



## **D1 - Embedded Linux**

### **Linux pour les systèmes embarqués**

#### **Objectifs**

- Utiliser les outils de développement natifs ou croisés
- Personnaliser, réduire et configurer le noyau Linux
- Embarquer Linux sur Flash EEPROM
- Détailler la procédure de boot Linux
- Monter un système de fichier distant
- Booter un noyau Linux distant
- Etudier les spécificités des noyaux 2.4 et 2.6

*Les exercices se font sur des cartes cibles :*

*Carte "SnowBall" de ST-Ericsson, basée sur un ARM Cortex/A9 double cœur, avec sonde JTAG Lauterbach.*

*Carte à base de processeur ARM9 d'Atmel, avec sonde JTAG Lauterbach.*

*Nous utilisons le dernier noyau disponible sur [www.kernel.org](http://www.kernel.org)*

#### **Matériel**

- Un PC Linux par binôme
- Une carte embarquée par binôme, au choix:
  - 1 carte à base de PowerPC 405 GPr (KSBC 0405 de chez Ecrin)
  - 1 carte à base d'ARM 926 (AT91SAM9261-EK de chez Atmel)
  - 1 carte simulée (Qemu) à base d'ARM 926 (carte de référence "Versatile" de chez ARM)
- Support de cours
- CDROM avec documentation et outils utilisés

#### **Pré-requis**

- Connaissance de Linux utilisateur

#### **Plan**

##### 1er jour

##### **Architecture de Linux**

- Présentation de Linux
- Licences GPL et open source
- Distributions Linux
- Architecture et modularité de Linux

### Les outils Linux pour l'embarqué

- Les firmware/bootloader (Uboot, Redboot,...)
- Les bibliothèques adaptées à l'embarqué (uClibc, dietlibc,...)
- Les IHM adaptées à l'embarqué (miniGUI, Qt/embedded, Nano-X,...)
- Busybox, le "couteau suisse" de Linux embarqué
- Les distributions spécialisées
  - Commerciales (MontaVista, Sysgo, Timesys, Windriver,...)
  - Open Source (ELDK, Koan, RTAI,...)

### Les chaînes de compilation croisée

- Chaînes complètes (ELDK, CodeSourcery, ...)
- Outils de compilation de chaînes (Crosstool-ng, Buildroot, ...)
- Compilation manuelle

*Exercice : Construction d'une chaîne avec Crosstool-ng et avec Buildroot. Etude des différences.*

## 2ème jour

### Création du noyau Linux pour l'embarqué

- Téléchargement des sources
- Etude du Makefile du noyau
- Configuration du noyau
- Compilation du noyau et des modules
- Installation des modules et du noyau

### Etude du support des MTD dans Linux (Memory Technology Devices)

- Les mémoires de type NOR
- Les mémoires de type NAND

*Exercice : Configuration et compilation d'un noyau Linux adapté à la cible*

### Création/modification du Board Support Package (BSP)

- Etude des BSP ARM
  - Les numéros de machine ARM
  - Organisation des fichiers sources
  - Les Makefiles
  - Les fichiers de configuration
  - La struct machine\_desc

*Exercice : Modification du BSP pour le support et la partitionnement d'une mémoire NAND*

### Création du système de fichier racine

- Périphériques, programmes, bibliothèques, ...
- Installation des modules
- Recherche et installation des bibliothèques dynamiques nécessaires

- Vérification de la cohérence du système de fichier
- Utilisation d'outils de génération de systèmes de fichiers racine embarqués (BuildRoot, LTIB, ...)

*Exercice : Création d'un système de fichier racine avec Buildroot.*

### 3ème jour

#### **Choix du type de système de fichier**

- Ramdisk/initrd
- Systèmes de fichier adapté aux flashes (JFFS2, UBIFS, ...)
- Systèmes de fichier adaptés à l'embarqué (ROMFS, CRAMFS, ...)
- Systèmes de fichiers compatibles "desktop" (EXT2, FAT)

#### **Boot de Linux**

- Partitionnement d'un disque Linux
- Bootloaders (lilo, grub, U-boot...)
- Paramètres du noyau au boot
- Etapes de chargement et de démarrage du noyau
- Choix des programmes lancés au démarrage (runlevels, services, suppression de l'IHM...)

*Exercice : Partitionnement dynamique de la NAND*

*Exercice : Flashage du noyau et du système de fichier*

*Exercice : Boot depuis la mémoire flash*

### 4ème jour

#### **Développement croisé**

- Préparation de la machine hôte
  - Services réseau (DHCP, TFTP)
  - Partage du système de fichier (NFS)
  - Etude des outils de développement (gcc, gdb, les Makefiles)
  - Utilisation de l'IDE Eclipse pour la programmation et le debug croisé
- Préparation de la cible
  - Téléchargement d'un noyau par TFTP
  - Démarrage sur un système de fichier distant (diskless)
  - Utilisation de gdbserver pour le debug distant
- Utilisation de Qemu pour simuler une carte complète

*Exercice : Installation d'une distribution complète sur carte PowerPC, ARM ou sur simulateur d'ARM (Qemu)*

*Exercice : Compilation, test et debug distant d'une application embarquée*

*Exercice : Compilation et installation de modules noyau indépendants*

*Exercice : Compilation croisé de paquetage*

## Renseignements pratiques

**Duration :** 4 days

**Cost :** 1950 € HT

**Prochaines sessions :** du 28 February au 2 March 2012  
du 17 au 20 April 2012



SARL au capital de 15400€ - SIRET 449 597 103 00026 - RCS Nanterre - NAF 722C - Centre de Formation : 19, rue Pierre Curie - 92400 Courbevoie  
Siège social et administration : 21, rue Pierre Curie - 92400 Courbevoie - Tél. 01 41 16 80 10 - Fax. 01 41 16 07 78

Last site update: Mon 13 Feb 2012 02:28:18 PM CET

<http://www.ac6-formation.com/>