WNDRVR

WIND RIVER LINUX DEVICE DRIVER AND BOARD SUPPORT PACKAGE DEVELOPMENT

COURSE DESCRIPTION

The Wind River Linux Device Driver and Board Support Package Development 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 and BSPs in the Wind River® Linux environment.

- After this course, participants will be able to perform the following:
- Develop and manipulate Linux kernel modules
- Develop Linux device drivers for the various types of devices supported in Linux
- Describe the workings of the different kernel subsystems and how they impact the structure of a device driver
- Debug Linux device drivers
- Create a new BSP, and customize, patch, and validate the BSP
- Port Wind River Linux platform projects to new boards

PRODUCTS SUPPORTED

- Wind River Linux LTS 21
- The following targets are available: QEMU simulated target (Intel® x86-64)

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

- Developers new to device driver and BSP development in Linux
- Linux application developers who need insight into the Linux kernel
- Developers interested in the interface between the Linux kernel and device drivers
- Developers who plan to use Wind River Linux on an unsupported board (hardware architecture is supported)

PREREQUISITE SKILLS

- Familiarity with the Yocto build environment

Course title:	**Wind River Linux Device Driver and Board Support Package Development**
Duration:	Four days
Format:	Instructor-led lectures and hands-on lab sessions; instructor-led Live - Remote delivery available
Content:	**Day 1**: Introduction to Linux Device Drivers; Linux Kernel Source Code; Introduction to Linux Kernel Modules; Character Device Drivers **Day 2**: Managing Memory in the Linux Device Drivers; Concurrency in Linux Device Drivers; Handling Interrupts in Linux Device Drivers; Debugging Linux Device Drivers **Day 3**: Linux PCI Device Drivers; Linux USB Device Drivers; Block Device Drivers; Linux Network Device Drivers **Day 4**: Wind River Linux BSP Overview; Creating Wind River Linux BSPs; Additional BSP Considerations

- Familiarity with makefiles and the GNU toolchain
- Understanding of how to deploy and debug Linux-based applications in a cross-development environment
- C or C++ programming experience on Linux/UNIX
- Familiarity with the Helix Platform hypervisor

PREREQUISITE COURSES

- Introduction to Linux

RELATED COURSES

- Wind River Linux CLI Essentials

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

- Configuring the kernel
- Building the kernel
- Patching the kernel
- **LAB: Getting Started with the Wind River Linux Lab Environment**
- **LAB: Managing Simulated Targets from the Command Line**

INTRODUCTION TO LINUX KERNEL MODULES

- Overview
- Building modules
- Installing modules
- Managing modules
- Module parameters
- **LAB: Managing Kernel Modules**
- **LAB: Developing Kernel Modules**

CHARACTER DEVICE DRIVERS

- Overview
- Driver Life Cycle
- Major and Minor Numbers
- Character driver entry points
- Blocking operations
- Controlling a device
- Querying the 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**
- **LAB: Managing Execution of Driver Code**

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: Kernel Debugging with GDB**

Day 3

LINUX PCI DEVICE DRIVERS

- PCI configuration space
- Identifying devices
- Matching devices and drivers
- Driver registration
- The **probe( )** function
- Memory and I/O regions
- DMA
- The **remove( )** function
- **LAB: Driving Devices over PCI**

LINUX USB DEVICE DRIVERS

- USB architecture
- Matching devices and drivers
- Driver registration
- The **probe( )** function
- Communicating with the device
- **LAB: Driving Devices over USB**

BLOCK DEVICE DRIVERS

- Driver Life Cycle
- Major and Minor Numbers
- Block Layer Architecture
- Block Driver Entry Points
- Processing Requests
- Controlling a Device
- **LAB: Developing a Block Device Driver**

LINUX NETWORK DEVICE DRIVERS

- Overview of network devices
- Implementing the driver
- Packet transmission and reception
- NAPI drivers
- **LAB: Developing a Network Device Driver**

Day 4

WIND RIVER LINUX BSP OVERVIEW

- The role of a Wind River Linux BSP
- The Wind River Linux BSP structure
- Setting up the build environment
- Configuring and patching a kernel
- Configuring user space

CREATING WIND RIVER LINUX BSPS

- BSP development overview
- Starting from scratch
- Starting with third-party code
- Enabling supported boards
- Cloning BSPs
- Packaging Wind River Linux BSPs
- **LAB: Creating a BSP**
- **LAB: Customizing the BSP**
- **LAB: Patching the Kernel in the BSP Layer**

ADDITIONAL BSP CONSIDERATIONS

- BSP documentation
- Boot loaders
- Legal requirements
- Validating a BSP
- **LAB: Validating the Kernel**

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

WNDRVR

Wind River is a global leader in delivering software for the intelligent edge. Its comprehensive portfolio is supported by world-class professional services and support and a broad partner ecosystem. Wind River is accelerating digital transformation of critical infrastructure systems that demand the highest levels of safety, security, and reliability.