

Android Internals

Knowing how Android works will help you make better use of the system and adapt it to work for you in novel applications. This course takes a deep dive into the internal workings of Android. You will learn about the architectural layers, the security model, and the main services such as activity manager, power manager and package manager. During the hands-on labs you will use an embedded development board – a BeagleBone Black – to create an embedded Android device which will include a single-use kiosk-mode app. You will find out how to import non-Android code, and how to interface with external hardware via USB and other peripheral interfaces. Other topics include debugging native code and profiling CPU, RAM, and power usage of the whole platform.

Duration

4 days

Audience

This course is ideal for engineers who want to know how Android works, and who want to customise and extend the platform

Prerequisites

Good knowledge C/C++ and Java

Familiarity with Linux development and command-line tools

Course materials

All students will receive:

- A printed copy of the presentations and lab notes
- Worked solutions to the problems, plus electronic copies of the course materials

About the trainer



Chris Simmonds has a wealth of experience in customising Android for embedded systems. He has been running training courses and workshops in Android since 2010 and has trained teams at many well-known organisations 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)7788 130719

Android Internals

Course outline

Introduction to Android

- Android architecture: the big picture
- Getting the source code: the Android Open Source Project (AOSP)
- How to configure AOSP for a specific device
- Building Android from source

The kernel

- Android-specific changes to Linux
- Getting the source for vendor kernels
- Building the kernel
- Booting the kernel: device trees, RAM disks and boot blobs

The native layer

- The init program and boot scripts
- Adding your own native services
- System properties: how they are defined and how to change them
- The Hardware Abstraction Layer (HAL)

The Android framework

- Binder and service manager
- Android services
- The interface between the framework and native layer: JNI

Services

- Activity manager: the application lifecycle and handling Intents
- Package manager: managing user IDs and permissions
- Power manager: power states and wake locks

Applications

- Application sandboxing
- Dalvik and ART
- Crating a new run-time via the Zygote
- Writing a single-use kiosk-mode application

Security

- User and group IDs
- Android permissions, and how to add your own
- How SE Linux adds an extra layer of protection
- Security threats and “rooting”

Interfacing with hardware

- Basic interfacing using GPIO
- Interfacing with serial ports
- Interfacing using i2c and SPI peripherals
- USB host and accessories

Non-Android code

- Problems with integrating non-Android code
- Using a chroot jail
- Using Unix sockets to link into the framework and apps

Debugging and profiling

- Debugging native code
- Profiling CPU usage using perf
- Profiling memory usage
- Profiling power usage