

Design of Distributed and Multi-Core Systems & Software

HILF! GmbH Microcomputer-Consulting, Bajuwarenring 17, 82041 Oberhaching/München
Tel.: +49 (89) 61 37 90 - 0, Fax: +49 (89) 62 52 14 5

Web: www.hilf.de
Email: info@hilf.de

Target group

real-time and embedded systems software system architects, project managers and technical consultants who will have responsibility for designing, structuring and implementing the RTOS/embedded software.

Prerequisites

Course participants are expected to have background in embedded and real-time software design like our course "Architectural Design of Real-Time Software"

Course description

This course examines the high-level design of embedded systems and software for distributed and multicore processing environments. It begins with a discussion of the basic concepts of distributed systems and multi-core systems-on-a-chip ("SoC's"). This is followed by an in-depth study of distributed control systems design, including examples from automotive applications and home automation. Guidelines are given for the design of large and complex distributed systems, with

examples from the worlds of transportation and high-performance communication systems. The course then shifts focus to the use of multi-core SoC's in embedded systems designs. This includes detailed study of both symmetric and asymmetric multiprocessing -- from the perspectives of hardware, software and operating systems support. It delves deeply into operating systems for multi-core SoC's, multi-core software architectural design, and special memory issues in multi-core software.

This course is far from a general course about system or software design theory, but rather it is highly focused on the practical design of multi-processor embedded systems and software that will operate in distributed and multicore processing configurations.

Course topics

- Definitions and Background
 - Distributed Systems
 - Multi-Core Systems-on-a-Chip
 - Differences: Multi-Core vs. Distributed
- Distributed Control Design
 - Loosely-Coupled Distributed Systems
 - Distributing a Control Capability
 - Examples: On-Board Automotive Networks
 - System Partitioning Guidelin
 - Exercise: Body Electronics Control for Airport Train
- Designing Complex Distributed Systems
 - Identifying Sub-systems: Sub-system Structuring Criteria
 - Distributed Applications: Logical vs. Physical Nodes
 - Design Models for Distributed Applications
 - Exercise: Multiprocessing System Performance Calculations
 - Decomposition of Sub-systems into Software Tasks (Optional)
 - Message Communication Performance Modeling (Optional)
 - Case Study: High-Performance IP Communication Router Design
- Multi-Core Systems-on-a-Chip (SoC's)
 - Amdahl's Law
 - Fine-Grained vs. Coarse-Grained Parallelism
 - Symmetric vs. Asymmetric Multiprocessing
 - Operating Systems for Embedded Multiprocessing
- Symmetric Multiprocessing ("SMP")
 - Organizing Software for SMP
 - Operating Systems Support for SMP
 - Spinlocks
 - Load Balancing vs. Processor Affinity
 - Hidden Dangers in Designing Software for SMP
- Asymmetric Multiprocessing ("AMP")
 - When to Use AMP
 - Operating Systems for AMP
 - Design Patterns for Multicore Software
 - Moving from Uni-Processing to AMP
- Memory Issues in MultiCore Software
 - Working with Cache Memory
 - Memory Contention
 - "False Sharing"
 - Memory Consistency and Inconsistency
- Examination