```java
public class CLHLock implements Lock {
  AtomicReference<QNode> tail;
  ThreadLocal<QNode> myPred;
  ThreadLocal<QNode> myNode;
  public CLHLock() {
    tail = new AtomicReference<QNode>(new QNode());
    myNode = new ThreadLocal<QNode>() {
      protected QNode initialValue() {
        return new QNode();
      }
    };
    myPred = new ThreadLocal<QNode>() {
      protected QNode initialValue() {
        return null;
      }
    };
  }
  public void lock() {
    QNode qnode = myNode.get();
    qnode.locked = true;
    QNode pred = tail.getAndSet(qnode);
    myPred.set(pred);
    while (pred.locked) {}
  }
  public void unlock() {
    QNode qnode = myNode.get();
    qnode.locked = false;
    myNode.set(myPred.get());
  }
  class QNode {
    volatile boolean locked = false;
  }
}
```

FIGURE 7.9 The CLHLock class.