SETC Training Corporate Logo

Linux Training: Professional Linux Device Driver Development (LIDDEV, 4 jours)

Description

The course Professional Linux Device Driver Development (Linux Training) covers everything that you ever wanted to know about Linux Device Driver Development. Starting with the basics, the course quickly proceeds through an in-depth exploration of User Space, Kernel Space, Character Drivers, Block Drivers, Network Drivers & more. The course concludes with an overview of a variety of topics including ALSA, Linux real time solutions & more.

Tarifs

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

Plan de cours

Devices in Linux
- Linux Device Drivers Overview
- Device Classes
- Creating Device Files
- Kernel Modules vs. Applications
- User Space Driver APIs

Loadable Modules
- Why Loadable Modules?
- Working with Loadable Modules
- Device Driver Code

Compiling, Loading, Exporting
- Setting up the Test System
- A Simple Module
- Compiling Modules
- Loading and Unloading Modules
- Initialization and Shutdown
- Exporting Symbols from Loadable Modules
- Working with Stacked Loadable Modules

Character Devices
- Major and Minor Numbers
- Registering Character Devices Files
- Driver Methods
- Working with User Spaces

Kernel Data Types (Overview Only)
- Standard C Types
- Working with Sizing Data Items
- Interface-Specific Types
- Linked Lists
- Other Issues

Debugging and Tracing
- Using printk
- Using /proc
- Using strace

- ksyms and ksymoops
- Using gdb, kgdb

Working with Queues

- Wait Queues
- Safe Blocking
- Schedule()
- Poll()

Working with Memory

- Linux Memory Management
- Working with mmap
- Direct I/O
- Direct Memory Access

Working with Hardware

- I/O Ports and Memory Mapping
- Allocating and Mapping I/O Space
- Working with I/O ports
- User Space Access

Interrupts

- Handling Interrupts
- Installing Interrupt Handlers
- Interrupt Sharing
- Kernel Restrictions
- Tasklets and Workqueues

USB Drivers

- USB Structure
- Working with Endpoints, Interfaces and Configurations
- USB Request Blocks
- Organization and Structure of Drivers
- Working with Gadget Drivers
- Accessing Hardware from User Space

Timing

- Timer Interrupts
- Short Delays
- Task Queues
- Kernel Timers

CHAR Drivers

- Major and Minor Numbers
- scull Design
- scull's Memory Usage
- Char Device Registration
- Read and Write

Block Device Drivers (Overview Only)

- Block Device Drivers Overview
- Working with Header Files
- Registering Block Device Drivers
- Structure of block_device_operations
- Working with Special Methods

Network Drivers

- Network Drivers Overview
- Structure of net_device

Sockets
Naming Scheme and Registration
Network Driver Methods
Working with NAPI

Kernel Tree

Working with the Kernel Tree
The Kernel Layout
Makefile
Kconfig File

Additional Topics – Overview Only

Configuring and building the kernel
Booting via TFTP
Root Filesystem over NFS
Monitoring File Systems (iNotify)
Development environment NFS
Initializing using the Device Tree method instead of board file
UDEV
Linux Real-Time