D4 - Real-time Linux

Real-time Linux with RT-Preempt patch and Xenomai

Objectives

- Understand Real-Time programming
- Discover the various solutions under Linux
  - The Preempt_RT patch
  - Xenomai
  - Real-Time drivers and networking with Xenomai
  - Programming with Xenomai

Labs are conducted on the PC or on ARM-based target boards (Quad Cortex-A9 Sabrelite boards from NXP)
We use the latest available kernel supported by Xenomai

Course environment

- A PC for two trainees, with Linux and Xenomai on a target board
- Printed course material

Prerequisite

- Linux application programming skills (D0 - Linux user mode programmingcourse)
- Embedded Linux knowledge (D1 - Embedded Linux with Buildroot and Yoctocourse)
- For RTDM, Linux Driver Prgramming (D3 - Linux Driverscourse)
- Notions of real-time programming (RT1 - Real Time and Multi-Core programmingcourse)

Target Audience

- Any embedded systems engineer or technician with the above prerequisites.

Course Outline

First Day

Linux overview

- Linux
  - History
  - Version management
- The various licenses used by Linux (GPL, LGPL, etc)
- Linux distributions
- Linux architecture and modularity

Exercise: Boot Linux automatically starting a user application

The Linux Boot

- Linux kernel parameters
- The Linux startup sequence
- Various initialization systems (busybox init, system V init, systemd)
- Automatically starting an embedded system

Exercise: Boot Linux automatically starting a user application

The Linux kernel

- Downloading stable source code
  - Getting a tarball
  - Using GIT
- Configuring the kernel
- Compiling the kernel and its modules
  - The Linux build system
  - Modules delivered in-tree
  - Out-of-tree modules
- Installing the kernel and the modules
- The Linux Device Tree

Exercise: Configuring and compiling a target kernel for the target board

Second Day

Real-Time programming

- Scheduling
- Threads
  - Definition of a thread
  - POSIX threads
- Synchronization and communication primitives
  - Mutexes and Condition Variables
  - Barriers
  - Semaphores
  - Message queues
- Thread-specific Data

Exercise: Implement a multi-threaded server

- Classic real-time problems
  - Dead-Locks
  - Live-Locks
  - Priority Inversion

Exercise: Solve the Readers-Writer problem

Debug and Analysis Tools

- The Kernel tracing infrastructure
  - Tracepoints
  - The ftrace function tracer
  - Kprobes
  - Event tracers
- Performance monitoring in the Linux kernel
  - Perfcounters
  - Perf events
- Debugging the kernel using traces
- LTTng

Exercise: Trace context switches and measure latency times

Exercise: Use LTTng to trace multi-task context switches

Third Day

Real-Time Solutions for Linux

- The specificities of Real-Time
- Why Linux is not Real-Time

- Configuration Options in Vanilla Kernel
- The Preempt_RT patch
- The co-kernel approach

Exercise: Install Preempt_RT and check the effect on latencies

Xenomai

- Architecture
  - Adeos
  - Skins
  - Shadow Threads and Scheduling Domains
- Xenomai Schedulers
  - The Real-Time class schedulers
  - The Weak class schedulers
- Configuring Xenomai

Exercise: Install Xenomai
Exercise: Cross-compile an application for Xenomai

Fourth Day

Xenomai programming

- The Xenomai Skins
  - POSIX
  - RTDM
- Specificities of the POSIX skin
- Programming RTDM drivers
  - Creating a kernel module
  - Integration in the Linux Device Model
- Xenomai traces
- Porting to Xenomai

Exercise: Identify and Debug Spurious Relax problems
Exercise: Port an application on Xenomai and test real-time characteristics
Exercise: Write a simple RTDM driver

RTNet

- Overview of RTNet
  - Architecture
  - Non-determinism of Ethernet
  - Time Division Multiple Access
- Configuration
- Network Programming with RTNet

Exercise: Add RTNet support to the Xenomai kernel
Exercise: Test using udp client and server