Exam topics

May 20, 2016

1. Inter-thread visibility
   - visibility problem, the causes of the problem (memory/CPU)
   - JMM: happens-before, visibility guarantees
   - double-checked locking
   - safe publication

2. Inter-thread synchronization
   - atomicity and consistency; locks: optimistic/pessimistic, critical sections, read-write lock
   - locking patterns: fully synchronized, client-side, copy-on-iteration, copy-on-write, pros and cons
   - deadlock, solution for deadlock, refining locks
   - Java monitors: reason, common pattern, condition variables
   - Lock mechanics and price, standard synchronizers

3. Thread management
   - thread pros and cons, threading strategies
   - Executor framework, standard executors, Callable and Future interfaces, completion service
   - Cancellation reasons, thread interruption, non-interruptible blocking, executor shutdown
   - thread usage anti-patterns

4. Inter-thread communication
   - confinements: method, thread, object, group
   - total thread confinement, producer/consumer/transducer
   - deadlock free execution, problems with queues
   - back-pressure techniques, synchronous queue
   - aggregating queues, priority queues, delayed queues, paced execution
   - channels vs streams, sources of non-determinism, declarative concurrency???
5. Inter-thread collaboration
   - passive/active objects, actor model, example
   - message handling, state/message centric, methods as messages
   - message guards and priorities, postponed delivery and timeout
   - reasoning with actors, actor diagrams, state diagrams
   - actor with extra thread, actor supervision

6. Asynchronous execution
   - synchronous vs asynchronous, problems of synchronous
   - callback solutions: Pyramid of Doom, named callbacks, problems with each
   - promises, promise chaining with apply/compose/...
   - promise executors, promises within actors

7. Performance considerations
   - context switch problem, solutions to context switch, green threads
   - blocking IO problem, asynchronous IO

8. Erlang agents
   - declarative core, processes and mailboxes, receive
   - fault tolerance: system processes, links, exit signals

9. Clojure declarative core
   - what is declarativeness, why important, in FP
   - basics of DP: iterative and recursive computation

10. Immutable data structures
    - general idea, benefits, example with a binary tree (no need for sorted trees or RB trees), updating immutable state
    - immutable queue

11. Software transactional memory (STM)
    - ACID properties and isolation levels
    - Refs, STM in Clojure (how it works in general)
    - Ref history queue, write-skew problem, livelock problem

12. Clojure agents
    - General (how Clojure agents relate to other reference types), agent mechanics
    - Clojure agents with IO, STM and nested sends, watches in Clojure