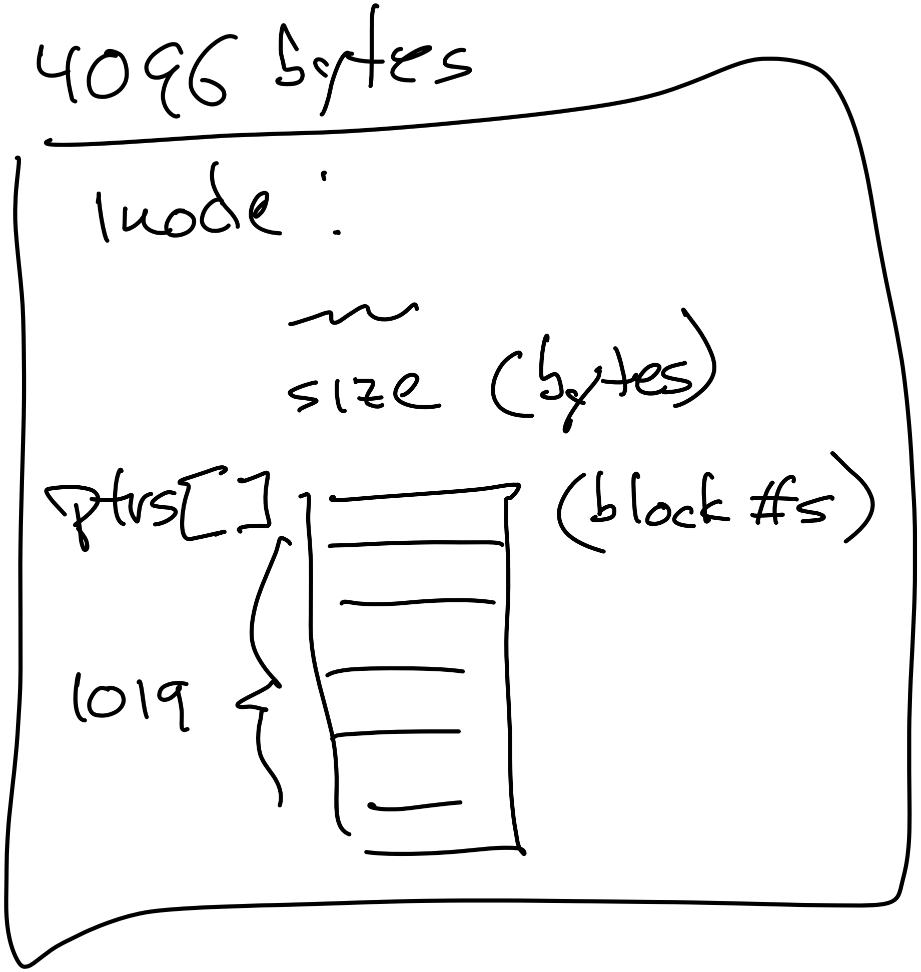
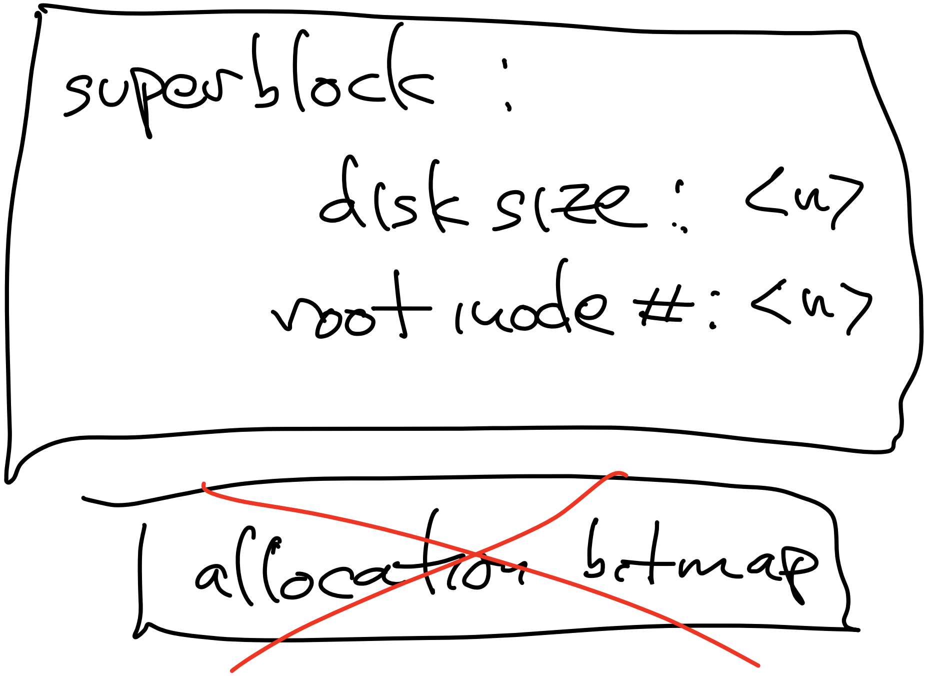
CS 3650 – Computer Systems
Spring 2024
Peter Desnoyers

Lecture 20, Thur Mar 21, 2024

Lab 4

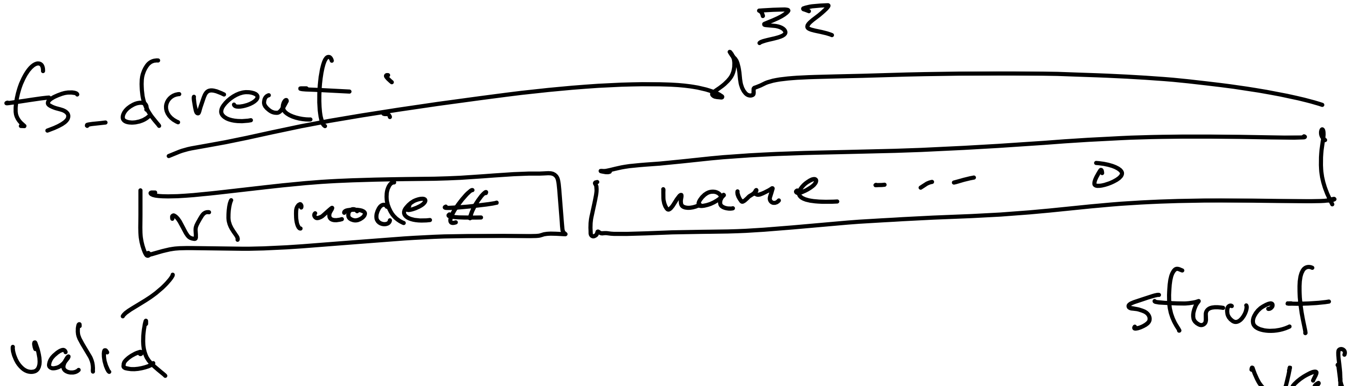


1019 · 4096 B

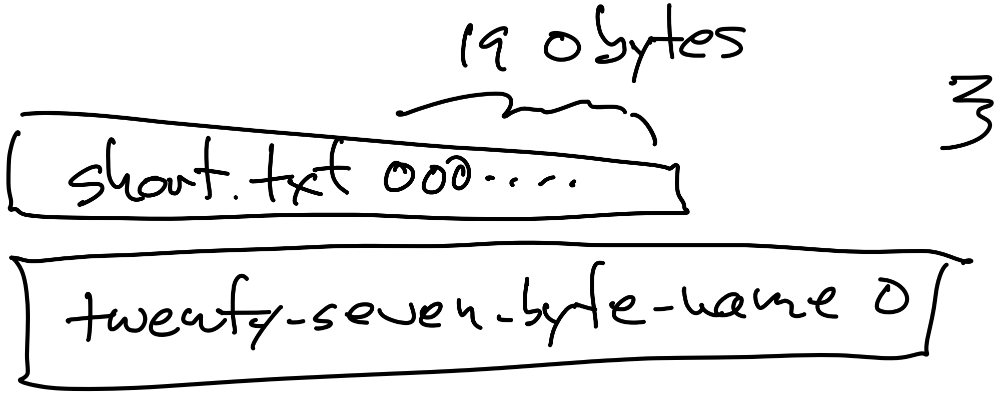


```
struct fs_inode in;
block_read(&in,
           block #,
           1)

in.ptrs[0] ←
```



```
struct {
  valid
  inode
  name[28]
}
```



```
struct fs_dirent d[128]
 block_read(&d, block#, 1)
if ( d[0].valid) printf("%s: %d",
                   d[0].name, d[0].inode)
```

FUSE readdir:

```
readdir( char *path,
         void *ptr,

int (*filler)(void *ptr,
              char *name,
              struct stat *st,
              ...)
```

```
for each name F in dir:
    ~~get attrs~~(F) → sb
    filler(ptr, F, &sb)
```

find file attributes
put them in sb

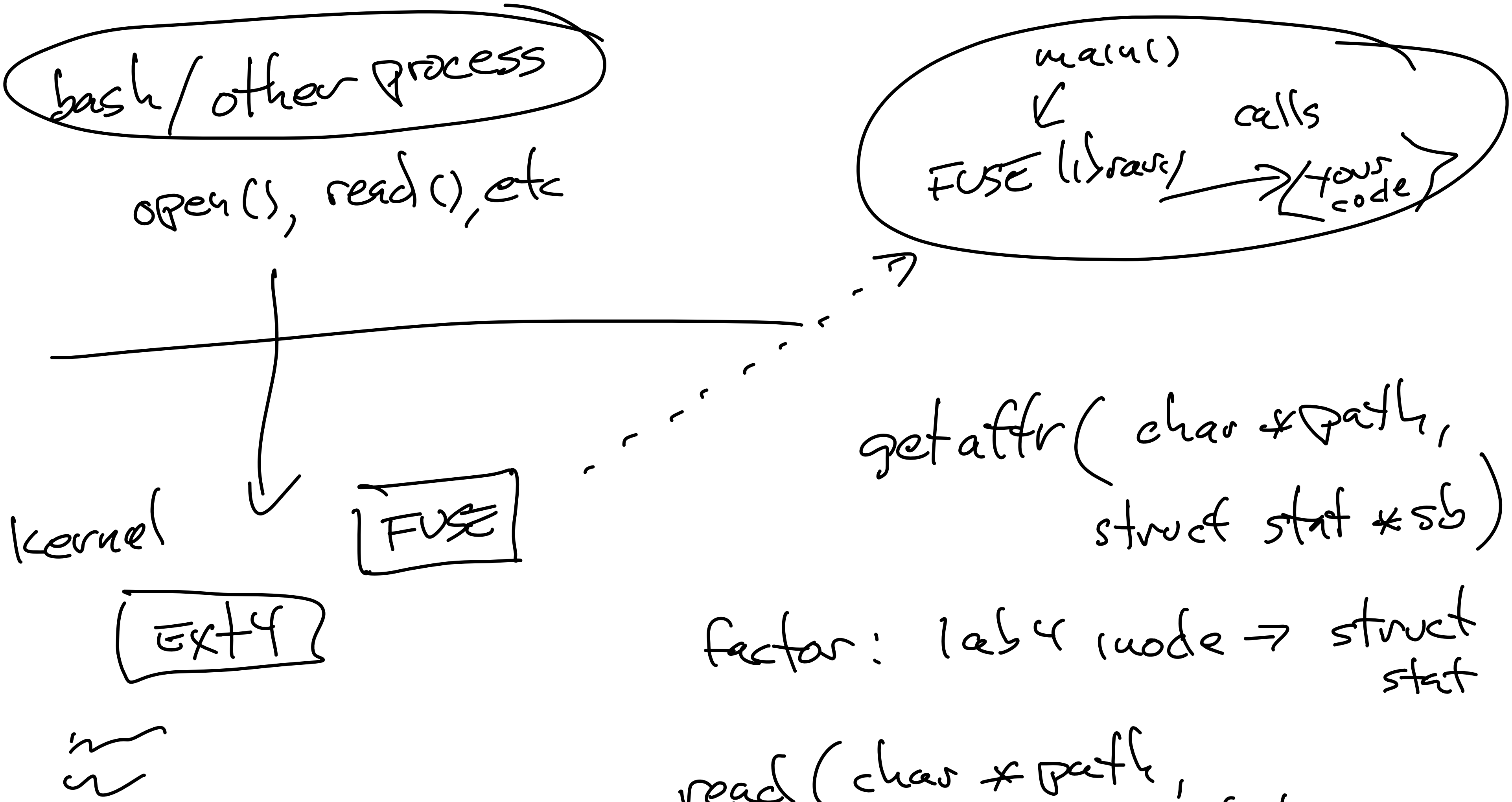
← struct stat

```
struct fuse_operations {
    int (*read)(char *path,
                char *buf,
                int len,
                int offset)
    int (*getattr)(...)
        ⋮
};
```

```
fs_ops.read(......)
```

maybe pass
NULL for
stat buf?
check...

How FUSE works



getattr(char *path,
struct stat *sb)

factor: lab4 inode → struct stat

read(char *path,
char *buf, int len,
int offset, ...)

Path translation

/home/pjd/file.txt

int pathlen

char * names[] → "home" → "pjd" → "file.txt"

path → inode #

```
inode = root_inode
for name in names
    read inode
    check ISDIR
    read DE block
    if DE[i].valid & name match
        inode = DE[i].inode
        ~~con~~ next name
```

inode #

```
inode = 1
_in = readblk 1
check ISDIR
read block
, name = "home"
inode = ...inode
```

DO NOT USE strcpy

memcpy ← copy bytes

(void * dst, void * src, int #bytes)

forget about EOF

syscalls: fd = open

val = read (fd, buf, len)

if val == 0

→ we're at end of file

read (offset, len)

→ offset ≥ file len: return 0 bytes

stdio : FILE*

int c = getc(fp)

if(c == EOF)

0...255 ←

or -1

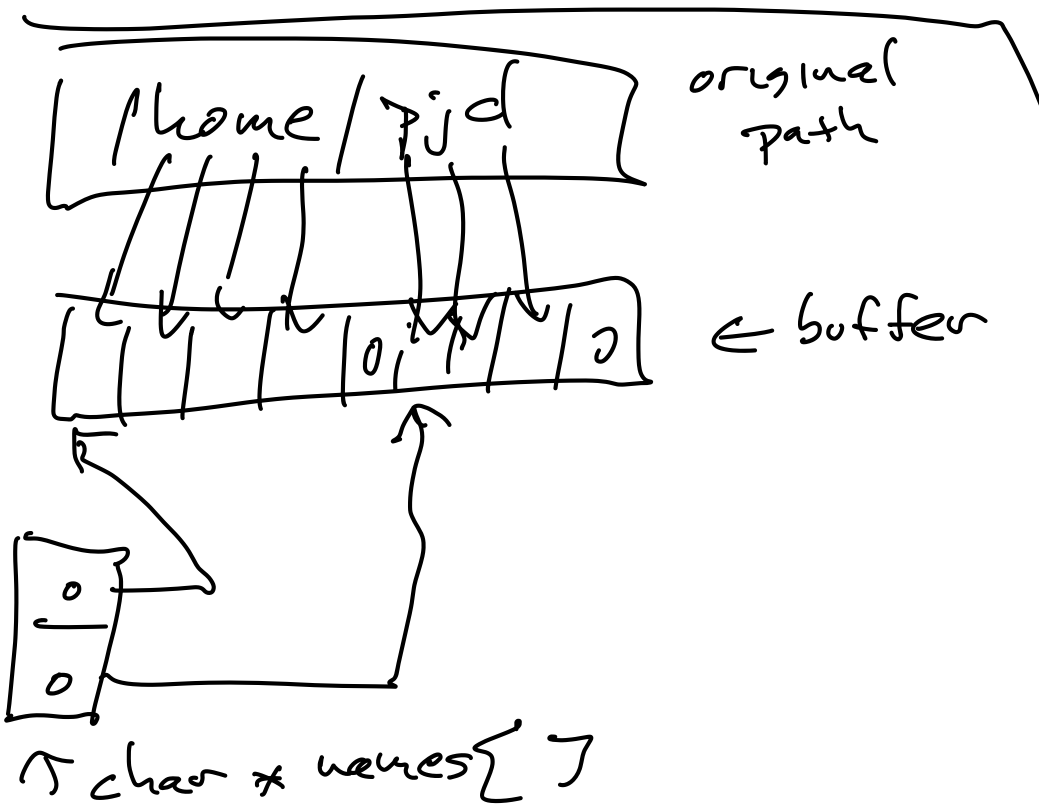
return values :

< 0 = error

≥ 0 = success

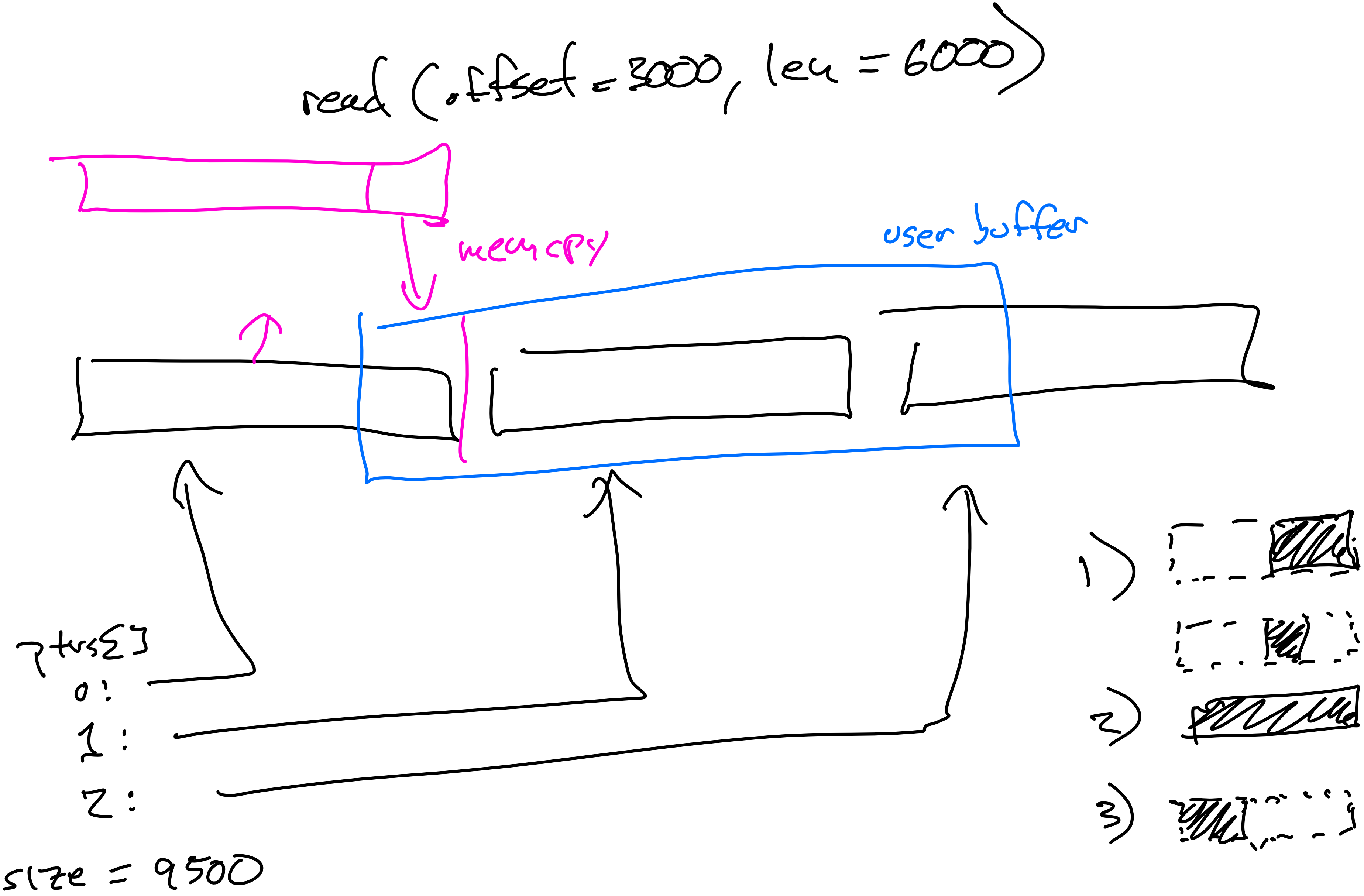
read : ≥ 0 : bytes read

```
if (not found)
    return -ENOENT;
```



```
getattr ( )
    split
    int i = translate (...)
    if i<0
        return i
```

read


read (offset = 3000, len = 6000)
memcpy
user buffer
ptrs[]
0:
1:
2:
size = 9500
1)
2)
3)

More file & related stuff

journalling (write-ahead log)

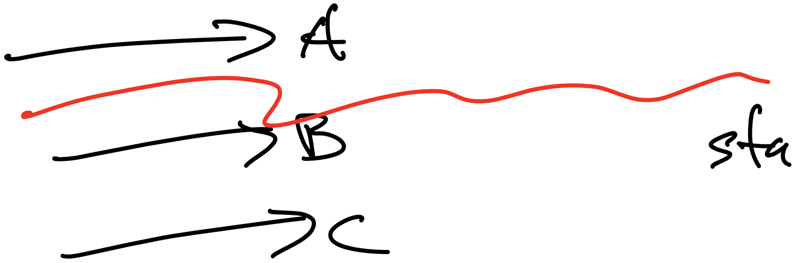
→ crash consistency

similar (sort of) to multi-thread

→ will do: A, B, C

checksum timestamp?

Idempotent



startup: read log

do A

do B

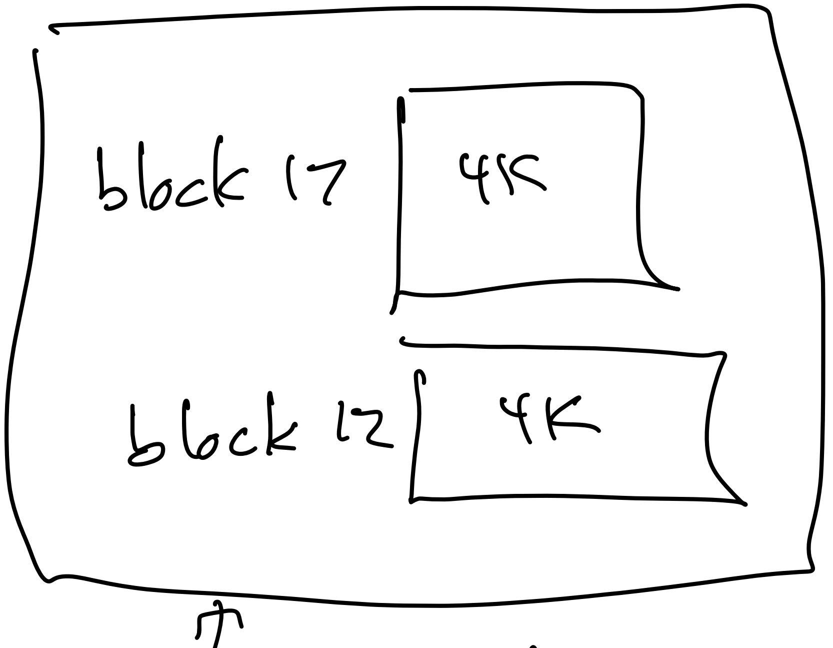
do C

Idempotent operation:

$$\left.\begin{array}{l}\text{1 time}\\[2em] N>1\text{ times}\end{array}\right\}\text{ same}$$

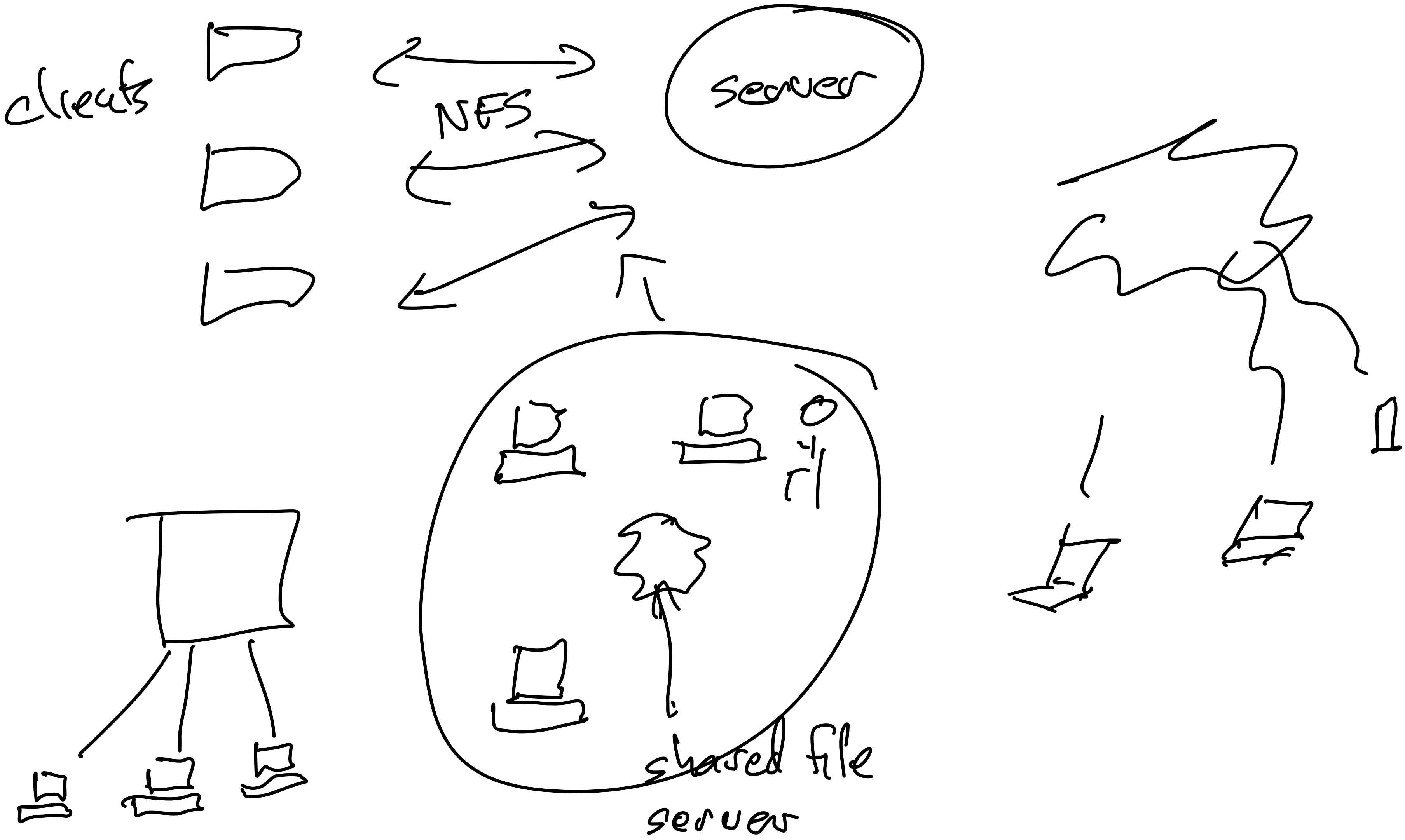
$x=1$ ✓

$x=x+1$ ✗



journal record

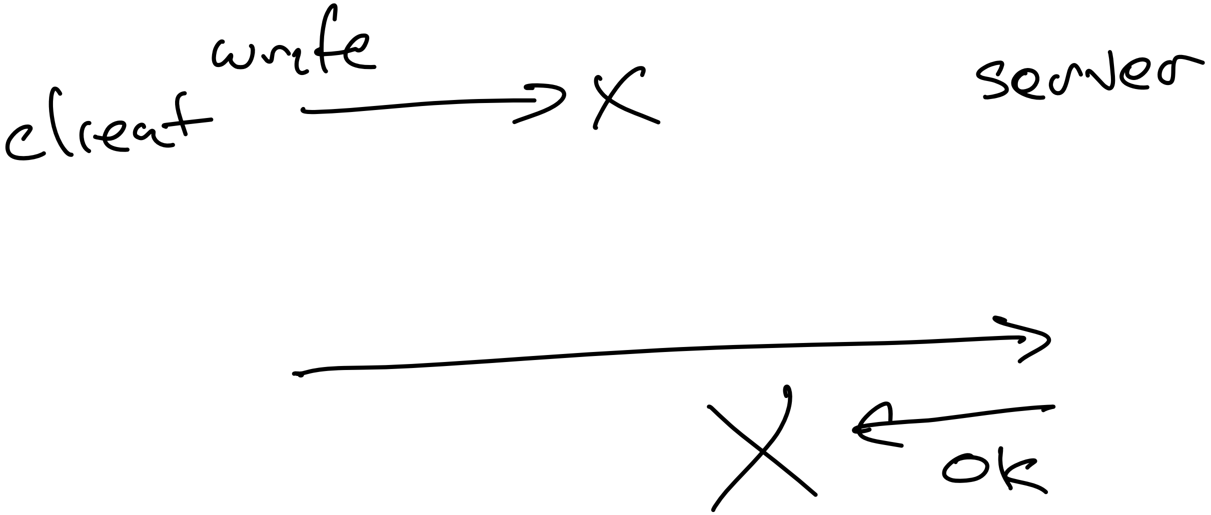
Network File System (NFS)



NFS network failure handling

idempotent

- READ (fileID, offset, len) →
- WRITE(fileID, offset, len, data) →



Not file systems

S3

key

PUT objname
<data>

GET objname

+ byte range request

DELETE

can't :

→ modify object

→ rename directories

key

"/dir/dir2/file"

"/dir/dir2/file2"

rename dir → xdir