Embedded Linux Networking training

Course duration

⏱ **3** days – 24 hours

Language

Materials   English

Oral Lecture   English
                 French

Trainer

One of the following engineers

- Maxime Chevallier

Contact

@ training@bootlin.com

☎ +33 484 258 097

bootlin
bootlin.com

Audience

Engineers working on networking support in Linux-based embedded devices

Training objectives

- Be able to understand the overall Linux kernel networking stack and configure complex network devices
- Be able to understand the flow of network packets in a Linux system, use different socket types, generate and filter traffic
- Be able to use the eBPF and XDP technologies for improved network traffic processing
- Be able to understand the architecture of Linux kernel network drivers
- Be able to understand how Ethernet PHYs and switches are supported in the Linux kernel
- Be able to debug and troubleshoot low-level network related issues

Prerequisites

- **Minimal experience in embedded Linux development**: participants should have a minimal understanding of the architecture of embedded Linux systems: role of the Linux kernel vs. user-space, development of Linux user-space applications in C. Following Bootlin's Embedded Linux course allows to fulfill this pre-requisite.
- **Experience with low-level development in Linux and hardware interfaces**: participants should have a minimal understanding of memory management, interaction with common hardware interfaces (registers, interrupts), the interaction between Linux user-space applications and the Linux kernel (system calls). Following Bootlin's Linux kernel driver development course allows to fulfill this pre-requisite.
- **Minimal English language level: B1**, according to the *Common European Framework of References for Languages*, for our sessions in English. See the CEFR grid for self-evaluation.

Pedagogics

- Lectures delivered by the trainer: 50% of the duration
- Practical labs done by participants: 50% of the duration
- Electronic copies of presentations, lab instructions and data files.

Certificate

Only the participants who have attended all training sessions, and who have scored over 50% of correct answers at the final evaluation will receive a training certificate from Bootlin.

Disabilities

Participants with disabilities who have special needs are invited to contact us at *training@bootlin.com* to discuss adaptations to the training course.

Required equipement

For on-site session delivered at our customer location, our customer must provide:

- Video projector
- One PC computer on each desk (for one or two persons) with at least 16 GB of RAM, and Ubuntu Linux 24.04 installed in a free partition of at least 30 GB
- Distributions other than Ubuntu Linux 24.04 are not supported, and using Linux in a virtual machine is not supported.
- Unfiltered and fast connection to Internet: at least 50 Mbit/s of download bandwidth, and no filtering of web sites or protocols.
- PC computers with valuable data must be backed up before being used in our sessions.

For on-site sessions organized at Bootlin premises, Bootlin provides all the necessary equipment.
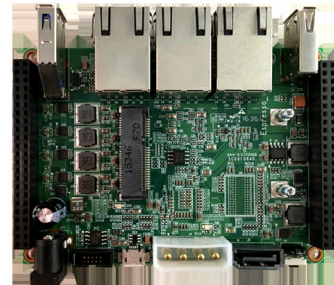
Hardware platform for practical labs

Hardware platform for practical labs

Globalscale EspressoBin board

- Dual Cortex A53 Marvell Armada 3720 SoC
- Onboard switch with 2x 1Gbps interfaces
- Extra 1Gbps interface
- 1GB RAM
- 1x SATA interface
- 1x USB 3.0 interface

Day 1 - Morning

Lecture	Networking stack and network devices in Linux	- Network stack overview in the linux kernel - What is a network interface, overview of a `net_device` - Overview of Ethernet, Wifi, CAN, Bluetooth, 802.15.4 - Stacked network devices and virtual network devices for VLAN, bridging, bonding - Switchdev and DSA devices - Control plane through *Netlink* and *ioctl*
Lab	Setting up and configuring network interfaces	- Basic setup with `iproute2` - Create bridges, VLAN interfaces with `iproute2` - Use network namespaces for interface isolation and testing - Basic use of `tcpdump` and `wireshark` - Using `ethtool` and `iproute2` to query the network interface features

Day 1 - Afternoon

Lecture	Path of a packet through the Linux networking stack	- Discover the *Socket API*, the various families and types of sockets - Sending and receiving data in userspace through sockets - Using traffic generators and analysers in userspace with *Scappy* and *Wireshark* - Path of a packet through the kernel, from a socket to a network driver - Traffic filtering through *Netfilter* and `iptables` - Traffic manipulation with the Traffic Control (`tc`) tool - Queueing control with `tc` for performance optimisation and Time-Sensitive Networking (TSN)
Lab	Sending and receiving traffic through sockets	- Write a small tool using the various socket types - Analyze the traffic through `wireshark` and `tcpdump` - Filtering the traffic with *Netfilter* and `tc` - Using traffic generators and performance measuring tools

Day 2 - Morning

Lecture	eBPF for networking	- Introduction to eBPF - Compiling and loading eBPF programs - BPF hooks in the networking stack - Introduction to XDP
Lab	Writing and using an XDP program	- Write and load a simple XDP program to filter incoming traffic - Use maps to configure the filter from userspace

Day 2 - Afternoon

Lecture	Network device drivers	- Overview of the hardware components and interfaces used in networking: MAC, PHY, MII, MDI, etc. - Infrastructure of a typical Ethernet controller driver - Sending and receiving packets with Napi - Managing buffers and queues - Packet timestamping for PTP - Overview of *ethtool* driver operations for configuration and reporting - Offloading network processing to the hardware
Lab	Advanced Ethernet configuration	- Investigating ethernet parameters controllable with *ethtool* - Using the various offloading features

Day 3 - Morning

Lecture	Ethernet PHYs and switch support	- Ethernet PHYs support in the kernel with *phylib* - Interacting with PHYs through MDIO - Dealing with the PHY to MAC connection with *phylink* - Switch support through the *DSA* framework - Dealing with switch operations with *switchdev*

Day 3 - Afternoon

Lecture	Network debugging and troubleshooting	- Analyzing performances and packet drops with monitoring tools - Debugging techniques for driver troubleshooting - Using tracing tools and `perf` for performance analysis - Diagnose hardware-related issues
Lab	Optimizing the speed in various scenarios	- Diagnosing and optimizing traffic speed - Analyzing and troubleshooting latencies