

6. SOBI: Crash safety

---

Problem: Crash can lead the

on-disk fs to be incorrect state

\$ make

!  $\leq$  × power failure

Solution:

logging

reboot

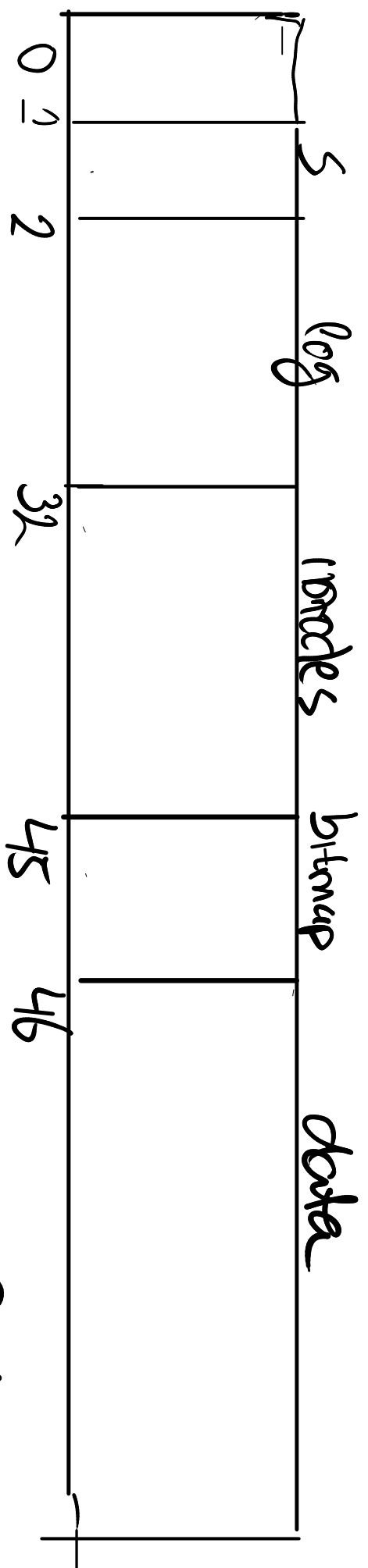
\$ ls?

# Risks

fs operation are multistep  
click operation

Crash may leave fs invariants  
violated

After Reboot:   
    crash again   
    no crash → but  $R/W$    
    data corruptly



```

$ echo "hi" > x
write: 33 allocate inode for x
write: 33 init inode x
write: 46 record x in /directory's data block
write: 32 update root inode
write: 33 update inode x

```

X power failure  
 ↳ lose inode

write 46  
 write 32  
 write 33  
 ↳ good  
 either.

# EX write X.

--- write "hi" to file x  
write: 45 set alloc bit in bitmap block  
write: 595 write h to allocated data block  
write: 595 write i to allocated data block  
write: 33 (size update, bn0)

lose data block

WRITE 33 (w.595) ← failure

WRITE 45  
block is shared  
among multiple files

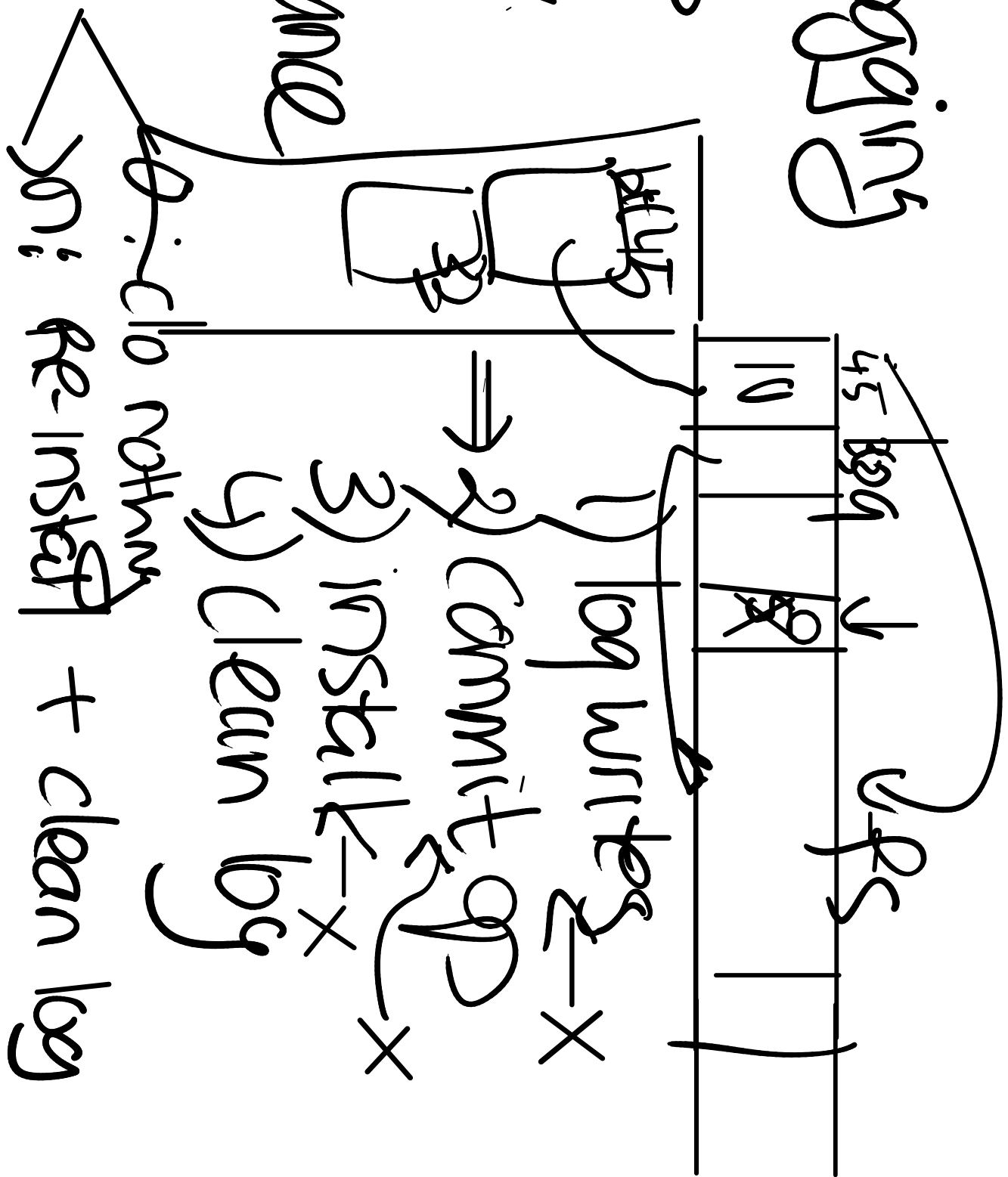
# Solution: Logging

atomic fs calls

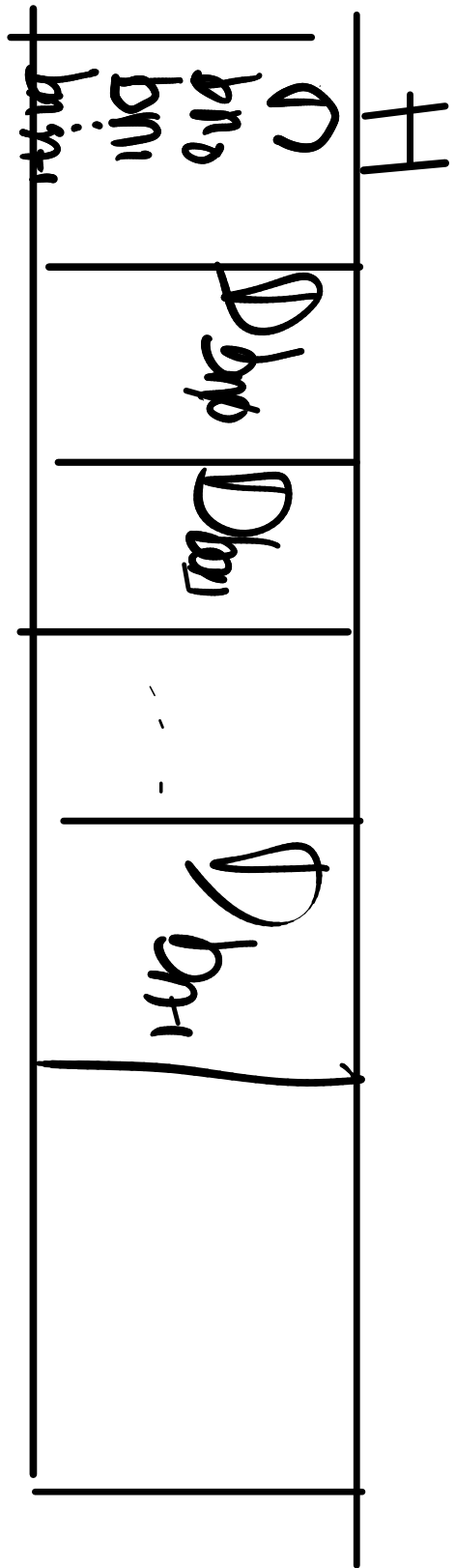
fast recovery

high performance

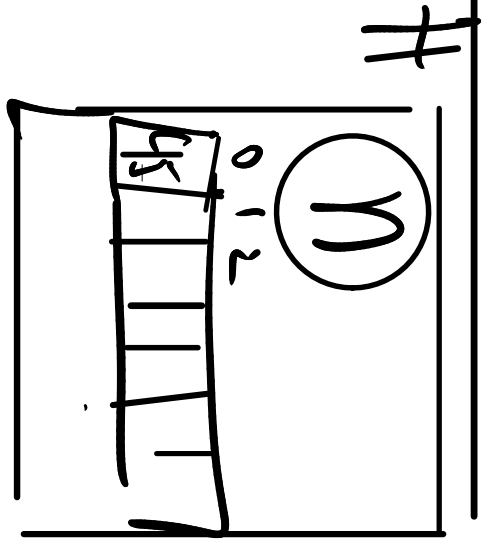
Reboot:



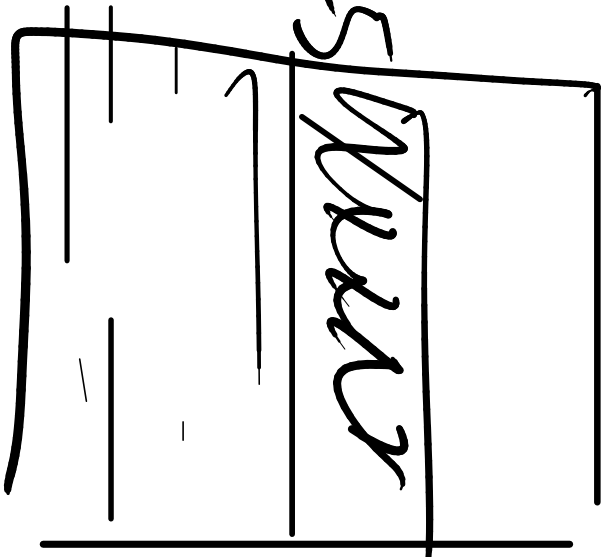
8  
disk



10  
pages



45  
NRRR



page.

Challenges: Eviction

Evict 45 + 45 more location.

? < bad: write-ahead  
rule violations

⇒ SG: don't evict blocks  
that are in log

Challenge \$s\$ op must fit in log.

max log size is 30

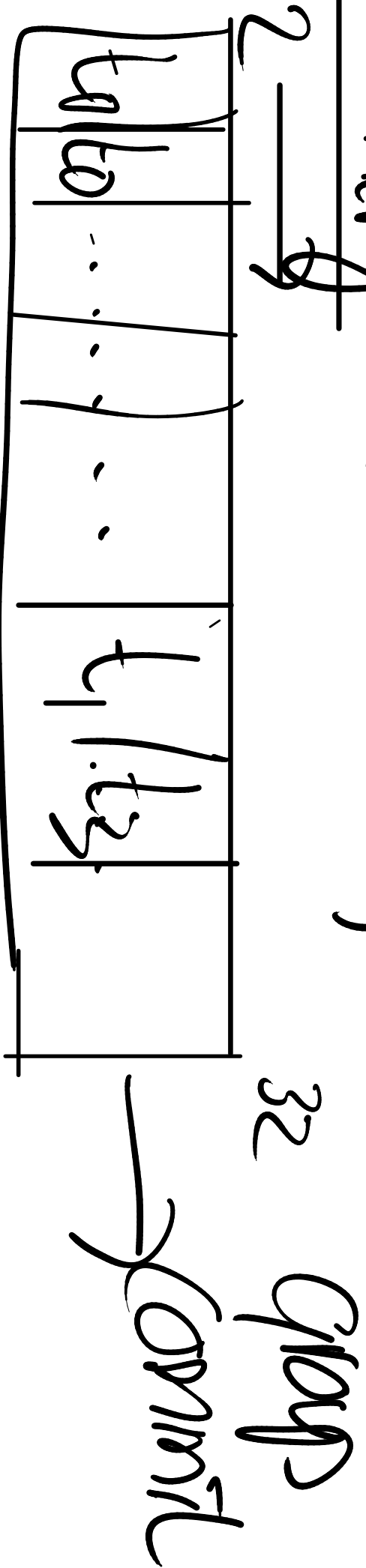
30) max # blocks a \$s\$ operation writes

write \$s \rightarrow\$ many transactions

→ cache \$\neq\$ log size.



Challenge: concurrent fs calls



⇒ all concurrent ops must file

⇒ limit #concurrent fs calls

# Summary

Logging for multi-step ops

BUT: performance?