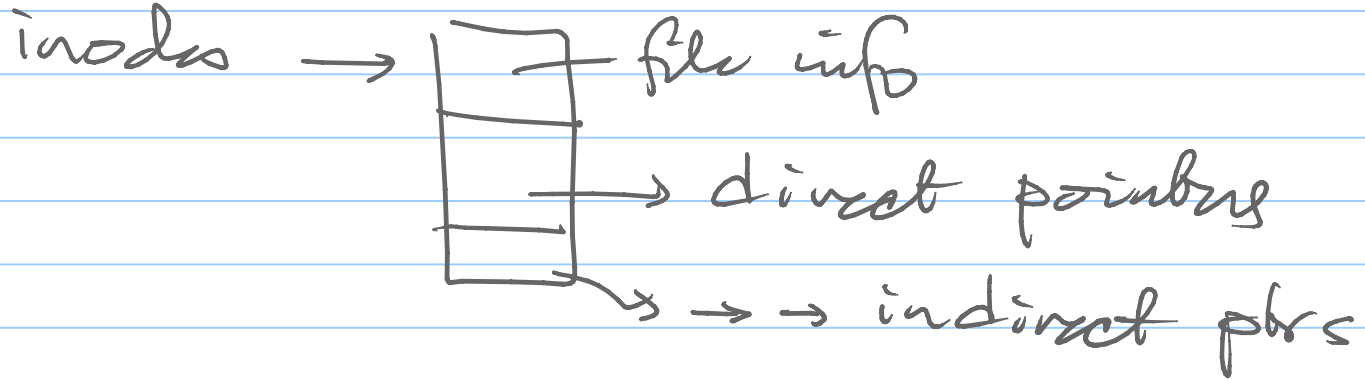
CSE 330: Operating Systems

Class: 27 **Date:** 4/19

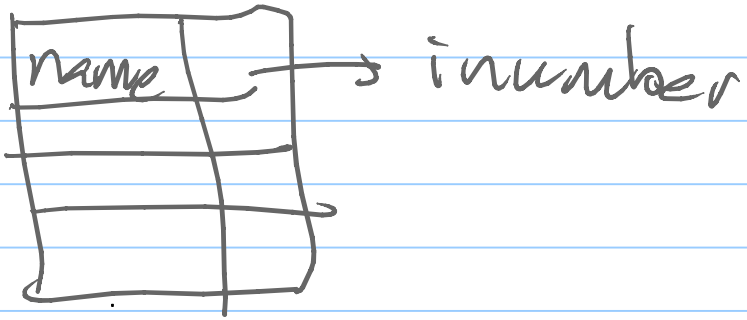
Spring 2018

Note Title

Files



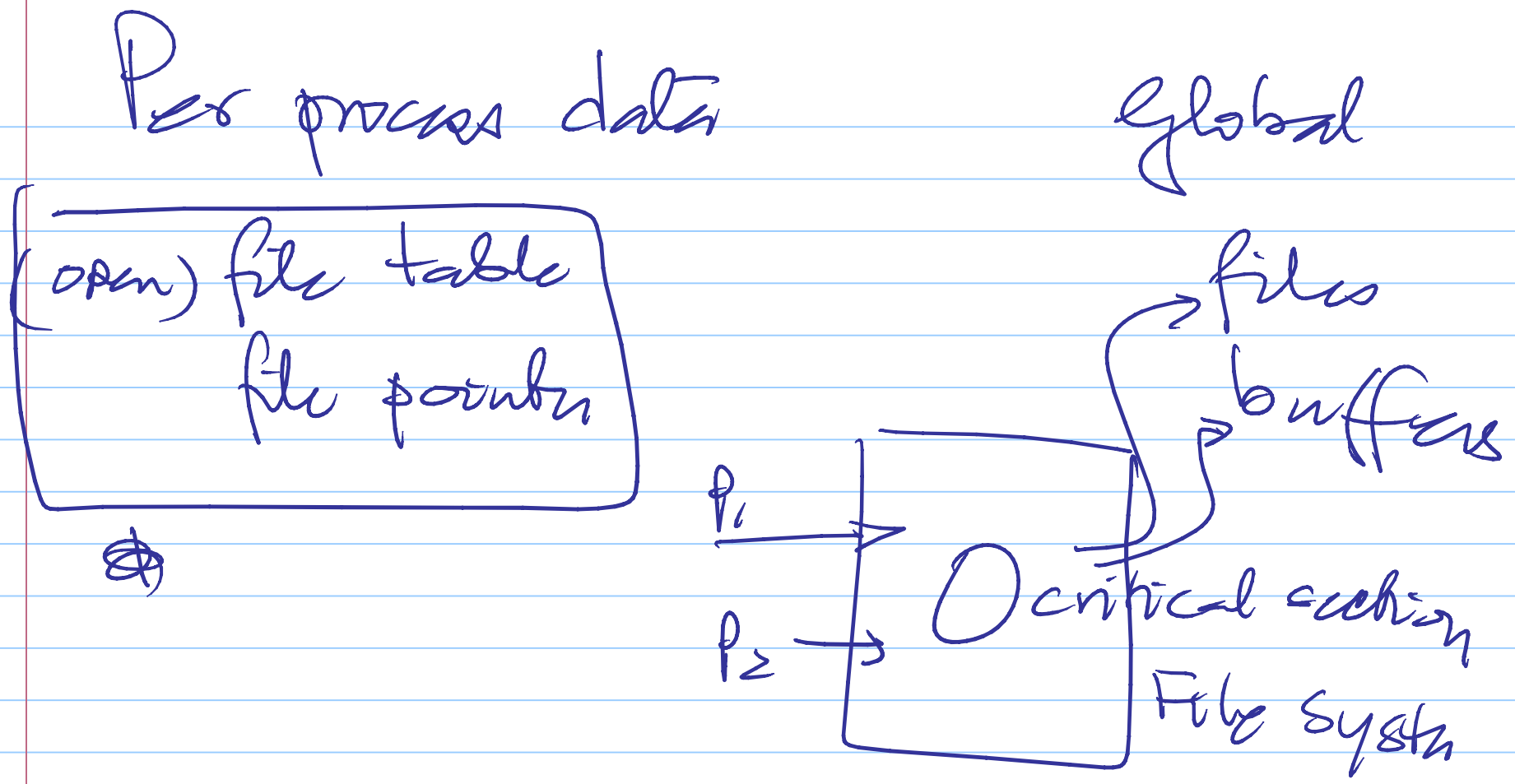
Directory



File System, system calls

- create
- unlink
- open
- read
- write
- close

(many more)


Per process data
(open) file table
file pointer
Global
files
buffers
P1
P2
critical section
File syst

file creation/deletion is atomic

- P_1 & P_2 create f1
  - someone creates
  - other gets "already created error"
- P_1 & P_2 delete
  - one deleted, file does not exist

P_1 P_2 → open f1

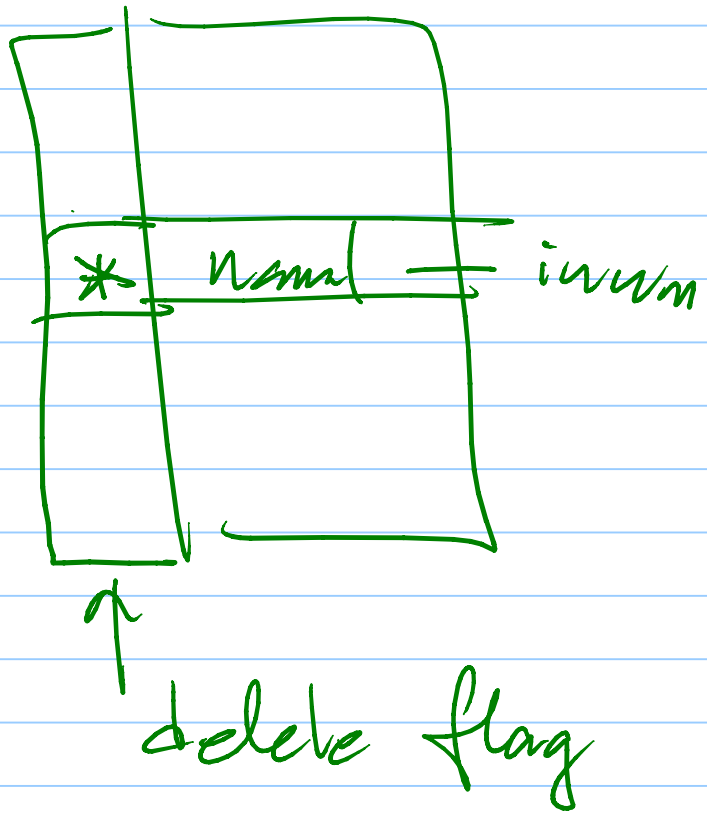
⋮

delete f1

→ 1st delete will close the file & delete dir entry

→ 2nd delete will delete data

file delete



Delete

↓

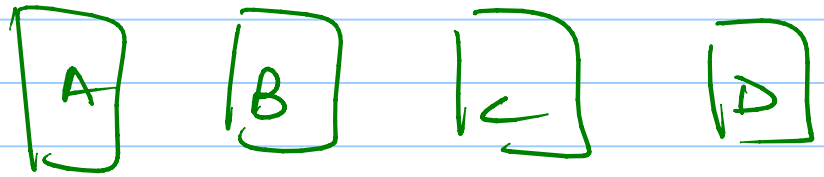
→ 1 add del flag to dir

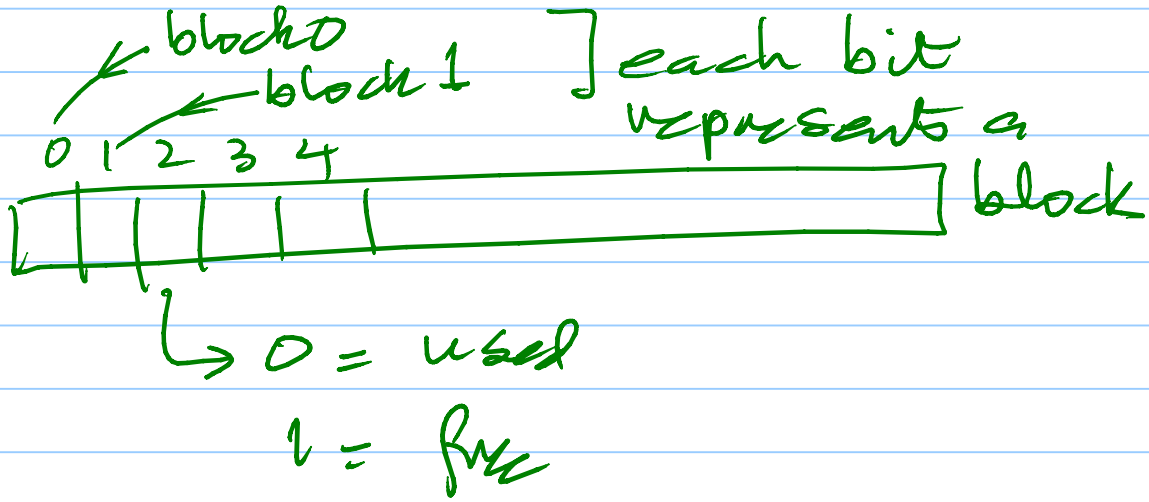
→ reclaim data blocks

reclaim data blocks

free list

- linked list
- file
- bit vector





Bit vector

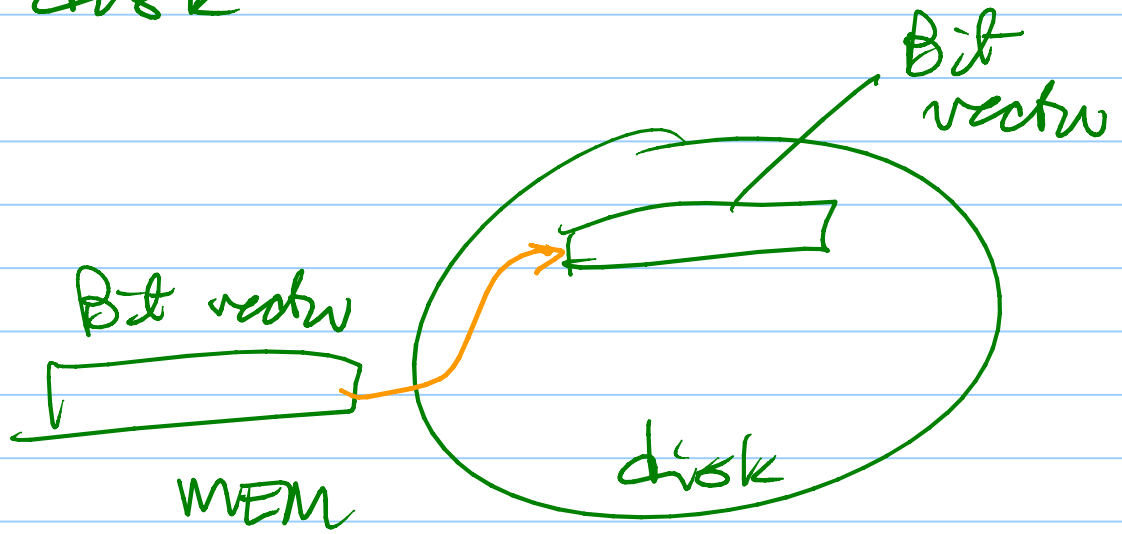
↳ in memory

↳ on disk

↓

Both

?



Power failure affects files too

- blocks written to buffer get lost
  - blocks added to file may be marked as "free"
    - → more

avoid disk inconsistencies

→ everything written instantly
(too slow)

→ delayed writes

↳ sequencing important

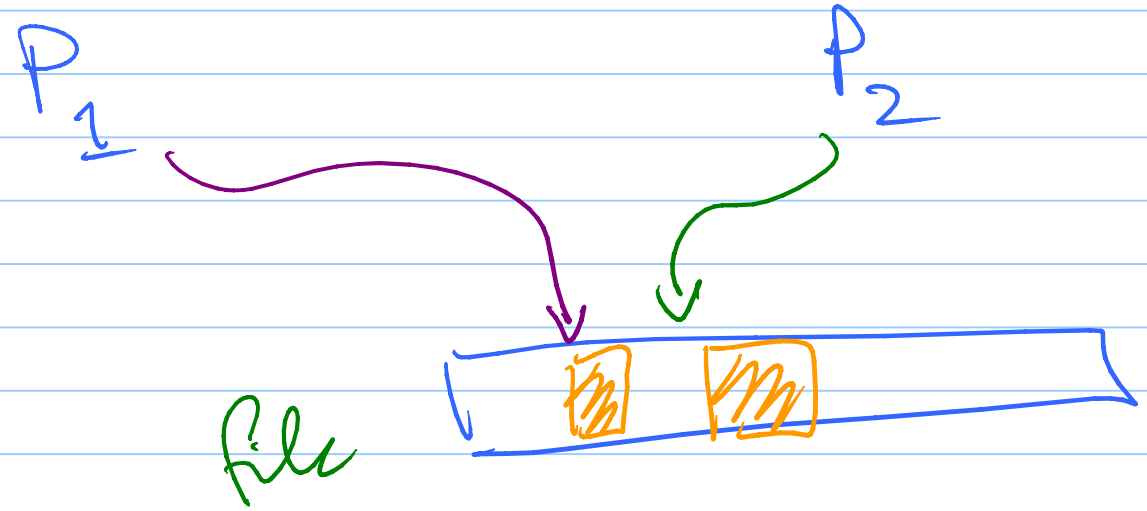
recover inconsistencies

↳ stuff on disk is outdated (metadata)

Checkdisk on startup

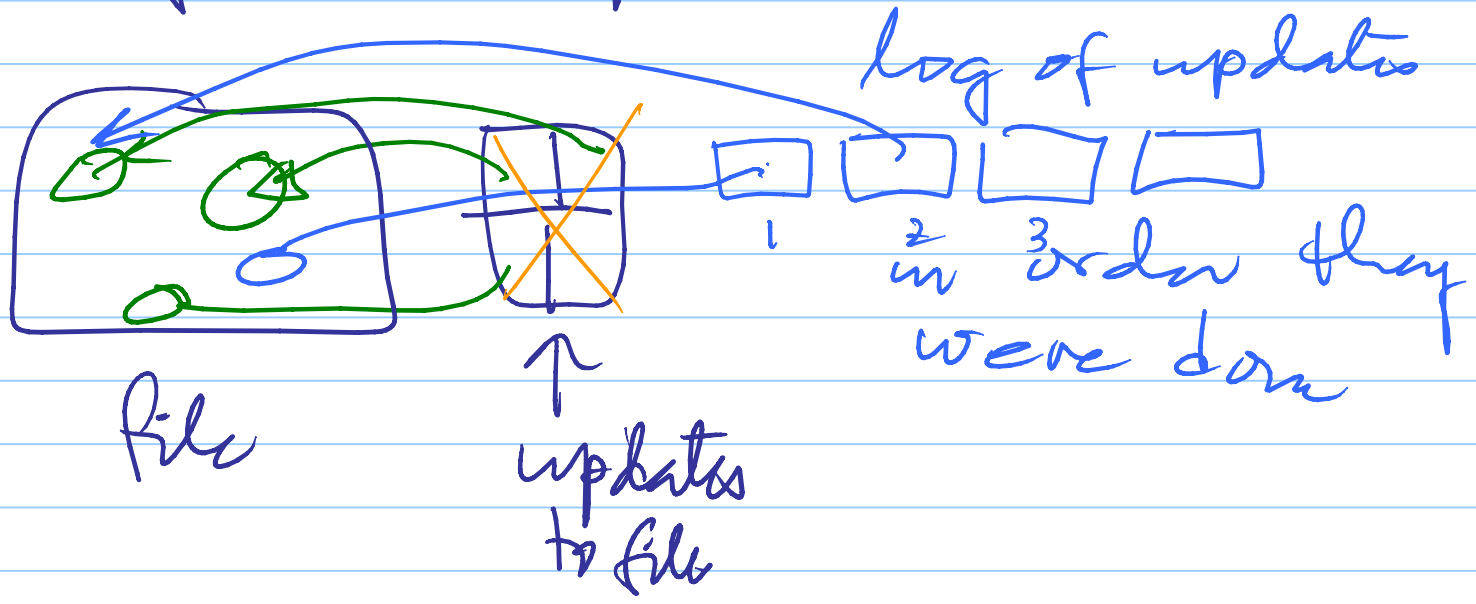
- chkdsk (win)
- fschk (Linux)

Concurrent file access



OS → does not control access / no locking
→ every read will return the most recent write

Journaling file systems



Log structured file system

→ there are no files

→ only logs.

Protection / Access control

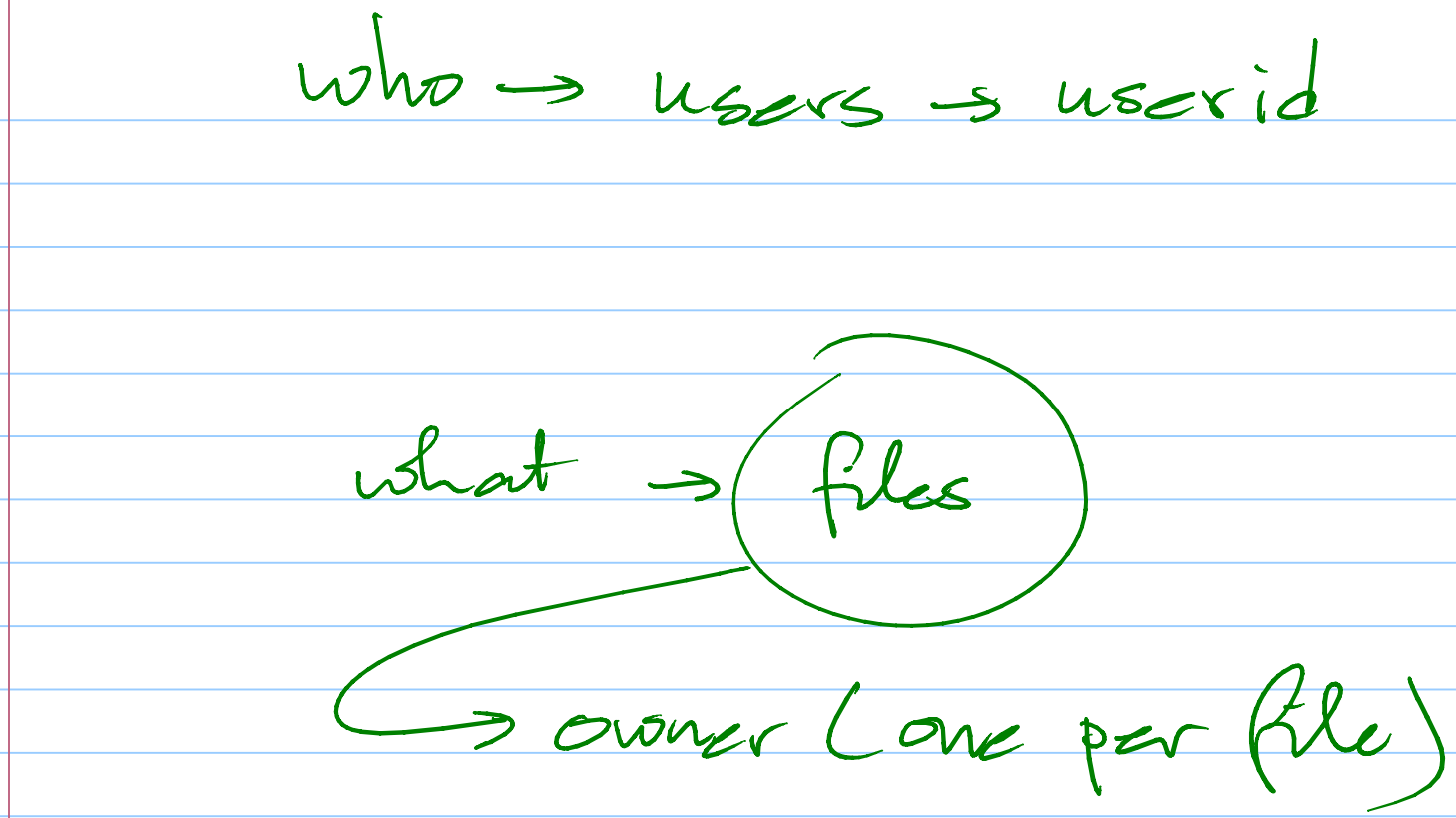
↳ low level] → processor (CPU) protection → CPU mode

→ memory protection →

↳ base/bound

↳ page tables

→ high level protection → who accesses what


who → users → userid
what → files
→ owner (one per file)

Owner can change file permission

↓

Application that runs under Owner's ID

Alice (human) → Computer

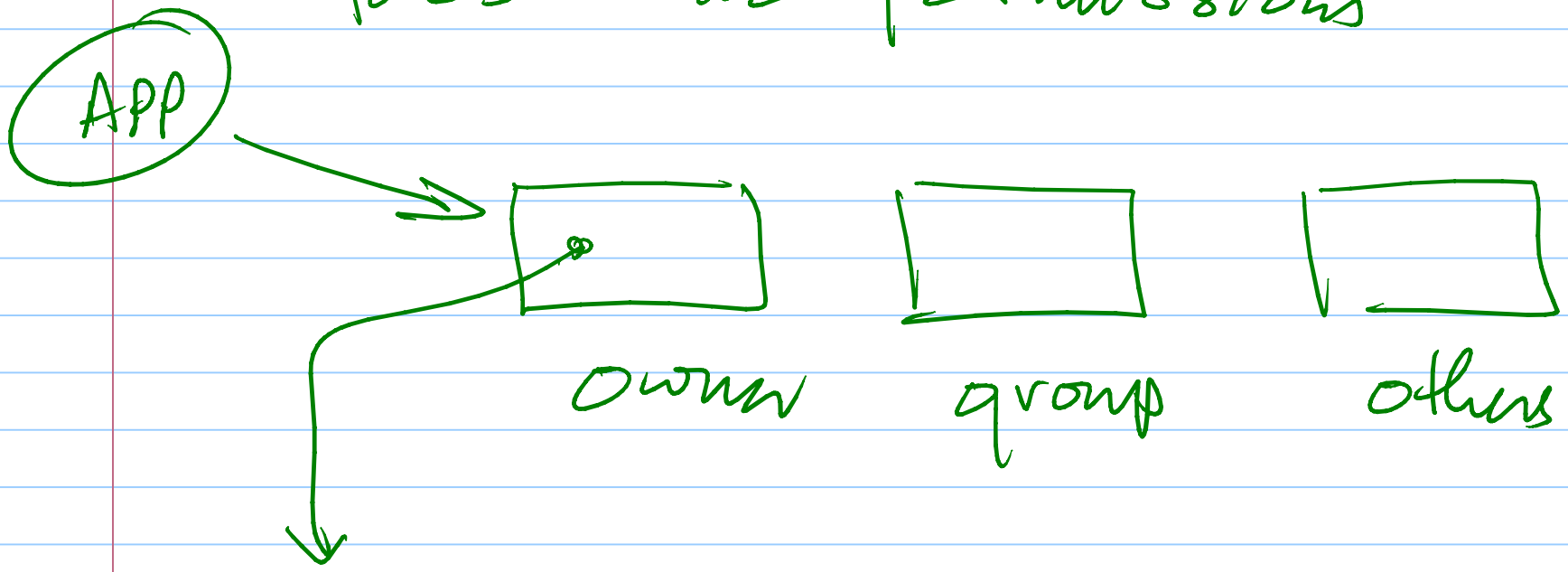
↓

runs apps

↓

all have Alice as UID

files have permissions



permissions are checked

~~at~~ by file system when "open" done

App

user id

file

owner ID

root, administrator

↓

Super User

→ can write/delete anything

→ can change permissions of anything

" " passwords of anyone

privilege escalation