



Installation, programmation, adaptation et écriture de drivers

Les applications industrielles sont de plus en plus souvent réalisées sous une version embarquée de Linux. De plus, l'environnement souvent très spécifique dans lesquelles s'exécutent ces systèmes font qu'il est parfois nécessaire d'adapter l'installation de Linux à l'environnement matériel.

ac6-formation vous propose des formations vous permettant de réaliser des applications sur Linux embarqué, mais aussi d'adapter ce système d'exploitation si le besoin s'en fait sentir.

Nous vous proposons également ([voir l'onglet "SysAdmin"](#)) toute une gamme de formations pour vous aider à mettre en place et administrer une infrastructure Linux, sur stations de travail et serveurs.

You can see detailed course descriptions of the various trainings by using the above navigation bar. You can also click on course identifiers in the following course briefs hereafter.

Cours principaux

[D0 - Programmation Linux et uClinux](#) La programmation utilisateur sur Linux et μ Clinux

[D1 - Embedded Linux](#) Linux pour les systèmes embarqués

[D1 - Embedded Linux](#) Implementing Linux on embedded systems

[D1L - Linux embarqué Freescale avec LTIB](#) Linux pour les systèmes embarqués Freescale à base ARM et PowerPC

[D3 - Linux drivers](#) Writing Linux drivers

Ce cours couvre les différentes techniques nécessaires à l'écriture de drivers Linux (2.4 et 2.6), la gestion de bus (PCI...), le hot-plug et l'auto-configuration des périphériques et la gestion multi-coeurs dans les drivers.

[D4 - Linux temps-réel Xenomai](#) Le temps-réel Linux avec RTAI et Xenomai

Ce cours présente les différents choix possibles pour réaliser un système temps-réel embarqué avec Linux et les critères de choix en fonction de la démarche (migration depuis un RTOS traditionnel, réécriture directe en environnement Linux, migration puis réécriture) et des contraintes temps-réel.

[D5 - Embedded GUI](#) Graphical User Interfaces for Embedded Linux

[D7 - Linux drivers hotplug and power management](#) Writing drivers with hot-plug and power management support

This course delves into the concepts of generic devices/drivers for Linux, the management of hotplug, as well as writing drivers for USB host and gadget. It also covers the whole issue of power management in recent versions of the 2.6 Linux kernel.

[D8 - USB Linux Drivers](#) Writing USB-2.0 and USB-3.0 host and gadget drivers on Linux

This course details the Linux driver model, the USB hotplug and power management architecture to write USB host (client) drivers as well as gadget drivers.

Autres cours

A1 - Linux administration Administration de base du système Linux
Ce cours est destiné à toute personne ayant à installer, configurer et administrer une machine Linux.

A2 - Administration Linux avancée Ce cours est destiné à toute personne ayant à installer, configurer et administrer un serveur Linux.

A3 - Samba Intégration de stations Windows avec un serveur Linux

A5 - Network security Sécurisation d'un serveur Linux, Internet ou Intranet

E1 - Eclipse Utilisation de l'environnement de développement Eclipse pour C, C++ et Java(TM)

G1 - Android Installation Android installation on a hardware platform

G2 - Android Programming Programming applications for the Android platform

G3 - Android Internals Porting the Android System and Framework Architecture

Installing Android on a new platform is a complex process requiring a deep understanding of the internals of the Android frameworks and the Hardware Abstraction Layer. This course explains how the frameworks are structured and can be adapted to a platform on which a basic Android port already exist.

G4 - Android Porting and Internals Porting the Android System and Framework

Installing Android on a new platform is a complex process; you need to port first the Linux kernel then install the Android platform. This second step requires a deep understanding of the internals of the Android frameworks and the Hardware Abstraction Layer. This course cover both phases, with a special emphasis on the multimedia frameworks.

N3I - IPV6 Architecture et programmation d'un réseau IPV6

Maîtriser la mise en place du nouveau protocole réseau IPV6 et savoir programmer une application utilisant IPV6

RT1 - Real Time and Multi-Core programming Programming Linux real-time and multi-core systems, avoiding common pitfalls

Real-time and embedded code, especially targetting multicore processors, cannot be effectively tested; it must be validated before coding. This training help you master multitask and real-time programming of multi-core processors, understanding how to effectively solve problems using the primitives provided by the underlying Operating System.

