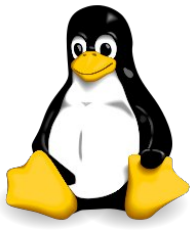
A.L.S.E.
http://www.alse-fr.com

Formation Pratique

Développer avec Linux Embarqué



Présentation

Pour de nombreuses raisons différentes, de plus en plus de systèmes embarqués adoptent **Linux Embarqué** comme *Operating System*, avec toute sa richesse et les bénéfices qui vont avec.

Cette **formation de quatre jours** permet de comprendre de quoi est constitué et comment fonctionne un Système Linux Embarqué. L'accent est mis sur chacun des composants ainsi que sur les nombreux outils disponibles pour le développement et la mise au point.

La formation commence à la configuration initiale pour aller jusqu'au produit fini, en mettant l'accent sur les outils libres, et présente également certaines solutions commerciales.

Ce cours s'inscrit dans le cadre d'une action d'acquisition ou de perfectionnement des connaissances.

Les **Exercices Pratiques**, qui occupent environ **50 % du temps de la formation** sont des éléments clés du succès de ces formations, et sont essentiel pour l'apprentissage.
Ils **assurent** également un **contrôle continu de l'acquisition des compétences.**
Ces exercices, très indépendants de la plateforme, seront menés sur un **kit Xilinx Zynq** (ARM CortexA9).

Les **Instructeurs ALSE** sont aussi et surtout des **Experts** qui utilisent à journée entière les Langages et les Méthodes de Conception qu'ils enseignent, pour concevoir et vérifier des systèmes complexes et performants. Ils savent partager leur savoir-faire avec passion et sont particulièrement appréciés des participants.

Pré-requis

La participation à cette formation demande une **connaissance préalable de l'environnement de travail Linux** (desktop) ainsi qu'une certaine culture des systèmes embarqués (RTOS ou *bare metal*).

Durée

Quatre (4) jours, soit **28 heures effectives** de cours ;
Typiquement 9h30 → 18H, le premier jour et 9H → 17h30 les jours suivants.

Lieu

Dans les locaux d'ALSE, **Paris XIII^{ème}**

Matériel

Pour les cours publics, le matériel (station de travail et plateforme embarquée) est fourni pour la durée du training.

Objectifs pédagogiques

- Comprendre et maîtriser la structure d'un système Linux Embarqué et ses différentes composantes.
- Construire et configurer un système Linux sur une cible donnée, incluant les systèmes de fichiers et le processus de démarrage (boot).
- Savoir comment développer, analyser, optimiser et mettre au point des applications sous cet environnement avec différents outils Linux (disponibles gratuitement).

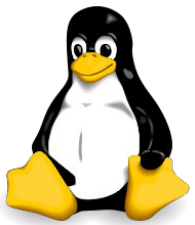
NB : à partir de quatre ou cinq stagiaires, une formation sur site est envisageable.

ALSE est un Organisme de Formation Professionnelle Continue déclaré auprès de la DIRECCTE depuis 1996 sous le numéro 26.21.01281.21 et enregistrée dans le Datadock. Cette formation peut donc être prise en charge dans ce cadre.

Programme de la Formation

Développer avec Linux Embarqué

(4 jours)



Introduction

- Course Objectives • Linux Background
- Open-Source Software • Licensing
- Linux Distributions
- Development platforms • Version control

Anatomy of an Embedded Linux System

- Toolchain
- Bootloader
- Linux kernel
- Filesystem
- Host to target communications

Practical Exercises :

* Booting a standard image
* Establishing target communications
* Exploring the target filesystem

Working with the Linux Kernel

- Scheduling, real-time and memory
- Kernel configuration • Building and booting the kernel
- Board support
- Kernel modules

Practical Exercises :

Configuring and building the kernel
Creating a kernel module

Debugging the Linux Kernel

- Kernel logs
- JTAG
- KGDB/KDB

Practical Exercises :

Using kernel logs
Using KGDB to debug a kernel module

Building Applications

- Compiling on the target • Cross-compiling
- Static and shared libraries
- Using portable build systems (e.g. Autotools)

Practical Exercises :

Cross-compiling
Creating static and shared libraries
Configuring and compiling an Autotools application

.../...

Debugging Applications

- Compiling for debugging
- Target debugging • Cross debugging
- Other user space tools
- Debugging user space segmentation faults

Practical Exercises :

Target debugging
Cross-debugging a segmentattion fault
Finding memory leaks with Valgrind
Generating and using core-dumps

Linux Applications:

- How a Linux application can be structured
- Managing processes and threads
- Inter-process communications
- Interfacing with the kernel

Practical Exercises :

Simple IPC
Threads and processes

Configuring Filesystems

- Filesystem contents
- BusyBox
- Filesystem types and locations
- NFS • RAM filesystems • Block-based filesystems
- Flash filesystems (JFFS2, UBIFS)

Practical Exercises :

Configuring and compiling BusyBox
Creating an initramfs
Writing a UBIFS into flash

Configuring the Bootloader

- The bootloader
- Bootloader choices
- Working with U-Boot
- Porting U-Boot to a new board

Practical Exercises :

Configuring the target root filesystem location
Flashing a kernel image in U-Boot

Trace & Profile

- Linux trace technology
- SysProf • OProfile
- FTrace • Perf
- Other Tools
- Use case examples

Practical Exercises :

Investigating scp behaviour with trace & profile tools

-=oOo=-