
Яндекс

Яндекс

Погружение в Android Multimedia

Mobius 2016

Основы

Android Audio/Video playback

* MediaPlayer
* AudioTrack
* MediaCodec
* MediaExtractor

OpenMAX

* OpenMAX DL (development layer)

Набор примитивов и интерфейсов по реализации кодеков вендорами

* OpenMAX IL (integration layer)

Набор интерфейсов по низкоуровневой работе с кодеками (OMXCodec.cpp)

* OpenMAX AL (application layer)

Набор высокоуровневых абстракций по работе с OpenMAX IL =)

OpenMAX IL (integration layer)

* OMXCodec.cpp
  (libstagefright.so)

доступен с Android 2.3

* MediaCodec.cpp ((libstagefright.so)

* frameworks/base/media/jni/
  android_media_MediaCodec.cpp (libmedia.so)

* MediaCodec.java доступен с Android 4.1

OpenMAX AL (application layer)14 api

* Поставляется вендором (libOpenMAXAL.so)

```
#include <OMXAL/OpenMAXAL.h>
#include <OMXAL/OpenMAXAL_Android.h>
```

* И все это богатство появилось с Android 4.0

* Если не хотите мучаться с MediaCodec, а высокопроизводительный плеер писать надо, то вам сразу сюда

media/NdkMediaCodec.h 21 api

* Добавлен API для работы с MediaCodec из C++ без OpenMAX AL

```
#include <media/NdkMediaCodec.h>
#include <media/NdkMediaExtractor.h>
```

* Связано это с тем, что в Android N теперь нельзя линковаться с системными либами

* А к какой либе надо было залинковаться, чтобы получить доступ к API который использует MediaCodec.java?

Подготовка h264 фрейма для

* Существует два формата формирования фреймов в h264: Annex B и AVC
* Изначально был только AVC, это формат фрейма использующийся в MPEG-4 AVC
* Но для потокового видео AVC не годится и был изобретено расширение Annex B
* Android “кушает” только AVC
* Соответственно, мы должны конвертировать фреймы из AnnexB в AVC

Вывод видео

Проблема частоты обновления экрана

* VSync - это синхронизация кадровой частоты

* Стандартная частота VSync у Android какая?

* Т.е. каждые 16мс происходит отрисовка нового кадра

* И на начало нового такта VSync у нас уже должен быть заготовлен кад
иначе нам надо будет ждать +16мс

* Для 50Hz надо использовать Three-two pull down

ANativeWindo android/native_window.h

android 2.3

* Единственно верный способ работать с окном из NDK

* Можно получить ANativeWindow через android/native_window_jni.h

```
ANativeWindow* ANativeWindow_fromSurface(JNIEnv* env, jobject surface);
```
android 2.3

```
ANativeWindow* ANativeWindow_fromSurfaceTexture(JNIEnv* env, jobject surfaceTexture);
```
android 4.0

* Содержит аллоцированные буферы на GPU, при аппаратном декодировании фрейм сразу попадает в эти буферы

Вывод аудио

AudioTrack

* Высокоуровневый интерфейс по работе с audio на

* Для нетребовательных приложений к задержке вывода сигнала

* Умеет играть файл целиком (MODE_STATIC) или по кусочкам (MODE_STREAM)

OpenSLES

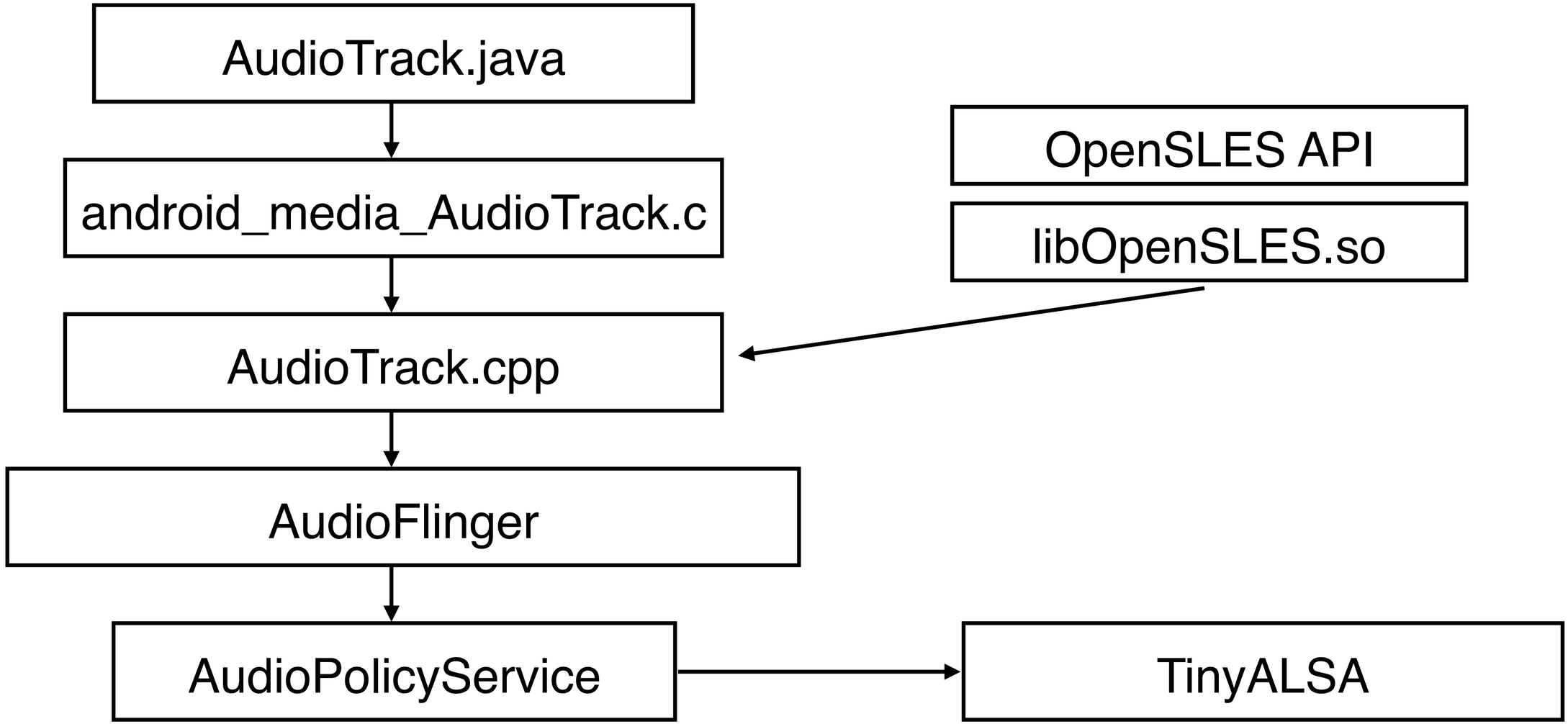
android 2.3

* Интерфейс предоставляемый Khronos по работе со звуком

* Это более низкоуровневый и гибкий интерфейс

* Если вам нужно писать какую-то сложную работу со звуком, например сведение треков вместе и наложение эффектов - то вам сразу сюда

AudioTrack vs OpenSLES



Какие библиотеки выбрать?

FFmpeg (LGPL)

https://github.com/FFmpeg/FFmpeg

https://github.com/WritingMinds/ffmpeg-android

* Комбайн по работе с мультимедиа

* Содержит реализации h264/h265/vp8/vp9 декодеров

* Содержит свой ресэмплер для звука

VLC (LGPL) (libvlc)

https://github.com/videolan/vlc

https://github.com/mrmaffen/vlc-android-sdk

* Это не только видео плеер, но и библиотека - libvlc

* Имеет свои уровни абстракции

* Имеет свои распаковщики для mp4

* Использует ffmpeg

GStreamer (LGPL)

https://cgit.freedesktop.org/gstreamer/

* Большой мультимедийный фреймворк конкурирующий с libvlc

* Куча своих наворотов

* Так же есть куча своих парсеров форматов

* И так же использует ffmpeg

ExoPlayer (Apache License)

https://github.com/google/ExoPlayer

* Плеер-фреймворк от Google написанный на Java

* Использует MediaCodec

* Адаптирован для проигрывания потоковых форматов видео (HLS/DASH/mp4/mpeg2ts)

* Используется в YouTube, Google Play Movies

* Основные проблемы это JNI overhead

ijkplayer (Apache

https://github.com/Bilibili/ijkplayer

* Работает на базе ffmpeg

* Есть MediaPlayer API Java-wrapper

* Есть интеграция с ExoPlayer (т.е. exoplayer начнет уметь играть то, что умеет играть FFmpeg)

* Внутри, по умолчанию, использует OpenMAX AL интерфейс предоставляемый в Android NDK

Выводы

* Работа с видео и аудио в Android многогранна и таит в себе множество нюансов

* Если планируется что-то высокопроизводительное писать, то сразу NDK

* Возможно вам хватит OpenMAXAL и OpenSLES

* Если minTargetSdk=21, то посмотрите на NdkMediaCodec

* Для быстрого бутстрапа рекомендую *ijkplayer*

Ссылки

1. https://source.android.com/devices/media/
2. https://source.android.com/devices/audio/index.html
3. http://mobilepearls.com/labs/native-android-api/ndk/docs/openmaxal/
4. https://developer.android.com/ndk/guides/audio/index.html
5. https://vec.io/posts/how-to-build-ffmpeg-with-android-ndk
6. https://wiki.videolan.org/AndroidCompile/
7. http://docs.gstreamer.com/display/GstSDK/Android+tutorials
8. https://developer.android.com/guide/topics/media/exoplayer.html
9. http://android-developers.blogspot.ru/2016/05/hardening-media-stack.html

Контакты

Дмитрий Полищук

Разработчик

 @dpolishuk

 deepol@yandex-