

5/10/24

Filesys Basics

→ Abstraction

→ Files: named persistent data

stored on disk! [→ data = file content
→ metadata = information about a file

file size, owner, type,
access time, creation time,
location of data blocks (layout)

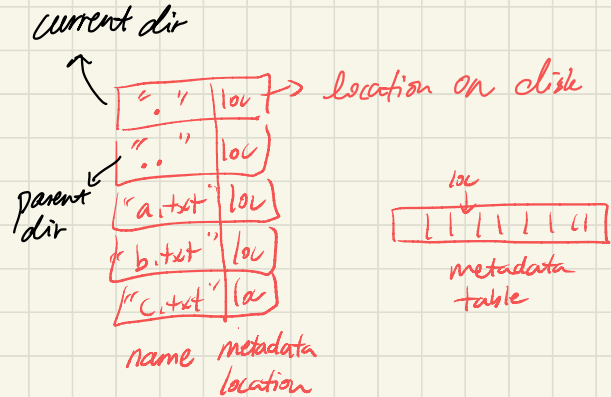
→ Directories: organize files

→ a special type of file (dir type)

→ data = list of directory entries

| | |
|----------|--------------|
| filename | metadata-loc |
|----------|--------------|

→ metadata = same as file



loc: could be just disk block #,
or could be index into a metadata table

→ Path `"/usr/jialin/..."` absolute path (starts from root)

`"jialin/Downloads/..."` relative path (starts from current working directory)

read `"home/tom/foo.txt"`

① read in metadata for root dir

File 2
"/"

| | |
|------|-----|
| bin | 737 |
| usr | 924 |
| home | 158 |

② from root's metadata, read root's data block

→ File 158
"/home"

③ from root's data, locate & read in home's metadata

| | |
|------|-----|
| mike | 682 |
| ada | 818 |
| tom | 830 |

④ from home's metadata, read in home's data block

→ File 830
"/home/tom"

⑤ from home's data, locate & read in tom's metadata

| | |
|---------|-----|
| music | 320 |
| work | 219 |
| foo.txt | 871 |

⑥ from tom's metadata, read in tom's data


→ File 871
"/home/tom/foo.txt"

⑦ from tom's data, locate & read in foo.txt's metadata

The quick brown fox jumped over the lazy dog.

⑧ from foo.txt's metadata, read in foo.txt's data!

Filesys Implementation

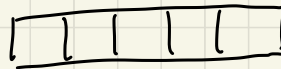
→ manage disk blocks & allocation 

→ track blocks/sectors usage via a **bitmap** *★ needs to be persisted, spans multiple contiguous blocks*

→ track metadata for each file/directory

→ **metadata table**

*also known as
inode, file record,
file header*



reserve sectors for metadata

→ store metadata together b/c of locality

→ also introduce a layer of indirection (inode #)

→ metadata of the filesystem

→ **superblock**: stored at a known location *(block #)* (# of blocks)

→ tracks bitmap region (starting block #, size of bitmap)

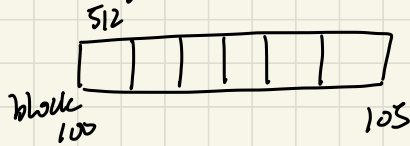
→ tracks metadata table

→ plus other fs info (block size, fs format)

Data Layout

→ how data is organized & stored on disk

→ **contiguous allocation** : store data in contiguous blocks



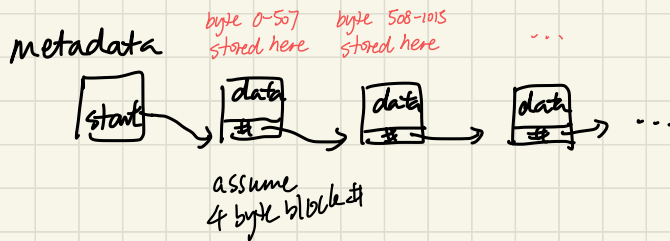
→ simple, small storage space

→ fast to locate any file offset data

$$\text{e.g. offset } 1000 = 1000 / 512 + \text{start} = 101$$

→ hard to grow

→ **linked allocation** : store data in any free block, each data block stores a pointer (disk block #) to the next data block



→ easy to grow, need to read

lots of blocks to find the desired data block