Contribution ID: **22** Type: **not specified**

Safe Systems with Linux

As Linux is increasingly deployed in systems with varying criticality constraints, distro providers are being expected to ensure that security fixes in their offerings do not introduce regressions for customer products that have safety considerations. The key question arises: How can they establish consistent linkage between code, tests, and the requirements that the code satisfies?

This MC addresses critical challenges in requirements tracking, documentation, testing, and artifact sharing within the Linux kernel ecosystem. Functionality has historically been added to the kernel with requirements explained in the email justifications for adding, but not formalized as "requirements"in the kernel documentation. While tests are contributed for the code, the underlying requirement that the tests satisfies is likewise not documented in a consistent manner.

Potential topics to be discussed:

- where should requirements that the kernel code and testing satisfies be tracked? In kernel documentation, in the code, etc.
- incorporating requirement linkage to the kernel code and tests that minimizes the impact to kernel maintainers and contributors.
- examples and strategies for enhancing documentation quality and level of detail within the Linux kernel so that effective safety analysis can be performed for products. Some starting points have been started [1], but what else is needed.
- connecting artifacts in a shareable format: how to effectively link and share testing, documentation, bug reports, and CVE information across multiple projects, infrastructures, and contribution processes.
- traceability and change identification in requirements to keep in sync with the evolving kernel code functionality and security fixes.
- increasing code test coverage of the Linux kernel to satisfy the higher safety assurance considerations. There's been some recent studies conducted by Boeing and the University of Illinois on various coverage types, that should be considered.
- requirements introduced by the Cyber Resilience Act in the EU [2] on product manufacturers might have on the Linux Kernel development process and documentation.
- improving systematic error responses when using Linux as well as runtime verification monitoring.

Last year, we had several talks on the need for safe systems [3][4] in various domains with Linux as a component (with varying safety criticality levels). This miniconference is targetted at getting those interested together, and working up a framework for collecting relevant evidence and sharing it.

MC Leads:
Kate Stewart, Philipp Ahmann

Potential Participants (not confirmed yet):

Syed Mohammed Khasim
Jonathan Corbet
Shuah Khan
Greg Kroah-Hartman
Chuck Wobler
Daniel Bristot de Oliveira

Thomas Gleixner
Gabrielle Paoloni
Olivier Charrier
Jiri Kosina
Joachim Werner
Paul Albertela
Bertrand Boisseau

[1] https://docs.kernel.org/admin-guide/workload-tracing.html
[2] https://digital-strategy.ec.europa.eu/en/policies/cyber-resilience-act
[3] https://lpc.events/event/17/contributions/1499/
[4] https://lpc.events/event/17/contributions/1518/

Primary authors: STEWART, Kate (Linux Foundation); AHMANN, Philipp (Robert Bosch GmbH)

Presenters: STEWART, Kate (Linux Foundation); AHMANN, Philipp (Robert Bosch GmbH)

Track Classification: LPC Microconference Proposals