technische universität dortmund

computer science 12

# Master Thesis

## Non-Volatile Unikernels: Library Support for Non-Volatile Memory in Library OSes

Christian Hakert
Dr.-Ing. Kuan-Hsun Chen
Otto-Hahn Str. 16
Technische Universität Dortmund
Email: christian.hakert@tu-dortmund.de
13.01.2021

Unikernels and Library OSes are a novel approach for the design of configurable and tailored kernels. Due to the fact that unnecessary code can be eliminated during configuration, unikernels achieve a minimal code size and maximal efficiency. For instance, if a bare-metal applications does not require unix like system services, but only a lightweight device driver layer instead, a unikernel can be configured to only provide such a lightweight interface and does not include the unused system services. This also increases the execution speed of the kernel. Unikraft [1], as a Unikernel,

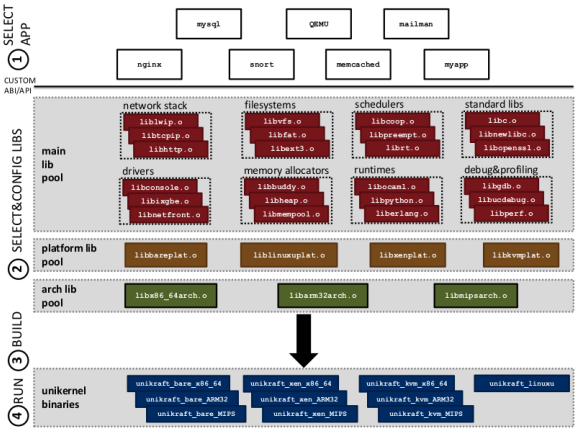


Figure 1: Unikraft Kernel Architecture

provides a large code basis and support for many applications (e.g. python support, unix like system services, scheduling, ...). Therefore Unikraft is a good candidate to develop future applications on embedded systems and provide a tailored execution environment for these applications.

The commercial rise of systems with non-volatile memory (NVM) as main memory further yields the chance to develop specific applications, which utilize the non-volatility of the underlying memory explicitly. These applications can save information during power cycles, without the need to explicitly store them in a volatile memory. This, for instance, allows to persist algorithm states in a simple manner.

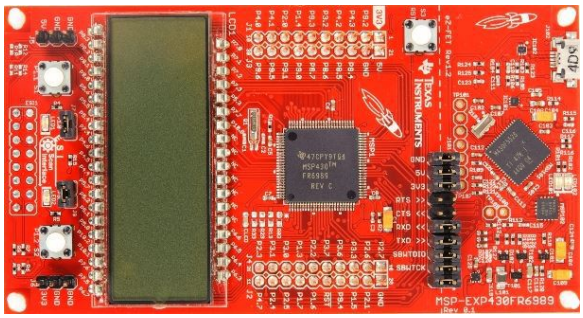


Figure 2: MSP430FR6989 Development System

**In this thesis**, students first are obligated to investigate the MSP430FR6989 architecture [2] and get familiar with the source tree of the Unikraft kernel. In the following, the required hardware support to run Unikraft on the given hardware has to be designed and implemented. Once the Unikernel fully executes on the target device, a specific library component for the support of non-volatile memory should be designed. This library shall provide a common interface to other libraries, which offers extended functionality for the NVM. The interface should not only allow to create persistent data structures, but also should provide checkpointing mechanisms, such that data structures can be always recovered to the recent state of integrity.

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

**Required Skills:**

- Knowledge of C and C++ programming
- Knowledge of operating system design and low level programming (e.g. lectures like BSB, RTOS)

**Acquired Skills after the thesis:**

- Deep Knowledge about the functioning of Library OSes
- Understanding of non-volatile memories and data persistency in main memory

[1] Kuenzer, S. (2018). Unikraft: Unikernels Made Easy.
[2] https://www.ti.com/product/MSP430FR6989