Class Discussion
Questions

- Why spin-wait instead of blocking?
- What is the downside of spin-waiting?
- Can you hold on to the CPU while waiting without causing contention for other resources?
  - If so, how do you know when to proceed?
Questions

- Is efficiency harder than correctness for spin locks?
- Which aspects of efficiency are important here?
- Is it important to optimize for the high-contention case?
- Why might you want to prioritize the “clear” requests over the “test and set” requests?
Questions

- What is the key idea behind “test and test and set”?
  - Why doesn’t it solve the problem in practice?
  - Why isn’t it very helpful to have a solution that works well only for long critical sections?
  - Is the performance really any better than “test and set”?
Questions

- How can a waiting process notice that a lock has been released?
- If it polls how can we avoid having all waiting processes ...
  - poll at the same time?
  - poll at the same place?
Questions

- If we use a shared counter ...
  - How can we update it?
- If we use a queue mechanism ...
  - how can we implement the enqueue and dequeue operations?
- If we use a ticketing mechanism ...
  - Does the hardware provide an atomic read and increment?
    - If not, how can we implement it?
  - How do we implement the signaling (ticket update) on different processor architectures?