We found several issues linked inherently with ISA of RISC-V itself when using ftrace after turning on kernel preemption. In RISC-V, we must use 2 instructions to perform a jump to a target which is further than 4KB, and we cannot promise any 2 instructions being executed on the same process context if preemption is enabled. However, this is how we patch code in ftrace in current implementation. Thus, we proposed a change that could possibly solve it, making kernel preemption work with ftrace. The patch has been published on the [mailing list](mailto:). We would like to share and discuss our thoughts on LPC. And the talk will cover following content:

- Current Implementation of RISC-V ftrace
- How does stop_machine() work
- Reviews of ftrace Implementations on other Architectures
- Mixing with Kernel Preemption
- Limitation of RISC-V ftrace due to RISC-V ISA
- Possible Solutions to Enable ftrace with a Preemptible Kernel
- Proposed Solution
- Experiment and Results

I agree to abide by the anti-harassment policy
Yes

**Primary author:** CHIU, Tao

**Presenter:** CHIU, Tao

**Session Classification:** RISC-V MC

**Track Classification:** LPC Microconference: RISC-V MC