

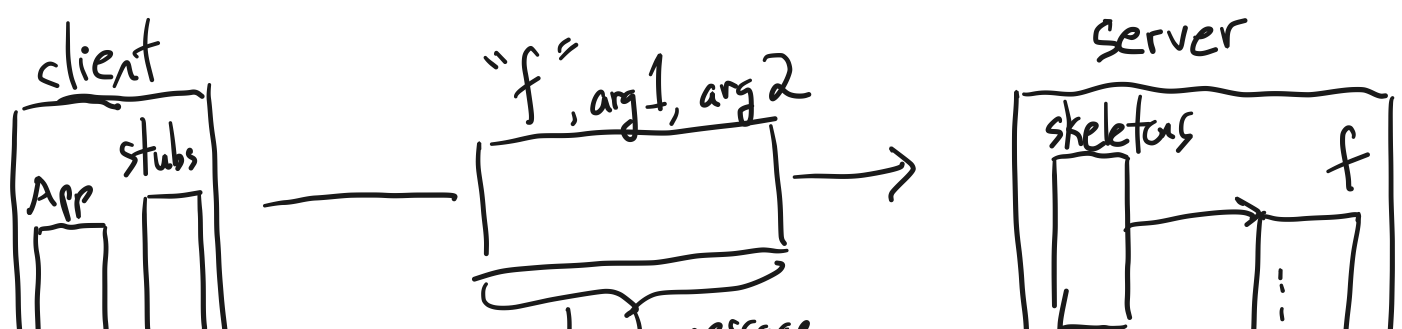
- 1. Last time
  - 2. Crash recovery, contd.
  - 3. RPC, client/server systems
  - 4. NFS
    - Intro + background
    - How it works
    - lab5 detour
    - Statelessness of protocol
    - Transparency
- 

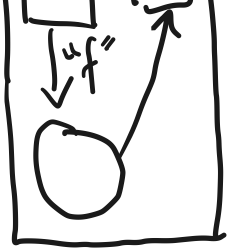
## 2. Crash recovery

See last time's scribble

---

## 3. RPC, client server





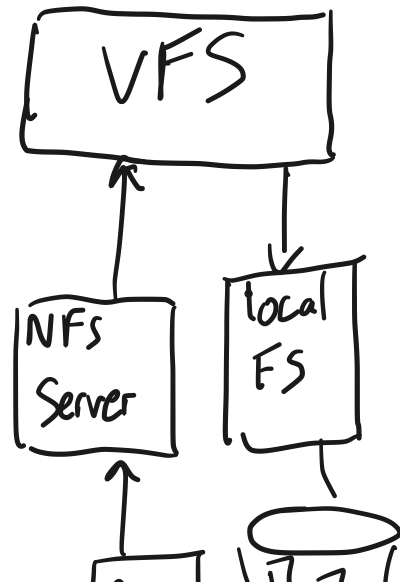
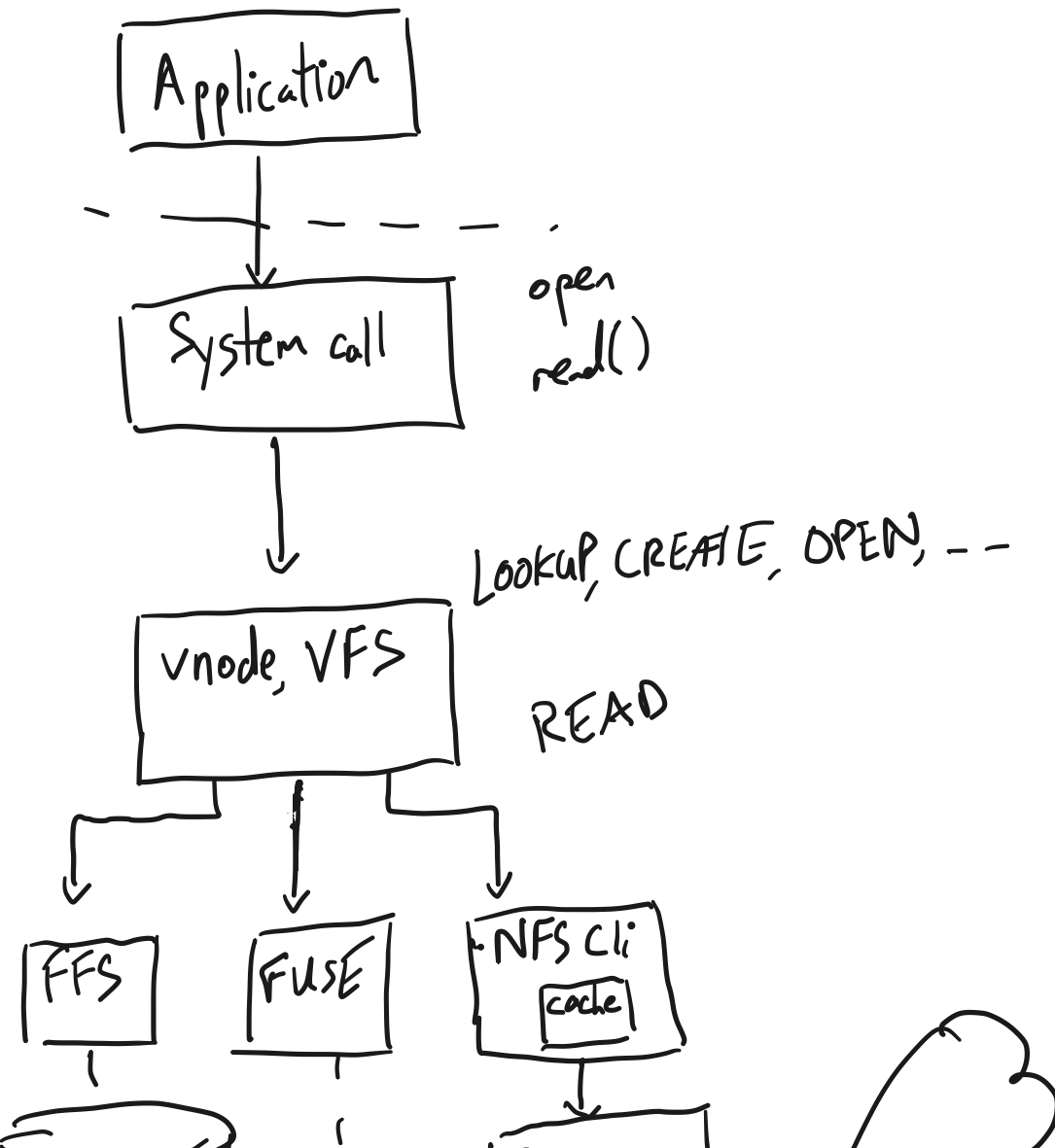
network message

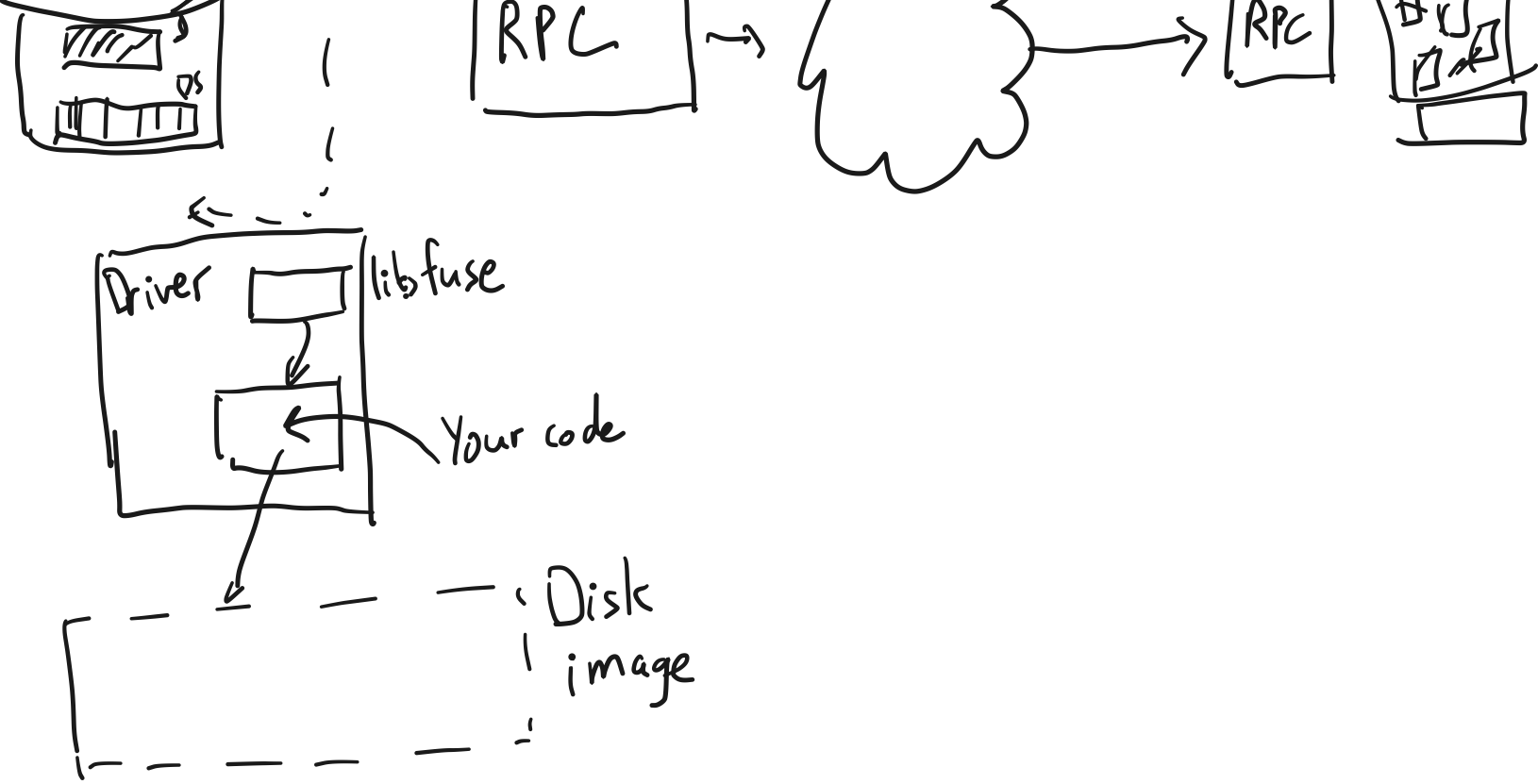


## 4. NFS

Client

Server





lab notes:

→ each inode gets its own disk block

→ pointers to pointers —

Client

XDR

time  
↓

Server

open("/usr/jo/lab1.c", --) ... LOOKUP("/usr/")  
open("/usr/jo/lab2.c")  
→ fh1 = [FS id | i# | gen#]

LOOKUP(fh1, "j0") →

← fh2

LOOKUP(fh2, "lab1.c")

← fh3 = [FS id | i# | gen#]

write(fd, buf, sz);

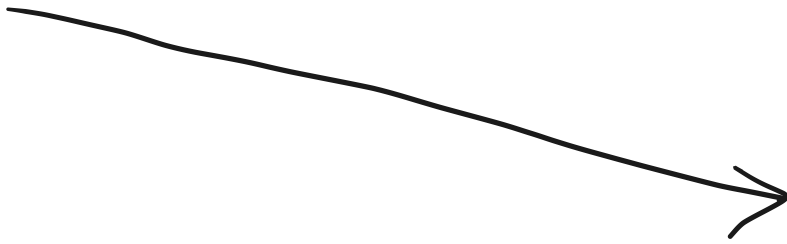
WRITE(fh3\_offset, data, size)

← return code

cli

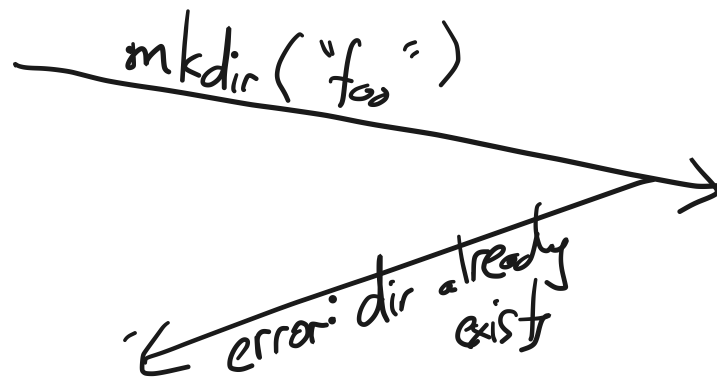
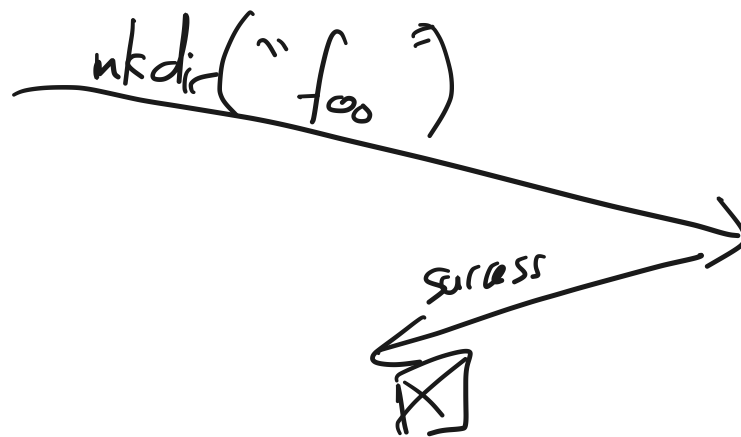
Serv.





write: idempotent

mkdir()



---

④ Transparency (or lack thereof)

- generation #      recall: FH = [FS ID | i# | gen #]

gen: 41

client A  write a file

client B  deletes ~~creates~~ a file

client C  creates a file <sup>gen:42</sup>

client A  write a file <sup>gen:41</sup> X  $41 < 42$

close()