



A1 - Linux administration

Administration de base du système Linux

Objectifss

- Apprendre à installer et administrer une machine Linux.
 - Installer Linux sur un poste de travail
 - Comprendre l'architecture de Linux
 - Gérer les paquetages logiciels
- Mettre en oeuvre le réseau
- Paramétrer l'interface graphique
- Sécuriser un système Linux

Matériel

- Un PC Linux par stagiaire
- Un support de cours par stagiaire

Pré-requis

- Connaissances générales en système d'exploitation.

Introduction to MCUXpresso SDK

- SDK structure and components
- Toolchains, CMake and Ninja integration
- Application structure and examples

MCUXpresso Config Tools

- Overview of the configuration tool suite (Pins, Clocks, Peripherals, Device settings)
- How Config Tools integrate with MCUXpresso SDK and West builds
- Generating initialization code (pin_mux.c/h, clock_config.c/h, peripheral setup)
- Using the graphical interface to configure GPIO, UART, and system clocks
- Exporting configuration files and re-integrating them into applications
- Limitations and best practices when combining with Kconfig/prj.conf

Course Environment

- Theoretical course
 - PDF material in English (printed for face-to-face); online over Teams.
 - Trainer assistance throughout.
- Practical activities (40-50% of duration)
 - Code examples, exercises and solutions.
 - Remote: one online Linux PC per trainee, with emulated or physical board as needed.
 - Face-to-face / on-site: one PC (one per two beyond six trainees), target board and install manual as needed.
- Downloadable preconfigured VM to redo the labs afterwards.
- Each session starts with a trainee check-in.

Target Audience

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

Course Outline

First Day

First day

First Day

Introduction

- Présentation de Linux
- La documentation

First day

Installation de Linux

- Les pré-requis
- Le formatage
- Le choix des paquets

First day

Les commandes d'administration

- L arborescence des fichiers
- Attributs des fichiers
- Commandes de gestion de fichiers et de répertoires
- Le shell
- L éditeur vi

Les scripts shell

- L exécution des scripts
- Les variables
- Les instructions de contrôle
- Les fonctions

La gestion des utilisateurs et des droits

- Les commandes de gestion des utilisateurs
- Les commandes de gestion des groupes
- Les droits et leur gestion

Second Day

Les disques et les systèmes de fichiers

- Les partitions sur les disques
- Les systèmes de fichiers

La sauvegarde

- Les différents outils de sauvegarde
- La commande tar
- La commande cpio

Les applications

- La gestion des processus
- L'automatisation des tâches
- syslog : la gestion des messages systèmes

Installation des applications

- Installation à partir des sources
- Les paquets RedHat
- Les paquets Debian

Second day

L'arrêt et le démarrage de Linux

- La séquence de démarrage
- Le démarrage automatique des services
- L'arrêt de la machine Linux

Le réseau

- Les commandes pour configurer TCP/IP
- Démarrage de services par inetd/xinetd
- services réseaux de base : telnet, ssh, ftp, wget

L'impression

- Le système CUPS
- Administration des impressions

L'environnement graphique

- X window
- KDE
- GNOME

La sécurité

- Les règles de base pour sécuriser une machine Linux
- Sécurisation des connexions

Open Portable Trusted Execution Environment (OP-TEE)

- Introduction to OP-TEE
- Key Features
- Hardware, software, and firmware requirements
- Architecture of OP-TEE
 - Components, modules, and communication channels
- Use Cases
 - Secure storage

- Secure communication
- Secure execution of applications
- OP-TEE build and deployment
 - Setting up the environment
 - Configuration of OP-TEE
 - Compilation of OP-TEE
- Comparison to other TEE solutions
- Trusted Applications (TA) on OP-TEE
 - The role of a TA in a secure system
 - Writing a Trusted Application
 - Loading and executing a Trusted Application within the OP-TEE runtime
 - Debugging and testing Trusted Applications
 - Communication between Trusted Applications and normal world applications
 - Best practices for creating secure Trusted Applications

Exercise: Build and install OP-TEE

Exercise: Write a TA application that communicates with a normal world application

Parallelism Problems Solutions

- Parallel programming problems
 - Uncontrolled parallel access
 - Deadlocks
 - Livelocks
 - Starvation

Exercise: The producer-consumer problem, illustrating (and avoiding) concurrent access problems

Exercise: The philosophers' dinner problem, illustrating (and avoiding) deadlock, livelock and starvation

Exercise: The readers-writer problem, illustrating complex concurrent access solving