Nowadays, multi-core systems have become the mainstream processors. To schedule real-time tasks on multi-core systems, there have been three widely adopted paradigms: partitioned, global, and semi-partitioned scheduling. In addition to the aforementioned task scheduling approaches, resource sharing should also be considered for multi-core systems.

Priority Inversion without using any protocol in a uniprocessor

Since the sharing resource access must ensure mutual exclusion, without any aid of protocols, a higher priority job may be blocked by a lower-priority job, which is called priority inversion. In uniprocessor system, locking protocols have been widely used to mitigate the effect of priority inversion based on priority inheritance, e.g., the priority inheritance protocol (PIP), the priority ceiling protocol (PCP), and the stack resource policy (SRP), which are already shown reasonably good at handling synchronization and mutual exclusions.

Since state-of-the-art uniprocessor resource sharing protocols cannot be directly applied on multi-core systems with multi-tasking, the problem is still an ongoing topic in the researches. Recently, The Multiprocessor Resource Sharing Protocol (MrsP) has been proposed in ECRTS’13 for solving the resource sharing problem in partitioned scheduling manner and implemented in two popular real-time operating system (RTOS): Litmus-RT and RTEMS [1]. Furthermore, in RTSS’16 a new partitioned scheduling for resource sharing has been proposed but not yet been implemented in any RTOSs. In this thesis, we will focus on analyzing the aforementioned protocols and evaluating the performance/pitfalls by using RTEMS specifically.

In this thesis, the student will first study the theory of MrsP in [1] and the source code of latest version RTEMS. By using the existing implementation of MrsP in RTEMS, the student should explore the performance/pitfalls of MrsP and implement the partition algorithm in [2] similarly in RTEMS. The students should evaluate targeted approaches with sound experiments on with a comprehensive implementation.

Other suggestions and related topics are also welcome. Please do not hesitate to make an appointment.

Required Skills:
- Good Knowledge of C
- Knowledge of Embedded System and Linux
- Real Time Scheduling knowledge is beneficial

Acquired Skills after the thesis:
- Knowledge of resource sharing protocols on multi-core systems and RTEMS.
- Design, Analysis, Implementation of system software on real-time embedded system

[1] Catellani et al., Challenges in the Implementation of MrsP, LNCS 2015