Développement de pilotes dans le noyau Linux

Réf NoyauLinux	**4 jours**

Objectifs de la formation :

Être capable de
- Compiler un noyau Linux ;
- Créer des modules ;
- Écrire des pilotes de périphériques : caractère, bloc, réseau sur des bus pci et usb,

Pré requis :	**Méthode et moyens** :
➢ Programmation en langage C ➢ Bases sur l'architecture d'un environnement Linux : noyau, module, descripteur de périphérique,	➢ 1 poste de travail par personne ➢ Groupe de 4 personnes maximum ➢ De nombreux exercices pratiques ➢ Méthode pédagogique active

Programme :

1) Programmation Linux en mode noyau

Cycles de développement du noyau, versions et diffusion des sources.
Mode superviseur et utilisateur, fonctionnement des appels-système.
Principes des modules et organisation des sources du noyau.
Outils de développement (Gcc, Kconfig et Makefile), et de débogage (Kgdb, Ltt).
Principe de compilation du noyau et des modules.
Dépendances et contrôle de versions des modules.
Exportation de symboles.

2) Éléments essentiels de programmation noyau

Périphériques : bloc, caractère, réseau et méthodes de développement.
Contextes de fonctionnement du noyau : appel-système, interruption et threads.
Synchronisation des appels-système par sémaphores et mutex.
Protection des variables globales par spinlocks.
Éléments temporels : ticks et jiffies, mesures horaires, attentes actives et sommeils.
Programmation d'actions différées et timers.
Représentation des processus et threads, tâche « current ».
Accès aux informations concernant le processus en cours d'exécution.
Préemptibilité du noyau 2.6.
Mémoires virtuelle et physique, espace d'adressage utilisateur et noyau.
Communication avec les processus par le système /proc.

3) Écriture de pilotes, périphérique en mode caractère

Écriture de pilotes de périphériques : fichiers spéciaux (/dev), numéros majeur et mineur
Fonctions essentielles de gestion de périphérique.
Fonctions avancées : paramétrage du périphérique (Ioctl), attentes d'événements, waitqueue.
Accès au matériel : ports d'entrées-sorties, mapping de ports ou de plages de mémoire physique, allocation de mémoire.
Les interruptions : activation/désactivations, handler, partages et imbrication Bottom half : Tasklet et workqueue, softirq.

4) Périphérique en mode bloc et système de fichiers

Principe des périphériques en mode bloc.
Enregistrement du driver.
Callback de lecture et écriture.
Support du formatage et opérations avancées
Ordonnanceur des entrées-sorties par bloc du noyau 2.6, choix d'une stratégie.
Conception/Implémentation/Enregistrement des systèmes de fichiers

Dominique COLOMBANI, pré Chabert, 38420 REVEL - SIRET 522 263 094 00016
Tél : 06 76 41 04 77 - Fax : 09 72 12 89 19 - Web http://www.dominique-colombani.fr
Email : pro@dominique-colombani.fr

Développement de pilotes dans le noyau Linux

5) Éléments avancés de programmation noyau

Allocations de mémoire : par zones physiques, virtuelles ou par pages.
Modèle des drivers du noyau 2.6 : Kset et Kobject.
Gestion des événements Hotplug.
Accès au bus PCI : adressage, drivers, auto-détection et interruptions.
Transferts DMA : organisation des adresses, allocation des buffers et transfert de données.

6) Interfaces et protocoles réseau

Gestion des interfaces réseau sous Linux.
Enregistrement d'une interface net_device.
Callback d'émission et réception.
Implémentation et options de la pile IPv4 sous Linux.
Transfert des paquets sk_buff.
Résolution d'adresse et détail du routage.

7) Drivers pour périphériques USB

Principe des périphériques USB.
Sous Système USB-core.
Exploration de sysfs
Accès aux « endpoints » de dialogue.
Types de périphériques.
Construction d'un URB.
Dialogue en modes « Contrôle » et « Interruption ».
Inscription d'un gestionnaire de communication en modes « Bulk » et « Isochronus ».

Dominique COLOMBANI, pré Chabert, 38420 REVEL - SIRET 522 263 094 00016
Tél : 06 76 41 04 77 - Fax : 09 72 12 89 19 - Web http://www.dominique-colombani.fr
Email : pro@dominique-colombani.fr