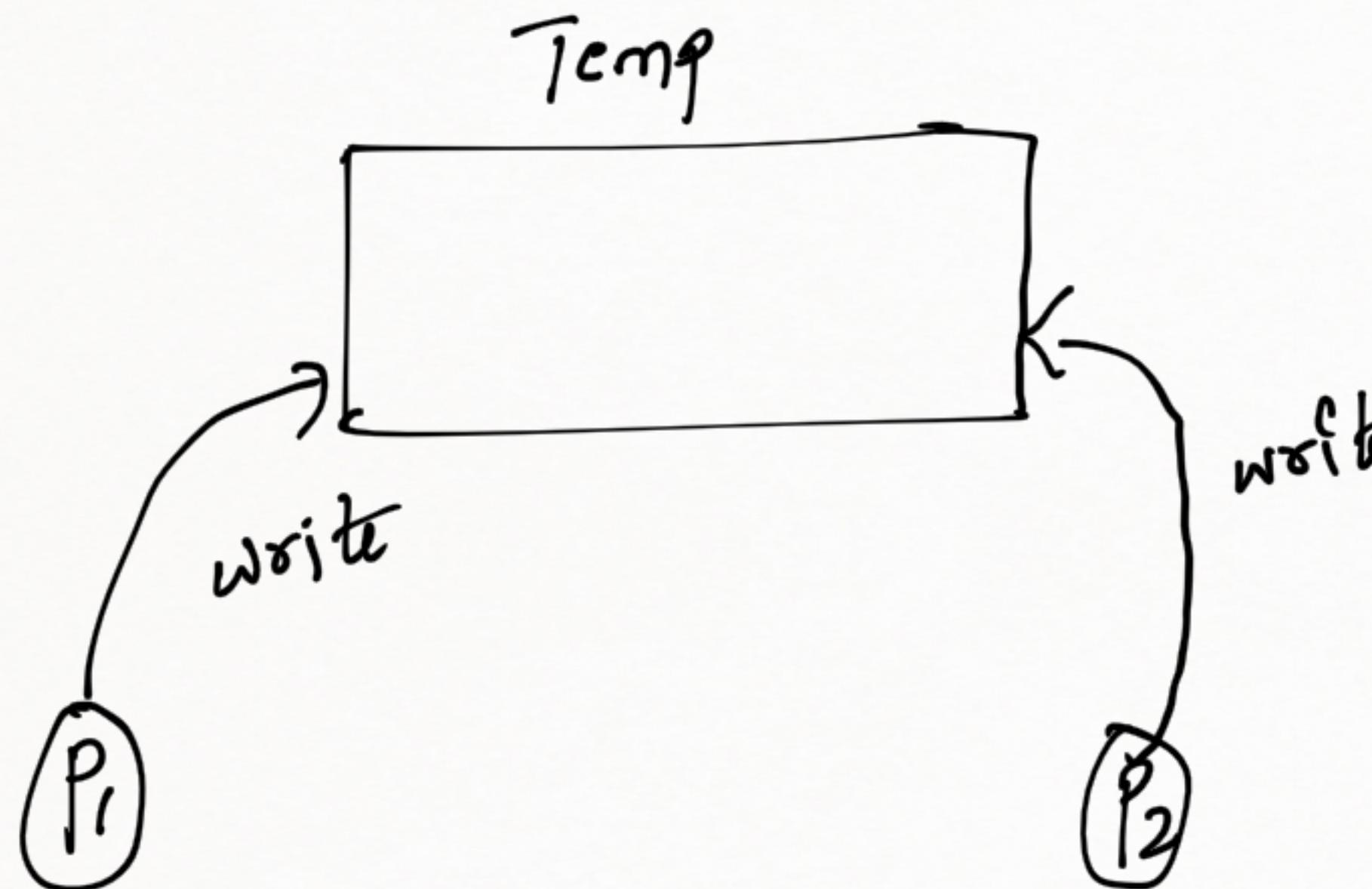
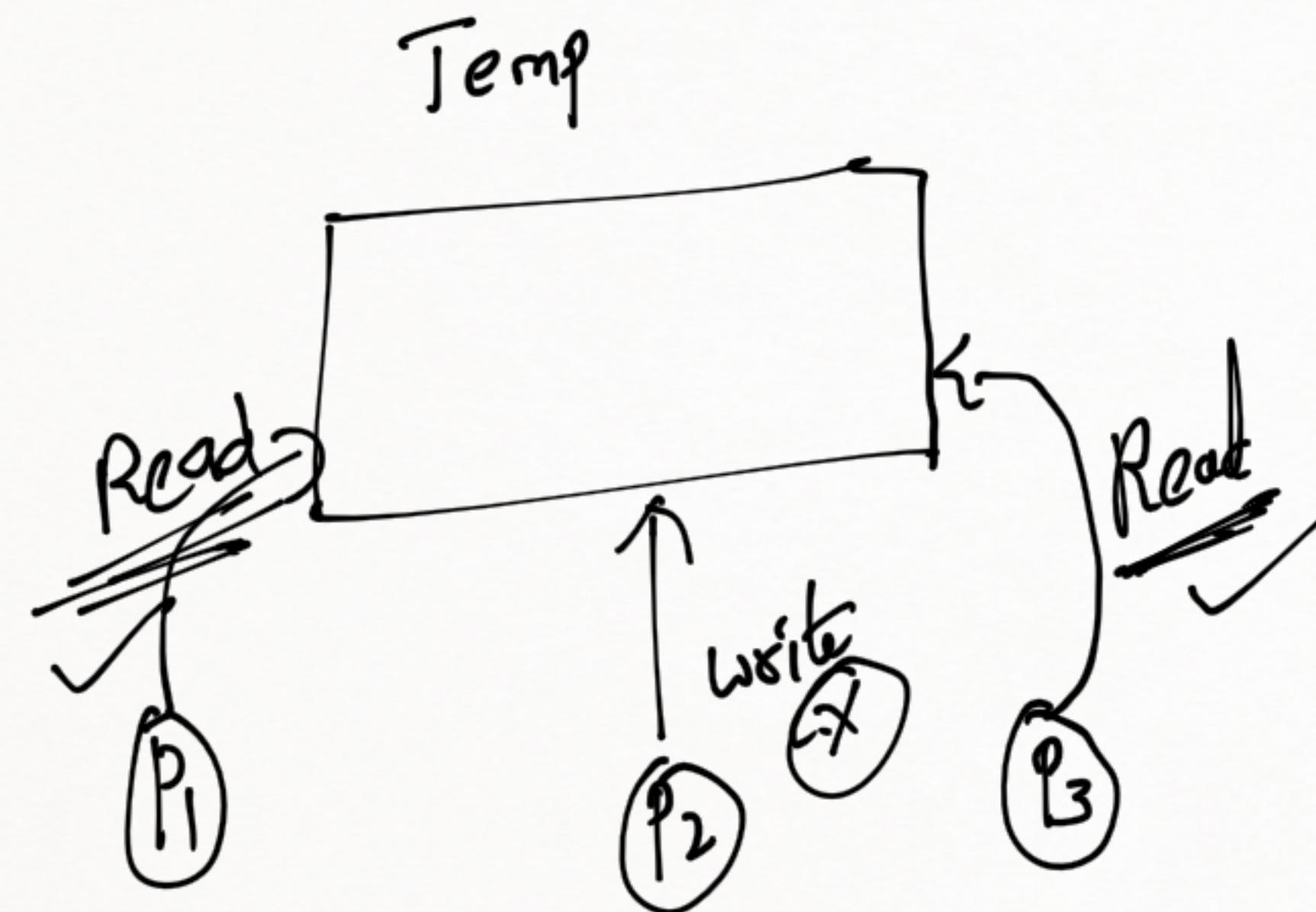


Date: 17-11-21

Advisory locking or Record locking



- Among P_1 & P_2 one of the process will apply write lock, if a process applies write no other process can apply write lock / Readlock.
- If one process writing data into a file (by applying write lock) no other process will allow to write (or) to read the data (write lock is exclusive lock)



- more than one process can read the data file by applying read lock
- no other process allowed to work the data
- read lock is shared lock

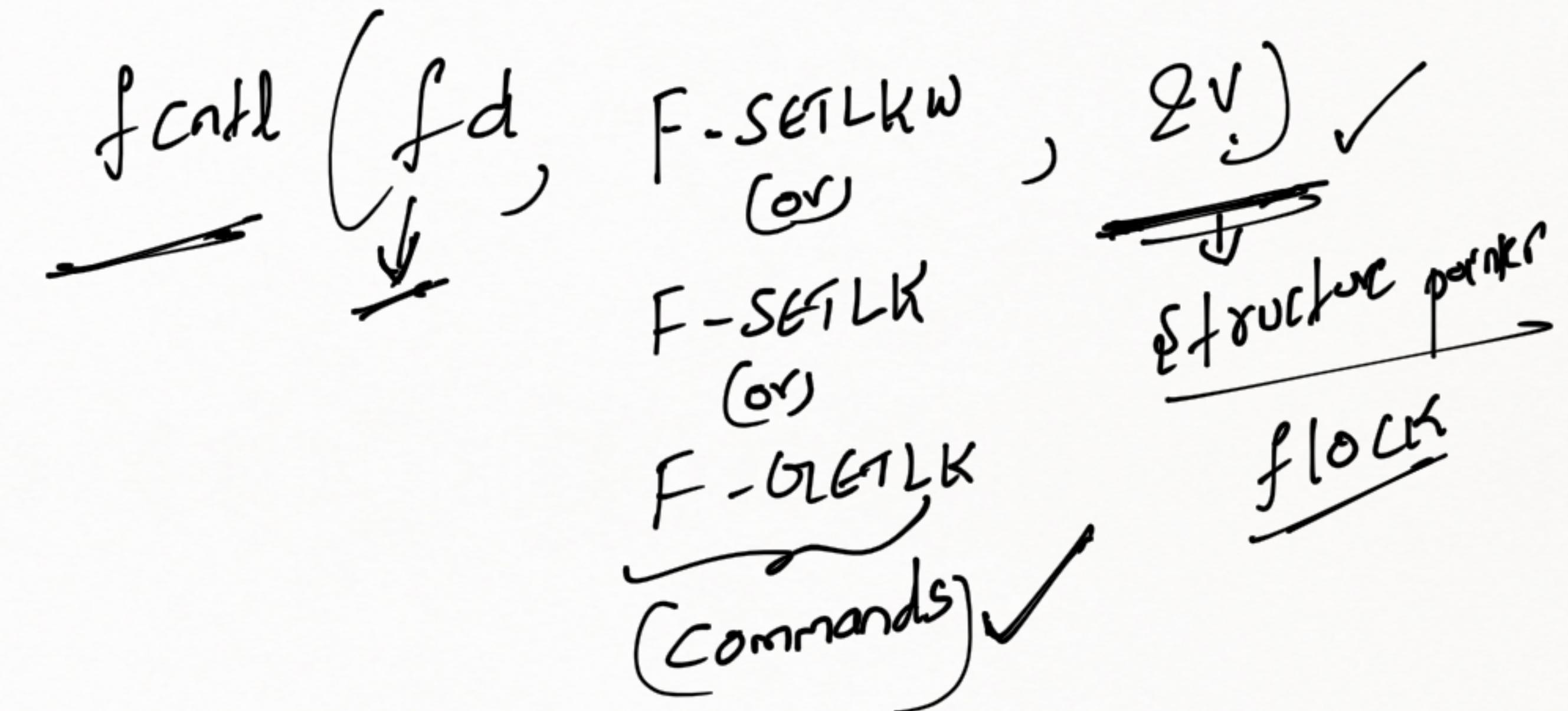
steps involved for

Record locking using fcntl

Stouch flock v;

locking mechanism are

- Apply to lock
- work the data
- release it lock



struct flock

{}

l-type \Rightarrow lock type write | Read | unlock

✓ short l-type:

✓ short l whence ✓

✓ off-t l-start; ✓

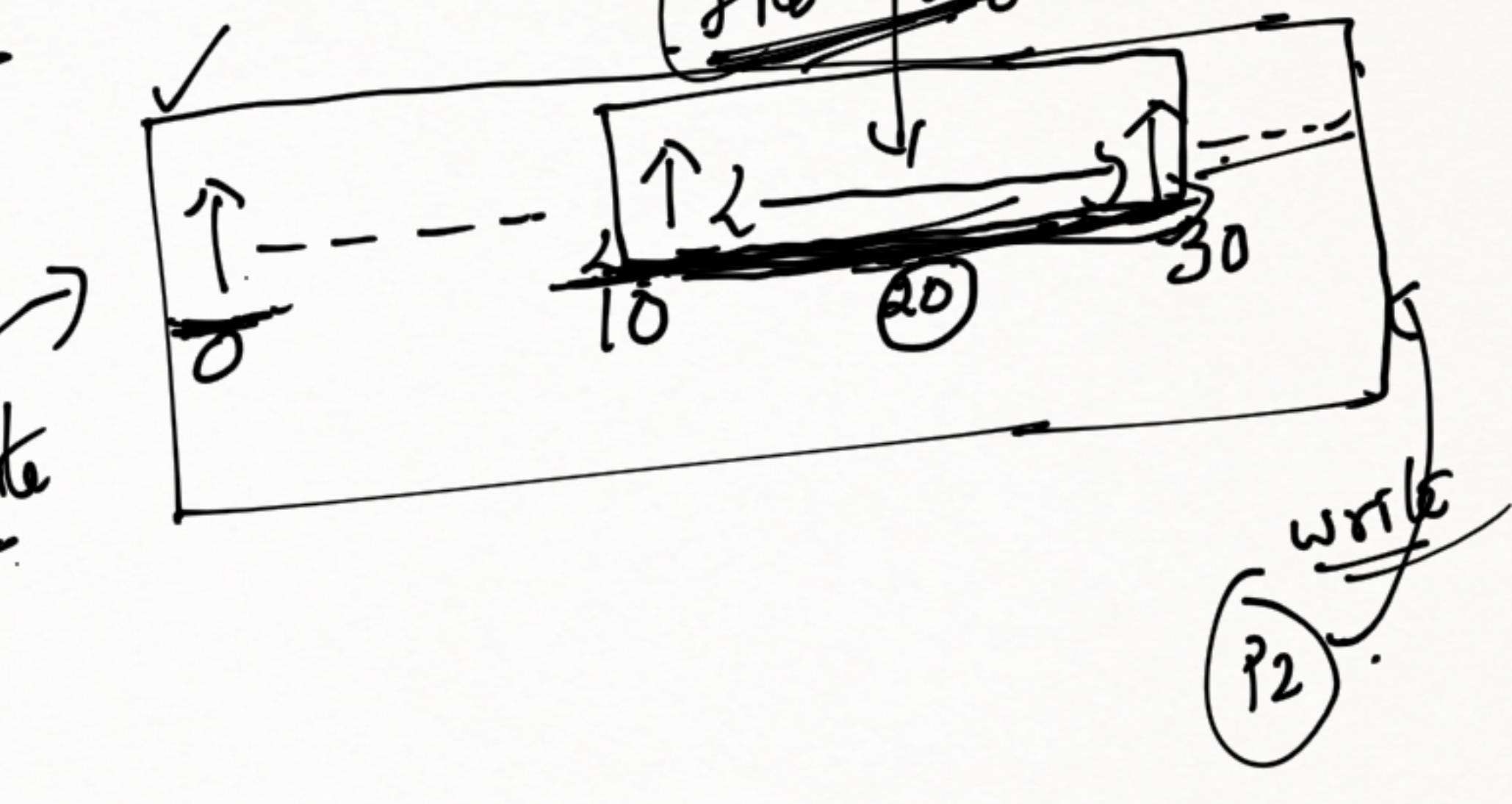
✓ off-t l-len;

✓ pfd-t l-pfd

struct flock v.

a. l-type = F-WRLCK; ✓
a. l-whence = SEEK-SCT (0) ✓
a. l-start = (10) ✓
a. l-Len = (20) ✓

specific file region lock
write



SEEK-SCT \Rightarrow 0

SEEK-CUR \Rightarrow 1

SEEK-END \Rightarrow 2

(file occupation)

complete file lock

Q. l-type = F-WRLCK

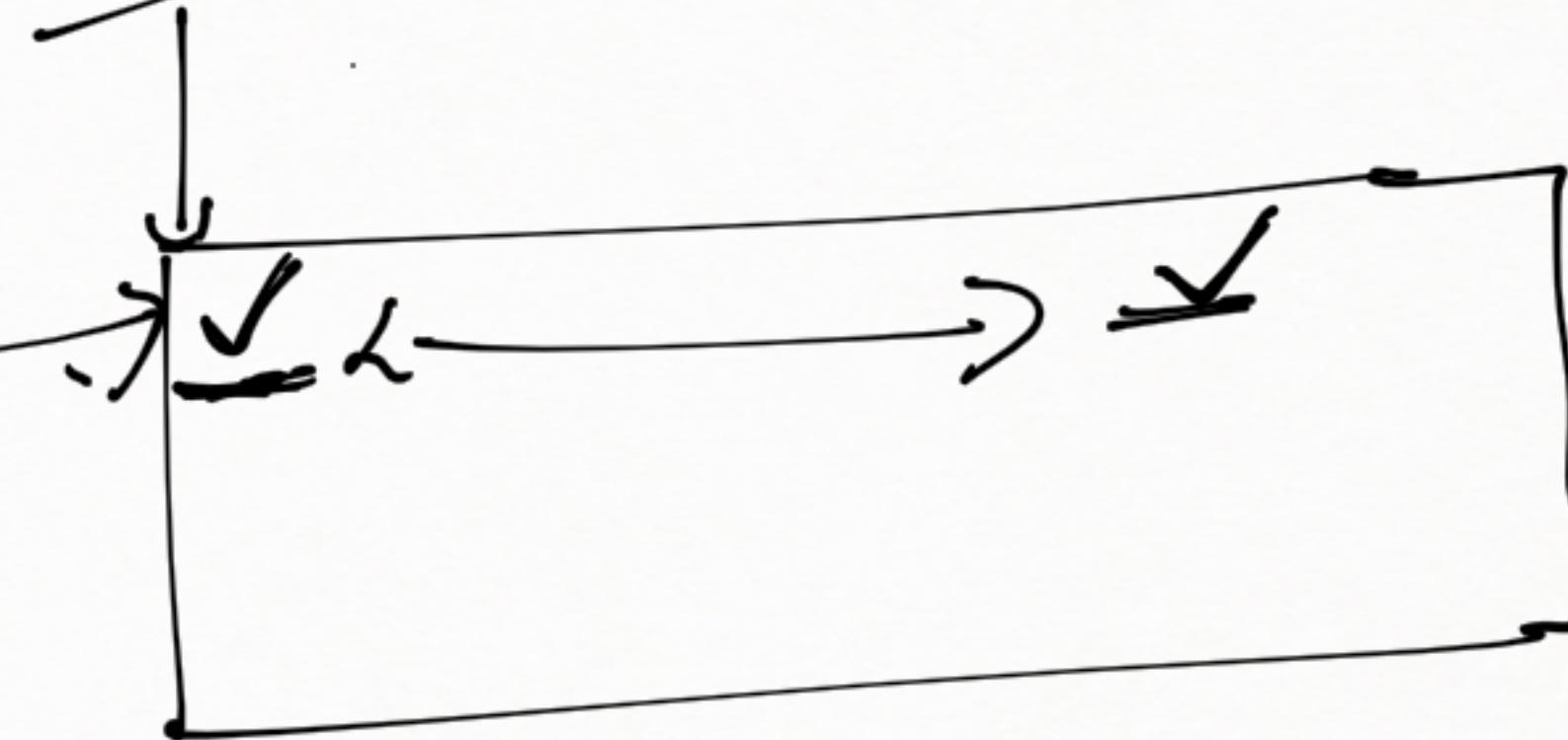
Q. l whence = 0;

Q. l-start = 0.

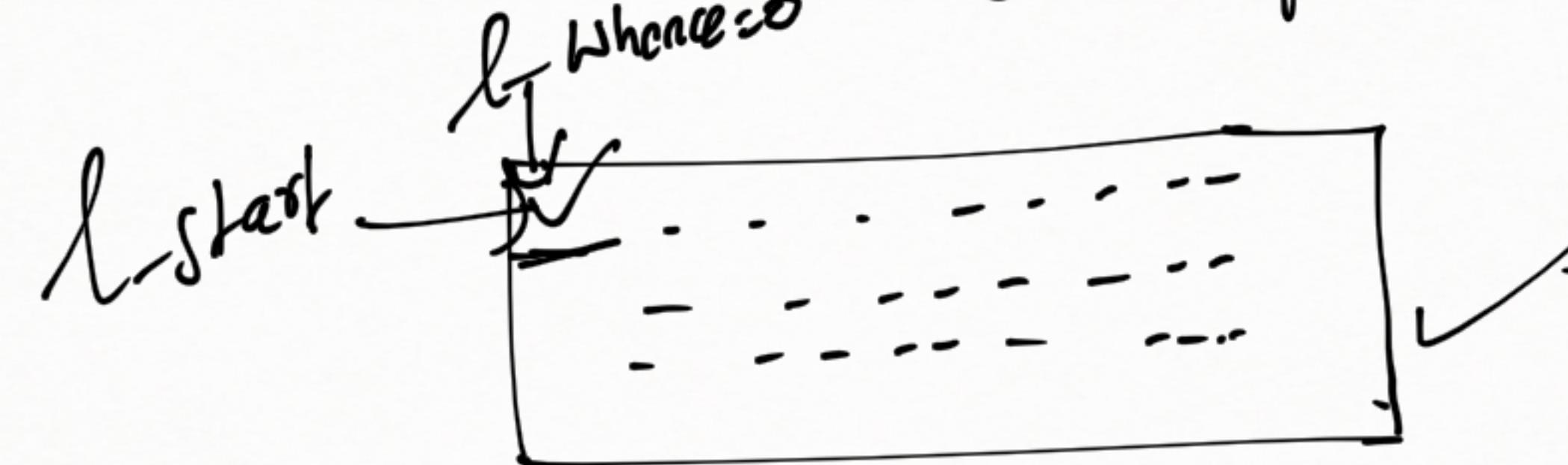
Q. l-len = 20/0

l whence = 0

l-start



is halving speed meaning that lock
Total file



→

P₁

q touch flock √;

v. l-type = F-WRLCK

v. l whence = 0;

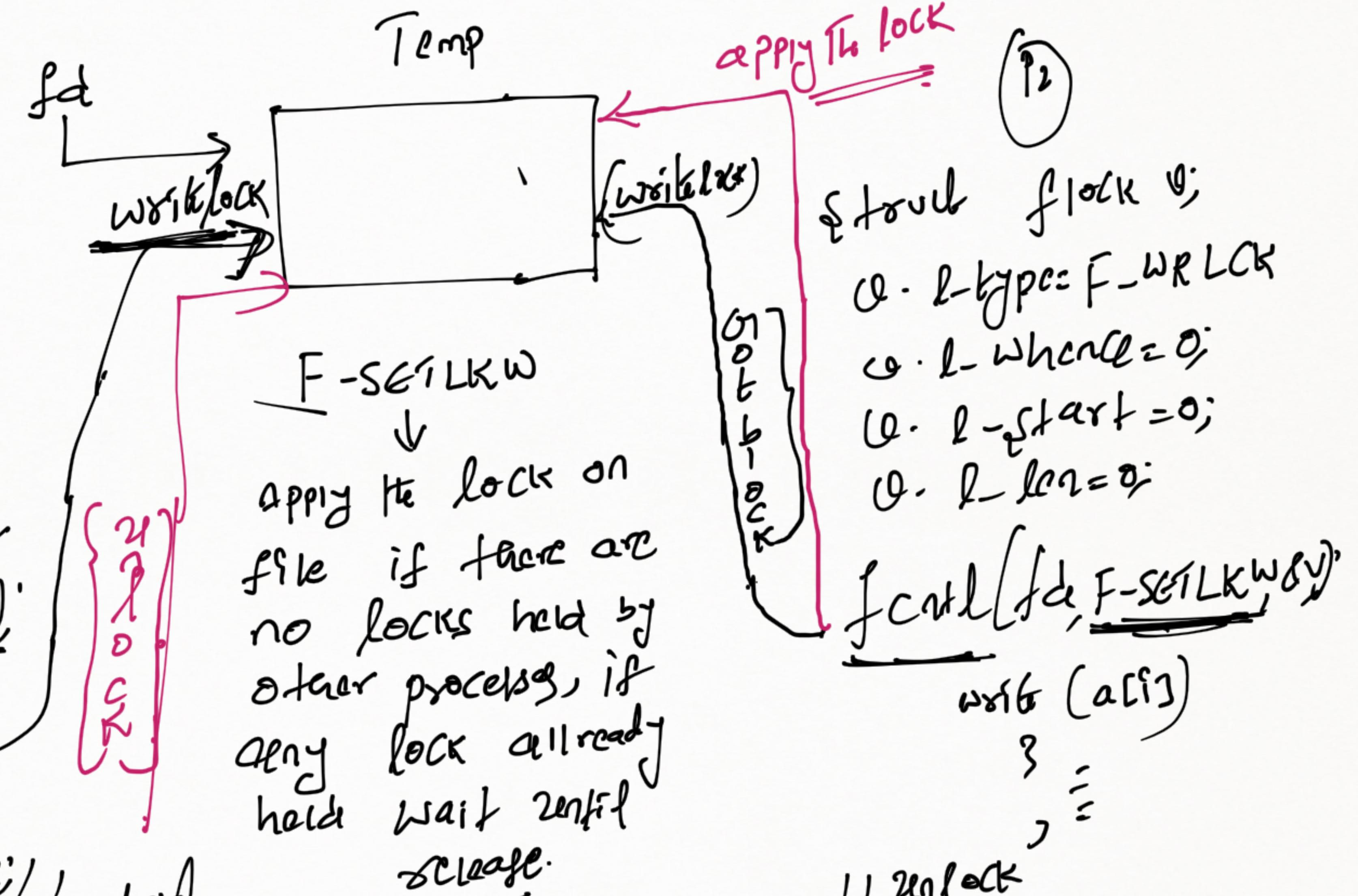
v. l start = 0;

v. l len = 0;

fctl(fd, F-SETLKW, &v)

while (a[i])
 3 // wait the data

fctl(fd, F-UNLCK, &v) // unlock



→ When lock released

- (i) If a process is terminated locks held by the process are automatically released
- (ii) By using l-type in structure flock
 - ↳ l-type = F_UNLOCK;
 - fcntl(fd, F_SETLK, &v);
- (iii) By closing the file descriptor
 - close(fd);
 - =