Topics

1. Concurrency [3]
5. Data Parallelism [1]
7. Safety and liveness [2]
8. Distributed systems [4]
9. Architectures [1]

9.1. Hardware architecture
- From switches to registers and adders
- CPU architecture
- Hardware concurrency

9.2. Language architecture
- Chapel
- Occam
- Rust
- Ada
- C++

24 Lectures

1. Concurrency [3]
   1.1. Forms of concurrency [1]
   - Coupled dynamical systems
   1.2. Models and terminology [1]
   - Abstractions
   - Interleaving
   - Atomicity
   - Proofs in concurrent and distributed systems
   1.3. Processes & threads [1]
   - Basic definitions
   - Process states
   - Implementations

   2.1. by shared variables [1]
   - Failure possibilities
   - Dekker's algorithm
   2.2. by test-and-set hardware support [0.5]
   - Minimal hardware support
   2.3. by semaphores [0.5]
   - Dijkstra definition
   - OS semaphores

   3.1. Shared memory synchronization [2]
   - Semaphores
   - Cond. variables
   - Conditional critical regions
   - Monitors
   - Protected objects
   - Asynchronous / synchronous
   - Remote invocation / rendezvous
   - Message structure
   - Addressing

   4.1. Correctness under non-determinism [1]
   - Forms of non-determinism
   - Non-determinism in concurrent/distributed systems
   - Is consistency/correctness plus non-determinism a contradiction?
   4.2. Select statements [1]
   - Forms of nondeterministic message reception

5. Data Parallelism [1]
   5.1. Data-Parallelism
   - Vectorization
   - Reduction
   - General data-parallelism
   5.2. Examples
   - Image processing
   - Cellular automata

   6.1. Problem definition and design space [1]
   - Which problems are addressed / solved by scheduling?
   6.2. Basic scheduling methods [1]
   - Assumptions for basic scheduling
   - Basic methods

7. Safety and liveness [2]
   7.1. Safety properties
   - Essential time-independent safety properties
   7.2. Livelocks, fairness
   - Forms of livelocks
   - Classification of fairness
   7.3. Deadlocks
   - Detection
   - Avoidance
   - Prevention (& recovery)
   7.4. Failure modes
   7.5. Idempotent & atomic operations
   - Definitions

8. Distributed systems [4]
   8.1. Networks [1]
   - OSI model
   - Network implementations
   8.2. Global times [1]
   - Synchronized clocks
   - Logical clocks
   8.3. Distributed states [1]
   - Consistency
   - Snapshots
   - Termination
   8.4. Distributed communication [1]
   - Name spaces
   - Multi-casts
   - Elections
   - Network identification
   - Dynamical groups
   8.5. Distributed safety and liveness [1]
   - Distributed deadlock detection
   8.6. Forms of distribution/redundancy [1]
   - Computation
   - Memory
   - Operations
   8.7. Transactions [2]

9. Architectures [1]
   9.1. Hardware architecture
   - From switches to registers and adders
   - CPU architecture
   - Hardware concurrency
   9.2. Language architecture
   - Chapel
   - Occam
   - Rust
   - Ada
   - C++

Laboratories & Assignments

Laboratories [11]

   - Program structures
   - Data structures

2. Tasks [1]
   - Generics
   - Abstract types

3. Protection [1]
   - Memory based synchronization

4. Task Lifetimes [1]
   - Creation
   - Termination

5. Communicating Tasks [1]
   - Rendezvous

6. Distributing Server [1]
   - Entry families
   - Requeue facility

7. Implicit Concurrency [1]

8. Synchronized Data [1]

9. Distribution [1]
   - Multi-core process creation, termination
   - Multi-core process communication

10. Pipelines [1]

Assignments [2]

1. Concurrent programming [15%]
   - Programming task involving:
     - Mutual exclusion
     - Synchronization
     - Message passing

2. Concurrent programming in multi-core systems [15%]
   - Multi-core programming task involving:
     - Process communication

Examinations [3]

1. Hurdle check [5%]
   - Week 4 lab exam

2. Mid-semester check [15%]
   - Exam or Self-test

3. Final exam [50%]
   - Examining the complete course

Marking

The final mark is based on the assignments [30%] plus the examinations [65%] plus the lab mark [5%].