A simple story of lock

2016-09-09

Question:

There is a resource need to be accessed by multiple readers and one writer.

Do we need use locks to control the accessing of the readers and the writer?

Question:

There is a resource need to be accessed by multiple readers and one writer.

Do we need use locks to control the accessing of the readers and the writer?

Need or do not need?

Quick answer:

Do not need.

- Only one writer.
- Speed up readers.

But, really?

First of all, WHY locks?

Please keep these in mind if you are going to use a lock.

- Atomicity
- Visibility
- Ordering

Rethinking

Do not need locks.

- We don’t care atomicity. (One writer)
- We don’t care visibility. (??)
- We don’t care ordering. (??)

Visibility

Visibility

- No locks.
  - Readers can access the resource during the writer is modifying the resource. (This might be a problem)

A good idea: reading after writing. (Ordering!)

Ordering

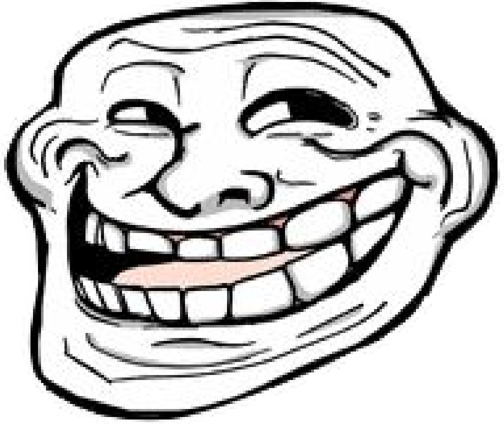
Ordering

Happened-before. (denoted: →)

“A relation between the result of two events, such that if one event should happen before another event, the result must reflect that, even if those events are in reality executed out of order (usually to optimize program flow).”

--wikipedia

It depens!



References:

http://tutorials.jenkov.com/java-concurrency/volatile.html

http://cs.nyu.edu/~lerner/spring12/Read04-ReadersWriters.pdf