HASH

```
          IDENT     HASH
I
          ERR       DAVE HAS SOURCE FOR HASH
*
          TITLE     USER DISK SUBPROCESS
          TITLE     HASH TABLE CONTROL
*
PARAMS    XTEXT
MACROS    XTEXT
SYSCALL   XTEXT
*
          EXT       OPERCL,IPLIST
          MACSET    OPERCL,IPLIST,ECS
BEADTST   XTEXT
*
*
          EXT       DDSIPL,GF.DDSF,GF.DDSL,GF.HSHF,GF.HSHL
          EXT       GF.SUSF,GF.SUSL,HASHIPL,HASHT1,HASHT2,HASHT3
          EXT       MAINCL,MYEVCH,MYIDENT,M.ADDS,M.DDSCT
          EXT       M.HASHL,READCAP,SLEEP,SUSPIPL,SYSERR
          EXT       WRITCAP
*
*PAGE               HASH...HASH...HASH
*
*
*                   HASH ACCESS TO FILE HEADER RECORD OR DISK ALLOCATION
*                   RECORD
*
*                   NOTES:    A) HASHING IS ON LOW ORDER 7 BITS OF THE
*                                UNIQUE NAME
*                             B) IF NO ENTRY IS FOUND, A NEW ENTRY IS
*                                ALLOCATED AND CHAINED TO THE HASH TABLE
*                                (1ST WD → UNIQUE NAME)
*                             C) LOCK IS SET IN THE 2ND WORD OF THE
*                                RECORD. SUSPENSE LIST IS USED IF RECORD
*                                IS LOCKED
*                             D) HASH LINKS ARE IN BITS 18-35
*                                OF THE SECOND WORD OF THE RECORD
*                             E) ORDER OF LOCK USE: ↑M.HASHL↑, ↑FHRLOCK↑
*                                OR ↑DARLOCK↑, AND ↑ALOCDDS↑.
*
*
*                   ON ENTRY: X1 = UNIQUE NAME (BITS 26-59)
*                             B4 = ↑M.FHRL↑ OR ↑M.DARL↑
*                             B5 = ↑FHRHASH↑ OR ↑DARHASH↑
```

```
*                               B6 = RETURN LINK
*                               B6-1 = FAIL RETURN IF NO DDS AVAIL
*
*
*                     ON EXIT:  B5 = 0 IF FOUND AN ENTRY;
*                                    1 IF NEW ENTRY MADE
*                               X5 = DDS ADDRESS OF ENTRY
*                               ** FILE IS LOCKED **
*
*
*                     REGISTERS USED: B5, A4/X4, A5/X5, A6/X6, A7/X7
*
*                     TEMPORARIES USED: HASHT1, HASHT2, HASHT3
*                               (NOTE : HASHT3 MUST = HASHT2+1)
*
*                     LOCKS USED: ↑M.HASHL↑, ↑FHRLOCK↑, ↑DARLOCK↑,
*                                 ↑ALOCDDS↑, ↑MYEVCH↑
*
*
*
          ENTRY     HASH
*
HASH      MX6       34                  UNIQUE NAME MASK
          BX5       X1*X6
          LX5       34
          MX6       60-7
          BX5       -X6*X5              HASH ADDRESS
          GETEVH    (MAINCL,M.HASHL)     RESERVE HASHING
*
          REHASH    B5,X5,HASHT1         READ THE HASH TABLE
          SA4       HASHT1
          ZR        X4,HASH2            JP IF NOT IN HASH TABLE
*
*
HASH1     REDDS     X4,HASHT2,2          READ HASH CHAIN ENTRY
          SA5       HASHT2
          BX6       X5-X1
          MX7       34
          BX6       X7*X6
          ZR        X6,HASH5            JP IF FOUND RECORD
*
          AX6       26                  CONSISTANCY CHECK (4 INSTR)
          MX7       60-7                HASH ADDR MUST BE SAME FOR
          BX6       -X7*X6              ALL RECORDS ON HASH CHAIN
          NZ        X6,HASH7
*
          SA4       HASHT3
          LX4       60-18
          SX4       X4                  HASH LINK TO NEXT RECORD
          NZ        X4,HASH1            LOOP TO END OF HASH CHAIN
*
*                   NO RECORD...MUST ALLOCATE NEW RECORD
*
HASH2     GETEVH    (MAINCL,M.DDSCT)    RESERVE ONE DDS RECORD
          SX5       X7-1
          NG        X5,HASH4A           JP IF OUT OF DDS
```

```
          SENDEV    *,X5
*
          GETEVH    (MAINCL,M.ADDS)     RESERVE DDS ALLOCATION
          ZR        X7,HASH4            JP IF OUT OF DDS SPACE
          BX5       X7                  ADDR OF NEW RECORD
          REDDS     X5,HASHT3,1
          SA4       HASHT3               GET LINK TO NEXT FREE BLOCK AND
          SENDEV    *,X4          RETURN LINK TO FREE SPACE EVENT CHAN
*
*                                       FORM PROTO-TYPE OF NEW RECORD
          SA4       HASHT1               OLD LINK FROM HASH TABLE
          LX4       18
          MX6       1                   SET LOCK IN NEW RECORD
          BX6       X6+X4
          SA6       HASHT2+1
*
          BX7       X1                  UNIQUE NAME FOR NEW RECORD
          SA7       HASHT2
*
          WRDDS     X5,HASHT2,2          WRITE PROTO-TYPE
*
*
          BX7       X1
          LX7       34                  RE-COMPUTE HASH ADDRESS
          MX6       60-7
          BX4       -X6*X7
          BX6       X5                  ADDR OF NEW RECORD
          SA6       HASHT1
          WRHASH    B5,X4,HASHT1         LINK NEW RECORD TO HASH TABLE
*
          SENDEV    (MAINCL,M.HASHL),B0   RELEASE HASHING
*
          SB5       1                   INDICATE NEW RECORD
          JP        B6                  RETURN
*
*                         OUT OF DDS SPACE
HASH4     SENDEV    (MAINCL,M.ADDS),B0   RELEASE ALLOCATION LOCK
          SENDEV    (MAINCL,M.HASHL),B0
          RJ        SYSERR
*
HASH7     RJ        SYSERR              HASH CHAIN SCREWED UP
*
*                         OUT OF DDS..SPECIAL RETURN
HASH4A    BX5       X7
          SENDEV    *,X5
          SENDEV    (MAINCL,M.HASHL),B0
          JP        B6-1
          SPACE     4
*
*
*                   FOUND THE RECORD
*
HASH5     GETEVH    (MAINCL,B4)         LOCK ALL RECOREDS
          REDDS     X4,HASHT2,2          TRY TO LOCK THIS RECORD
          SX5       X4+0                ADDR OF RECORD
          SA4       HASHT2+1             2ND WD OF RECORD
```

```
          NG        X4,HASH6              JP IF ALREADY LOCKED
          SB5       B0                    SIGNAL FOUND A RECORD
          MX6       1
          BX6       X4+X6                 SET LOCK
          SA6       A4
          WRDDS     *,*,*                 WRITE BACK LOCK
*
          SENDEV    *,B0
          SENDEV    (MAINCL,M.HASHL),B0
*
          JP        B6                    RETURN
*
*
*                   RECORD IS LOCKED...WAIT ON SUSPENSE LIST
*                                         X5 = ADDR OF LOCKED RECORD
*
HASH6     SX7       X4                    OLD HEAD OF SUSPENSE LIST
          BX6       X4-X7
          SA4       MYIDENT
          SA7       HASHT1                 PUT SELF AT HEAD OF SUSPENSE
          BX6       X6+X4                 LIST
          SA6       HASHT2+1
          WRDDS     *,*,*                 WRITE BACK LOCK / SUSPENSE STUFF
          WRSUSP    X4,HASHT1              WRITE SUSPENSE LIST LINK
*
          SENDEV    *,B0          RELEASE RECORDS
          SENDEV    (MAINCL,M.HASHL),B0   RELEASE HASHING
*
          GETEVH    MYEVCH       WAIT FOR WAKE UP
*
          NZ        X7,HASH               START OVER...RECORD MAY BE GONE
          BX4       X5                    ADDR OF RECORD
          GETEVH    (MAINCL,M.HASHL)       LOCK HASHING
          JP        HASH5                 GO TRY TO LOCK RECORDS
*
*                   END HASH...END HASH...END HASH
*PAGE               DELHASH...DELHASH...DELHASH
*
*
*
*                   DELETE HASH TABLE ENTRY
*
*
*                   NOTES:     A) THE UNIQUE NAME DRIVES A WALK DOWN
*                                 THE PROPER CHAIN FROM THE HASH TABLE
*                              B) HASH ADDR IS FORMED FROM THE LOW
*                                 ORDER 7-BITS OF THE UNIQUE NAME
*                              C) THE RECORD TO BE DELETED SHOULD BE
*                                 LOCKED ON ENTRY
*                              D) ALL RECORDS MUST BE LOCKED WHILE
*                                 PATCHING THE HASH CHAIN (UNLESS THE
*                                 RECORD BEING REMOVED IS 1ST ON THE
*                                 HASH CHAIN)
*                              E) ORDER OF LOCK USE IS ↑M.HASHL↑,
*                                 AND THEN ↑FHRLOCK↑ OR ↑DARLOCK↑
*
```

```
*
*
*                    ON ENTRY:  RECORD IS LOCKED
*                               X1 = UNIQUE NAME / DISK ADDRESS
*                               B4 = ↑M.FHRL↑ OR ↑ M.DARL↑
*                               B5 = ↑FHRHASH↑ OR ↑DARHASH↑
*                               B6 = RETURN LINK
*
*
*                    REGISTERS USED: A3/X3, A4/X4, A5/X5, A6/X6, A7/X7
*
*                    TEMPORARIES USED: HASHT1, HASHT2, HASHT3
*                              (NOTE : HASHT3 MUST = HASHT2+1)
*
*                    LOCKS USED: M.HASHL, FHRLOCK, DARLOCK, M.ADDS
*
*
*
            ENTRY     DELHASH
*
DELHASH     BX5       X1
            LX5       34
            MX6       60-7
            BX5       -X6*X5                HASH ADDR
            GETEVH    (MAINCL,M.HASHL)       RESERVE HASHING
*
            REHASH    B5,X5,HASHT1           READ HASH TABLE
            SA4       HASHT1
            ZR        X4,DELHASH3           CONSISTANCY CHECK
*
            REDDS     X4,HASHT2,2            MUST CHASE 1ST LINK EXPLICITLY
            SA3       HASHT2                 IN CASE THE HASH TABLE MUST BE
            BX6       X3-X1                 PATCHED
            MX7       34
            BX6       X7*X6
            NZ        X6,DELHASH2           JP IF NO MATCH
*
            SA3       HASHT3                 PATCH HASH TABLE
            LX3       60-18
            SX6       X3                    LINK TO NEXT RECORD
            SA6       HASHT1
            WRHASH    *,*,*                 WRITE INTO HASH TABLE
            JP        DELHASH5              CONTINUE WITH RELEASE OF RECORD
*
*
*                                 SEARCH FOR DESIRED RECORD
DELHASH1    REDDS     X4,HASHT2,2            READ NEXT RECORD
            SA3       HASHT2
            BX6       X1-X3
            MX7       34
            BX6       X7*X6
            ZR        X6,DELHASH4           JP IF FOUND THE RECORD
DELHASH2    MX7       60-7
            AX6       26                    CONSISTANCY CHECK (HASH ADDR
            BX6       -X7*X6                MUST BE SAME FOR ALL RECORDS
            NZ        X6,DELHASH8           ON SAME HASH CHAIN)
```

```
*
          BX5       X4                  ADDR OF OLD RECORD
          SA4       HASHT3
          AX4       18
          SX4       X4                  ADDR OF NEXT RECORD
          NZ        X4,DELHASH1         CONSISTANCY CHECK
*
DELHASH3  RJ        SYSERR              REACHED END OF HASH CHAIN
*
*
*                             PATCH UP HASH CHAIN
DELHASH4  SA3       HASHT3               CURRENT RECORD
          GETEVH    (MAINCL,B4)         LOCK ALL RECORDS
          REDDS     X5,HASHT2,2          READ FROM PREVIOUS RECORD
          SA5       HASHT3
          LX3       60-18
          SX6       X3                  ADDR OF NEXT RECORD
          LX5       60-18
          SX7       X5
          BX5       X5-X7
          BX6       X5+X6               SET NEW LINK
          LX6       18
          SA6       A5
          WRDDS     *,*,*               WRITE BACK PREVIOUS RECORD
          SENDEV    *,B0
*
*
*                             RELEASE DDS SPACE
DELHASH5  GETEVH    (MAINCL,M.ADDS)     X4= ADDR OF RECORD TO RELEASE
          SA7       HASHT1               OLD HEAD OF FREE LIST
          WRDDS     X4,HASHT1,1
          SENDEV    *,X4
*
*                             INCR DDS ALLOCATION
          GETEVH    (MAINCL,M.DDSCT)
          SX5       X7+1
          SENDEV    *,X5
*
*
*                             UNLOCK HASHING
          SENDEV    (MAINCL,M.HASHL),B0
*
*
*                             WAKE UP SUSPENSE LIST
          LX3       18
          SX4       X3                  SUSPENSE LIST HAAD LINK
          ZR        X4,DELHASH7
*
DELHASH6  RESUSP    X4,HASHT3            READ NEXT PROCESS ON SUSP LIST
*
          SENDEV    (SLEEP,X4),-3        SEND ↑FILE GONE↑
*
          SA4       HASHT3               NEXT PROCESS
          NZ        X4,DELHASH6         LOOP TILL END OF SUSPENSE LIST
*
*
```

```
*                                   EXIT
DELHASH7  JP        B6                      RETURN
*
*
DELHASH8  RJ        SYSERR                  HASH CHAIN SCREWED UP
*
*
*                   END DELHASH...END DELHASH...END DELHASH
*
*
*
          END
```