Steering Xenomai into the Real-Time Linux Future

Jan Kiszka | Embedded Linux Conference, March 13, 2018

Agenda

Xenomai – what is this again?

Do we still need it?

Looking back & current status

Midterm changes to come

Architectural outlook

Summary

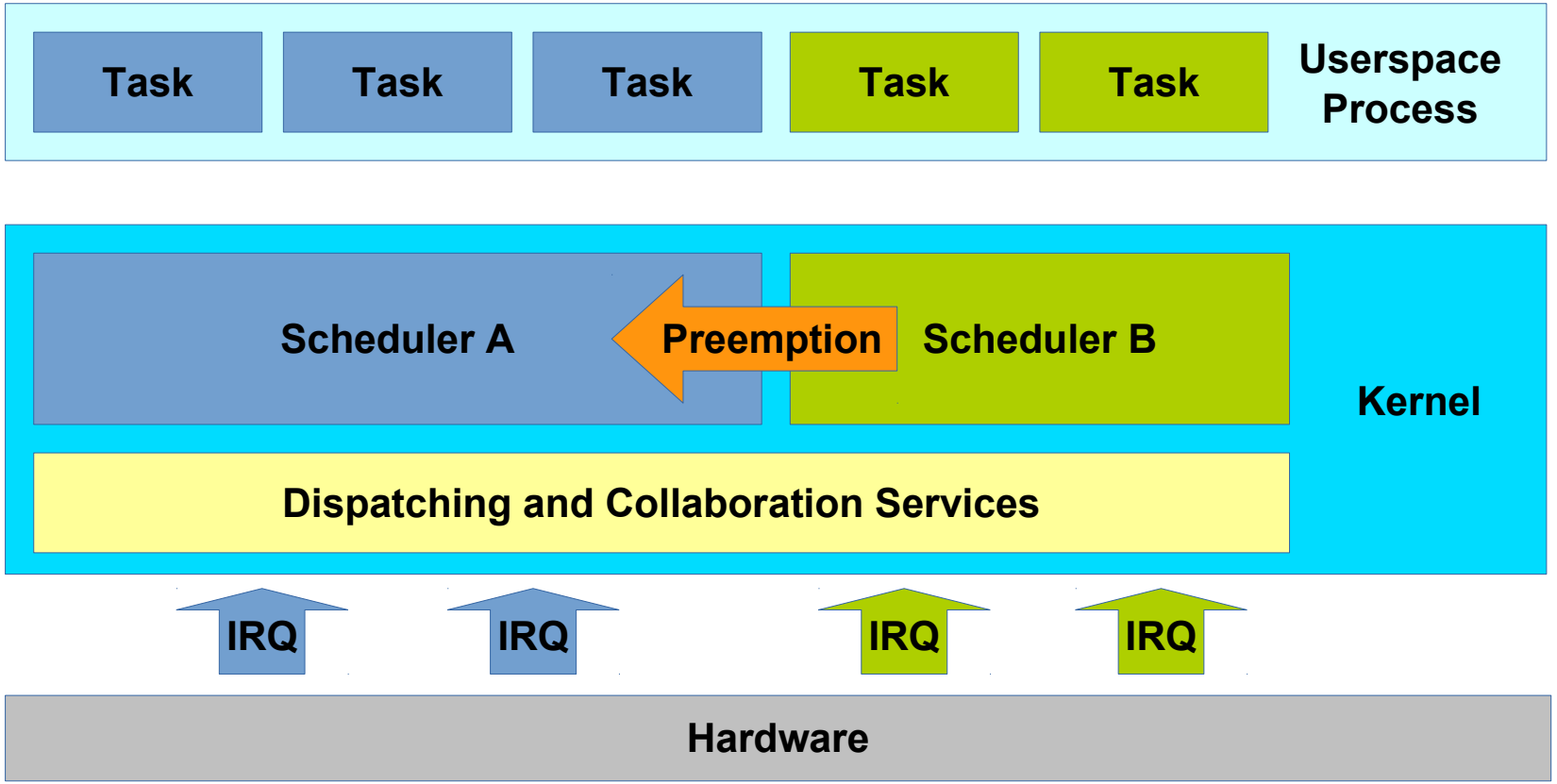
Xenomai is an RTOS-to-Linux Portability Framework

It comes in two flavors

- as **co-kernel** extension for (patched) Linux
- as libraries for **native Linux** (including PREEMPT-RT)

It’s also the only remaining product-grade co-kernel for Linux

Co-Kernel in a Nutshell



When do you want a co-kernel?

Accurate modeling of legacy RTOS behavior

- Non-POSIX scheduling policies
- Special APIs with POSIX-incompatible impact (specifically on scheduling)

Strong separation of RT and non-RT code paths

- Avoid tricky configuration (build-time & runtime) for stable PREEMPT-RT deployments
- Large RT code base, 3rd-party libraries, not only Linux/RT-aware developers
- Use RT → non-RT switch signaling to preserve RT architecture

Latency and/or performance concerns

- PREEMPT-RT == more context switches
- Problematic on low-end hardware or co-located high-throughput workload

When do you NOT want a co-kernel?

None of the aforementioned reasons apply

- RT turns out to be a soft requirement
- PREEMPT-RT is “good enough”
- Application’s RT architecture is manageable

You are concerned about co-kernel integration & maintainability

- Good point, more about this later...
- Just keep in mind: PREEMPT-RT requires maintenance as well, out-of-tree as well as in-tree

Xenomai History

Xenomai 1.0

- Announced in 2001 – as portability framework for RTOS applications
- Required a real-time basis
- Development of ADEOS layer for Linux and RTAI
- Merged with RTAI => RTAI/fusion

Xenomai 2.0

- Departed from RTAI in 2005 – incompatible design goals
- Evolved ADEOS to I-pipe layer (also used by RTAI)
- Ported to 6 architectures

Xenomai 3.0

- Released in 2015 after >5 years of development
- Rework of in-kernel core (now POSIX-centric)
- Support for native Linux

Xenomai Applications

- **Machine / motion control systems, PLCs**
- **Printing machines** (manroland)
- **Printers / copying machines** (Xerox)
- **3D printers** (see talk on Wednesday)
- **Network switches** (e.g. Ruggedcom)
- **Magnetic resonance tomographs** (Siemens Healthcare)
- **OROCOS** (OSS robotics framework)
- **Robotic research projects**
- **Manned spaceflight software reference platform** (NASA)
- **…**

The Shadow User Base

- **Many more applications in the shadow**
- **Some known to the maintainers (e.g. autonomous logistics vehicles), some suspected**
- **Embedded World Exhibition 2018 (Nuremberg, Germany)**
  - Service providers advertising Xenomai support
  - NXP offers BSPs, demonstrates TSN
  - Known to many hardware and service companies
- **Over 700 subscribers on the mailing list**
- **So... a healthy project?**

Key Contributors of the Last 5 Years

- **Philippe Gerum** → **customer-funded** + **personal budget**
- **Gilles Chanteperdrix** → **spare-time**, partly **employer-funded**
- **Jan Kiszka** → **employer-funded**
- **Jorge Ramirez-Ortiz** → employer-funded, partly spare-time
- **Henning Schild** → employer-funded
- **Dmitriy Cherkasov** → customer-funded, now spare-time

Gilles Chanteperdrix, 1975-2016

He is sorely missed.

Xenomai in Danger

- **Too much work for a single maintainer with few contributors**
- **At Siemens, discussion started in 2017**
  - Migrate or invest?
  - Coordinated effort of internal users
  - Decision: **invest into Xenomai**
- **“RTnet, Analogy and the elephant in the room”**
  - External call for contributions
  - Raised awareness, brought in first commitments

“New year, new roles”

- **Philippe will step back from project lead**
  - continues to support with reviews and specific tasks
  - concentrates on new co-kernel architecture (more later)
- **/me will take over project lead**
- **Switch to take place summer..autumn this year**
- **New I-pipe work-split**
  - ARM: Philippe Gerum
  - ARM64: Dmitriy Cherkasov
  - PPC32: Steven Seeger
  - x86: Jan Kiszka
  - Integration: Philippe

Maintaining the I-pipe Kernel Patch

- **Out-of-tree is always a challenge**
- **Only limited support feasible, currently**
  - 4.4 (all archs, may be limited to x86 & ARM)
  - 4.9 (all archs, pending issues on x86)
  - 4.14 (no x86 yet, adaptations for FPU changes needed)
  - Stable updates can lag behind
- **Discontinued archs: NIOS2, SH, Blackfin, PPC64, ARM < v7**
- **4.14: I-pipe now patch queue of logical increments** (easier to base on own trees)
- **New policy: only maintain patches for latest LTS**
  - Additional kernels depending on contributors
  - Siemens will offer 4.4, soon based on CIP SLTS kernel

Xenomai 3.0 & 3.1

- **Xenomai 3.0.7 (current stable) to be released soon**
  - RTnet fixes merged
  - Last topic: I-pipe patch updates
- **Criteria for 3.1 release not yet set in stone**
  - Will introduce ARM64
  - Several core improvements like fast prio-ceiling mutexes and fast setscheduler
  - 4.14 shall be supported for all target archs
- **Reminder: Xenomai 2 is UNMAINTAINED!**

Xenomai Driver Stacks

- **RTnet: refreshed recently (3.0.7), but needs more love**
  - Drop old drivers, refresh current ones
  - Rethink core for upcoming **TSN** architecture
- **UART, GPIO, SPI and CAN currently look good**
- **Analogy (analogue I/O) is orphaned**

- **We need driver subsystem maintainers!**
- **We'll drop unmaintained / broken drivers** (after prior warning)

Improving the Infrastructure

- **Project hosting will switch to gitlab.denx.de**
  - **`philippes_tasks--`** (many thanks to Denx!)
- **Private vs. public CI**
  - Denx may provide private (maintainers-only) CI with test farm integration
  - Public CI allows reuse by contributors, but limited to qemu tests
  - Idea: try Travis CI, **contributors welcome!**
- **On-device testing**
  - We will define reference boards
  - Manual distributed testing, **contributors welcome!**
  - Test farm needed, central (Denx?) or distributed?
  - Copy distributed LAVA deployments of AGL, CIP etc.?

And now for something completely different?

Dovetail, Steely – Rethinking Co-Kernels for Linux

- **Goals**
  - Improve integration of co-kernel enabling with Linux kernel
    (in contrast to I-pipe / Xenomai: abstract kernel away)
  - Further simplify maintenance (as long as out-of-tree)
  - Provide a chance to upstream
- **2 Elements**
  - Dovetail - interrupt routing, co-kernel hooks ~= I-pipe
  - Steely - co-kernel implementation ~= Xenomai Cobalt
- **Ongoing development**
  - **Not** Xenomai-compatible
  - **Not** product-ready
  - Do **not** use to fly to Mars

Looking into Dovetail

- **Interrupt pipeline**
  - Prioritize selected interrupts (some can become “NMIs”, doing “out-of-band” work)
  - Solely builds upon irqchip abstraction
  - Reuses existing locking (even lockdep works)
- **Task steeling**
  - Remove Linux task from standard scheduler
  - Return it again
- **Kernel event propagation to co-kernel**
  - Syscalls
  - Faults
  - Signals
- **Check out git.xenomai.org/linux-steely.git**
- **Take a look at Documentation/dovetail/***

Steely – The In-Tree User of Dovetail

- **POSIX-compatible RTOS core**
- **Demonstrates in-tree usage of Dovetail interfaces**
- **Fundamental rework of Xenomai Cobalt**
  - More fine-grained locking (Xenomai 3: single core lock)
  - Scalable (>6 cores)
  - Uses standard clocksources
  - CPU frequency changes supported
  - Compatibility to Xenomai could be added once matured
- **Open issue: real-time drivers**
  - Only minimal set available
  - How to make upstream drivers co-kernel aware?
- **Userspace: git.xenomai.org/steely.git** (usage very similar to Xenomai 3)

Status

- **Redesign shrunk code base significantly**
  - I-pipe → Dovetail: ~ 50%
  - Xenomai → Steely: < 50%
- **347 files changed, 53661 insertions(+), 594 deletions(-)**
  - Dovetail: ~14 files changed, 3828 insertions(+)
  - Steely: ~113 files changed, 41955 insertions(+)
  - Rest is hooks in / adaptions of existing code
- **Works on ARM i.MX 6 and 7, ARM64 WiP**
- **Again: ongoing construction work, rebasing, not ready for production use**
  - But ready for a try

Summary

- **The Co-kernel is here to stay**
  - Used in production for >20 years
  - Valid use cases aside PREEMPT-RT
- **Industrial usage of real-time Linux has a problem**
  - Unhealthy imbalance between give and take
  - Not only Xenomai is suffering from this
- **The Xenomai project is alive and kicking – but needs more active users**
  - Stand up and provide feedback, publicly!
  - We would welcome more core hackers...
  - ...but we have plenty of other tasks as well

Any Questions?

Thank you!

http://xenomai.org

Jan Kiszka <jan.kiszka@siemens.com>
Xenomai mailing list <xenomai@xenomai.org>