

SYSTEM PROGRAMMING TECHNIQUES

FNDBLK

DETERMINING THE PHYSICAL ADDRESS OF A BLOCK WITHIN A DISK FILE

**Date: 5-Jan-72
File: FNDBLK.RNO
Edition: 1**

The information in this document is subject to change without notice and should not be construed as a commitment by Digital Equipment Corporation.

Actual distribution of the software described in this specification will be subject to terms and conditions to be announced at some future date by Digital Equipment Corporation.

1.0 INTRODUCTION

The following simple procedure will allow a non-privileged program to find the physical block number corresponding to a relative block, R, in a file.

1.1 Read the HOME Block

For the unit you're interested in :

```
LOOKUP HOME.SYS[1,4]
```

Read the file until you find a block whose first word is SIXBIT 'HOM'. (This should be block N+1, where N is the number of blocks per cluster.)

Save the following words:

```
10 HOMLUN - Logical unit number within the structure.
16 HOMCNP - Count-pointer for retrieval pointers.
20 HOMCLP - Address-pointer for retrieval pointers.
21 HOMBPC - Blocks per cluster.
```

1.2 Read the First RIB of the File

```
LOOKUP FILE.EXT
USETI 0
INPUT
```

N.B. - If writing, the file should be in UPDATE mode - write it once, then CLOSE, LOOKUP, ENTER. Otherwise the Retrieval Information Block (RIB) won't have been written on the disk.

1.3 Scan the First RIB for the Relative Block, R

Start with BASE=0.

Retrieval pointers start at relative word N of the RIB, where $N = RH(\text{word } 0)$.

If you find a retrieval pointer that equals 0, reread these instructions. You lost.

If you find a pointer that equals $XWD\ 0,400000+n$ you have seen a "unit-change-pointer". The following retrieval pointers, until another unit-change is found, refer to unit

n, where n is the relative unit within the file structure of the unit (eg DSKBn). All RIBs start with a unit-change-pointer as the first retrieval pointer.

If you find a pointer with LH non-0, then:

The number of blocks described by this pointer = $L = \text{HOMBPC} * K$, where K is obtained by an LDB c(HOMCNP).

If $\text{BASE} = \langle R \rangle \text{BASE} + L$, you won.

Otherwise, $\text{BASE} = \text{BASE} + L$, try the next pointer.

If the next pointer equals X'00, -i the block you are looking for is not in this RIB. To read the next RIB:

USETI -n (n=2 to read the 2nd RIB etc.)

INPUT

Set BASE equal to the contents of word 33 of the RIB and scan this RIB as described above.

If you win, the physical address equals $R - \text{BASE} + J$, where J is obtained from an LDB c(HOMCNP)*HOMBPC. The unit on which the block resides is found from the last unit-change pointer. It should be the same as HOMLUN.

2.0 The following program shows how to find the physical block number that corresponds to a relative block in a file.

```

      TITLE      FNDBLK
;THIS PROGRAM FINDS OUT THE PHYSICAL BLOCK NUMBER
;CORRESPONDING TO BLOCK #D1153 IN FILE FOO
ST:   RESET
      INIT 14
      SIXBIT /DSK/
      HOMHDR
      HALT
      LOOKUP HOMBLK
      HALT
HOMLUP: INPUT
      ILDB 1, HOMHDR+1
      CAME 1, ['HOM  ']
      JRST HOMLUP

;FOUND THE HOME BLOCK
      HRRZ 1, HOMHDR+1
      MOVE 2, 10(1) ;HOMLUN
      MOVEM ^, LUN
      MOVE 2, 16(1) ;COUNT-FIELD PNTR
      HLLM 2, CNP
      MOVE 2, 20(1) ;ADDRESS-FIELD PNTR
      HLLM 2, CLP
      MOVE 10, 21(1) ;BLOCKS PER CLUSTER
      RELEASE

```

```

;READ THE PRIME RIB OF FILE 'FOO'
  INIT      14
  SIXBIT    /DSK/
  RIBHDR
  HALT
  LOOKUP    FOO
  HALT
  USETI     0
  INPUT
  SETOM     RIB
;INDICATE WE JUST READ
;THE PRIME RIB(-1)

;FIND THE RETRIEVAL PNTRS
  ILDB      1,RIBHDR+1
  ADD       1,RIBHDR+1
  MOVE      2,(1)
  TRZ       2,400000
;RH=LOC OF POINTERS
;l=LOC OF 1ST POINTER
;SHOULD BE A UNIT-CHANGE
;1ST UNIT (WITHIN STR)
;OF FILE
  ADDI      1,1
;POINT TO 1ST REAL
;POINTER
  SETZ      3,
;3 IS BASE ADR. OF THE
;POINTER
  HLL       1,CNP
;1 IS A POINTER TO
;POINTER COUNTS
LOOP:  LDB      4,1
  IMULI     4,(10)
;COUNT FIELD
;TIME NUMBER OF BLOCKS
;PER CLUSTER
  ADD       4,3
  CAIG      3,+D1153
;TOP ADR+1 OF PNTR
;IS 1153 PAST START
;OF PNTR
  CAIG      4,+D1153
  SKIPA     3,4
; AND NOT PAST END
;NO; SET 3=START OF NEXT
;POINTER
  JRST      FOUND
;FOUND THE POINTER
  ADDI      1,1
;STEP TO NEXT POINTER
  SKIPN     4,(1)
;IS THERE ONE?
  HALT
;THAT'S TOO BAD
  CAIN      4,-1
;PICKED UP RIBCOD WORD?
  JRST      NXTRIB
;YES, PROPER POINTER IS
;PROBABLY IN EXTENDED RIB
  TLNE      4,-1
;IS IT A UNIT-CHANGE?
  JRST      LOOP
;NO, SEE IF IT'S THE
;RIGHT ONE
  TRZ       4,400000
;ZAP THE EXTRA BIT
  MOVE      2,4
;AND SAVE THE UNIT NUMBER
;IN 2

;HERE TO READ THE NEXT RIB FOR FILE 'FOO'
NXTRIB:  SOS      5,RIB
;POINT TO NEXT RIB
  USETI     0(5)
;USETI TO NTH RIB

```

```

INPUT
ILDB      1,RIBHDR+1      ;RH = OFFSET OF POINTERS
MOVE      6,RIBHDR+1      ;START OF DATA BLOC
MOV       3,33(6)         ;SET BASE TO BASE OF THIS
                                ;RIB
ADD       1,RIBHDR+1      ;1=LOC OF FIRTS POINTER,
                                ;THIS RIB
MOVE      2,(1)           ;GET THE FIRST POINTER
HLL       1,CNP           ;1 IS A POINTER TO
                                ;POINTER COUNTS
JRST      LOOP           ;NOW GO SCAN THIS RIB

FOUND:    CAME           2,LUN      ;IS IT THE RIGHT UNIT?
          HALT           ;NO
          HLL           1,CLP      ;YES, SET TO
                                ;ADDRESS-POINTER
          LDB           4,1
          IMULI        4,(10)      ;1ST BLOCK DESCRIBED BY
                                ;POINTER
          SUBI         3,+D1153    ;DISTANCE FROM START
                                ;OF POINTER
          SUB          4,3         ;4 CONTAINS THE PHYSICAL
                                ;BLOCK!!!!

EXIT
;STORAGE
HOMHDR:   BLOCK        3
RIBHDR:   BLOCK        3
HOMBLK:   SIXBIT      /HOME/
          SIXBIT      /SYS/
          0
          XWD         1,4
FOO:      SIXBIT      /FOO/
          0
          0
          0

RIB:      0
CLP:      0
CNP:      0
LUN:      0
END       ST

```

