

Homework 7

Professor: Maurice Herlihy

TA: James Scherick

Problem 1. The `ReentrantReadWriteLock` class provided by the `java.util.concurrent.locks` package does not allow a thread holding the lock in read mode to upgrade its lock to write mode (if it tries, the reader thread will block). Justify this design decision by sketching what it would take to permit such lock upgrades.

Problem 2. Reimplement the `SimpleReadWriteLock` class using Java **synchronized**, `wait()`, `notify()`, and `notifyAll()` constructs in place of explicit locks and conditions.

Hint: you must figure out how methods of the inner read-write lock classes can lock the outer `SimpleReadWriteLock` object.

Problem 3. For the `LockFreeHashSet`, design a lock-free data structure to replace the fixed-size bucket array. Your data structure should allow an arbitrary number of buckets.

Problem 4. Modify the `StripedHashSet` to allow resizing of the range lock array using read/write locks.