CS744

Lecture 18

1. practice lab exercises
2. Lab 4
   - pthreads
   - http
3. midsem
   - ~~12th Oct~~
   - 11th Oct
   - Sunday 2pm

Lecture 14

* the file abstraction

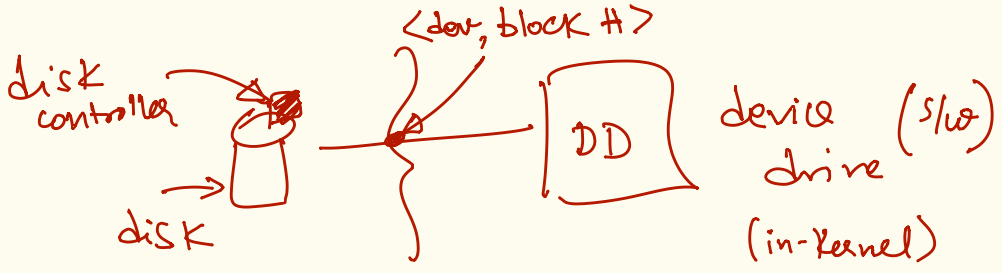
- inode
  - naming
  - access permissions
  - timestamps
  - size
  - datablocks — set of blocks storing the file "data".
    - DIRECT — linear order of blocks. {b23, b44, b79, ...} — block ids
    - INDIRECT — block which stores direct mappings.
    - DOUBLE INDIRECT
    - TRIPLE INDIRECT

$\frac{512KB}{512} \approx 1KB$

---

* disk / storage

the ISA (h/w) abstraction — fixed-sized blocks



~~DD~~ disk ⟷ DD

pio ~ in/out

blocks/port mmio

cpu, instructions

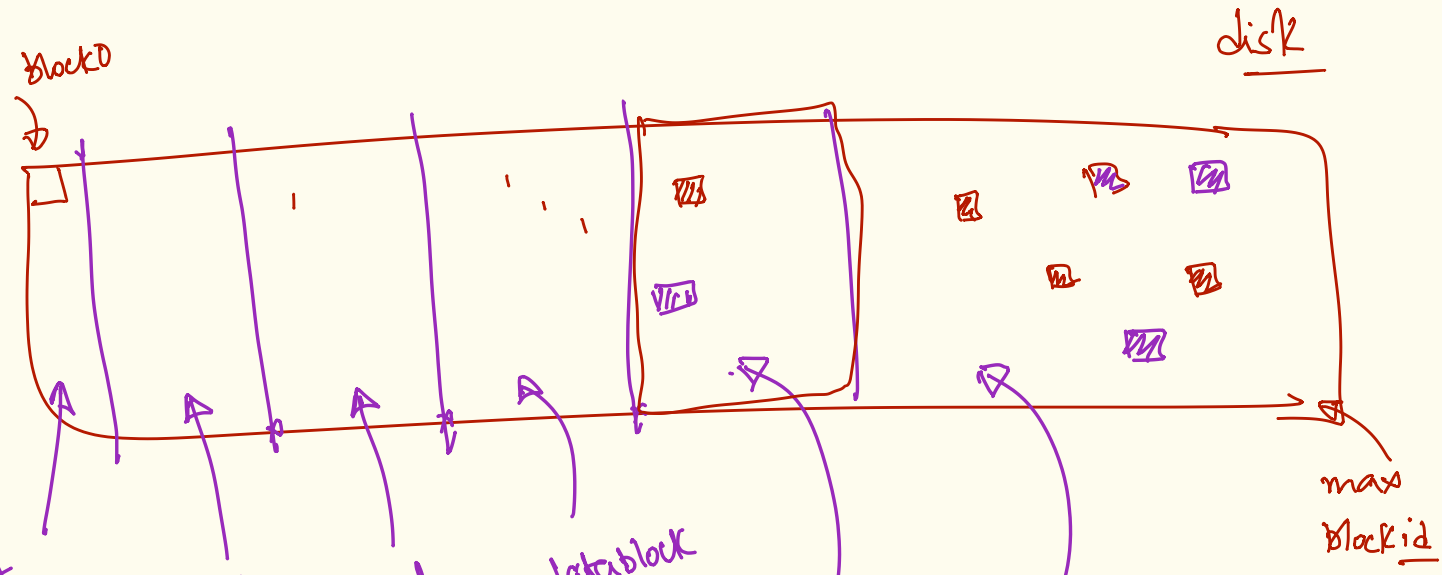
ISA

reg1 ← R/w

reg2 ← block # ← disk abstraction!

reg3 ← memory address

in/out



boot block

information for booting purpose (stores loc^n of the OS program!)

super block

device & FS-specifics.

- which FS?
- where are boundaries of meta & data
- size of inode etc.

inode bitmap

bookkeeping free/used inodes

datablock bitmap

bookkeeping free/used datablocks

inodes

inode information

datablocks

set of all data blocks.

int

```
fd = open("foo.txt");
read(fd, buf, len);
write(fd, ...);
```

open

involved.

- read disk metadata to check if filename ↔ inode exists
- interpret inode for access permission.
- creates an in-memory copy of inode & setup fd & file object

```
newfd = dup(fd);
```

fd table / process

0 STDIN
1 STDOUT
2 STDERR
6
7

created on open.

file object

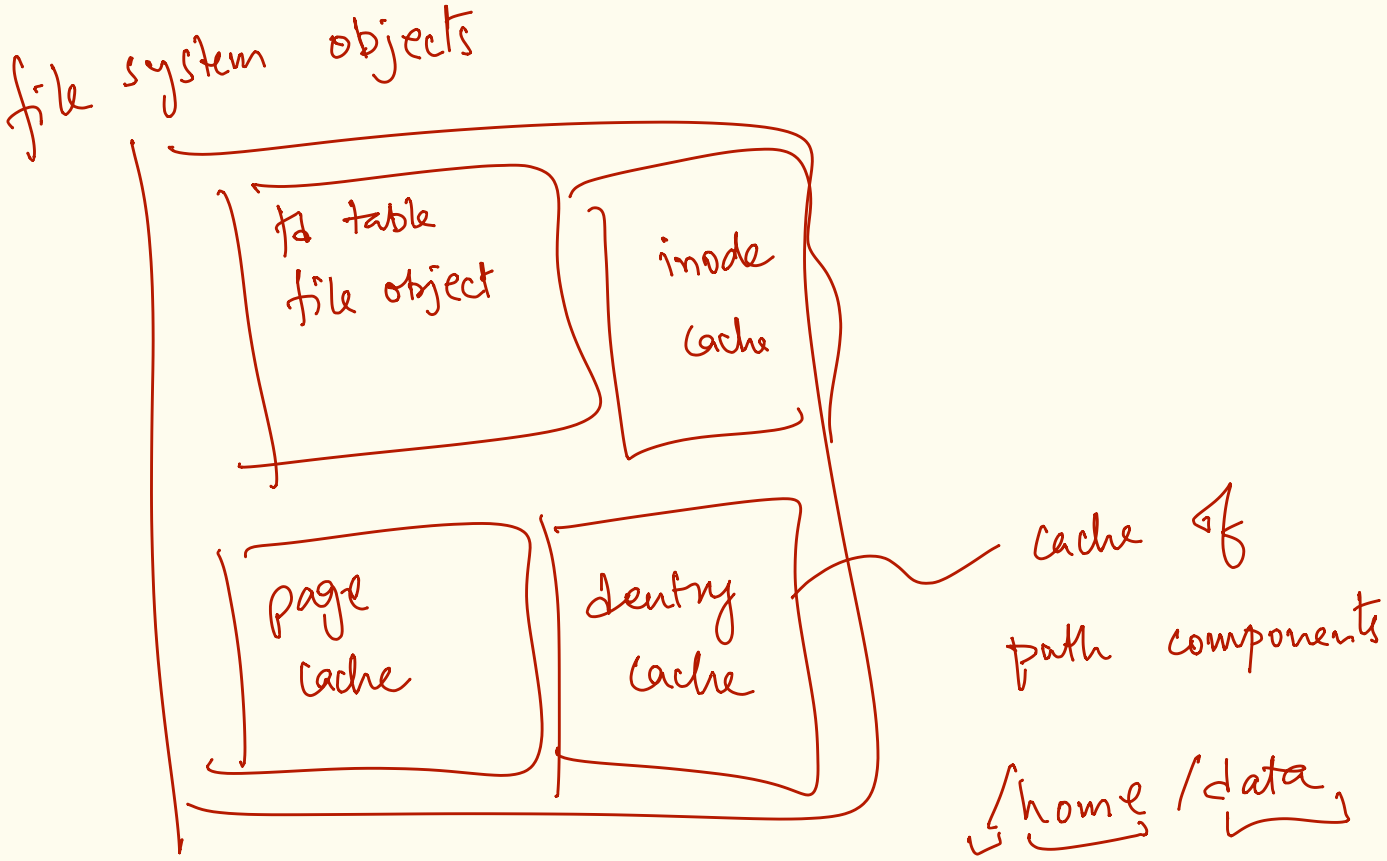
refcnt
offset = 0
inode ptr

inode cache

in-memory copy of inode on disk

<inode, offset>

page cache (disk / buffer cache)


file system objects
fd table
file object
inode
cache
page
cache
dentry
cache
cache of
path components
/home /data