G3 - Les entrailles d'Android

Les frameworks Android et la mise en œuvre de la HAL

Objectifs

- Explorer l'architecture du code source d'Android
  - Le processus init d'Android
  - Les services système
  - Le Binder d'Android
  - La sécurité d'Android et SELinux
  - Le framework applicatif d'Android
  - La couche d'abstraction matérielle d'Android
  - Le framework multimédia d'Android et OpenMAX
- Comprendre la structure statique et dynamique des frameworks
  - Structure des classes
  - Répartition entre code Java et code C++
  - Flux de données à travers les frameworks
  - Structure de contrôle des frameworks

À qui s'adresse ce cours ?

- Aux ingénieurs qui doivent travailler sur le portage d'Android sur une nouvelle plateforme
  - Écrire la HAL d'une nouvelle carte
  - Déboguer la HAL et les frameworks Android
  - Étendre la plateforme Android pour des usages spécifiques

Prérequis

  - Bonne expérience du kernel Linux et de la programmation de drivers
  - Connaissance de l'installation d'Android
  - Connaissance de base de la structure d'une application Android
  - Bonne maîtrise de la programmation C++ et Java
  - Connaissance de base de Java

Environnement du cours

  - Support de cours imprimé (en anglais).
  - Un PC Linux pour deux stagiaires (64 Go de RAM, 1 To de disque libre, 8 cœurs).
  - Débogage sur émulateurs Android (Goldfish ou Cuttlefish)

Audience visée

- Tout ingénieur ou technicien en systèmes embarqués possédant les prérequis ci-dessus.

Plan du cours

Présentation de l'architecture d'Android

- Linux et Android
- Les licences d'Android

Premier jour

Les extensions Android du kernel Linux

- Accéder au code source du kernel
- Les drivers du kernel spécifiques à Android
  - Ashmem
  - Logger
  - Low_memory_killer
  - Timed_output
  - Timed_gpio
  - Gestion des boutons et du clavier
- Les kernels GKI d'Android
- Le débogueur du kernel Android

Le système de build d'Android

- La base de code d'Android
- Construire Android
  - L'environnement de build d'Android
  - Les émulateurs Android (Goldfish et Cuttlefish)
  - Le système de build d'Android
  - Les fichiers Android.mk
- Ajouter de nouveaux composants au système de build
  - Composants Java
  - Composants natifs
  - Applications
- Créer une nouvelle plateforme Android
- Migrer vers Soong
  - Les fichiers Blueprint
- Android Treble
  - Les partitions Android
  - L'image système partagée d'Android (SSI)
  - Le contrôle des interfaces
- Débogage

L'initialisation du système Android

- Les propriétés Android
- L'initialisation d'Android
  - Initialisation d'Android et gestion des périphériques
  - Le langage d'initialisation d'Android
  - La séquence de démarrage d'Android
  - Le processus “zygote”
- Security Enhanced Android
  - La sécurité dans Android et SELinux
  - Les commandes de la toolbox

    - La sécurité et le système de build
    - Les fichiers SEPolicy

L'interface native d'Android

- La bibliothèque C bionic
    - Pourquoi une nouvelle bibliothèque C
    - Les fonctionnalités de bionic spécifiques à Android
    - Ce qui manque dans bionic
- Ajouter des composants natifs
    - Ajouter des exécutables natifs
    - Définir des méthodes Java en C++
    - JNI pour Android
- Interface de plateforme pour les composants natifs
    - Accéder aux propriétés système
    - Accéder au système de log d'Android
    - Interagir avec les services démons
- Le NDK Android et JNI
- SWIG

Deuxième jour

Le Binder d'Android

- L'architecture Binder d'Android
    - Pourquoi un nouveau mécanisme d'IPC
    - Android comme système massivement réparti
    - Le Binder en action
    - Le driver kernel du Binder
- Mise en œuvre de Binder
    - Le langage AIDL
    - L'outil AIDL
    - Les classes Java de Binder
    - Les classes C++ d'implémentation de Binder
- Écrire des services
    - Services Java standard
    - Les services et le Binder
    - Liaison de service
    - Stable AIDL
- Mise en œuvre de Binder
    - Les classes Java de Binder
    - Le comptage de références dans les frameworks C++ d'Android
    - Les classes C++ d'implémentation de Binder
    - Implémenter un service C++
    - Le backend NDK d'AIDL
- Les services système
    - Qu'est-ce qu'un service système
    - Services statiques et services dépendant du contexte
    - Structure d'un service système
    - Ajouter un nouveau service système
    - Le processus système ServiceManager

Troisième jour

Le Power Manager d'Android

- La gestion de l'énergie dans Android
- L'API driver
- L'API en mode utilisateur
- L'API Java

La couche d'abstraction matérielle

- Pourquoi une HAL ?
- HAL conventionnelles et HAL historiques
  - Architecture
  - Les principaux composants HAL
  - La HAL Sensor historique
- Les HAL HIDL
  - Types de HAL
  - Le Hardware Interface Definition Language
- Migrer de HIDL vers AIDL

Quatrième jour

Le framework de capteurs d'Android

- Les capteurs dans Android
  - Les types de capteurs
  - Le Sensor Manager
  - Accéder aux capteurs
- Architecture du framework
  - Découverte des capteurs
  - Calibration des capteurs
- Le Location Manager et le géocodage

Le framework multimédia d'Android

- Le multimédia dans un appareil Android
  - Formats de données et formats de fichiers
  - Codec et démultiplexage
- Le multimédia pour les applications
  - Lecture audio et vidéo (classe MediaPlayer)
  - Capture audio et vidéo (classe MediaRecorder)
- Architecture du framework
  - Architecture générale du framework
  - Flux généraux de données et de contrôle
  - La couche de service MediaPlayer
  - Stagefright et OpenMAX

Stagefright et OpenMAX pour le multimédia Android

- Le multimédia dans un appareil Android
- StageFright dans le framework média d'Android
  - L'ancien framework OpenCore
  - La structure des classes de Stagefright
- Présentation d'OpenMAX
  - Le Khronos Group

    - OpenMAX/DL : la couche de développement
    - OpenMAX/IL : la couche d'intégration
    - OpenMAX/AL : la couche applicative
    - OpenMAX et OpenSL/ES
- OpenMAX dans le framework média d'Android
    - Interface entre Android et OpenMAX
    - L'architecture d'OpenMAX/IL
- Intégration d'OpenMAX/IL dans Stagefright
    - Enregistrement des composants
    - Configuration des composants
    - Les particularités des composants
- L'implémentation Bellagio d'OpenMAX/IL
    - Implémentation LGPL du cœur d'OpenMAX/IL
    - Exemple d'implémentation de composants

Cinquième jour

Le Surface Flinger

- Présentation
- Le Hardware composer

L'Audio Manager d'Android

- Le routage audio
- Architecture de l'Audio manager
    - Le système audio
    - L'Audio policy manager
    - L'Audio policy service
- Le flux de contrôle
    - Contrôle de la lecture et de l'enregistrement
    - Génération de la source de temps

L'Audio Flinger

- Flux audio en sortie et en entrée
    - Gestion des tampons
- L'Audio track
    - Présentation
    - Cycle de vie d'une track
    - Flux de données
- Le mixeur audio :
    - Présentation
    - Cycle de vie du mixeur
    - Fast mixer
    - Rééchantillonnage
    - Volume

Les composants audio de la HAL

- Les politiques audio
    - Politiques audio et périphériques de politique
    - Services fournis par la politique audio de la HAL
    - Utilisation depuis l'audio policy manager
    - Utilisation des services audio par la politique audio de la HAL
    - Duplication de sortie
    - Suspend/resume

- Les périphériques audio
  - Classes de périphériques audio
  - Flux de données
  - Interaction avec l'audio track et l'enregistrement
- Flux et effets audio