

Embedded Android

There are many courses that teach you how to write applications for Android. This is not one of them. Instead, this course focuses on the Android platform itself. It will teach you how to take the source code from the Android Open Source Project (AOSP) and build a custom Android image for a wide range of applications. Typical use-cases include test equipment, point of sale, and industrial control terminals. Usually, Android is **embedded** into the system so that the end user sees only the user interface that controls the device.

During this four day course you will design your own product using the latest version of Android, **Android 10**. Beginning with a clean copy of AOSP, you will build system images and use them to boot the target device. You will create a new Android **System Service**, defining the interface using the Android Interface Definition Language, **AIDL**. You will learn about the Android security framework and **SELinux**, and add SELinux policy files for the changes you have made. Later, you will learn about the **Hardware Abstraction Layer**, the **HIDL** interface language, and **binderized** and **passthrough** transports. Finally, there is the all-important topic of debugging and profiling the final system.

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. At the moment it is only available on-line, using a virtual classroom. You will be building AOSP in the cloud and downloading images so they can be tested locally using the Android emulator.

Duration

4 days

Prerequisites

Good knowledge of C/C++ and Java

Familiarity with embedded Linux concepts (toolchain/bootloader/kernel/root filesystem)

Familiarity with Linux command-line tools such as make, grep, and find

Course materials

All students will receive:

- Electronic copies presentations and lab notes
- Worked solutions to the problems

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

Embedded Android

Course outline

Introduction to Android

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

Creating a new Android product

- The structure of an Android device
- Setting basic hardware characteristics
- Adding the device to the “lunch” menu

The kernel

- Looking at some Android-specific changes to Linux
- The Google common kernel and vendor kernels
- Building Linux from source

Booting Android

- Options for storage layout: separate system and vendor partitions; system-as-root; A-B “seamless update”
- Flashing images using fastboot
- Android init: events and services
- Understanding run command (.rc) scripts

The Android build system

- Building a product from packages and modules
- Creating packages and modules using Android Blueprint (Android.bp) and Android Makefiles (Android.mk)

The Android framework

- Understanding Android services
- The role of binder and AIDL
- Adding a custom system service inside a persistent app
- Accessing the service from a platform library

Applications

- Application sandboxing
- Writing system and privileged apps
- Calling platform libraries

Security

- How Android permissions work and how to add your own
- POSIX user ID and group ID within Android
- Understanding and modifying SELinux policy files
- Security threats and “rooting”

The Hardware Abstraction layer

- Types of HAL: binderized; passthrough and same process (sp)
- HIDL – the HAL Interface Definition language
- Run-time linking and the Vendor Native Development Kit (VNDK)
- The Vendor Interface (VINTF)

Testing: CTS and VTS

- Validating your Android implementation using the Compatibility Test Suite, CTS
- Testing your vendor HAL using the Vendor Test Suite, VTS

Integrating non-Android code

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

Graphics

- How SurfaceFlinger orchestrates image rendering
- Internal and external displays

Debugging and profiling

- Debugging native code with GDB
- Decoding crash dumps (Tombstones)
- Profiling CPU usage using perf