Announcements

- Office hours
  - W office hour will be 10-11 not 11-12 starting this week
- Midterm is next Tuesday
  - Covers through lecture on Thursday
- Project #2 is available on the web

Using Test and Test for Mutual Exclusion

```
repeat
        while test-and-set(lock);
        // critical section
        lock = false;
        // non-critical section
until false;
```

Note: no priority based on wait time

- bounded waiting time version

```
repeat
        waiting[i] = true;
        key = true;
        while waiting[i] and key
            key = test-and-set(lock);
        waiting[i] = false;
        // critical section
        j = (i + 1) % n
        while (j != i) and (!waiting[j])
            j = (j + 1) % n;
        if (j == i)
            lock = false;
        else
            waiting[j] = false;
        // non-critical section
until false;
```

wait until released or no one busy

look for a waiting process

no process waiting

release process j

Semaphores

- getting critical section problem correct is difficult
  - harder to generalize to other synchronization problems
  - Alternative is semaphores
- semaphores
  - integer variable
  - only access is through atomic operations
- P (or wait)

  while s <= 0;

  s = s - 1;
- V (or signal)

  s = s + 1
- Two types of Semaphores
  - Counting (values range from 0 to n)
  - Binary (values range from 0 to 1)