

NFS

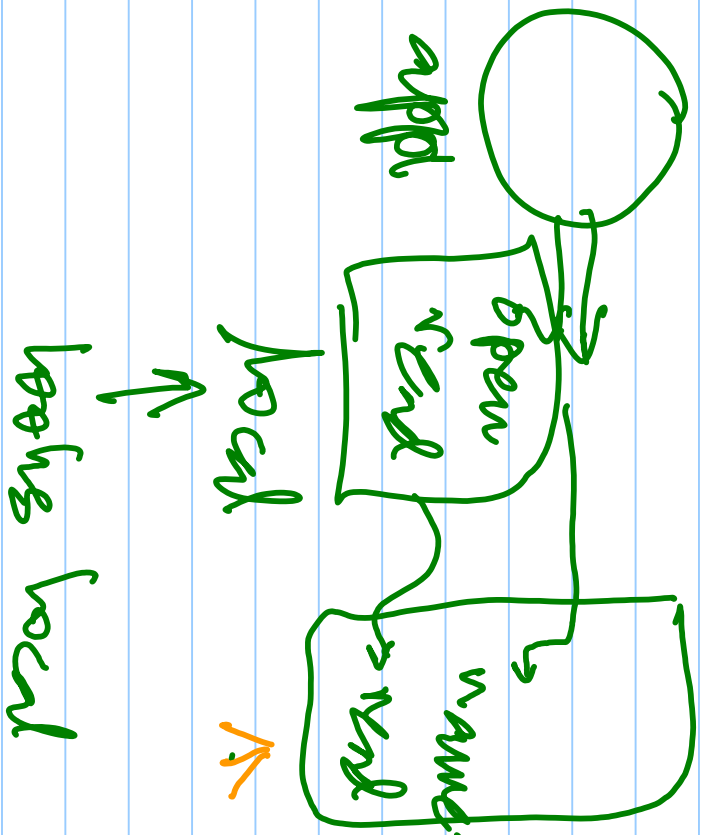
?

in kernel

kernel

remote → nfsd

local / not local



local

-

Single system image

↳ a cluster of computers

looks like a single computer

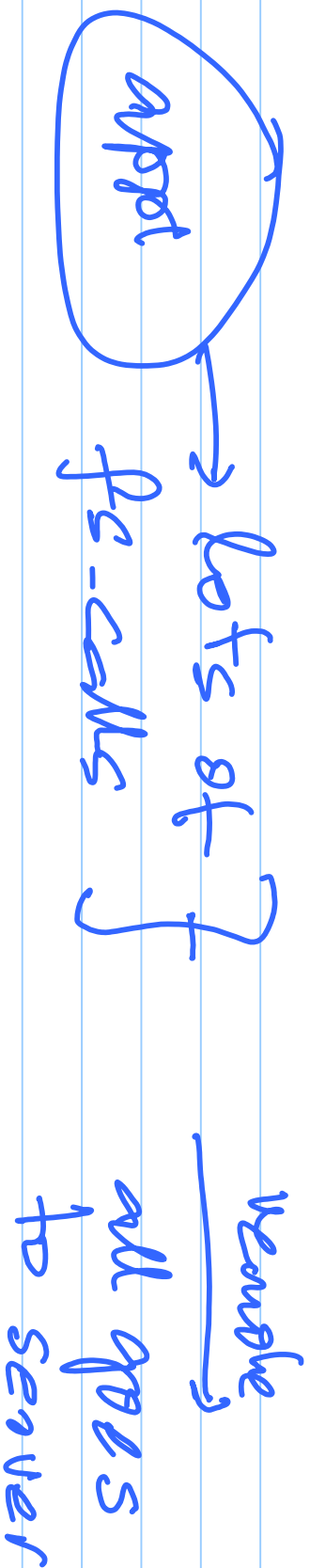
↳ done by an OS (FDS)

↳ file system

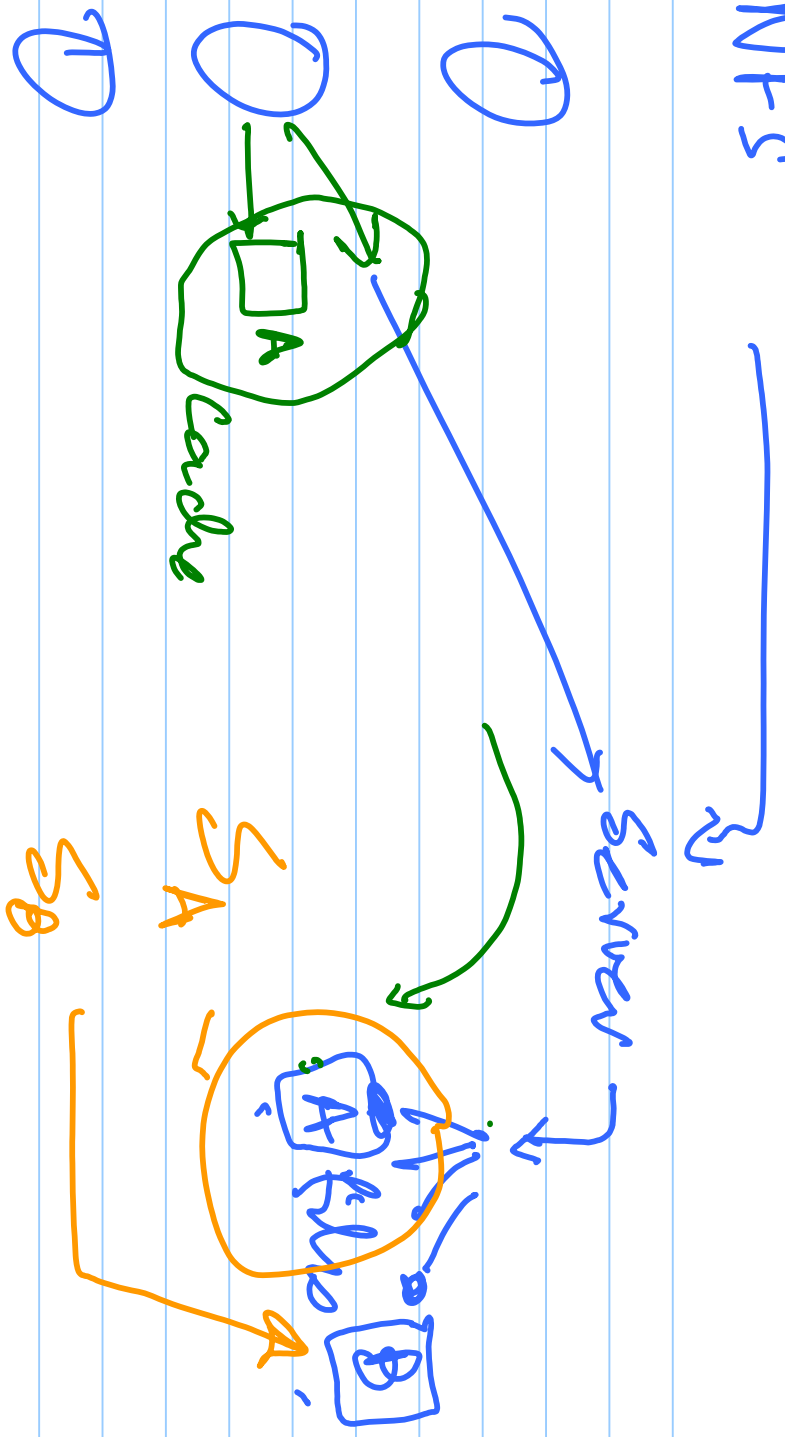
+++.

NFS

- stateless server → HTML
- Server failure tolerance
- band scalability

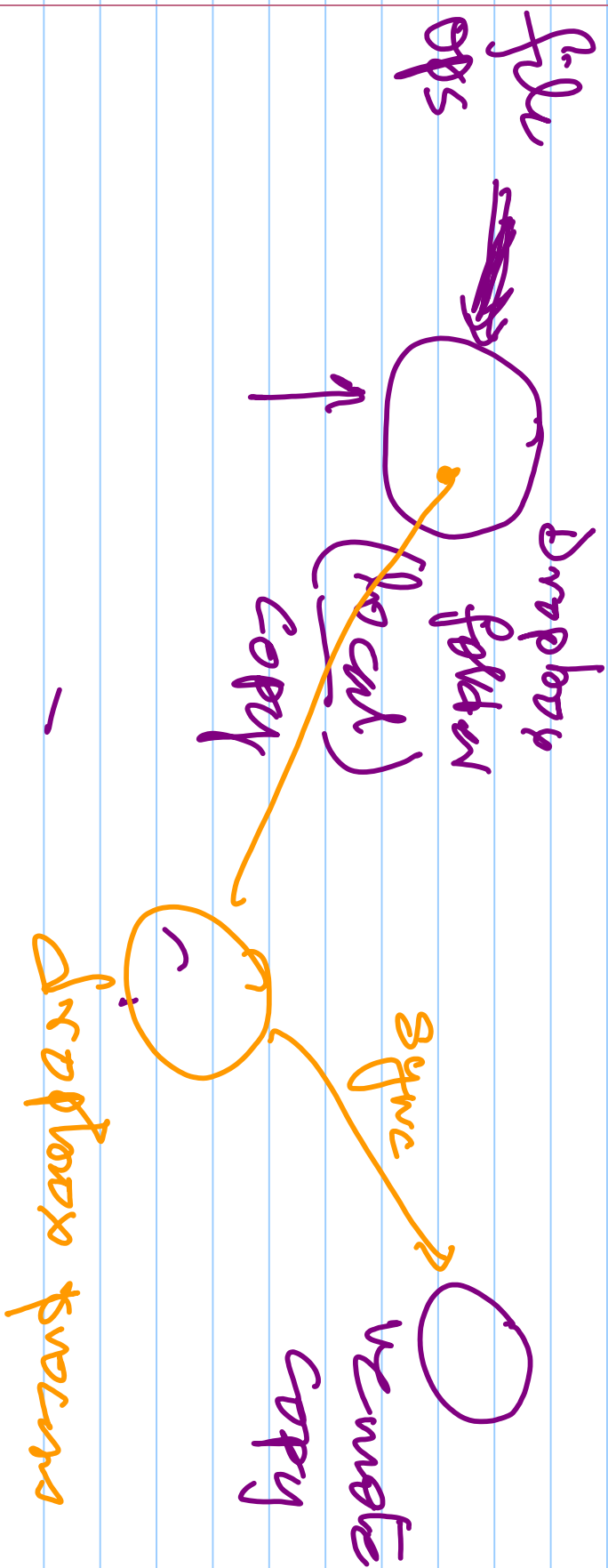
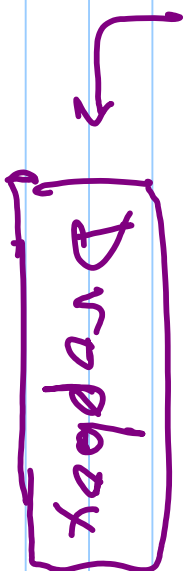


NFS



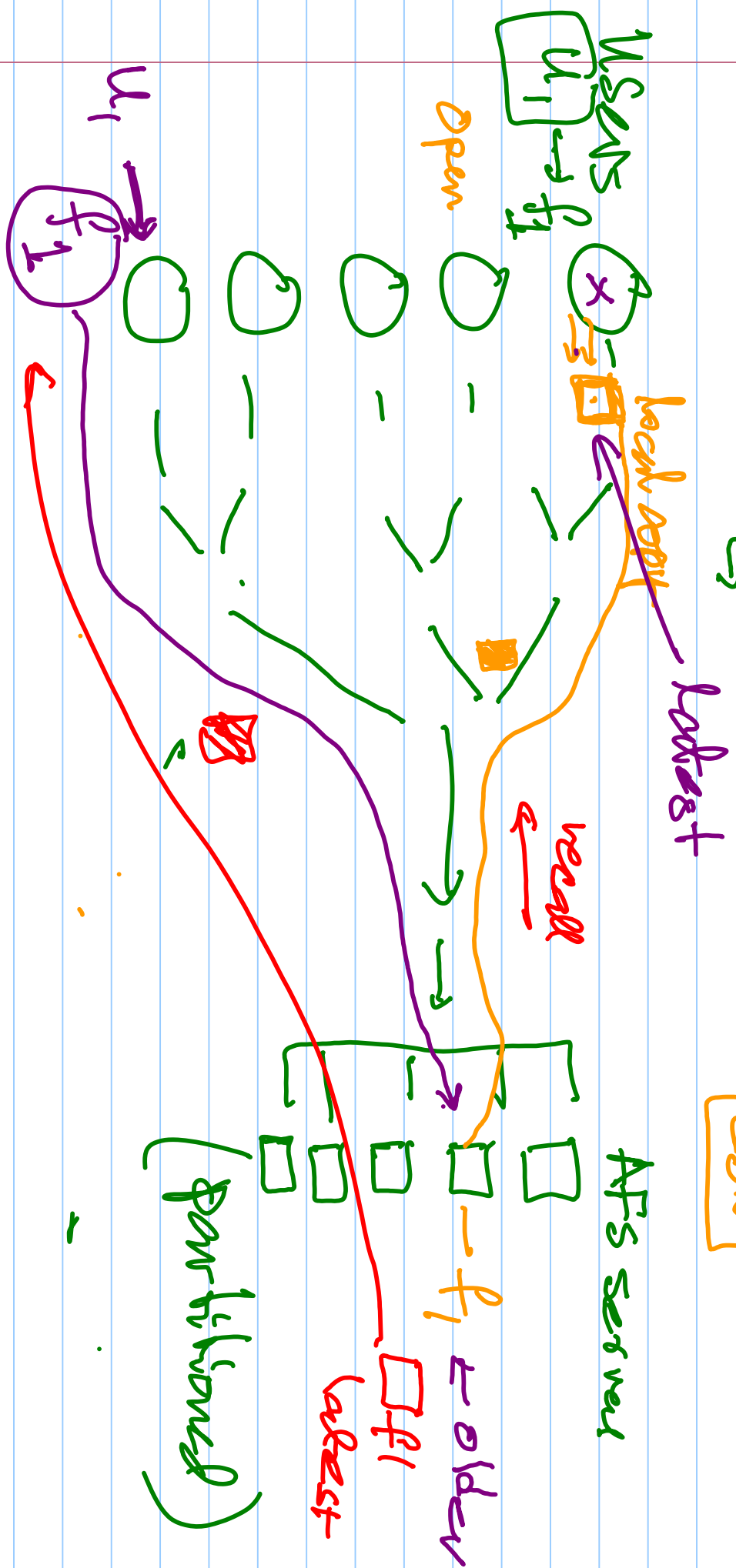
- ① Client caching
- ② Partition → multiple servers

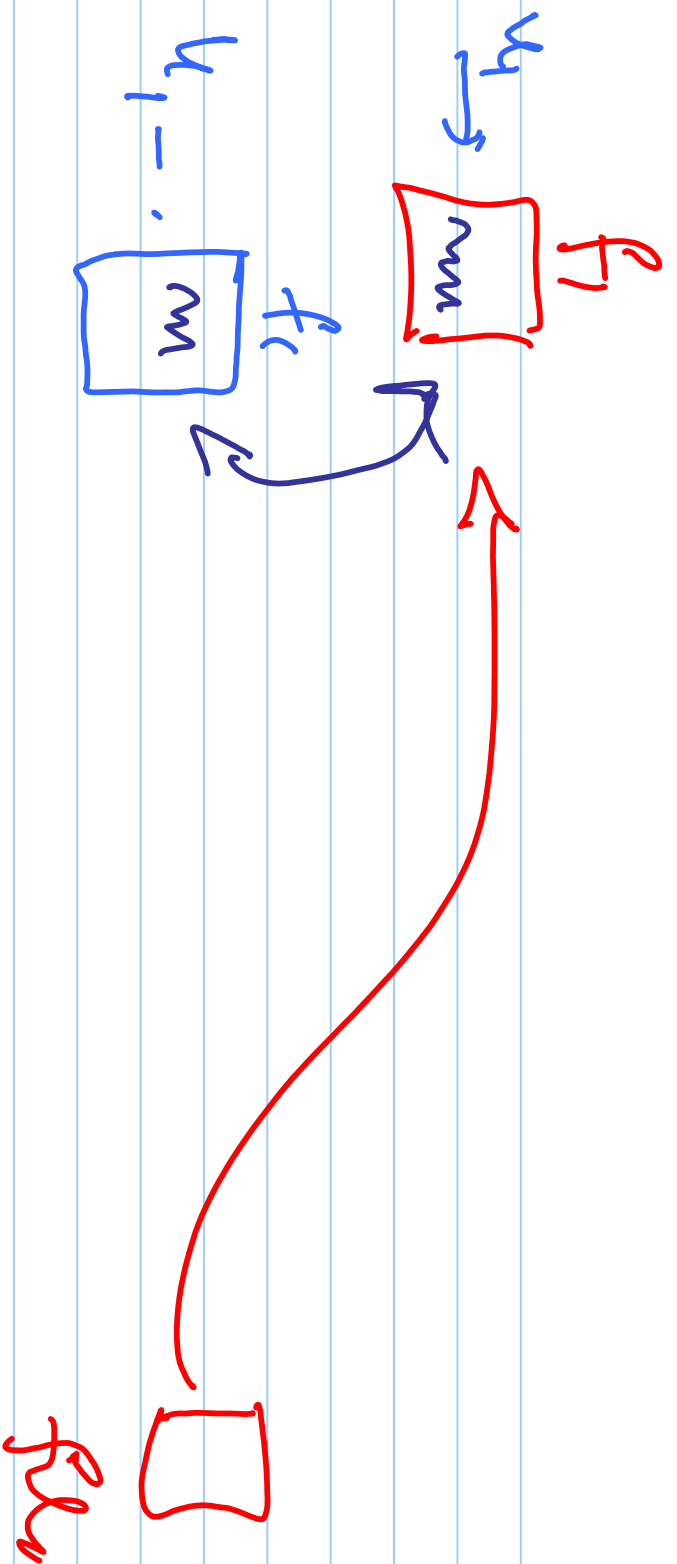
NFS → AFS



AFS → Andrew File System

CDN





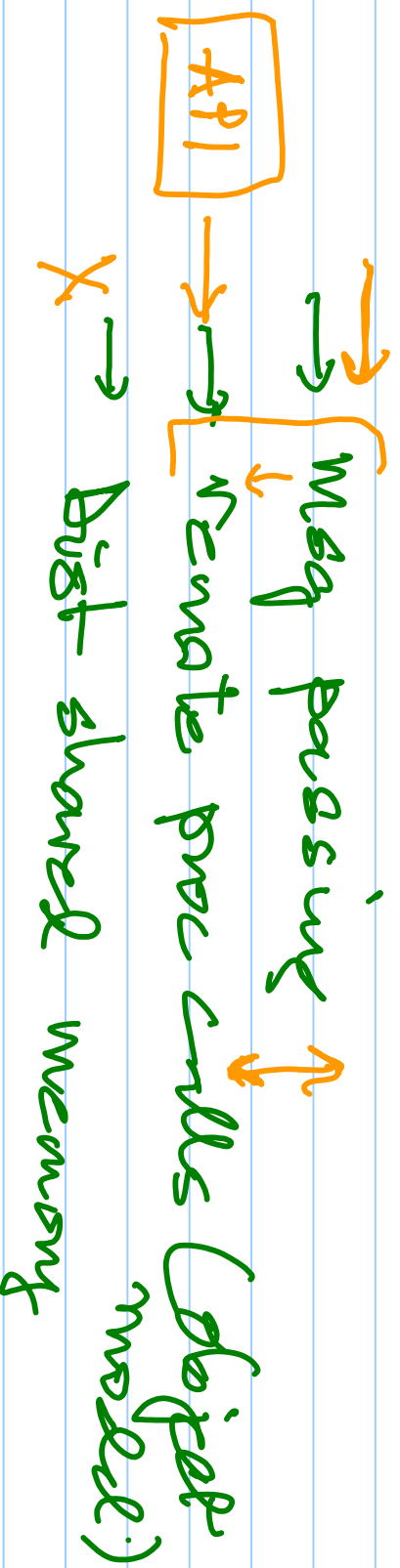
$u_1 - m$ — not n needed

- # NFS
 - single system image
 - coherent
 - Server load is too high
 - Scalable

Distributed OS \rightarrow built on

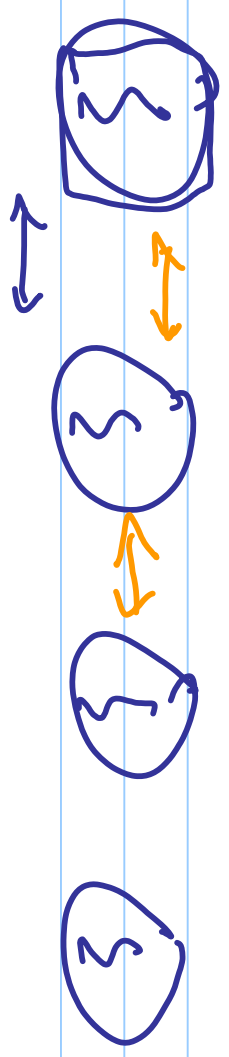
a distributed system model

\rightarrow a programming method that ~~is~~
works on 1 machine or n
machines \rightarrow



Message passing

Process

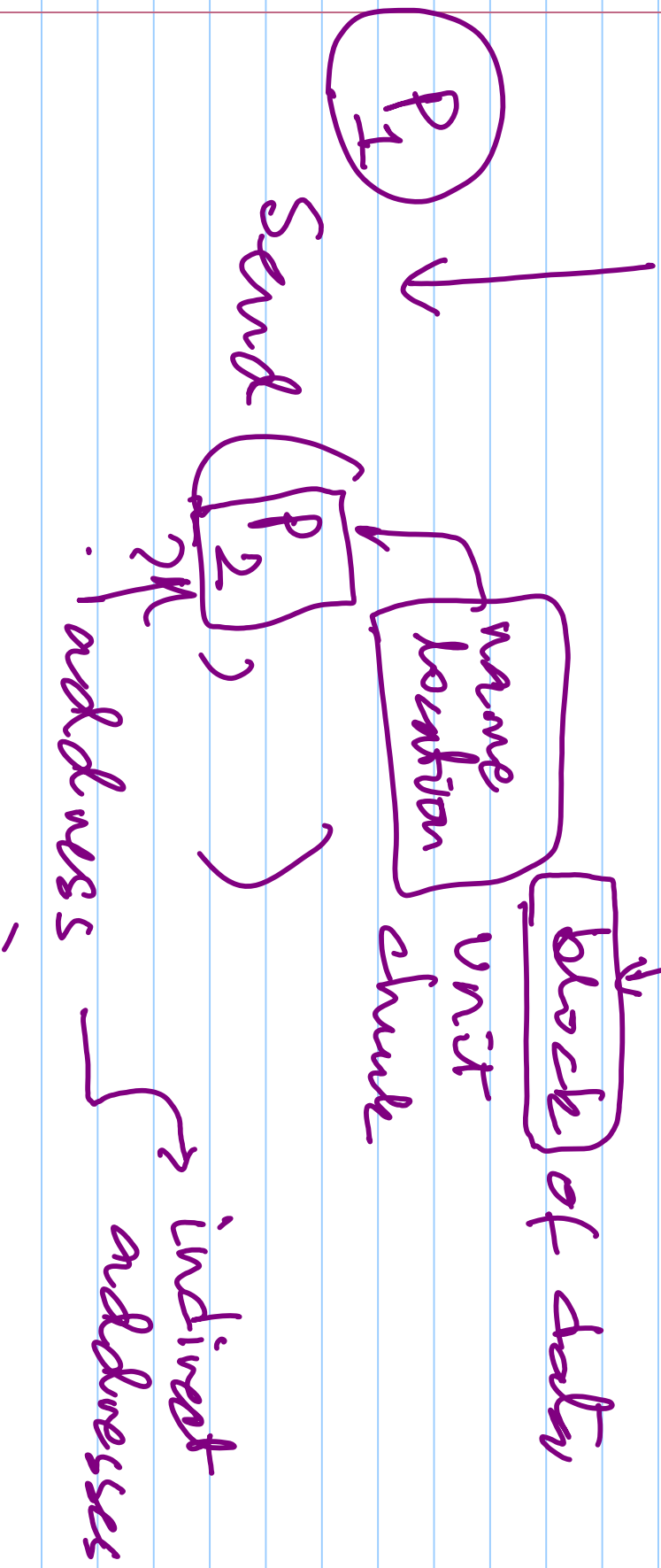
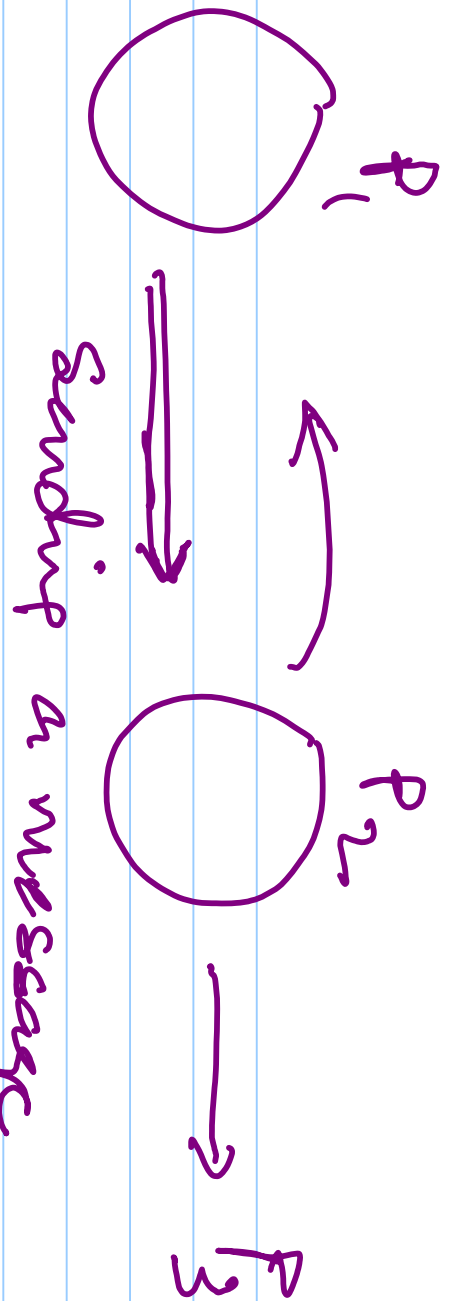


Communicate

shared memory

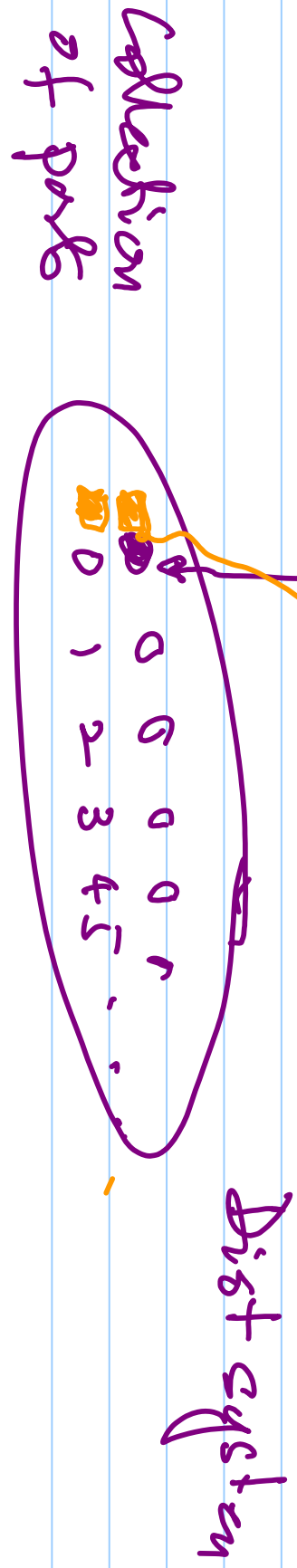
→ message

No shared memory



Process send msg
to ports

$P_1 \rightarrow \text{send} (\text{"ports"}, \text{msg})$
(array)



port → has an id



member
address

→ has a sequence of msgs
stored in it

processes can send → port

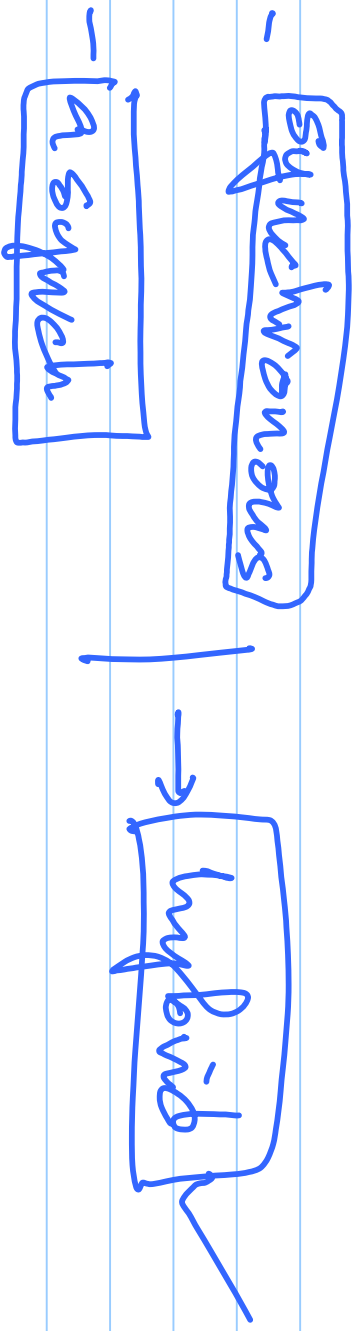
”

”

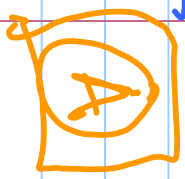
rec from a port

P_2 rev (port, [msg])
any port id → output parameter

Semantics of send & rev



→ Send (port, msg)



non blocking → puts msg in port & returns

(easy)

(B)

blocking → if puts msg in port & returns after P₂ recvs the msg
(sync)

→ A, but if port is full, return error
→ A " " " " " " block
→ A " " " " " " block + timeout

Send
B → block till read (sync)
→ block till read or
timeout
rcv → Do not block (async)

async → nothing holds ← BAD
Send puts in path (or not)
good
rcv checks & gets if available
Sync → Send/rcv both block till
msg is read

Send does not block

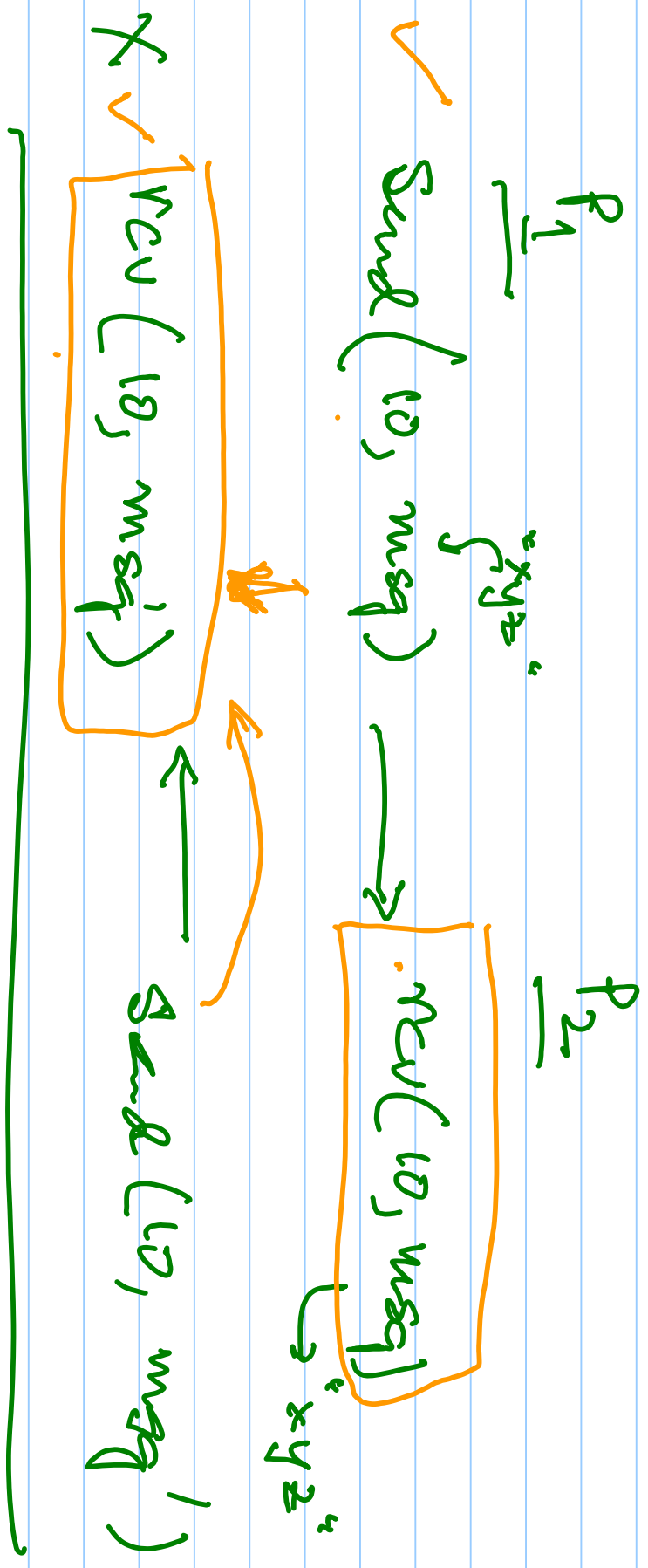
- if port has space

- blocks if port full

recv → blocks if port is

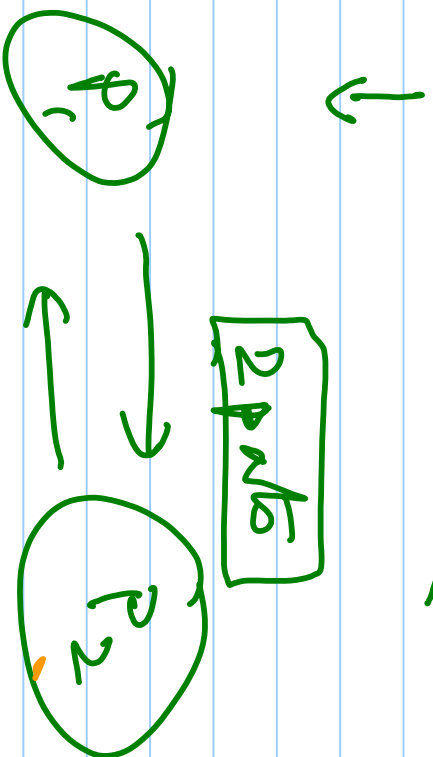
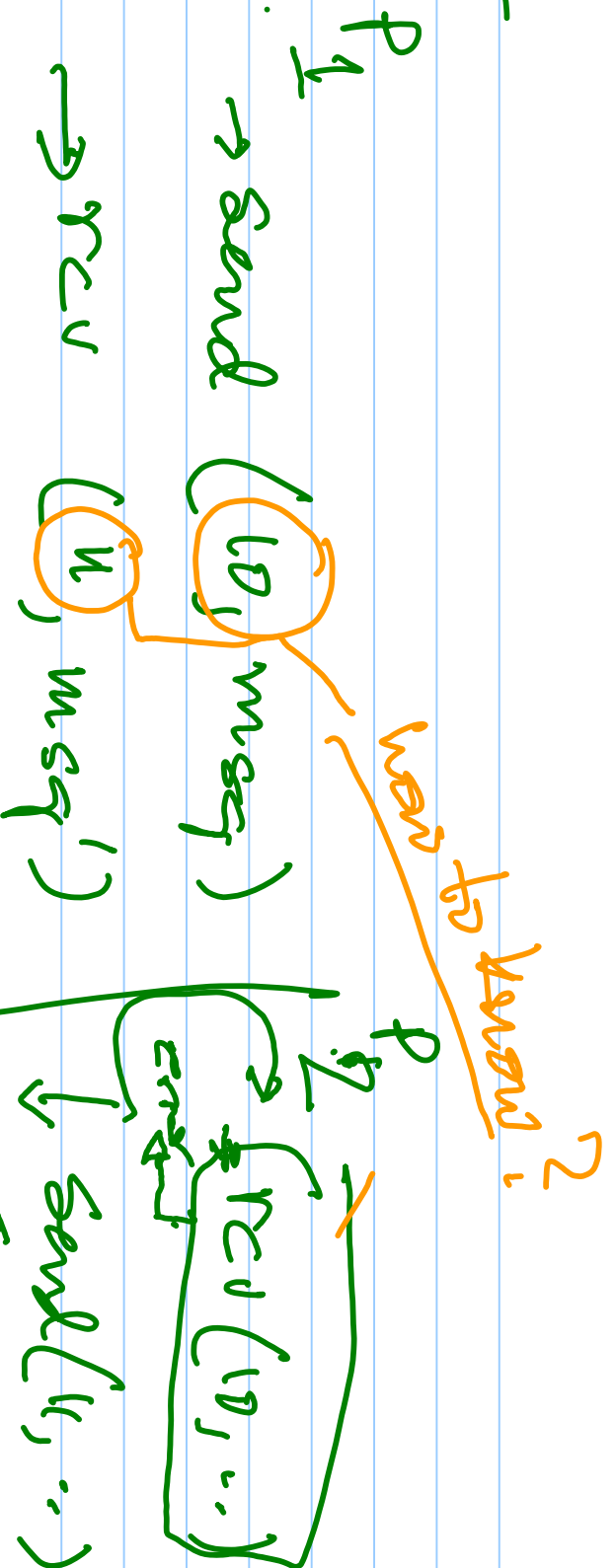
empty ... till a msg
arrives

+ no timeouts

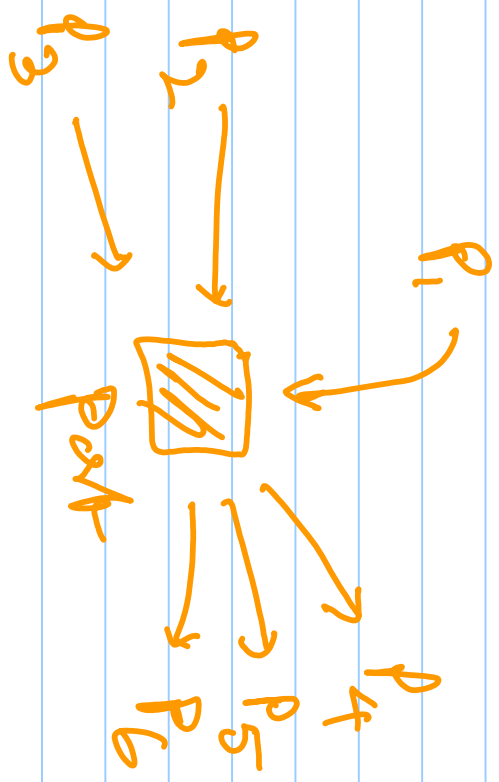
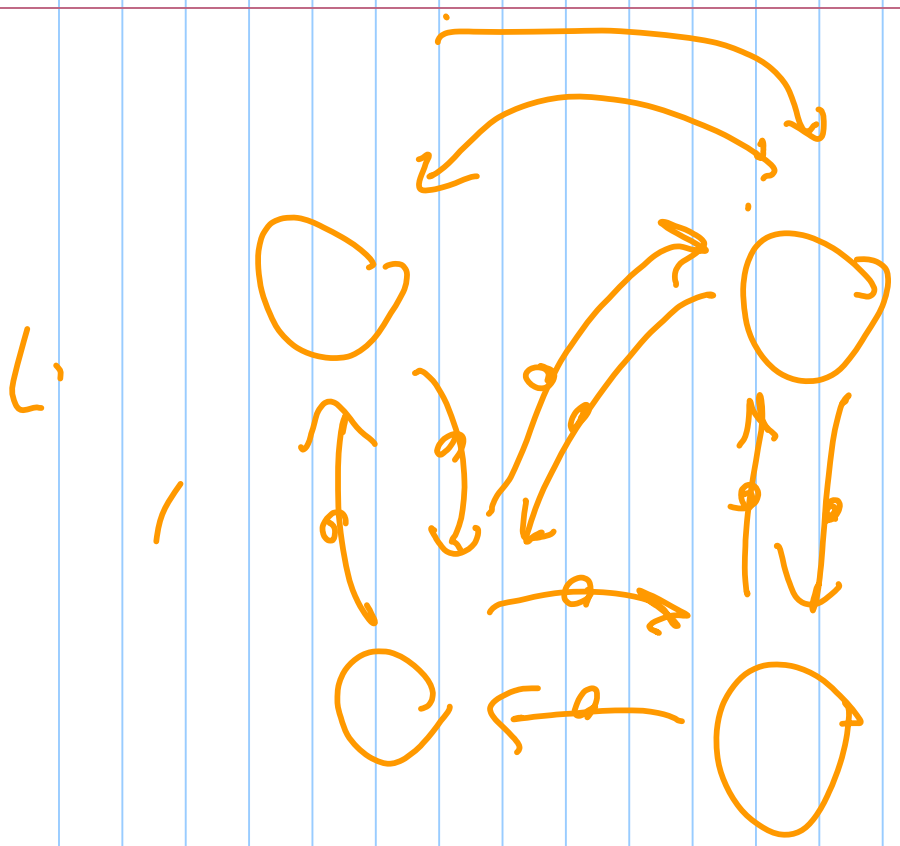


⇒ sends were blocking → this will work
 else will not work

2 way



3



1

— msgs will
be sequenced in
the packet in
order rec'd

— delivered in same order
packets are FIFO
