Contribution ID: **70** Type: **not specified**

Formalizing Kernel Synchronization Primitives with PREEMPT_RT

Monday 20 September 2021 10:00 (45 minutes)

In certain corners of the Linux Kernel, manual locking and lockless-synchronization primitives are developed instead of using the existing (and default) kernel locking APIs. This is obviously frowned upon, but still exists for historical reasons or because developers think that the subsystem in question is special enough to warrant manual synchronization primitives.

Adopting the PREEMPT_RT patch for mainline quickly exposes such cases, as these manual synchronization mechanisms are usually written with invalid assumptions regarding PREEMPT_RT that either negatively affect preemptibility or directly result in livelocks.

In the non-mainline parts of the PREEMPT_RT patch, such offending call sites were directly dealt with using "#ifdef" shortcuts. This can be an acceptable solution for an external patch, but for mainline the bar is much higher: the kernel is first surveyed for similar patterns in other subsystems and then, in cooperation with the locking subsystem maintainers, new official kernel locking mechanisms are created for such call sites.

This is better for the kernel ecosystem in general, and better for the call sites themselves. Using official kernel locking mechanisms comes with the benefits of reliability, thorough reviews, and lockdep validation.

In this session, four cases will be presented —from actual experiences while pushing some of the remaining parts of the PREEMPT_RT patch mainline: new sequence counter APIs, modified APIs for disabling tasklets, local locks, and generic patterns to avoid using the low-level in_irq/softirq/interrupt() macros in non-core kernel code.

The session will end with a discussion of some of the remaining manual locking mechanisms yet to be converted for proper mainline PREEMPT_RT inclusion.

I agree to abide by the anti-harassment policy

I agree

Primary author: DARWISH, Ahmed S.

Presenter: DARWISH, Ahmed S.

Session Classification: LPC Refereed Track

Track Classification: LPC Refereed Track (Closed)