Contribution ID: **175**   Type: **not specified**

Status Report: Broken Dependency Orderings in the Linux Kernel

Wednesday 14 September 2022 10:45 (45 minutes)

Potentially broken dependency orderings in the Linux kernel have been a recurring theme on the Linux kernel mailing list and even Linux Plumbers Conference. The Linux kernel community fears that with ever-more sophisticated compiler optimizations, it would become possible for modern compilers to undermine the Linux kernel memory consistency model when optimizing code for weakly-ordered architectures, e.g. ARM or POWER.

Specifically, the community was worried about address and control dependencies being broken, with the latter having already seen several unfruitful [PATCH RFC]'s on LKML.

This "fear" of optimizing compilers eventually lead to READ_ONCE() accesses being promoted to memory-order-acquire semantics for arm64 kernel builds with link-time optimization (LTO) enabled, leaving valuable performance improvements on the table as this imposes ordering constraints on non-dependent instructions.

However, the severity of this problem had not been investigated as of yet, with previous discussions lacking the evidence of concrete instances of broken dependency orderings.

We are pleased (or not so pleased) to report that, based on our work, we have indeed found broken dependency orderings in the Linux kernel. We would now like to open the discussion about, but not limited to, the severity of the broken dependencies we found thus far, whether they warrant dedicating more attention to this problem, and potential (lightweight or heavyweight) fixes.

I agree to abide by the anti-harassment policy

Yes

Primary authors: MAINAS, Charalampos (Technical University of Munich); ELVER, Marco (Google); FINK, Martin (Technical University of Munich); HEIDEKRÜGER, Paul (Technical University of Munich); Prof. BHA-TOTIA, Pramod (Technical University of Munich); Dr CHAKRABORTY, Soham (Delft University of Technology)

Presenters: ELVER, Marco (Google); HEIDEKRÜGER, Paul (Technical University of Munich)

Session Classification: Toolchains

Track Classification: Toolchains Track