SPTCC 2017
Exercises for the locking class (Nir Shavit)

Assignment 1. Design an isLocked() method that tests whether a thread is holding a lock (but does not acquire that lock). Give implementations for

- Any testAndSet() spin lock
- The CLH queue lock, and
- The MCS queue lock.

Assignment 2. Figure 1 shows an alternative implementation of CLHLock in which a thread reuses its own node instead of its predecessor node. Explain how this implementation can go wrong and how the MCS lock avoids the problem even though it also reuses thread-local nodes.

```
1  public class BadCLHLock implements Lock {
2          AtomicReference<Qnode> tail = new Qnode();
3          ThreadLocal<Qnode> myNode;
4          public void lock() {
5              Qnode qnode = myNode.get();
6              qnode.locked = true;
7              Qnode pred = tail.getAndSet(qnode);
8              while (pred.locked) {}
9          }
10         public void unlock() {
11             myNode.get().locked = false;
12         }
13 }
```

Figure 1: An incorrect attempt to implement a CLHLock