EUKL: A Prototype of a Linux-Based Unikernel for Resource-Constrained Embedded Systems

Yoshifumi Shu
shu.yoshifumi@ertl.jp
Nagoya University
Nagoya, Aichi, Japan

Yutaka Matsubara
yutaka@ertl.jp
Nagoya University
Nagoya, Aichi, Japan

Yixiao Li
liyixiao@ertl.jp
Nagoya University
Nagoya, Aichi, Japan

Hiroaki Takada
hiro@ertl.jp
Nagoya University
Nagoya, Aichi, Japan

ACM Reference Format:
Yoshifumi Shu, Yutaka Matsubara, Yixiao Li, and Hiroaki Takada. 2024. EUKL: A Prototype of a Linux-Based Unikernel for Resource-Constrained Embedded Systems. In . ACM, New York, NY, USA, 1 page. https://doi.org/10.1145/nnnnnnn.nnnnnnn

1 Introduction

Resource-constrained embedded systems are embedded systems with strict hardware resource constraints and typically lack an MMU (Memory Management Unit) and hardware-assisted virtualization support. These significantly contribute to minimizing SWaP-C (size, weight, power, and cost), thus they are widely used in real-world products. There is a demand to utilize the high-functional software assets available in Linux to reduce development costs in embedded systems that require real-time capability and reliability. Ensuring real-time capability with Linux alone is challenging, hence the virtualization approach that runs Linux and an RTOS (Real-Time Operating System) simultaneously is commonly used to meet this demand. However, this approach cannot be adopted in resource-constrained embedded systems due to the lack of hardware-assisted virtualization support. To the best of our knowledge, there is no approach to simultaneously support real-time applications and Linux applications in a single resource-constrained embedded system without compromising real-time capability and reliability.

To address this issue, we focus on UKL (Unikernel Linux)[1], a prior work on a Linux-based unikernel, which converts Linux into a unikernel-style OS running on bare metal or inside a VM. We consider an architecture that runs a Linux-based unikernel on top of an RTOS. While the Linux-based unikernel provides a Linux execution environment, the RTOS protects real-time applications from the Linux kernel and its applications. In this poster, we present EUKL, a fork of UKL, running as an RTOS application.

Conference'17, July 2017, Washington, DC, USA
© 2024 Copyright held by the owner/author(s). Publication rights licensed to ACM.
ACM ISBN 978-x-xxxx-xxxx-x/YY/MM
https://doi.org/10.1145/nnnnnnn.nnnnnnn

2 The Design of EUKL

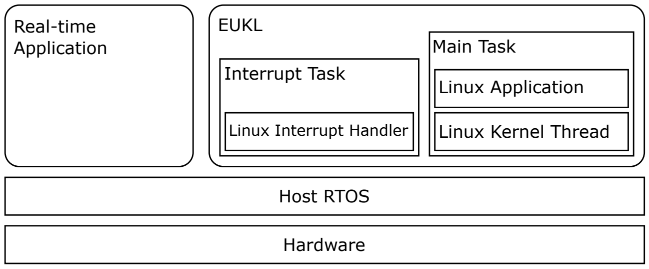

Figure 1: Overview of EUKL architecture.

Figure 1 shows the overall architecture of a resource-constrained embedded system that utilizes EUKL. All the Linux code runs inside RTOS tasks.

The design of EUKL involves the following three steps. First, Nommu Linux (Linux using no MMU) is converted into a Linux-based unikernel in the same way as UKL. Next, we port it onto an RTOS by removing privileged code and virtualizing interrupts to leverage the temporal and spatial isolation provided by the RTOS. Finally, LTO (Link Time Optimization) is applied to both the Linux kernel and user applications to improve resource efficiency.

3 Evaluation

Our early-stage implementation of EUKL targets the Arm Cortex-M4 @ 168 MHz and TOPPERS/ASP3 RTOS, and runs a hello world application in C as a test Linux application. The impact on the worst-case execution time in a real-time application is calculated as (3μs + (the processing time of interrupt handler in the RTOS kernel)) × (the number of interrupts assigned to EUKL). Compared to vanilla Linux, the ROM usage decreases by 30%. These results show that the impact on real-time capability is sufficiently minimal and resource efficiency is improved, demonstrating that our approach is practical and beneficial for resource-constrained embedded systems.

References

[1] Ali Raza, Thomas Unger, Matthew Boyd, Eric B Munson, Parul Sohal, Ulrich Drepper, Richard Jones, Daniel Bristot De Oliveira, Larry Woodman, Renato Mancuso, Jonathan Appavoo, and Orran Krieger. 2023. Unikernel Linux (UKL). In *Proceedings of the Eighteenth European Conference on Computer Systems* (Rome, Italy) *(EuroSys '23)*. Association for Computing Machinery, New York, NY, USA, 590–605. https://doi.org/10.1145/3552326.3587458