

File Systems

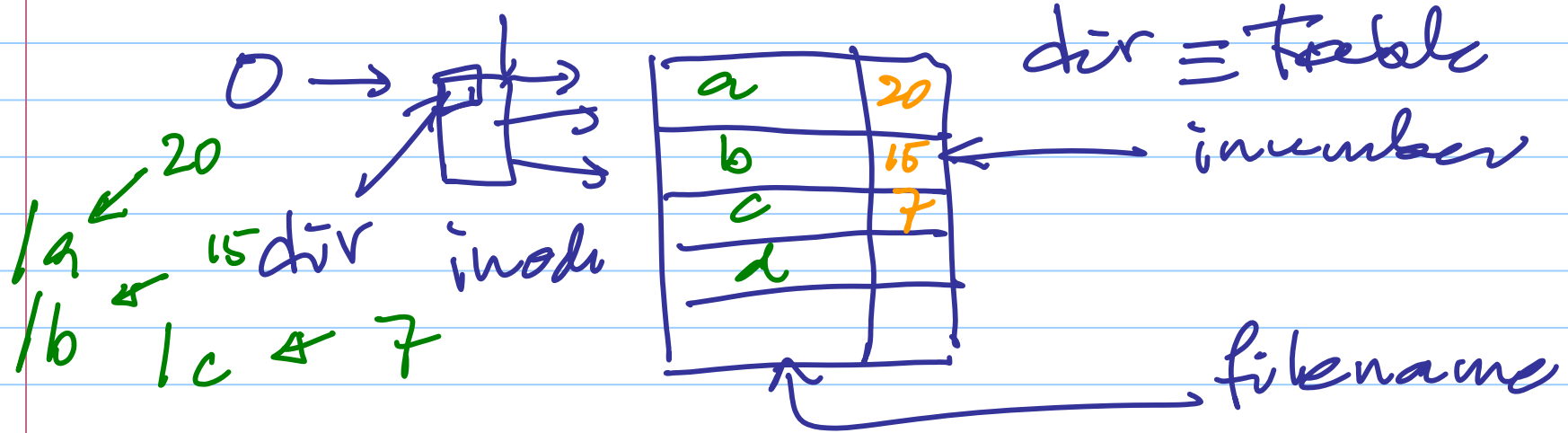
- inodes
- index structures
- file data
- directory tree

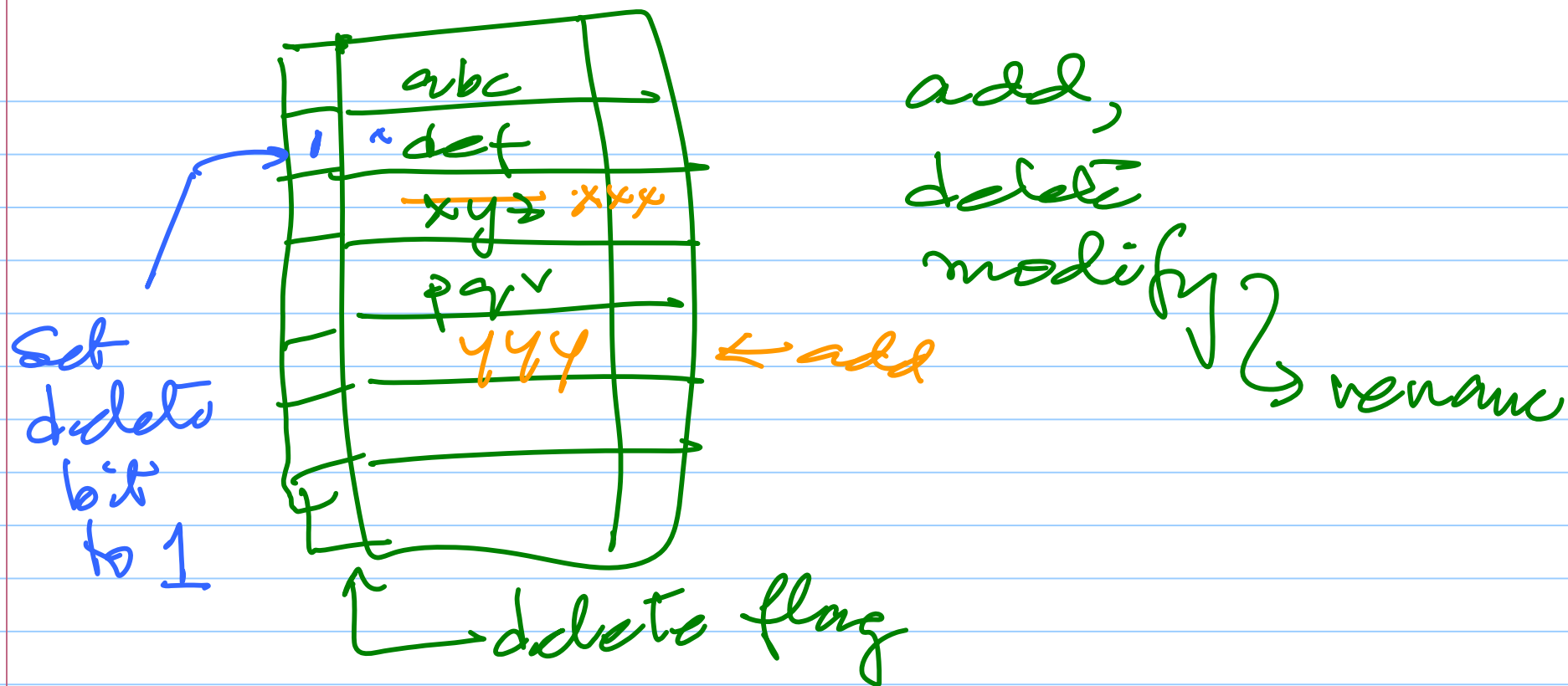
—

inumber 0 \rightarrow root directory \rightarrow /

directory is a file

\rightarrow but in inode it is marked.





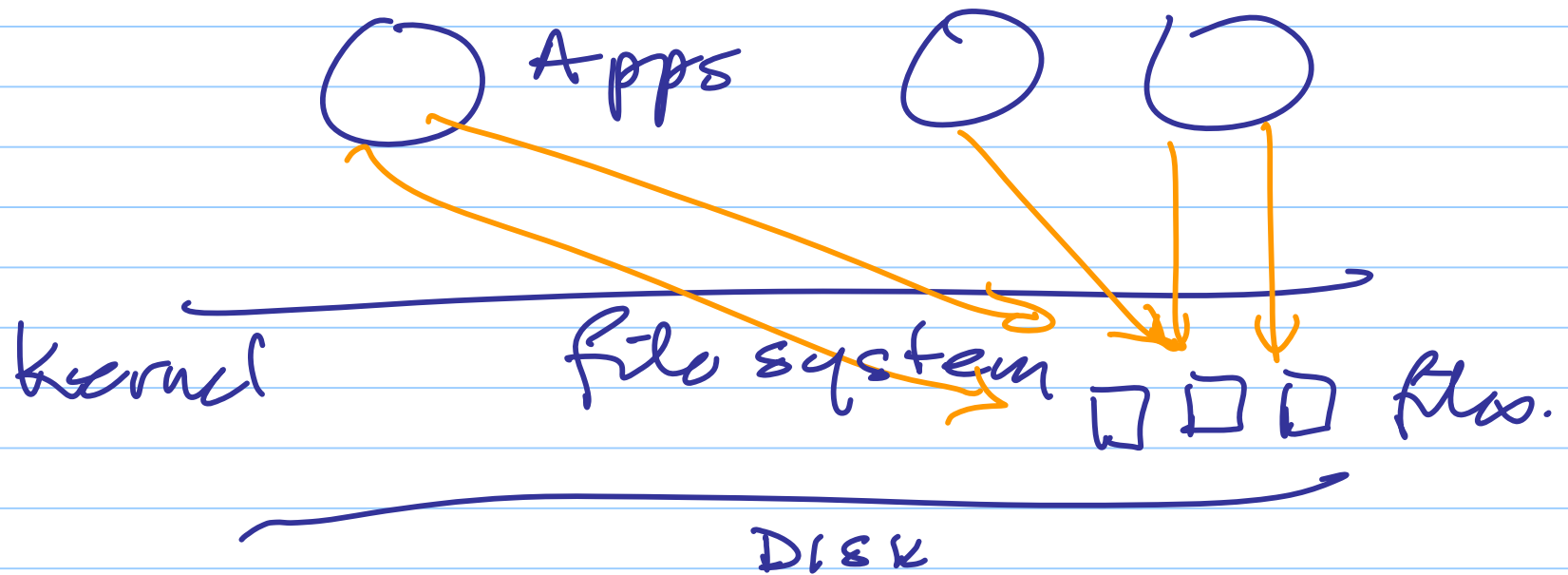
free list → bit vector
→ one large file
→ linked list

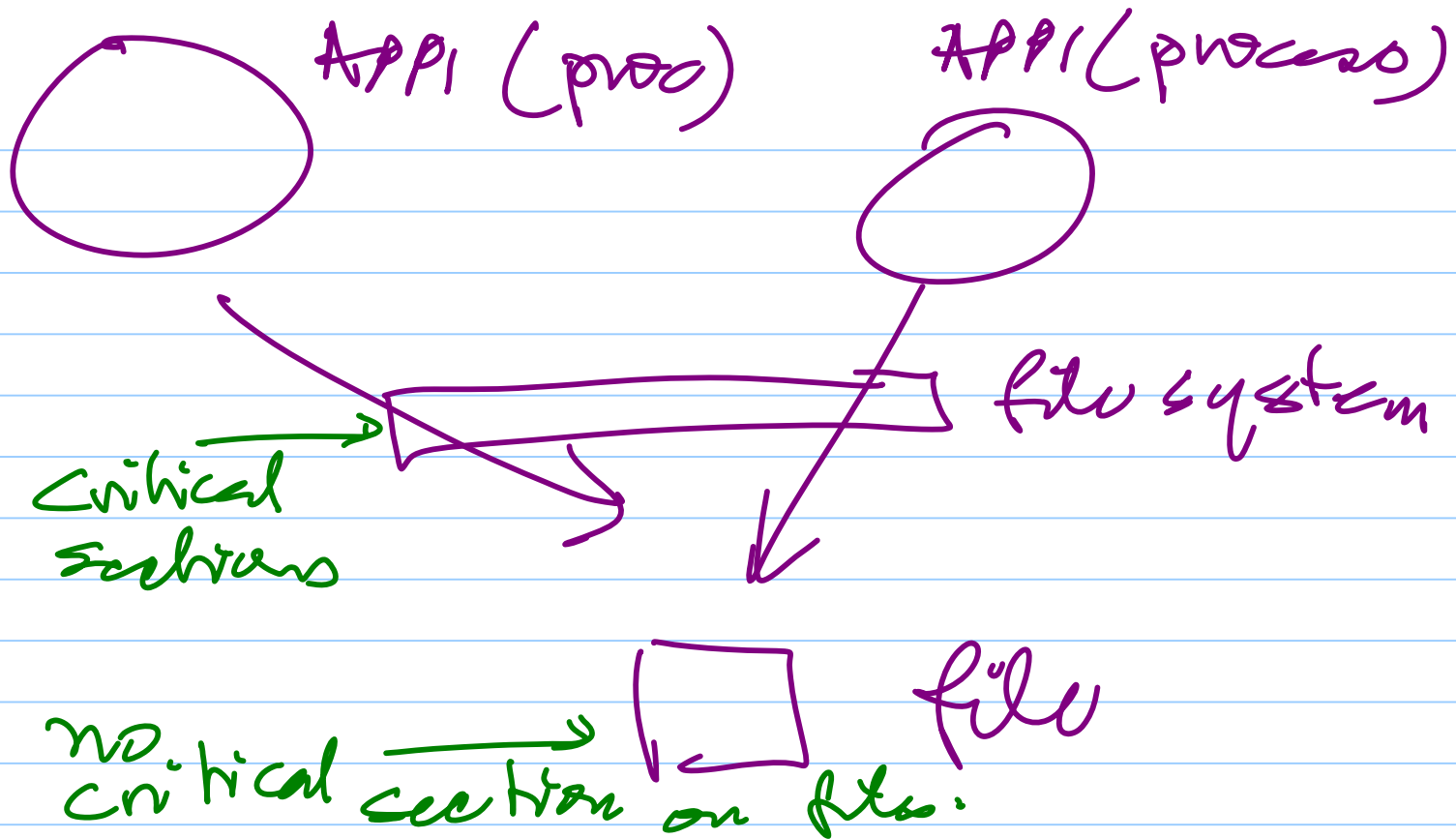
} ?

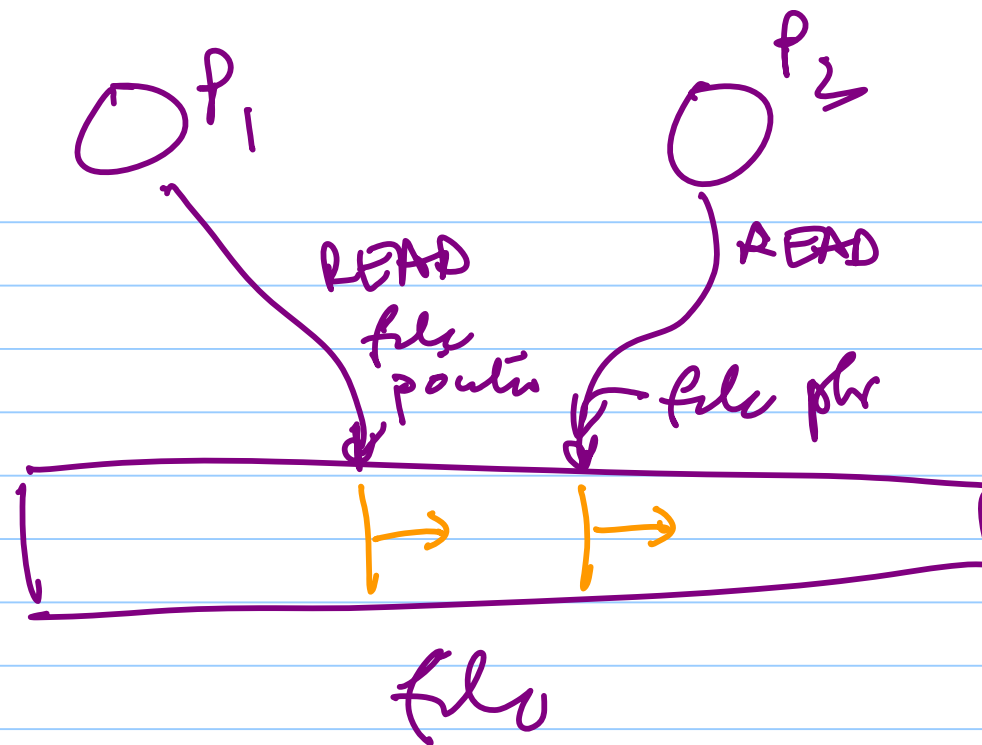
delete a file (unlink)

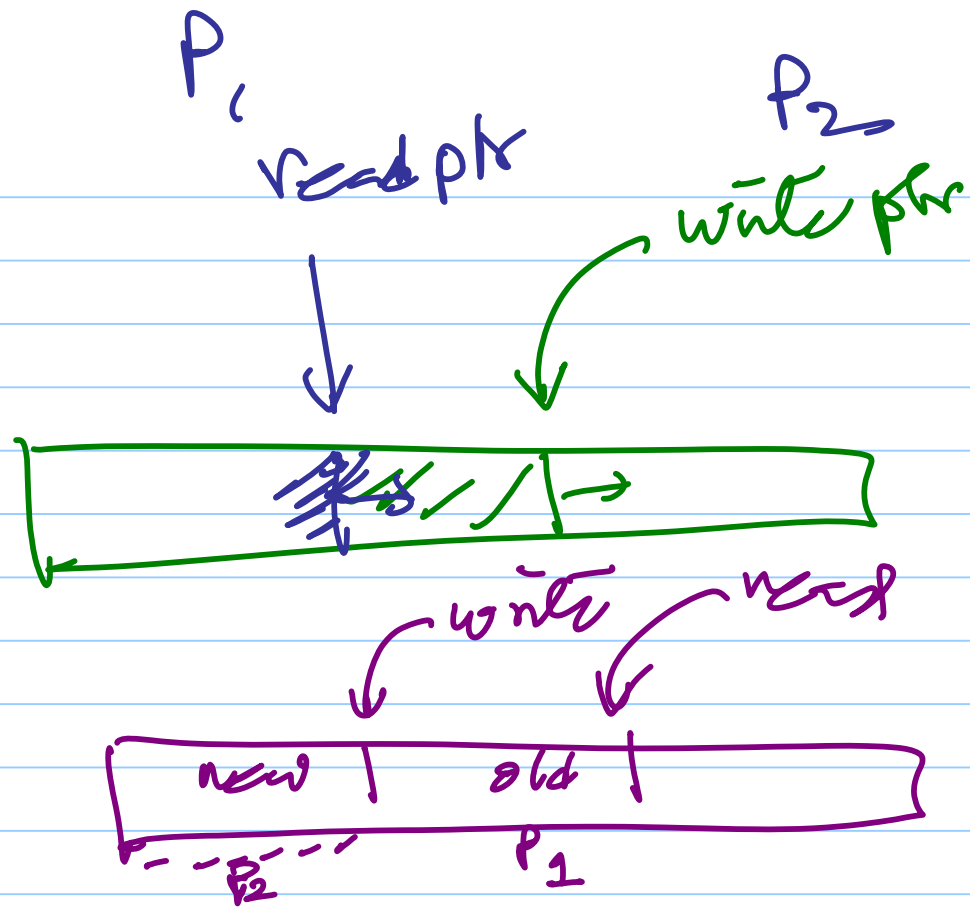
- delete dir entry
- add all blocks to free list

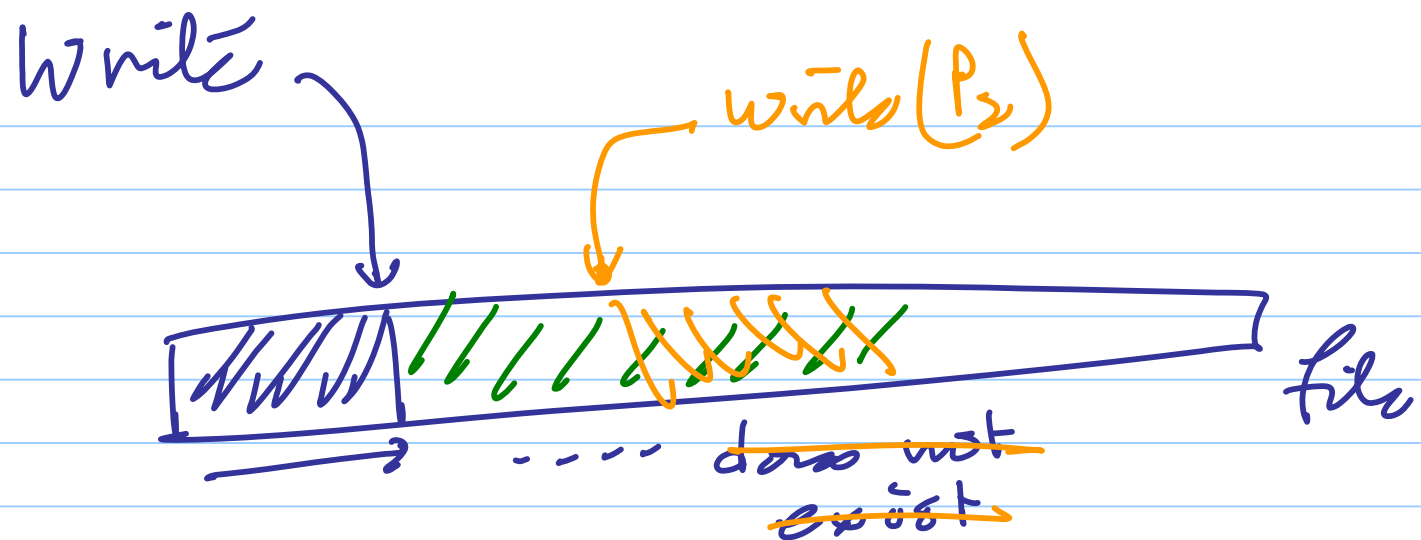
Runtime Semantics



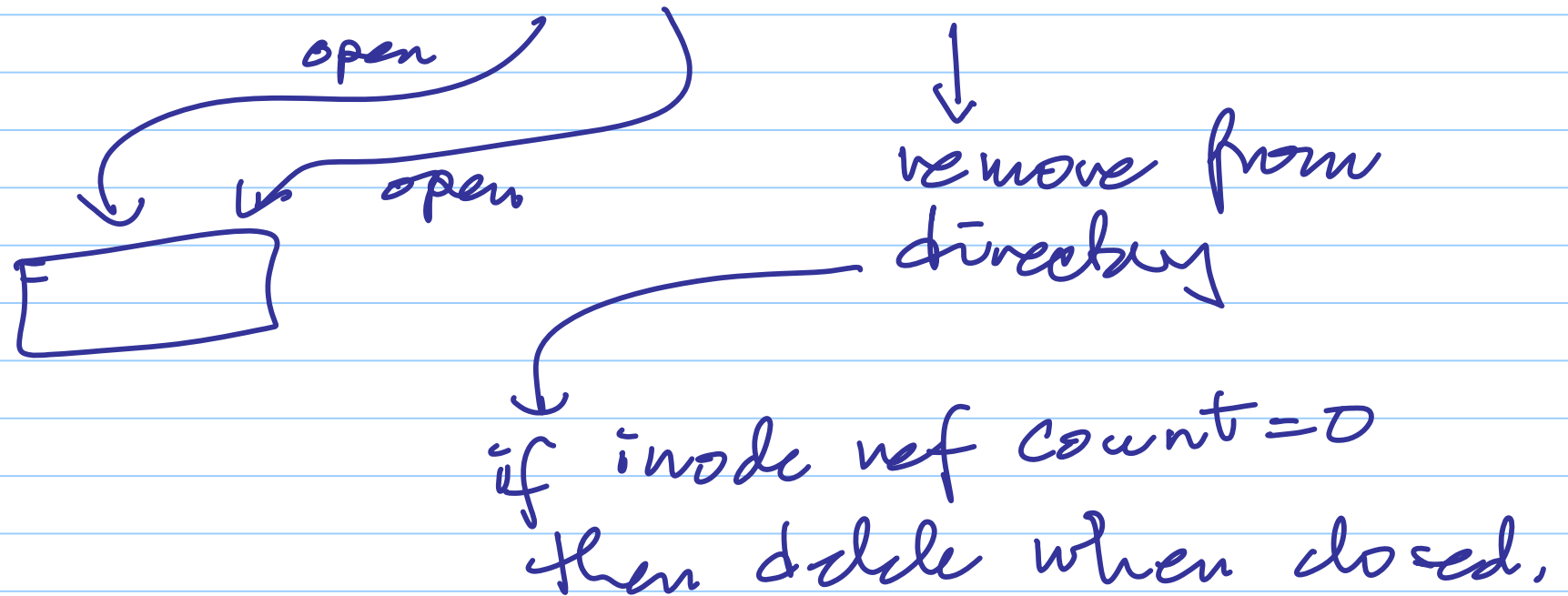








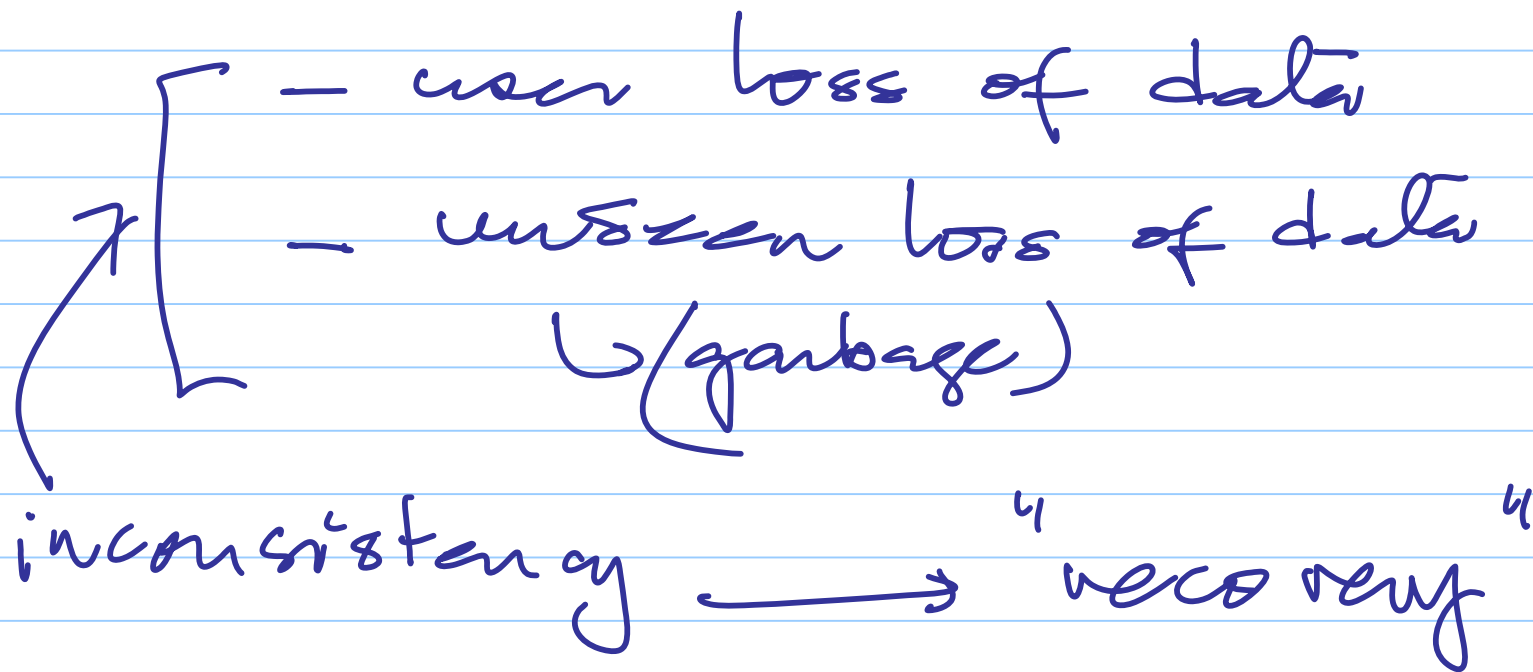
Concurrent R, W + delete



Consistency & recovery

→ buffered in memory
(meta data + regular data)
— fail (power or crash)

Leads to

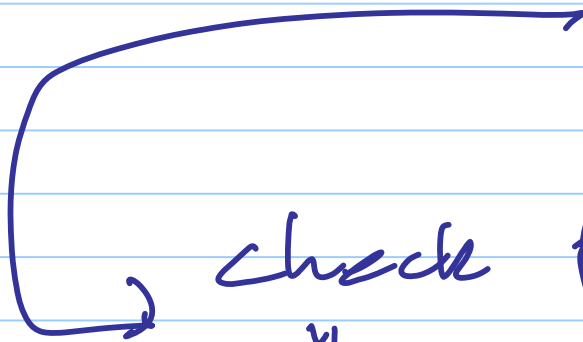


Recovery

fsck (unix)



file system check
(run on boot)



check free space
" inodes in directories
find lost chains...

journaling file systems

(e.g. Linux ext3, ext4.)



updates to file system data
(meta + real) is written to
"journal" → log

Log structured file system

↳ the journal is the
file system (!)

Database file systems

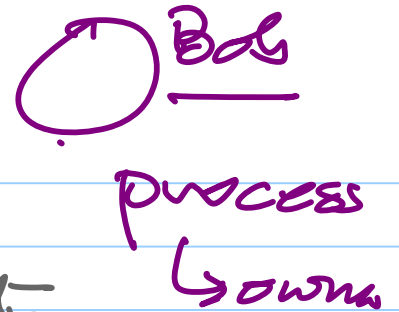
- combine logs
- before and after images
- "commit" logs.

Protection

- memory protection (boundaries)
- CPU protection (modes)

- ↓
- Access Control ✓
 - Process rights ✓

Access control (Unix)

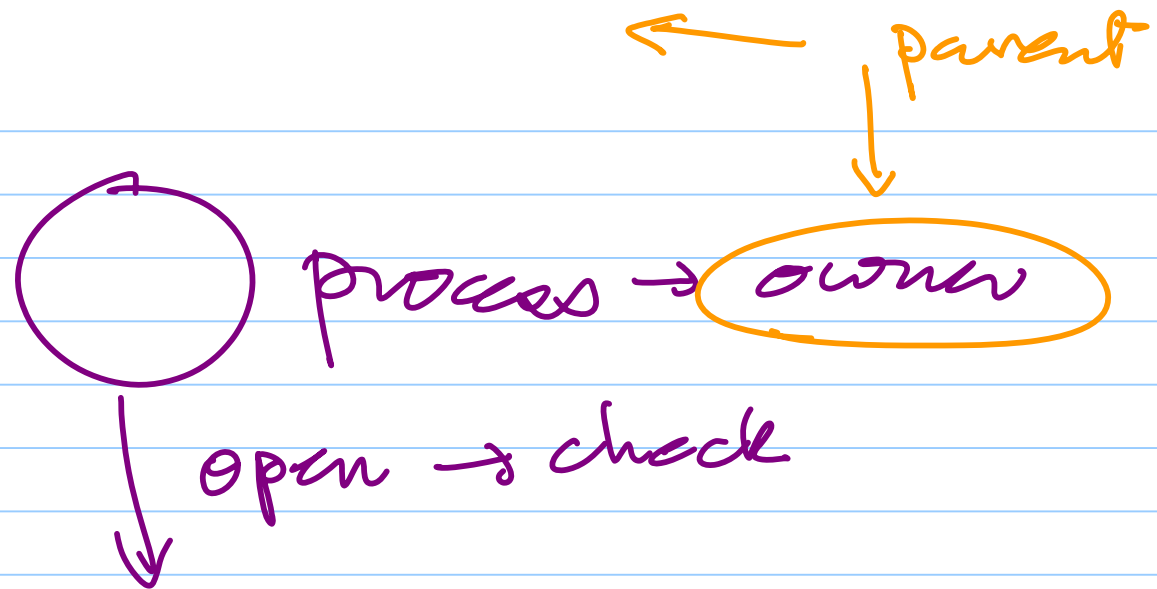


files have owners ^{+ grant} & rights

owner = Alice group = AGroup

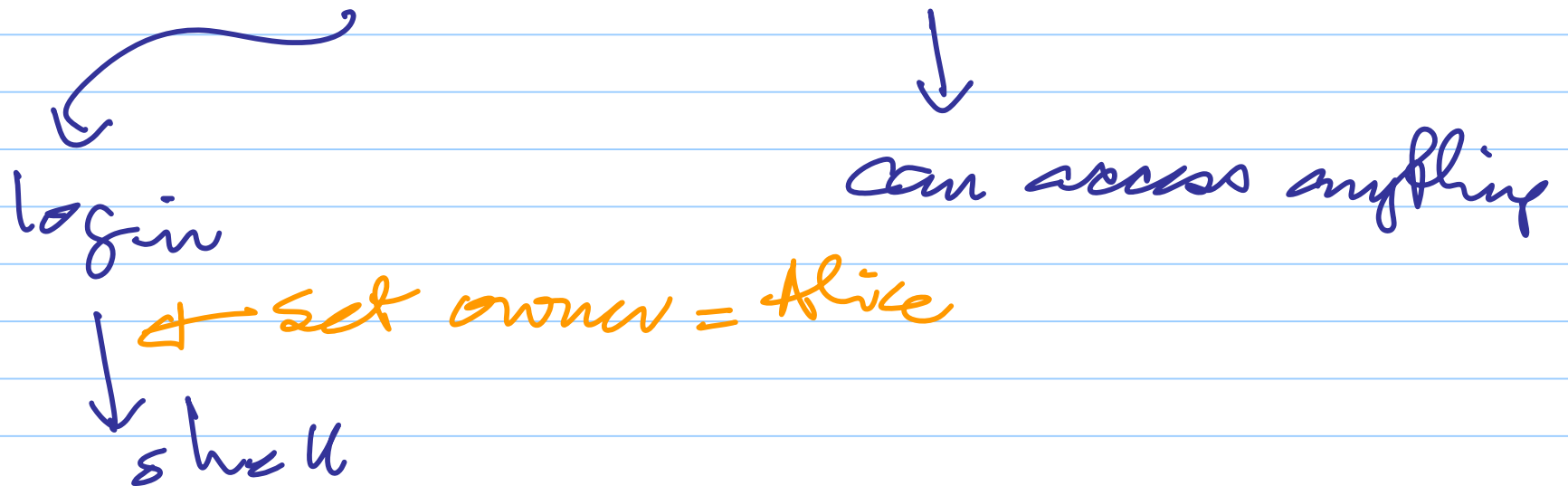
access rights

$rw\<, \underline{rw\<}, \underline{rw\<}$
 ↑ ↑ ↑
 owner group other



file has 9 permission bits

ultimate parent process (init)
has owner "root" \rightarrow uid 0



Root → Alice → root
(normal)

↓
system
can → & verify
(manual)

↑
priv
escalation
↓
"Setuid"