Contribution ID: **188** Type: **not specified**

Auto-detecting sleeping lock calls in non-preemptible context via static analysis

Wednesday 18 September 2024 12:00 (45 minutes)

Calling sleeping locks in a non-preemptible context is not allowed because it causes a "BUG: scheduling while atomic" error. This problem is particularly relevant for PREEMPT_RT kernels, which convert all spin locks into sleeping locks. As a result, unexpected scheduling can occur in non-preemptible contexts. One way to detect this issue is by annotating such sleeping functions with might_resched(), which triggers a warning on PREEMPT_RT systems.

Although PREEMPT_RT has been around for a while, new bugs of this type continue to emerge from various subsystems. Given the straightforward nature of this bug, I developed a prototype static tool based on graph search called rtlockscope. This tool aims to scan the entire kernel source code for such issues. Rtlockscope is similar to Gary Guo's klint, which detects this problem in Rust code. However, unlike klint, rtlockscope cannot rely heavily on scheduling/preemption annotations because the Linux kernel code lacks them. Therefore, the autodetection must be more sophisticated, which is the primary challenge.

The current (unfinished) state of rtlockscope will be presented, along with some ideas for improving it.

Primary author: GLOZAR, Tomas (Red Hat)

Presenter: GLOZAR, Tomas (Red Hat)

Session Classification: LPC Refereed Track

Track Classification: LPC Refereed Track