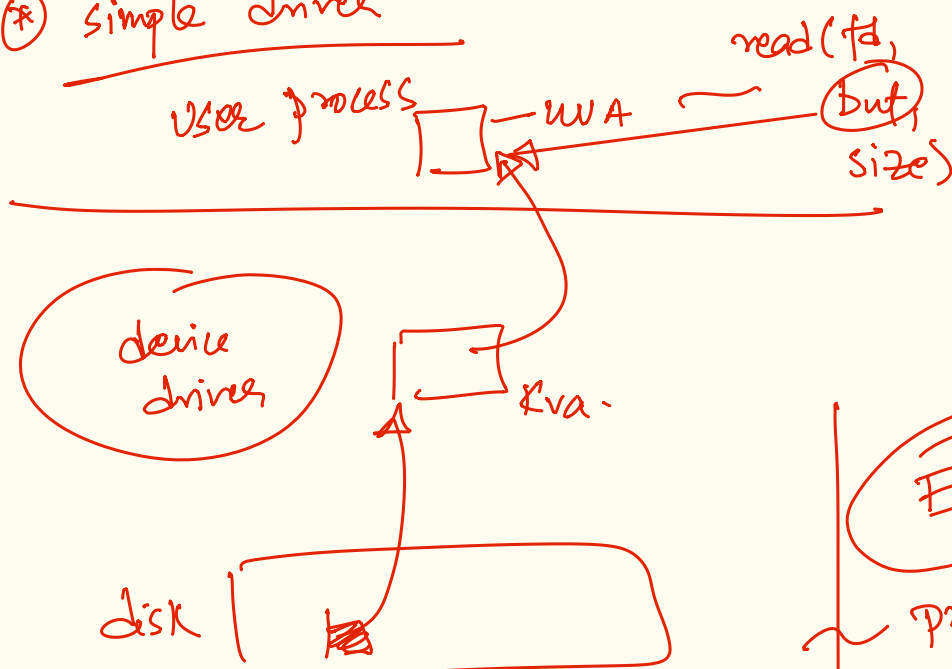


the file system design.

* simple driver



* Lab today

redux Lab 10

Lab 11

* Lab-Lite

Lab 9 redux
& Lab 6

FS

providing the file abstraction
to organize & store
data in a persistent manner.

- naming
- access control / permissions
- attributes: size, timestamps
- collection of data blocks (ordered)

FS metadata

access & mgmt. of
files.

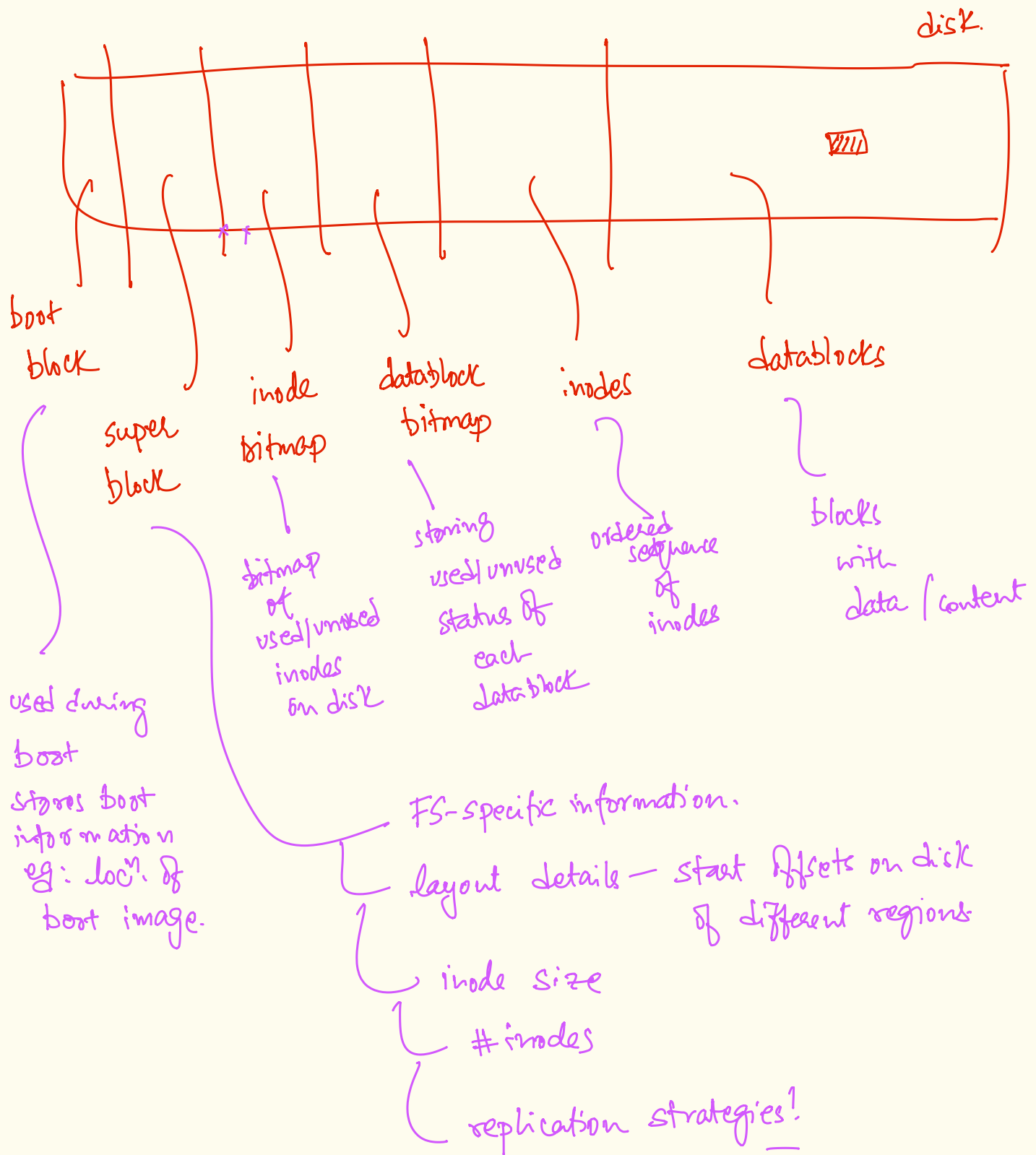
- inode

* metadata / inodes

have to be persistent!

* stored on disk?

the FS has an approach to use disk.
every technique



$fd = \text{open}("foo.txt");$

file descriptor
(number)
unix-like

sys call to
open a file
intent to use
a file.

similar to
malloc

intent / declaration
to use memory.

- (i) find inode for foo.txt on disk (if not in memory)
- (ii) check access permissions!
- (iii) create in-memory copy of the inode.
- (iv) setup fd & struct file objects.

PCB.

fd table

list
of open files



|file object|

struct file

offset
refcnt
inode*

in-memory
inode
cache

dup(4)

<fd, offset>

<device, block>