Contribution ID: **254** Type: **not specified**

The Clone Wars

Friday, 28 August 2020 10:20 (45 minutes)

Linux gained a new process creation system call clone3() in 2019 for the 5.3 release. It provides a superset and hopefully cleaner semantics than legacy clone().
I'd like to discuss a few things related to it:

- How to expose this safely to other libraries: various libraries in userspace want to make use of it to get access to new features such as CLONE_INTO_CGROUP (notably systemd for this one) and others. What are the adoption blockers? Can we sanely deal with deadlocking issues due to atfork handlers? Should we even expose the separate stack to userspace?
- Improving the stack handling: the legacy clone syscall exposes a stack (and on some architectures a stack size) argument to userspace. clone3() does this too because we didn't want to regress any use-cases so legacy-clone() callers could migrate to clone3(). There's a few differences though. clone3() requires a stack size argument to be passed and it doesn't require userspace to know in which direction the stack grows. Each architecture will do the right thing in the kernel instead. However, it still seems that we require userspace to do too much. When I look at what each clone() implementation is doing in the glibc source code in *pure assembly* my head starts spinning. How can we make this easier? Can we come up with a scheme that makes it almost trivial to use the stack argument in userspace?

I agree to abide by the anti-harassment policy

I agree

Primary author: BRAUNER, Christian (Canonical)

Presenter: BRAUNER, Christian (Canonical)

Session Classification: GNU Toolchain MC

Track Classification: GNU Toolchain MC