



## D0 - Programmation Linux et uClinux

### La programmation utilisateur sur Linux et $\mu$ Clinux

#### Objectifs

- Découvrir Linux et ses outils de développement et de mise au point
- Programmer et déboguer des applications Linux
  - Programmation multi-tâches et temps réel
  - Communications inter-processus
  - Entrées-sorties
- Linux et  $\mu$ Clinux

*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.*

*Pour uClinux, carte à base de Cortex/M3 (STM32) de STMicroelectronics, avec sonde JTAG Lauterbach.*

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

#### Matériel

- Un PC Linux par binôme
- Support de cours
- CDROM avec documentation et exercices corrigés

#### Pré-requis

- Connaissance de Linux utilisateur.
- Connaissances générales en programmation C.

#### Plan

##### 1er jour

#### Présentation & Architecture

- Historique
- Licences GPL et open source
- Distributions et versions de Linux et uClinux

- Architecture du kernel Linux et uClinux

## Utilisation de Linux

- Le système de fichier
- Utilisation du shell et des scripts
- L'éditeur vi
- Administration de base

## Le développement d'applications

- Développement d'applications Linux
  - différence entre mode utilisateur et mode noyau
  - structure d'une application Linux
- Développement d'applications uClinux
  - les formats d'exécutables uClinux
  - les options de compilation
- Développement de bibliothèques Linux
  - statiques (.lib)
  - dynamiques (.so)
- Développement de bibliothèques uClinux
  - statiques (.lib)
  - dynamiques (.so)
- Outils de développement
  - les outils de compilation
  - la documentation (pages "man", info)
  - les makefile
  - environnements intégrés (Eclipse, Kdevelop,...)
- Outils de debug et de mise au point
  - gdb
  - debug mémoire avec dmalloc et electric fence
  - analyse runtime complète avec Valgrind

*Exercice : Ecriture d'une bibliothèque mathématique simple compilée statiquement puis dynamiquement pour Linux et uClinux*

## 2ème jour

### Entrées/Sorties et Signaux

- Interface standard des entrées/sorties
  - API
  - inodes et données
  - verrous
- Programmation réseau
  - sockets
  - protocoles UDP et TCP

*Exercice : Programmation d'une application client/serveur. Comparaison entre TCP/IP et UDP/IP.*

- Entrées-sorties asynchrones
  - mode non bloquant
  - E/S multiplexées
  - E/S notifiées
  - Asynchrone enchainé (API aio POSIX)

*Exercice : Gestion de plusieurs connexions concurrentes par multiplexage d'I/O*

- Signaux POSIX et timers

- types de signaux
- réception d'un signal
- fonctions utilisables dans un gestionnaire de signal
- timers en mode utilisateur

### 3ème jour

#### **Programmation Multitâches**

- Les processus
  - définition
  - structures du noyau
  - sécurité
  - cycle de vie (fork, exec,...)
  - Différence entre Linux et uClinux

*Exercice : Gestion de plusieurs connexions concurrentes en multi processus (fork)*

- Les threads
  - définition (threads utilisateur et noyau)
  - threads POSIX (mutex, variables conditionnelles, Thread Specific Data,...)

*Exercice : Gestion de plusieurs connexions concurrentes en multitâches (pthreads)*

- Gestion mémoire
  - gestion de la mémoire par la MMU (adresses virtuelles/physiques)
  - pagination et protection
  - swap
  - allocation mémoire
  - caches
  - différences entre Linux et uClinux
- Ordonnancement
  - la commutation de contexte
  - politique d'ordonnancement "normale" (niveaux nice)
  - politique d'ordonnancement temps réel (FIFO ou Round Robin)
  - principe et performances de l'algorithme
  - ordonnancement SMP (Symmetrical Multi Processor)

### 4ème jour

#### **Communication entre tâches (IPC POSIX)**

- Présentation de la norme POSIX
- Communication entre tâches
  - Mapping de fichier en mémoire
  - Mémoire partagée
  - Files de messages
  - Pipes
- Synchronisation des tâches
  - sémaphores
  - mutex
  - signaux
- Les IPCs System V sont décrites en annexe

*Exercice : Mise en place d'un watchdog sur timeout pour fermer les connexions inactives sur un serveur*

## Renseignements pratiques

**Duration :** 4 days

**Cost :** 1950 € HT

**Prochaines sessions :** du 10 au 13 July 2012

du 11 au 14 September 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: Tue 22 May 2012 10:50:29 AM CEST

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