A **very bad** implementation of a lock

```
struct lock {
}

void acquire (lock) {
    disable_interrupts();
}

void release (lock) {
    enable_interrupts();
}
```

➡ Disabling interrupts blocks notification of external events that could trigger a context switch

- Can miss or delay important events
- The thread is no longer preemptive

Native Lock Implementation

```
struct lock {
    int held = 0;
    queue Q;
}

void acquire (lock) {
    disable_interrupts();
    while (lock->held){
        enqueue(lock->Q, current_thread);
        thread_block(current_thread);
    }
    lock->held = 1;
    enable_interrupts();
}

void release (lock) {
    disable_interrupts();
    if (!isEmpty(lock->Q)){
        thread_unblock(dequeue(lock->Q));
    }
    lock->held = 0;
    enable_interrupts();
}
```