QEMU Status Report

KVM Forum 2024

Paolo Bonzini, Red Hat

Distinguished Engineer

2023 highlights

- Accelerators
- Build system
- Retrocomputing
- New devices
- Cleanups
- Infrastructure
- Security

Red Hat

Accelerators

- Removed HAX

- TCG plugin support on Windows

- KVM

  - Xen-on-KVM supports PV console and network devices

  - SEV-SNP support

  - New architecture: loongarch

Build system

- Python virtual environment
  - Meson, Sphinx, Avocado all use the same Python interpreter
  - Easier to use "non-platform" Python on CentOS 8 and SLES 15
  - Python 3.8 required
- Automatic dependency download
  - pip
  - Meson subprojects (libfdt, libslirp)
- Functional tests without Avocado

Retrocomputing

- 68k improvements (Mac OS, A/UX, NetBSD, Linux)
- 64-bit HPPA
- Deprecation and removal of ancient ARM boards
- Removal of NiosII and CRIS targets
- Rewritten ESP SCSI device
- Fixes to x86 TCG for 16-/32-bit protected mode

New devices

- virtio-sound

- virtio-gpu rutabaga
  - Alternative to virglrenderer
  - Supports multiple GPU protocols over virtio
  - Wayland passthrough

Cleanups

- Final version of -audiodev/-audio
- Absence of host libraries can be used to disable boards
  - Not just devices
  - Example: libfdt

Infrastructure

- qemu.org and patchew moved to OSUOSL
- CI using Kubernetes runners on Azure
- New sponsorship: DigitalOcean!

Security

- More C compiler hardening (-fzero-call-used-regs, -ftrivial-auto-var-init=zero)
- Coverity runs via Gitlab pipeline
- <suspense>

What was next in 2020?

Red Hat

More gitlab?

- Static site generation ✅
- Primary repository ✅
- Release process
- Issue tracking ✅
- Wiki

Red Hat

More API, less command line?

- Extend -preconfig, allow configuration with QMP ✖
- "Official" bindings for QAPI ✖

Rust-y QEMU?

- "[RFC PATCH v1 0/6] Implement ARM PL011 in Rust" (June 2024)
- Lots of discussion, but a generally positive attitude
- No one seems to be scared of learning (more) Rust
  - Technical debt is a concern
  - Preserve expectations while tipping into a completely different ecosystem
- Merge early, iterate later

Build system integration

- cargo+make vs. rustc+meson
  - New Meson required
  - Requires handwritten meson.build for all dependent crates
  - Work upstream into making meson understand Cargo.lock
- Some level of cargo integration still useful for clippy?
- Cross-compilation
  - Procedural macros and their dependencies
- kconfig

Red Hat

Minimum supported Rust version

- Debian has 1.63.0
- Useful features from newer versions
  - C string literals, offset_of! (1.77.0)
  - Easier configuration of Clippy (1.74.0)
- None of these a blocker (offset_of! a bit harder)
- Might end up requiring 1.74.0+ for development

Red Hat

CI and linting

- clippy adds warnings on every new release
  - Good: A CI job that builds with nightly rust
  - Bad: broken CI every time a new warning is added
- Disable all lint groups, explicitly list desirable warnings
- Non-fatal CI job with lint groups enabled
- For every warning that happens in that job, decide whether to enable it

Red Hat

QEMU APIs in Rust

- Immediate goal: No undefined behavior
  - Only one &mut live at a single time
  - All callbacks should take &self
- Goal: Devices should use (mostly) *safe* Rust code
  - Wrap calls to C functions with Rust *bindings*
  - Error, QOM, character devices
- Lower priority goal: Devices can use *idiomatic* Rust code
  - Should come as a byproduct

Red Hat

Idiomatic Rust code

```
void pl011_realize(DeviceState *dev, Error **errp)
pub fn realize(&self) -> Result<(), qemu::Error>
```

- Automatically generate the *extern "C"* callbacks
- Wrap C types with conversion functions
- Automatic reference counting
- Type-safe casting
- …

Who to learn from?

- Mesa – early adopter of Rust with Meson
- glib – bindings, object system
- Linux – interoperability between C and Linux code

Red Hat

Thank you

in linkedin.com/company/red-hat

▶ youtube.com/user/RedHatVideos

f facebook.com/redhatinc

🐦 twitter.com/RedHat

Red Hat