

將 FFmpeg 移植到 Android

原文連結 (CSDN)

緣起

對於尚未了解 FFmpeg 的同學，可以先去看看 FFmpeg 常用基本命令，仔細研究其功能——如音視頻合成、播放各類編碼、視頻截取、多圖合成視頻與音頻混流、格式轉換等。它能為很多應用場景帶來強大且有趣的功能。

以「配音秀」類應用為例，配音不僅趣味十足，也往往能讓產品更具吸引力。我們計劃開發一個配音模塊，進而看中了 FFmpeg，嘗試將其移植到 Android 平台。前後花了 3~4 天，嘗試了多個版本，網上不少文章都沒能成功，直到發現一篇名為《android 使用 ffmpeg 並調用接口》的教程，才最終搞定。實測表明，只有 FFmpeg 1.2 搭配 ndk-r9 的組合才能成功移植。

思路

FFmpeg 是一個用 C 語言編寫的項目，其中包含一個 `main()` 函數。我們的目標是：

1. 使用 Android NDK 編譯出 `libffmpeg.so`。
2. 借助該庫編譯並修改 `ffmpeg.c` 文件，將原本的 `main()` 函數改名為 `video_merge(int argc, char **argv)`，這樣就能從 JNI 裡直接調用它來完成視頻合成等操作。

例如，可通過類似以下方式來實現視頻合成（對應命令行 `ffmpeg -i src1 -i src2 -y output`）：

```
video_merge(5, argv); // 其中 argv 模擬命令行參數
```

環境

- 操作系統：Ubuntu 12.04
- FFmpeg 版本：1.2

- NDK 版本：ndk-r9

在動手之前，建議先參考一些相關教程，碰到問題再回來比照本文，避免走太多彎路。

修改接口與 Android.mk

在為 FFmpeg 編寫 JNI 接口時，需要編寫 `Android.mk` 文件來鏈接庫，進而生成可用的 `.so` 文件。一些示例的 `Android.mk` 在不同環境下可能無法直接運行成功。其作用是告訴 NDK 哪些源文件需要編譯、鏈接到哪個庫等信息。

我採用了一種「兩次編譯、再鏈接」的方法：

1. 先編譯得到一個名為 `myffmpeg` 的共享庫。
2. 在另一個模塊 `ffmpeg-jni` 中，再將 `myffmpeg` 鏈接進去，最終生成所需的 `.so`。

另外，需要將編譯好的 `libffmpeg.so` 放置在 `jni` 目錄下，以保證鏈接時能夠找到它。

調試 FFmpeg

移植 FFmpeg 之後，為了通過 JNI 調用函數，往往需要在 C 層調試。如果能像在命令行一樣看到詳細的日誌輸出，就能更輕鬆地定位問題。

在 Eclipse 下，按住 Ctrl 點擊類似 `av_log` 的調用位置，可以追蹤到 `ffmpeg/libavutil/log.c` 裡 `av_log_default_callback` 函數的實現。它會調用 Android 的 `__android_log_print` 打印到 Logcat。通過查看這些輸出，就能獲知 FFmpeg 的內部狀態，用來排查合成失敗、不支持特定編碼等問題。

有時，FFmpeg 會拋出異常導致 App 崩潰，可以使用以下命令來定位：

```
adb shell logcat | ndk-stack -sym obj/local/armeabi
```

如果 FFmpeg 原始的 `main()` 最後有一個 `exit(0)`，請記得註釋掉，否則會導致應用退出。

內存洩漏與 Service 方案

合成完成後，若再次調用時出現 "INVALID HEAP ADDRESS IN dlfree ffmpeg" 錯誤，多半是 FFmpeg 內存釋放不完全導致。一個折中辦法是將合成過程放在一個單獨的 Service 裡，合成完畢後殺掉該 Service，以清理資源。

```
<!-- AndroidManifest.xml -->  
<service android:name=".FFmpegService" />
```

通過註冊 Receiver 等方式，在合成完成後自行結束 Service，即可避免重複調用時的內存問題。

可能遇到的問題

- **AAC 文件播放異常**

有些機型（如小米 2s）可能無法通過默認的 MediaPlayer 播放 AAC 編碼的音頻。

- **編碼器支持不足**

如果要支持 AMR-NB、MP3 等，需要在編譯 FFmpeg 時手動啟用相應的選項。如果編譯腳本無法找到相關庫或頭文件，會報錯終止。

- **合成速度**

合成一個 10 秒的 1280×720 視頻並混合音頻，可能要花費幾十秒到一分鐘不等。用戶體驗上，或許先讓用戶試聽再決定是否最終合成會更好。

在配音秀的具體實現中，常見的做法是：1. 提前下載好原始視頻、字幕，以及「已去除需配音片段」的音頻文件。2. 錄製用戶聲音，合成時僅需將錄音與靜音片段合併即可。3. 若對本地合成時間不滿意，可選擇將音視頻數據上傳到服務器，由服務器端進行合成，完成後再下載。

在 Eclipse 中使用 NDK

不一定要在命令行輸入 `ndk-build`。只需在 Eclipse 中右鍵項目，選擇 **Android Tools → Add Native Support**，之後每次點擊「Run」都會自動執行 `ndk-build`。

一鍵生成 JNI 頭文件

編寫 JNI 函數頭文件比較繁瑣，可以通過 javah 命令自動生成。

在 Eclipse 裡，可以配置為一個外部工具，用類似如下命令來生成頭文件：

```
javah -jni -classpath bin/classes -d jni com.example.ffmpeg.MyFFmpeg
```

執行後，會在 jni 目錄下產生類似 com_example_ffmpeg_MyFFmpeg.h 的文件，然後只需在你的 C 代碼中 #include 並實現對應函數即可。

小結

FFmpeg 在 Android 上的移植涉及多方知識，包括 NDK 環境配置、C/C++ 編譯鏈接、JNI 調用、音視頻編解碼等。若遇到無法合成、不支持某些格式或鏈接報錯等問題，需要仔細排查配置與日誌輸出。希望本文能夠幫助你避開一些坑。如果你也在使用 FFmpeg，歡迎在評論區分享經驗或問題，互相交流與學習。