

Keonics Certified Embedded Linux

Duration: 3 Months

Linux Device Driver Programming

Summary

The course aims at teaching the basics of device driver programming for the Linux kernel. The course covers LED driver, input driver and char device driver development. The course is taught on an ARM board with simple devices like the GPIOs, LEDs, seven segment displays, keypads, temperature sensors and EEPROM devices. The goal is to focus on learning the kernel interface, while still programming real hardware.

Since device driver development requires knowledge in varied disciplines, a quick review of the fundamentals of file IO, file systems, microprocessors and kernel modules is included.

Pre-requisite

- C Programming
- Programming in GNU/Linux
- Microprocessors

Syllabus

File I/O

- File Descriptors
- Opening and Closing Files
- Reading and Writing to Files Sequentially
- Reposition Read/Write File Offset
- File Tables and File Descriptor Tables

Filesystems

- Filesystems Internals - inodes, dentry
- Pseudo filesystems - proc, sysfs

Introduction to Devices

- Device Interfacing
- Controlling LEDs from Userspace
- Controlling Seven segment displays

Accessing Drivers

- Unix/Linux Device Abstraction
- Device Files
- Types of Drivers
- Mouse Device Access
- Sound Mixer Device

Kernel Compilation and Kernel Modules

- Kernel Compilation
 - Kernel configuration
 - Building the kernel
 - Booting from the new kernel
- Kernel Modules
 - Introduction
 - Advantages
 - Listing, Loading, Unloading

Writing Kernel Modules

- Module Programming vs Application Programming
- Hello World Module
 - Module Programming Basics
 - Compiling
 - Loading and Unloading
 - Module Parameters

Input Drivers

- Kernel Timers
- System Ticks
- Scheduling timers
- Keypad Input driver

Char Drivers

- Major and Minor Numbers
- Data Structures
 - File Operations Structure
 - File Structure
- Registering Char Driver
- Seven segment display

Debugging Techniques

- Debugging by Printing
 - Kernel Log Mechanism
 - Turning on/off Debug Messages
 - Rate Limiting
- Debugging by Querying
 - debugfs
- Debugging by Watching
 - strace
- Debugging System Faults
 - Decoding Oops
 - Recovering from a System Hang

Interrupts and Blocking I/O

- Interrupts
- Installing Interrupt Handlers
- Blocking I/O
- Process states
- Wait Queues

Concurrency and Locking

- Sources of Concurrency
- Semaphores
- Spinlocks
- Common Locking Problems
- Atomic Variables

Device Model

- I2C and SPI bus
- Bus Model
- Device and Driver Binding
- Module Autoloading