

Safety Critical and High Availability Systems

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

This course is intended for practicing real-time and embedded systems software system architects, project managers and technical consultants who have responsibility for designing, structuring and implementing the software for real-time and embedded computer systems in applications that could, when they fail, threaten the well-being or life of people.

Prerequisites

Course participants are expected to be familiar with general embedded and real-time software design. This knowledge can be gained by attending a prerequisite embedded software design course such as "Architectural Design of Real-Time Software".

Course description

The Safety Critical part examines the design of embedded systems and software that are to provide services in applications that could, when they fail, threaten the well-being or life of people. It offers practical guidance on how to address safety concerns when designing safety critical software in fields such as medical, automotive, avionics, nuclear and chemical process control.

High availability systems must tolerate both expected and unexpected faults. Their design is based on redundant hardware and software combined in ways that will achieve "five-nines" (99.999%) or greater availability, equivalent to less than 1 second of downtime per day. Basic hardware N-plexing and voting issues are discussed, followed by an in-depth study of a number of backward error recovery fault tolerance techniques including static N-version programming, Checkpoint-Rollback, Process Pairs, and Recovery Blocks. Examples from Space Shuttle and Airbus 330/340 are being showed.

Course topics

- Definitions and Background
 - Hazards and Risks
 - Safety vs. Fault Tolerance
 - Design Issues for Safety
 - Redundancy
 - Approaches to Dependability
 - Code-Level Safety: MISRA-C and LINT
 - Examples: Automotive Brake-by-Wire, Steer-by-Wire
- Preparatory Analyses
- Fundamental Safety Design Patterns
- Multi-Channel Design Patterns
- Design Patterns for High Availability and Safety
 - Monitor-Actuator Pattern
 - Extended Example: Medical Respiratory Ventilator
 - The Safety Executive
 - Extended Example: Automotive Drive-by-Wire
 - Extended Example: Airbus A330/340 Fly-by-Wire
 - A Cookbook for Safety-Critical Design
- C Language in Critical / High Availability Systems
 - Software Robustness: MISRA-C, LINT, Static Code Analyzers
 - Exercise: C-Language Shenanigans
- Concepts for Backward Error Recovery
- System and Software Design Patterns for High Availability
- Technical Issues in High Availability Design
 - RAID: Redundant Arrays of Inexpensive Disks
 - Exercise: Hamming Codes
 - Failover Management
 - Data Replication
 - Dealing with Software Design Faults
 - Extended Example: Airbus A330/340 Fly-by-Wire
- Underlying Principles
 - Fault Avoidance vs. Tolerance
 - Failure Curves
 - Redundancy
 - Replication vs. Functional Redundancy vs. Analytic Redundancy
 - Dynamic vs. Static Redundancy
 - Extended Example: Space Shuttle Software
- Final Examination