Linux Features for Safety-Critical Systems WG - Annual Update

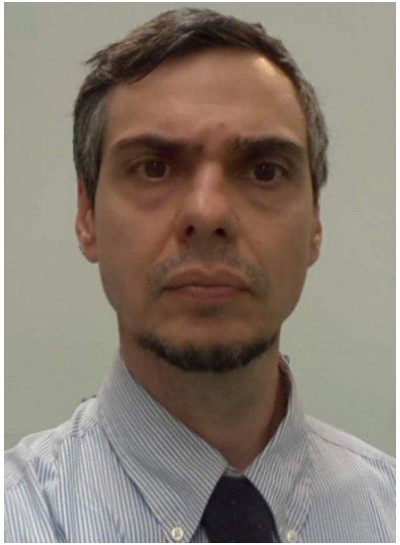
Alessandro Carminati
Red Hat

ELISA
Enabling Linux in Safety Applications

Aerospace · Automotive · Linux Features
Medical Devices · OS Engineering Process
Safety Architecture · Space Grade Linux · Systems · Tools

The Host

- Hostname: Alessandro Carminati
- Linux Kernel Developer @ Red Hat (automotive distribution)
- Contributing at Architecture, Tools and Linux Features WGs.
- Focus on the Linux Kernel safety, core of the GNU Linux safety.



Working Group Statement [-]

- **Kernel Feature Investigation:** Explore and evaluate Linux kernel features for safety-critical systems.
- **Building a Collaborative Community:** Connect kernel developers and safety system producers to share insights.
- **Deepen Practical Knowledge:** Understand feature limitations, real-world performance, and areas for improvement.
- **Supporting Kernel Evolution:** Propose best practices and eventual kernel patches for better safety integration.

Achievements of 2024 (Overview)

- **Restart & Realignment:** Defined priorities and set a clear agenda after a significant restart phase.
- **Minimal Kernel Configuration:** Explored alternatives and selected the target platform for configuration work.
- **Collaboration with Other WGs:** Productive discussions, particularly with the architecture working group.
- **Linux Minimal Features (WIP):** Defined the problem, developed a methodology, and conducted initial investigations.

Minimal Kernel Configuration

- **Why Minimal Matters for Safety:** Reducing enabled features lowers complexity and the risk of failures.
- **Safety Requirements Across Industries:** Diverse needs (automotive, aerospace, healthcare) call for a common baseline with essential features.
- **Defining a Minimal Configuration:** Establishing a streamlined kernel that supports critical features without adding unnecessary complexity.
- **Role of Kernel Configuration:** Limiting enabled drivers and features improves safety by reducing potential failure points.

Minimal Kernel Configuration

- **Architecture is Key:** Kernel configuration depends on knowing the target architecture.
- **Exploring Alternatives:** Evaluated architectures (x86, ARM, RISC-V) and system types (physical vs. emulated).
- **Target Decision:** Selected **aarch64 QEMU** for its controlled testing environment and community accessibility.
- **Strategic Value:** QEMU (using virtio) has silicon implementation proposals, offering real-world hardware impact.

Linux Minimal Features Investigation - Intro

- **Architecture WG Initiative:** Investigation prompted by a key question from the Architecture WG.
- **Defining Core Requirements:** Identifying the essential kernel features to run even a basic program.
- **Guiding Kernel Configuration:** Helps strip down the kernel to improve simplicity and predictability.
- **Cross-Industry Relevance:** Provides a common feature set for safety-critical applications across various sectors.

Linux Minimal Features Investigation - Methodology

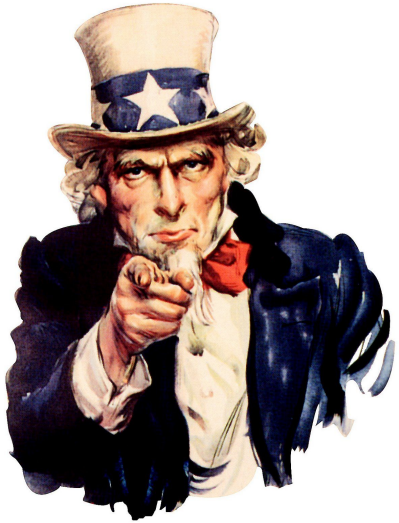
- **Exploring Approaches:** Manual code review, QEMU debugging, and dynamic tracing methods considered.
- **Challenges:** Manual investigation impractical; QEMU debugging: required extensive coding work.
- **Chosen Method:** Dynamic tracing using ftrace for minimal kernel interactions.
- **Why ftrace:** Lightweight, works out of the box but adds kernel complexity.
- **Custom Tooling:** Launcher program minimizes OS interference during tracing.

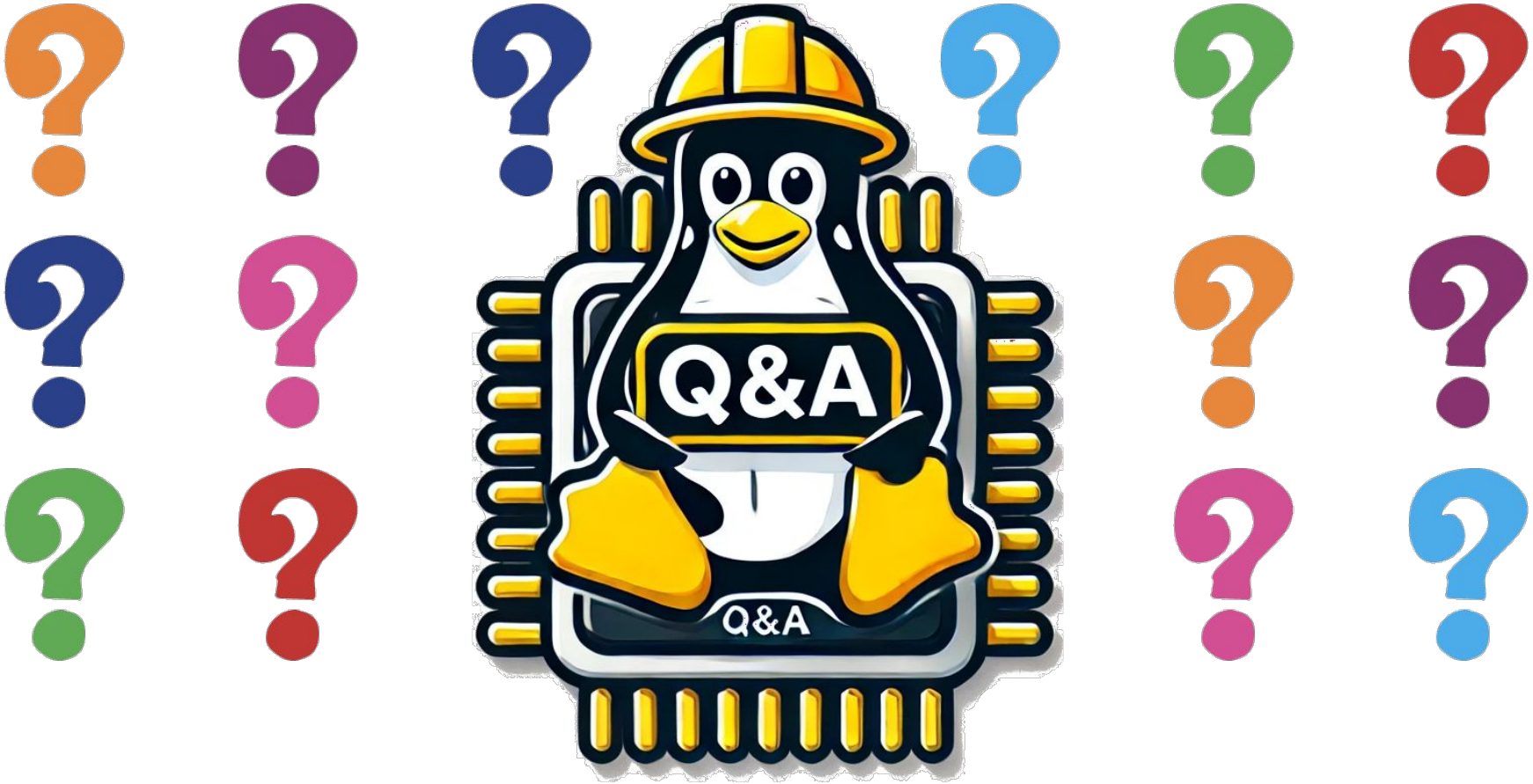
Linux Minimal Features Investigation - Results

- **Essential Kernel Mechanisms:** Identified key components required for minimal application.
- **Methodology Development:** Established a systematic approach for feature investigation.
- **Randomization vs. Predictability:** Balancing security-driven randomization with safety-critical system requirements.
- **Next Steps:** Further minimal config and feature probing needed; feature list to be published soon.

Next Steps

- **Deep Dive into Features:** Analyze existing features for complexity and optimization.
- **Collaboration Opportunities:** Leverage insights from tools and architecture working groups.
- **Original Agenda Completion:** Continue investigations on fundamental operations.
- **Open Invitation:** I WANT YOU
  - Meetings: bi-weekly Tue 13.00 CE(S)T https://elisa.tech/community/meetings/
  - Mailing list: https://lists.elisa.tech/g/linux-features
  - Linux Features git: https://github.com/elisa-tech/wg-lfscs




Q&A
Q&A

ELISA
Enabling Linux in
Safety Applications

Thanks