CS 3650 – Computer Systems
Spring 2024
Peter Desnoyers

Lecture 17, Tue Mar 12, 2024

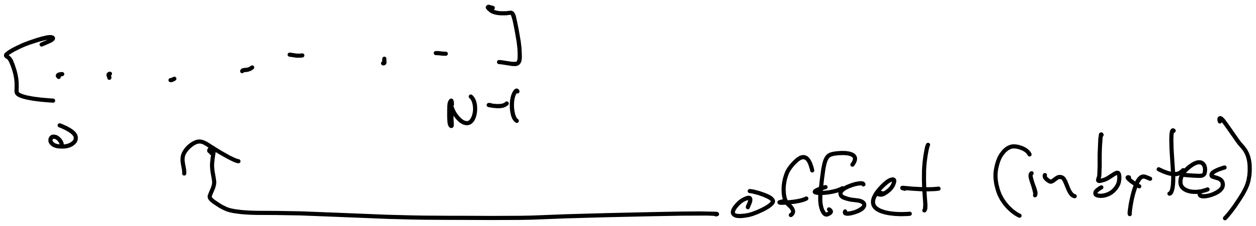
File Systems

what's a file system?

open(path, ...)
read / write
close

→ File: sequence of bytes

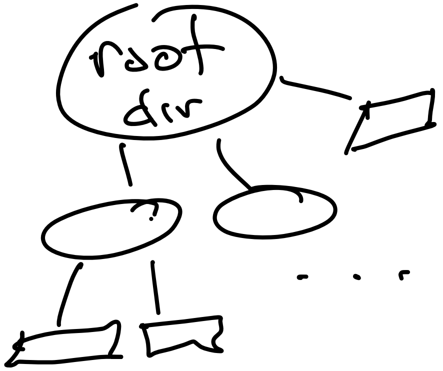
length=N



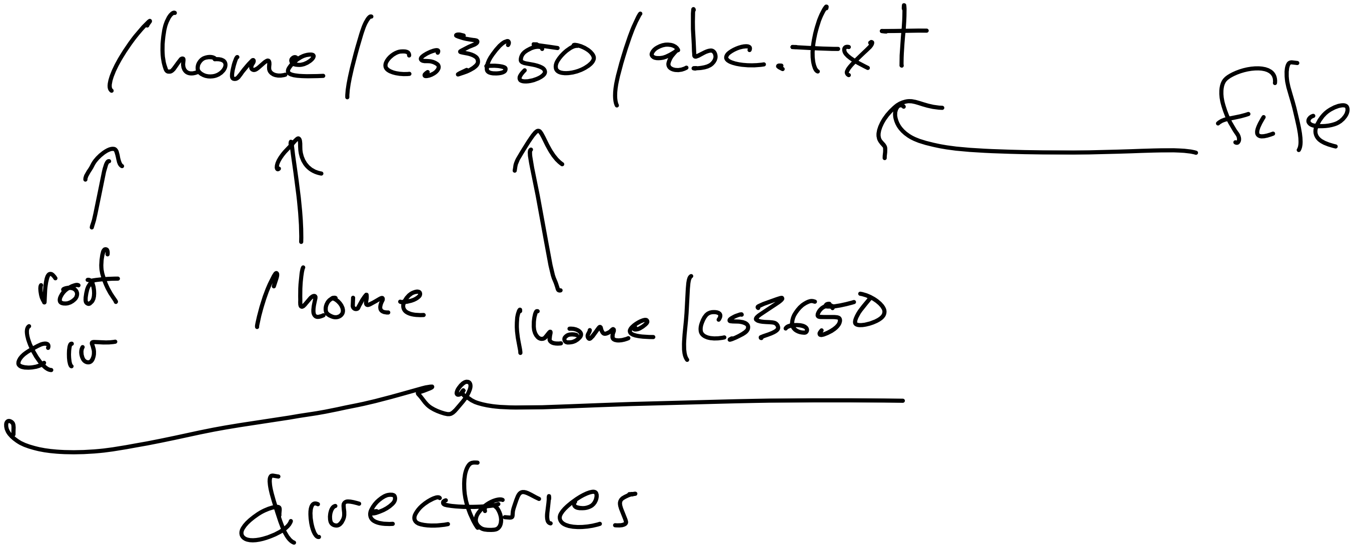
/dev/null

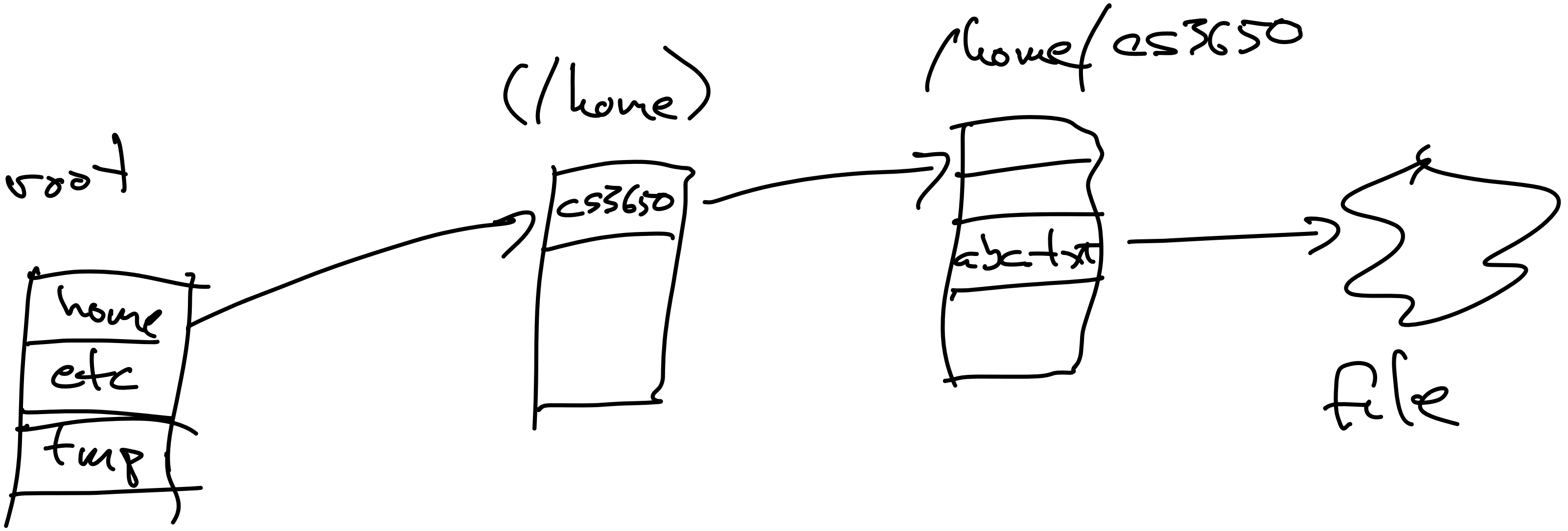
→ hierarchical file system
object = file | directory

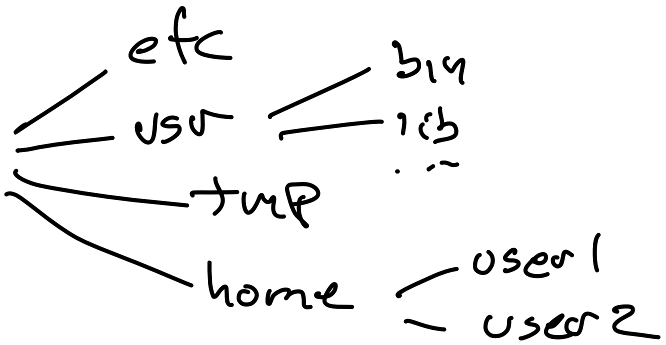
directory = map<name ⇒ object>



```
echo foo > abc.txt
```







typical Linux directory structure

- multiple file systems

disk
file
system

c:\Users\...

D:\USB-DRIVE1

E:\network-drive-stuff

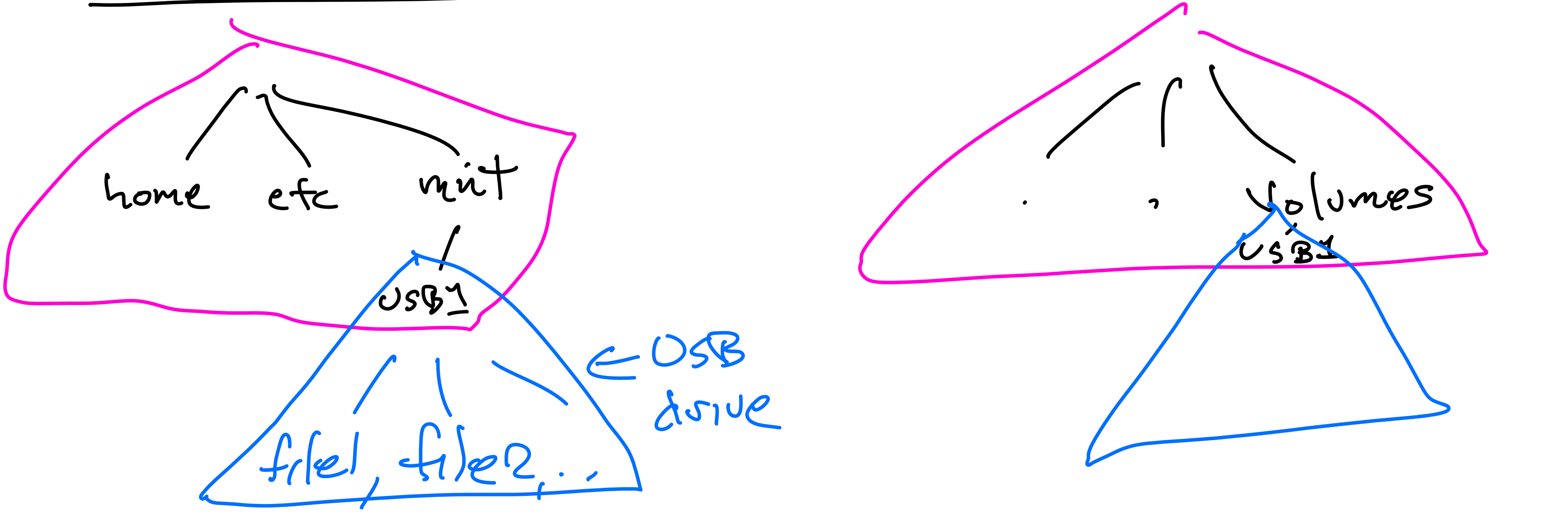
⋮

path relative to that file system

file system ("drive") identifier

NOT unix-like

Unix: file system mounting



really simple in-memory model of FS:

```
class obj:
    type : DIR | FILE
    dict dir    (name → obj)
    data        (bytes - content)
```

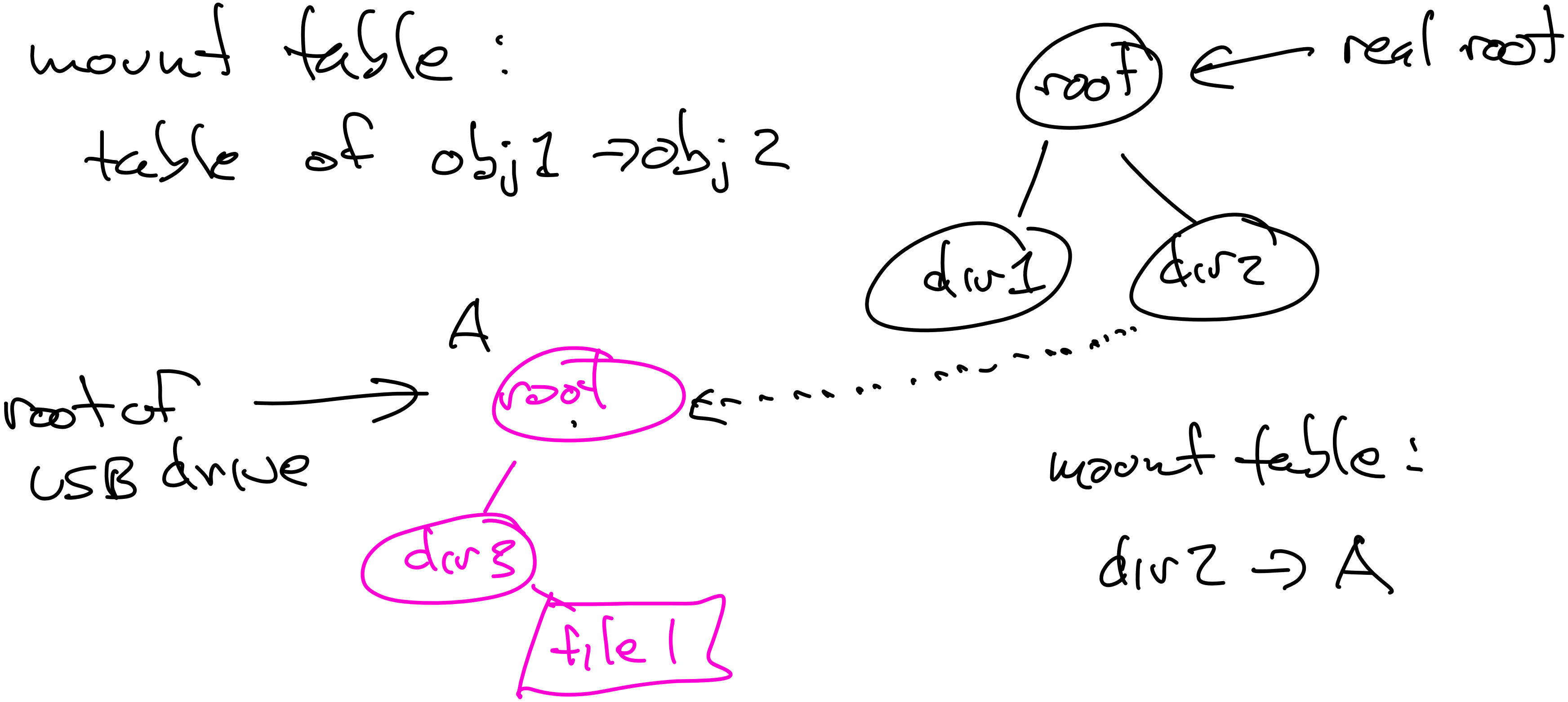
root = obj()

/a/b/c

'a' 'b' 'c'

mount table:

table of obj1 → obj2



mount table:

dir2 → A

File system operations:

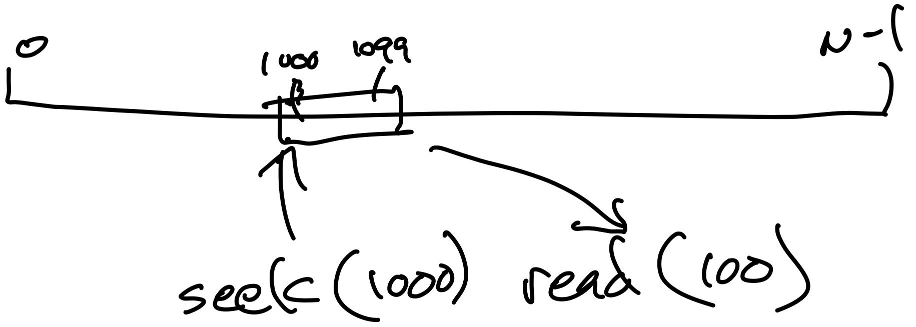
(create) (open w/ O_CREAT)

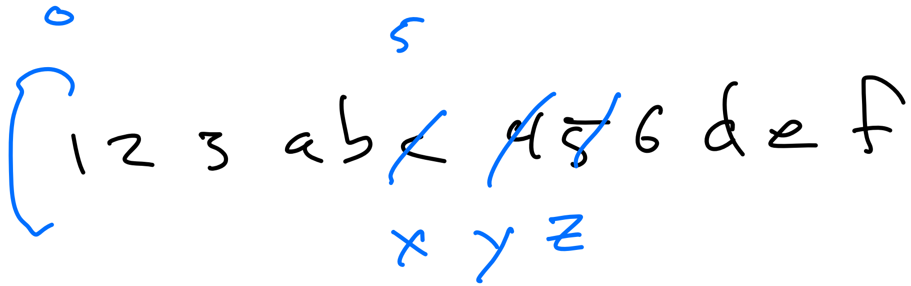
open / close

read / write / lseek

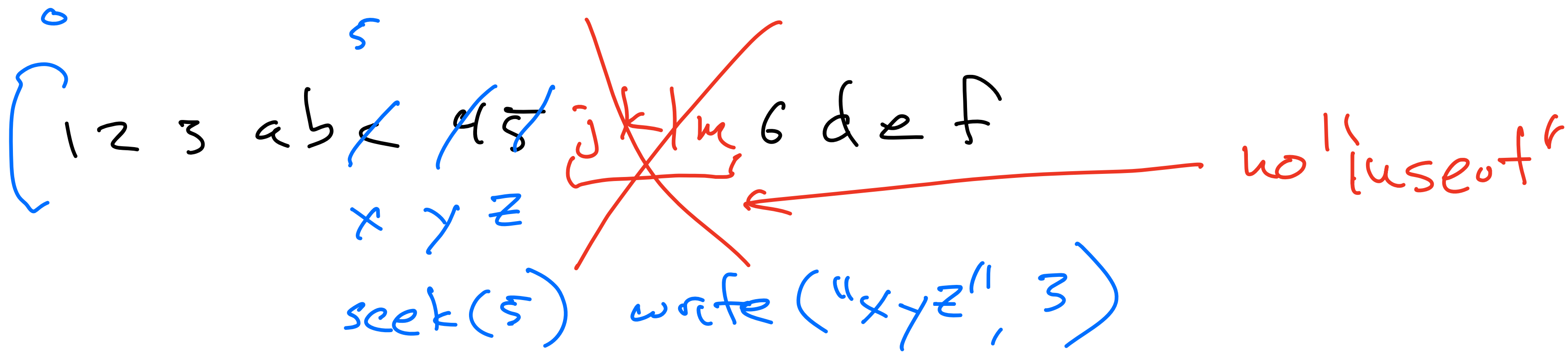
change current file offset

unlink





seek(5) write("xyz", 3)



a file is an array of bytes

(create) (open w/ O_CREAT)

open / close

read / write / lseek

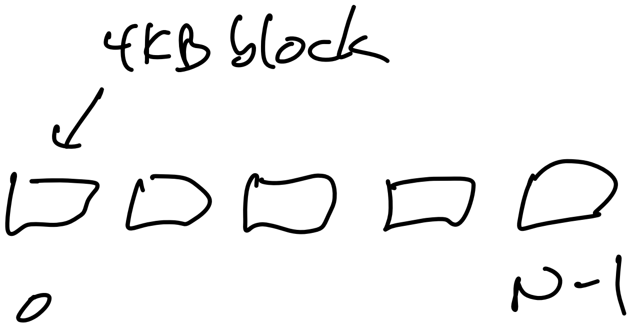
unlink

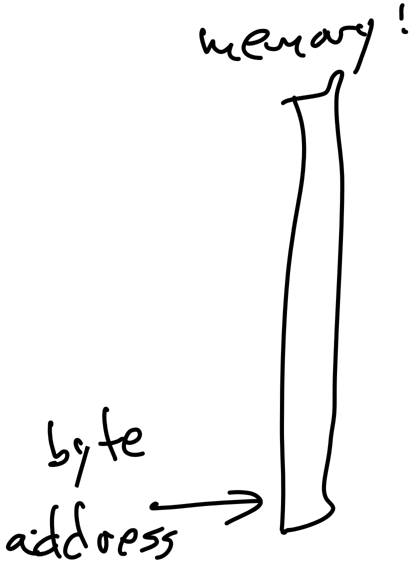
i.e. delete

syscall	CLI
mkdir	→ mkdir
rmdir	→ rmdir
opendir readdir closedir	→ ls

How do we implement a file system?

1) on a block device



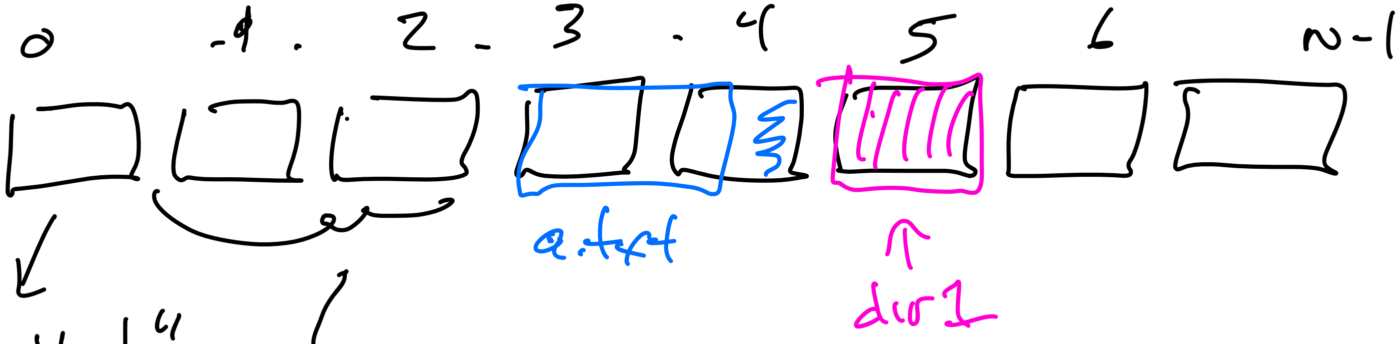


operations:

read block X (len = Y blocks)

write block X (.. len ≥ 1)

1) CD-ROM file system



"superblock"
- version = 1

root dir
start = 1
len = 2

root dir:

name	type	start	len
a.txt	F	3	5000
dir1	D	5	4K

array of

```
struct {
  char name[11]
  file/dir
  int start_block
  int len_in_bytes
```

len_in_blocks, too?

File system characteristics

HOSW
file sys chapter

1) how does it identify a file?

2) " organize blocks in a file?

3) ~~" it handle free space?~~

starting block
N

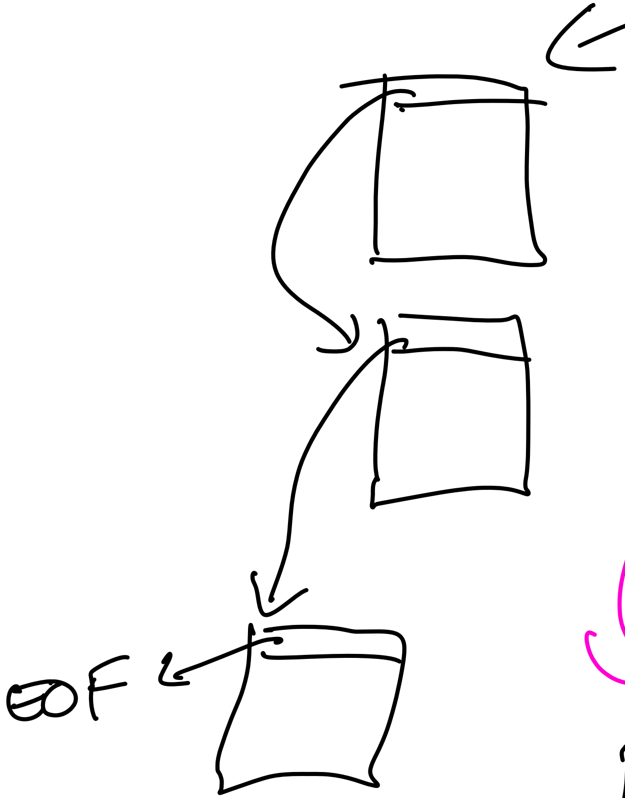
linear range

N, N+1, N+2 ...

common trick: dir. entry has "~~deleted~~" Flag
"valid"

MS-DOS File System ("FAT" - file allocation table)

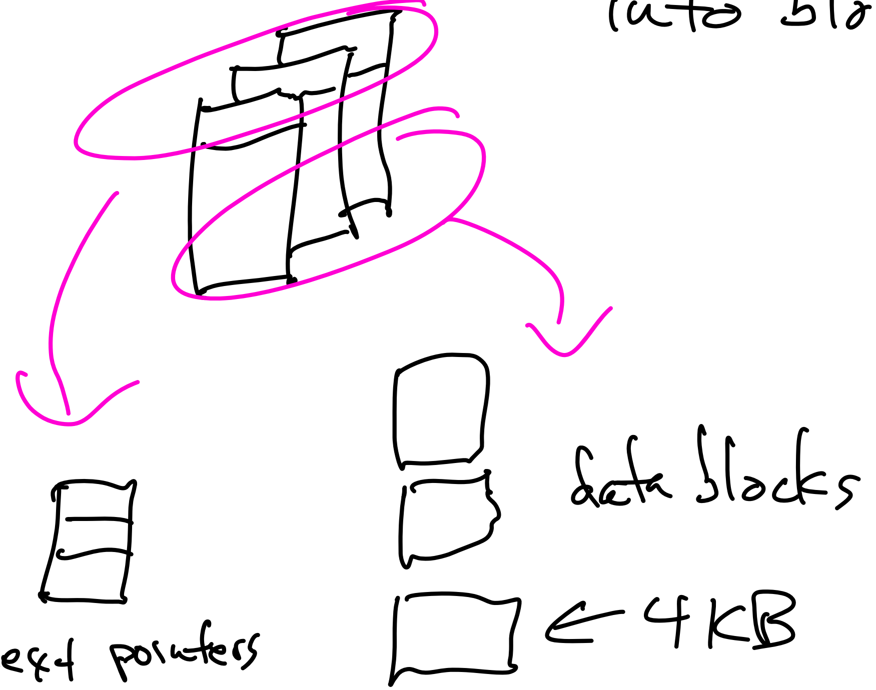
dir. entry: char name[11]
int start_block
int len_in_bytes

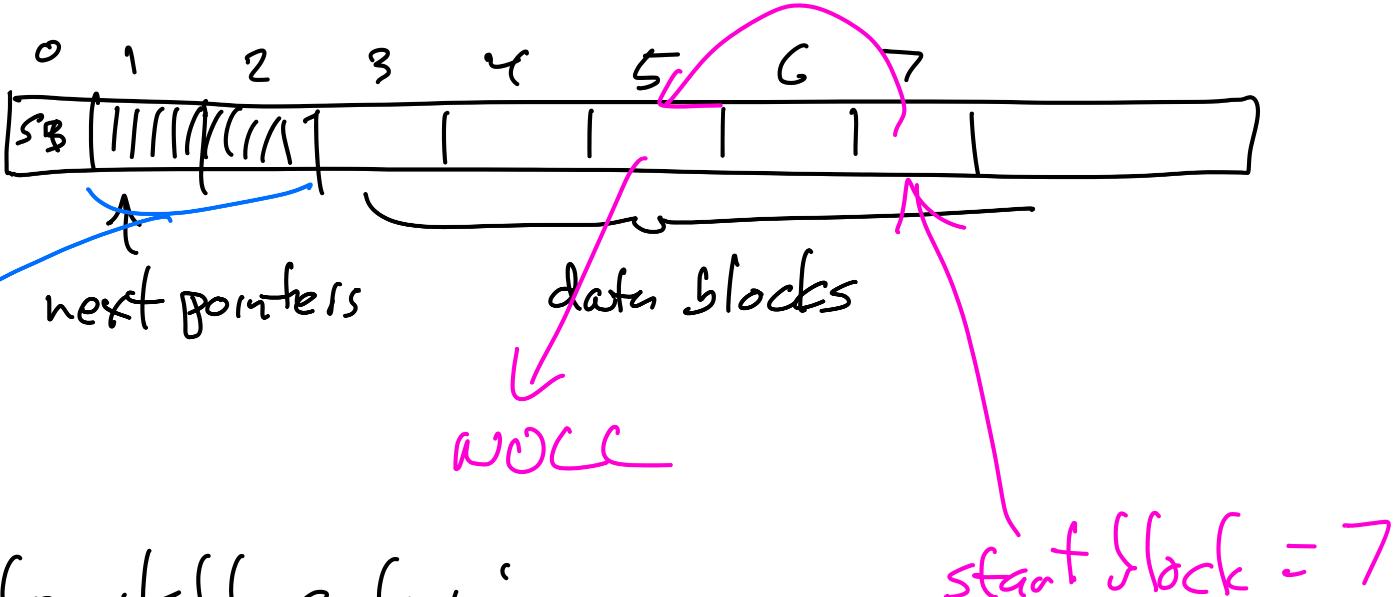


problem: how to fit pointer into block?

4KB

512B





pointer table entry:

- block #
- last block of file
- unused

1) link files

2) indicate free space

1) how does it identify a file?

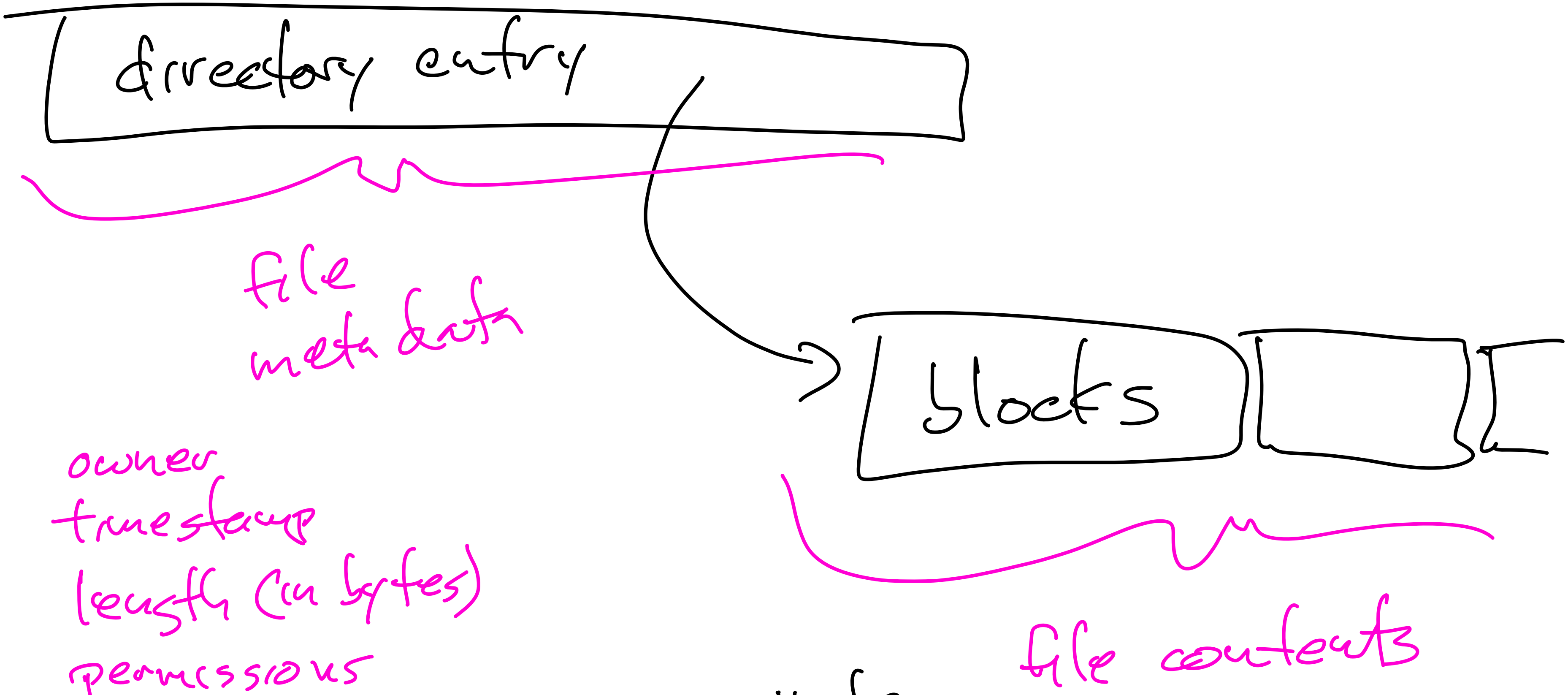
2) " organize blocks in a file?

3) " it handle free space?

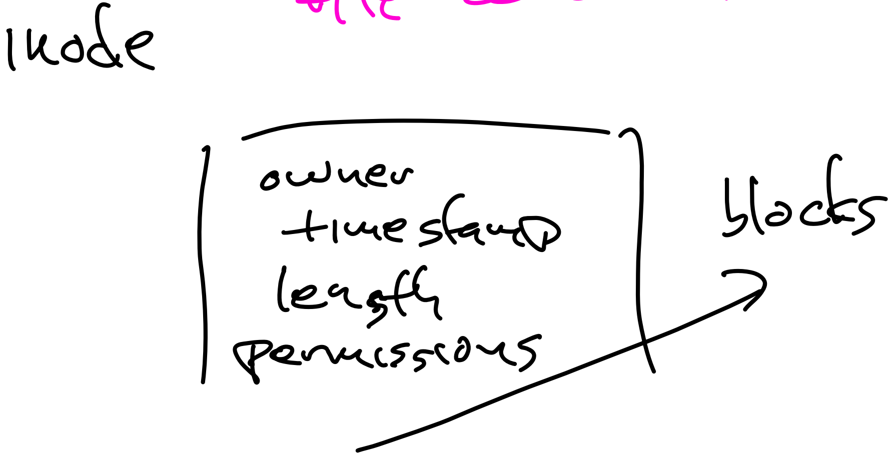
starting block #

linked list

combined w/ pointers in file alloc table



owner
timestamp
length (in bytes)
permissions



directory:

"name", <file#>

"name, < >

⋮