Contribution ID: **240** Type: **not specified**

Overflowing with Fear: Detecting and Mitigating Implicit Panics in Rust

One of the main selling points for Rust's inclusion in the kernel is safety, which is strongly associated with a reduction of runtime panics. Yet, in Rust an integer overflow or out-of-bounds array access translates into an implicit panic, inserted without any warning to the programmer.

The inability to easily identify where these implicit panic sites are introduced creates a blind spot when writing critical kernel code. Code that processes untrusted user-space data is particularly vulnerable to hitting these conditions, and a formal guarantee that e.g. a given function is panic-free would be extremely helpful.

This topic aims to describe the conditions under which such panic sites can be inserted, review existing solutions from the user-space ecosystem (like the no-panic crate), and discuss how this issue can be mitigated in the kernel.

Potential solutions include new tooling or compiler support to flag potential panic sites, or establishing stricter coding rules to forbid them altogether through e.g. checked variants of potentially panicking operations.

Primary author: COURBOT, Alexandre (NVIDIA)

Presenter: COURBOT, Alexandre (NVIDIA)

Session Classification: Rust MC

Track Classification: Rust MC