

Mastering Embedded Linux

This intensive four-day course will teach you the techniques required to build Linux into embedded systems. During the hands-on sessions you will learn about the fundamental elements that underpin all embedded Linux projects: the toolchain, the bootloader, the Linux kernel and the root filesystem. You will create each of these elements from scratch, and discover out how to automate the process using tools such as Buildroot and the Yocto Project. In later sessions you will delve into architectural issues such as file system layout, how to split functions between user and kernel space and real-time programming. You will also find out how to debug and profile code at both the application level and within the kernel so that you will be able to identify bugs and resolve performance bottlenecks.

Duration

4 days

Audience

This course is ideal for software engineers who are familiar with embedded devices but need to apply that knowledge to Linux development, and to those who are familiar with Linux, but want to apply that knowledge to embedded systems

Prerequisites

Essential: good knowledge of the C programming language, since this is used in the

programming portions of the course

Desirable: *either* a good background in embedded devices, *or* a reasonable proficiency in Linux command-line tools. Delegates with neither will find the learning curve rather steep

Course materials

All students will receive:

- A printed copy of the presentations and lab notes
- Worked solutions to the lab sessions
- A free copy of the trainer's book, "Mastering Embedded Linux Programming"

About the trainer



Chris Simmonds has been using Linux in embedded systems for over 15 years. He has been running training courses and workshops in embedded Linux since 2002 and has delivered hundreds of sessions to many well-known companies in the UK, Europe, USA, South America and SE Asia. He is the author of the book "Mastering Embedded Linux Programming", and is a frequent presenter at open source and embedded conferences, including Embedded Linux Conference and Embedded World. You can see some of his work on the "Inner Penguin" blog at www.2net.co.uk

Enquiries and bookings

Please email training@2net.co.uk or call +44 (0)1962 869003

Mastering Embedded Linux

Course outline

Embedded Linux

- Linux as an embedded operating system
- Working with open source licenses

Toolchain, bootloader and kernel

- The toolchain: choosing, installing and testing
- The bootloader: how to load Linux into memory using U-Boot
- The Linux kernel: customising and cross-compiling
- Device trees: how they work and how to modify them to fit your hardware

Root filesystem and networking

- Directory layout
- Important programs: init and the shell
- Creating a basic network configuration
- Creating user accounts

Linux build systems

- Buildroot
- The Yocto Project

Init, device manager and log daemon

- Choices for init: Busybox, SysV, systemd
- Examples using Busybox init and systemd
- Device managers: populating /dev
- Options for the system log daemon

File systems and flash memory

- Types of flash memory: NOR, NAND and eMMC
- Choosing the right file system: JFFS2, UBIFS, EXT4 and tmpfs
- Designing a robust in-field update mechanism

Linux kernel programming and device drivers

- Writing kernel code: kernel modules
- Anatomy of a simple device driver
- Kernel debugging: interpreting an oops message; interactive debugging using kgdb

Developing and debugging

- Application program interfaces: the POSIX standard
- Calling device drivers from user space
- Debugging applications remotely using gdbserver

Real-time

- Linux scheduling policies
- Kernel preemption and scheduling latencies
- Real-time: hard or soft? Impact on kernel and applications
- Approaching hard real-time with the PREEMPT_RT patch