TockWorld 7

Userspace Libraries Discussion

- Audience breakdown, ~70% C apps, ~40% Rust apps (have any)

- libtock-rs
  - Not much activity – perhaps b/c few public projects
    - limiting factor is relocations, make very difficult to use
    - → suggest: push for reloc. "in any way possible", even if toolchain does it?
    - "out of the question" to use two apps
  - Some ARM outreach from Rust core, answer is "we have reloc." ish
    - everything works until dynamic dispatch
    - possibly could get support if more ask
    - → FD-PIC ARM-specific option, superset of needs (e.g. shared libs)
    - → lowRISC working on ePIC standard, crawling
  - Can we use gcc somehow?
    - partial impl. exists to call into gcc backend.

  - Notwithstanding reloc., conclusion is that libtock-rs is fine?
    - MS hesitation – writing syscall binding twice (automation could help)
      - very used to C apps, no patterns/examples
    - libtock-c/-rs wrappers look very different
      - can "making C look more like Rust" help?
    - scopes issues?
      - embassy/futures incompatible with our -rs
      - no examples of async on libtock-platform
      - built on Pin abstraction, Johnathan has local test that is sound, but has 10% of overhead.
  - One effort is to make libtock-rs a "backend" for embassy
    - for teaching, aligns with embedded-hal
    - some summer interns working on this
    - Futures model does not allow creation of future with buffer/memory that future does not own
      - e.g. printing content of something else demands copy

(another test impl HIF disposal)

- What is "the bar" / the objective for libtock-rs
  - libtock-c has guide now
    - plus examples of let's track something
- Would separate -sync and -async impl's help?
  - probably, but for the overhead of duplicating low-level driver
  - autogenerator could alleviate this though.
- Could "syscall.json" lead to / enable autogen, in userspace and kernel?
- Adding a new driver hard b/c lockstep changes needed in kernel, -c, -rs; big barrier
- Without central / automated, changes don't always make it everywhere, i.e. Alarm syscall

---

- representative applications "upstream" / open
  - TPM2 is open source... non-trivial eng. to use w/ upstream Tock, but maybe worth it?
  - More active involvement in OpenSK
    - they have done a good job of freshly releases, but all volunteer time
  - Challenge: Getting 90% to "upstreamable" is there, but last 10% of small blockers (e.g. openSK patches...)
    - "is it useful if it is out of date"
  - Can we find more "fixed applications"
    - e.g. soil moisture board, uSD thread reboot / permanote
    - Challenge: Usually requires specialty hardware
  - Is there an equivalent to TODOMVC from WebDev?

(complex HW req's)

Breakout Candidates:
- syscall.json
- Flagship example (HW+SW)

- Is libtock-rs FFI to C reasonable?
  - TI SP does it
  - Works for isolated library type, less clear about C that know about Tock
  - libtock-rs safety really limits options here

Contrib's & Engagement

- Contrib pain points
  - Time Cost
    - not sure what you can do for our company to assign more time to upstreaming
    - can we as a project provide data etc. to help convince management
  - Specificity
    - local impl solves a problem, but not deemed generic enough (relive to upstream)
    - (same), time into approvals etc if will just be rejected
      - more clarity in what kind of things will be accepted
  - → Venue or mechanism to discuss early plans (pre-pre RFC, maybe design doc)
    - challenge: this becomes extra step that slows downstream development
    - would something more one-way work, not design discussion, but notification of this is what we are working on
    - (what exists for this now?
      - GH, slack, email Amit...
      - no clear answer
        - problematic as can't see whole picture
        - we are good at q&a reply here, GH less so
          - possibly evident by lower traffic volume here.
      - → Discord? (or other forum?)
        Zulip?
      - conversation-based medium effective, but 90-day slack is painful

- ENGAGEMENT
  - How did y'all find us?
    - Word of mouth in company
    - Search / explored for 100% Rust OS, Tock one of only options
      - * decent documentation
        - examples can be used
        - can get up & running quickly
  - How can we find you?
    - We are finding users via word of mouth
    - Ferros Sys. - We book cheap table at by events (trade shows) where likely to see knows you. Low-cost
      - → Venues?? .... crickets....
    - Hesitant to talk externally in early use
    - Tock Hackathon?
      - First time many took embedded systems. Similarly Rust.

Governance

Structure

- Silence as enthusiastic consent
- Should there be a 'kernel WG', hard to manage granularity
- Externally, can be hard to find relevant WG, default to core?
- Focused WG with clear conceptual mandate work well
- Core as fallback not necessarily bad, get centralized sense of unmet demand
- A passive WG not a bad thing as a means of establishing clearer, more fine-grained ownership
- Userland WGs?
- Orbit fan vs. RISC-V?
  - ↳ Start with overlapping groups, open door to eventual split

More focused WGs lower barrier to entrance, easier to ping local WG than worry about 'bothering all of core'.

FUNDING

Have a lot of internal test infra - we already have to use, more overhead here not necessarily desirable

Greg

MS — others need clearer corp-sec-reqts, but of interest.

Certification (at particular kernels?)

- ↳ often case-by-case, less about blanket cert - often can use, but proof of possible that "if you choose to, cert. is possible"
- ↳ challenge: C-oriented standards cannot map to Rust
  - foundation could help push standard changes
- ↳ challenge: for some companies, certification is the selling point, many find some externally self-interested parties pushing back
- ↳ challenge: rolling certs, every PiR - requires paperwork about why changes okay / safe -
  - → Eclipse foundation has cert. experience
    - ↳ bad funding experience w.r.t. chip vendors actually paying up funds
  - Starting ones? FIPS, Common Criteria
    - Euro: Any IoT must be 'safety-certified', CRA bringing hammer