Formation Android AOSP

Durée : 3 jours - 21 heures

Prérequis : The skills required for the internship are the following:

UNIX/Linux knowledge (using the shell !) - Some Java knowledge - Some C language knowledge

Publics concernés : Android applications developers, Linux developers, Linux projects managers

Matériel :

The following table shows the PC requirements (type of CPU, RAM, free space on the hard disk).

CPU : i5 or better (i7 preferred)

RAM : 16Gb or more (32Gb preferred)

Disk : 500 Gb or more

Inscription en inter entreprise : 2 000 € HT/pers - (Les repas ne sont pas compris)

Objectifs :

- Describing how the Android OS works (AOSP)
- Explaining Treble and HIDL projects
- Explaining development methods for the system's low level layers (NDK, C/C++)
- Describing AOSP customization for a new platform
- Introducing Android security
- Introducing Android Automotive OS (AAOS)
- Introducing certification procedures (CDD/*TS)

Déroulé :

A practical work is available for each item with a "star" (*).

Introduction to Android

- History
- Market share
- Android "fragmentation"
- Android vs Linux
- Android for the industry (pros and cons)

Using AOSP

- What is AOSP ?
- Licensing
- The "repo" tool
- Downloading the sources *
- Compiling an AOSP image (for the emulator) *
- Describing source content
- Customizing / defining a target
- Adding BusyBox *
- Updating the boot animation *
- Adding an application to the AOSP build *
- Using a real board (fastboot)

The Android kernel

- History and introduction
- Linux kernel cross-compilation
- Linux kernel “androidism”
- Compiling the kernel for Android with GKI (Generic Kernel Image) *
- Using a real board (BSP)

Android architecture

- Android layers
    - The Android framework
    - Native libraries
    - HAL (Hardware Abstraction Layer)
    - The Linux kernel
- Binder
- System server
- Booting Android
- Android graphic system
- Focus on Treble / HIDL *
- Focus on Android Auto Automotive OS (AAOS) *
- Testing the Vehicle HAL *

Development

- Android SDK
- ADB *
- User space development with the NDK (C/C++) *
- Remote debugging *
- JNI (Java Native Interface with C/C++) *
- Example of hardware device access (Java → kernel) *

Security

- UID/GID
- Capabilities
- Permissions
- Verity (dm-verity)
- Secure boot (AVB)
- Encryption (dm-crypt)
- SELinux *
- Seccomp
- OTA

Certification

- CDD (Compatibility Definition Document)
- CTS (Compatibility Test Suite)
- VTS (Vendor Test Suite)