

CS202° File System II

Scheduling Note

11/21 Tuesday ✓

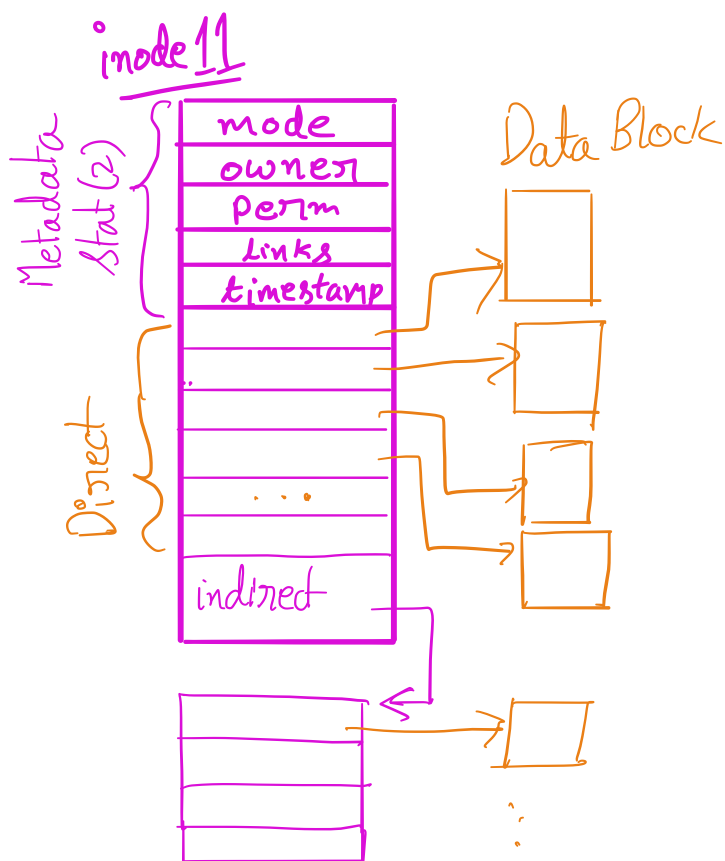
11/23 Thursday X (Turkey, Pies, etc.)

11/28 Tuesday X

11/30 Thursday ✓

Last Class

file → inode
("a") (#11)



Directories

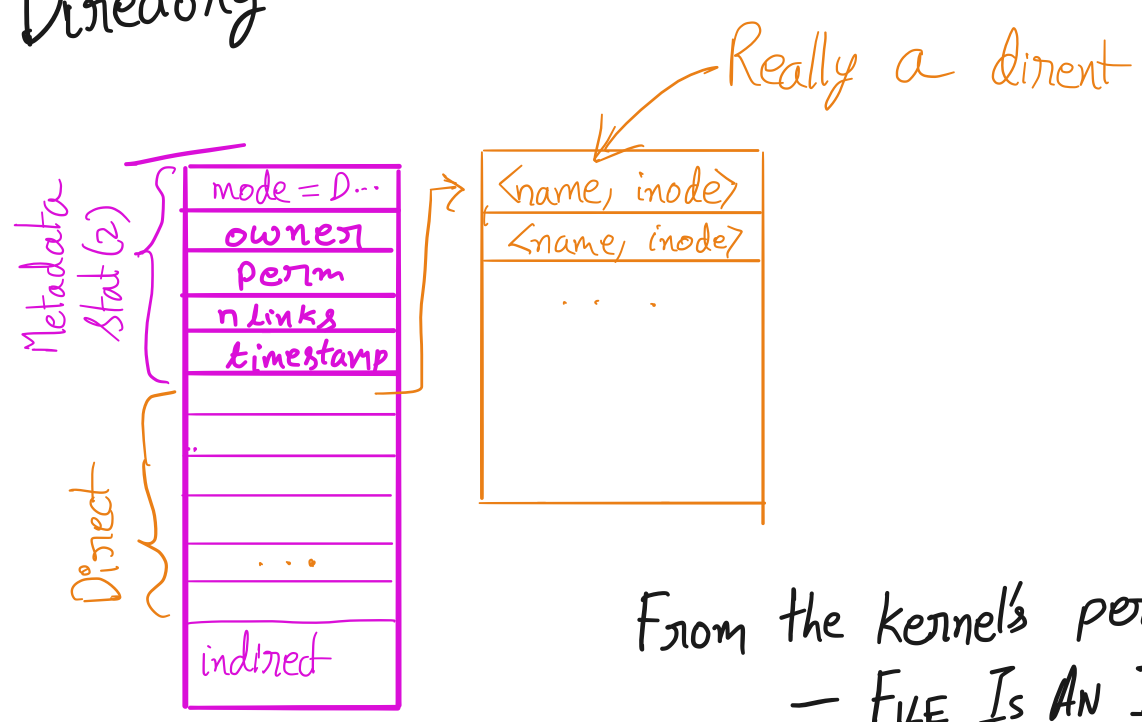
Lab 2: opendir → readdir → struct dirent
 struct dirent
 struct dirent
 ...

```

struct dirent {
    ino_t    d_ino;
    char    d_name[];
    ushort  d_reclen;
    ...
}
  
```

← Inode Number
 ← file name

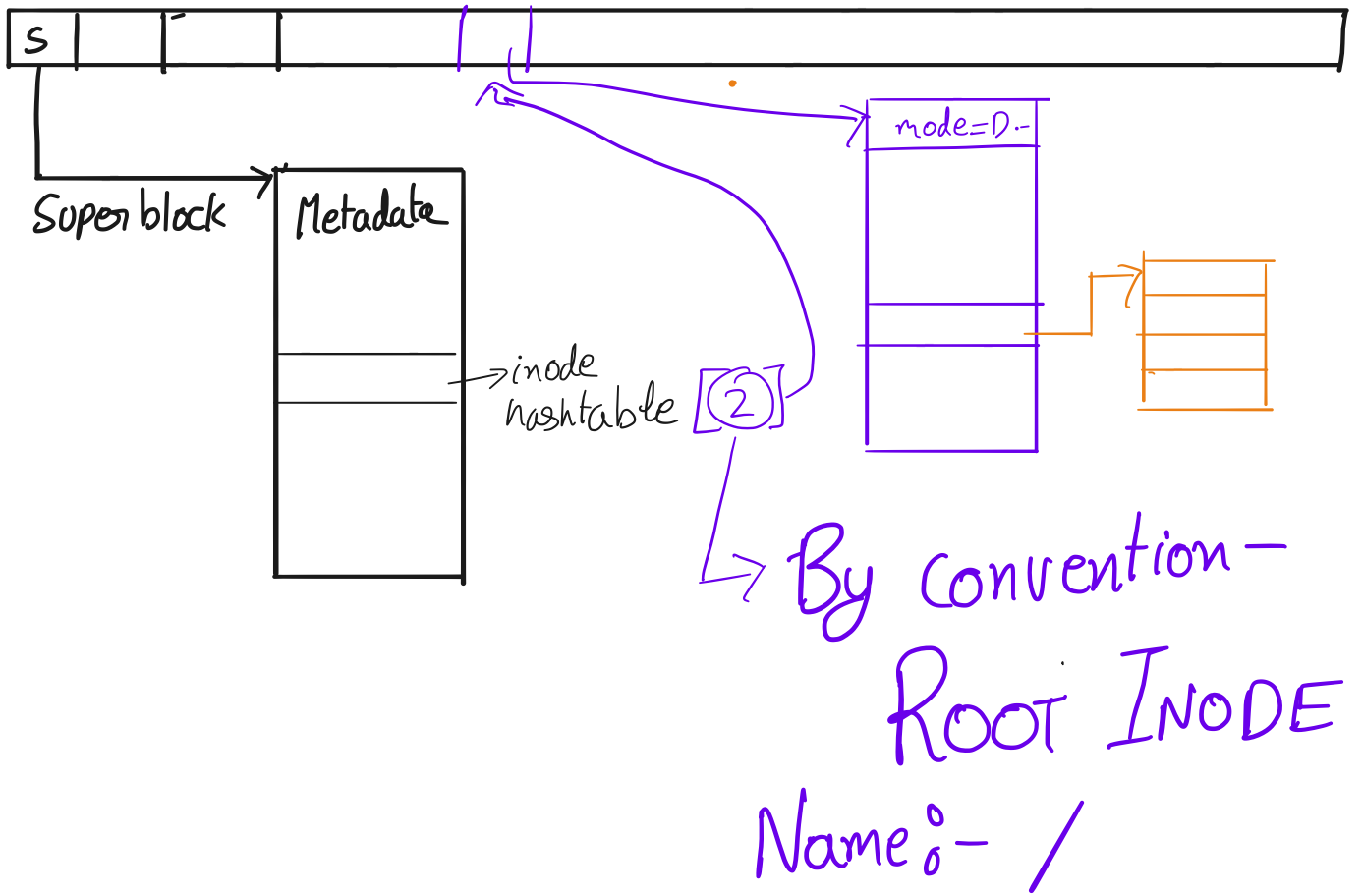
Directory



From the kernel's perspective
 - FILE IS AN INODE #
 - NAMES ARE IN DIRECTORIES

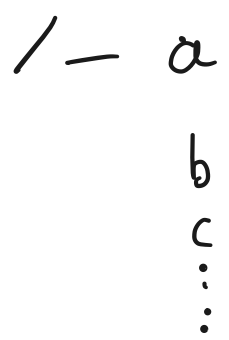
FILE SYSTEM LAYOUT (ABSTRACT)

- INTERMEDIATE STEP TO EXPLORE DIRECTORY LAYOUT



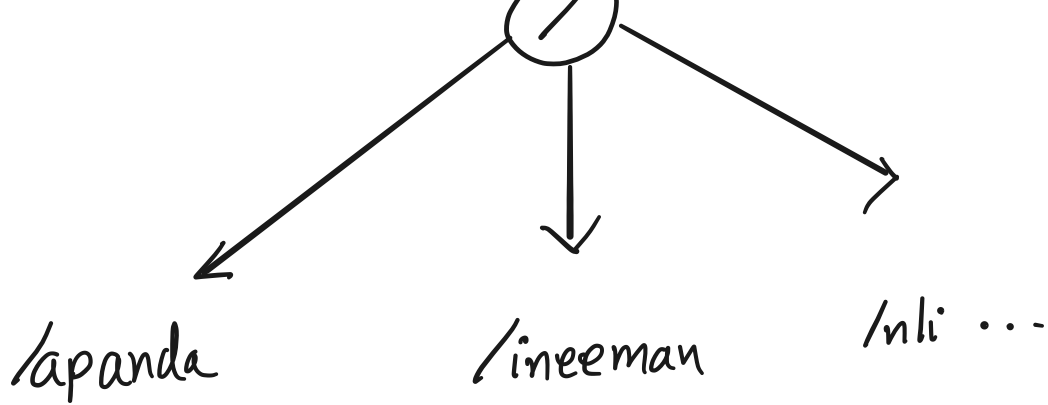
Possible directory organizations

(a) FLAT: SINGLE DIRECTORY FOR ALL



PROBLEMS?

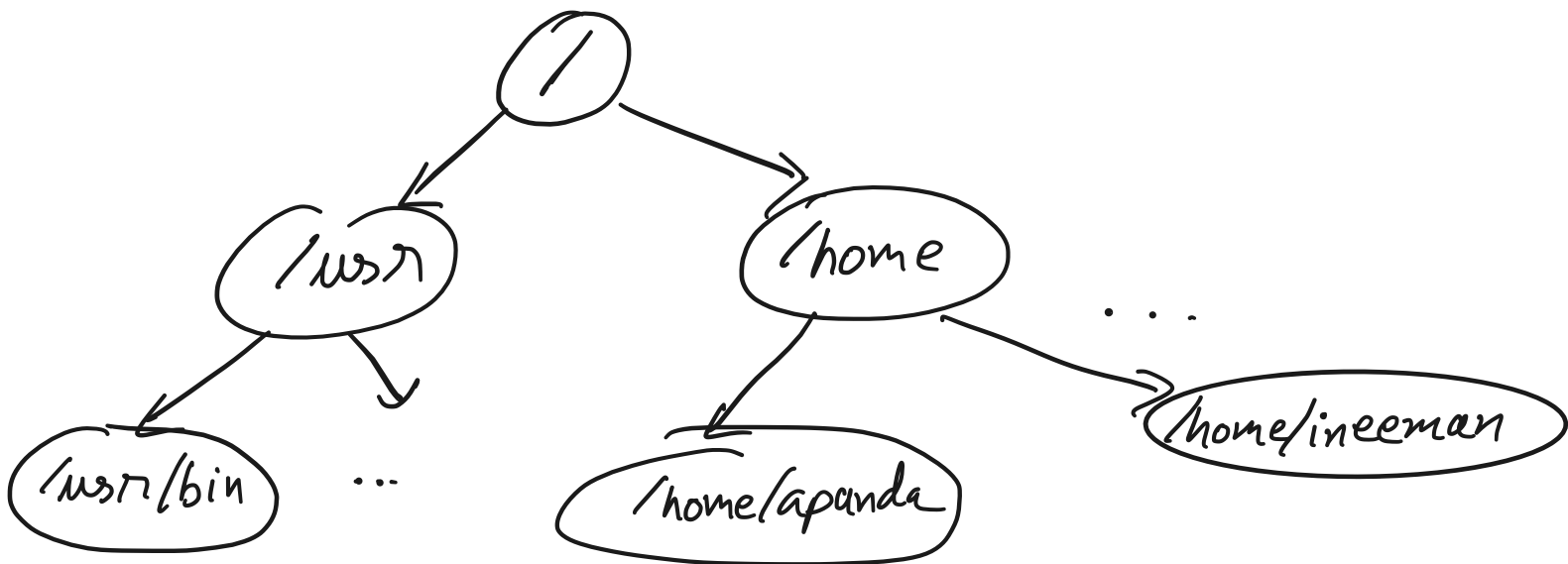
(b) DIRECTORY PER USER



BENEFITS?

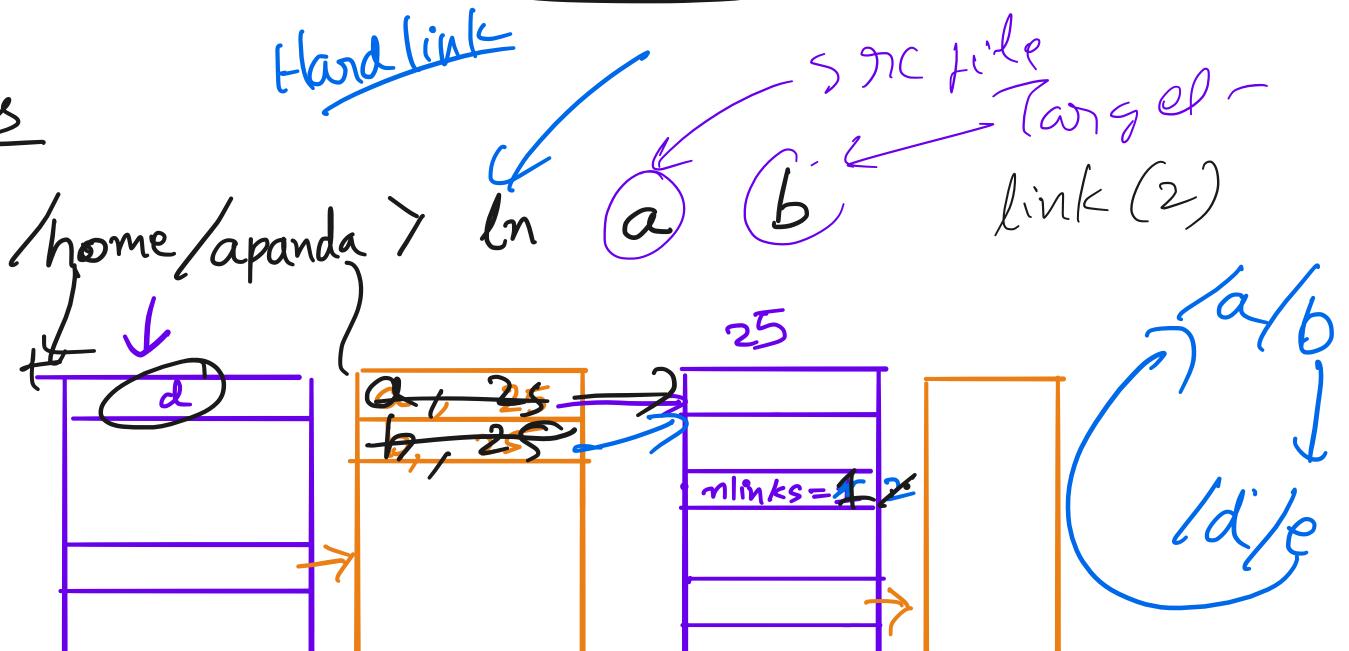
PROBLEMS

(C) Hierarchical



Links

Hard link



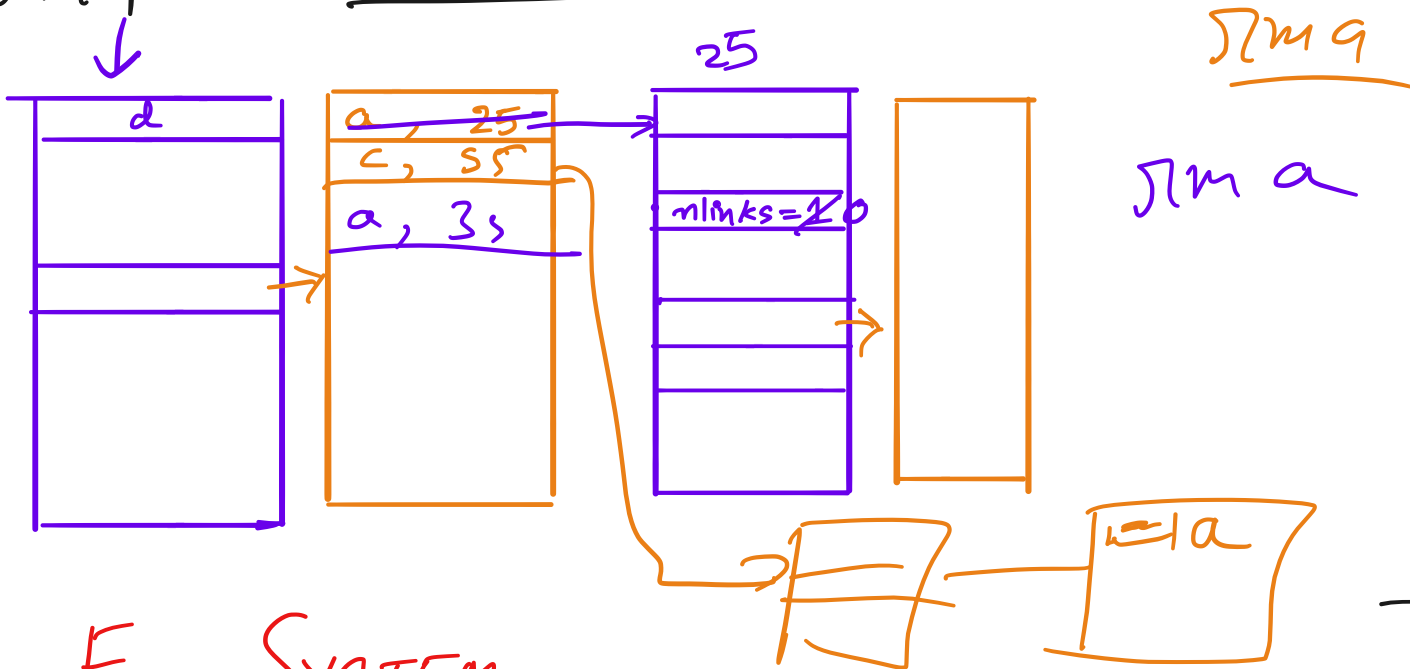


`/home/apanda> rm b ; rm a`
 (`unlink("b")`)

`rm dir`
`rm -r`



`/home/apanda> ln -s a c`



FAST FILE SYSTEM

Learning: Anatomy of a more complex file system

Goal: Used by *BSD until recently
 - OpenBSD uses an extended version (FFS 2)

`stat("/a/b") -> 42`

Design Goal: Improve locality for

- Directories

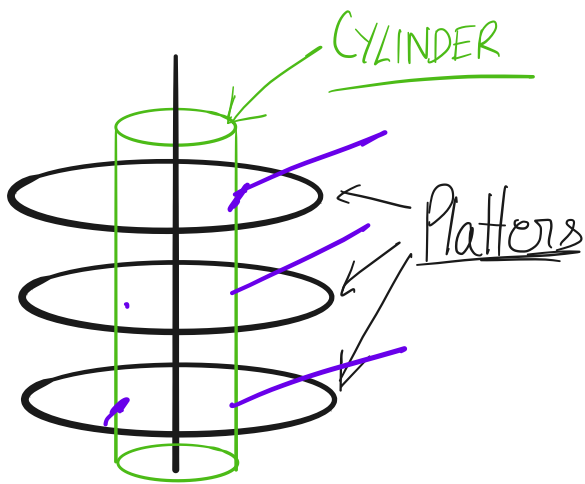
[FILES IN A DIRECTORY OFTEN
ACCESSED TOGETHER: THINK ls]

- Files

[GOOD FOR SEQUENTIAL ACCESS]

How? ALLOCATE BLOCKS BETTER.

CYLINDERS: OUR FRIEND FROM TWO CLASSES AGO



Observations:

- MORE STORAGE (SECTORS) THAN ON A SINGLE TRACK. Why?
- NO SEEK NECESSARY WHEN READING FROM THE SAME CYLINDER. Why?

CYLINDER GROUP

CORE ABSTRACTION: COMBINE BLOCKS FROM 1 OR MORE ADJACENT CYLINDERS

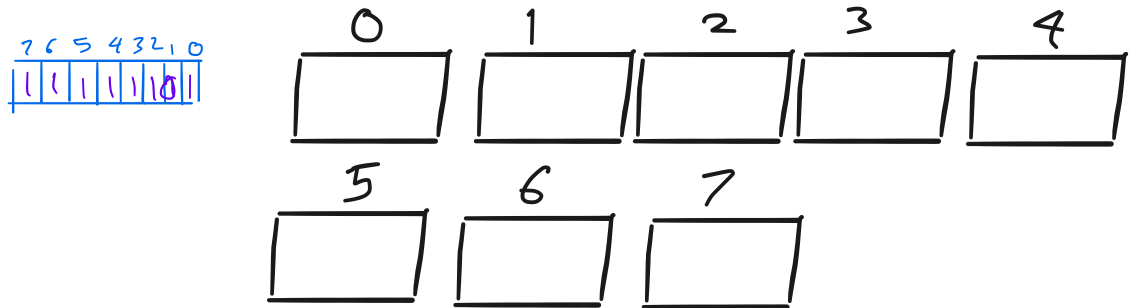
CG	S	DESC	Block Bitmap	inode bitmap	INODES	DATA BLOCKS
----	---	------	--------------	--------------	--------	-------------

FFS



Bitmaps : A quick way to track what blocks & inodes are free

YOU WILL IMPLEMENT ONE IN
STEP 1 OF LAB 5



(a) Allocating directory

↳ Place in cylinder group with largest # of free inodes

(b) Allocating file inode

↳ If possible, place in same cyl. group as parent directory

(Hash to choose random cyl. group otherwise)

© Data blocks

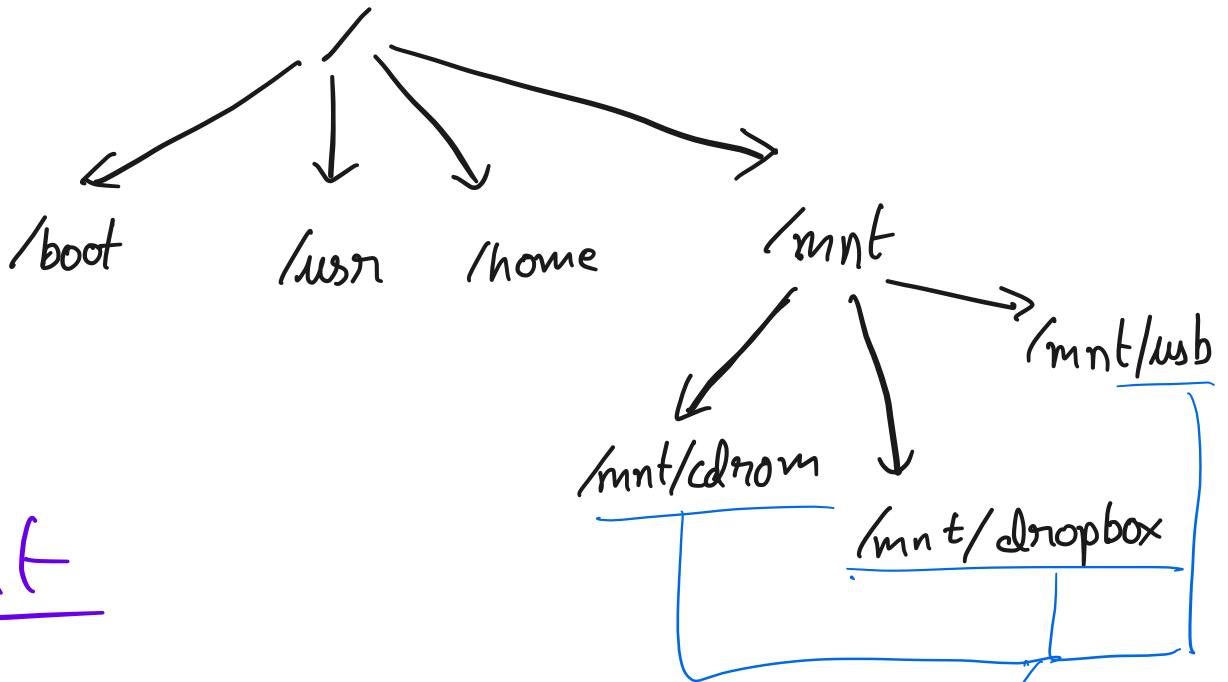
File $< 48\text{KB}$: In same cyl. group as inode

$> 48\text{KB}$: Redirected to another cyl. group; prefer one with large number of free data blocks

Buffer Cache

Why multiple super blocks?

VFS



mount

file sy

open("/home/out.txt")

open("/mnt/dropbox/out.txt")

Might all have
different file
systems

inode → File metadata, authoritative copy on disk.

LAYOUT + CONTENTS DICTATED BY FILE SYSTEM

vnode → Logical representation in memory, providing common interface for the OS.

LAYOUT + CONTENTS DICTATED BY KERNEL

LAYOUT + CONTENTS & DETAILS

```
struct fuse_operations fs_oper = {  
    .getattr      = fs_getattr,  
    .readlink    = fs_readlink,  
    .mknod       = fs_mknod,  
    .mkdir       = fs_mkdir,  
    .opendir     = fs_open, // No difference between open and opendir.  
    .readdir     = fs_readdir,  
    .unlink      = fs_unlink,  
    .rmdir       = fs_rmdir,  
    .symlink     = fs_symlink,  
    .rename      = fs_rename,  
    .link        = fs_link,  
    .chmod       = fs_chmod,  
    .chown       = fs_chown,  
    .truncate    = fs_truncate,  
    .open        = fs_open,  
    .read        = fs_read,  
    .write       = fs_write,  
    .statfs     = fs_statfs,  
    .fsync       = fs_fsync,  
    .ftruncate   = fs_ftruncate,  
    .fgetattr    = fs_fgetattr,  
    .utimens     = fs_utimens,  
};
```

FUNCTION
POINTERS

FDT

