Wrapping System Calls in Glibc

Maciej W. Rozycki <macro@wdc.com>
Senior Technologist, R&D Engineering

Sep 10th, 2019
Message and Request for Feedback

• Message from Florian Weimer
  – On behalf of the GNU C Library project

• The messengers
  – Dmitry V. Levin
  – Myself

• GNU Tools Cauldron 2019
  – Montréal, Canada, Sep 12-15, 2019
  – The GNU C Library BoF
Motivation

• Type checking
• Portability across Linux ports
• POSIX thread cancellation
• Debugging
Overview

• Intent is to have syscall wrappers in glibc
  – That are useful for general application use
  – Guidelines: [https://sourceware.org/glibc/wiki/Consensus#WIP:_Kernel_syscalls_wrappers](https://sourceware.org/glibc/wiki/Consensus#WIP:_Kernel_syscalls_wrappers)

• However missing ones may only be gradually added

• Syscall wrappers need documentation in the manual
  – Copyright assignment required with FSF

• The glibc project suffering from shortage of reviewers
Specific Notes

• No userspace syscall emulation, ENOSYS
• Architecture-independent wrapper names
• Multiplexing syccalls problematic
  – Type checking not possible when argument types vary
  – Variadic calls even more problematic
    • E.g. `open(2)/openat(2)` would not pass the `mode` argument with the newly-added `O_TMPFILE` flag
    • Cannot be called with a non-variadic function prototype on some architectures
  – Separate wrappers e.g. for individual `futex(2)` requests
• New syccalls added across all architectures in one kernel release
Interfaces

• Use `ssize_t` or `size_t` type respectively for buffer sizes in the wrapper
  – Even if the kernel uses `int` or `unsigned int`

• Avoid `long` type for flags

• Using `off64_t` * type for file offsets is fine
  – Plain `off64_t` is problematic

• Errors via `errno` and special return value
  – Except for `pthread_*` 

• Types and constants in a separate header for each syscall
Q&A
About the Messengers

• Dmitry V. Levin
  – The co-founder and the chief software architect of BaseALT
  – Contributor to the GNU C Library since 2001
  – A long time contributor to various free software projects, including strace, Linux kernel, Linux-PAM, and many others

• Myself
  – 20+ year supporter of free software movement
  – Contributor and reviewer for: Linux, GCC, glibc, binutils, GDB
  – x86 APIC, MIPS architecture, FDDI networking
  – Enthusiast of (running Linux on) DEC hardware: VAX, MIPS, Alpha
  – Now helping Western Digital with RISC-V effort