

1. Build a simple concurrent counter and measure how long it takes to increment the counter many times as the number of threads increases. How many CPUs are available on the system you are using? Does this number impact the performance?
2. Build a version of the approximate counter. Measure its performance as a function of the number of threads, as well as the threshold.
3. Build the version of a concurrent linked list described in Section 29.2 of <https://pages.cs.wisc.edu/~remzi/OSTEP/>. Write program(s) to compare the performance of regular and concurrent versions of linked list.
4. Build the version of a concurrent queue described in Section 29.3 of <https://pages.cs.wisc.edu/~remzi/OSTEP/>. Write program(s) to compare the performance of regular and concurrent versions of queue.
5. Implement ticket lock. Write program(s) to demonstrate its working.
6. Xv6 always disables interrupts for `spinlocks`. This is a conservative approach. It is sufficient to ensure that if a spinlock is used by an interrupt handler, a CPU must never hold that lock with interrupts enabled. Use this idea to improve performance of `spinlocks` in xv6.