

Embedded Linux

This training course will help you get up and running with embedded Linux. During the hands-on sessions, you will learn how to use open source components, including the Yocto Project, to build a reliable Linux system that will run on a typical embedded development board, the BeagleBone Black. You will learn about cross toolchains, the role of the bootloader, the importance of device trees, and the Linux kernel. You will delve into architectural issues such as file system layout, how to split functions between user and kernel space and real-time programming.

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
- A USB flash drive containing worked solutions to the problems, plus electronic copies of the course materials
- 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

Embedded Linux

Course outline

Embedded Linux

- Linux as an embedded operating system
- Working with open source licenses
- Embedded Linux build systems: introduction to the Yocto Project

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

The Yocto Project

- Layers and recipes
- Understanding Bitbake
- Creating your own meta BSP layer
- Customising U-Boot and the kernel recipes

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

- Real-time: hard or soft? Impact on kernel and applications
- Approaching hard real-time with the PREEMPT_RT patch