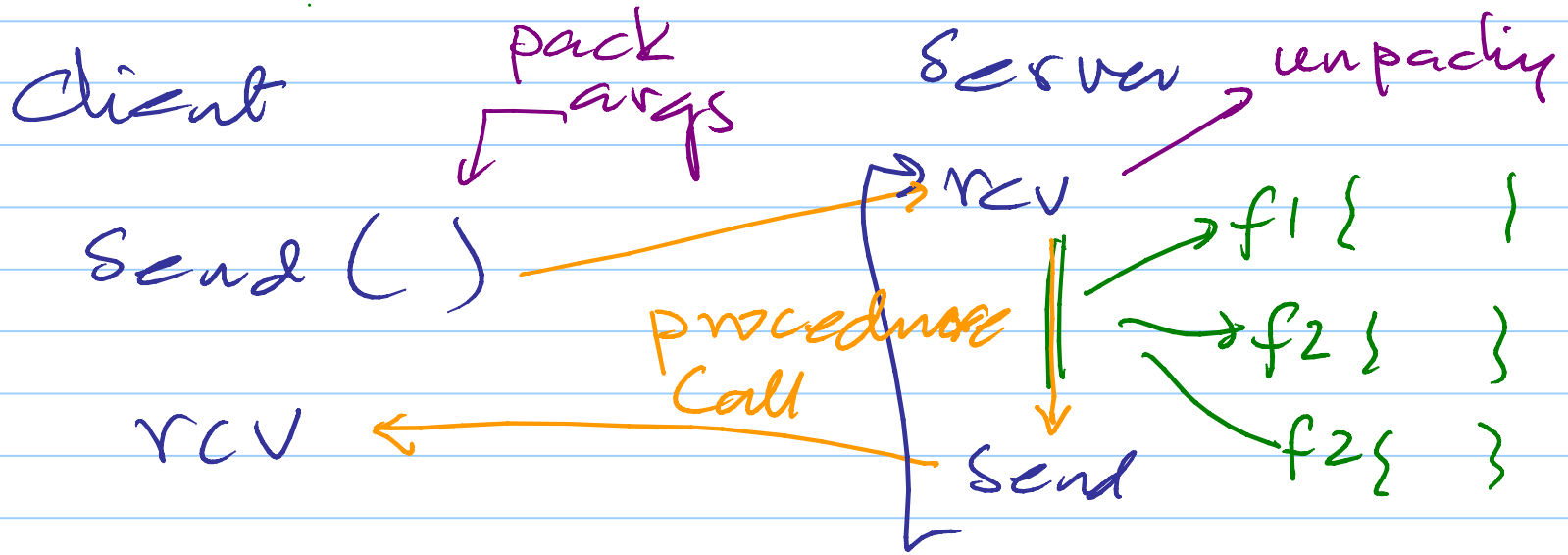
CSE 531: Distributed and Multiprocessor Operating Systems

Fall 2017

Class: 23 Date: 11/9

Note Title

Remote procedure calls (RPC)



RPC systems use a language (IDL)

IDL → Interface definition language

OLD - DCE-RPC, CORBA, DCOM

NEW - Java RMI, SOAP, SOA

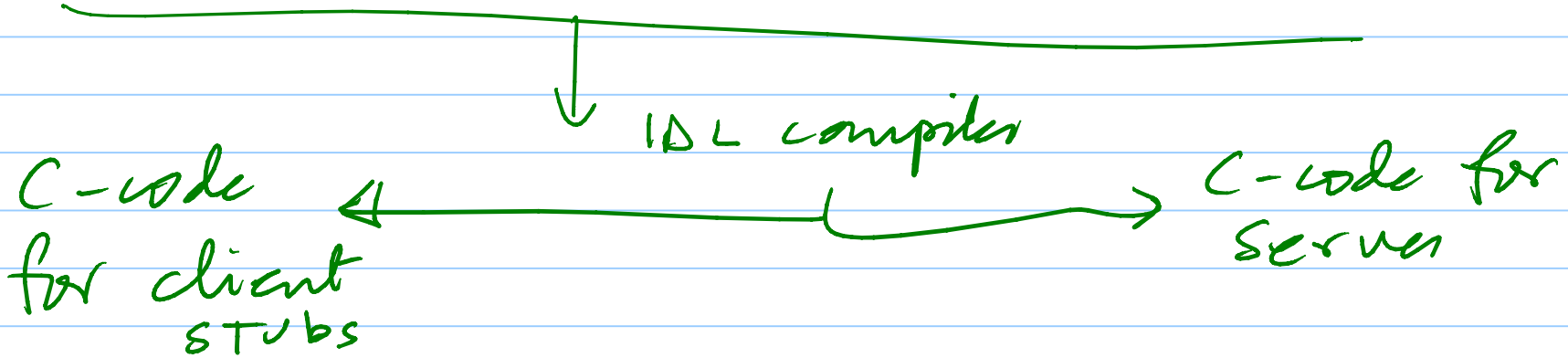
SOAP → simple object access protocol

SOA → service oriented architecture

Server code is in IDL

ext function f1(args) { body }

ext function f2(arg) { body }



Server → f1(arg1) f2(arg2)

client stub

+ initialization code for port

f1(arg1){

op-code = 0

code to pack args (marshalling)

send(sp, msg) ~~+~~ inserts replyport

rcv(replyport, msg)

unpack args (un-marshall)

}

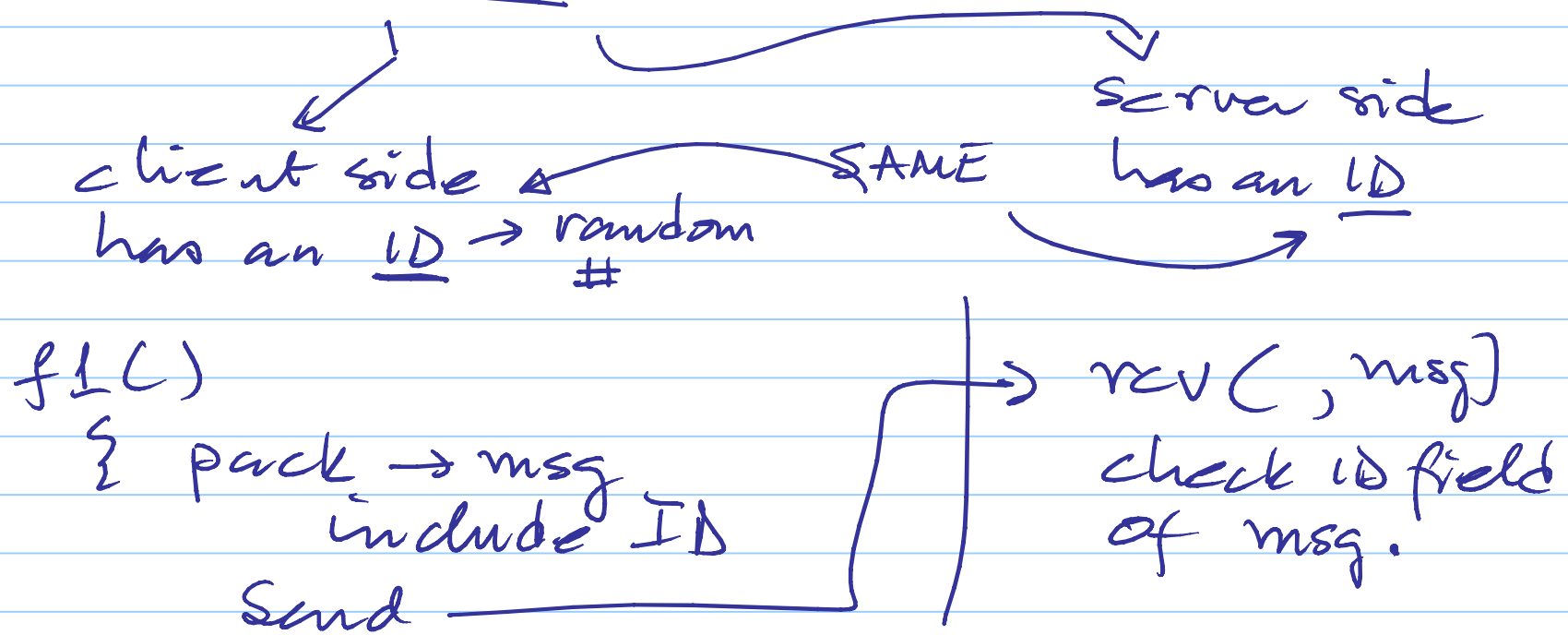
main()

f1(i, j, k)

Server code

```
main()
  { register a port
    ∞ loop
        rcv(sp, msg)
          unpack ms → op-code
                    - ↳ further unpack
                          arg
          switch op-code                    / send
              0 → f1(arg1) { code ,         }
              1 → f2(arg2) { code ,         send
```

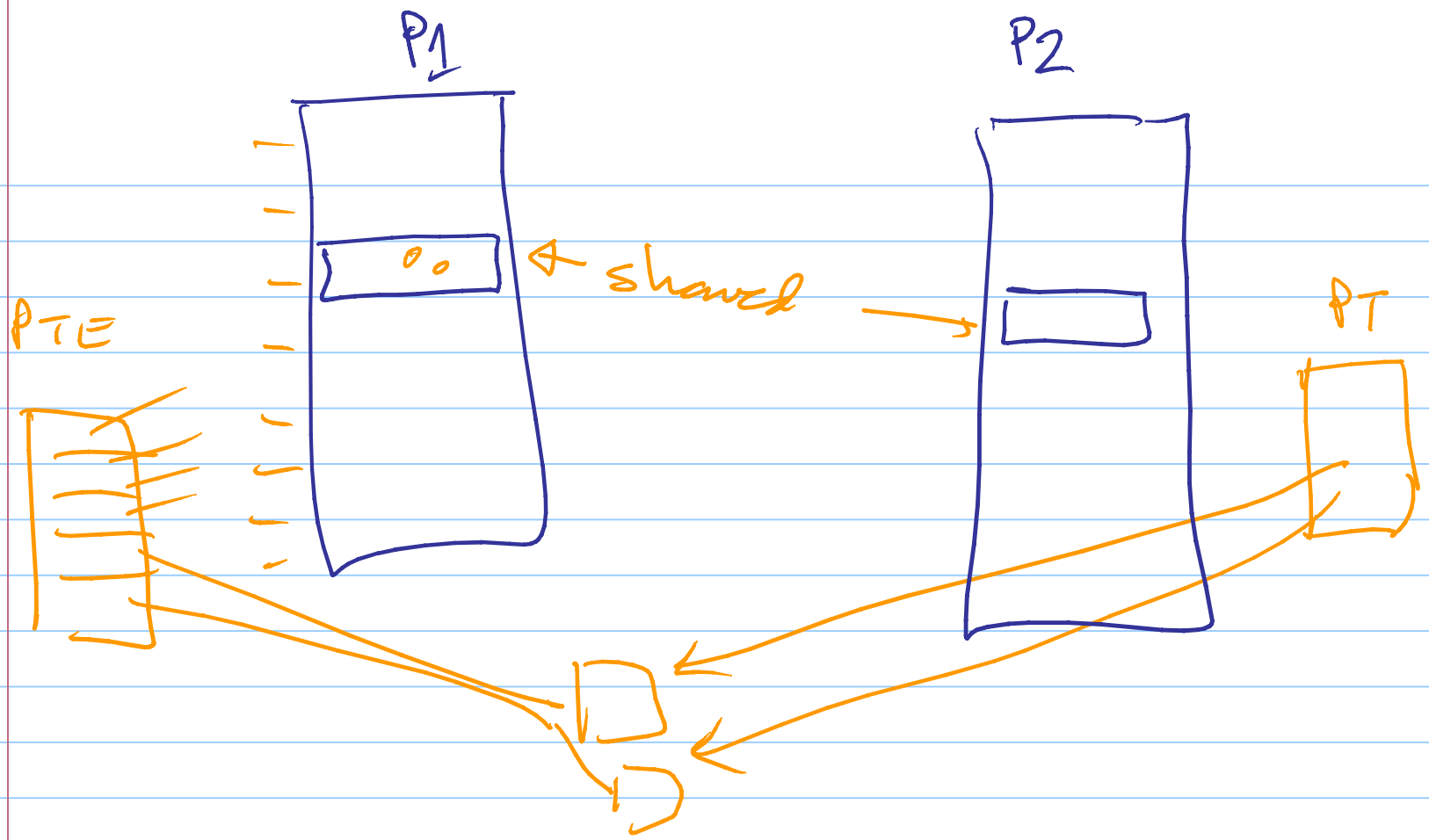
Version check



Distributed Shared Memory (DSM)

- Distributed means 'no shared mem'

Shared mem → one segment of RAM
↳ two or more threads
↳ two or more processes


P1
P2
shared
PTE
PT

DSM → 2 or more processes on different machines share one (or more) segments of memory.

↳ @ RAM Level

LOAD/STO access

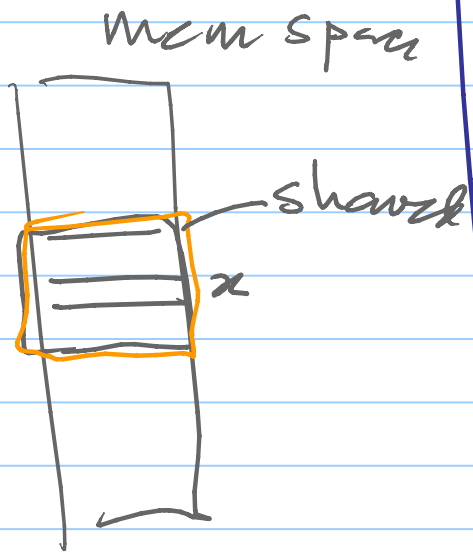
Process → DSM client

int x ...

{

x = x + 1

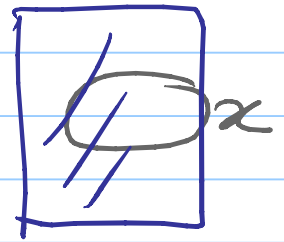
}



DSM Server

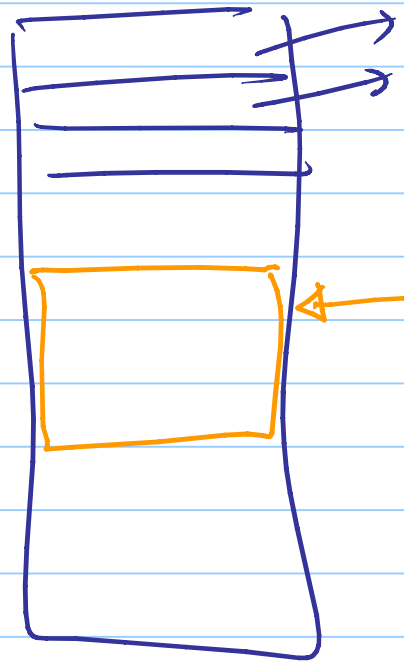
→ pkt → rcv

send



shared mem segment & contents

DSM client (process) page table



these entries are marked

1) invalid
2) no read access
3) no write access

client read page fault

```
read_pf
    { ask server for copy of page
      copy rcvd page to mem
      update PTE to point to
        set valid
        set read OK
        (write not OK }
    }
```

for reading

- will get it
- no locking

write pf handler

for writing

{ is the page in r-mode?

↳ NO → Ask server for page

get page, install

set W bit to OK

set R bit to OK

(No locking)

}

YES

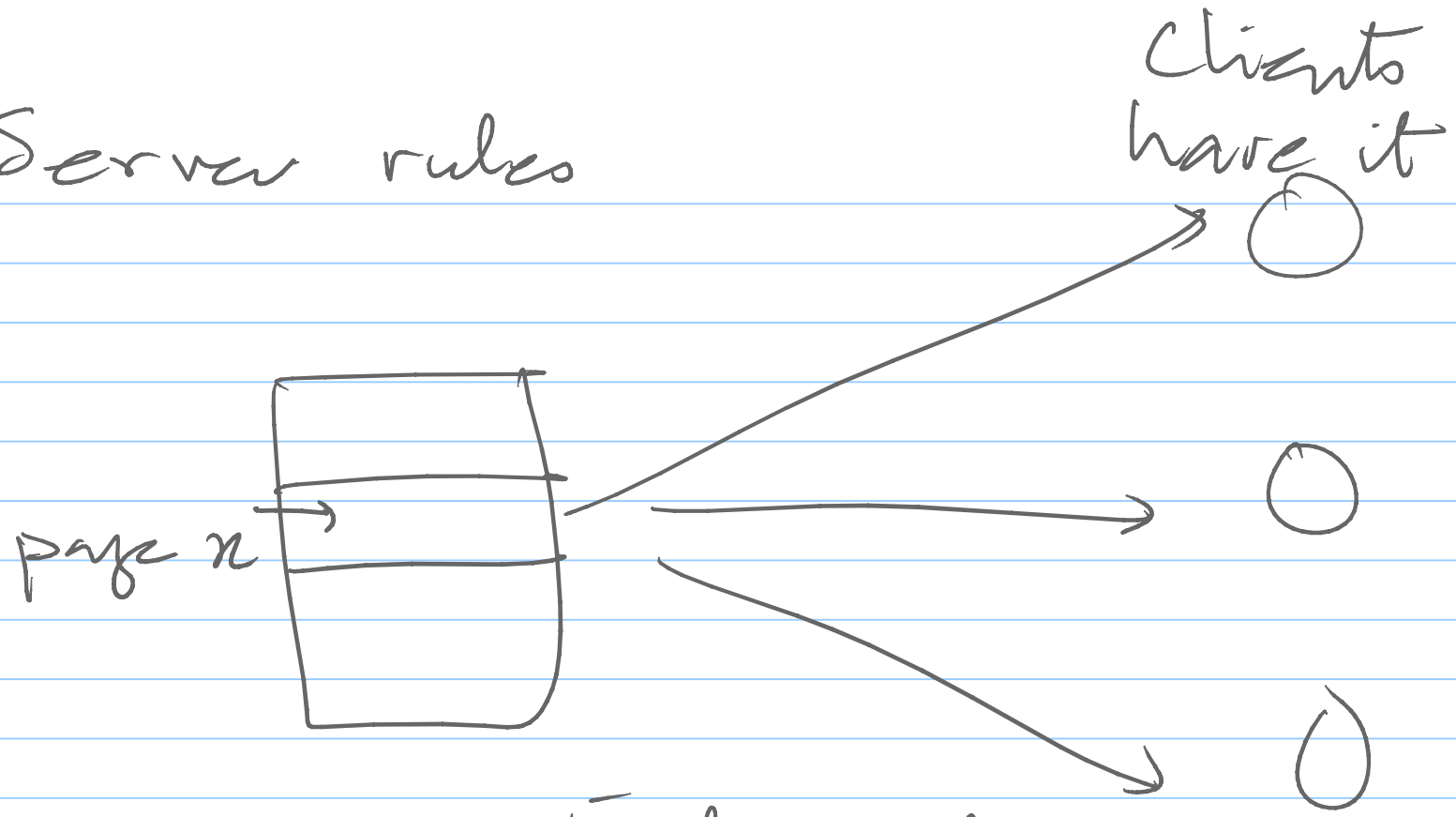
- Same

But no need

to get the data

- server gets request for page in W-mode

Server rules



RULE → multiple read copies OR 1 write copy

invalidation

P_1, P_2 has x in R mode

P_3 gets w-page fault

ask server for w mode on x

server invalidates P_1, P_2

gives copy to P_3

Power

LOAD n

STD n

Server code

→ r_page fault (request)

{ if page in R mode or none mode

{ send copy of page,
add client to list of owners
set mode = R }

if page in W_mode

{ send invalidate to owner
get copy of page (latest)
send copy
add owner
mode = R }

W - page fault request

{ page in none mode → Send page
Set owner
Set mode = W

page in R mode

↳ send invalidates (R) to all owners
wait for ack
send page ... etc

page in W mode
↳ send w-invalidate
get copy of page, send copy
set mode, owner

---

invalidation server at every client

→ r_invalidate
  → set r, w → none
  → send ack to server

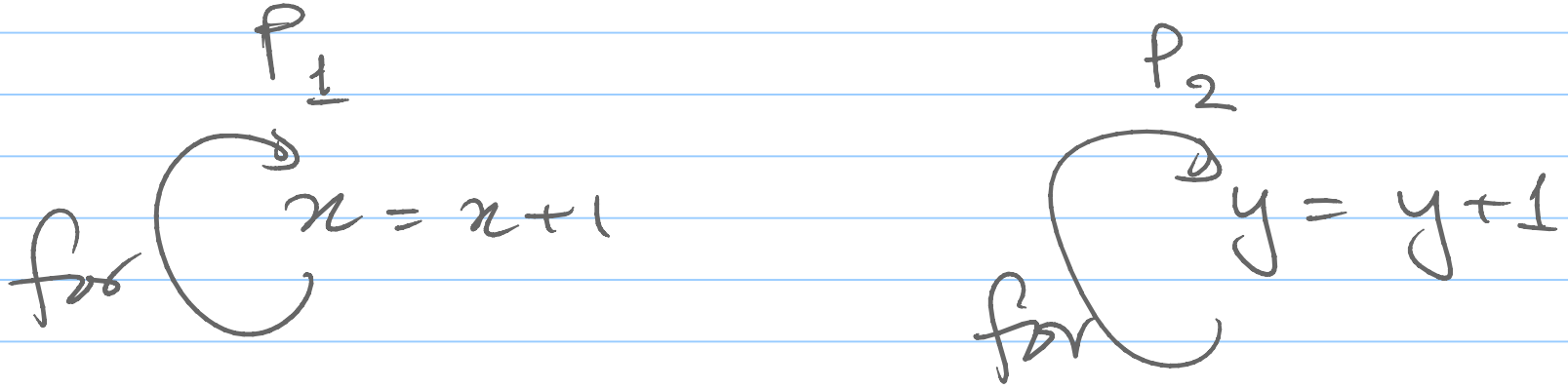
→ w_invalidate
  - set r, w → none
  - send page to server

→ no locking → race conditions

→ use locking

→ page shuffling is reduced.

DSM has false sharing



x & y are on same page!

→ page shuttling happens.

Release Consistent DSM

- no page shuttling
- not coherent
  (but looks coherent)
- no false sharing