

Master/Bachelor Thesis

Implementing and Verifying Resource Synchronization Protocols in RTOSes

Junjie Shi
Prof. Dr. Jian-Jia Chen
Technische Universität Dortmund
Email: junjie.shi@tu-dortmund.de
April 4, 2022

To prevent race conditions or data corruptions, concurrently accessing the same resource is prohibited by exploiting mutual exclusion. Several semaphore-based protocols have been developed to provide a deadlock-free resource synchronization and mitigate the priority inversions. In modern multiprocessor platforms, many resource synchronization protocols have been proposed, e.g., Multiprocessor Priority Ceiling Protocol (MPCP) [5] and Distributed Priority Ceiling Protocol (DPCP) [6] and implemented on various real-time operating systems (RTOSes), e.g., RTEMS [2].

To ensure the correctness of such implementations, dedicated corner cases will be designed and deployed to testify whether the system behaves as expected. However, such case-based validation is usually not sufficient, since it is not possible to test over all the sensitive cases empirically. Alternatively, Gadia et al. [3] use Java to model the implementation of Priority Inheritance Protocol on uniprocessor in the RTEMS and validate the model by using Java Pathfinder to exhaustively detect potential data races, deadlocks, and priority inversions. However, the required effort cannot easily scale to validate the other advanced protocols or even for multiprocessor systems.

Towards this, formal verification has been adopted recently. Gu et al. in [4] develop a practical concurrent OS kernel named CertiKOS, and verify its (contextual) functional correctness with a proof assistant Coq. Although some researches continue to provide certified real-time systems on the same vein, we are more interested to verify the existing implementations on off-the-shelf real-time platforms. We can foresee that such an alternative trait is challenging but potentially effective to the relevant researches.

In this thesis, the student is expected to convert the existing implementations of DPCP, MPCP, FMLP, and DGA [7] on the latest version of RTEMS, i.e., 5.1, at first. Afterwards, all the necessary properties of the assumptions (HW and OS) used by these

protocols must be defined and modeled. Based on the proposed models, the student has to formulate the properties of protocols that can be validated, e.g., ceiling priority boost, migration mechanism, and forced execution order. Here, we assume the basic components provided in RTOSes and hardware are correct, e.g., priority based scheduler. At the end, the student is supposed to validate/verify how these necessary properties are matched prove the correctness of considered implementation of DPCP, MPCP, FMLP, and DGA on RTEMS by using Frama-C [1].

Required Skills:

- Knowledge of C and C++ programming
- Knowledge of Real-Time System
- Knowledge of System Programming and
- Architecture knowledge is beneficial

Acquired Skills after the work:

- Knowledge of Resource Synchronization
- Knowledge of Real-Time Operating Systems
- Design, Analysis, Implementation of system software on real-time system

References

- [1] Frama-C software analyzers. <https://frama-c.com/>. visited on 15.10.2021.
- [2] RTEMS. <http://www.rtems.org/>.
- [3] S. Gadia, C. Artho, and G. Bloom. Verifying nested lock priority inheritance in RTEMS with java pathfinder. In K. Ogata, M. Lawford, and S. Liu, editors, *Formal Methods and Software Engineering - 18th International Conference on Formal Engineering Methods, ICFEM 2016, Tokyo, Japan, November 14-18, 2016, Proceedings*, volume 10009 of *Lecture Notes in Computer Science*, pages 417–432, 2016.
- [4] R. Gu, Z. Shao, H. Chen, X. N. Wu, J. Kim, V. Sjöberg, and D. Costanzo. Certikos: An extensible architecture for building certified concurrent OS kernels. In *12th USENIX Symposium on Operating Systems Design and Implementation, OSDI 2016, Savannah, GA, USA, November 2-4, 2016*, pages 653–669, 2016.
- [5] R. Rajkumar. Real-time synchronization protocols for shared memory multiprocessors. In *Proceedings, 10th International Conference on Distributed Computing Systems*, pages 116 – 123, 1990.
- [6] R. Rajkumar, L. Sha, and J. P. Lehoczky. Real-time synchronization protocols for multiprocessors. In *Proceedings of the 9th IEEE Real-Time Systems Symposium (RTSS '88)*, pages 259–269, 1988.
- [7] J. Shi, J. D. T. Pham, M. Münch, J. V. Hafemeister, J. Chen, and K. Chen. Supporting multiprocessor resource synchronization protocols in RTEMS. *CoRR*, abs/2104.06366, 2021.