Contribution ID: **26**　　　　　　　　　　　　　　　　　　Type: **not specified**

pidfds: Process file descriptors on Linux

Wednesday, 11 September 2019 12:00 (45 minutes)

Traditionally processes are identified globally via process identifiers (PIDs). Due to how pid allocation works the kernel is free to recycle PIDs once a process has been reaped. As such, PIDs do not allow another process to maintain a private, stable reference on a process. On systems under pressure it is thus possible that a PID is recycled without other (non-parent) processes being aware of it. This becomes rather problematic when (non-parent) processes are in charge of managing other processes as is the case for system managers or userspace implementations of OOM killers.

Over the last months we have been working on solving these and other problems by introducing pidfds – process file descriptors. Among other nice properties, the allow callers to maintain a private, stable reference on a process.

In this talk we will look at challenges we faced and the different approaches people pushed for. We will see what already has been implement and pushed upstream, look into various implementation details and outline what we have planned for the future.

I agree to abide by the anti-harassment policy

I confirm that I am already registered for LPC 2019

Primary author: Mr BRAUNER, Christian

Presenter: Mr BRAUNER, Christian

Session Classification: LPC Refereed Track