1.3 Lecture Summary

1.3 Unstructured Locks

Lecture Summary: In this lecture, we introduced *unstructured locks* (which can be obtained in Java by creating instances of `ReentrantLock`()), and used three examples to demonstrate their generality relative to structured locks. The first example showed how explicit `lock`() and `unlock`() operations on unstructured locks can be used to support a *hand-over-hand* locking pattern that implements a non-nested pairing of lock/unlock operations which cannot be achieved with synchronized statements/methods. The second example showed how the `tryLock`() operations in unstructured locks can enable a thread to check the availability of a lock, and thereby acquire it if it is available or do something else if it is not. The third example illustrated the value of *read-write locks* (which can be obtained in Java by creating instances of `ReentrantReadWriteLock`()), whereby multiple threads are permitted to acquire a lock `L` in “read mode”, `L.readLock`().`lock`(), but only one thread is permitted to acquire the lock in “write mode”, `L.writeLock`().`lock`().

However, it is also important to remember that the generality and power of unstructured locks is accompanied by an extra responsibility on the part of the programmer, e.g., ensuring that calls to `unlock`() are not forgotten, even in the presence of exceptions.

Optional Reading:

1. Tutorial on Lock Objects in Java

2. Documentation on Java’s Lock interfaces