

Bachelor Thesis

Supporting Multi-processor Resource Synchronization on Erika RTOS

Junjie Shi
Dr. Kuan-Hsun Chen
Prof. Dr. Jian-Jia Chen
Otto-Hahn Str. 16
Technische Universität Dortmund
12.May.2020

To prevent race conditions or data corruptions, concurrently accessing the same resource is prohibited by exploiting mutual exclusion. Thus, many semaphore based resource synchronization protocols have been widely used to realize mutual exclusion and prevent deadlocks and priority inversions. In modern multi-processor platforms, many resource synchronization protocols have been proposed, i.e., Multiprocessor Priority Ceiling Protocol (MPCP) [1] and Distributed Priority Ceiling Protocol (DPCP) [2]. However, the performance of aforementioned protocols in real operating systems by considering overheads are still lack of research.

To evaluate the performance in real implementation, we target on Erika Enterprise [3], a royalty free automotive OSEK/VDX certified Hard Real Time Operating System (RTOS). ERIKA v3 has supported hard real-time with Fixed Priority (FP) scheduling and Immediate Priority Ceiling Protocol (I-PCP). In Erika OS, Earliest Deadline First (EDF), resource reservation and limited preemption schedulers are supported. In addition, both partitioned and global scheduling are supported by default. However, not many resource synchronization protocols are supported in Erika OS, i.e., only I-PCP is supported by default.

In this work, the student has to first study the existing implementation of resource synchronization protocols, i.e., I-PCP in the latest stable version of Erika OS [4]. Then, the student need to implement the MPCP and DPCP after obtaining the knowledge of these two protocols. One example of how to implement a multiprocessor resource synchronization protocol, i.e., flexible spin-lock model (FSLM) can be found in [5]. In the end, the student has to evaluate the performance of the new approaches with I-PCP on Erika OS. Also, the evaluation for the same approach on different platforms is preferred, load the implementation on both ARM based platform and X86 base platform (more information related to the architectures supported in Erika OS can be found in [5]).

Required Skills:

- Knowledgeable of C and C++ programming
- Basic knowledge of tasks and schedulers
- Knowledge of System Programming
- Real Time Operating System knowledge is beneficial

Acquired Skills after the work:

- Knowledge of Real-Time Operating System
- Knowledge of Resource Synchronization
- Knowledge of Open-Source Software
- Design, Analysis, Implementation of system software on real-time system, e.g., version control, coding convention, etc.

[1] Sha L, Rajkumar R, Lehoczky Priority inheritance protocols: An approach to real-time synchronization[J]. IEEE Transactions on computers, 1990, 39(9): 1175-1185.

[2] Rajkumar et al., Real-time synchronization protocols for multiprocessors RTSS, 1988

[3] <http://www.erika-enterprise.com/>

[4] <https://github.com/evidence/erika3>

[5] https://pdfs.semanticscholar.org/e5fb/917c0b2cd46f4d0f04cc3a84f8d26956256f.pdf?_ga=2.208846420.757358935.1589189328-1178069878.1581856499

[6] <http://www.erika-enterprise.com/index.php/erika3/supported-architectures.html>