Linux Device Drivers

Wind River Education enables you to unleash the power of Wind River’s technology. Our training and mentoring empower developers with the knowledge and proficiency required to program and manage device software faster and more reliably. Reduce your project risks and shorten your development timelines by equipping your engineers with the right training from our experts.

Course Description

The Linux Device Drivers course provides engineers with a fast, cost-effective way to acquire the skills necessary to develop, deploy, and debug their own customized Linux device drivers.

After taking this course, participants will be able to:

• Develop and manipulate Linux kernel modules.
• Develop Linux device drivers for the various types of devices supported in Linux.
• Understand how the different kernel subsystems work and what effect they have on the structure of a device driver.
• Debug Linux device drivers.

Products Supported

• Linux kernel 2.6.33

Who Should Attend

• Anyone new to device driver development in Linux
• Linux application developers who are looking to gain more insight into how the Linux kernel works
• Developers interested in the interface between the Linux kernel and device drivers

Prerequisite Skills

• Familiarity with makefiles and GNU toolchain
• Understanding of various methods used to deploy and debug Linux-based applications in a cross-development environment
• C or C++ programming experience on Linux/UNIX

Prerequisite Courses

• Introduction to Linux

Related Courses

• Wind River Linux and Workbench Essentials
• Wind River Linux Board Support Package
• Wind River Linux 3.0 for Intermediate Platform Developers

Course Title: Linux Device Drivers
Duration: Four days
Format: Instructor-led lectures and hands-on lab sessions; instructor-led Live Remote delivery available
Price: Contact your local sales representative

Course Format

• This four-day expert-led course consists of lectures and lab sessions.
• Attendees use Linux kernel 2.6.33 to gain experience with the topics presented.
• Participants examine and exercise simulated network topologies in hands-on labs.
• Participants receive individual guidance from an expert engineer who has extensive experience with Wind River technologies.

Global Reach of Wind River Education Services

With more than 30 years of device software experience, we provide education services in every region of the world. You can rely on our expertise—acquired delivering hundreds of classes each year to thousands of students—to provide a highly effective learning experience, wherever your developers are located.

Private Classes

Private classes are conducted at your location, scheduled for your convenience. Private classes include the use of a preconfigured laboratory environment that may consist of a connection to a remote lab environment or equipment that we bring to your facility. Private classes can be tailored to your specific needs by adding or removing topics from multiple courses, maximizing the benefit of your time in class.

Mentoring

Our Mentoring programs are provided by experienced engineers whose coaching can increase your team’s productivity while reducing your project’s risk.

Whether you need assistance with product installation and configuration, advice on development workflow, or optimization best practices, mentoring can shorten your
trial-and-error cycle, document recommended procedures, and ensure your developers are using tools and technology efficiently. If you have limited time to resolve a particular issue, a Wind River expert can evaluate your system and development environment, and assist you in building a debug framework, instrumenting code, verifying software updates, and general hands-on debugging.

All of our education services are led by expert engineers who are closely connected to the Wind River technical community for access to specific expertise.

Syllabus

Day 1

**Introduction to Linux Device Drivers**
- Linux Architecture Overview
- Linux Device Driver Overview
- Device Driver Types
- Linux Device Model
- In-Tree vs. Out-of-Tree Driver Development

**Linux Kernel Source Code**
- Source Code Organization
- The Kernel Configurator
- The Kernel Build System
- Working with Kernel Patches
- Development Environment Lab

**Introduction to Linux Kernel Modules**
- Anatomy of a Kernel Module
- Module Licensing
- Building Modules
- Managing Modules
- Module Parameters
- Kernel Module Lab

**Character Device Drivers**
- Driver Life Cycle
- Major and Minor Numbers
- Character Driver Entry Points
- Blocking Operations
- Controlling a Device
- Querying Read/Write Ability
- Restricting Operations
- Character Device Driver Lab

Day 2

**Managing Memory in the Linux Device Drivers**
- How Linux Manages Memory
- Allocating Memory with kmalloc()
- Page-Based Memory Allocation
- Manipulating Memory
- Memory Mapped i/o

Day 3

**Concurrency in Linux Device Drivers**
- Concurrency
- Race Conditions
- Locking Primitives
- Deadlock
- Atomic Variables
- Concurrency Management Lab

**Managing Time in Linux Device Drivers**
- HZ
- Jiffies and Jiffies_64
- Delaying Execution
- Deferring Execution
- Time Management Lab

**Handling Interrupts Linux Device Drivers**
- How Interrupts Work
- IRQs
- Top Half Processing
- Bottom Half Processing
- Handling interrupts Lab

**Debugging Linux Device Drivers**
- Debugging by Printing
- Debugging by Querying
- Debugging by Observation
- Using a Kernel Debugger
- Debugging Drivers Lab

**Linux TTY Device Drivers**
- TTY Architecture
- TTY Driver Life Cycle
- TTY Driver Entry Points
- Line Settings
- struct tty_struct
- Using TTY Core Lab

**Linux PCI Device Drivers**
- PCI Configuration Space
- Identifying Devices
- Matching Devices and Drivers
- Driver Registration
- Probe() Function
- Memory and I/O Regions
- DMA
- Remove() Function
- Using PCI Core Lab
Day 4

**Linux USB Device Drivers**
- USB Architecture
- Matching Devices and Drivers
- Driver Registration
- Probe Function
- Communicating with URBs
- Communicating without URBs
- Using USB Core Lab

**Block Device Drivers**
- Driver Life Cycle
- Major and Minor Numbers
- Block Driver Entry Points
- Processing Requests
- Controlling a Device
- Block Device Driver Lab

**Linux Network Drivers**
- Overview of Network Devices
- Driver Registration
- Network Driver Entry Points
- Controlling Interfaces
- Packet Transmission
- Interrupt-driven Packet Reception
- NAPI Drivers
- Network Device Driver Lab

**Wind River World Headquarters**
500 Wind River Way
Alameda, CA 94501
USA
Toll-free: 800-545-9463
Tel.: 510-748-4100
Fax: 510-749-2454
training@windriver.com

**Wind River EMEA**
Steinheilstrasse 10
85737 Ismaning
Germany
Tel.: +49 89 962 445 0
Fax: +49 89 962 445 999
emea-training@windriver.com
http://education.windriver.com