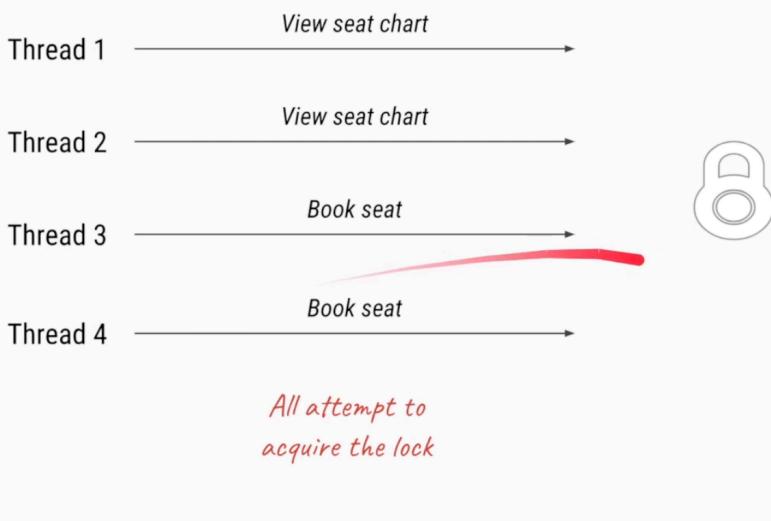


# Multithreading 9 : ReadWriteLock vs ReentrantLock

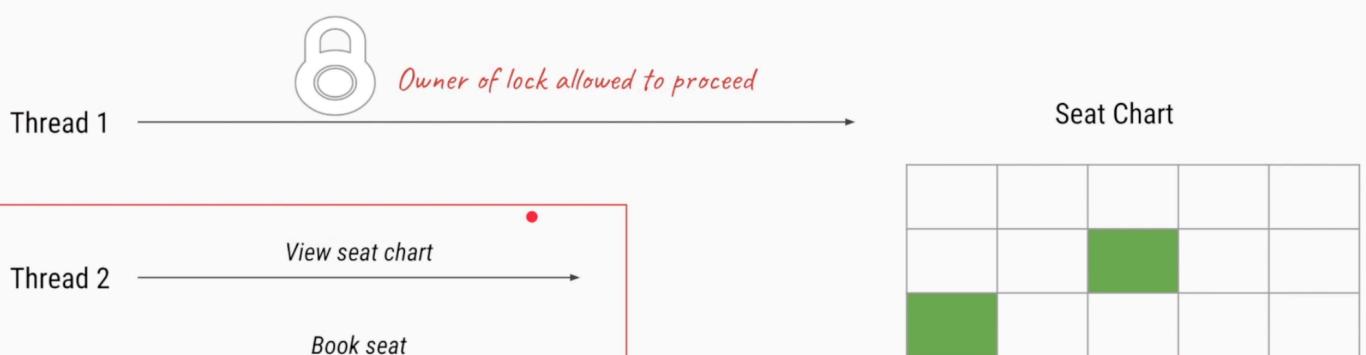
## ReadWriteLock Use Case:

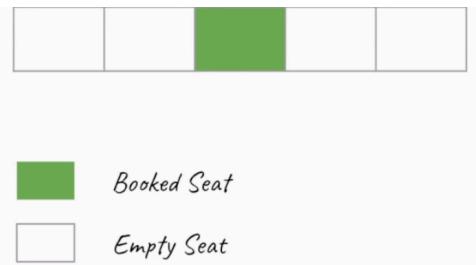
Use case updated - Allow viewing seat chart



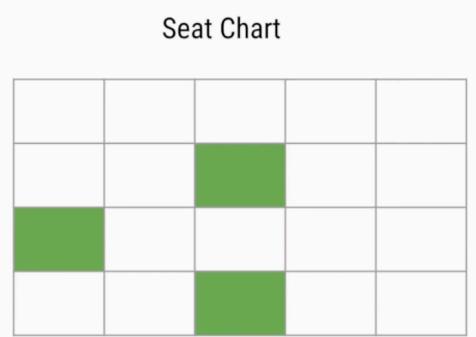
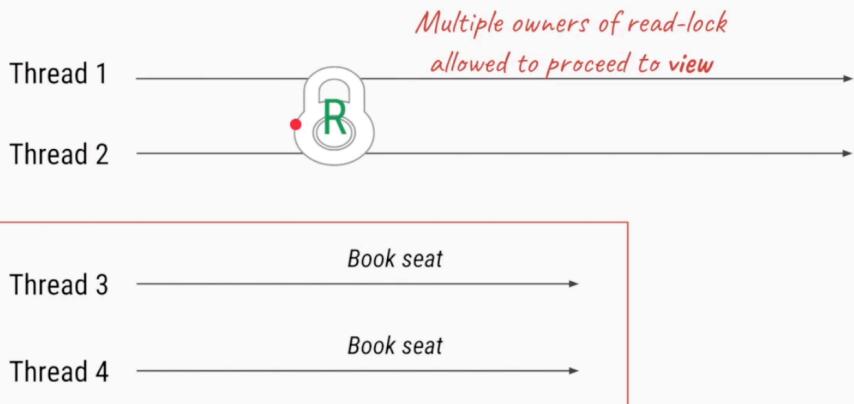
Conceptual example

Inefficient





## More efficient



*Threads which wanted to write/update go into wait state*

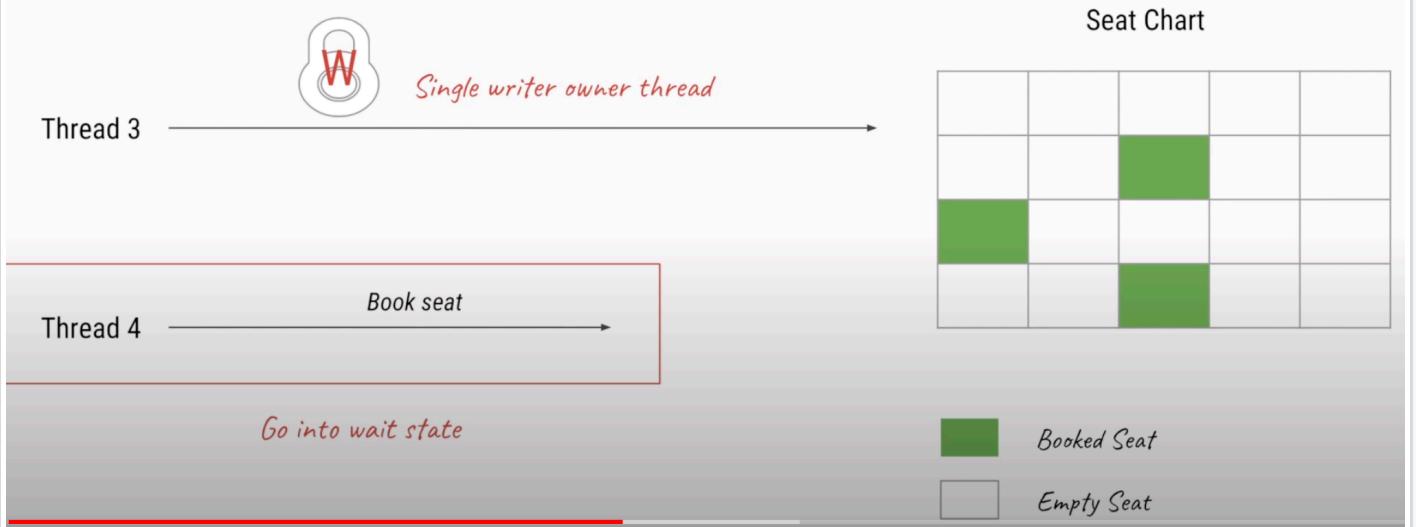


## Multiple reader threads at a time





Only 1 writer thread at a time



Only 1 writer thread at a time



## ReentrantLock

One thread at a time

## ReadWriteLock

One writer thread at a time  
OR  
Multiple reader threads at a time

```

private ReentrantReadWriteLock lock = new ReentrantReadWriteLock();

private ReentrantReadWriteLock.ReadLock readLock = lock.readLock();
private ReentrantReadWriteLock.WriteLock writeLock = lock.writeLock();

private void readResource() {
    readLock.lock();
    // view the resource
    readLock.unlock();
}

private void writeResource() {
    writeLock.lock();
    // update the resource
    writeLock.unlock();
}

public static void main(String[] args) {
    Thread t1 = new Thread(() -> obj.readResource()); t1.start();
    Thread t2 = new Thread(() -> obj.readResource()); t2.start();
}
  
```

```
Thread t2 = new Thread(() -> obj.readResource(), t2.start());
Thread t3 = new Thread(() -> obj.writeResource()); t3.start();
Thread t4 = new Thread(() -> obj.writeResource()); t4.start();
```

## Note

ReadLock and WriteLock though are 2 separate instances  
only 1 will be allowed at a time...

Either  
ReadLock is being used (by n threads)  
OR  
WriteLock is being used (by 1 thread)

But never both at same time

## Wait queue in ReentrantLock

*Current thread owner*

Thread 1



Thread 2

Thread 3

Thread 4

Wait Queue

*All threads waiting for the lock*

## What about wait queues in ReadWriteLock?

*Current writelock thread owner*

Thread 1



Thread 2 (R)

Thread 3 (W)

Thread 4 (R)

Wait Queue

*All threads waiting for the lock*

## What about wait queues in ReadWriteLock?

*New Lock owners*

Thread 2

Thread 4



Thread 3 (W)

*All threads waiting for the lock*

Wait Queue

## What about wait queues in ReadWriteLock?

New Lock owners

Thread 2
Thread 4
Thread 5



Possibility that writer thread keeps waiting in face of incoming new reader threads, thus this is not allowed

Thread 3 (W)			
--------------	--	--	--

Wait Queue

All threads waiting for the lock

It dependent on implementation on ReadWriteLock that it is allow to access Thread 5 before giving access Write access thread Thread 3.

## What about wait queues in ReadWriteLock?

New Lock owner

Thread 3
----------



Thread 5 (R)			
--------------	--	--	--

Wait Queue

All threads waiting for the lock

As per write implementation of Retentrant-write lock it allow thread 3 (W) first then only provide access for thread 5(R)

## Difference

ReentrantLock	ReadWriteLock
Single lock per lock instance	Separate read & write locks per lock instance
	Allows write lock owner thread to also acquire read lock
newCondition	Only write lock allows newCondition
Good for general purpose locking	Good for frequent reads and infrequent writes
	Can be more performant for right use-cases