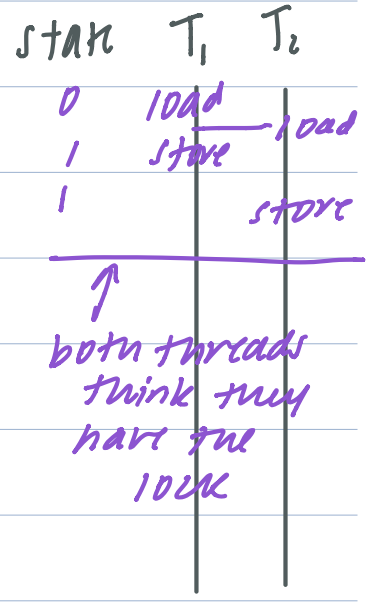
WRONG mutex implementation

```
std::atomic<int> stat = UNLOCKED;  //0
void lock() {
    while(stat.load() == LOCKED) { }
    stat.store(LOCKED);
}
void unlocked() {
    stat.store(UNLOCKED);
}
```

read, modify, write is not atomic

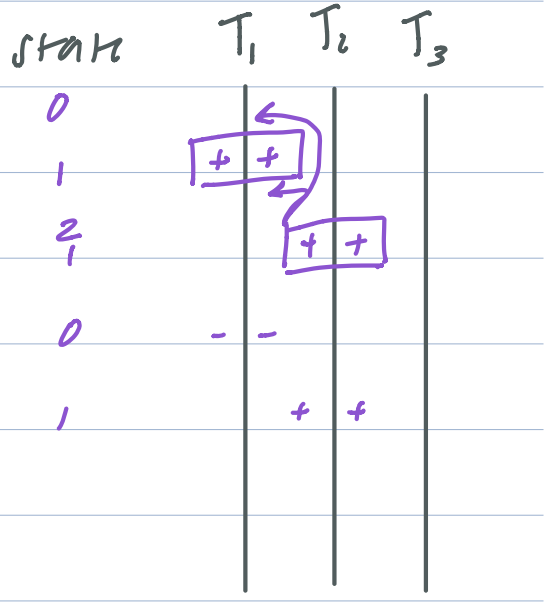


CORRECT mutex implementation

```
std::atomic<int> stat = UNLOCKED;
void lock() {
    while(--stat != LOCKED) {
        --stat;
    }
}
void unlock() {
    --stat;
}
```

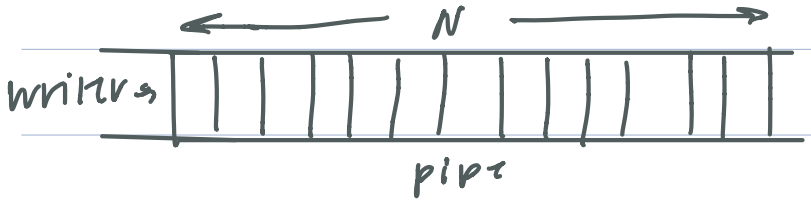
equivalent to

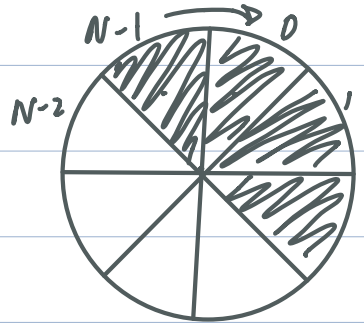
```
while(stat++ != UNLOCKED) {
    stat--;
}
```



Bounded Buffer

* want reader and writer to act independently





1. capacity
2. position of next byte to read
3. position of next byte to write

2 "try again" conditions:

1. partially work
2. didn't write anything

— SY

Condition variables

solve wake up / sleep problem

bbuf

bbuf.write("ABCDE", 5) → 4

bbuf.write("EFGH")

bbuf.write("IJKL")

bbuf.read(X, 3)

bbuf.read(X, 3)

bbuf.read(X, 3)