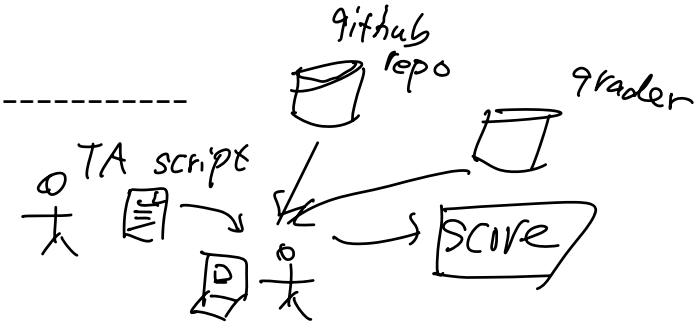
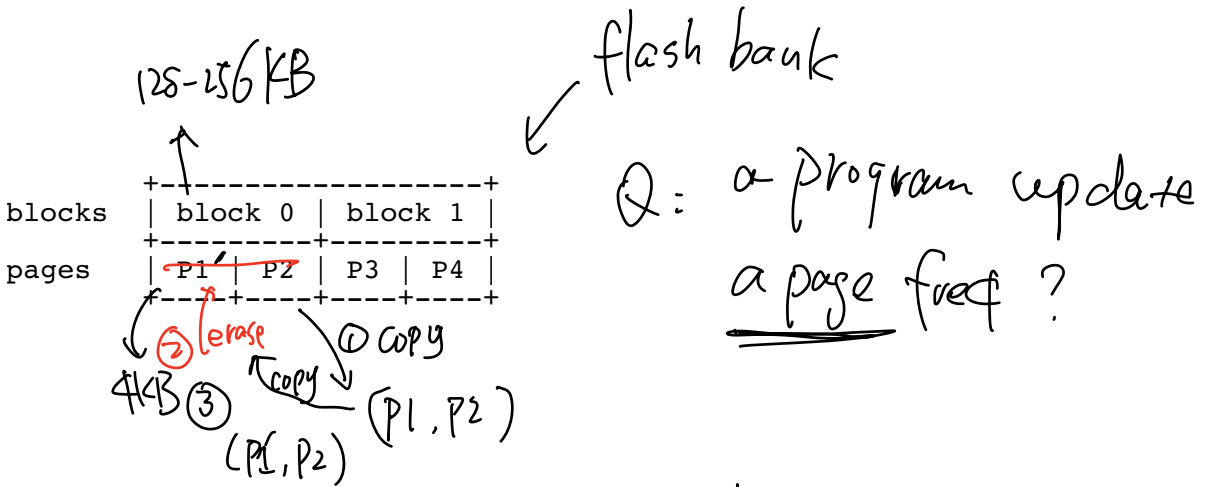
Week 13.a
CS 5600
04/03 2023

1. SSD continued
2. Intro to fs
3. Files
4. Directories

- lab3.
- lab5
- SSD.
  - read: 1 page, ~10s μs
  - erase: 1 block, ~1s ms (10x)
  - program: 1 page, ~100s μs (10x)



- Wear-out:



Q: a program update a page free?

- FTL: flash translation layer

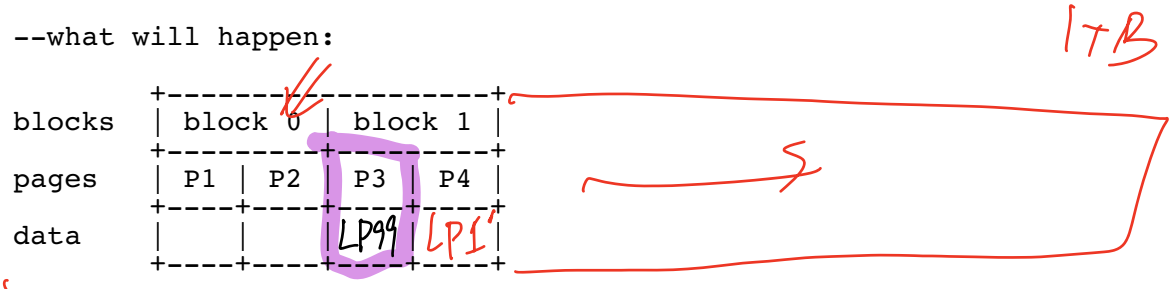
  logical page → FTL → physical page

- log-structured FTL

* an example:
  --Given a flash bank has two blocks; each has two pages.
  --there are four writes to pages:
    wirte(logic_page_1)
    wirte(logic_page_10)
    wirte(logic_page_99)

LP1, LP10, LP99

  --what will happen:



mapping:
  LP1 => ~~P1~~ P4 , LP10 => P2 P500 , LP99 => P3 P999

LP1'

Question:
  what will happen if the following op is write(logic_page_1')?

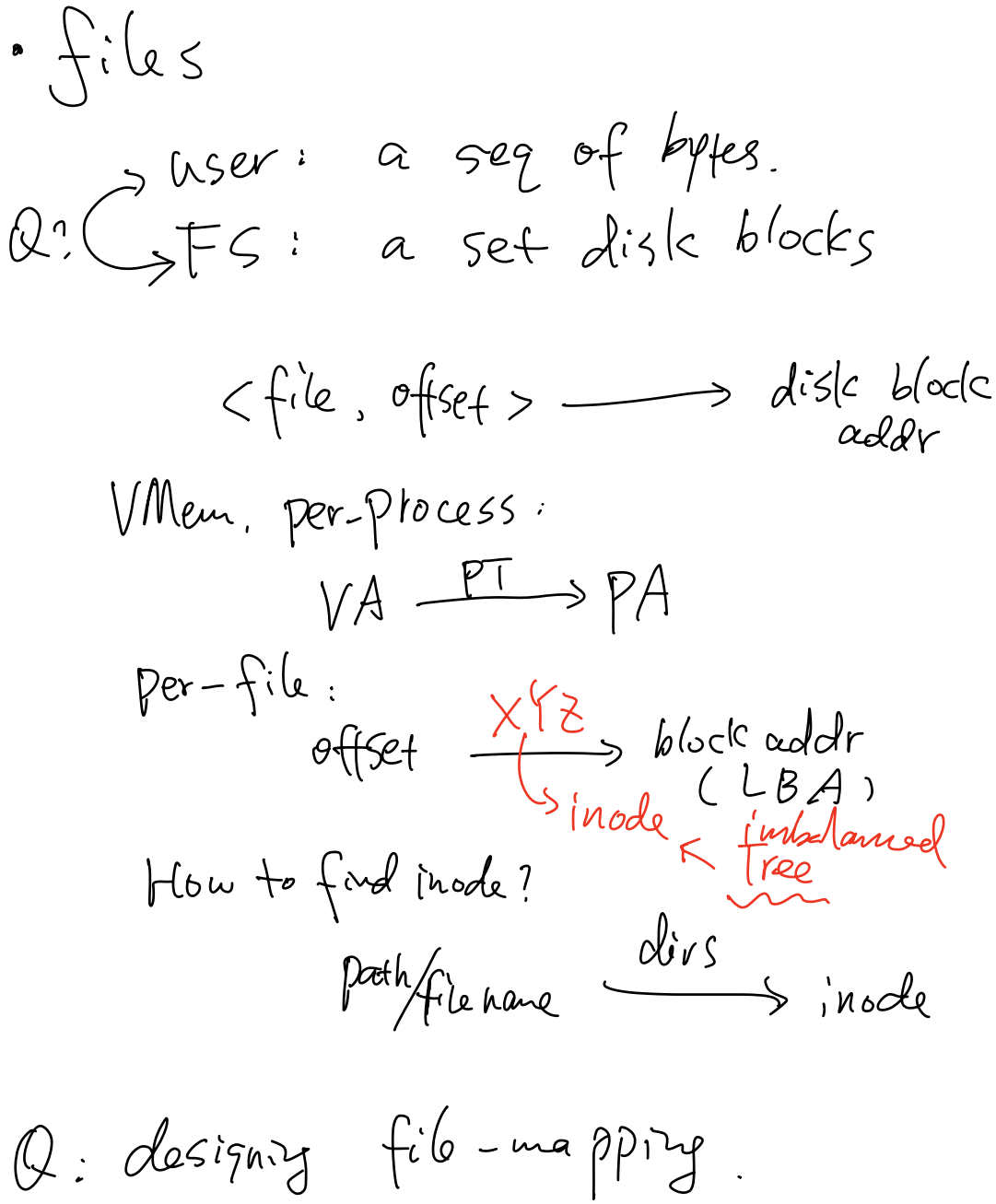
update

GC: garbage collection

- file system,
  - filetype
  - create/delete files

  - persistency &
  - "name" a seq of bytes (file)
  - human-understandable way to name (dirs)

- Where are FSes implemented?



Q: designing file-mapping.

- design parameters:

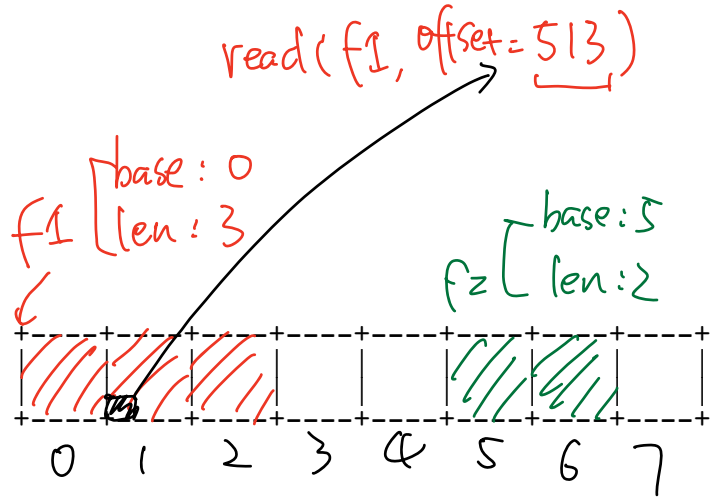
 * small files (most files are small)
   vs.
   large files (much of the disk is allocated to large files)

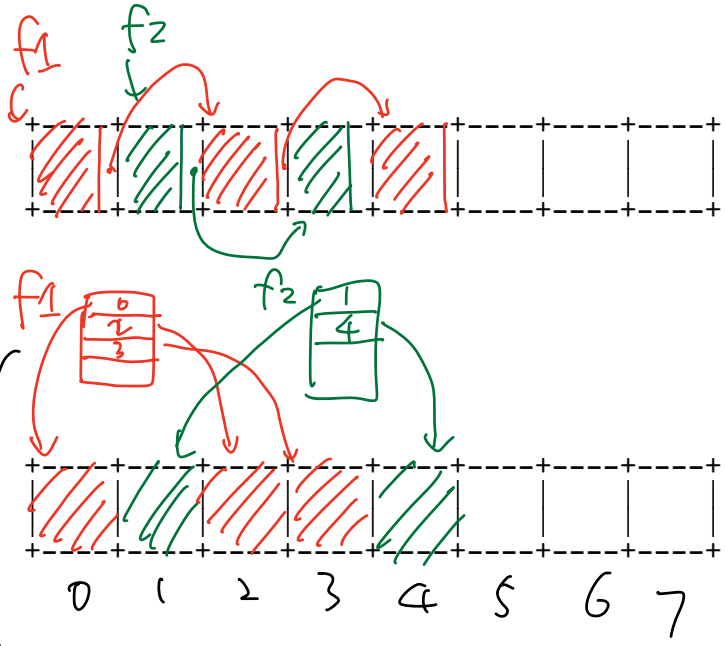
 * access patterns:
   sequential access vs.
   random accesses vs.
   keyed accesses

 * disk utilization (metadata overhead and fragmentation)

- 3 Candidates

  (A) Continuous allocation (extent-based file)
  (B) linked files
  (C) indexed files.



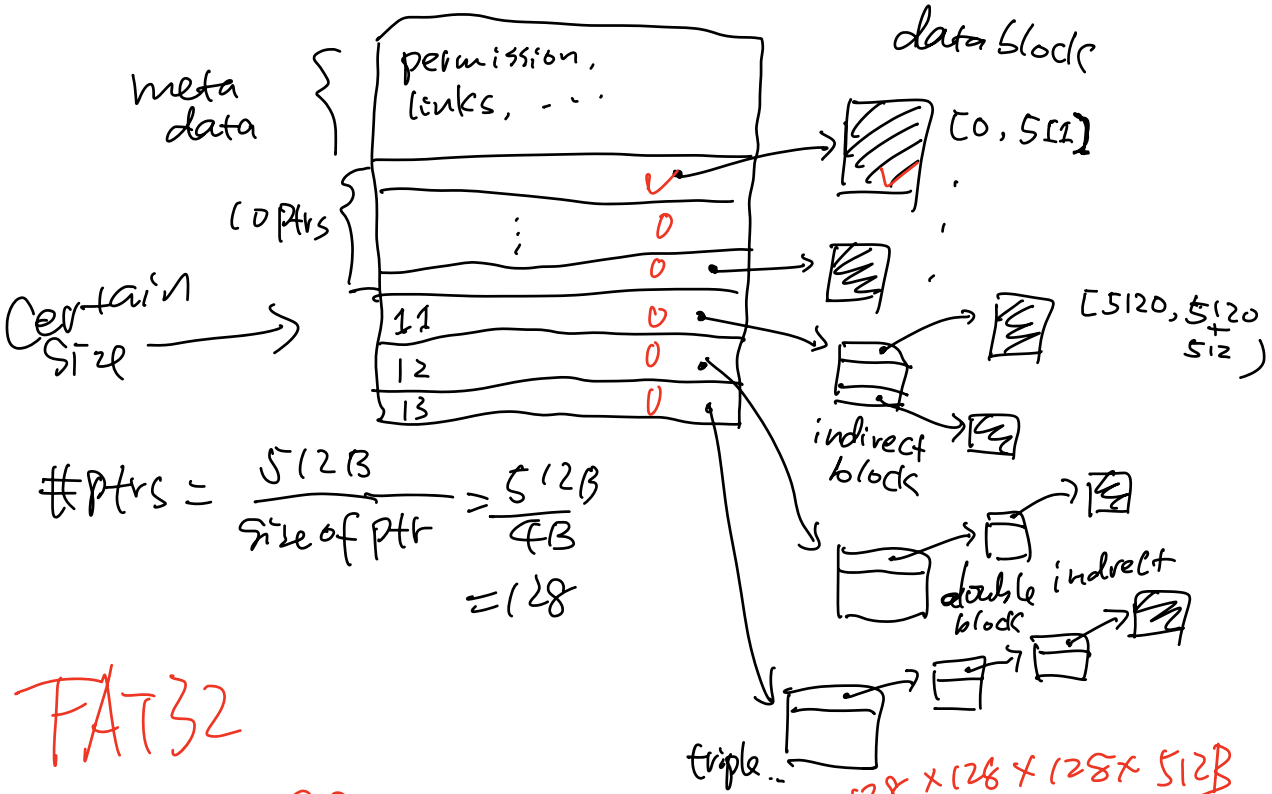


Q: How large is the table?

LARGE NUMBER

$$\#entries = \frac{\text{size of a file}}{\text{block size}}$$

. Unix inode

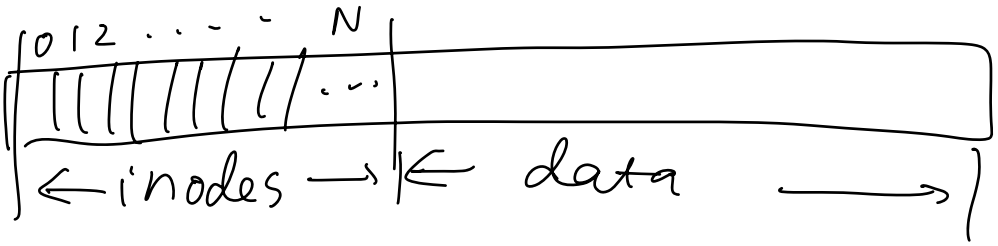


$$\#ptrs = \frac{512B}{\text{size of ptr}} = \frac{512B}{4B} = 128$$

FAT32

↳ 4GB

disk



$ df -i ~