Formation développement de drivers Linux et noyau

Des drivers Linux prêts pour un produit et pour la revue upstream, avec les frameworks du noyau utilisés comme leurs mainteneurs l'entendent.

5 jours	Intermédiaire à avancé	Sur site ou à distance	Français ou anglais
Durée	Niveau	Format	Langue (supports en anglais)

Présentation

Cette formation apprend à écrire des drivers Linux prêts pour la production et pour une soumission upstream. Elle part des sources du noyau et des modules, passe par le modèle de périphériques, la gestion mémoire, les interruptions et les verrous, puis écrit des drivers platform, I2C et PCI qui utilisent le DMA, les horloges, les resets et le runtime PM comme un driver en a besoin.

Objectifs

À la fin de la formation, les participants savent :

- compiler, configurer et démarrer le noyau, et écrire des modules noyau ;
- écrire des drivers platform, I2C et PCI rattachés via le modèle de périphériques et le device tree ;
- allouer la mémoire, mapper les registres et utiliser le DMA correctement ;
- gérer les interruptions, le travail différé et les verrous ;
- utiliser les horloges, les resets et le runtime PM dans un driver ;
- déboguer un driver avec printk, ftrace, kprobes et l'analyse des oops.

Public

Ingénieurs qui écrivent ou maintiennent des drivers Linux pour du matériel spécifique, et développeurs BSP qui doivent comprendre les drivers qu'ils intègrent.

Prérequis

Une bonne maîtrise du C et de la ligne de commande Linux. Des notions de Linux embarqué sont un plus.

Programme

La gestion de l'énergie fait partie de chaque driver : elle est traitée du point de vue du driver, et non comme un sujet à part.

Jour 1

- Le noyau Linux, son histoire et son processus de développement
- Récupérer, configurer, compiler et démarrer le noyau
- Les modules noyau

Jour 2

- Les frameworks du noyau et le modèle de périphériques, les drivers caractère
- Décrire et accéder au matériel : drivers platform, identification des périphériques et données associées
- La gestion mémoire : allocation, mémoire d'entrées-sorties, remapping, endianness

Jour 3

- Dormir, attendre et ordonnancer
- Les interruptions et leurs handlers
- Le partage des ressources et les verrous
- Les drivers I2C : le bus, le modèle I2C de Linux, le device tree, le dialogue avec le composant

Jour 4

- Le DMA : l'API DMA générique, DMAEngine, IOMMU et DMA périphérique
- Les drivers PCI : énumération, enregistrement, probe, espace de configuration, BAR et DMA

Jour 5

- Les horloges et les resets côté consommateur
- Le runtime PM et les callbacks de mise en veille dans un driver
- Le débogage du noyau : printk, kprobes, ftrace, analyse des oops et des panics, perf

Travaux pratiques et matériel

Les travaux pratiques se font sur des machines QEMU et sur les cartes de notre laboratoire : Arm 32 bits : STM32MP157C-DK2, BeagleBone Black ; Arm 64 bits : Raspberry Pi 4 et 5, NXP i.MX8M Mini, Khadas VIM3 ; x86 : QEMU x86, Machines x86_64 génériques, Kits de développement Intel. D'autres cartes, ou le matériel du client, peuvent être utilisés sur demande.

Les sessions ont lieu sur site, en France, en Belgique, en Suisse, en Allemagne et au-delà, avec le matériel fourni par LABCSMART, ou à distance. À distance, chaque participant dispose de son propre compte SSH et de son espace de travail sur notre serveur de build, et réalise les travaux pratiques sur des machines QEMU et sur de vraies cartes. Les exercices sont tirés de nos projets clients. Les formations sont données en français ou en anglais ; les supports de cours, eux, sont en anglais.

Modules optionnels

Chaque module peut s'ajouter à la formation, ou en remplacer une partie, pour les équipes qui travaillent sur un sous-système donné.

- Drivers audio Linux avec ALSA SoC : PCM, contrôles, DAPM, drivers codec, platform et machine (3 jours)
- Drivers caméra Linux avec V4L2 : capteurs, sous-périphériques, MIPI CSI-2, contrôles, media controller, périphériques mémoire à mémoire (3 jours)
- Drivers de contrôleurs PCIe host et endpoint (2 jours)
- Les fournisseurs d'horloges et le common clock framework (1 jour)

Qui anime cette formation

La formation est conçue et animée par le fondateur de LABCSMART, ingénieur Linux embarqué et noyau Linux avec plus de onze ans d'expérience industrielle. Il contribue au noyau Linux en amont pour les SoC Renesas RZ, dans les sous-systèmes PCIe, audio, Ethernet, thermique, DMA et horloges, et a écrit deux livres publiés chez Packt, *Linux Device Driver Development, Second Edition* (2022) et *Mastering Linux Device Driver Development* (2021), présentés sur notre page de publications.

Organiser une session pour votre équipe. Chaque formation est composée de modules indépendants. Une équipe peut retirer les modules dont elle n'a pas besoin et en ajouter d'autres à la place, et nous construisons aussi des formations sur mesure autour d'un SoC, d'une carte ou d'un projet. Contact : contact@labcsmart.com, https://labcsmart.com/fr/contact/. Page de la formation : https://labcsmart.com/fr/formations/formation-drivers-linux/