5/1/69

disk representation of file

pointer block

data block

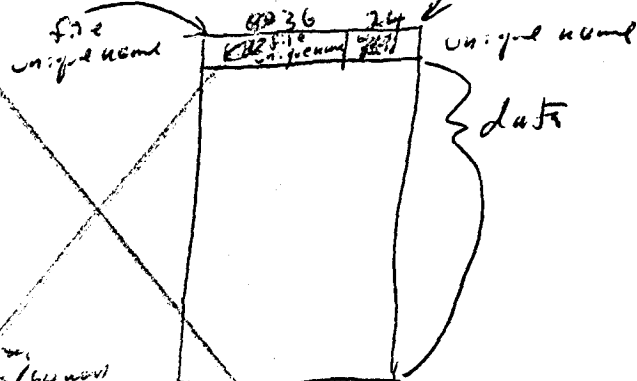[data diagram with "file unique name" pointing to block labeled "8036 24", "unique name", "data"]

disk address

file unique name
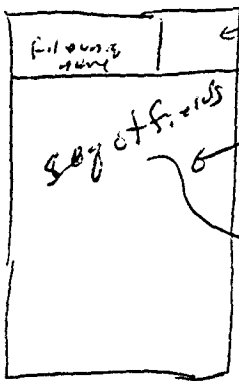
disk address
24
pacred
∴ 128 disk
addresses / 64 word block

object fields
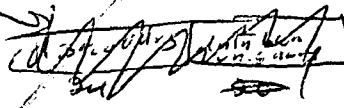
pointer blocks in fixed area, fixed location
data written anywhere and fix alloc tables

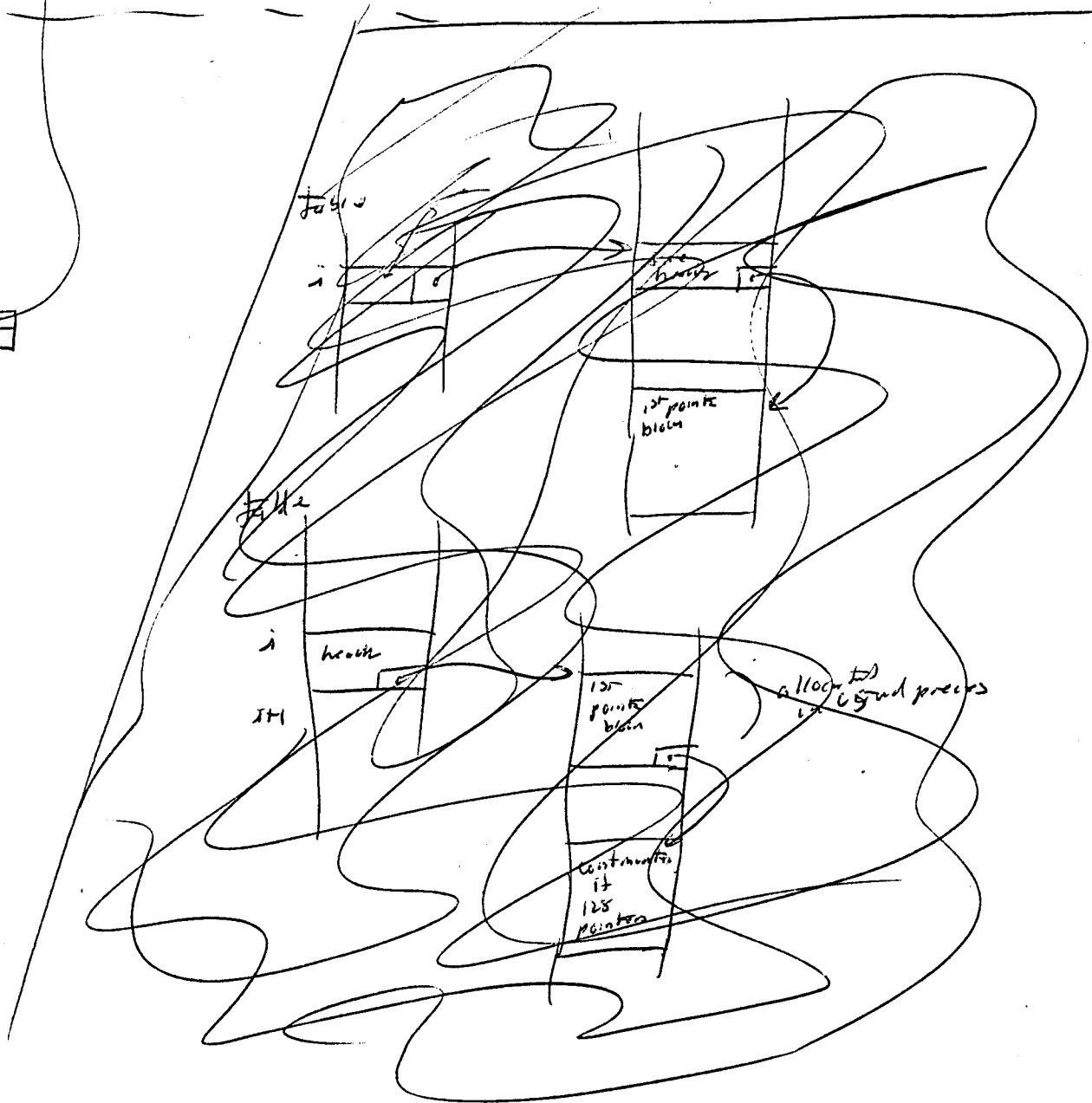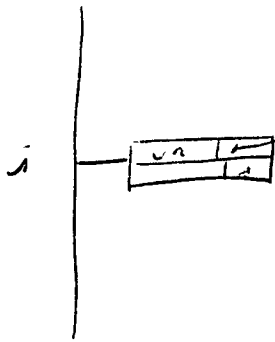[handwritten box with calculations: 44 bits data block, 64 point block, 64 bits data block, 256, ...]
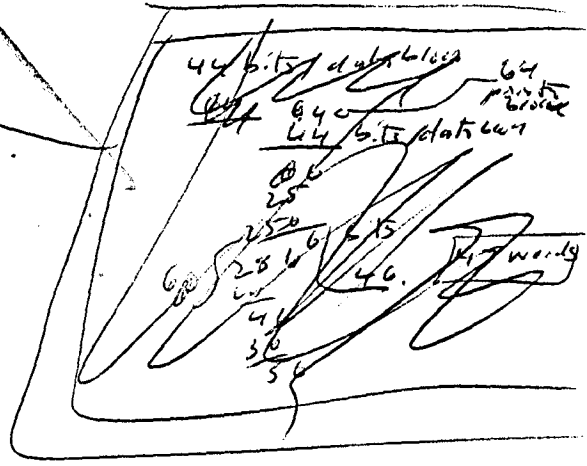
header of
a file (or
other objects) has
a list of access
keys. (How stored?)

ecs representation of a swappable file

ecs file

capability

user has

system has

c-list

i

index

1st pointer
block

1st pointer
block

head

allocated pages

contains
it
128
pointers

table

5/22/67 also indicates if 2nd level file 0

file header

2 pieces of 1st pointer block if 128 wds 1 piece if 16 wds

disk address of point block

it points to pointer block disc address (24) 2 pointers (18+2) (if 128 wd)

| 24 | 18 | 18 | 6 |

if pointer to data block
old disk address (24)
new disk address (24)
attachment count (9)
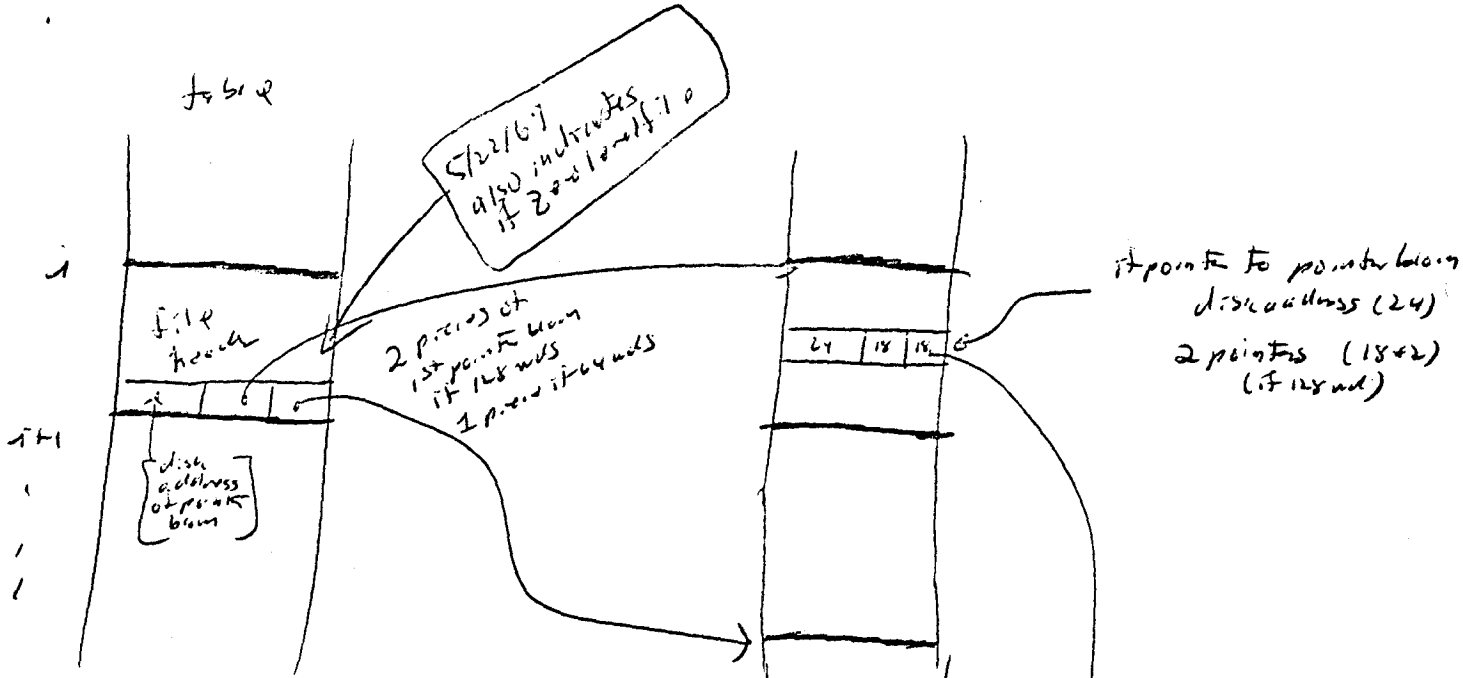status date (3)
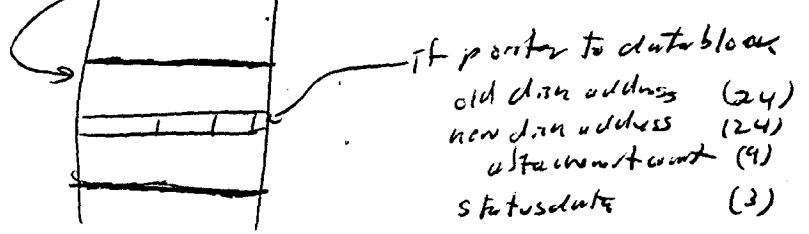
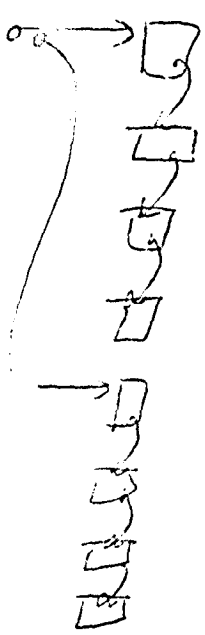5/22/69

pointer blocks allocated in 16 wd pieces

represented as

for a 128 pointer pointer block

If a process desires to attach wds of a file following happens:

   internal tables checked, ~~~~~~~~~~~~~~~

      needed pointer blocks read in (possible block at process have)

      if data blocks need to be read in, proper reads scheduled (status changed)

      attachment count on data blocks upped.

a diskread is scheduled by usual method and an internal table indexed by
    diskrequest slot # will contain pertinent data. i.e. file
                                        pointer address in ~~~
                                        file address of block

a control process will do actual copy of data to file and change status
          also wake up any processes waiting on this read. (They will be chained
    to file header]

If a process tries to read or write etc a missing block in ecs representation
    that is attached, the "disk process descriptor" is chained to the file header
    and the "ecs process" waits on an associated event channel. (associated to the disk process)

If a process tries to read or write etc on unattached data block.
          it is   attached, {read or} detached
                            {rewritten}

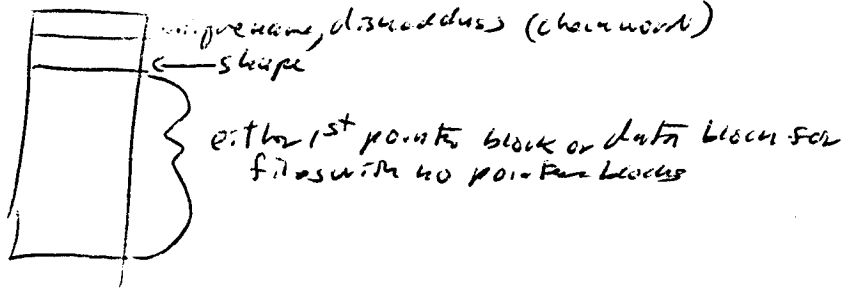| Open actually close a file | constructs header, maybe gets 1st pointer block — (if wasn't already there) etc. |

5/22/69

disk file   (on disk)

identified by, { unique name
                { disk address
                  size of header

1st block of file [diagram] ← unique name, disk address (chain work)
                            ← shape

either 1st pointer block or data block for
files with no pointer blocks

format pointer block   24 bits per disk address

[packed tight in to words   including partial
  pointers in some wds $\frac{60}{24}$ period]

_____

directories etc will be implemented in files

128 ...