Contribution ID: **2** Type: **not specified**

Safe Systems with Linux MC

Description

As Linux continues to be deployed in systems with varying criticality constraints, progress needs to be made in establishing consistent linkage between code, tests, and requirements, to improve overall efficiency and ability to support necessary analysis.
This MC addresses critical challenges in expectation management (aka requirements tracking), documentation, testing, and artifact sharing within the Linux kernel ecosystem. While tests are contributed for the code, traditionally the underlying requirement that the tests satisfies is likewise not documented in a structured manner. This has resulted in a large amount of "tribal knowledge" associated with subsystems, which results in technical debt when maintainers stop working on subsystems.

Taking in the feedback from last year's "Safe Systems with Linux" miniconference 1, on how we can improve the documentation of the kernel's design [1a] the ELISA (Enabling Linux in Safety Applications) community has focused on prototyping a template for capturing the requirements with volunteer linux kernel subsystem maintainers. The ELISA architecture team 2 has been meeting weekly and has developed a structured approach for documenting testable expectations with a template that allows embedding requirements directly with relevant code (as requested in the initial workshop) while maintaining machine readability and forming a base for improving testing with initiatives like KernelCI. The prototype format got initial review and feedback in December at the ELISA workshop at Goddard [3] and after incorporating that feedback in the workshop in Lund in May [4].

Initial pilots in the TRACING subsystem [5] have demonstrated the value of this approach, even resulting in the identification and fixing of previously unknown issues. [6,7]

Building on the last year's discussions, the goal of this miniconference is to get wider feedback from additional maintainers and developers of different subsystems on the approach being proposed.

Potential Topics

- **Progress on Linux Kernel Requirements Framework**
  Discussing the SPDX-based template for low-level requirements, lessons learned from initial pilots, and plans for wider adoption.
- **Technical Debt Reduction**
  How documented requirements capture understanding of original functionality, and can be leveraged for verification when code needs to be rewritten (ie. C to Rust), etc.
- **Requirements-Driven Testing**
  How documented requirements can drive test case development and validation. Connecting relevant test cases with specific requirements and code, should be able to yield more efficient testing.
- **Semantic Aspects of Kernel Requirements**
  Exploring how to properly document expected behaviors with consideration for design elements that impact or are impacted by these behaviors.
- **Practical Implementation Challenges**
  Addressing the balance between detailed requirements documentation and maintaining kernel development velocity.

- **Required tools for automation**
  Progress on tools to generate, validate, and track work products increasing dependability throughout the kernel development process.
- **Industry Adoption**
  How safety-critical industries are beginning to leverage these developments for certification and compliance purposes. How their safety engineers can participate in contributing formalized requirements to the kernel and providing linkage.
- **Requirements as an Education Tool**
  How linux kernel documentation can mine the requirements, and help new contributors understand kernel functionality and design intent and attract new upsteam developers

Summary

Last year, we established the need for better documentation of requirements in safe systems. This year, we will showcase concrete progress, including a working framework for Linux Kernel Low Level Requirements with practical implementations in the TRACING subsystem. This MC aims to bring together kernel maintainers, developers, with safety architects and industry stakeholders to expand adoption of these practices and address remaining challenges in building safe systems with Linux. It should engage with testing and documentation centric activities and how all parts can link together.

Potential Participants

Steve Rostedt
Greg Kroah-Hartman
Thomas Gleixner
Jonathan Corbet
Tim Bird
Shuah Khan
Gustavo Padovan
Gabrielle Paoloni
Chuck Wolber
Luigi Pellecchia
Alessandro Carminati
Wolfram Sang (Renesas BSP)
Vincent Mailhol (CAN Subsystem)
Kate Stewart
Philipp Ahmann
Nicole Pappler

References

\1 LPC 2024 Safe Systems with Linux Miniconf: https://lpc.events/event/18/sessions/187/#20240920

[1a] Kernel design documentation improvement: https://www.youtube.com/watch?v=stqGiy85s_Y

\2 ELISA Architecture meetings: https://lists.elisa.tech/g/safety-architecture

[3] NASA workshop: https://www.youtube.com/watch?v=_N3l_EEV8uM

[4] Link to Lund workshop session: https://drive.google.com/file/d/1–e82k80_D79ycJdFbEwLQ0kpbD0pMR9/view?usp=sharing

[5] Drafting requirements in tracing subsystem: https://github.com/torvalds/linux/compare/master...elisa-tech:linux:linux_requirements_

[6] Patch on LKML: https://lkml.org/lkml/2025/3/21/1128

[7] Patch on LKML: https://lkml.org/lkml/2025/5/21/850

Primary authors: STEWART, Kate (Linux Foundation); AHMANN, Philipp (Etas GmbH (BOSCH))

Presenters: STEWART, Kate (Linux Foundation); AHMANN, Philipp (Etas GmbH (BOSCH))