



Contribution ID: 192

Type: **not specified**

From Concerns to Code: On Rust Driver Infrastructure and Maintenance

Thursday 11 December 2025 12:45 (45 minutes)

About a year ago, I presented the initial efforts to upstream the fundamental infrastructure needed to enable complex Rust drivers in the Linux kernel – work that started with the Nova GPU driver. At the same time, much of the discussion around Rust in the kernel centered on concerns about (long-term) maintenance and the potential burden of supporting a second language.

Today, that foundational infrastructure is upstream. This includes core pieces of driver support such as the generic device/driver model, PCI, platform and auxiliary bus support, device resource management, memory-mapped I/O, memory allocation primitives, and DMA. In this talk, I will focus on two aspects:

First, we will look at how this infrastructure - especially driver-core infrastructure - has turned out in practice. Using a real upstream driver as a reference, I will highlight the benefits Rust's type system brings, walk through some interesting implementation details, and also discuss shortcomings that still require cooperation with the C side to address.

Second, we will revisit the maintenance concerns mentioned previously. Now that the corresponding infrastructure is upstream and active driver development in Rust is happening in-tree, we can make an initial assessment of how those concerns have played out in practice. As a maintainer of both Rust and C kernel infrastructure, I will share my experience from both directions: collaborating with C maintainers while maintaining Rust code, and working with Rust contributors in the context of C infrastructure that I maintain.

Primary author: Mr KRUMMRICH, Danilo (Red Hat)

Presenter: Mr KRUMMRICH, Danilo (Red Hat)

Session Classification: LPC Refereed Track

Track Classification: LPC Refereed Track