



# EL-504: Developing Linux Device Drivers

## Course Description:

Implementing Linux on custom hardware will, in most cases, require you to write device drivers. This course demonstrates how to write code that runs within the Linux kernel to handle hardware events and present a standard interface to applications.

This course presents a detailed view of Linux device drivers with an emphasis on topics specific to embedded environments: cross compilation; remote debugging and real-time. It uses a combination of theory and practice, using a development board with an ARM core and a recent version of Linux 2.6. No prior knowledge of Linux device drivers is assumed, making it ideal for engineers porting from code from an RTOS to Linux.

## Course Objectives:

- Demonstrate how to write drivers for custom hardware
- Provide insight into porting drivers from an RTOS to Linux, e.g. the separation between application and kernel code
- Describe the development tools needed, including debug strategies
- Examine the way drivers can affect real-time behaviour and best practice to avoid scheduling latencies

## Delegates will learn:

- How to write kernel modules
- How to create robust drivers using mutexes and spinlocks to serialise access to shared data
- How to debug kernel code running on a remote embedded target
- How to handle interrupts, including deferred processing using tasklets and work queues
- How to access hardware resources, including devices connected via PCI
- The details of memory management and memory mapping techniques

## Pre-Requisites:

- Good 'C' programming skills
- General knowledge of an RTOS or embedded operating systems
- Knowledge of Linux or Unix is useful, but not essential

## 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.

## Duration:

Five days.

## Course Material:

- Student workbook

## Related courses:

- EL-503: Developing for Embedded Linux
- EL-301: Graphics for Embedded Linux

## Laboratory Exercises:

During the lab sessions, students will write several fully-function device drivers, including a fifo, a RAM disk and a loop-back network interface. All exercises are developed and cross-compiled on a PC running Linux and downloaded to an ARM9 development board as the target (the Digi ConnectCore Wi-9C), emphasising the issues encountered when writing for embedded platforms.

## Course Outline:

### Writing Kernel Modules

- Structure of a kernel module
- compiling and loading modules

### Introduction to character device drivers

- Major and minor numbers
- basic operations – open, read, write and release
- example driver based on a fifo

### Debugging Kernel code and device drivers

- kernel oops messages
- Debugging with gdb and kgdb

### The Linux driver model

- sysfs and the /sys directory
- adding device classes and class attributes

### Task synchronisation

- Putting tasks to sleep using wait queues
- Re-entrancy issues
- mutexes, semaphores and spinlocks

### Real Time Linux

- How real-time is Linux?
- things that affect real-time behaviour
- kernel pre-emption modes

### Input and output

- Interfacing with the real world
- accessing memory and i/o mapped resources

### Time

- Delays and sleeps
- Using kernel timers.

### Interrupts

- Installing an interrupt handler; interrupt context and process context
- deferred processing using a bottom half or tasklet.

### Memory management

- allocating memory by pages and bytes
- slab caches
- Techniques to map device memory directly into user space using mmap
- getting direct access to user buffers

### Block Device Drivers

- Anatomy of a block device: example RAM disk driver

### Devices on a PCI bus

- How to access PCI hardware
- an outline PCI driver

### Network Device Drivers

- Anatomy of a network device: example loop-back interface

### Board Support Packages

- Customising the Linux configuration menus

FEABHAS

## Feabhas Ltd

5, Lowesden Works  
Lambourn Woodlands  
Hungerford, Berkshire  
RG17 7RY, UK

Tel: +44 (0) 1488 73050

Fax: +44 (0) 1488 73051

## Email:

info@feabhas.com

## Web:

www.feabhas.com