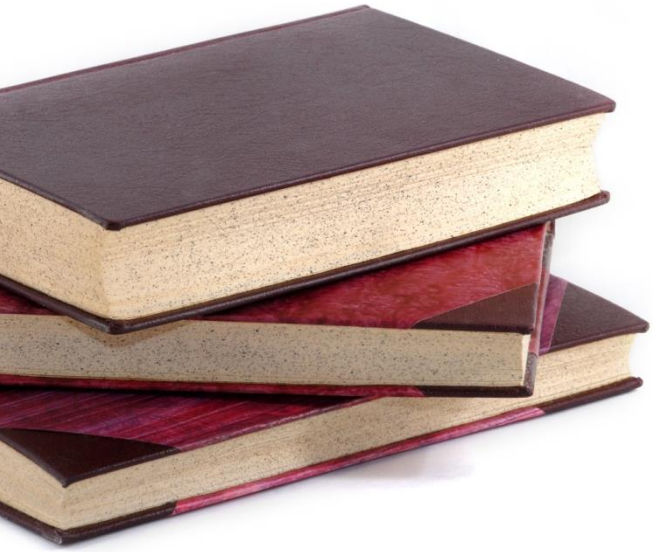
MPSIT

Lecture 7



7

Real-time

22 noiembrie 2016

Exam questions

1. Define the ADT elements
2. Toaster is:
   a) A graphical interface
   b) A web interface
   c) A Yocto Project external component
3. From which framework was wic inspired
4. Opkg represents:
   a) An embedded devices permissions escalation tool
   b) An embedded devices management system
   c) An embedded devices package management system

Exam answers

1. A cross-toolchain, user-space tools, the qemu environment, Eclipse IDE
2. B
3. Mic – Meego Image Creator
4. C

Outline

- GPOS vs RTOS
- PREEMPT_RT
- Yocto Project –rt kernel
- Linux real-time apps
- Benchmarking
- Meta-realtime

Recap: external tools

- Wic: an external solution to a Bitbake internal limitation
- Swabber: stable but not relevant components
- LAVA: external community automation framework
- Build Appliance: Yocto Project VM
- Matchbox: Handheld devices X11 environment solution
- Extra: opkg, pseudo, eglibc, cross-prelink are the Yocto Project contributions to the embedded Linux community

Outline

- GPOS vs RTOS
- PREEMPT_RT
- Yocto Project –rt kernel
- Linux real-time apps
- Benchmarking
- Meta-realtime

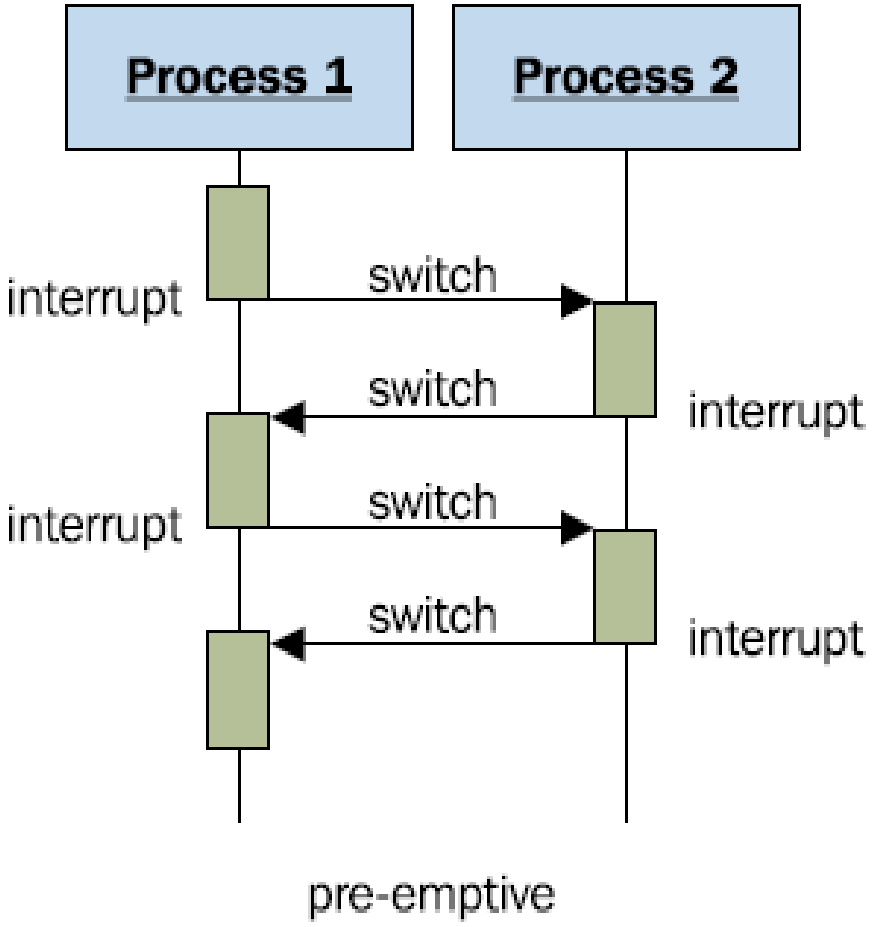
GPOS

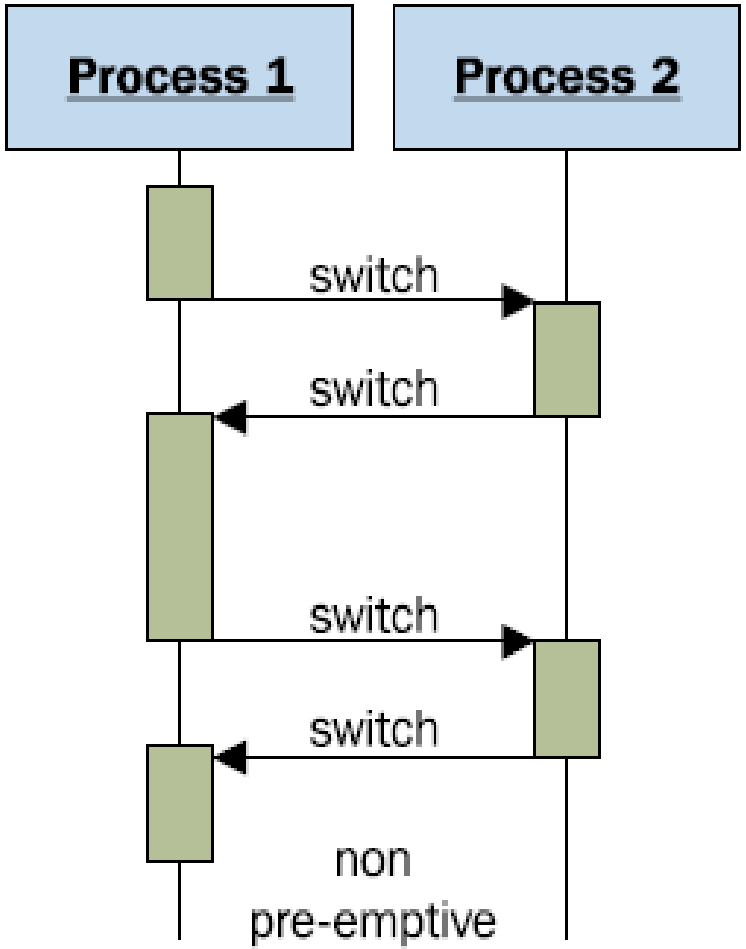
- General Purpose Operating System
- High throughput task scheduling
- Designed for high end, general purpose systems
- Not preemptible
- Linux & Windows distributions

RTOS

- Real time operating system
- Priority based task scheduling
- Designed for a low end, stand alone device
- Preemptible
- FreeRTOS, VxWorks, QNX Neutrino, Windows CE etc.

Preemptive vs non-preemptive





RTOS types

- Hard real-time:
  - deadline miss will result in a complete system failure
- Firm real-time:
  - deadline miss is acceptable
  - system can be degraded
  - result is not useful anymore
- Soft real-time
  - meeting the deadline is seem more like a goal
  - missing a deadline only degrades the usefulness of the received result and of the system

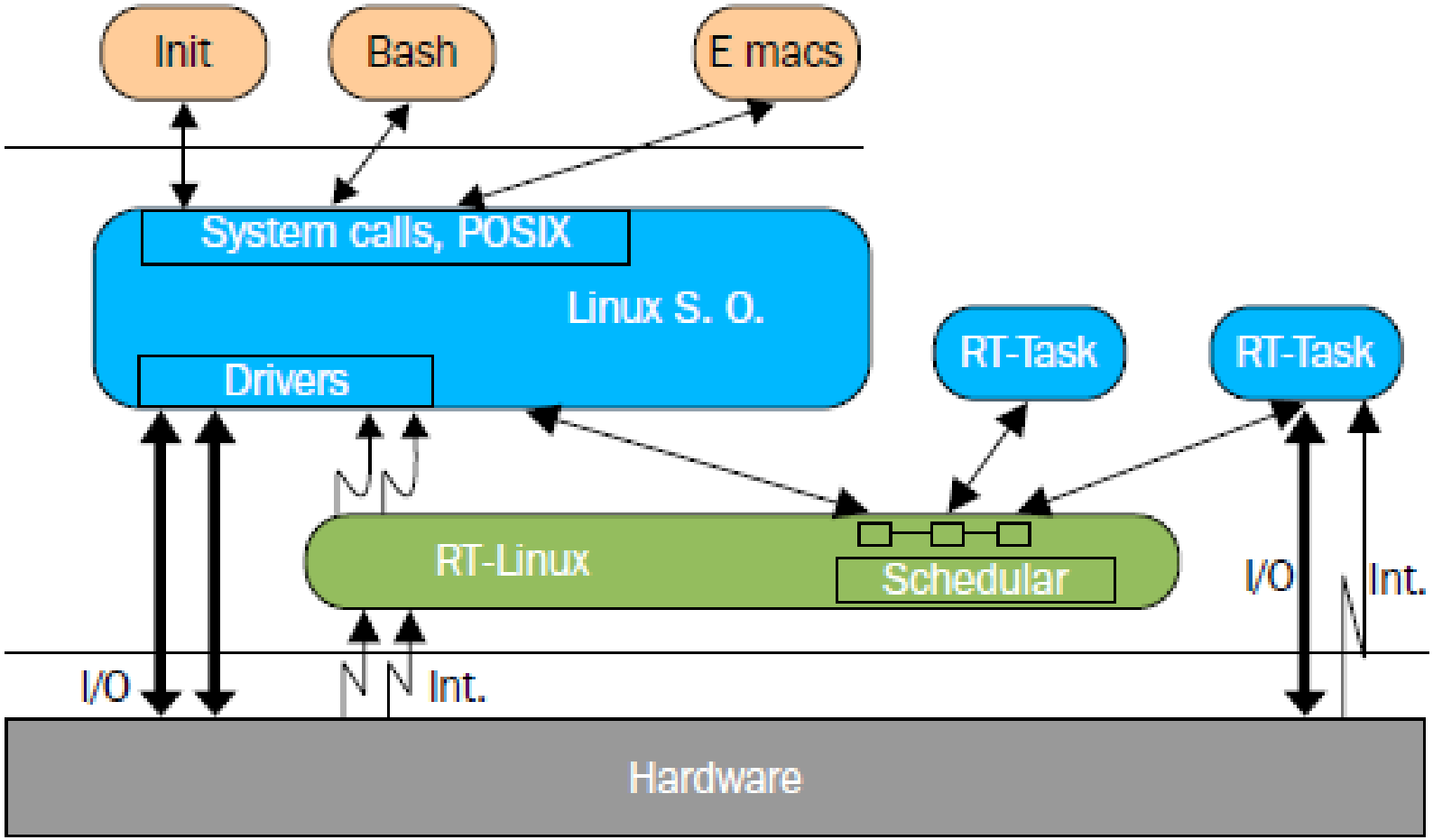
Linux not RTOS material

- Paging: inability to know when the translation between a virtual page and a page on the disk will happen.
- Coarsed-grained synchronization: once inside the kernel context a process cannot be preempted.
- Batching: operations can be batched for more effective use of resources.
- Request reordering: The I/O requests can be reordered for a more effective use of hardware.
- Fairness in scheduling: UNIX heritage, scheduler tries to be fair with all running processes.

RTOS option for Linux

- Improve the latency: SCHED_FIFO & SCHED_RR intended as real-time policies and for time critical applications.
- A more preemptive implementation: spinlock mechanism used for SMP, interrupt handling modifications, new scheduler and in general very serious kernel changes.
- Interrupt abstraction: run Linux with the priority of an idle task, fakes the disabling of an interrupt for the real-time kernel
  - RTLinux
  - RTAI
  - Xenomai

RTLinux architecture

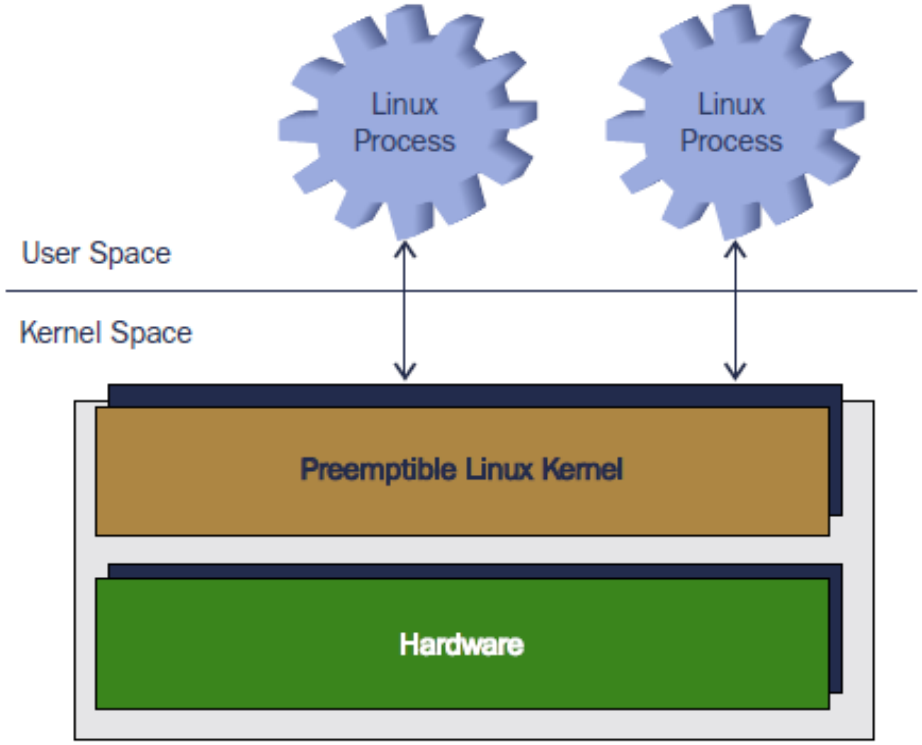


Outline

- GPOS vs RTOS
- PREEMPT_RT
- Yocto Project –rt kernel
- Linux real-time apps
- Benchmarking
- Meta-realtime

PREEMPT_RT

- Transforms Linux into a real-time solution
- Standard Linux only offers soft real-time
- Ingo Molnar`s PREEMPT_RT + Thomas Gheixner's high-res timer = hard real-time



PREEMPT_RT transformation process

- Protects critical sections with the preemptible **rwlock_t preemptible** and **spinlock_t**.
- The locking mechanisms is preempted using rtmutexes.
- A priority inversion and priority inheritance mechanism is implemented for mutexes, spinlocks and rw_semaphores.
- Use high resolution timer inside Linux timer API.
- Implement the usage of kernel threads for interrupt handlers.

Apply PREEMPT_RT patch

- Get the Linux kernel version: https://www.kernel.org/
  - wget https://www.kernel.org/pub/linux/kernel/v3.x/linux-3.12.38.tar.xz
- Get the corresponding rt patches for the kernel: https://www.kernel.org/pub/linux/kernel/projects/rt/
  - wget https://www.kernel.org/pub/linux/kernel/projects/rt/3.12/patch- 3.12.38-rt52.patch.gz
- Patch the source code with PREEMP_RT patches
  - tar xf linux-3.12.38.tar.xz
  - cd linux-3.12.38/
  - gzip -cd ../patch-3.12.38-rt52.patch.gz | patch -p1

Configure PREEMPT_RT

- Different from one architecture to another
- QEMUARM machine from poky.
  - CONFIG_GENERIC_LOCKBREAK=y
  - CONFIG_TREE_PREEMPT_RCU=y
  - CONFIG_PREEMPT_RCU=y
  - CONFIG_UNINLINE_SPIN_UNLOCK=y
  - CONFIG_PREEMPT=y
  - CONFIG_PREEMPT__LL=y
  - CONFIG_PREEMPT_COUNT=y
  - CONFIG_DEBUG_PREEMPT=y
  - CONFIG_RCU_CPU_STALL_VERBOSE=y
  - CONFIG_PREEMPT_RT_FULL=y
  - CONFIG_HZ_1000=y
  - CONFIG_HZ=1000

Outline

- GPOS vs RTOS
- PREEMPT_RT
- Yocto Project –rt kernel
- Linux real-time apps
- Benchmarking
- Meta-realtime

Yocto Project -rt kernel

- The PREEMPT_RT patched kernel recipe
- Three recipes available
  - linux-yocto-rt_4.1.bb
  - linux-yocto-rt_4.4.bb
  - linux-yocto-rt_4.8.bb
- Difference given by the LINUX_KERNEL_TYPE:
  - Standard: Includes the generic Linux kernel policy of the Yocto Project linux-yocto kernel recipes
  - Preempt-rt: Applies the PREEMPT_RT patches and the configuration options required to build a real-time Linux kernel
  - Tiny: Defines a bare minimum configuration meant to serve as a base for very small Linux kernels

PREEMPT_RT disadvantages

- Reduction of latency was done by forcing the kernel to preempt.
  - Adds a number of context switches in the process
  - Lowest priority tasks wait longer than they would in a normal Linux kernel
- The patch needs porting and adapting from one kernel to another.
  - In-house Linux kernel knowledge required

Outline

- GPOS vs RTOS
- PREEMPT_RT
- Yocto Project –rt kernel
- Linux real-time apps
- Benchmarking
- Meta-realtime

Real-time apps

- Real-time applications require a determinism operating system and hardware
  - Low-latency interrupt handling
  - The mechanism around ISR latencies should register values around tens of microseconds
- Required kernel configuration:
  - On-demand CPU scaling: helps with creation of long-latency events in low-power consumption mode
  - NOHZ: disables the timer interrupt

When writing applications

- Applications:
  - Disable the swap support to diminish latencies caused by page faults
  - The use of global variables or arrays should be kept to a minimum
  - Use priority inheritance futexes
  - Avoid input/output operations and data sharing
- Device drivers:
  - Interrupt handling done in a thread context
  - Hardware interrupt context can be used with the IRQF_NODELAY but avoid functions as wake_up(), up() or complete()

Outline

- GPOS vs RTOS
- PREEMPT_RT
- Yocto Project –rt kernel
- Linux real-time apps
- Benchmarking
- Meta-realtime

Technical terms

- Interrupt latency: the time that has elapsed since an interrupt was generated and until the execution has started.
- Scheduling latency: the time between the wake up signal of an event and a scheduler that has the opportunity to schedule a thread for it.
- Worst-case latency: the time that has passed since a demand was issued and until the response to that demand was received.
- Context-switch: the switching of the CPU from one process or thread to another

LPPTest

- Included in PREEMPT_RT patch
- Contains a Linux driver that changes a bit on a parallel port
- Identifies the response time
- Another driver responds to the bit change
- User application measures the results
- Two machines are required

RealFeel

- Test interrupt processing
- Uses /dev/rtc for periodic interrupts
- Measures the duration between interrupts and compares it with the expected value
- Prints the variation from the expected value
- Log file keeping for later processing

LRTB

- Linux Real-Time Benchmarking Framework
- Set of scripts and drivers
- Evaluate various performance counters
- Programs: hackbench, lmbench, dohell, cyclictest, hourglass, unixbench, cache-calibrator

Outline

- GPOS vs RTOS
- PREEMPT_RT
- Yocto Project –rt kernel
- Linux real-time apps
- Benchmarking
- Meta-realtime

Meta-realtime

- Initiative maintained by Bruce Ashfield from Windriver
- A place where real-time activities related to the Linux kernel or system development gets gathered.
- Placeholder for PREEMPT_RT, SCHED_DEADLINE, POSIX real-time and similar solutions
- Poor in content though but it does constitute a starting point for real-time related work

Schedtool-dl

- Scheduler testing tool for deadline scheduling
- Appeared from the need to change or make queries of the CPU-scheduling policies and even processes levels available under Linux
- Lock processes on various CPUs for SMP/NUMA

Rt-app

- Test application for real-time simulation load on a system
- Support for:
  - SCHED_FIFO
  - SCHED_OTHER
  - SCHED_RR
  - SCHED_DEADLINE
- Support for Adaptive Quality of Service Architecture (AQuoSA) framework

The End