Redox – A µkernel-based Unix-like OS

Jacob Lorentzon

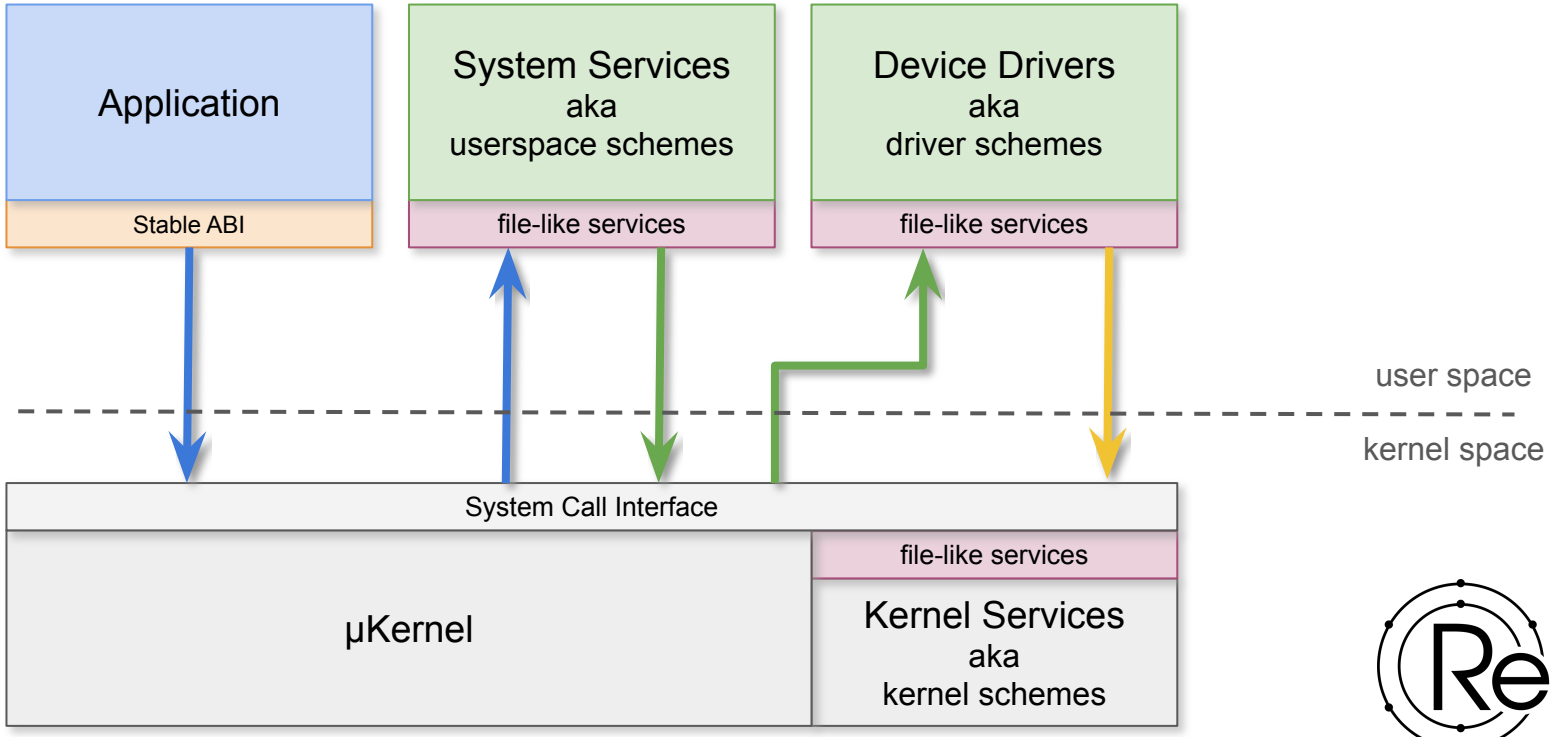
Redox OS

- Unix-like OS, on continuously shrinking µkernel
- Plan 9-inspired userspace filesystems
- Community-developed since 2015
- Written in Rust
  - Including our libc, relibc!
  - Some 3rd-party exceptions
- POSIX source-level compatibility
- Recent focus and progress on porting software
  - COSMIC apps
  - nushell
  - RustPython
  - GCC

Me

- Redox contributor since 2019
- Redox Summer of Code 2020..=2023
  - I/O
  - Userspaceification of fork/execv
  - Demand paging implementation
- NLnet project (2024-2025)
  - Userspace signal handling
  - Userspace process management

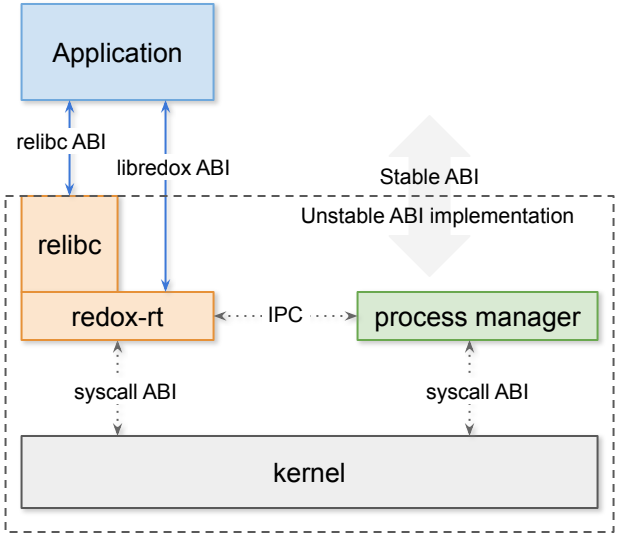
Architecture



Redox OS

Architecture

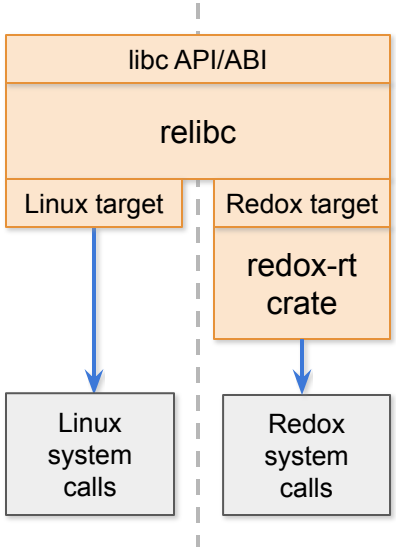
- File syscalls are handled by “schemes”
- Drivers run in separate userspace programs
- Kernel <30 kSLoC (~100k with dependencies)
- Syscall ABI intentionally kept unstable, shifting the stability layer to `redox-rt`
- POSIX and many other crucial parts moving incrementally to userspace libs



Redox OS

Relibc

- C library written in Rust
  - Even headers! (apart from macros, etc.)
  - Rustifying over time, reducing `unsafe {}`
- Redox and Linux support
- Focus on most of POSIX
- Source-level compatibility
- Two backends
  - `relibc` -> raw syscalls (Linux)
  - `relibc` -> `redox-rt` (Redox)



Redox OS

Status

- Increasing POSIX coverage, for porting
- Dynamically linked relibc/redox-rt is close
- Signals will allow moving more state to userspace
- Although ‘blocking’ bugs in FS and netstack,
- Self-hosting: cargo, gcc, and rustc themselves now mostly work!

Demo

Thanks for listening!

Questions?

Links

- https://redox-os.org/
- https://nlnet.nl/project/RedoxOS-Signals/
- https://fosdem.org/2025/schedule/event/fosdem-2025-5670-posix-signals-in-user-space-on-the-redox-microkernel/
- https://fosdem.org/2025/schedule/event/fosdem-2025-5973-redox-os-a-microkernel-based-unix-like-os/