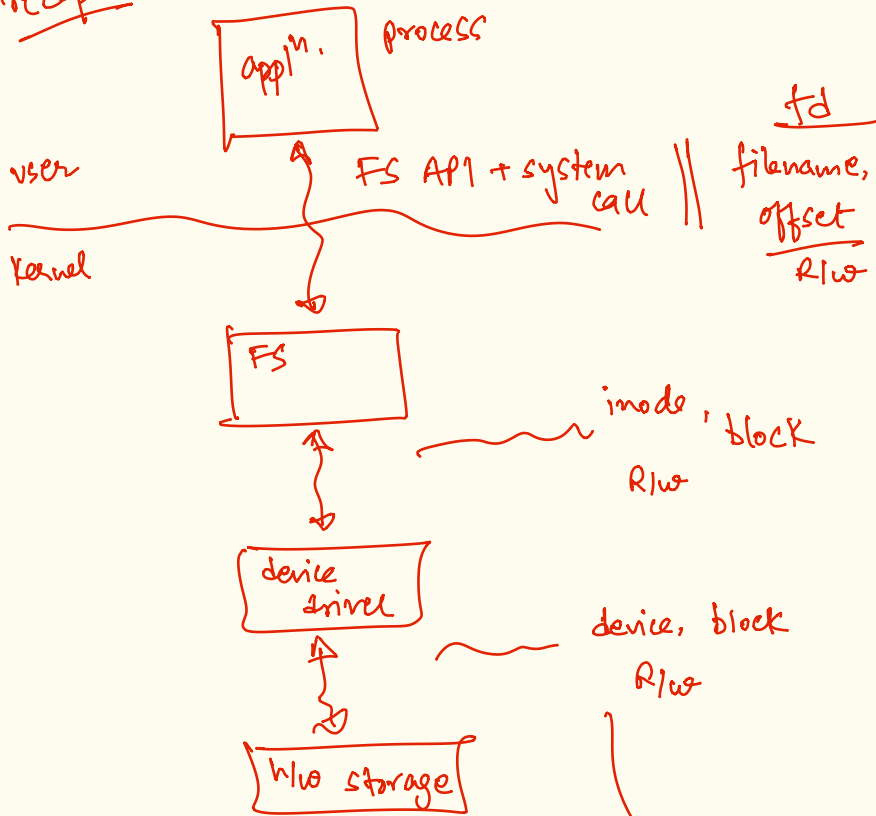


# Lecture # 23

# the file system

recap



file system requirements list.

- uniform interface / abstraction across devices
- persistence
- reliability
- performance
- "manage" the disk storage

(HDD, flash, tape, n/w storage)

simple device / disk interface



— busy  
idle

R  
W

addr.

num

sample device driver pseudo-code

```
while (status == BUSY);
```

```
cmd ← R;
```

```
maddr ← buffer;
```

```
blocknum ← block num;
```

```
status ← busy;
```

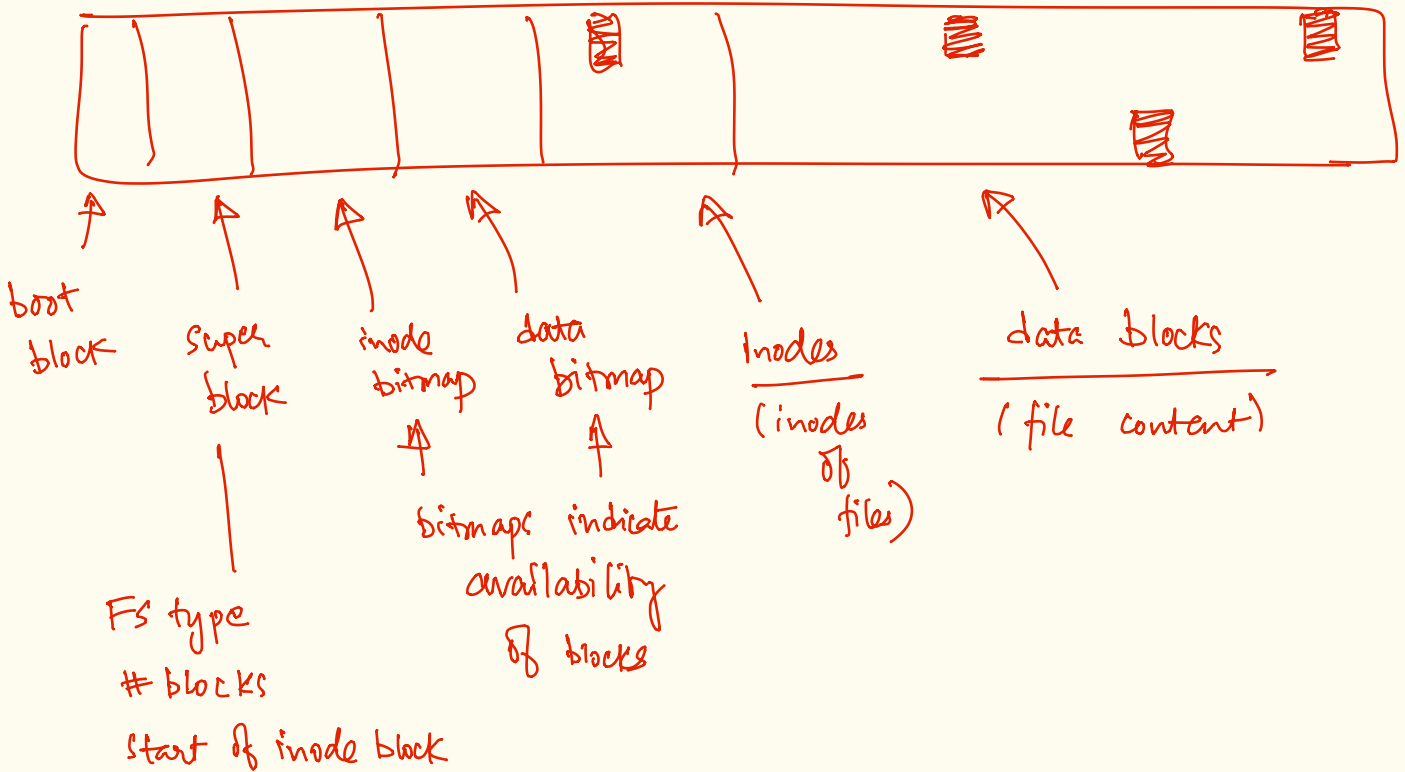
address in memory to store disk content from where to ~~copy~~ write to disk.

// ~~data~~ block on disk

# the file abstraction design + FS operations.

- Key element: inode — associated with a file  
└ metadata for a file  
└ size, permissions, timestamps  
└ datablocks

FS layout on the disk via a FS



~~fd = read (file~~  
`read (fd, buf, 10);`

⊙ `fd` → inode number  
`inode` → contents of inode  
`offset` → appr. data block (from inode info.)

# e.g: read (fd, buf, 10); // assume inode number

~~block~~ <sup>Block from disk</sup> read inode into memory <sub>(of fd)</sub> } inodes are on disk.  
of fd is 23

(block num in the inodes region of disk)

- interpret the inode

- check permissions
- check of exceptions
- offset → datablock number

- read datablock from disk to memory ↻ loop to satisfy size of read.

Q

files  
the file abstraction

← root directory

- files + directories (organization for files)

- hierarchical
- files + sub-directories.

- how are directories stored? by FS.

as files.  
contents of directory file are (filenames, inode id, number, ...)

- "/" → root directory

home	63
usr	72
bin	58
proc	123

```
# fd = open ("/home / foo.txt");
```

read block num associated with inode # 2

- read inode of '/' known inode #  
 check permissions

- read data blocks of '/'

lookup for "home" & lookup its inode num.

(blocknum string)  
 - read inode of home

↓ convert identify the block num

- read data of home

- read inode of foo.txt

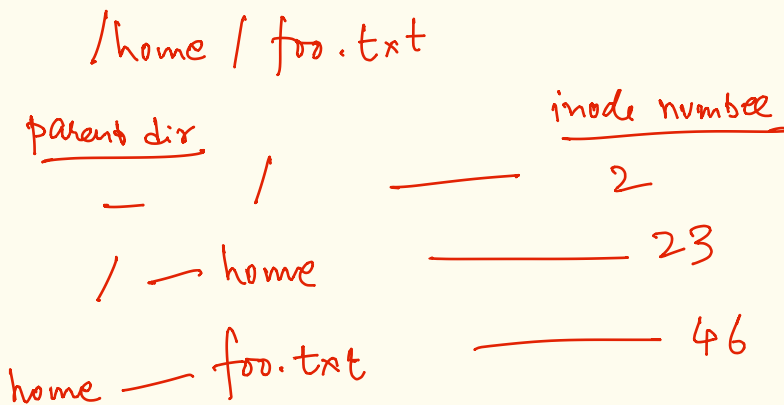
$n=3$  ← length of the path / home / foo.txt

cost of lookups =  $(2n-1)$  } costly!

① dentry cache

→ directory entry cache

in-memory cache storing path components to inode number mappings



③ disk cache page

mapping of offset to data content.

② inode cache.

in-memory store caches

inode numbers to inode mapping content