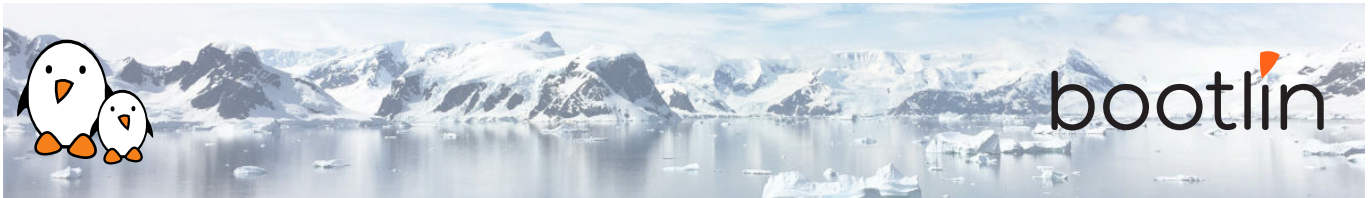


Boot Time Optimization Training

1-day session

Title	Boot Time Optimization Training
Overview	Measuring boot time Reducing user space boot time Reducing kernel boot time Bootloader optimizations Advanced techniques and alternatives Practical labs with ARM boards (Beagle Bone Black boards).
Materials	Will soon be published on https://bootlin.com/doc/training/boot-time .
Duration	Two days - 16 hours. 25% of lectures, 75% of practical labs.
Trainer	One of the engineers listed on https://bootlin.com/training/trainers/
Language	Oral lectures: English or French. Materials: English.
Audience	People developing embedded Linux systems. People supporting embedded Linux system developers.
Prerequisites	Knowledge and practice of Unix or GNU/Linux commands People lacking experience on this topic should get trained by themselves, for example with our freely available on-line slides: https://bootlin.com/blog/command-line/ Knowledge and practice of embedded Linux system development



Required equipment	<p>For on-site sessions only. Everything is supplied by Bootlin in public sessions.</p> <ul style="list-style-type: none">• Video projector• PC computers with at least 8 GB of RAM, and Ubuntu Linux installed in a free partition of at least 30 GB. Using Linux in a virtual machine is not supported, because of issues connecting to real hardware.• We need Ubuntu Desktop 18.04 (Xubuntu and other variants are fine). We don't support other distributions, because we can't test all possible package versions.• Connection to the Internet (direct or through the company proxy).• PC computers with valuable data must be backed up before being used in our sessions. Some people have already made mistakes during our sessions and damaged work data.
Materials	Print and electronic copies of presentations and labs. Electronic copy of lab files.

Day 1 - Morning

Lecture - Principles	Lab - Preparing the system
<ul style="list-style-type: none">• How to measure boot time• Main ideas	<ul style="list-style-type: none">• Compiling the bootloader, kernel and root filesystem• Setting up the system, customizing the build system

Lecture - Measuring time
<ul style="list-style-type: none">• Generic software techniques• Hardware techniques• Specific solutions for each stage



Day 1 - Afternoon

Lab - Measuring time

- Modify the system to measure time at various steps
- Measure total boot time by toggling a GPIO

Lecture - Toolchain optimizations

- Introduction to toolchains
- C libraries
- Size information
- Measuring executable performance with time

Lecture - Application optimization

- Using `strace`
- Other profiling techniques

Lab - Application optimization

- Finding unnecessary configuration options in applications
- Modifying configuration options through Buildroot
- Experiments with `strace` to trace program execution

Day 2- Morning

Lecture - Optimizing system initialization

- Using Bootchart
- Optimizing init scripts

Lab - Optimizing system initialization

- Using Buildroot to remove unnecessary scripts and commands
- Access-time based technique to identify unused files
- Simplifying BusyBox
- Starting the application as the init program



Lecture - Filesystem optimizations

- Available filesystems, performance and boot time aspects
- Tweaks for reducing boot time
- Booting on an initramfs

Lab - Filesystem optimizations

- Trying and measuring two block filesystems: ext4 and SquashFS.
- Trying and measuring the initramfs solution. Constraints due to this solution.

Day 2 - Afternoon

Lecture - Kernel optimizations

- Using *Initcall debug* to generate a boot graph
- Compression and size features
- Tweaks to reduce boot time

Lab - Kernel optimizations

- Generating and analyzing a boot graph for the kernel
- Find and eliminate unnecessary kernel features
- Find the best kernel compression solution for our system

Lecture - Bootloader optimizations

- Compiling U-Boot with less features
- U-Boot configuration settings that impact boot time
- Optimizing kernel loading
- Skipping the bootloader

Lab - Bootloader optimizations

- Using the above techniques to make the bootloader as quick as possible.