

FORMATIONS LINUX

MNIS – Tour de l’Horloge - 4, place louis Armand – 75012 Paris

TEL : 0950 070814

ARM

ASSEMBLEUR ET ARCHITECTURE ARM

Durée : 5 jours

Prix : 1740€

GROUPE DE FORMATIONS

La formation fait partie du groupe de formation « Développement »

Développement

ARM	Assembleur et architecture ARM	5	
LGC	Langage C	5	DEBUTANT
LIS	Développement d’applications sous Linux	5	
PTR	Programmation temps réel	4	
SHL	Ecrire des scripts Shell	4	DEBUTANT
KSH	Ecrire des scripts Korn Shell	4	
LDI	Noyau Linux et développement de drivers	5	AVANCÉ
DRN	Développement de drivers réseau sous Linux	5	AVANCÉ
TCP	Développement TCP-IP sous UNIX	5	AVANCÉ

QUEL OBJECTIF

Apprendre l’assembleur ARM, programmer les coprocesseurs des SoC "System on Chip" ARM9 et Cortex-A et Cortex-M, mettre en œuvre un moniteur pour Trustzone.

POUR QUI

Cette formation s’adresse aux développeurs de firmware ou de logiciels bas niveau pour architecture ARM ainsi qu’aux architectes désirant connaître les « System on Chip » ARM

PRE-REQUIS

Connaissances de base de la programmation et des micro-processeurs. La connaissance du système Linux facilitera la mise en œuvre des travaux pratiques.

POUR QUOI

Vous travaillez ou allez travailler sur une architecture ARM-v7 et vous voulez approfondir vos connaissances.

DEROULE DE LA FORMATION

INTRODUCTION A ARM

Présentation de l'architecture ARM : de v5 à v7.

Le processeur ARM, fonctionnement, les registres.

Les instructions ARM. L'architecture load/store.

Architecture ARMv5 : TCM, Jazelle, domaines, MMU.

Architecture ARMv7 : caches, mode monitor, trustzone.

Travaux pratiques

Prise en main de l'environnement de développement. Compilateur, éditeur de liens et simulateur.

Ecriture d'un programme ARM.

PRESENTATION DE L'ASSEMBLEUR ARM

Les opérations de base de l'assembleur ARM.

L'exécution conditionnelle et les branches.

Les opérations arithmétiques et logiques.

Accès à la mémoire, la gestion des piles.

Les opérations atomiques.

Le mode THUMB

Travaux pratiques

Ecriture de différents petits programmes en assembleur ARM.

INTERFACE C ET ASSEMBLEUR ARM

Utiliser l'assembleur ARM pour optimiser les applications.

Intégration de routines assembleur dans un programme C.

Bibliothèques de fonctions assembleur.

Travaux pratiques

Ecriture d'une bibliothèque de synchronisation de threads en assembleur.

GESTION DES EXCEPTIONS ET DES PERIPHERIQUES

La gestion des exceptions et des interruptions par le processeur ARM.

Le vecteur d'interruption.

Les différentes façons de gérer une interruption.

Présentation de quelques périphériques standard, PIC, timer, UART.

Travaux pratiques

Mise en oeuvre d'une interruption timer sur une carte versatile.

PROGRAMMATION SYSTEME ARM9

La gestion du processeur par un OS.

Modes de fonctionnement. Passage en mode superviseur.

Gestion du CPSR et des coprocesseurs.

Le coprocesseur système CP15.

La gestion de la MMU, des domaines et des caches.

Les barrières, la Tiny Coupled Memory, le Fast Context Switch.

Travaux pratiques

Passage en mode superviseur et retour en mode utilisateur. Construction d'un mini-OS, activation de la protection mémoire par domaines.

PROGRAMMATION SYSTEME CORTEX-A

Le Cortex A pour sécuriser les applications sensibles.

Mise en oeuvre de la protection par trustzone.

Mise en oeuvre d'un moniteur sécurisé.

Gestion des coprocesseurs, de la MMU et des caches L1 et L2.

FCSE et ASID.

Travaux pratiques

Passage en mode moniteur et gestion d'un mini OS. Activation de la protection mémoire par trustzone.

PROGRAMMATION SYSTEME CORTEX-M

Le Cortex-M pour développer rapidement une application embarquée.

Le boot du Cortex-M

Spécificité pour la gestion des exceptions sur Cortex-M

Gestion des coprocesseurs, de la MPU et des caches L1 et L2.

Travaux pratiques

Ecriture d'une mini-application sur le Cortex-M.