

A theorem for the RT scheduling latency (and a measuring tool too!)

Friday, 28 August 2020 07:00 (45 minutes)

Defining Linux as an RTOS might be risky when we are outside of the kernel community. We know how and why it works, but we have to admit that the black-box approach used by `cyclictest` to measure the `PREEMPT_RT`'s primary metric, the scheduling latency, might not be enough for trying to convince other communities about the properties of the `kernel-rt`.

In the real-time theory, a common approach is the categorization of a system as a set of independent variables and equations that describe its integrated timing behavior. Two years ago, Daniel presented a model that could explain the relationship between the kernel events and the latency, and last year he showed a way to observe such events efficiently. Still, the final touch, the definition of the bound for the scheduling latency of the `PREEMPT_RT` using an approach accepted by the theoretical community was missing. Yes, it was.

Closing the trilogy, Daniel will present the theorem that defines the scheduling latency bound, and how it can be efficiently measured, not only as a single value but as the composition of the variables that can influence the latency. He will also present a proof-of-concept tool that measures the latency. In addition to the analysis, the tool can also be used in the definition of the root cause of latency spikes, which is another practical problem faced by `PREEMPT_RT` developers and users. However, discussions about how to make the tool more developers-friendly are still needed, and that is the goal of this talk.

The results presented in this talk was published at the ECRTS 2020, a top-tier academic conference about real-time systems, with reference to the discussions made in the previous edition of the Linux Plumbers.

I agree to abide by the anti-harassment policy

I agree

Primary author: BRISTOT DE OLIVEIRA, Daniel (Red Hat, Inc.)

Presenter: BRISTOT DE OLIVEIRA, Daniel (Red Hat, Inc.)

Session Classification: LPC Refereed Track

Track Classification: LPC Refereed Track (Closed)