Proposal for a Master Thesis at the
Chair of Robotic and Embedded system

The exploration and implementation of real-time operating systems by posix threads

Advisors: Biao Hu
Dr. Kai Huang

Supervisor: Prof. Dr. Alios Knoll

Project Description

In the context of TU9 project, research at the Institute of Robotics and Embedded Systems is dealing with the challenge of providing high-performance ECUs as an enabling technology applicable in the automotive field, which will tackle the problem of integrating tasks with different criticalities.

Background The implementation and evaluation of real-time operating systems are important as the long response time of tasks incurred by the operating system may significantly hamper the system performance, and sometimes endanger people's life, such as the long response time of safe airbag.

An increasing important trend in the design of real-time and embedded systems is the integration of components with different levels of criticality onto a common hardware platform. At the same time, these platforms are migrating from single cores to multi-cores and, in the future, many-core architectures. Criticality is a designation of the level of assurance against failure needed for a system component. A mixed criticality system (MCS) is one that has two or more distinct levels (for example safety critical, mission critical and non-critical). Perhaps upto five levels may be identified (see, for example, the IEC 61508, DO-178B, DO-254 and ISO 26262 standards). Typical names for the levels are ASILs (Automotive Safety and Integrity Levels), DALs (Design Assurance Levels) and SILs (Safety Integrity Levels).

Most of the complex embedded systems found in, for example, the automotive and avionics industries are evolving into mixed criticality systems in order to meet stringent non-functional requirements relating to cost, space, weight, heat generation and power consumption (the latter being of particular relevance to mobile systems). Indeed the software standards in the European automotive industry (AUTOSAR[1]) and in the avionics domain (ARINC[2]) address mixed criticality issues; in the sense that they recognise that MCSs must be supported on their platforms.

The Thesis: In this thesis, the student will implement the current state-of-art scheduling algorithm in the real system. There is already a framework named SF3P [1] that successfully implement some common scheduling policies (fixed priority, earliest deadline first) by the posix thread. Under this framework, the sudent is supposed to implement the basic mixed-criticality scheduling algorithm.

Requirements

The basic requirements will be that

1. Familiar with the linux programming environment and Posix thread programming.

2. Programming: C/C++

3. Interest in the real-time scheduling operating system.

[1] http://www.autosar.org/
[2] http://www.arinc.com/

Arrangement

The project will include the following phases:

- Learn the framework of SF3P.
- Implement your own the scheduling algorithm under the SF3P.

Kind of Work

- 20% theory
- 50% implementation
- 20% evaluation
- 10% documentation

Contact

Biao Hu	Dr. Kai Huang
MI 03.07.059	MI 03.07.042
hub@in.tum.de	huangk@in.tum.de
Phone: +49.89.289.18128	Phone: +49.89.289.18111

References

[1] A. Gomez, L. Schor, P. Kumar, and L. Thiele, "Sf3p: a framework to explore and prototype hierarchical compositions of real-time schedulers," in *Rapid System Prototyping (RSP), 2014 25th IEEE International Symposium on*. IEEE, 2014, pp. 2–8.