

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 (cours [D0 - Programmation en mode utilisateur Linux](#))
- Embedded Linux knowledge (cours [D1 - Linux embarqué avec Buildroot et Yocto](#))
- For RTDM, Linux Driver Programming (cours [D3 - Drivers Linux](#))
- Notions of real-time programming (cours [RT1 - Programmation Temps-Réel et Multi-Cœurs](#))

Environnement du cours

- Cours théorique
 - Support de cours au format PDF (en anglais) et une version imprimée lors des sessions en présentiel
 - Cours dispensé via le système de visioconférence Teams (si à distance)
 - Le formateur répond aux questions des stagiaires en direct pendant la formation et fournit une assistance technique et pédagogique
- Activités pratiques
 - Les activités pratiques représentent de 40% à 50% de la durée du cours
 - Elles permettent de valider ou compléter les connaissances acquises pendant le cours théorique.
 - Exemples de code, exercices et solutions
 - Pour les formations à distance:
 - ▶ Un PC Linux en ligne par stagiaire pour les activités pratiques, avec tous les logiciels nécessaires préinstallés.
 - ▶ Le formateur a accès aux PC en ligne des stagiaires pour l'assistance technique et pédagogique
 - ▶ Certains travaux pratiques peuvent être réalisés entre les sessions et sont vérifiés par le formateur lors de la session suivante.
 - Pour les formations en présentiel:
 - ▶ Un PC (Linux ou Windows) pour les activités pratiques avec, si approprié, une carte cible embarquée.
 - ▶ Un PC par binôme de stagiaires s'il y a plus de 6 stagiaires.
 - Pour les formations sur site:
 - ▶ Un manuel d'installation est fourni pour permettre de préinstaller les logiciels nécessaires.
 - ▶ Le formateur vient avec les cartes cible nécessaires (et les ramène à la fin de la formation).
- Une machine virtuelle préconfigurée téléchargeable pour refaire les activités pratiques après le cours
- Au début de chaque session (demi-journée en présentiel) une période est réservée à une interaction avec les stagiaires pour s'assurer que le cours répond à leurs attentes et l'adapter si nécessaire

Audience visée

- Tout ingénieur ou technicien en systèmes embarqués possédant les prérequis ci-dessus.

Plan du cours

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