**HASH**

<table>
<thead>
<tr>
<th>IDENT</th>
<th>HASH</th>
</tr>
</thead>
<tbody>
<tr>
<td>ERR</td>
<td>DAVE HAS SOURCE FOR HASH</td>
</tr>
<tr>
<td>TITLE</td>
<td>USER DISK SUBPROCESS</td>
</tr>
<tr>
<td>TITLE</td>
<td>HASH TABLE CONTROL</td>
</tr>
<tr>
<td>PARMS</td>
<td>XTEXT</td>
</tr>
<tr>
<td>MACROS</td>
<td>XTEXT</td>
</tr>
<tr>
<td>SYSCALL</td>
<td>XTEXT</td>
</tr>
<tr>
<td>EXT</td>
<td>OPERCL, IPLIST</td>
</tr>
<tr>
<td>MACSET</td>
<td>OPERCL, IPLIST, ECS</td>
</tr>
<tr>
<td>READST</td>
<td>XTEXT</td>
</tr>
<tr>
<td>EXT</td>
<td>DDISPL, GF, DDSF, GF, DODSL, GF, HSHF, GF, HSHL</td>
</tr>
<tr>
<td>EXT</td>
<td>GF, SUSF, GF, SUSL, HASHPL, HASHT1, HASHT2, HASHT3</td>
</tr>
<tr>
<td>EXT</td>
<td>MAINL, MYEVC, MYIDENT, M, ADDS, M, DDSCT</td>
</tr>
<tr>
<td>EXT</td>
<td>M, HASHL, READCAP, SLEEP, SUSPL, SYSERR</td>
</tr>
<tr>
<td>EXT</td>
<td>WRITCAP</td>
</tr>
</tbody>
</table>

*PAGE

**HASH**...**HASH**...**HASH**

**HASH** ACCESSES 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

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, FHRXL OR +M, DARL*

B5 = +FHRHASH+ OR +DARHASH+
B6 = RETURN LINK
B6-1 = FAIL RETURN IF NO DDS AVAILABLE

ON EXIT: 
B5 = 0 IF FOUND AN ENTRY;
1 IF NEW ENTRY MADE
X5 = DDS ADDRESS OF ENTRY
** FILE IS LOCKED **


TEMPORARIES USED: HASHT1, HASHT2, HASHT3
(NOTE: HASHT3 MUST = HASHT2+1)

LOCKS USED: *M,HASHT1, *FHRLOCK*, *DARLOCK*,
*ALOCDDS*, *MYEVCH*

ENTRY  HASH

HASH  MX6  34  UNIQUE NAME MASK
     BX5  X1*X6
     LX5  34
     MX6  60-7
     BX5  -X6*X5
GETEVH (MAINCL,M,HASHT1) RESERVE HASHING

REHASH B5,X5,HASHT1 READ THE HASH TABLE
SA4  HASHT1
ZR  X4,HASHT2 JP IF NOT IN HASH TABLE

HASHT1 REDDS X4,HASHT2,2 READ HASH CHAIN ENTRY
SA5  HASHT2
RX6  X5,X1
MX7  34
RX6  X7*X6
ZR  X6,HASH5 JP IF FOUND RECORD

AX6  26  CONSISTENCY CHECK (4 INSTR)
MX7  60-7 HASH ADDR MUST BE SAME FOR
RX6  -X7*X6 ALL RECORDS ON HASH CHAIN
N7  X6,HASH7

SA4  HASHT3
LX4  60-13
SX4  X4
NZ  X4,HASHT1 HASH LINK TO NEXT RECORD

* 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 GET LINK TO NEXT FREE BLOCK AND
SA4 HASHT3 RETURN LINK TO FREE SPACE EVENT CHAN
SENDEV *,X4

SA4 HASHT1 FORM Proto-TYPE OF NEW RECORD
LX4 18 OLD LINK FROM HASH TABLE
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 RE-COMPUTE HASH ADDRESS
LX7 34
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),80 RELEASE HASHING
SB5 1 INDICATE NEW RECORD
JP R6 RETURN

HASHT4 SENDEV (MAINCL,M.ADDS),80 RELEASE ALLOCATION LOCK
SENDEV (MAINCL,M.HASHL),80 TRY TO LOCK NEW RECORD
RJ SYSERR

HASHT7 RJ SYSERR HASH CHAIN SCREWED UP

HASHT4A BX5 X7 OUT OF DDS SPACE SPECIAL RETURN
SENDEV *,X5
SENDEV (MAINCL,M.HASHL),80
JP B6-1
SPACE 4

FOUND THE RECORD

HASH5 GETEVH (MAINCL,R4) LOCK ALL RECORDS
REDDS X4,HASHT2,2 TRY TO LOCK THIS RECORD
SX6 X4+0 ADDR OF RECORD
SA4 HASHT2+1 2ND WD OF RECORD
**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 *H.LOCK*, and then *FHRLOCK* or *DARLOCK*.
ON ENTRY: RECORD IS LOCKED
X1 = UNIQUE NAME / DISK ADDRESS
B4 = *FHRLOCK+ OR +DARLOCK+
B5 = *FHRHASH+ OR +DARHASH+
B6 = RETURN LINK


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
GETENVH (MAINCL,M.HASHL) HASH ADDR
* 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
BX6 X3=X1
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
BX6 X7=X6
NZ X6,DELHASH8 CONSISTANCY CHECK (HASH ADDR
MUST BE SAME FOR ALL RECORDS
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 |

**DELEASH3**
- RJ  | SYSERR | REACHED END OF HASH CHAIN

**DELEASH4**
- SA3  | HASHT3 | CURRENT RECORD
- GETEVH  | (MAINCL,M.ADDS) |
- REDDS  | X5,HASHT2,2 | LOCK ALL RECORDS
- SA5  | HASHT3 |
- LX3  | 60-18 |
- SX6  | X3 |
- LX5  | 60-18 |
- SX7  | X5 |
- BX5  | X5-X7 |
- AX6  | X5+X6 |
- LX6  | 18 |
- SA6  | A5 |
- WRDDS  | **,**,* |
- SENDEV  | **,**BO |

**DELEASH5**
- GETEVH  | (MAINCL,M.ADDS) |
- SA7  | HASHT1 |
- WRDDS  | X4,HASHT1,1 |
- SENDEV  | **,**X4 |

**INCR DDS ALLOCATION**
- GETEVH  | (MAINCL,M.DDSCIT) |
- SX5  | X7+1 |
- SENDEV  | **,**X5 |

**SENDEV**
- (MAINCL,M,HASHTL),BO |

**WAKE UP SUSPENSE LIST**
- LX3  | 18 |
- SX4  | X3 |
- ZR  | X4,DELHASH7 |

**DELEASH6**
- RESUSP  | X4,HASHT3 | READ NEXT PROCESS ON SUSP LIST
- SFNDEV  | (SLEEP,X4),-3 | SEND *FILE GONE*
- SA4  | HASHT3 |
- NZ  | X4,DELHASH6 | LOOP TILL END OF SUSPENSE LIST