```c
typedef struct __lock_t {
    int         flag;   // state of lock: 1=held, 0=free
    int         guard;  // use to protect flag, queue
    queue_t     *q;     // explicit queue of waiters
} lock_t;

void lock_init(lock_t *lock) {
    lock->flag = lock->guard = 0;
    lock->q    = queue_init();
}

void lock(lock_t *lock) {
    while (xchg(&lock->guard, 1) == 1)
        ; // spin
    if (lock->flag == 0) { // lock is free: grab it!
        lock->flag  = 1;
        lock->guard = 0;
    } else {               // lock not free: sleep
        queue_push(lock->q, gettid());
        lock->guard = 0;
        park();            // put self to sleep
    }
}

void unlock(lock_t *lock) {
    while (xchg(&lock->guard, 1) == 1)
        ; // spin
    if (queue_empty(lock->q))
        lock->flag = 0;
    else
        unpark(queue_pop(lock->q));
    lock->guard = 0;
}
```