Contribution ID: **146** Type: **not specified**

System call wrappers for glibc

Friday 28 August 2020 09:35 (45 minutes)

Most programmers prefer to call system calls via functions from their C library of choice, rather than using the generic syscall function or custom inline-assembler sequences wrapping a system callinstruction. This means that it is desirable to add C library support for new system calls, so that they become more widely usable.

This talk covers glibc-specific requirements for adding new system call wrappers to the GNU C Library (glibc), namely code, tests, documentation, patch review, and copyright assignment (not necessarily in that order). Developers can help out with some of the steps even if they are not familiar with glibc procedures or have reservations about the copyright assignment process.

I plan to describe the avoidable pitfalls we have encountered repeatedlyover the years, such as tricky calling conventions and argument types, or multiplexing system calls with polymorphic types. The ever-present temptation of emulating system calls in userspace is demonstrated with examples.

Finally, I want to raise the issue of transition to new system call interfaces which are a superset of existing system calls, and the open problems related to container run-times and sandboxes with seccomp filters—and the emergence of non-Linux implementations of the Linux system call API.

The intended audience for this talk are developers who want to help with getting system call wrappers added to glibc, and kernel developers who define new system calls or review such patches.

I agree to abide by the anti-harassment policy

I agree

Primary author: WEIMER, Florian (Red Hat)

Presenter: WEIMER, Florian (Red Hat)

Session Classification: GNU Toolchain MC

Track Classification: GNU Toolchain MC