



Contribution ID: 276

Type: **not specified**

Pollable ptraced pidfds

Wednesday, 14 September 2022 15:00 (1h 30m)

When a process wants to ptrace a child without imposing unacceptable signal-handling latencies, it has to `waitpid()` on it, so that when a signal is received it is immediately detected and can be dispatched to the tracee. But if that process also wants to do anything else at all, it cannot be stuck in `waitpid()`: it must be able to go off and do that other work. So it must use `waitpid(WNOHANG)`. To avoid even worse latencies if that work takes time, it is best done in other threads (or processes). But ptracing is thread-specific, and only the ptracing thread can make changes in the traced process or receive information about it via `waitpid()`. So when the other work the process is doing needs any changes to be done in the traced child, the process must inform the ptracing thread of it, so that thread can make them.

Now if `ptrace waitpid()`s were pollable via `pidfds`, you could use one `poll()` in the ptracing thread to receive messages about work to be done in the traced child *and* get told about changes of state in the traced child that require attention from the ptracing thread. But right now that is impossible: `waitpid()` only wakes up polled `pidfds` if the `pid` is an actual child, not a tracee, and you can only `waitpid()` on `pids`, not `fds`. So the problem `pidfds` aim to solve, i.e. that there are two incompatible waiting systems, one for `pids` and one for `fds`, *still exists* for ptraced children. I think this should be fixed, and fixing it is not hard: but fixing it without breaking existing users of `pidfds` might be harder, since they won't expect ptraced children to wake up `poll()`ed `pidfds`, because they didn't historically. I have ideas and even working code, but I need some advice about how to make it upstream- ready: maybe it's acceptable to break compatibility in this minor way or maybe we need to do something cleverer. I don't know. Could anyone advise?

I agree to abide by the anti-harassment policy

Yes

Primary author: ALCOCK, Nick (Oracle Corporation)

Presenter: ALCOCK, Nick (Oracle Corporation)

Session Classification: BOFs Session

Track Classification: Birds of a Feather (BoF)