OFence: Pairing Barriers to Find Concurrency Bugs in the Linux Kernel

Julia Lawall, Inria-Paris

September 14, 2023

1 Context

Memory barriers (also known as memory fences) make it possible to ensure that one set of memory operations completes before another set of memory operations. As such, in some circumstances memory barriers are sufficient to provide synchronization without the cost of locks. Nevertheless, the use of memory barriers makes code difficult to understand; except for possible informal comments, there is no indication of which memory operations must be on one side of the barrier or the other and there is no indication of how barriers in different parts of the code relate to each other.

The paper "OFence: Pairing Barriers to Find Concurrency Bugs in the Linux Kernel", published at EuroSys 2023, takes a first step in the understanding of memory barriers in the Linux kernel, by identifying common patterns of correct and incorrect barrier usage. Nevertheless, it has been possible to classify only part of the barriers in the Linux kernel according to these patterns, barrier usages remain difficult to understand, and no reliable strategy has been devised for detecting missing barriers.

2 This project

The goal of this project is to replicate and scale up the work carried out around barriers in OFence. Directions include the following:

- Replicate the identification of memory reads and writes that must be protected by barriers, as considered in OFence.
- Extend the analysis of OFence to cases where some reads and writes are protected not by barriers, but by other synchronization primitives that also have barrier semantics.
- Identify parts of the Linux kernel code where concurrency is possible, and where concurrency is impossible (typical of intialization functions).
- Use the information about where concurrency is possible to identify places where needed barriers are missing.
- Propose annotations or abstractions for kernel code that can make it clear where and why barriers are needed.

3 Caveat

Understanding code that uses barriers is very difficult. A good knowledge of the C language, of Limux kernel code, of program analysis, and of concurrency is required.

4 Supervision and Location

The project will be primarily supervised by Julia Lawall of the Whisper team at Inria Paris. Baptiste Lepers will participate in the supervision. The project can be carried out at Inria or remotely.

5 References

The OFence paper:

Baptiste Lepers, Josselin Giet, Willy Zwaenepoel, Julia Lawall:
OFence: Pairing Barriers to Find Concurrency Bugs in the Linux Kernel.
EuroSys 2023: 33-45
https://inria.hal.science/hal-04109096v1

A paper presenting a promising approach to identifying Linux kernel code where concurrency is possible:

Jia-Ju Bai, Julia Lawall, Qiu-Liang Chen, Shi-Min Hu:
Effective Static Analysis of Concurrency Use-After-Free Bugs in Linux Device Drivers.
USENIX Annual Technical Conference 2019: 255-268
https://inria.hal.science/hal-02182516v1