LINUX DEVICE DRIVERS

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 this course you will be able to do the following:
• Develop and manipulate Linux kernel modules
• Develop Linux device drivers for the various types of devices supported by Linux
• Describe the workings of the different kernel subsystems and how they impact the structure of a device driver
• Debug Linux device drivers

PRODUCTS SUPPORTED
• Wind River Linux 7.0
• QEMU-simulated target for Intel® x86-64 processors

COURSE FORMAT
• This four-day, expert-led course consists of lectures and lab sessions.
• Students gain hands-on experience and receive personal guidance from expert Wind River instructors.
• Specific questions are addressed.

AUDIENCE
• Anyone new to device driver development in Linux
• Linux application developers who need 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

RELATD COURSES
• Wind River Linux and Workbench Essentials
• Wind River Linux BSP Development

SYLLABUS
Day 1
INTRODUCTION TO LINUX DEVICE DRIVERS
• Linux architecture overview
• Linux device driver overview
• Device driver types
• Linux device model

LINUX KERNEL SOURCE CODE
• Source code organization
• The kernel configurator
• The kernel build system
• Working with kernel patches
• LAB: Getting Started with the Wind River Linux Lab Environment
INTRODUCTION TO LINUX KERNEL MODULES
- Overview
- Anatomy of a kernel module
- Module licensing
- Building modules
- Installing modules
- Managing modules
- Module parameters
- LAB: Developing a Basic Kernel Module

CHARACTER DEVICE DRIVERS
- Overview
- Driver lifecycle
- Major and minor numbers
- Character driver entry points
- Blocking operations
- Controlling a device
- Querying read/write ability
- Restricting operations
- LAB: Developing a Character Device Driver

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
- Accessing user space memory
- Implementing the mmap operation
- LAB: Managing Memory in Kernel Code

CONCURRENCY IN LINUX DEVICE DRIVERS
- Concurrency
- Race conditions
- Locking primitives
- Deadlock
- Atomic variables
- LAB: Managing Concurrency in Kernel Code

MANAGING TIME IN LINUX DEVICE DRIVERS
- Measuring time in the kernel

- Delaying execution
- Deferring execution
- LAB: Managing Execution of Driver Code

Day 3
HANDLING INTERRUPTS IN LINUX DEVICE DRIVERS
- How interrupts work
- IRQs
- Interrupt handlers
- LAB: Implementing Interrupt Handlers

DEBUGGING LINUX DEVICE DRIVERS
- Debugging by printing
- Debugging by querying
- Debugging by observation
- Using a kernel debugger
- LAB: Implementing Debugging Strategies in Kernel Code
- LAB: Configuring KGDB
- LAB: Debugging the Kernel with GDB

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
- LAB: Driving Devices over PCI

Day 4
LINUX USB DEVICE DRIVERS
- USB architecture
- Matching devices and drivers
- Driver registration
- Probe function
- Communicating with the device
- LAB: Driving Devices over USB
BLOCK DEVICE DRIVERS
- Driver lifecycle
- Major and minor numbers
- Block driver entry points
- Processing requests
- Controlling a device
- LAB: Developing a Block Device Driver

NETWORK DEVICE DRIVERS
- Overview of network devices
- Driver registration
- Network driver entry points
- Controlling interfaces
- Packet transmission
- Packet reception
- LAB: Developing a Network Device Driver

GLOBAL REACH OF WIND RIVER EDUCATION SERVICES
With more than 30 years of device software experience, Wind River provides education services in every region of the world. Our private classes can be tailored to your needs by adding or removing topics from multiple courses. If you have more specific project challenges, Wind River Mentoring provides coaching by experienced engineers to help you integrate Wind River solutions into your environment. And when you’re too busy to attend a whole class, our On-Demand Learning options provide around-the-clock access to advanced and specialized topics. 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.

CONTACT US
For more information about Wind River Education Services, visit www.windriver.com/education/.

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

Wind River is a world leader in embedded software for intelligent connected systems. The company has been pioneering computing inside embedded devices since 1981, and its technology is found in nearly 2 billion products. To learn more, visit Wind River at www.windriver.com.

©2015 Wind River Systems, Inc. The Wind River logo is a trademark of Wind River Systems, Inc., and Wind River and VxWorks are registered trademarks of Wind River Systems, Inc. Rev. 12/2015