Problem Set 3
Due Sunday February 12, 11:59pm

1. What does it mean to turn interrupts off? Why is it important to turn off interrupts in the process subsystem code? Before a spinlock is acquired?

2. What happens when a process wakes up another process? How does a sleeping process get to run again?

3. Why does an xv6 process need two stacks?

4. The latest xv6 distribution implements blocking locks (acquiresleep). When would a blocking lock be more appropriate than a spinlock, and vice versa?

5. List the set of spinlocks in xv6 which CANNOT be converted to blocking locks, and explain why.

6. Add blocking lock condition variables to xv6. The blocking lock condition variable passes a pointer to the blocking lock to release.

7. Add support for kernel threads in xv6. A kernel thread is an asynchronous procedure call within the kernel. You may assume kernel threads exit but are never killed (they don't run user code). Hint: Linux reuses the process control block for kernel threads, so the primary work here is to be able to start a process that only runs in the kernel.

8.Write code for a laptop, desktop, or server machine to determine the cache coherence miss penalty between each pair of processors. For this you may want to pin each thread to a processor before running the test.