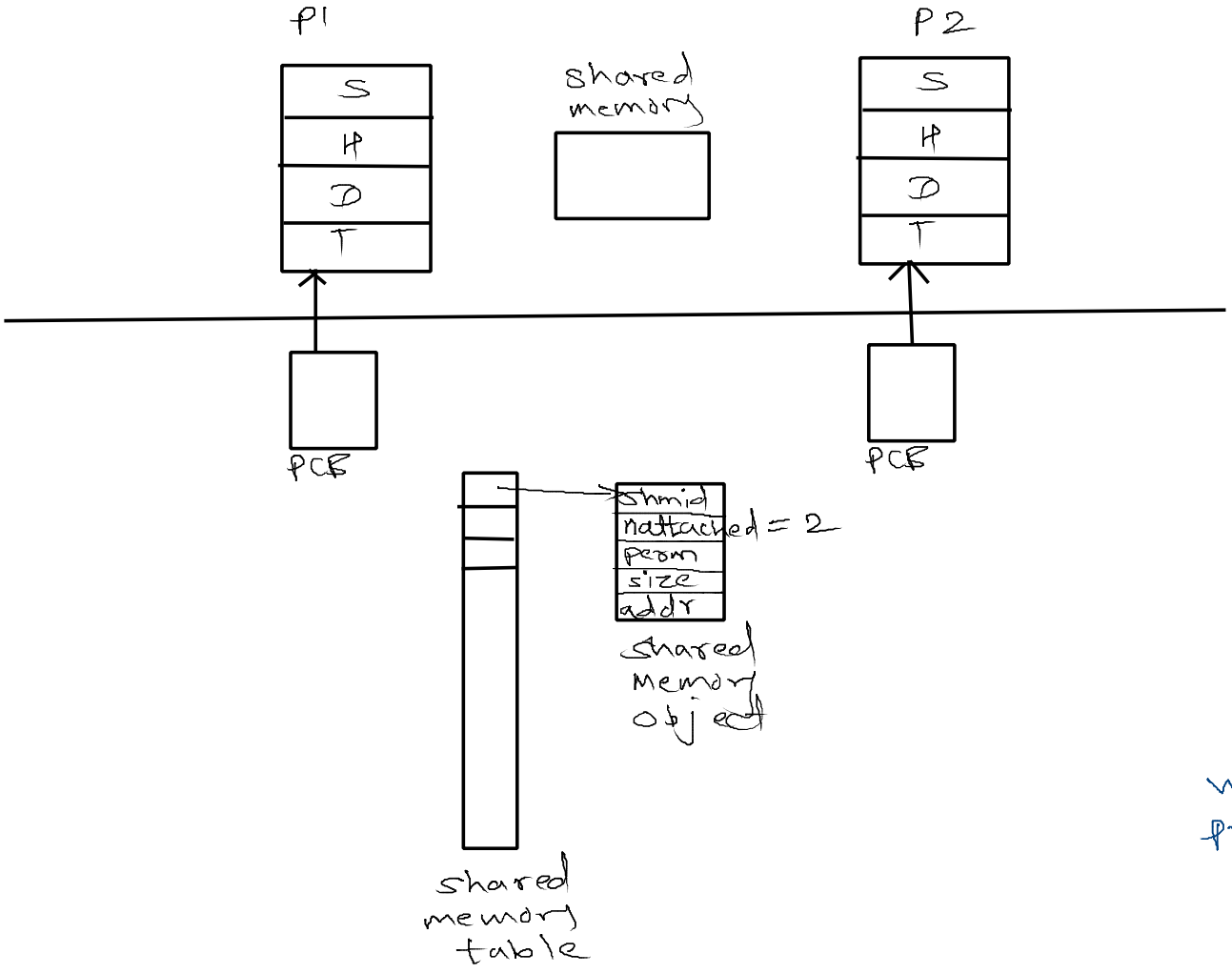
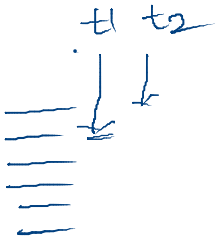
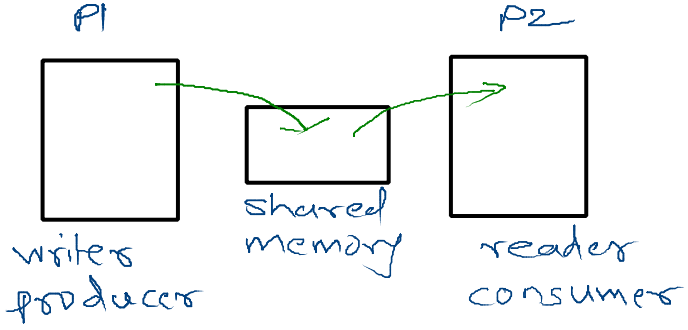
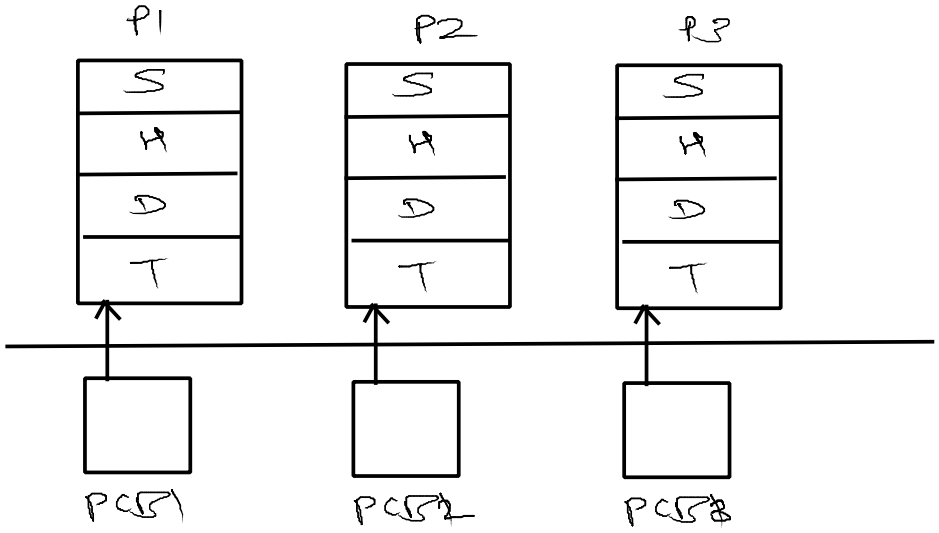
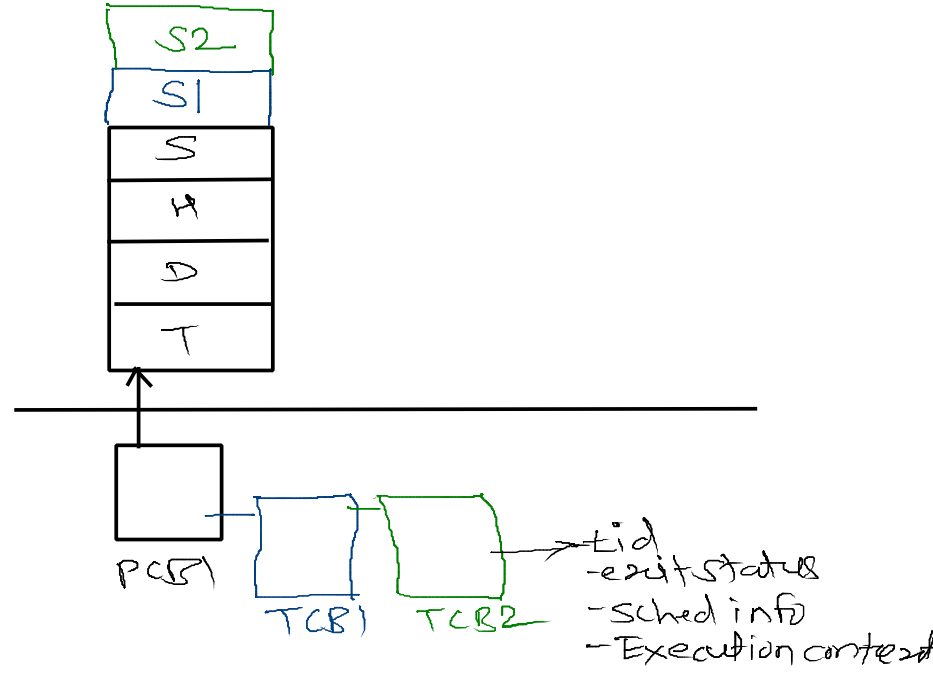

P1
S
H
D
T
shared memory
P2
S
H
D
T
PCB
PCB
shmid
nattached = 2
perm
size
addr
shared memory obj
shared memory table


t1
t2


P1
P2
writer producer
shared memory
reader consumer

Thread - Light weight process

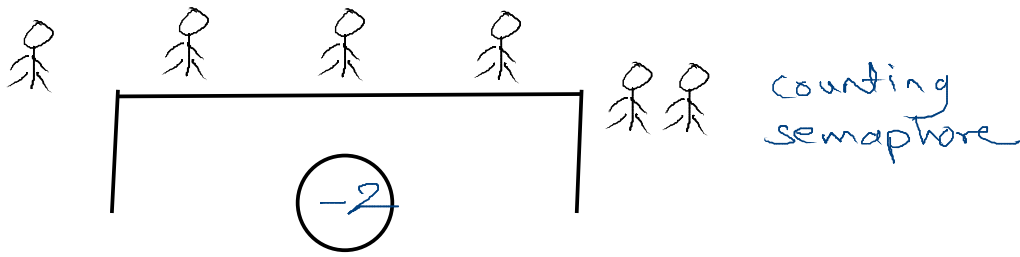




Semaphore

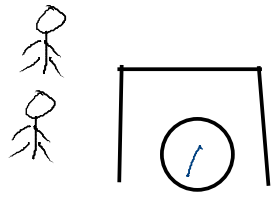
- internally it is a counter
-

- Dec / wait / P operations:
  - decrement count
  - if cnt < 0, block the current process
  - before using common resource



counting semaphore

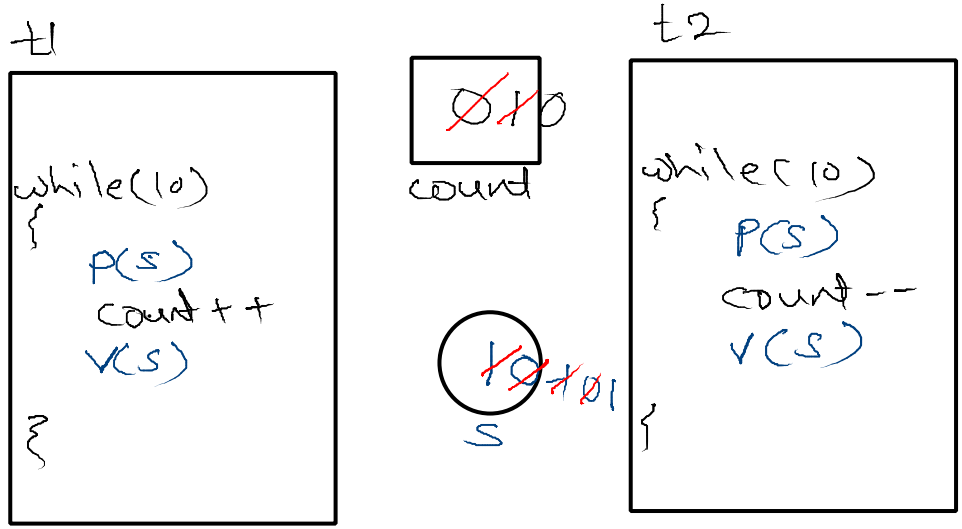
- INC / post / V operation:
  - increment count
  - if someone is blocked on that counter wakeup it.
  - after leaving common resource



Binary semaphore

Mutex (Mutual Exclusion)

- binary semaphore
- lock / unlock
- the process who lock the mutex will become owner of that mutex
- only owner can unlock mutex



P1

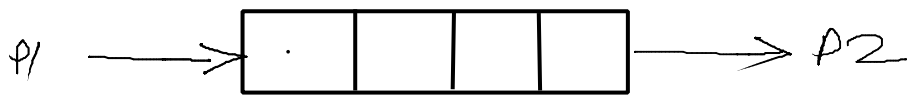
P(b)
P(e)

V(b)
V(f)

P2

P(b)
P(f)

V(b)
V(e)



P — full – counting semaphore
e — empty – counting semaphore
b — at a time one — binary semaphore