

## Developing for Embedded Linux

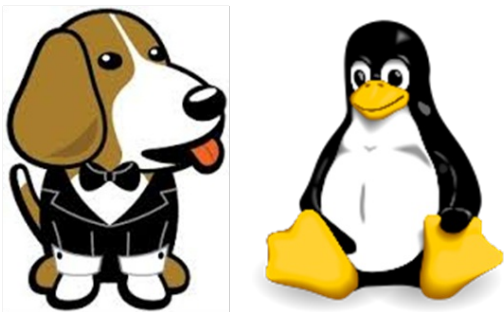
---

<b>Course category</b>	Embedded Linux
<b>Training area</b>	Operating Systems
<b>Course code</b>	EL-503
<b>Duration</b>	5 days
<b>Additional information</b>	For price: Contact Feabhas (available for on-site delivery only)

Linux is often used as an embedded operating system, and yet many still regard it as something of a black art.

This course sheds light and brings clarity by showing exactly how to deploy Linux on a typical embedded target board through a combination of theory and practice.

Starting with a board without an operating system, delegates will go through a simulated product cycle during which they will build and boot a Linux kernel, build a root file system, write a device driver and a multi-threaded application. Finally, they will review the performance of the resulting system and consider what changes could be made to improve its real-time performance.



### Overview:

A five day course showing how to implement Linux on a typical development board (ARM Cortex-A8).

### Course objectives:

- Describe the four essential components of an embedded project: toolchain, kernel, bootloader and root file system
- Demonstrate how to control hardware from a device driver (in outline, see EL-504: Developing Linux Device Drivers for a more in-depth treatment)
- Provide an overview of application development, profiling and debugging
- Show how to configure flash memory for robust code and data storage and a look at new developments in that space.

- Look at user space development, process handling and the IPC facilities provided by the Linux Kernel

### **Delegates will learn:**

- How to configure and build a customised Linux kernel
- How to construct a compact root file system from scratch
- How to develop and debug code for the target board, using the Eclipse IDE
- How to write single and multi-threaded programs using POSIX functions
- How to use Linux for product development and the benefits it brings
- Gain an in-depth understanding of modern Linux for embedded and what that means for them

### **Who should attend?**

Software engineers who are developing applications for embedded or real-time Linux. Engineers wishing to assess the suitability of Linux for their next application.

### **Pre-requisites:**

- Good 'C' programming skills
- General knowledge of an RTOS or embedded operating systems
- Experience of using Linux or a version of Unix or attendance on course EL-205 : Linux Core Skills

### **Duration:**

Five days

### **Course materials:**

Student Workbook

### **Course workshop:**

The target platform will be the BeagleBone Black which uses an ARM Cortex-A8 and will help delegates understand the issues encountered when writing for embedded platforms.

The course presents embedded and real-time concepts applied to Linux using this target. The host development system is a standard PC running Linux. We use the target as an example of a modern embedded system which can control and interact with many available interfaces including USB.

Lab sessions follow a logical sequence, and result in a Linux-powered web-controlled rocket launcher.

### **Introduction**

- What defines embedded Linux
- The 4 basic elements: toolchain, bootloader, kernel, root file system

## **The Linux Kernel**

- Virtual memory & processes
- Configuration and cross-compiling

## **Booting Linux**

- The Linux boot sequence
- Boot-loaders: U-boot
- loading images using TFTP
- The init process including systemd

## **The root file-system**

- Creating a minimal root filesystem with Busybox
- Choosing a C library
- Creating a RAM disk image

## **Network configuration**

- Static and dynamic IP addresses
- mounting the root file system over NFS

## **Device drivers**

- Creation and manipulation of device driver
- Hotpluggable drivers
- kernel modules

## **Debugging**

- Logging using syslogd
- remote debugging using Eclipse and gdbserver.

## **POSIX programming:**

- Processes: fork and exit, scheduling
- signals and signal handlers

- time and timers
- pipes, message queues, semaphores, shared memory

## **POSIX Threads**

- Threads vs processes
- synchronising threads using mutexes and condition variables.

## **Flash memory**

- The Linux Memory Technology Devices Layer and eMMC

## **File systems**

- File-systems for embedded Linux
- UBIFS, squashfs, tmpfs, jffs2 and more

## **Profiling**

**Feabhas Ltd - UK Office** 15-17 Lotmead Business Park, Wanborough, Swindon, SN4 0UY, UK ·

+44 (0) 1793 792909 · [info@feabhas.com](mailto:info@feabhas.com) · [www.feabhas.com](http://www.feabhas.com)