The Xenomai Project

http://freesoftware.fsf.org/projects/xenomai/

The Open Group Conference

Paris, April 2002

Philippe Gerum, rpm@xenomai.org

Project ID

- What is Xenomai?
  - A GNU/Linux-based real-time framework
  - A foundation for writing real-time interfaces
- What are the main features?
  - A collection of (traditional) RTOS API emulators
  - A scalable run-time system (timeliness requirements)
  - A simulation system aimed at debugging tasks

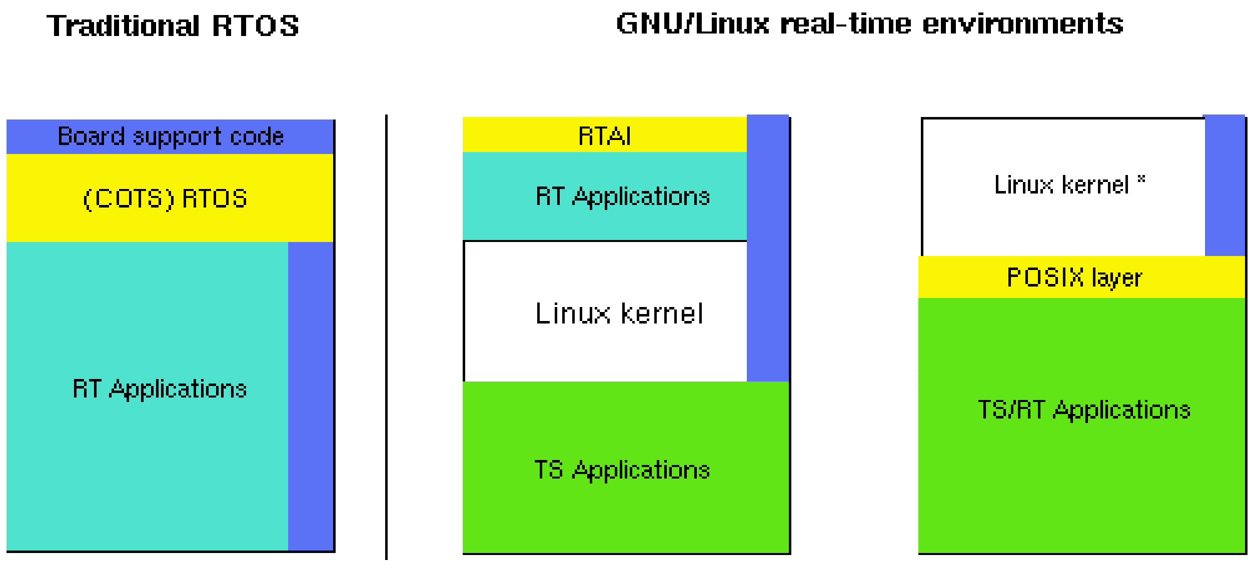
Why Xenomai?

- Help for migration of real-time systems
  - Versatility of GNU/Linux
  - Commonality of traditional RTOS features
- Integrate the emulation & simulation approaches
  - Prototyping and first order port
  - RTOS-awareness of debugging support
  - Early development stages made simpler
  - Test harnessing

The real-time infrastructure

- The traditional RTOS layout
  - Small and specific “h/w glue” code (i.e. BSP)
  - COTS (nearly) platform-independent kernel
- The real-time Linux duality
  - Supervisor-mode executive, and/or
  - Standard Linux kernel

Compared infrastructures



* Additional FIFO scheduling and/or fine-grain preemption support

🟦 Hardware control layer
🟨 Service layer

A basic view of a RTOS

- The hardware control layer
  - Manages parts of the bare silicon
  - Handles the external events
- The software service layer
  - Creates, manages and synchronizes tasks
  - Implements a set of programming facilities
  - Provides a kernel API

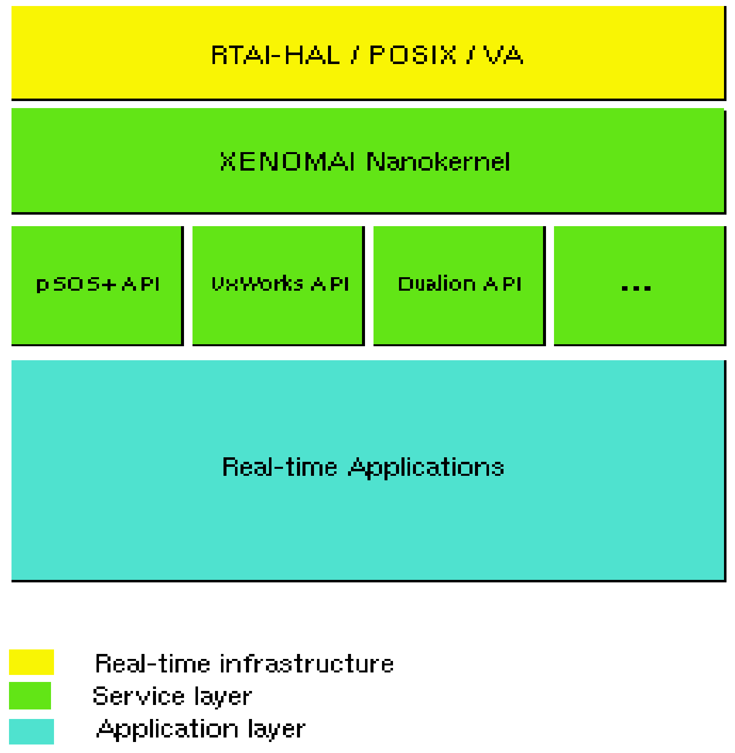
The software service layer

- The kernel code and its interface are usually highly integrated. For instance, RTAI's **rtai_sched** module directly exports to the application layer its own implementation of:
  - Preemptive scheduling
  - Inter-tasks synchronization and messaging
  - Precision timers
  - Memory allocation (optionally)

The Xenomai approach

- The hardware control is left to a real-time host infrastructure (e.g. RTAI-HAL).
- A splitted view of the software service layer
  - A single nanokernel providing generic services
  - Any number of real-time interfaces using these services
- A small and well-defined interface between the real-time infrastructure and the nanokernel

A layered view of Xenomai

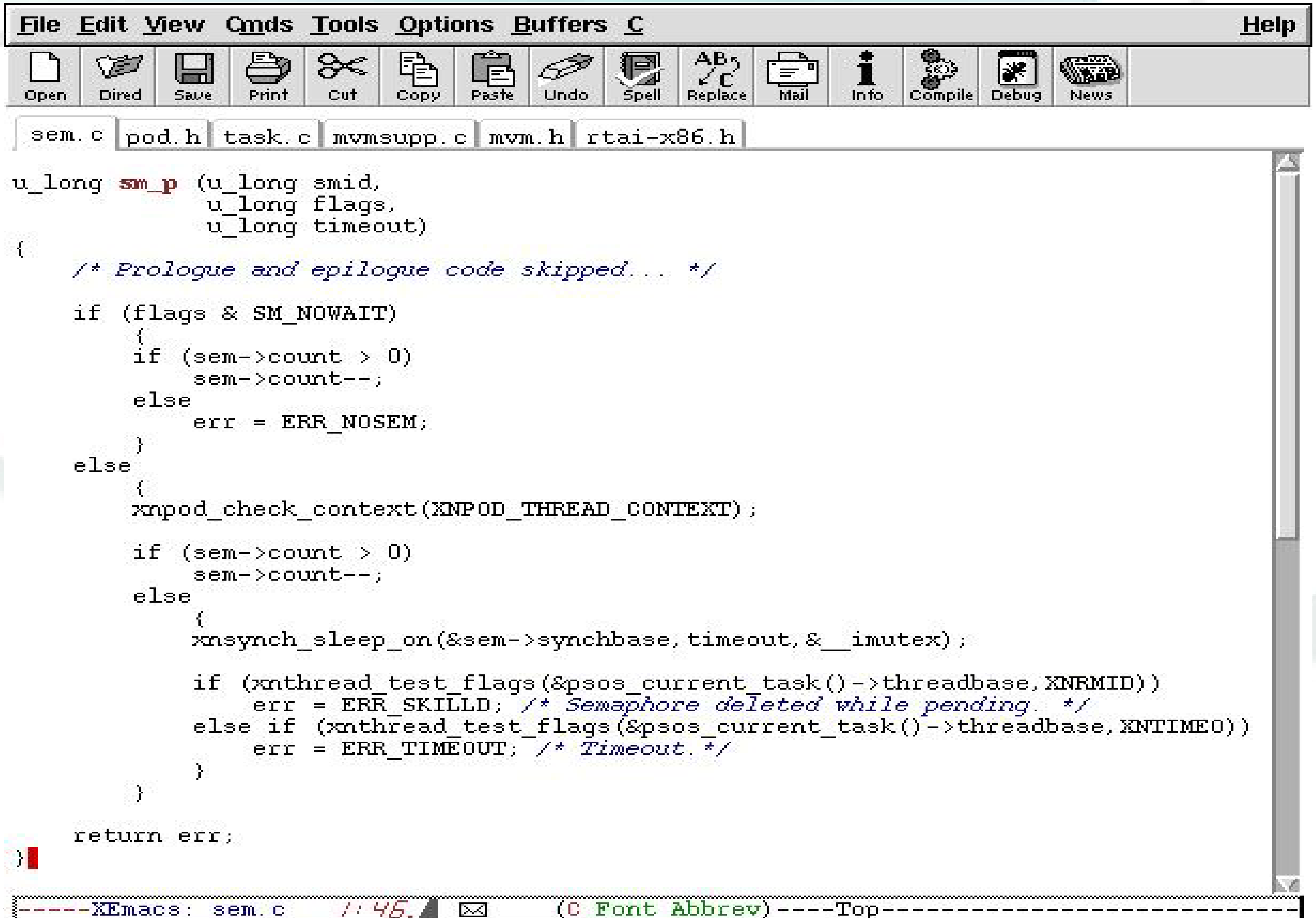


The Xenomai nanokernel

/ Offers generic real-time services on top of a real-time infrastructure:

- Fixed-priority, FIFO, preemptive multi-threading
- Thread synchronization
- Timer and clock management
- Memory allocation (bounded worst-case time)
- Interrupt and signal management

A simple code fragment...


File Edit View Cmds Tools Options Buffers C Help
Open Dired Save Print Cut Copy Paste Undo Spell Replace Mail Info Compile Debug News
sem.c pod.h task.c mvmsupp.c mvm.h rtai-x86.h

```
u_long sm_p (u_long smid,
             u_long flags,
             u_long timeout)
{
    /* Prologue and epilogue code skipped... */

    if (flags & SM_NOWAIT)
        {
        if (sem->count > 0)
            sem->count--;
        else
            err = ERR_NOSEM;
        }
    else
        {
        xnpod_check_context(XNPOD_THREAD_CONTEXT);

        if (sem->count > 0)
            sem->count--;
        else
            {
            xnsynch_sleep_on(&sem->synchbase,timeout,&__imutex);

            if (xnthread_test_flags(&psos_current_task()->threadbase,XNRMID))
                err = ERR_SKILLD; /* Semaphore deleted while pending. */
            else if (xnthread_test_flags(&psos_current_task()->threadbase,XNTIMEO))
                err = ERR_TIMEOUT; /* Timeout.*/
            }
        }

    return err;
}
```

```
-----XEmacs: sem.c      /:46.      (C Font Abbrev)----Top-------------------------------
```

Advantages

- Selectable real-time infrastructure
  - Depends on the required degree of timeliness
    - RTAI's HAL for hard real-time
    - LinuxThreads (POSIX 1003.1c) for soft/firm real-time
  - Seamless simulation support using the Minute Virtual Machine
    - Runs Xenomai's nanokernel any client interface
    - Runs the original RTAI's uniprocessor scheduler
    - Comes with a RTOS-aware graphic debugger

Advantages (2)

- A common base for porting real-time interfaces
  - Traditional RTOS API emulators
  - Home-grown interfaces
- Behavioral compatibility with traditional RTOS
  - Straightforward implementation of emulators
  - Simplified application port

Xenomai's current status

- A single nanokernel running on top of multiple real-time infrastructures
  - RTAI's HAL for hard real-time
  - LinuxThreads for soft real-time
  - Minute Virtual Machine (MVM) for simulation
- API emulators for pSOS+, VxWorks, VRTXsa
- uITRON-compliant API
- Dualion experimental API

Next (planned) steps

- Documentation effort
- LTT port
- Enrich the RTOS emulator collection

The Xenomai Project

http://freesoftware.fsf.org/projects/xenomai/

The Open Group Conference

Paris, April 2002

Philippe Gerum, rpm@xenomai.org