

Embedded Android

Android is becoming a popular choice as an embedded operating system, largely because of its great UI and familiar programming interface. Android is often used in applications such as point of sale, test and measurement, industrial control, and information kiosks. The source code for Android is available through the Android Open Source Project (AOSP).

In this four day course you will design your own Android product, selecting the packages and configuration appropriate for the target platform. Beginning with a clean copy of AOSP, you will build the system images and flash them using fastboot. You will add a custom native service – a web server – and integrate it with the build. Then you will learn about the Android security framework and SELinux, and add SELinux policy files for the changes you have made. Later modules describe how SurfaceFlinger renders graphics for various displays, and how the framework integrates with specific hardware using the vendor HAL and the HAL Interface Definition Language, HIDL. Finally, there is the all-important topic of debugging and profiling the final system. All code and examples are based on **Android version 8.1 “Oreo”** and covers the new hardware abstraction layers introduced in **Project Treble**.

Roughly half of the course is taken by hands-on lab sessions during which you will apply the theory to create an embedded Android device.

Duration

4 days

Prerequisites

Good knowledge of 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

Audience

This course is ideal for system architects, engineers and project leaders who want to know how Android works, and how to customize and extend the platform

About the trainer



Chris Simmonds has a wealth of experience in customizing 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

Embedded Android

Course outline

Introduction to Android

- Android architecture: the big picture
- The impact of Project Treble
- The Android Open Source Project (AOSP)
- Building Android from source

Creating a new Android product

- Creating a new product
- Setting basic hardware characteristics
- Adding the product to the “lunch” menu

The kernel

- Android-specific changes to Linux
- Google common kernel and vendor kernels
- Building Linux from source
- Understanding and modifying device trees

Booting Android

- Learning how the bootloader starts Android
- Flashing images using fastboot
- Customizing the boot scripts
- Controlling system behaviour using system properties

The Android build system

- Building a product from packages and modules
- Android makefiles (Android.mk)
- Android blue print files (Android.bp)

The Hardware Abstraction layer

- Changes to the HAL since Project Treble
- The Vendor INTerface: VINTF
- Types of HAL
 - legacy HAL
 - passthrough and Same Process
 - binderized HAL
- HIDL – the HAL Interface Definition
- Run-time linking: the Vendor Native Development Kit (VNDK)

Testing; CTS and VTS

- Testing the application interface using the Compatibility Test Suite, CTS
- Testing the Vendor HAL using the Vendor Test Suite, VTS

The Android framework

- Understanding Android services
- The role of binder and AIDL in defining interfaces to services
- Using JNI to call native code

Graphics

- How SurfaceFlinger orchestrates image rendering
- Internal and external displays
- Streaming multi-media

Applications

- Application sandboxing
- Dalvik and ART
- How to write a single-use **kiosk-mode** application

Debugging and profiling

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

Security

- User and group IDs
- Android permissions
- Understanding and modifying SELinux policy files
- Security threats and “rooting”

Integrating non-Android code

- Problems with integrating non-Android code
- Using a chroot jail