

Topics for the Final

- Reader-Writer Locks (Lecture 12, 16)
 - Properties of the reader lock
 - Properties of the writer lock
 - Lock upgrading
 - Implementation with wait-notify
- Lists (Lecture 12, 14, 17)
 - Concurrent add/remove/contains/iterate
 - Coarse-grained
 - Fine-grained hand-over-hand locking
 - Optimistic locking
 - Lazy list
 - Logical/physical remove
 - Validation
 - Non-blocking
 - CAS, limitations
 - Marked reference
 -
- Wait-notify (Lecture 16)
 - Semantics of wait
 - Semantics of notify
 - Relation with locks
 - Synchronized
 - Conditions
 - Illegal Monitor Exception
 - Lost wakeups
 - What are they
 - How to prevent
 - Interrupted Exception
- Task scheduling (Lecture 16, 17, 19)
 - Threads vs tasks
 - Thread pools
 - Asynchronous execution in Java: ExecutorService, Future, Task
 - Scheduling: Central queue, Work dealing, Work stealing, Work balancing
- Consensus (Lecture 19)
 - Semantics of the consensus problem
 - Consensus number: Registers, RMW, CAS, LL/SC
 - Consensus hierarchy
- Non-blocking progress (Lecture 19, 20)
 - Lock-free vs Wait-free
 - Relation to blocking progress

- Universal Lock-free construction
 - Interface
 - Relationship with consensus objects
- Universal Wait-free construction
 - Helping, announce array
- Queues (Lecture 17, 20)
 - Blocking Bounded queue with a single lock and multiple wait conditions
 - Blocking Bounded queue with enq and deq locks
 - Non-blocking unbounded queue
- ABA problem (Lecture 20)
 - What is it
 - Examples
 - How to prevent it
 - Tagged reference, double CAS, DWCAS
- Stacks (Lecture 21)
 - Non-blocking stack based on CAS
 - Back-off through elimination array
- Hashing (Lecture 22, 23)
 - Hash table, buckets, nodes, hash function, collisions, load-factor
 - Parallel add/remove
 - Parallel resize
 - Coarse-grained locking
 - Striped Locking
 - Non-blocking: DCAS, recursive split ordering
- Transactional Memory (Lecture 23, 24)
 - Transaction properties: atomicity, serializability, isolation, composability
 - Transaction operations: commit (validate, write-back), abort
 - Conflicts: Read-read, read-write, write-write
 - Hardware Transactional Memory
 - Isolation through caches and buffers
 - Atomicity through MESI
 - Implementation in modern hardware