$Q_PRIVMSK], PRV$V_BYPASS] = 1;
  3
  3 PMS_TOT_WRITE = .PMS_TOT_WRITE + 1;
P 3 STATUS = $QIO (
P 3                 EFN  = EFN,
P 3                 ASTADR = CONTINUE_THREAD,
P 3                 ASTPRM = .BASE,
P 3                 CHAN = .IO_CHANNEL,
P 3                 FUNC = (IO$_WRITELBLK
P 3                         OR .ACP$GB_DATACHK[ACP$V_WRITECHK]
P 3                         ^ $BITPOSITION (IO$V_DATACHECK)),
P 3                 IOSB = IO_STATUS,
P 3                 P1   = .BUFFER,
P 3                 P2   = 512,
P 3                 P3   = .BFRD [BFRD$L_LBN]
  3                 );
  3
  3 (PTR [PHD$Q_PRIVMSK]) = .SAVE_PRIV [2];
  3 (PTR [PHD$Q_PRIVMSK]+4) = .SAVE_PRIV [3];
  3 PTR = .CTL$GL_PCB;
  3 PTR [PCB$W_DIOCNT] = .PTR [PCB$W_DIOCNT] - 1;
  3 PTR [PCB$W_ASTCNT] = .PTR [PCB$W_ASTCNT] - 1;
  3 (PTR [PCB$Q_PRIV]) = .SAVE_PRIV [0];
  3 (PTR [PCB$Q_PRIV]+4) = .SAVE_PRIV [1];
  3
  2 END;            ! of block defining PTR, SAVE_PRIV
  2
  2 IF NOT .STATUS
  2 THEN IO_STATUS = .STATUS
  2 ELSE WAIT_FOR_AST();
  2
  2 ! Restore CURRENT_UCB to IO_CHANNEL.
  2 !
  2
  2 IO_CCB [CCB$L_UCB] = .CURRENT_UCB;
  2
  2 IF .IO_STATUS
```

```
2 THEN
3     BEGIN
3     LOCAL
3         CLUSTER,
3         LCKINDX;
3
3     BFRD [BFRD$V_DIRTY] = 0;
3
3 ! Now let's find the lock that backs this buffer and bump it's
3 ! sequence number so the rest of the world will know we've modified
3 ! this buffer.
3 !
3
3     LCKINDX = 0;
3
3     DO
4         BEGIN
4
4         IF .BFRD [BFRD$L_LOCKBASIS] EQL .LB_BASIS [.LCKINDX]
4         THEN
4             EXITLOOP;
4
4         LCKINDX = .LCKINDX + 1;
4         END
3     UNTIL .LCKINDX EQL LB_NUM;
3
3     IF .LCKINDX EQL LB_NUM
3     THEN
3         BUG_CHECK (XQPERR, 'no backing lock for dirty buffer');
3
3     CLUSTER = 0;
3
3     IF .BBLOCK [CURRENT_UCB [UCB$L_DEVCHAR2], DEV$V_CLU]
3         AND .CLU$GL_CLUB NEQ 0
3     THEN
3         CLUSTER = 1;
3
3     CASE .BFRD [BFRD$B_BTYPE] FROM 0 TO 5 OF
3     SET
3     [HEADER_TYPE]:
3         IF .CLUSTER
3         THEN
3             LB_HDRSEQ [.LCKINDX] = .LB_HDRSEQ [.LCKINDX] + 1;
3
3     [DIRECTORY_TYPE, DATA_TYPE]:
3         IF .CLUSTER
3         THEN
3             LB_DATASEQ [.LCKINDX] = .LB_DATASEQ [.LCKINDX] + 1;
3
3     [BITMAP_TYPE]:
3         (LB_DATASEQ [0])<0,16,0> = .(LB_DATASEQ [0])<0,16,0> + 1;
3
3     [INDEX_TYPE]:
3         (LB_DATASEQ [0])<16,16,0> = .(LB_DATASEQ [0])<16,16,0> + 1;
3
3     [QUOTA_TYPE]:
3         SAVE_VC_FLAGS<1,15,0> = .SAVE_VC_FLAGS<1,15,0> + 1;
```

```
3
3       TES;
3
3       END
2  ELSE
3       BEGIN
3
3  ! clean up will iterate on write failures if the volume is write locked until
3  ! the best effort clean up has been done. However, if the write error was on the
3  ! storage bitmap, we allocation lock the volume to prevent further damage.
3  !
3
3       IF .BFRD [BFRD$V_POOL] EQL 0        ! this is storage bitmap pool
3           AND SURFACE_ERROR (.IO_STATUS [0])
3       THEN
3           CURRENT_VCB [VCB$V_NOALLOC] = 1;
3
3       INVALIDATE (.BUFFER);
3       ERR_EXIT (.IO_STATUS[0]);
2       END;
2
1  END;                                          ! end of routine WRITE_BLOCK


                                                        .EXTRN  CLU$GL_CLUB, ACP$V_WRITECHK
                                                        .EXTRN  BUG$_BADBUFADR

                               003C 00000               .ENTRY  WRITE_BLOCK, Save R2,R3,R4,R5           ; 2309
              55 00000000G   00   9E 00002               MOVAB   CTL$GL_PCB, R5
              5E          10      C2 00009               SUBL2   #16, SP
              54        00A8  CA  9E 0000C               MOVAB   168(BASE), R4                           ; 2349
     52     04  AC          FC  BA  C3 00011               SUBL3   @-4(BASE), BUFFER, R2                   ; 2357
              52 00000200   8F  C6 00017               DIVL2   #512, R2
                          52      D6 0001E               INCL    INDX
              50          FC  AA  D0 00020               MOVL    -4(BASE), R0                            ; 2359
         16   A0          52      B1 00024               CMPW    INDX, 22(R0)
                          04      1B 00028               BLEQU   1$
                               FEFF 0002A               BUGW                                            ; 2361
                              0000* 0002C               .WORD   <BUG$_BADBUFADR!4>
              50          FC  AA  D0 0002E  1$:          MOVL    -4(BASE), R0                            ; 2363
              52          20      C4 00032               MULL2   #32, R2
              52          18  A0  C0 00035               ADDL2   24(R0), R2
              52          20      C2 00039               SUBL2   #32, BFRD
       FF74   DA          0C  A2  D0 0003C               MOVL    12(BFRD), @-140(BASE)                   ; 2368
              53          65      D0 00042               MOVL    CTL$GL_PCB, PTR                         ; 2375
                          3E  A3  B6 00045               INCW    62(PTR)                                 ; 2376
                          38  A3  B6 00048               INCW    56(PTR)                                 ; 2377
              6E        0084  C3  7D 0004B               MOVQ    132(PTR), SAVE_PRIV                     ; 2378
       0084   C3 20000080   8F  C8 00050               BISL2   #536871040, 132(PTR)                    ; 2381
              53 00000000G   00   D0 00059               MOVL    CTL$GL_PHD, PTR                         ; 2382
         08   AE          63      7D 00060               MOVQ    (PTR), SAVE_PRIV+8                      ; 2383
              63 20000080   8F  C8 00064               BISL2   #536871040, (PTR)                       ; 2386
                        08EC  CA  D6 0006B               INCL    2284(BASE)                              ; 2388
                          7E      7C 0006F               CLRQ    -(SP)                                   ; 2401
                          7E      D4 00071               CLRL    -(SP)
                          08  A2  DD 00073               PUSHL   8(BFRD)
```

```
                        7E      0200    8F  3C 00076        MOVZWL  #512, -(SP)                                  ;
                                  04    AC  DD 0007B        PUSHL   BUFFER                                       ;
                                        5A  DD 0007E        PUSHL   BASE                                         ;
                               0000G    CF  9F 00080        PUSHAB  CONTINUE_THREAD                              ;
                                  88    AA  9F 00084        PUSHAB  -120(BASE)                                   ;
  50 00000000G   9F       01           00G  EF 00087        EXTZV   S^ACP$V_WRITECHK, #1, @#ACP$GB_DATACHK, R0   ;
                 50       50            0E  78 00090        ASHL    #14, R0, R0                                  ;
                 7E       50            20  C9 00094        BISL3   #32, R0, -(SP)                               ;
                                FF78    CA  DD 00098        PUSHL   -136(BASE)                                   ;
                                        1E  DD 0009C        PUSHL   #30                                          ;
           00000000G      00            0C  FB 0009E        CALLS   #12, SYS$QIO                                 ;
                          63      08    AE  7D 000A5        MOVQ    SAVE_PRIV+8, (PTR)                           ; 2403
                          53            65  D0 000A9        MOVL    CTL$GL_PCB, PTR                              ; 2405
                          3E            A3  B7 000AC        DECW    62(PTR)                                      ; 2406
                          38            A3  B7 000AF        DECW    56(PTR)                                      ; 2407
                   0084   C3            6E  7D 000B2        MOVQ    SAVE_PRIV, 132(PTR)                          ; 2408
                          06            50  E8 000B7        BLBS    STATUS, 2$                                   ; 2413
                     88   AA            50  D0 000BA        MOVL    STATUS, -120(BASE)                           ; 2414
                                        05  11 000BE        BRB     3$                                           ;
                  0000G   CF            00  FB 000C0 2$:    CALLS   #0, WAIT_FOR_AST                             ; 2415
                   FF74   DA      94    AA  D0 000C5 3$:    MOVL    -108(BASE), @-140(BASE)                      ; 2420
                          51      88    AA  D0 000CB        MOVL    -120(BASE), R1                               ; 2422
                          6B            51  E9 000CF        BLBC    R1, 14$                                      ;
                     18   A2            04  8A 000D2        BICB2   #4, 24(BFRD)                                 ; 2429
                                        50  D4 000D6        CLRL    LCKINDX                                      ; 2436
              0080 CA40           10    A2  D1 000D8 4$:    CMPL    16(BFRD), 128(BASE)[LCKINDX]                 ; 2441
                                        07  13 000DF        BEQL    5$                                           ;
                                        50  D6 000E1        INCL    LCKINDX                                      ; 2445
                          05            50  D1 000E3        CMPL    LCKINDX, #5                                  ; 2447
                                        F0  12 000E6        BNEQ    4$                                           ;
                          05            50  D1 000E8 5$:    CMPL    LCKINDX, #5                                  ; 2449
                                        04  12 000EB        BNEQ    6$                                           ;
                                      FEFF 000ED            BUGW                                                 ; 2451
                                     0000* 000EF            .WORD   <BUG$_XQPERR!4>                              ;
                                        53  D4 000F1 6$:    CLRL    CLUSTER                                      ; 2453
                          51      94    AA  D0 000F3        MOVL    -108(BASE), R1                               ; 2455
                          0B      3C    A1  E9 000F7        BLBC    60(R1), 7$                                   ;
                           00000000G    9F  D5 000FB        TSTL    @#CLU$GL_CLUB                                ; 2456
                                        03  13 00101        BEQL    7$                                           ;
                          53            01  D0 00103        MOVL    #1, CLUSTER                                  ; 2458
                 05       00      19    A2  8F 00106 7$:    CASEB   25(BFRD), #0, #5                             ; 2460
  001F         0015     001C    000C       0010B 8$:        .WORD   9$-8$,-                                      ;
                        0023    0015       00113                    11$-8$,-                                     ;
                                                                    10$-8$,-                                     ;
                                                                    12$-8$,-                                     ;
                                                                    10$-8$,-                                     ;
                                                                    13$-8$                                       ;
                          60            53  E9 00117 9$:    BLBC    CLUSTER, 18$                                 ; 2463
                                0094  CA40  D6 0011A        INCL    148(BASE)[LCKINDX]                           ; 2465
                                        04 0011F            RET                                                  ; 2463
                          57            53  E9 00120 10$:   BLBC    CLUSTER, 18$                                 ; 2468
                                      6440  D6 00123        INCL    (R4)[LCKINDX]                                ; 2470
                                        04 00126            RET                                                  ; 2468
                                  64    B6 00127 11$:       INCW    (R4)                                         ; 2473
                                        04 00129            RET                                                  ;
                          02      A4    B6 0012A 12$:       INCW    2(R4)                                        ; 2476
                                        04 0012D            RET                                                  ;
```

```
        50      A4   AA              0F      01  EF 0012E 13$:    EXTZV   #1, #15, -92(BASE), R0              ; 2479
    A4  AA               0F          01      50  D6 00134         INCL    R0
                                             50  F0 00136         INSV    R0, #1, #15, -92(BASE)
                                             04  0013C            RET                                          ; 2422
                                     03  18  A2  93 0013D 14$:    BITB    24(BFRD), #3                         ; 2492
                                             09  12 00141         BNEQ    15$
                         000001F4    8F      51  D1 00143         CMPL    R1, #500                             ; 2493
                                             1B  13 0014A         BEQL    16$
                         0000005C    8F      51  D1 0014C 15$:    CMPL    R1, #92
                                             12  13 00153         BEQL    16$
                         000000BC    8F      51  D1 00155         CMPL    R1, #188
                                             09  13 0015C         BEQL    16$
                         00002144    8F      51  D1 0015E         CMPL    R1, #8516
                                             08  12 00165         BNEQ    17$
                                     50  98  AA  D0 00167 16$:    MOVL    -104(BASE), R0                       ; 2495
                               0B    A0      10  88 0016B         BISB2   #16, 11(R0)
                                         04  AC  DD 0016F 17$:    PUSHL   BUFFER                               ; 2497
                            0000V    CF      01  FB 00172         CALLS   #1, INVALIDATE
                                         88  AA  BF 00177         CHMU    -120(BASE)                           ; 2498
                                             04  0017A 18$:       RET                                          ; 2501

; Routine Size:  379 bytes,    Routine Base:  $CODE$ + 07BE
```

```
1 GLOBAL ROUTINE CREATE_BLOCK (LBN, COUNT, TYPE) : L_NORM =
1
1 !++
1 !
1 ! FUNCTIONAL DESCRIPTION:
1 !
1 !       This routine fabricates block buffer(s) containing the designated
1 !       block(s). The type code is as for READ_BLOCK and determines the buffer
1 !       pool to be used.
1 !
1 ! CALLING SEQUENCE:
1 !
1 ! INPUT PARAMETERS:
1 !
1 ! IMPLICIT INPUTS:
1 !
1 ! OUTPUT PARAMETERS:
1 !
1 ! IMPLICIT OUTPUTS:
1 !       NONE
1 !
1 ! ROUTINE VALUE:
1 !       address of buffer
1 !
1 ! SIDE EFFECTS:
1 !       buffer zeroed and recorded as a block read
1 !
1 !--
1
2 BEGIN
2
2 BIND_COMMON;
2
2 LOCAL
2       BUFFER,
2       FOUND_COUNT,
2       BFRD            : REF BBLOCK,
2       INDX0;
2
2 INDX0 = FIND_BUFFER (.LBN, .TYPE, 1, FOUND_COUNT);
2
2 BFRD = .CACHE_HDR [F11BC$L_BFRDBAS] + .INDX0*BFRD$S_BFRDDEF;
2
2 BUFFER = .CACHE_HDR [F11BC$L_BUFBASE] + .INDX0*512;
2 CH$FILL (0, 512, .BUFFER);
2
2 IF (.LBN + 1) NEQ 0
2 THEN
3     BEGIN
3     BFRD [BFRD$V_DIRTY] = 1;
3     BFRD [BFRD$V_VALID] = 1;
3     END;
2
2 RETURN .BUFFER;
2
1 END;                                  ! end of routine CREATE_BLOCK
```

```
                                    00FC 00000        .ENTRY  CREATE_BLOCK, Save R2,R3,R4,R5,R6,R7          ; 2502
                  5E              04  C2 00002        SUBL2   #4, SP                                         ;
                                  5E  DD 00005        PUSHL   SP                                             ; 2541
                                  01  DD 00007        PUSHL   #1                                             ;
                          0C      AC  DD 00009        PUSHL   TYPE                                           ;
                          04      AC  DD 0000C        PUSHL   LBN                                            ;
            F989  CF              04  FB 0000F        CALLS   #4, FIND_BUFFER                                ;
                  52              50  D0 00014        MOVL    R0, INDX0                                      ;
                  51      FC      AA  D0 00017        MOVL    -4(BASE), R1                                   ; 2543
        50        52              05  78 0001B        ASHL    #5, INDX0, R0                                  ;
        56        50      18      A1  C1 0001F        ADDL3   24(R1), R0, BFRD                               ;
        50        52              09  78 00024        ASHL    #9, INDX0, R0                                  ; 2545
        57        50      FC      BA  C1 00028        ADDL3   @-4(BASE), R0, BUFFER                          ;
 0200   8F        00      6E      00  2C 0002D        MOVC5   #0, (SP), #0, #512, (BUFFER)                   ; 2546
                                  67     00034                                                               ;
        50        04      AC      01  C1 00035        ADDL3   #1, LBN, R0                                    ; 2548
                                  04  13 0003A        BEQL    1$                                             ;
                  18      A6      0C  88 0003C        BISB2   #12, 24(BFRD)                                  ; 2552
                          50      57  D0 00040 1$:    MOVL    BUFFER, R0                                     ; 2555
                                      04 00043        RET                                                    ; 2557

; Routine Size:  68 bytes,    Routine Base:  $CODE$ + 0939
```

```
1 GLOBAL ROUTINE UNHOOK_BFRD (INDXARG, BFRDARG) : L_NORM NOVALUE =
1
1 !++
1 !
1 ! FUNCTIONAL DESCRIPTION:
1 !
1 ! This routine extricates a BFRD and its associated BFRL from
1 ! their hash lookup lists.  It also takes care of the BFRL if
1 ! this was the last reference to it.
1 ! The caller of this routine is responsible for clearing or
1 ! rewriting other relevant fields in the BFRD.
1 !
1 !--
1
2 BEGIN
2
2 BIND_COMMON;
2
2 EXTERNAL ROUTINE
2         DEL_EXTFCB      : L_NORM,
2         NUKE_HEAD_FCB   : L_NORM NOVALUE;
2
2 LOCAL
2         BFRD            : REF BBLOCK,
2         INDX            : WORD,
2         INDX_ADDR,
2         BFRL            : REF BBLOCK;
2
2 BFRD = .BFRDARG;
2 INDX = .INDXARG;
2
2 IF .BFRD [BFRD$V_DIRTY]
2 THEN
2     BUG_CHECK (XQPERR, 'Should not find dirty buffer');
2
2 BFRD [BFRD$V_VALID] = 0;
2
2 IF .BFRD [BFRD$B_BTYPE] EQL DIRINDX_TYPE
2 THEN
3     BEGIN
3     LOCAL
3         BUFR    : REF BBLOCK,
3         DIRFCB  : REF BBLOCK;
3
3 ! DIRINDX_TYPE blocks are not hashed by lbn, and the field is used
3 ! to point to the directory fcb.
3
3     IF (DIRFCB = .BFRD [BFRD$L_LBN]) NEQ 0
3     THEN
4         BEGIN
4
4         IF .DIRFCB [FCB$B_TYPE] NEQ DYN$C_FCB
4             OR .DIRFCB [FCB$L_DIRINDX] EQL 0
4         THEN
4             BUG_CHECK (XQPERR, 'not legit dirfcb');
4
4 ! This will break the association of this directory index buffer
```

```
4 ! from the directory fcb.
4 !
4
4          DIRFCB [FCB$L_DIRINDX] = 0;
4
4          IF .DIRFCB [FCB$W_REFCNT] EQL 0
4          THEN
4              IF TESTBITSC (DIRFCB [FCB$V_DIR])
4
4 ! from this point on, because the refcnt was zero, and the DIR
4 ! flag was set and is now clear, the search_fcb routine will no
4 ! longer find this fcb, so we are free to deallocate it.
4 ! Furthermore, because the DIR was set, we know that no search_fcb
4 ! had found it at the time we cleared it because search_fcb will
4 ! unconditionally clear it.
4 !
4              THEN
5                  BEGIN
5
5 ! Note that we are under the cache interlock here, and assume that
5 ! the these routines will do nothing that could cause a stall.
5 !
5
5                  DEL_EXTFCB (.DIRFCB);
5                  NUKE_HEAD_FCB (.DIRFCB);
4                  END;
4
4          BFRD [BFRD$L_LBN] = 0;
4
3          END;                ! of bfrd$l_lbn (dirfcb pointer) neq zero
3
2      END;
2
2 IF .BFRD [BFRD$L_LBN] NEQ 0
2 THEN
2
2 ! BFRD$L_LBN neq 0 means that this buffer is in the hash lookup list.
2 ! We need to remove it from the list it is currently
2 ! in as we are about to use the same buffer for some other LBN.
2 !
2
3      BEGIN
3
3      INDX_ADDR = LOOKUP_LBN (.BFRD [BFRD$L_LBN]);
3
3      IF .(.INDX_ADDR)<0,16> EQL .INDX
3      THEN
3          (.INDX_ADDR)<0,16> = .BFRD [BFRD$W_NXTBFRD]
3      ELSE
4          BEGIN
4          DO
4              INDX_ADDR = BBLOCK [ BFRD_ADDR (.(.INDX_ADDR)<0,16>), BFRD$W_NXTBFRD]
4          UNTIL .(.INDX_ADDR)<0,16> EQL .INDX;
4
4          (.INDX_ADDR)<0,16> = .BFRD [BFRD$W_NXTBFRD];
3          END;
2      END;
```

```
  2
  2 ! If it had a buffer lock hooked up with it, adjust the reference count
  2 ! on the lock descriptor, and remove the lock descriptor from its hash
  2 ! lookup list if this is the last reference to it.
  2 !
  2
  2 IF .BFRD [BFRD$W_BFRL] NEQ 0
  2 THEN
  3     BEGIN
  3
  3     BFRL = BFRLD_ADDR (.BFRD [BFRD$W_BFRL]);
  3
  3     BFRL [BFRL$W_REFCNT] = .BFRL [BFRL$W_REFCNT] - 1;
  3     IF .BFRL [BFRL$W_REFCNT] EQL 0
  3     THEN
  4         BEGIN
  4
  4         IF .BFRL [BFRL$L_LKID] NEQ 0
  4         THEN
  5             BEGIN
  6             IF NOT $DEQ (LKID = .BFRL [BFRL$L_LKID])
  5             THEN
  5                 BUG_CHECK (XQPERR, 'unexpected lock manager reaction');
  4             END;
  4
  4 ! Hash on the lockbasis,parlkid pair and scan the list to find the
  4 ! desired bfrl, if necessary.  Pull it from the list when located.
  4 !
  4
P 4         INDX_ADDR = LOOKUP_LOCK (.BFRL [BFRL$L_LCKBASIS],
  4                                  .BFRL [BFRL$L_PARLKID]);
  4
  4         IF .(.INDX_ADDR)<0,16> EQL .BFRD [BFRD$W_BFRL]
  4         THEN
  4             (.INDX_ADDR)<0,16> = .BFRL [BFRL$W_NXTBFRL]
  4         ELSE
  5             BEGIN
  5
  5             DO
  5                 INDX_ADDR = BBLOCK [ BFRLD_ADDR (.(.INDX_ADDR)<0,16>), BFRL$W_NXTBFRL]
  5             UNTIL .(.INDX_ADDR)<0,16> EQL .BFRD [BFRD$W_BFRL];
  5
  5             (.INDX_ADDR)<0,16> = .BFRL [BFRL$W_NXTBFRL];
  4             END;
  4
  4         BFRL [BFRL$L_LKID] = 0;
  4         BFRL [BFRL$L_LCKBASIS] = 0;
  4         BFRL [BFRL$L_PARLKID] = 0;
  4
  4         BFRL [BFRL$W_NXTBFRL] = .CACHE_HDR [F11BC$W_FREEBFRL];
  4         CACHE_HDR [F11BC$W_FREEBFRL] = .BFRD [BFRD$W_BFRL];
  3         END;
  3
  3 ! There is no longer a BFRL associated with this BFRD.
  3 !
  3
  3     BFRD [BFRD$W_BFRL] = 0;
```

```
; 1751          2729 3
; 1752          2730 2       END;
; 1753          2731 2
; 1754          2732 1 END;


                                                        .EXTRN  DEL_EXTFCB, NUKE_HEAD_FCB
                                                        .EXTRN  SYS$DEQ

                            007C 00000                  .ENTRY  UNHOOK_BFRD, Save R2,R3,R4,R5,R6          ; 2558
             55      FC  AA   9E 00002                  MOVAB   -4(BASE), R5                              ; 2572
             53      08  AC   D0 00006                  MOVL    BFRDARG, BFRD                             ; 2586
             52      04  AC   B0 0000A                  MOVW    INDXARG, INDX                             ; 2587
   04        18      A3   02  E1 0000E                  BBC     #2, 24(BFRD), 1$                          ; 2589
                        FEFF 00013                      BUGW                                              ; 2591
                       0000* 00015                      .WORD   <BUG$_XQPERR!4>                           ;
             18      A3   08  8A 00017 1$:              BICB2   #8, 24(BFRD)                              ; 2593
             06      19   A3  91 0001B                  CMPB    25(BFRD), #6                              ; 2595
                          35  12 0001F                  BNEQ    5$                                        ;
             54      08   A3  D0 00021                  MOVL    8(BFRD), DIRFCB                           ; 2605
                          2F  13 00025                  BEQL    5$                                        ;
             07      0A   A4  91 00027                  CMPB    10(DIRFCB), #7                            ; 2609
                          06  12 0002B                  BNEQ    2$                                        ;
                   00B0   C4  D5 0002D                  TSTL    176(DIRFCB)                               ; 2610
                          04  12 00031                  BNEQ    3$                                        ;
                        FEFF 00033 2$:                  BUGW                                              ; 2612
                       0000* 00035                      .WORD   <BUG$_XQPERR!4>                           ;
                   00B0   C4  D4 00037 3$:              CLRL    176(DIRFCB)                               ; 2618
                     18   A4  B5 0003B                  TSTW    24(DIRFCB)                                ; 2620
                          13  12 0003E                  BNEQ    4$                                        ;
   0E        22      A4   00  E5 00040                  BBCC    #0, 34(DIRFCB), 4$                        ; 2622
                          54  DD 00045                  PUSHL   DIRFCB                                    ; 2638
          0000G      CF   01  FB 00047                  CALLS   #1, DEL_EXTFCB                            ;
                          54  DD 0004C                  PUSHL   DIRFCB                                    ; 2639
          0000G      CF   01  FB 0004E                  CALLS   #1, NUKE_HEAD_FCB                         ;
                     08   A3  D4 00053 4$:              CLRL    8(BFRD)                                   ; 2642
                     08   A3  D5 00056 5$:              TSTL    8(BFRD)                                   ; 2648
                          3D  13 00059                  BEQL    9$                                        ;
             51           65  D0 0005B                  MOVL    (R5), R1                                  ; 2658
             50      14   A1  3C 0005E                  MOVZWL  20(R1), R0                                ;
 7E    00    08      A3   01  7A 00062                  EMUL    #1, 8(BFRD), #0, -(SP)                    ;
 50    50            8E   50  7B 00068                  EDIV    R0, (SP)+, R0, R0                         ;
                          50  D5 0006D                  TSTL    R0                                        ;
                          03  18 0006F                  BGEQ    6$                                        ;
             50           50  CE 00071                  MNEGL   R0, R0                                    ;
             54    10 B140    3E 00074 6$:              MOVAW   @16(R1)[R0], INDX_ADDR                    ;
             52           64  B1 00079                  CMPW    (INDX_ADDR), INDX                         ; 2660
                          16  13 0007C                  BEQL    8$                                        ;
             50           65  D0 0007E 7$:              MOVL    (R5), R0                                  ; 2666
             51           64  3C 00081                  MOVZWL  (INDX_ADDR), R1                           ;
             51           20  C4 00084                  MULL2   #32, R1                                   ;
             51      18   A0  C0 00087                  ADDL2   24(R0), R1                                ;
             54      FE   A1  9E 0008B                  MOVAB   -2(R1), INDX_ADDR                         ;
             52           64  B1 0008F                  CMPW    (INDX_ADDR), INDX                         ; 2667
                          EA  12 00092                  BNEQ    7$                                        ;
             64      1E   A3  B0 00094 8$:              MOVW    30(BFRD), (INDX_ADDR)                     ; 2669
```

```
                                  1A   A3   B5 00098 9$:     TSTW    26(BFRD)                         ; 2678
                                       01   12 0009B         BNEQ    10$                              ;
                                            04 0009D         RET                                      ;
                             51        65   D0 0009E 10$:    MOVL    (R5), R1                         ; 2682
                             50   1A   A3   3C 000A1         MOVZWL  26(BFRD), R0                     ;
                             50        10   C4 000A5         MULL2   #16, R0                          ;
                             50   1C   A1   C0 000A8         ADDL2   28(R1), R0                       ;
                             52   F0   A0   9E 000AC         MOVAB   -16(R0), BFRL                    ;
                             02   A2   B7 000B0              DECW    2(BFRL)                          ; 2684
                                  72   12 000B3              BNEQ    15$                              ; 2685
                             04   A2   D5 000B5              TSTL    4(BFRL)                          ; 2689
                                  15   13 000B8              BEQL    11$                              ;
                                  7E   7C 000BA              CLRQ    -(SP)                            ; 2692
                                  7E   D4 000BC              CLRL    -(SP)                            ;
                             04   A2   DD 000BE              PUSHL   4(BFRL)                          ;
               00000000G     00   04   FB 000C1              CALLS   #4, SYS$DEQ                      ;
                             04   50   E8 000C8              BLBS    R0, 11$                          ;
                                     FEFF 000CB              BUGW                                     ; 2694
                                    0000* 000CD              .WORD   <BUG$_XQPERR!4>                  ;
                             51        65   D0 000CF 11$:    MOVL    (R5), R1                         ; 2702
               50       08   A2   0C   A2   C1 000D2         ADDL3   12(BFRL), 8(BFRL), R0            ;
                             56   24   A1   3C 000D8         MOVZWL  36(R1), R6                       ;
     7E          00          50        01   7A 000DC         EMUL    #1, R0, #0, -(SP)                ;
     50          50          8E        56   7B 000E1         EDIV    R6, (SP)+, R0, R0                ;
                                       50   D5 000E6         TSTL    R0                               ;
                                       03   18 000E8         BGEQ    12$                              ;
                             50        50   CE 000EA         MNEGL   R0, R0                           ;
                             54   20 B140   3E 000ED 12$:    MOVAW   @32(R1)[R0], INDX_ADDR           ;
                        1A   A3        64   B1 000F2         CMPW    (INDX_ADDR), 26(BFRD)            ; 2704
                                       17   13 000F6         BEQL    14$                              ;
                             50        65   D0 000F8 13$:    MOVL    (R5), R0                         ; 2711
                             51        64   3C 000FB         MOVZWL  (INDX_ADDR), R1                  ;
                             51        10   C4 000FE         MULL2   #16, R1                          ;
                             51   1C   A0   C0 00101         ADDL2   28(R0), R1                       ;
                             54   F0   A1   9E 00105         MOVAB   -16(R1), INDX_ADDR               ;
                        1A   A3        64   B1 00109         CMPW    (INDX_ADDR), 26(BFRD)            ; 2712
                                       E9   12 0010D         BNEQ    13$                              ;
                             64        62   B0 0010F 14$:    MOVW    (BFRL), (INDX_ADDR)              ; 2714
                                  04   A2   7C 00112         CLRQ    4(BFRL)                          ; 2717
                                  0C   A2   D4 00115         CLRL    12(BFRL)                         ; 2719
                             50        65   D0 00118         MOVL    (R5), R0                         ; 2721
                             62   26   A0   B0 0011B         MOVW    38(R0), (BFRL)                   ;
                             50        65   D0 0011F         MOVL    (R5), R0                         ; 2722
                        26   A0   1A   A3   B0 00122         MOVW    26(BFRD), 38(R0)                 ;
                                  1A   A3   B4 00127 15$:    CLRW    26(BFRD)                         ; 2728
                                            04 0012A         RET                                      ; 2732

; Routine Size:  299 bytes,    Routine Base:  $CODE$ + 097D
```


```
RD
V0
;
```

```
2733  1 ROUTINE FREE_ONE (POOL) : L_JSB NOVALUE =
2734  1
2735  1 !++
2736  1 !
2737  1 ! FUNCTIONAL DESCRIPTION:
2738  1 !
2739  1 ! This routine tosses a buffer off the in-process list to make room
2740  1 ! for a new one.  This only happens when we already have used up
2741  1 ! the minimum number of buffers for this pool, and there are not
2742  1 ! enough available in the cache to steal more from it.
2743  1 !
2744  1 !--
2745  1
2746  2 BEGIN
2747  2
2748  2 BIND_COMMON;
2749  2
2750  2 LOCAL
2751  2         INDX,
2752  2         POOL_LRU        : REF BBLOCK,
2753  2         FREED           : REF BBLOCK;
2754  2
2755  2 BFRS_USED [.POOL] = .BFRS_USED [.POOL] - 1;
2756  2
2757  2 REMQUE (.BFR_LIST [.POOL, QFLNK], FREED);
2758  2
2759  2 INDX = (.FREED - .CACHE_HDR [F11BC$L_BFRDBAS])/BFRD$S_BFRDDEF;
2760  2
2761  2 IF .FREED [BFRD$V_DIRTY]
2762  2 THEN
2763  3     BEGIN
2764  3     RELEASE_CACHE ();
2765  3     WRITE_BLOCK (.INDX*512 + .CACHE_HDR [F11BC$L_BUFBASE]);
2766  3     SERIAL_CACHE ();
2767  3     END;
2768  2
2769  2 ! Insert at front for now to recycle immediately.
2770  2 !
2771  2
2772  2 POOL_LRU = CACHE_HDR [F11BC$Q_POOL_LRU] + .POOL*8;
2773  2 INSQUE (.FREED, POOL_LRU [QFLNK]);
2774  2 FREED [BFRD$W_CURPID] = 0;
2775  2
2776  1 END;


                       0C  BB 00000 FREE_ONE:
                                            PUSHR   #^M<R2,R3>                        ; 2733
           50      0C  AE  D0 00002         MOVL    POOL, R0                          ; 2755
               F4 AA40  B7 00006            DECW    -12(BASE)[R0]                     ;
           50      0C  AE  D0 0000A         MOVL    POOL, R0                          ; 2757
           51  CC AA40  7E 0000E            MOVAQ   -52(BASE)[R0], R1                 ;
           52      00  B1  0F 00013         REMQUE  @0(R1), FREED                     ;
           50      FC  AA  D0 00017         MOVL    -4(BASE), R0                      ; 2759
```

```
50          52     18   A0  C3 0001B         SUBL3   24(R0), FREED, R0                ;
53          50          20  C7 00020         DIVL3   #32, R0, INDX                    ;
16     18   A2          02  E1 00024         BBC     #2, 24(FREED), 1$                ; 2761
                     FC0D   30 00029         BSBW    RELEASE_CACHE                    ; 2764
53          53          09  78 0002C         ASHL    #9, R3, R3                       ; 2765
            50     FC   BA  D0 00030         MOVL    @-4(BASE), R0                    ;
                      6043  9F 00034         PUSHAB  (R0)[R3]                         ;
       FCDA   CF        01  FB 00037         CALLS   #1, WRITE_BLOCK                  ;
                      F79F  30 0003C         BSBW    SERIAL_CACHE                     ; 2766
            50     0C   AE  D0 0003F  1$:    MOVL    POOL, R0                         ; 2772
            50     FC BA40  7E 00043         MOVAQ   @-4(BASE)[R0], POOL_LRU          ;
            50          28  C0 00048         ADDL2   #40, POOL_LRU                    ;
            60          62  0E 0004B         INSQUE  (FREED), (POOL_LRU)              ; 2773
                   1C   A2  B4 0004E         CLRW    28(FREED)                        ; 2774
                        0C  BA 00051         POPR    #^M<R2,R3>                       ; 2776
                            05 00053         RSB                                      ;

; Routine Size:  84 bytes,    Routine Base:  $CODE$ + 0AA8
```

```
1 GLOBAL ROUTINE RELEASE_LOCKBASIS (LCKINDX) : L_NORM =
1
1 !++
1 !
1 ! FUNCTIONAL DESCRIPTION:
1 !
1 ! This routine scans in-process buffers and prepares them for lowering
1 ! the synchronization lock associated with LOCKBASIS.
1 ! All dirty buffers associated with the given LOCKBASIS must have been
1 ! written to disk previously.  This routine runs under the cache serialization
1 ! interlock, and if a conversion is required, will exit without releasing
1 ! the cache serialization interlock.
1 !
1 !--
1
2 BEGIN
2
2 BIND_COMMON;
2
2 EXTERNAL
2         CLU$GL_CLUB     : ADDRESSING_MODE (GENERAL);
2
2 LOCAL
2         LOCKBASIS,
2         LKID_ADDR,
2         INDX;
2
2 LABEL
2         FIXBFRL,
2         LBMATCH;
2
2 LKID_ADDR = 0;
2 INDX = 0;
2 LOCKBASIS = .LB_BASIS [.LCKINDX];
2
2 SERIAL_CACHE ();
2
2 INCR POOL FROM 0 TO 3
2 DO
3     BEGIN
3     LOCAL
3         I,
3         QHEAD   : REF BBLOCK,
3         BFRL    : REF BBLOCK,
3         BFRD    : REF BBLOCK;
3
3     QHEAD = BFR_LIST [.POOL, QFLNK];
3     BFRD = .QHEAD [QFLNK];
3
3     I = .BFRS_USED [.POOL];
3
3     UNTIL .I EQL 0
3     DO
4         BEGIN
4         LOCAL
4             NXTBFRD;
4
```

```
4               I = .I - 1;
4               NXTBFRD = .BFRD [BFRD$L_QFL];
4
4       ! Check this BFRD for LOCKBASIS match.  Note that we are scanning an
4       ! in-process list which can only contain buffers from a given volume
4       ! or volume set.  Because the LOCKBASIS contains the RVN, it is not
4       ! necessary to make a separate check for a volume set match.
4       ! When the buffer was initially brought into the in-process list, it
4       ! matched correctly on the desired UCB.
4       !
4
4               IF .BFRD [BFRD$L_LOCKBASIS] EQL .LOCKBASIS
4               THEN
5       LBMATCH:    BEGIN
5
5                   LOCAL
5                       POOL_LRU : REF BBLOCK;
5
5                   IF NOT .BFRD [BFRD$V_VALID]
5                   THEN
6                       BEGIN
6                       LOCAL
6                           INDX0;
6
6                       INDX0 = (.BFRD - .CACHE_HDR [F11BC$L_BFRDBAS])/BFRD$S_BFRDDEF;
6
6                       UNHOOK_BFRD (.INDX0 + 1, .BFRD);
6
6                       RETURN_BFRD (.BFRD);
6
6                       LEAVE LBMATCH;
5                       END;
5
5       ! This is a valid buffer to be returned to the cache.
5       !
5
6       FIXBFRL:            BEGIN
6
6                       LOCAL
6                           LBINDX_ADDR,
6                           PARLKID;
6
6                       IF .BFRD [BFRD$V_DIRTY]
6                       THEN
6                           BUG_CHECK (XQPERR, 'should not have dirty buffers now');
6
7                       PARLKID = .(IF .CURRENT_VCB [VCB$W_RVN] EQL 0
7                                   THEN CURRENT_VCB [VCB$L_VOLLKID]
6                                   ELSE CURRENT_RVT [RVT$L_STRUCLKID]);
6
6                       IF .BFRD [BFRD$W_BFRL] NEQ 0
6                       THEN
6
6       ! We're already hooked up to a BFRL.  This means we found it already
6       ! in the cache in the first place.  Nothing needs to be done with
6       ! the BFRL.  However, let's be paranoid and make sure that it is
6       ! hooked up to the correct lock.
```

```
6 !
6
7                 BEGIN
7                 BFRL = BFRLD_ADDR (.BFRD [BFRD$W_BFRL]);
7
7                 IF .BFRL [BFRL$L_LCKBASIS] NEQ .LOCKBASIS
7                     OR .BFRL [BFRL$L_PARLKID] NEQ .PARLKID
7                 THEN
7                     BUG_CHECK (XQPERR, 'wrong lockbasis');
7
7                 LEAVE FIXBFRL;
7                 END
6             ELSE IF .INDX NEQ 0
6             THEN
6
6 ! This BFRD is not hooked up with a BFRL yet, but we've already
6 ! got a BFRL in hand (from an earlier hit) so just hook it up
6 ! and bump the REFCNT in the BFRL.
6 !
7                 BEGIN
7
7                 BFRD [BFRD$W_BFRL] = .INDX;
7                 BFRL [BFRL$W_REFCNT] = .BFRL [BFRL$W_REFCNT] + 1;
7
7                 LEAVE FIXBFRL;
6                 END;
6
6 ! There is no BFRL for this BFRD yet, so let's first see if a BFRL
6 ! for this LOCKBASIS is already in the cache.
6 !
6
6             LBINDX_ADDR = LOOKUP_LOCK (.LOCKBASIS, .PARLKID);
6             INDX = .(.LBINDX_ADDR)<0,16>;
6
6             WHILE .INDX NEQ 0
6             DO
7                 BEGIN
7
7                 BFRL = BFRLD_ADDR (.INDX);
7
7                 IF .BFRL [BFRL$L_LCKBASIS] EQL .LOCKBASIS
7                     AND .BFRL [BFRL$L_PARLKID] EQL .PARLKID
7                 THEN
7                     EXITLOOP;
7
7                 INDX = .BFRL [BFRL$W_NXTBFRL];
6                 END;
6
6             IF .INDX EQL 0
6             THEN
6
6 ! We did not find a BFRL in the cache already, so take one off
6 ! the free list and use it.
6 !
7                 BEGIN
7
7                 IF (INDX = .CACHE_HDR [F11BC$W_FREEBFRL]) EQL 0
```

```
7                         THEN
7                             BUG_CHECK (XQPERR, 'no free lock block');
7
7   ! Take our new BFRL out of the free BFRL list.
7   ! Link it into the front of the hash chain.
7   ! Fill in the lockbasis and parent lock ID fields.
7   !
7
7                         BFRL = BFRLD_ADDR (.INDX);
7                         CACHE_HDR [FT1BC$W_FREEBFRL] = .BFRL [BFRL$W_NXTBFRL];
7                         BFRL [BFRL$W_NXTBFRL] = .(.LBINDX_ADDR)<0,16>;
7                         (.LBINDX_ADDR)<0,16> = .INDX;
7
7                         BFRL [BFRL$L_LCKBASIS] = .LOCKBASIS;
7                         BFRL [BFRL$L_PARLKID] = .PARLKID;
7
7   ! Test for LOCKBASIS of -1.  -1 is used for all buffers backed by the
7   ! volume allocation lock.  Because there is a volume allocation lock
7   ! associated with the VCB at all times, there is no need to separately
7   ! keep a lock on those buffers in the cache.  However, we will keep
7   ! a BFRL around to avoid a lot of special casing, but it's LKID field
7   ! will be zero.
7   !
7
7                         IF (.LOCKBASIS + 1) NEQ 0
7                             AND .BBLOCK [CURRENT_UCB [UCB$L_DEVCHAR2], DEV$V_CLU]
7                             AND .CLU$GL_CLUB NEQ 0
7                         THEN
7                             LKID_ADDR = BFRL [BFRL$L_LKID];
7
6                         END;
6
6   ! Fill in the BFRD field that points to the BFRL.  Bump the REFCNT
6   ! in the BFRL.
6   !
6
6                     BFRD [BFRD$W_BFRL] = .INDX;
6                     BFRL [BFRL$W_REFCNT] = .BFRL [BFRL$W_REFCNT] + 1;
6
5                     END;                            ! of FIXBFRL block
5
5   ! Store the current sequence number from the backing lock for this buffer.
5   ! This must be done here because multiple buffers may be backed by a
5   ! single lock.  If one of them has been modified, the sequence number
5   ! for that lock will have changed, yet unmodified buffers backed by
5   ! the same do not have the same sequence number, even though they
5   ! are still valid.  Hence, stamp all buffers with the current sequence
5   ! number.
5
5                 CASE .BFRD [BFRD$B_BTYPE] FROM 0 TO 6 OF
5                 SET
5                 [HEADER_TYPE]:
5                     BFRD [BFRD$L_SEQNUM] = .LB_HDRSEQ [.LCKINDX];
5
5                 [DIRECTORY_TYPE, DATA_TYPE]:
5                     BFRD [BFRD$L_SEQNUM] = .LB_DATASEQ [.LCKINDX];
5
```

```
5              [DIRINDX_TYPE]:
6                   BEGIN
6                   LOCAL
6                       DIRINDX      : REF BBLOCK FIELD (DIRC);
6
6                   BFRD [BFRD$L_SEQNUM] = .LB_HDRSEQ [.LCKINDX];
6                   DIRINDX = .CACHE_HDR [F11BC$L_BUFBASE]
6                           + 512*(.BFRD - .CACHE_HDR [F11BC$L_BFRDBAS])/BFRD$S_BFRDDEF;
6                   DIRINDX [DIRC$L_DATASEQ] = .LB_DATASEQ [.LCKINDX];
5                   END;
5
5              [BITMAP_TYPE]:
5                   BFRD [BFRD$L_SEQNUM] = .(LB_DATASEQ [0])<0,16,0>;
5
5              [INDEX_TYPE]:
5                   BFRD [BFRD$L_SEQNUM] = .(LB_DATASEQ [0])<16,16,0>;
5
5              [QUOTA_TYPE]:
5                   BFRD [BFRD$L_SEQNUM] = .SAVE_VC_FLAGS<1,15,0>;
5
5              TES;
5
5
5  ! Take the buffer off of the in-process queue and put it back
5  ! on the appropriate pool LRU list in the cache.
5  ! Clear out CURPID in the BFRD to note we aren't using it anymore.
5  !
5
5              REMQUE (.BFRD, BFRD);
5              BFRS_USED [.POOL] = .BFRS_USED [.POOL] - 1;
5              BFRD [BFRD$W_CURPID] = 0;
5              POOL_LRU = CACHE_HDR [F11BC$Q_POOL_LRU] + .POOL*8;
5              INSQUE (.BFRD, .POOL_LRU [QBLNK]);
4              END;                ! of LOCKBASIS match.
4
4          BFRD = .NXTBFRD;
4
3          END;                ! of loop through pool bfrds.
3
3      IF .QHEAD NEQA .BFRD
3      THEN
3          BUG_CHECK (XQPERR, 'screwed up in-process list');
3
2      END;            ! of loop through all pools
2
2  IF .LKID_ADDR EQL 0
2  THEN
2      RELEASE_CACHE ();
2
2  RETURN .LKID_ADDR
2
1  END;
```

```
                                 0BFC 00000          .ENTRY  RELEASE_LOCKBASIS, Save R2,R3,R4,R5,R6,R7,-  ; 2777
                                                             R8,R9,R11                                     ;
            5E              10  C2 00002             SUBL2   #16, SP                                       ;
            57          FC  AA  9E 00005             MOVAB   -4(BASE), R7                                  ; 2792
            5B        00A8  CA  9E 00009             MOVAB   168(BASE), R11                                ;
                        0C  AE  D4 0000E             CLRL    LKID_ADDR                                     ; 2808
                            56  D4 00011             CLRL    INDX                                          ; 2809
            50          04  AC  D0 00013             MOVL    LCKINDX, R0                                   ; 2810
            59        0080 CA40 D0 00017             MOVL    128(BASE)[R0], LOCKBASIS                      ;
                          F76A  30 0001D             BSBW    SERIAL_CACHE                                  ; 2812
                            53  D4 00020             CLRL    POOL                                          ; 2814
       08   AE          CC AA43 7E 00022 1$:         MOVAQ   -52(BASE)[POOL], QHEAD                        ; 2823
            54          08  BE  D0 00028             MOVL    @QHEAD, BFRD                                  ; 2824
       04   AE          F4 AA43 3C 0002C             MOVZWL  -12(BASE)[POOL], I                            ; 2826
                        04  AE  D5 00032 2$:         TSTL    I                                             ; 2828
                            03  12 00035             BNEQ    3$                                            ;
                          019A  31 00037             BRW     29$                                           ;
                        04  AE  D7 0003A 3$:         DECL    I                                             ; 2834
            6E              64  D0 0003D             MOVL    (BFRD), NXTBFRD                               ; 2835
            59          10  A4  D1 00040             CMPL    16(BFRD), LOCKBASIS                           ; 2845
                            20  12 00044             BNEQ    4$                                            ;
  1E   18   A4              03  E0 00046             BBS     #3, 24(BFRD), 5$                              ; 2852
            50              67  D0 0004B             MOVL    (R7), R0                                      ; 2858
  50        54          18  A0  C3 0004E             SUBL3   24(R0), BFRD, R0                              ;
            50              20  C6 00053             DIVL2   #32, INDX0                                    ;
                            54  DD 00056             PUSHL   BFRD                                          ; 2860
                        01  A0  9F 00058             PUSHAB  1(INDX0)                                      ;
     FE21   CF              02  FB 0005B             CALLS   #2, UNHOOK_BFRD                               ;
            50              54  D0 00060             MOVL    BFRD, R0                                      ; 2862
                         0000V  30 00063             BSBW    RETURN_BFRD                                   ;
                          0165  31 00066 4$:         BRW     28$                                           ; 2864
  04   18   A4              02  E1 00069 5$:         BBC     #2, 24(BFRD), 6$                              ; 2876
                              FEFF 0006E             BUGW                                                  ; 2878
                             0000* 00070             .WORD   <BUG$_XQPERR!4>                               ;
            50          98  AA  D0 00072 6$:         MOVL    -104(BASE), R0                                ; 2880
                        0E  A0  B5 00076             TSTW    14(R0)                                        ;
                            06  12 00079             BNEQ    7$                                            ;
            50          7C  A0  9E 0007B             MOVAB   124(R0), R0                                   ; 2881
                            04  11 0007F             BRB     8$                                            ;
            50          9C  AA  D0 00081 7$:         MOVL    -100(BASE), R0                                ; 2882
            55              60  D0 00085 8$:         MOVL    (R0), PARLKID                                 ; 2880
                        1A  A4  B5 00088             TSTW    26(BFRD)                                      ; 2884
                            25  13 0008B             BEQL    11$                                           ;
            50              67  D0 0008D             MOVL    (R7), R0                                      ; 2894
            51          1A  A4  3C 00090             MOVZWL  26(BFRD), R1                                  ;
            51              10  C4 00094             MULL2   #16, R1                                       ;
            51          1C  A0  C0 00097             ADDL2   28(R0), R1                                    ;
            52          F0  A1  9E 0009B             MOVAB   -16(R1), BFRL                                 ;
            59          08  A2  D1 0009F             CMPL    8(BFRL), LOCKBASIS                            ; 2896
                            06  12 000A3             BNEQ    9$                                            ;
            55          0C  A2  D1 000A5             CMPL    12(BFRL), PARLKID                             ; 2897
                            04  13 000A9             BEQL    10$                                           ;
                              FEFF 000AB 9$:         BUGW                                                  ; 2899
                             0000* 000AD             .WORD   <BUG$_XQPERR!4>                               ;
                          00A1  31 000AF 10$:        BRW     19$                                           ; 2901
                            56  D5 000B2 11$:        TSTL    INDX                                          ; 2903
                            48  12 000B4             BNEQ    16$                                           ;
```

```
                              51            67  D0 000B6        MOVL    (R7), R1                                ; 2922
                  50          59            55  C1 000B9        ADDL3   PARLKID, LOCKBASIS, R0                  ;
                              58      24    A1  3C 000BD        MOVZWL  36(R1), R8                              ;
     7E           00          50            01  7A 000C1        EMUL    #1, R0, #0, -(SP)                       ;
     50           50          8E            58  7B 000C6        EDIV    R8, (SP)+, R0, R0                       ;
                                            50  D5 000CB        TSTL    R0                                      ;
                                            03  18 000CD        BGEQ    12$                                     ;
                              50            50  CE 000CF        MNEGL   R0, R0                                  ;
                              58      20 B140   3E 000D2 12$:   MOVAW   a32(R1)[R0], LBINDX_ADDR                ;
                              56            68  3C 000D7        MOVZWL  (LBINDX_ADDR), INDX                     ; 2923
                                            20  13 000DA 13$:   BEQL    15$                                     ; 2925
                              50            67  D0 000DC        MOVL    (R7), R0                                ; 2929
                  51          56            04  78 000DF        ASHL    #4, INDX, R1                            ;
                              51      1C    A0  C0 000E3        ADDL2   28(R0), R1                              ;
                              52      F0    A1  9E 000E7        MOVAB   -16(R1), BFRL                           ;
                              59      08    A2  D1 000EB        CMPL    8(BFRL), LOCKBASIS                      ; 2931
                                            06  12 000EF        BNEQ    14$                                     ;
                              55      0C    A2  D1 000F1        CMPL    12(BFRL), PARLKID                       ; 2932
                                            05  13 000F5        BEQL    15$                                     ;
                              56            62  3C 000F7 14$:   MOVZWL  (BFRL), INDX                            ; 2936
                                            DE  11 000FA        BRB     13$                                     ; 2925
                                            56  D5 000FC 15$:   TSTL    INDX                                    ; 2939
                                            4C  12 000FE 16$:   BNEQ    18$                                     ;
                              50            67  D0 00100        MOVL    (R7), R0                                ; 2947
                              56      26    A0  3C 00103        MOVZWL  38(R0), INDX                            ;
                                            04  12 00107        BNEQ    17$                                     ;
                                          FEFF     00109        BUGW                                            ; 2949
                                         0000*     0010B        .WORD   <BUG$_XQPERR!4>                         ;
                              50            67  D0 0010D 17$:   MOVL    (R7), R0                                ; 2956
                  51          56            04  78 00110        ASHL    #4, INDX, R1                            ;
                              51      1C    A0  C0 00114        ADDL2   28(R0), R1                              ;
                              52      F0    A1  9E 00118        MOVAB   -16(R1), BFRL                           ;
                              51            67  D0 0011C        MOVL    (R7), R1                                ; 2957
                       26     A1            62  B0 0011F        MOVW    (BFRL), 38(R1)                          ;
                              62            68  B0 00123        MOVW    (LBINDX_ADDR), (BFRL)                   ; 2958
                              68            56  B0 00126        MOVW    INDX, (LBINDX_ADDR)                     ; 2959
                       08     A2            59  D0 00129        MOVL    LOCKBASIS, 8(BFRL)                      ; 2961
                       0C     A2            55  D0 0012D        MOVL    PARLKID, 12(BFRL)                       ; 2962
                              50      01    A9  9E 00131        MOVAB   1(R9), R0                               ; 2972
                                            15  13 00135        BEQL    18$                                     ;
                              50      94    AA  D0 00137        MOVL    -108(BASE), R0                          ; 2973
                              0D      3C    A0  E9 0013B        BLBC    60(R0), 18$                             ;
                                   00000000G 00  D5 0013F       TSTL    CLU$GL_CLUB                             ; 2974
                                            05  13 00145        BEQL    18$                                     ;
                       0C     AE      04    A2  9E 00147        MOVAB   4(R2), LKID_ADDR                        ; 2976
                       1A     A4            56  B0 0014C 18$:   MOVW    INDX, 26(BFRD)                          ; 2984
                                      02    A2  B6 00150        INCW    2(BFRL)                                 ; 2985
                              51      14    A4  9E 00153 19$:   MOVAB   20(BFRD), R1                            ; 3000
                       06     00      19    A4  8F 00157        CASEB   25(BFRD), #0, #6                        ; 2997
     0050            001A          004B       000E     0015C 20$:   .WORD   21$-20$,-                           ;
                     0024          0056       001A     00164                24$-20$,-                           ;
                                                                        22$-20$,-                               ;
                                                                        25$-20$,-                               ;
                                                                        22$-20$,-                               ;
                                                                        26$-20$,-                               ;
                                                                        23$-20$                                 ;
                              50      04    AC  D0 0016A 21$:   MOVL    LCKINDX, R0                             ; 3000
```

```
                         61     0094 CA40   D0 0016E          MOVL    148(BASE)[R0], (R1)                ;
                                       42   11 00174          BRB     27$                                ;
                         50       04   AC   D0 00176  22$:    MOVL    LCKINDX, R0                        ; 3003
                         61          6B40   D0 0017A          MOVL    (R11)[R0], (R1)                    ;
                                       38   11 0017E          BRB     27$                                ;
                         50       04   AC   D0 00180  23$:    MOVL    LCKINDX, R0                        ; 3010
                         61     0094 CA40   D0 00184          MOVL    148(BASE)[R0], (R1)                ;
                         50            67   D0 0018A          MOVL    (R7), R0                           ; 3011
            51           54       18   A0   C3 0018D          SUBL3   24(R0), BFRD, R1                   ; 3012
            51           51            09   78 00192          ASHL    #9, R1, R1                         ;
                         51            20   C6 00196          DIVL2   #32, R1                            ;
                         51            60   C0 00199          ADDL2   (R0), DIRINDX                      ;
                         50       04   AC   D0 0019C          MOVL    LCKINDX, R0                        ; 3013
                  08     A1          6B40   D0 001A0          MOVL    (R11)[R0], 8(DIRINDX)              ;
                                       11   11 001A5          BRB     27$                                ; 2997
                         61            6B   3C 001A7  24$:    MOVZWL  (R11), (R1)                        ; 3017
                                       0C   11 001AA          BRB     27$                                ;
                         61       02   AB   3C 001AC  25$:    MOVZWL  2(R11), (R1)                       ; 3020
                                       06   11 001B0          BRB     27$                                ;
   61      A4    AA      0F            01   EF 001B2  26$:    EXTZV   #1, #15, -92(BASE), (R1)           ; 3023
                         54            64   0F 001B8  27$:    REMQUE  (BFRD), BFRD                       ; 3033
                                  F4 AA43   B7 001BB          DECW    -12(BASE)[POOL]                    ; 3034
                                  1C   A4   B4 001BF          CLRW    28(BFRD)                           ; 3035
                         50       00 B743   7E 001C2          MOVAQ   @0(R7)[POOL], POOL_LRU             ; 3036
                         50            28   C0 001C7          ADDL2   #40, POOL_LRU                      ;
                  04     B0            64   0E 001CA          INSQUE  (BFRD), @4(POOL_LRU)               ; 3037
                         54            6E   D0 001CE  28$:    MOVL    NXTBFRD, BFRD                      ; 3040
                                     FE5E   31 001D1          BRW     2$                                 ; 2828
                         54       08   AE   D1 001D4  29$:    CMPL    QHEAD, BFRD                        ; 3044
                                       04   13 001D8          BEQL    30$                                ;
                                          FEFF 001DA          BUGW                                       ; 3046
                                         0000* 001DC          .WORD   <BUG$_XQPERR!4>                    ;
 FE3E           53       01            03   F1 001DE  30$:    ACBL    #3, #1, POOL, 1$                   ; 2814
                                  0C   AE   D5 001E4          TSTL    LKID_ADDR                          ; 3050
                                       03   12 001E7          BNEQ    31$                                ;
                                     F9F9   30 001E9          BSBW    RELEASE_CACHE                      ; 3052
                         50       0C   AE   D0 001EC  31$:    MOVL    LKID_ADDR, R0                      ; 3054
                                            04 001F0          RET                                        ; 3056

; Routine Size:  497 bytes,    Routine Base:  $CODE$ + 0AFC
```

```
GLOBAL ROUTINE RESET_LBN (BUFFER, LBN) : L_NORM NOVALUE =

!++
!
! FUNCTIONAL DESCRIPTION:
!
!       This routine changes the resident LBN of the indicated block
!       and marks it as dirty and valid.
!       It is assumed this will be followed soon by a buffer write
!       operation for the new lbn.
!
! CALLING SEQUENCE:
!       RESET_LBN (ARG1, ARG2)
!
! INPUT PARAMETERS:
!       ARG1: address of block buffer
!       ARG2: new LBN
!
! IMPLICIT INPUTS:
!       buffer descriptor arrays
!
! OUTPUT PARAMETERS:
!       NONE
!
! IMPLICIT OUTPUTS:
!       NONE
!
! ROUTINE VALUE:
!       NONE
!
! SIDE EFFECTS:
!       backing LBN for buffer altered
!       buffer marked as dirty and valid.
!
!--

BEGIN

BIND_COMMON;

LOCAL
    INDX,
    BFRD        : REF BBLOCK,
    INDX_ADDR;

INDX = (.BUFFER - .CACHE_HDR [F11BC$L_BUFBASE])/512 + 1;

BFRD = BFRD_ADDR (.INDX);

SERIAL_CACHE ();

IF .BFRD [BFRD$L_LBN] NEQ 0
THEN

! BFRD$L_LBN neq 0 means that this buffer is in the hash lookup list.
! We need to remove it from the list it is currently
! in prior to resetting the LBN field.
```

```
2 !
2
3     BEGIN
3
3     INDX_ADDR = LOOKUP_LBN (.BFRD [BFRD$L_LBN]);
3
3     IF .(.INDX_ADDR)<0,16> EQL .INDX
3     THEN
3         (.INDX_ADDR)<0,16> = .BFRD [BFRD$W_NXTBFRD]
3     ELSE
4         BEGIN
4         DO
4             INDX_ADDR = BBLOCK [ BFRD_ADDR (.(.INDX_ADDR)<0,16>), BFRD$W_NXTBFRD]
4         UNTIL .(.INDX_ADDR)<0,16> EQL .INDX;
4
4         (.INDX_ADDR)<0,16> = .BFRD [BFRD$W_NXTBFRD];
3         END;
2     END;
2
2 ! Store the desired LBN and hook it up in the LBN hash list.
2 !
2
2 BFRD [BFRD$L_LBN] = .LBN;
2
2 INDX_ADDR = LOOKUP_LBN (.LBN);
2 BFRD [BFRD$W_NXTBFRD] = .(.INDX_ADDR)<0,16>;
2 (.INDX_ADDR)<0,16> = .INDX;
2
2 BFRD [BFRD$V_VALID] = 1;
2 BFRD [BFRD$V_DIRTY] = 1;
2
2 RELEASE_CACHE ();
2
1 END;                                    ! end of routine RESET_LBN


                                  003C 00000        .ENTRY  RESET_LBN, Save R2,R3,R4,R5        ; 3057
                   55       FC   AA   9E 00002      MOVAB   -4(BASE), R5                       ; 3093
       52     04   AC       00   B5   C3 00006      SUBL3   @0(R5), BUFFER, R2                 ; 3102
                   52 00000200   8F   C6 0000C      DIVL2   #512, R2                           ;
                   53       01   A2   9E 00013      MOVAB   1(R2), INDX                        ;
                   50            65   D0 00017      MOVL    (R5), R0                           ; 3104
       52          53            05   78 0001A      ASHL    #5, INDX, R2                       ;
                   52       18   A0   C0 0001E      ADDL2   24(R0), R2                         ;
                   52            20   C2 00022      SUBL2   #32, BFRD                          ;
                              F571    30 00025      BSBW    SERIAL_CACHE                       ; 3106
                            08   A2   D5 00028      TSTL    8(BFRD)                            ; 3108
                                 41   13 0002B      BEQL    4$                                 ;
                   51            65   D0 0002D      MOVL    (R5), R1                           ; 3118
                   50       14   A1   3C 00030      MOVZWL  20(R1), R0                         ;
 7E    00     08   A2            01   7A 00034      EMUL    #1, 8(BFRD), #0, -(SP)             ;
 50    50          8E            50   7B 0003A      EDIV    R0, (SP)+, R0, R0                  ;
                                 50   D5 0003F      TSTL    R0                                 ;
                                 03   18 00041      BGEQ    1$                                 ;
```

```
                          50        50  CE 00043        MNEGL   R0, R0                                  ;
                          50  10 B140  3E 00046 1$:     MOVAW   @16(R1)[R0], INDX_ADDR                  ;
    53          60        10        00  ED 0004B        CMPZV   #0, #16, (INDX_ADDR), INDX              ; 3120
                                    18  13 00050        BEQL    3$                                      ;
                          51        65  D0 00052 2$:    MOVL    (R5), R1                                ; 3126
                          54        60  3C 00055        MOVZWL  (INDX_ADDR), R4                         ;
                          54        20  C4 00058        MULL2   #32, R4                                 ;
                          54    18  A1  C0 0005B        ADDL2   24(R1), R4                              ;
                          50    FE  A4  9E 0005F        MOVAB   -2(R4), INDX_ADDR                       ;
    53          60        10        00  ED 00063        CMPZV   #0, #16, (INDX_ADDR), INDX              ; 3127
                                    E8  12 00068        BNEQ    2$                                      ;
                          60    1E  A2  B0 0006A 3$:    MOVW    30(BFRD), (INDX_ADDR)                   ; 3129
                    08    A2    08  AC  D0 0006E 4$:    MOVL    LBN, 8(BFRD)                            ; 3136
                          54        65  D0 00073        MOVL    (R5), R4                                ; 3138
                          51    14  A4  3C 00076        MOVZWL  20(R4), R1                              ;
    7E          00  08    AC        01  7A 0007A        EMUL    #1, LBN, #0, -(SP)                      ;
    51          51        8E        51  7B 00080        EDIV    R1, (SP)+, R1, R1                       ;
                                    51  D5 00085        TSTL    R1                                      ;
                                    03  18 00087        BGEQ    5$                                      ;
                          51        51  CE 00089        MNEGL   R1, R1                                  ;
                          50  10 B441  3E 0008C 5$:     MOVAW   @16(R4)[R1], INDX_ADDR                  ;
                    1E    A2        60  B0 00091        MOVW    (INDX_ADDR), 30(BFRD)                   ; 3139
                          60        53  B0 00095        MOVW    INDX, (INDX_ADDR)                       ; 3140
                    18    A2        0C  88 00098        BISB2   #12, 24(BFRD)                           ; 3143
                             F955  30 0009C             BSBW    RELEASE_CACHE                           ; 3145
                                    04 0009F            RET                                             ; 3147

; Routine Size:  160 bytes,     Routine Base:  $CODE$ + 0CED
```

```
1 GLOBAL ROUTINE MARK_DIRTY (BUFFER) : L_NORM NOVALUE =
1
1 !++
1 !
1 ! FUNCTIONAL DESCRIPTION:
1 !
1 !       This routine marks the indicated buffer for write-back.
1 !
1 ! CALLING SEQUENCE:
1 !       MARK_DIRTY (ARG1)
1 !
1 ! INPUT PARAMETERS:
1 !       ARG1: address of block buffer
1 !
1 ! IMPLICIT INPUTS:
1 !       CURRENT_VCB: VCB of volume
1 !
1 ! OUTPUT PARAMETERS:
1 !       NONE
1 !
1 ! IMPLICIT OUTPUTS:
1 !       NONE
1 !
1 ! ROUTINE VALUE:
1 !       NONE
1 !
1 ! SIDE EFFECTS:
1 !       buffer marked for write-back
1 !
1 !--
1
2 BEGIN
2
2 BIND_COMMON;
2
2 LOCAL
2       BFRD    : REF BBLOCK,
2       INDX0;
2
2 INDX0 = (.BUFFER - .CACHE_HDR [F11BC$L_BUFBASE])/512;
2
2 IF .INDX0<0,16> GEQU .CACHE_HDR [F11BC$W_BFRCNT]
2 THEN
2    BUG_CHECK (BADBUFADR, 'buffer addr out of range');
2
2 BFRD = .CACHE_HDR [F11BC$L_BFRDBAS] + .INDX0*BFRD$S_BFRDDEF;
2
2 IF NOT .BFRD [BFRD$V_VALID] OR .BFRD [BFRD$W_CURPID] EQL 0
2 THEN
2    BUG_CHECK (XQPERR, 'expect it to be valid and in-process');
2
2 BFRD [BFRD$V_DIRTY] = 1;
2
1 END;                                  ! end of routine MARK_DIRTY
```

```
                                     0000 00000       .ENTRY  MARK_DIRTY, Save nothing                ; 3148
      50     04   AC         FC   BA   C3 00002       SUBL3   @-4(BASE), BUFFER, R0                   ; 3187
                  50  00000200   8F   C6 00008       DIVL2   #512, INDX0                             ;
                  51         FC   AA   D0 0000F       MOVL    -4(BASE), R1                            ; 3189
             16   A1              50   B1 00013       CMPW    INDX0, 22(R1)                           ;
                                  04   1F 00017       BLSSU   1$                                      ;
                                     FEFF 00019       BUGW                                            ; 3191
                                    0000* 0001B       .WORD   <BUG$_BADBUFADR!4>                      ;
                  51         FC   AA   D0 0001D 1$:   MOVL    -4(BASE), R1                            ; 3193
                  50              20   C4 00021       MULL2   #32, R0                                 ;
                  50         18   A1   C0 00024       ADDL2   24(R1), BFRD                            ;
      05     18   A0              03   E1 00028       BBC     #3, 24(BFRD), 2$                        ; 3195
                             1C   A0   B5 0002D       TSTW    28(BFRD)                                ;
                                  04   12 00030       BNEQ    3$                                      ;
                                     FEFF 00032 2$:   BUGW                                            ; 3197
                                    0000* 00034       .WORD   <BUG$_XQPERR!4>                         ;
             18   A0              04   88 00036 3$:   BISB2   #4, 24(BFRD)                            ; 3199
                                       04 0003A       RET                                             ; 3201

; Routine Size:  59 bytes,    Routine Base:  $CODE$ + 0D8D
```

```
1 ROUTINE RETURN_BFRD (BFRD) : L_JSB_1ARG NOVALUE =
1
1 !++
1 !
1 ! This routine completes the return of the given BFRD to the appropriate
1 ! pool in the cache.  It expects that it has already been unhooked from
1 ! the hash lists and the BFRL taken care of.
1 !
1 !--
1
2 BEGIN
2
2 MAP
2     BFRD : REF BBLOCK;
2
2 BIND_COMMON;
2
2 LOCAL
2     POOL,
2     POOL_LRU      : REF BBLOCK;
2
2 POOL = .BFRD [BFRD$V_POOL];
2 POOL_LRU = CACHE_HDR [F11BC$Q_POOL_LRU] + .POOL*8;
2
2 ! Credit our buffers in use count if this was on our in-process list.
2 !
2
2 IF .BFRD [BFRD$W_CURPID] NEQ 0
2 THEN
2     BFRS_USED [.POOL] = .BFRS_USED [.POOL] - 1;
2
2 BFRD [BFRD$L_LBN] = 0;
2 BFRD [BFRD$L_UCB] = 0;
2 BFRD [BFRD$L_LOCKBASIS] = 0;
2 BFRD [BFRD$L_SEQNUM] = 0;
2 BFRD [BFRD$W_CURPID] = 0;
2 BFRD [BFRD$W_NXTBFRD] = 0;
2
2 ! Remove from either the in-process list or it's current position in
2 ! the cache LRU (whether curpid was zero or not) and move to the head
2 ! of the cache LRU list.
2 !
2
2 REMQUE (.BFRD, BFRD);
2 INSQUE (.BFRD, POOL_LRU [QFLNK]);
2
1 END;



                                    52  DD 00000 RETURN_BFRD:
                                                 PUSHL    R2                                        ; 3202
  51      18  A0        02          00  EF 00002 EXTZV    #0, #2, 24(BFRD), POOL                    ; 3224
                        52     FC BA41  7E 00008 MOVAQ    @-4(BASE)[POOL], POOL_LRU                 ; 3225
```

```
          52            28  C0 0000D        ADDL2   #40, POOL_LRU                 ;
                   1C   A0  B5 00010        TSTW    28(BFRD)                      ; 3230
                        04  13 00013        BEQL    1$                            ;
                   F4 AA41  B7 00015        DECW    -12(BASE)[POOL]               ; 3232
                   08   A0  7C 00019 1$:    CLRQ    8(BFRD)                       ; 3234
                   10   A0  7C 0001C        CLRQ    16(BFRD)                      ; 3236
                   1C   A0  D4 0001F        CLRL    28(BFRD)                      ; 3238
          50            60  0F 00022        REMQUE  (BFRD), BFRD                  ; 3246
          62            60  0E 00025        INSQUE  (BFRD), (POOL_LRU)            ; 3247
                        04  BA 00028        POPR    #^M<R2>                       ; 3249
                            05 0002A        RSB                                   ;

; Routine Size:  43 bytes,    Routine Base:  $CODE$ + 0DC8
```

```
GLOBAL ROUTINE INVALIDATE (BUFFER) : L_NORM NOVALUE =

!++
!
! FUNCTIONAL DESCRIPTION:
!
!       This routine invalidates the indicated buffer.
!
! CALLING SEQUENCE:
!       INVALIDATE (ARG1)
!
! INPUT PARAMETERS:
!       ARG1: address of block buffer
!
! IMPLICIT INPUTS:
!       NONE
!
! OUTPUT PARAMETERS:
!       NONE
!
! IMPLICIT OUTPUTS:
!       NONE
!
! ROUTINE VALUE:
!       NONE
!
! SIDE EFFECTS:
!       buffer contents forgotten
!
!--

BEGIN

BIND_COMMON;

LOCAL
        POOL,
        BFRD    : REF BBLOCK;

BFRD = ((.BUFFER - .CACHE_HDR [F11BC$L_BUFBASE])/512)*BFRD$S_BFRDDEF
        + .CACHE_HDR [F11BC$L_BFRDBAS];

BFRD [BFRD$V_DIRTY] = 0;
BFRD [BFRD$V_VALID] = 0;

IF .BFRD [BFRD$W_CURPID] NEQ .(CTL$GL_PCB [PCB$L_PID])<0,16>
THEN
    BUG_CHECK (XQPERR, 'expected this to be in-process');

! Pull this buffer out of it's current position and place it at the
! front of the list so it will be the next buffer tossed off of our
! in-process list.
!

POOL = .BFRD [BFRD$V_POOL];
REMQUE (.BFRD, BFRD);
INSQUE (.BFRD, BFR_LIST [.POOL, QFLNK]);
```

```
; 2335          3307  2
; 2336          3308  2 IF .BUFFER EQL .BITMAP_BUFFER
; 2337          3309  2 THEN
; 2338          3310  2     BITMAP_VBN = 0;
; 2339          3311  2
; 2340          3312  1 END;                                       ! end of routine INVALIDATE



                                      0000 00000          .ENTRY  INVALIDATE, Save nothing                  ; 3250
                       50       FC  AA  D0 00002          MOVL    -4(BASE), R0                              ; 3289
          51       04  AC           60  C3 00006          SUBL3   (R0), BUFFER, R1                          ;
                       51 00000200  8F  C6 0000B          DIVL2   #512, R1                                  ;
                       51           20  C4 00012          MULL2   #32, R1                                   ;
                       51       18  A0  C0 00015          ADDL2   24(R0), BFRD                              ; 3290
                   18  A1           0C  8A 00019          BICB2   #12, 24(BFRD)                             ; 3293
                       50 00000000G 00  D0 0001D          MOVL    CTL$GL_PCB, R0                            ; 3295
                   60  A0       1C  A1  B1 00024          CMPW    28(BFRD), 96(R0)                          ;
                                    04  13 00029          BEQL    1$                                        ;
                                      FEFF 0002B          BUGW                                              ; 3297
                                     0000* 0002D          .WORD   <BUG$_XQPERR!4>                           ;
  50       18  A1          02       00  EF 0002F  1$:     EXTZV   #0, #2, 24(BFRD), POOL                    ; 3304
                           51       61  0F 00035          REMQUE  (BFRD), BFRD                              ; 3305
                           50  CC AA40  7E 00038          MOVAQ   -52(BASE)[POOL], R0                       ; 3306
                           60       61  0E 0003D          INSQUE  (BFRD), (R0)                              ;
                   BC  AA       04  AC  D1 00040          CMPL    BUFFER, -68(BASE)                         ; 3308
                                03  12 00045              BNEQ    2$                                        ;
                       B4  AA       D4 00047              CLRL    -76(BASE)                                 ; 3310
                                    04 0004A      2$:     RET                                               ; 3312

; Routine Size:  75 bytes,    Routine Base:  $CODE$ + 0DF3
```

```
1 GLOBAL ROUTINE WRITE_HEADER : L_NORM NOVALUE  =
1
1 !++
1 !
1 ! FUNCTIONAL DESCRIPTION:
1 !
1 !       This routine writes out the currently resident file header.
1 !
1 ! CALLING SEQUENCE:
1 !       WRITE_HEADER ()
1 !
1 ! INPUT PARAMETERS:
1 !       NONE
1 !
1 ! IMPLICIT INPUTS:
1 !       FILE_HEADER: address of current file header
1 !
1 ! OUTPUT PARAMETERS:
1 !       NONE
1 !
1 ! IMPLICIT OUTPUTS:
1 !       IO_STATUS: status of I/O transfer
1 !
1 ! ROUTINE VALUE:
1 !       NONE
1 !
1 ! SIDE EFFECTS:
1 !       checksum checked, header written
1 !
1 !--
1
2 BEGIN
2
2 BIND_COMMON;
2
2 EXTERNAL ROUTINE
2       CHECKSUM        : L_NORM;       ! compute file header checksum
2
2
2 ! The checksum of the header should be good, since all routines that modify
2 ! the header bless it with a new checksum when they are finished. Check the
2 ! checksum and write the header.
2 !
2
2 IF NOT CHECKSUM (.FILE_HEADER)
2 THEN BUG_CHECK (WRTINVHDR, FATAL, 'ACP attempted to write an invalid file header');
2
2 WRITE_BLOCK (.FILE_HEADER);
2
1 END;                                  ! end of routine WRITE_HEADER


                                        .EXTRN  CHECKSUM, BUG$_WRTINVHDR

                     0000 00000         .ENTRY  WRITE_HEADER, Save nothing          ; 3313
        04   AA   DD 00002              PUSHL   4(BASE)                             ; 3357
```

```
        0000G  CF          01  FB 00005         CALLS   #1, CHECKSUM                          ;
               04          50  EB 0000A         BLBS    R0, 1$                                ;
                             FEFF 0000D         BUGW                                          ; 3358
                            0000* 0000F         .WORD   <BUG$_WRTINVHDR!4>                    ;
                     04    AA  DD 00011 1$:     PUSHL   4(BASE)                               ; 3360
        F967   CF          01  FB 00014         CALLS   #1, WRITE_BLOCK                       ;
                               04 00019         RET                                           ; 3362

; Routine Size:  26 bytes,    Routine Base:  $CODE$ + 0E3E
```

```
GLOBAL ROUTINE WRITE_DIRTY (LOCKBASIS) : L_NORM NOVALUE =

!++
!
! FUNCTIONAL DESCRIPTION:
!
!       This routine writes all buffers which were modified back to the
!       disk from whence they came.
!
! INPUT PARAMETERS:
!
! SIDE EFFECTS:
!       dirty buffers written.
!
!--

BEGIN

BIND_COMMON;

INCR POOL FROM 0 TO 2
DO
    BEGIN
    LOCAL
        I,
        QHEAD   : REF BBLOCK,
        BFRD    : REF BBLOCK;

    QHEAD = BFR_LIST [.POOL, QFLNK];
    BFRD = .QHEAD [QFLNK];

    I = .BFRS_USED [.POOL];

    UNTIL .I EQL 0
    DO
        BEGIN
        I = .I - 1;

        IF (.LOCKBASIS EQL 0
            OR .BFRD [BFRD$L_LOCKBASIS] EQL .LOCKBASIS)
            AND .BFRD [BFRD$V_DIRTY]
        THEN
            WRITE_BLOCK (((.BFRD-.CACHE_HDR [F11BC$L_BFRDBAS])/BFRD$S_BFRDDEF)*512
                         + .CACHE_HDR [F11BC$L_BUFBASE]);

        BFRD = .BFRD [BFRD$L_QFL];
        END;

    IF .QHEAD NEQA .BFRD
    THEN
        BUG_CHECK (XQPERR, 'in-process queue screwed up');

    END;

END;                            ! of routine WRITE_DIRTY
```

```
                              003C 00000       .ENTRY  WRITE_DIRTY, Save R2,R3,R4,R5                    ; 3363
                         53     D4 00002       CLRL    POOL                                             ; 3383
          54      CC AA43       7E 00004 1$:   MOVAQ   -52(BASE)[POOL], QHEAD                           ; 3391
          52             64     D0 00009       MOVL    (QHEAD), BFRD                                    ; 3392
          55      F4 AA43       3C 0000C       MOVZWL  -12(BASE)[POOL], I                               ; 3394
                         55     D5 00011 2$:   TSTL    I                                                ; 3396
                         33     13 00013       BEQL    5$                                               ;
                         55     D7 00015       DECL    I                                                ; 3399
                  04     AC     D5 00017       TSTL    LOCKBASIS                                        ; 3401
                         07     13 0001A       BEQL    3$                                               ;
      04  AC      10     A2     D1 0001C       CMPL    16(BFRD), LOCKBASIS                              ; 3402
                         20     12 00021       BNEQ    4$                                               ;
  1B  18  A2             02     E1 00023 3$:   BBC     #2, 24(BFRD), 4$                                 ; 3403
          51      FC     AA     D0 00028       MOVL    -4(BASE), R1                                     ; 3405
  50      52      18     A1     C3 0002C       SUBL3   24(R1), BFRD, R0                                 ;
          50             20     C6 00031       DIVL2   #32, R0                                          ;
  50      50             09     78 00034       ASHL    #9, R0, R0                                       ;
          51             61     D0 00038       MOVL    (R1), R1                                         ; 3406
                       6140     9F 0003B       PUSHAB  (R1)[R0]                                         ;
      F923  CF           01     FB 0003E       CALLS   #1, WRITE_BLOCK                                  ;
          52             62     D0 00043 4$:   MOVL    (BFRD), BFRD                                     ; 3408
                         C9     11 00046       BRB     2$                                               ; 3396
          52             54     D1 00048 5$:   CMPL    QHEAD, BFRD                                      ; 3411
                         04     13 0004B       BEQL    6$                                               ;
                              FEFF 0004D       BUGW                                                     ; 3413
                             0000* 0004F       .WORD   <BUG$_XQPERR!4>                                  ;
  AF      53             02     F3 00051 6$:   AOBLEQ  #2, POOL, 1$                                     ; 3383
                                04 00055       RET                                                      ; 3417

; Routine Size:  86 bytes,    Routine Base:  $CODE$ + 0E58
```

```
GLOBAL ROUTINE TOSS_CACHE_DATA (LCKINDX) : L_NORM NOVALUE =

!++
!
! FUNCTIONAL DESCRIPTION:
!
!       This routine invalidates all data blocks associated with
!       the given lock basis, after first writing dirty blocks.
!       The data sequence number is incremented, which invalidates
!       copies of these blocks on other nodes, and blocks found in
!       our cache will be marked invalid, causing them to be tossed
!       from the cache when the serialization lock is lowered.
!       Only the data block pool is scanned.
!
! SIDE EFFECTS:
!       dirty buffers written, appropriate buffers invalidated
!
!--

BEGIN

BIND_COMMON;

LOCAL
    I,
    LOCKBASIS,
    QHEAD       : REF BBLOCK,
    BFRD        : REF BBLOCK;

LOCKBASIS = .LB_BASIS [.LCKINDX];

QHEAD = BFR_LIST [1, QFLNK];
BFRD = .QHEAD [QFLNK];

I = .BFRS_USED [1];

UNTIL .I EQL 0
DO
    BEGIN
    I = .I - 1;

    IF .BFRD [BFRD$L_LOCKBASIS] EQL .LOCKBASIS
    THEN
        BEGIN

        IF .BFRD [BFRD$V_DIRTY]
        THEN
            WRITE_BLOCK (((.BFRD-.CACHE_HDR [F11BC$L_BFRDBAS])/BFRD$S_BFRDDEF)*512
                        + .CACHE_HDR [F11BC$L_BUFBASE]);

        BFRD [BFRD$V_VALID] = 0;
        END;

    BFRD = .BFRD [BFRD$L_QFL];
    END;                ! of loop through in-process data block pool

IF .QHEAD NEQA .BFRD
```

```
; 2506          3475  2 THEN
; 2507          3476  2     BUG_CHECK (XQPERR, 'in-process data block queue screwed up');
; 2508          3477  2
; 2509          3478  2 LB_DATASEQ [.LCKINDX] = .LB_DATASEQ [.LCKINDX] + 1;
; 2510          3479  2
; 2511          3480  1 END;                          ! of routine TOSS_CACHE_DATA



                                       003C 00000       .ENTRY  TOSS_CACHE_DATA, Save R2,R3,R4,R5          ; 3418
                    51       F4   AA    9E 00002        MOVAB   -12(BASE), R1                               ; 3437
                    50       04   AC    D0 00006        MOVL    LCKINDX, R0                                 ; 3447
                    55     0080 CA40    D0 0000A        MOVL    128(BASE)[R0], LOCKBASIS                    ;
                    53       D4   AA    9E 00010        MOVAB   -44(BASE), QHEAD                            ; 3449
                    52            63    D0 00014        MOVL    (QHEAD), BFRD                               ; 3450
                    54       02   A1    3C 00017        MOVZWL  2(R1), I                                    ; 3452
                                  54    D5 0001B 1$:    TSTL    I                                           ; 3454
                                  31    13 0001D        BEQL    4$                                          ;
                                  54    D7 0001F        DECL    I                                           ; 3457
                    55       10   A2    D1 00021        CMPL    16(BFRD), LOCKBASIS                         ; 3459
                                  24    12 00025        BNEQ    3$                                          ;
          1B       18   A2        02    E1 00027        BBC     #2, 24(BFRD), 2$                            ; 3463
                    51       FC   AA    D0 0002C        MOVL    -4(BASE), R1                                ; 3465
          50        52      18   A1    C3 00030        SUBL3   24(R1), BFRD, R0                            ;
                    50            20    C6 00035        DIVL2   #32, R0                                     ;
          50        50            09    78 00038        ASHL    #9, R0, R0                                  ;
                    51            61    D0 0003C        MOVL    (R1), R1                                    ; 3466
                                6140    9F 0003F        PUSHAB  (R1)[R0]                                    ;
               F8C9 CF            01    FB 00042        CALLS   #1, WRITE_BLOCK                             ;
                 18 A2            08    8A 00047 2$:    BICB2   #8, 24(BFRD)                                ; 3468
                    52            62    D0 0004B 3$:    MOVL    (BFRD), BFRD                                ; 3471
                                  CB    11 0004E        BRB     1$                                          ; 3454
                    52            53    D1 00050 4$:    CMPL    QHEAD, BFRD                                 ; 3474
                                  04    13 00053        BEQL    5$                                          ;
                                      FEFF 00055        BUGW                                                ; 3476
                                     0000* 00057        .WORD   <BUG$_XQPERR!4>                             ;
                    50       04   AC    D0 00059 5$:    MOVL    LCKINDX, R0                                 ; 3478
                         00A8 CA40    D6 0005D        INCL    168(BASE)[R0]                               ;
                                        04 00062        RET                                                 ; 3480

; Routine Size:  99 bytes,    Routine Base:  $CODE$ + 0EAE


; 2512          3481  1
```

```
GLOBAL ROUTINE KILL_CACHE (UCB) : L_NORM NOVALUE =

!++
!
! This routine scans through all BFRD's in the cache tossing out
! all those that match the specified UCB.
!
!--

BEGIN

BIND_COMMON;

LOCAL
    BFRD : REF BBLOCK,
    PIDINDX : WORD;

BFRD = .CACHE_HDR [F11BC$L_BFRDBAS];
PIDINDX = .CT[$GL_PCB [PCB$L_PID];

SERIAL_CACHE ();

INCR INDX0 FROM 0 TO (.CACHE_HDR [F11BC$W_BFRCNT] - 1)
DO
    BEGIN

    IF .BFRD [BFRD$L_UCB] EQLA .UCB
    THEN
        BEGIN

        IF .BFRD [BFRD$W_CURPID] NEQ 0
        THEN
            BEGIN
            IF .BFRD [BFRD$W_CURPID] EQL .PIDINDX
            THEN
                BFRD [BFRD$V_VALID] = 0;        ! it will get tossed on unlock
            END
        ELSE
            BEGIN
            UNHOOK_BFRD (.INDX0+1, .BFRD);
            RETURN_BFRD (.BFRD);
            END;
        END;

    BFRD = .BFRD + BFRD$S_BFRDDEF;

    END;

RELEASE_CACHE ();

END;


                    003C 00000        .ENTRY  KILL_CACHE, Save R2,R3,R4,R5                 ; 3482
```

```
              50        FC  AA  D0 00002        MOVL    -4(BASE), R0                    ; 3499
              52        18  A0  D0 00006        MOVL    24(R0), BFRD                    ;
              50 00000000G  00  D0 0000A        MOVL    CTL$GL_PCB, R0                  ; 3500
              55        60  A0  B0 00011        MOVW    96(R0), PIDINDX                 ;
                      F35D  30 00015            BSBW    SERIAL_CACHE                    ; 3502
              50        FC  AA  D0 00018        MOVL    -4(BASE), R0                    ; 3504
              54        16  A0  3C 0001C        MOVZWL  22(R0), R4                      ;
              53            01  CE 00020        MNEGL   #1, INDX0                       ;
                            2B  11 00023        BRB     4$                              ;
      04      AC        0C  A2  D1 00025 1$:    CMPL    12(BFRD), UCB                   ; 3508
                            21  12 0002A        BNEQ    3$                              ;
                        1C  A2  B5 0002C        TSTW    28(BFRD)                        ; 3512
                            0C  13 0002F        BEQL    2$                              ;
              55        1C  A2  B1 00031        CMPW    28(BFRD), PIDINDX               ; 3515
                            16  12 00035        BNEQ    3$                              ;
      18      A2            08  8A 00037        BICB2   #8, 24(BFRD)                    ; 3517
                            10  11 0003B        BRB     3$                              ; 3512
                            52  DD 0003D 2$:    PUSHL   BFRD                            ; 3521
                        01  A3  9F 0003F        PUSHAB  1(INDX0)                        ;
    FA25      CF            02  FB 00042        CALLS   #2, UNHOOK_BFRD                 ;
              50            52  D0 00047        MOVL    BFRD, R0                        ; 3522
                      FE6A  30 0004A            BSBW    RETURN_BFRD                     ;
              52            20  C0 0004D 3$:    ADDL2   #32, BFRD                       ; 3526
  D1          53            54  F2 00050 4$:    AOBLSS  R4, INDX0, 1$                   ; 3504
                      F779  30 00054            BSBW    RELEASE_CACHE                   ; 3530
                            04 00057            RET                                     ; 3532

; Routine Size:  88 bytes,    Routine Base:  $CODE$ + 0F11
```

```
1 GLOBAL ROUTINE WRONG_LOCKBASIS (HEADER) : L_NORM NOVALUE =
1
1 !++
1 !
1 ! This routine returns the given buffer as a result of discovering
1 ! we had the wrong lockbasis when finding it.
1 ! If it was read from disk, we must invalidate it, as we stored the
1 ! wrong lockbasis in its bfrd.  If it was found in the cache,
1 ! simply return it and credit us for returning it.
1 !
1 !--
1
2 BEGIN
2
2 BIND_COMMON;
2
2 LOCAL
2      POOL,
2      BFRD          : REF BBLOCK,
2      POOL_LRU      : REF BBLOCK;
2
2 SERIAL_CACHE ();
2
2 BFRD = ((.HEADER - .CACHE_HDR [F11BC$L_BUFBASE])/512)*BFRD$S_BFRDDEF
2         + .CACHE_HDR [F11BC$L_BFRDBAS];
2
2 IF .BFRD [BFRD$W_BFRL] EQL 0
2 THEN
2      BFRD [BFRD$V_VALID] = 0
2 ELSE
3      BEGIN
3      POOL = .BFRD [BFRD$V_POOL];
3      REMQUE (.BFRD, BFRD);
3      BFRS_USED [.POOL] = .BFRS_USED [.POOL] - 1;
3      BFRD [BFRD$W_CURPID] = 0;
3      POOL_LRU = CACHE_HDR [F11BC$Q_POOL_LRU] + .POOL*8;
3      INSQUE (.BFRD, .POOL_LRU [QBLNK]);
2      END;
2
2 RELEASE_CACHE ();
1 END;
```

```
                                    000C 00000     .ENTRY  WRONG_LOCKBASIS, Save R2,R3     ; 3533
                               F318   30 00002     BSBW    SERIAL_CACHE                    ; 3554
             50          FC      AA   D0 00005     MOVL    -4(BASE), R0                    ; 3556
   51    04  AC                  60   C3 00009     SUBL3   (R0), HEADER, R1                ;
             51    00000200      8F   C6 0000E     DIVL2   #512, R1                        ;
             51                  20   C4 00015     MULL2   #32, R1                         ;
             51          18      A0   C0 00018     ADDL2   24(R0), BFRD                    ; 3557
                         1A      A1   B5 0001C     TSTW    26(BFRD)                        ; 3559
                                 06   12 0001F     BNEQ    1$                              ;
   18    A1                      08   8A 00021     BICB2   #8, 24(BFRD)                    ; 3561
                                 1C   11 00025     BRB     2$                              ;
```

```
     50    18   A1         02      00  EF 00027 1$:    EXTZV   #0, #2, 24(BFRD), POOL                   ; 3564
                           51      61  0F 0002D        REMQUE  (BFRD), BFRD                             ; 3565
                              F4 AA40  B7 00030        DECW    -12(BASE)[POOL]                          ; 3566
                              1C   A1  B4 00034        CLRW    28(BFRD)                                 ; 3567
                           50 FC BA40  7E 00037        MOVAQ   @-4(BASE)[POOL], POOL_LRU                ; 3568
                           50      28  C0 0003C        ADDL2   #40, POOL_LRU                            ;
                      04   B0      61  0E 0003F        INSQUE  (BFRD), @4(POOL_LRU)                     ; 3569
                                 F732  30 00043 2$:    BSBW    RELEASE_CACHE                            ; 3572
                                       04 00046        RET                                              ; 3573

; Routine Size:  71 bytes,    Routine Base:  $CODE$ + 0F69
```

```
1 GLOBAL ROUTINE KILL_BUFFERS (POOL, LOCKBASIS) : L_NORM NOVALUE =
1
1 !++
1 !
1 ! This routine scans through all BFRD's in the cache tossing out
1 ! all those in the specified POOL for the CURRENT_UCB.
1 ! If the directory data pool is specified, LOCKBASIS must match also.
1 !
1 !--
1
2 BEGIN
2
2 BIND_COMMON;
2
2 LOCAL
2     BFRD : REF BBLOCK,
2     BFRD_CNT : WORD,
2     START_BFRD : WORD,
2     PIDINDX : WORD;
2
3 BEGIN
3
3 BIND
3     POOLCNT = CACHE_HDR [F11BC$W_POOLCNT] : VECTOR [,WORD];
3
3 START_BFRD = 0;
3
3 IF .POOL NEQ 0
3 THEN
4     BEGIN
4     START_BFRD = .POOLCNT [0];
4     IF .POOL EQL 2
4     THEN START_BFRD = .START_BFRD + .POOLCNT [1];
3     END;
3
3 BFRD_CNT = .POOLCNT [.POOL];
2 END;
2
2 BFRD = .CACHE_HDR [F11BC$L_BFRDBAS] + .START_BFRD*BFRD$S_BFRDDEF;
2 PIDINDX = .CT[$GL_PCB [PCB$L_PID];
2
2 SERIAL_CACHE ();
2
3 INCR INDX0 FROM .START_BFRD TO (.START_BFRD + .BFRD_CNT - 1)
2 DO
3     BEGIN
3
3     IF .CURRENT_UCB EQL .BFRD [BFRD$L_UCB]
4         AND (.POOL NEQ 1 OR
4             .BFRD [BFRD$L_LOCKBASIS] EQL .LOCKBASIS)
3     THEN
4         BEGIN
4
4         IF .BFRD [BFRD$W_CURPID] NEQ 0
4         THEN
5             BEGIN
5             IF .BFRD [BFRD$W_CURPID] EQL .PIDINDX
```

```
5                  THEN
5                      BFRD [BFRD$V_VALID] = 0;          ! it will get tossed on unlock
5                  END
4              ELSE
5                  BEGIN
5                  UNHOOK_BFRD (.INDX0+1, .BFRD);
5                  RETURN_BFRD (.BFRD);
4                  END;
3              END;
3
3          BFRD = .BFRD + BFRD$S_BFRDDEF;
3
2          END;
2
2      RELEASE_CACHE ();
2
1      END;
```

```
                                         003C 00000          .ENTRY  KILL_BUFFERS, Save R2,R3,R4,R5               ; 3574
51          FC    AA 00000078      8F    C1 00002            ADDL3   #120, -4(BASE), R1                           ; 3597
                                   53    B4 0000B            CLRW    START_BFRD                                   ; 3599
                  50          04   AC    D0 0000D            MOVL    POOL, R0                                     ; 3601
                                   0C    13 00011            BEQL    1$                                           ;
                  53               61    B0 00013            MOVW    (R1), START_BFRD                             ; 3604
                  02               50    D1 00016            CMPL    R0, #2                                       ; 3605
                                   04    12 00019            BNEQ    1$                                           ;
                  53          02   A1    A0 0001B            ADDW2   2(R1), START_BFRD                            ; 3606
                  54             6140    B0 0001F 1$:        MOVW    (R1)[R0], BFRD_CNT                           ; 3609
                  50          FC   AA    D0 00023            MOVL    -4(BASE), R0                                 ; 3612
                  51               53    3C 00027            MOVZWL  START_BFRD, R1                               ;
                  51               20    C4 0002A            MULL2   #32, R1                                      ;
52                51          18   A0    C1 0002D            ADDL3   24(R0), R1, BFRD                             ;
                  50 00000000G     00    D0 00032            MOVL    CTL$GL_PCB, R0                               ; 3613
                  55          60   A0    B0 00039            MOVW    96(R0), PIDINDX                              ;
                                 F296    30 0003D            BSBW    SERIAL_CACHE                                 ; 3615
                  50               53    3C 00040            MOVZWL  START_BFRD, R0                               ; 3617
                  54               54    3C 00043            MOVZWL  BFRD_CNT, R4                                 ;
                  54               50    C0 00046            ADDL2   R0, R4                                       ;
                  53               53    3C 00049            MOVZWL  START_BFRD, INDX0                            ;
                                   53    D7 0004C            DECL    INDX0                                        ;
                                   38    11 0004E            BRB     6$                                           ;
            0C    A2          94   AA    D1 00050 2$:        CMPL    -108(BASE), 12(BFRD)                         ; 3621
                                   2E    12 00055            BNEQ    5$                                           ;
                  01          04   AC    D1 00057            CMPL    POOL, #1                                     ; 3622
                                   07    12 0005B            BNEQ    3$                                           ;
            08    AC          10   A2    D1 0005D            CMPL    16(BFRD), LOCKBASIS                          ; 3623
                                   21    12 00062            BNEQ    5$                                           ;
                              1C   A2    B5 00064 3$:        TSTW    28(BFRD)                                     ; 3627
                                   0C    13 00067            BEQL    4$                                           ;
                  55          1C   A2    B1 00069            CMPW    28(BFRD), PIDINDX                            ; 3630
                                   16    12 0006D            BNEQ    5$                                           ;
            18    A2               08    8A 0006F            BICB2   #8, 24(BFRD)                                 ; 3632
                                   10    11 00073            BRB     5$                                           ; 3627
```

```
                                   52  DD 00075 4$:     PUSHL   BFRD                                             ; 3636
                        01         A3  9F 00077         PUSHAB  1(INDX0)                                         ;
                F94E  CF           02  FB 0007A         CALLS   #2, UNHOOK_BFRD                                  ;
                      50           52  D0 0007F         MOVL    BFRD, R0                                         ; 3637
                                 FD93  30 00082         BSBW    RETURN_BFRD                                      ;
                      52           20  C0 00085 5$:     ADDL2   #32, BFRD                                        ; 3641
             C4       53           54  F2 00088 6$:     AOBLSS  R4, INDX0, 2$                                    ; 3617
                                 F6A2  30 0008C         BSBW    RELEASE_CACHE                                    ; 3645
                                       04 0008F         RET                                                      ; 3647

; Routine Size:  144 bytes,    Routine Base:  $CODE$ + 0FB0


; 2682          3648  1
; 2683          3649  1 END
; 2684          3650  0 ELUDOM




;                                    PSECT SUMMARY
;
;       Name                         Bytes                     Attributes
;
;   $CODE$                               4160  NOVEC,NOWRT,  RD ,  EXE,NOSHR,  LCL,  REL,  CON,NOPIC,ALIGN(2)



;                               Library Statistics
;
;                                              -------- Symbols --------       Pages          Processing
;       File                                   Total     Loaded   Percent       Mapped         Time
;
;   _$255$DUA28:[SYSLIB]LIB.L32;1              18619        90         0       1000            00:01.9




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

; Size:          4153 code + 7 data bytes
; Run Time:         04:03.1
; Elapsed Time:     07:51.7
; Lines/CPU Min:     900
; Lexemes/CPU-Min: 64276
; Memory Used:  476 pages
; Compilation Complete
```

0172 AH-BT13A-SE
VAX/VMS V4.0

DIGITAL EQUIPMENT CORPORATION
CONFIDENTIAL AND PROPRIETARY

RWVB
LIS

RDBLOK
LIS

REQUEU
LIS

RWATTR
LIS

REMOVE
LIS

RDHEDR
LIS

RETDIR
LIS