D8 - Drivers USB Linux

Ecriture de drivers USB-2.0 et USB-3.0 sous Linux

Objectifs

- Apprendre à créer des drivers device et gadgets pour USB-2.0 et USB-3.0
  - Explorer la richesse de Linux concernant les drivers USB hôte.
  - Comprendre le support Linux des gadgets USB.
  - Découvrir le support du standard OTG (2.0 et 3.0)
- Comprendre les spécificités du noyau Linux dans la gestion des devices et des drivers.
- Savoir paramétrer le noyau Linux à la compilation et en fonctionnement pour une gestion optimale du hotplug.
  - Comprendre comment sont générés les événements hotplug et savoir les utiliser dans l'écriture de ses drivers.
  - Installer et utiliser les projets externes hotplug : Udev, libusb, etc...
- Connaître les évolutions de Linux jusqu'au noyaux 2.6.3ç et 3.x.
- Maîtriser les techniques de debugging noyau.

Les exercices se font en utilisant l'environnement de développement intégré System Workbench for Linux - Basic Editionqui est remis à tous nos stagiaires pour leur permettre de continuer, après la formation,à travailler dans un environnement convivial et efficace.

-->

Matériel

- Un pc par binôme
- Une carte cible par binôme
- Support de cours

Pré-requis

- Bonne pratique de la programmation en C sous Linux.
- Connaissance de la programmation Linux kernel (niveau cours D3)

Prerequisite

- Good practice of C programming on Linux
- Good knowledge of Linux kernel and driver programming (see our cours D3 - Drivers Linuxand cours D7 - Power Management in Linux Driverss)

Environnement du cours

- Cours théorique
  - Support de cours imprimé et au format PDF (en anglais).
  - Le formateur répond aux questions des stagiaires en direct pendant la formation et fournit une assistance technique et pédagogique.
- Activités pratiques
  - Les activités pratiques représentent de 40% à 50% de la durée du cours.
  - Elles permettent de valider ou compléter les connaissances acquises pendant le cours théorique.
  - Exemples de code, exercices et solutions
  - Un PC (Linux ou Windows) par binôme de stagiaires (si plus de 6 stagiaires) pour les activités pratiques avec, si approprié, une carte cible embarquée.

    - Le formateur accède aux PC des stagiaires pour l'assistance technique et pédagogique.
- Une machine virtuelle préconfigurée téléchargeable pour refaire les activités pratiques après le cours
- Au début de chaque demi-journée une période est réservée à une interaction avec les stagiaires pour s'assurer que le cours répond à leurs attentes et l'adapter si nécessaire

Audience visée

- Tout ingénieur ou technicien en systèmes embarqués possédant les prérequis ci-dessus.

Modalités d'évaluation

- Les prérequis indiqués ci-dessus sont évalués avant la formation par l'encadrement technique du stagiaire dans son entreprise, ou par le stagiaire lui-même dans le cas exceptionnel d'un stagiaire individuel.
- Les progrès des stagiaires sont évalués de deux façons différentes, suivant le cours:
    - Pour les cours se prêtant à des exercices pratiques, les résultats des exercices sont vérifiés par le formateur, qui aide si nécessaire les stagiaires à les réaliser en apportant des précisions supplémentaires.
    - Des quizz sont proposés en fin des sections ne comportant pas d'exercices pratiques pour vérifier que les stagiaires ont assimilé les points présentés
- En fin de formation, chaque stagiaire reçoit une attestation et un certificat attestant qu'il a suivi le cours avec succès.
    - En cas de problème dû à un manque de prérequis de la part du stagiaire, constaté lors de la formation, une formation différente ou complémentaire lui est proposée, en général pour conforter ses prérequis, en accord avec son responsable en entreprise le cas échéant.

Plan

1er jour

Programmation noyau (rappels)

- Le développement de module noyau.
- Les objets du noyau.
- Le système de fichier sysfs.

Exercise : *Ecriture d'un module noyau illustrant la création et l'utilisation des ksets et kobjs.*

Hotplug

- Hotplug Kernel : Les uevents
- Hotplug Utilisateur : Udev
- Hotplug Utilisateur : Hal et Dbus

Exercise : *Ecriture d'un module noyau émettant ses propres événements hotplug.*
Exercise : *Compilation croisé, paramétrage et utilisation de l'outil Udev.*

2ème jour

Devices et Drivers

- Le modèle device/driver sous Linux.
- Les classes et types de périphériques.
- Les types de bus.
- Devices & drivers génériques.
- Devices & drivers système.
- Devices & drivers plate-forme.

Exercise : *Ecriture d'un device et d'un driver plate-forme illustrant les mécanismes de matching interne.*

Drivers USB

- Le bus USB.
- Les périphériques USB.
- L'interface utilisateur USB.
- Les descripteurs USB.
- Les requêtes USB.
- Les pilotes USB.

Exercise : *Ecriture d'un module noyau usb illustrant l'utilisation des urbs.*
Exercise : *Ecriture d'un version du même module utilisant les requêtes usb synchrones.*

3ème jour

La libUSB

- Les bibliothèques libUSB.
- La libUSB 0.1.12.
- La libUSB 1.0

Exercise : *Compilation croisée de la libusb.*
Exercise : *Ecriture d'un driver usb tournant dans l'espace utilisateur.*

Drivers USB gadget

- Les pilotes gadget USB.
- Les pilotes gagdet USB composites.
- Les pilotes gadget USB OTG.

Exercise : *Ecriture d'un driver gadget coté cible et du driver correspondant coté pc.*

Renseignements pratiques

Renseignements : 3 jours