

Xenomai brief introduction



Paulo Pedreiras
pbrp@ua.pt

DETI/IT/University of Aveiro

Real-Time Operative Systems Course Unit
Oct/2021

Outlook



- What is Xenomai
- Architecture
- API
- Developing an application
- Summary and info sources

What is Xenomai?

- Xenomai is a RTOS-to-Linux Portability Framework
- It now comes in two flavors
 - As a dual-kernel / co-kernel extension for (patched) Linux
 - More stringent real-time requirements
 - More complex installation
 - As libraries for native Linux (including PREEMPT-RT)
 - Less predictable
 - Can be used on standard Linux

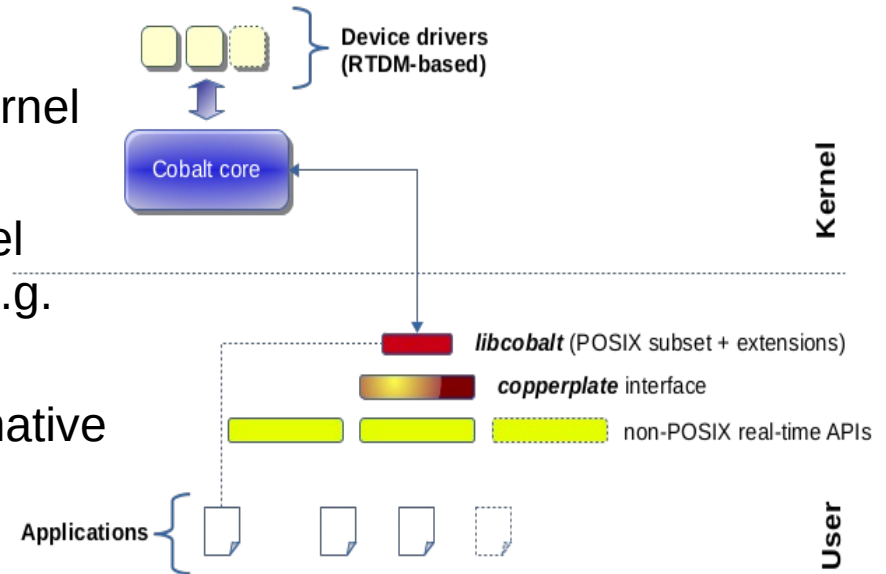
For what is it used?

- Many areas:
 - Machine control systems, PLCs
 - Printing machines (manroland)
 - Printers / copying machines
 - Network switches (e.g. Ruggedcom)
 - Magnetic resonance tomographs (Siemens Healthcare)
 - OROCOS (OSS robotics framework)
 - Many robotic research projects
 - And many others ...

Xenomai architecture

- Co-kernel: Cobalt

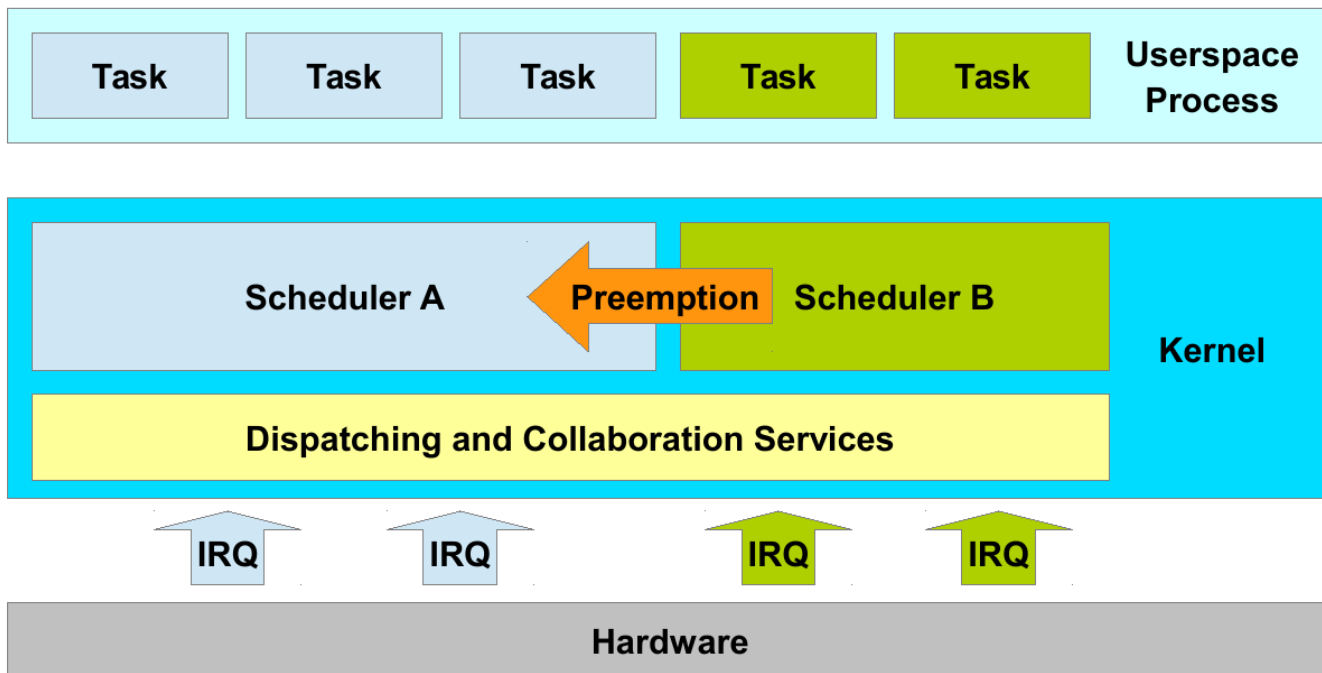
- Supplements Linux with a real-time co-kernel running side-by-side with it.
- Cobalt extension built into the Linux kernel and deals with all time-critical activities (e.g. interrupts, scheduling)
- The Cobalt core has higher priority than native kernel activities
- Best real-time performance, but requires kernel patch (Adeos) and compilation



https://gitlab.denx.de/Xenomai/xenomai/-/wikis/Start_Here

Xenomai architecture

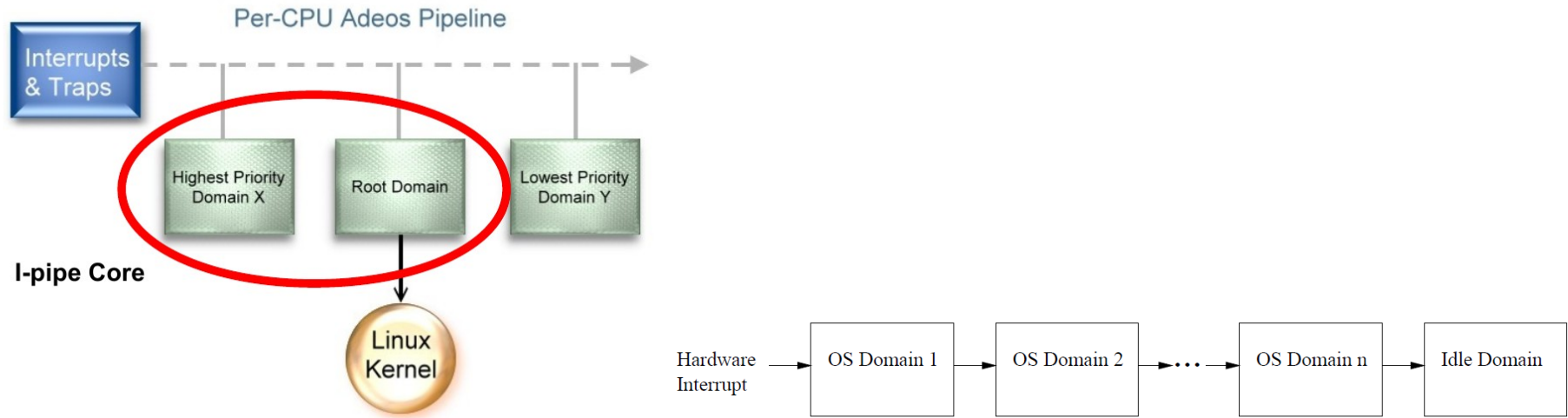
- What is a co-kernel architecture



Xenomai architecture

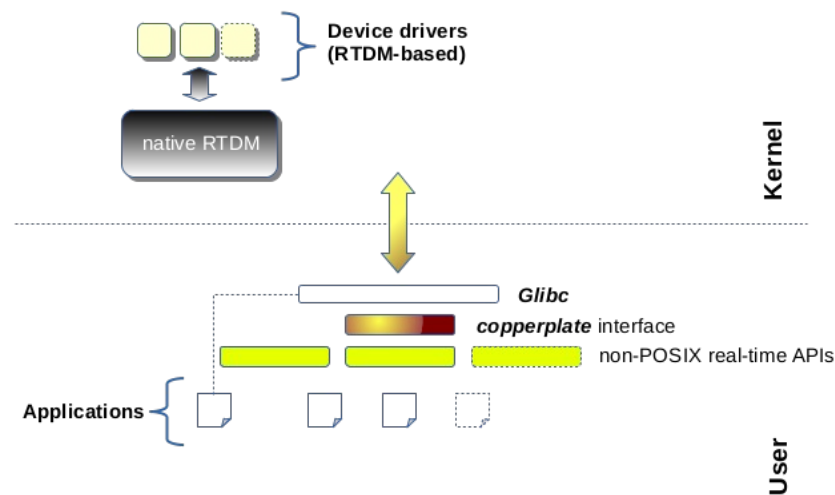
- What is a co-kernel architecture

- Based on Adeos: nanokernel hardware abstraction layer (HAL), or hypervisor, that operates between computer hardware and the operating system



Xenomai architecture

- Single (Linux) kernel: Mercure
 - Uses only the native real-time capabilities of the native Linux kernel
 - PREEMPT-RT enhances the real-time behavior
 - Not mandatory
 - Depends on app requirements with respect to latency/jitter
 - All the non-POSIX RTOS APIs that Xenomai provides are emulated over the native Linux threading
 - “Skins” are also supported



https://gitlab.denx.de/Xenomai/xenomai/-/wikis/Start_Here

Xenomai architecture

- Task migration: Cobalt only
 - Every cobalt thread is also a Linux task
 - Thread states are shared/synchronized among domains
 - Only one can run at a time
 - Migration to RT:
 - On syscall `xnshadow_harden()`
 - Linux task suspended; Cobalt thread resumed
 - Migration to Linux
 - On syscall `xnshadow_relax()`, or fault/trap, signal
 - Cobalt thread suspended; Linux task resumed

Xenomai architecture

- Real-Time Drive Model
 - Goals and principles
 - Provide environment for co-kernel real-time drivers
 - Service interface towards applications and other drivers
 - Low-level primitives from implementing drivers
 - Linux for non-RT purposes: setup / shutdown, resource discovery and claiming, etc.
 - Two types of RTDM devices
 - Character device (open/close, read, write, ioctl)
 - Protocol device (socket, bind, send, recv, etc.)

Xenomai API

- Vision and goals
 - Run API emulation over standard Linux/POSIX
 - Enable seamless migration between co-kernel and native Linux deployments
- Three real-time APIs/skins available
 - VxWorks
 - pSOS
 - **Alchemy (former “native skin”) (we will focus on this one)**

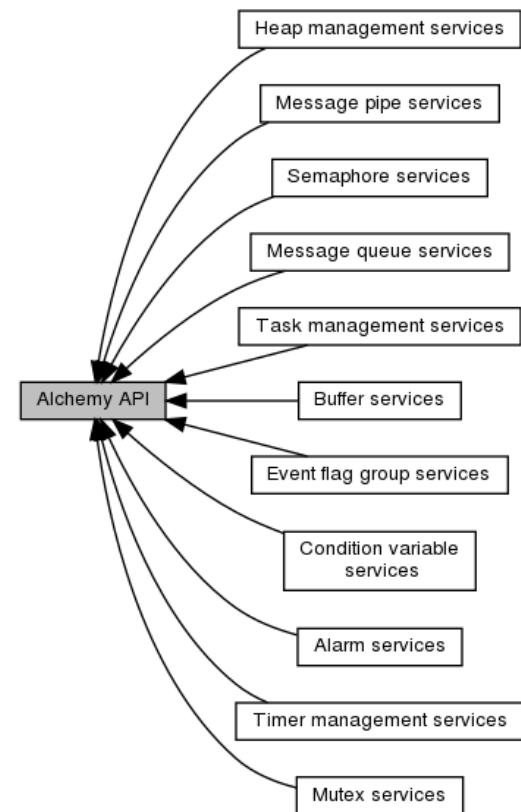
Xenomai API

- Documentation

- https://xenomai.org/documentation/xenomai-3/html/xen03prm/group__alchemy.html

- Rich API

- Task management (create, delete, ...)
 - IPC and sync:
 - message queues, semaphores, mutex, ...
 - Alarms
 - Condition variables
 - ...



Xenomai: developing an application

- **Includes and main()**

```
#include <stdio.h>
...
// Alchemy API (former native skin)
#include <alchemy/task.h>
#include <alchemy/timer.h>
...
/* Main*/
int main(int argc, char *argv[]) {
    ...
    /* Create RT tasks */
    err=rt_task_create(&task_a_desc, "Task a",...);
    err=rt_task_create(&task_b_desc, "Task b",...);
    ...
    /* Start RT task */
    rt_task_start(&task_a_desc, &task_code,...);
    rt_task_start(&task_b_desc, &task_code,...);
    ...

    /* wait for termination signal */
    Loop forever/until termination event
    return 0;
}
```

Xenomai: developing an application

- **Task structure**

```
void task_code(void *args) {  
    ...  
    /* Initializations */  
    /* Set task as periodic (if suitable)*/  
    err=rt_task_set_periodic(NULL, TM_NOW, taskArgs->taskPeriod_ns);  
    ...  
    /* Task cyclic code  
    for(;;) {  
        err=rt_task_wait_period(&overruns);  
        if(err) {  
            Handle error (print message, terminate, ...)  
            break;  
        }  
    }  
    return;  
}
```

Xenomai: developing an application

- **Compiling and linking**

```
# Standard Makefile to build Xenomai 3 applications, with alchemy skin
# Use xeno-config to get the correct compile and link flags
XENO_CONFIG := /usr/xenomai/bin/xeno-config
CFLAGS := $(shell $(XENO_CONFIG) --skin=alchemy --cflags)
LDFLAGS := $(shell $(XENO_CONFIG) --skin=alchemy --ldflags)
CC := $(shell $(XENO_CONFIG) --cc)

EXECUTABLE := periodicTask

all: $(EXECUTABLE)

%: %.c
    $(CC) -o $@ $< $(CFLAGS) $(LDFLAGS)

clean:
    rm $(EXECUTABLE)
```

Summary

- Xenomai
 - Cobalt vs Mercure
 - Cobalt: High determinism, but complex set-up, platform limitation (due to Adeos),...
 - Mercure: Lower determinism, but works on standard Linux
- Skins/APIs that emulate for diverse RTOS
 - VxWorks, pSOS, Alchemy
 - Facilitates application porting
 - Rich set of RT services (task scheduling, IPC, ...)

Info sources

- Overview and Architecture
 - https://gitlab.denx.de/Xenomai/xenomai/-/wikis/Introducing_Xenomai_3
 - <https://elinux.org/images/7/76/Kiszka.pdf>
- Installation
 - <https://xenomai.org/documentation/xenomai-3/html/README.INSTALL/>
- Application development
 - <https://xenomai.org/documentation/xenomai-3/html/xeno3prm/index.html>
 - https://gitlab.denx.de/Xenomai/xenomai/-/wikis/Building_Applications_For_Xenomai_3
 - <https://www.ashwinnarayan.com/post/xenomai-realtime-programming/>
 - <https://www.ashwinnarayan.com/post/xenomai-realtime-programming-part-2/>
- Other
 - <https://www.diva-portal.org/smash/get/diva2:1251188/FULLTEXT01.pdf>