Learning Goals for TTK4145

These are roughly sorted by the order the topic will be lectured.

Course Learning Goals

- General maturation in software engineering/computer programming.
- Ability to use (correctly) and evaluate mechanisms for shared variable synchronization.
- Understanding how a deterministic scheduler lays the foundation for making real-time systems.
- Insight into principles, patterns and techniques for error handling and consistency in multi thread / distributed systems.
- Knowledge of the theoretical foundation of concurrency, and ability to see how this can influence design and implementation of real-time systems

Learning goals: Code Quality

- Be able to write software following selected Code Complete Checklists for modules, functions, variables and comments.
- Be able to critizise program code based on the same checklists.

Learning goals: Fault Tolerance Basics

- Understand and use terms (like): Reliability. Failure vs fault vs error. Failure modes. Acceptance test. Fault prevention vs. tolerance. Redundancy, Static vs. Dynamic. Forward/ Backward error recovery.
- Understand, use and evaluate techniques (like): N-version programming. Recovery blocks. Error detection. Failure mode merging. Acceptance tests

Learning goals: Fault model and software fault masking

- Understanding of the three cases in low level design for fault tolerance by redundancy: Storage, Computation and Communication.
- Understanding of the work method: 1) Find error model 2) detect errors and merge failure modes (+error injection for testing) 3) handling/masking with redundancy ...aiming for progression of fail fast, reliable and available systems
- Ability to Implement (simple) Process Pairs-like systems

Learning goals: Transaction Fundamentals

- Knowledge of eight "design patterns" (Locking, Two-Phase Commit, Transaction Manager, Resource Manager,Log,Checkpoints,Log Manager, Lock Manager), how they work and which problems they solve. Ability to utilize these patterns in highlevel design.
- Comprehension of terms: Optimistic Concurrency Control, Two-phase commit optimizations, Heuristic Transactions, Interposition.

Learning Goals: Atomic Actions

- A thorough understanding of the problems Atomic Actions are meant to solve and how these motivates the different aspects of Atomic Actions.
- Ability to use and implement Atomic Actions, including the mechanisms providing the start, side and end boundaries.
- Understanding the motivation for using Asynchronous Notification in Atomic Actions
- A coarse knowledge of how the mechanisms for Asynchronous Notification in C/Posix, ADA and Java works.

Learning Goals: Shared Variable Synchronization

- Ability to create (error free) multi thread programs with shared variable synchronization.

- Thorough understanding of pitfalls, patterns, and standard applications of shared variable synchronization.
- Understanding of synchronization mechanisms in the context of the kernel/HW.
- Ability to correctly use the synchronization mechanisms in POSIX, ADA (incl. knowledge of requeue and entry families) and Java.

Learning Goals: Scheduling

- Be able to prove schedulability using the utilization test and the response time analysis for simple task sets.
- Know and evaluate the assumptions underlying these proofs and what is proven.
- Understand the bounded and unbounded priority inversion problems.
- Understand how the ceiling and inheritance protocols solves the unbounded priority inversion problem.
- Understand how the ceiling protocol avoids deadlocks.