```
public class ALock implements Lock {
  ThreadLocal<Integer> mySlotIndex = new ThreadLocal<Integer> (){
    protected Integer initialValue() {
      return 0;
    }
  };
  AtomicInteger tail;
  volatile boolean[] flag;
  int size;
  public ALock(int capacity) {
    size = capacity;
    tail = new AtomicInteger(0);
    flag = new boolean[capacity];
    flag[0] = true;
  }
  public void lock() {
    int slot = tail.getAndIncrement() % size;
    mySlotIndex.set(slot);
    while (!flag[slot]) {};
  }
  public void unlock() {
    int slot = mySlotIndex.get();
    flag[slot] = false;
    flag[(slot + 1) % size] = true;
  }
}
```

FIGURE 7.7 Array-based queue lock.