SETC Training Corporate Logo

Formation Linux: Développement professionnel de pilotes de périphériques Linux (LIDDEVFR, 4 jours)

Description

Le cours Formation Linux : Développement professionnel de pilotes de périphériques Linux couvre tout ce que vous avez toujours voulu savoir sur le développement de pilotes de périphériques Linux. En commençant par les bases, le cours procède rapidement à une exploration approfondie de l'espace utilisateur, de l'espace noyau, des pilotes de caractères, des pilotes de blocs, des pilotes de réseau et plus encore. Le cours se termine par un aperçu d'une variété de sujets, y compris ALSA, les solutions Linux en temps réel et plus encore.

Tarifs

- Tarification: $3,750/person
- Rabais de 10% lorsque vous inscrivez 3 personnes.

Plan de cours

Périphériques sous Linux

- Aperçu des pilotes de périphériques Linux
- Classes de dispositifs
- Création de fichiers de périphériques
- Modules du noyau et applications
- API des pilotes de l'espace utilisateur

Modules chargeables

- Pourquoi des modules chargeables ?
- Travailler avec des modules chargeables
- Code du pilote de périphérique

Compilation, chargement, exportation

- Mise en place du système de test
- Un module simple
- Compilation des modules
- Chargement et déchargement des modules
- Initialisation et arrêt
- Exportation de symboles à partir de modules chargeables
- Travailler avec des modules chargeables empilés

Périphériques de caractères

- Numéros majeurs et mineurs
- Enregistrement des fichiers de périphériques de caractères
- Méthodes de conduite
- Travailler avec les espaces utilisateurs

Types de données du noyau (vue d'ensemble seulement)

- Types C standard
- Travailler avec le dimensionnement des éléments de données
- Types spécifiques à l'interface
- Listes liées
- Autres questions

Débogage et traçage

- Utilisation de printk

Utilisation de /proc
Utilisation de strace
ksyms et ksymoops
Utilisation de gdb, kgdb
Travailler avec des files d'attente
Files d'attente
Blocage sécurisé
Programme()
Sondage()
Travailler avec la mémoire
Gestion de la mémoire sous Linux
Travailler avec mmap
E/S directes
Accès direct à la mémoire
Travailler avec le matériel
Ports d'E/S et mappage de la mémoire
Allocation et mappage de l'espace E/S
Travailler avec les ports d'E/S
Accès à l'espace utilisateur
Interruptions
Gestion des interruptions
Installation des gestionnaires d'interruptions
Partage des interruptions
Restrictions du noyau
Tasklets et Workqueues
Pilotes USB
Structure de l'USB
Travailler avec les terminaux, les interfaces et les configurations
Blocs de requêtes USB
Organisation et structure des conducteurs
Travailler avec des pilotes de gadgets
Accès au matériel depuis l'espace utilisateur
Timing
Interruptions de la minuterie
Délais courts
Files d'attente de tâches
Timers du noyau
Pilotes CHAR
Numéros majeurs et mineurs
Conception de l'écume
Utilisation de la mémoire de scull
Enregistrement de l'appareil Char
Lire et écrire
Pilotes de périphériques de bloc (vue d'ensemble uniquement)
Aperçu des pilotes de périphérique de bloc
Travailler avec des fichiers d'en-tête
Enregistrement des pilotes de périphérique de bloc
Structure de block_device_operations
Travailler avec des méthodes spéciales
Pilotes réseau

Aperçu des pilotes de réseau
Structure de net_device
Douilles
Schéma de dénomination et enregistrement
Méthodes du pilote réseau
Travailler avec NAPI
Arbre du noyau
Travailler avec l'arbre du noyau
L'agencement du noyau
Makefile
Fichier Kconfig
Sujets supplémentaires - Vue d'ensemble uniquement
Configuration et construction du noyau
Démarrage via TFTP
Système de fichiers racine sur NFS
Surveillance des systèmes de fichiers (iNotify)
Environnement de développement NFS
Initialisation à l'aide de la méthode de l'arbre des dispositifs au lieu du fichier de la carte
UDEV
Linux Real-Time