将 FFmpeg 移植到 Android

https://blog.csdn.net/lzw_java/article/details/21564091

缘起

对于那些尚未了解 ffmpeg 的同学，请移步了解 FFmpeg 常用基本命令，仔细研究一番，看看它的功能是否能够为你的应用带来更多乐趣和强大功能。

它具备合成音视频、播放各种编码的音视频、截取特定片段、将多张图片合成视频并与音频合并、格式转换等诸多功能。

在类似配音秀的应用中，配音是一种非常有趣的体验，我们也有意开发一个配音模块。因此，我们找到了 ffmpeg 这个工具，试图将其移植到 Android 平台。大概花了我 3 到 4 天的时间，经历了多个版本的尝试，网上很多的文章都没有帮助，直到我找到了那篇名为《android 使用 ffmpeg 并调用接口》的文章，才最终获得了成功。我尝试了很多版本，只有 ffmpeg 1.2 和 ndkr9 的组合才能成功移植。

思路

ffmpeg 是一个用 C 语言编写的程序，其中包含有一个 main 函数。我们首先在 Android NDK 下编译出 libffmpeg.so 文件，然后利用这个库来编译 ffmpeg.c 文件。我们将 ffmpeg.c 文件中的 main 函数修改为 video_merge 函数，即 video_merge(int argc,char **argv)。接下来，就可以像下面这样合成视频：

这个操作类似在命令行下调用：ffmpeg -i src1 -i src2 -y output

环境

我们使用的操作系统是 Ubuntu 12.04。

在开始之前，先查看一下相关的教程，如果遇到问题再来参考这篇文章，介绍在那篇教程下可能会遇到的问题。

修改接口调用

在修改接口调用的时候，那篇文章使用了这个 Android.mk 文件：

然而，我一直无法成功运行这个文件。

这个文件用于链接库，也就是为当前模块进行编译。它会在一个目录中寻找所需的函数等内容，然后链接到 libffmpeg.so 文件上。这里的`-l` 表示链接的意思，而 `ffmpeg` 表示库的名称。至于 `lib` 和`.so` 这两部分是系统自动加上的。

因此，我采用了这个方法：

请注意，这里首先编译了一个共享库 `myffmpeg`，然后通过链接到最终需要的 `ffmpeg-jni`。

在这个过程中，需要将 libffmpeg.so 文件放置在 jni 目录下，像这样：

调试 ffmpeg

移植 ffmpeg 后，我们也使用了 JNI 来调用函数，但可能会遇到许多问题。那么如何进行调试呢?

如果我们能像在命令行下一样获得调试信息就好了：

这个日志可以帮助我们找到相应的位置，在附近添加一个 Log.i 语句即可。

在 Eclipse 下，你可以通过按住 Ctrl 键并点击类似于 `av_log` 的位置，从原始的 main 函数入口处跳转到函数位置，逐步跟踪。它位于 `ffmpeg/libavutil/log.c` 文件中的 `av_log_default_callback` 函数内，就像这样：

请注意到了 `LOGD` 吗，它是：

这里的... 表示可变长参数。可变长参数的最简单例子就是 printf 函数，有时候我们可以这样写 `printf("%d", a)`，也可以这样写 `printf("%d %d", a, a)`，前者有两个参数，后者有三个参数。使用类似的语法实现。

而 `__android_log_print` 则是 Android 提供的用于打印到 Logcat 中的函数。

通过这种方式，你就可以查看输出日志，帮助解决无法合成、不支持 mp3 或 amr、或者视频问题等等。

有时候可能会突然出现异常。

如何知道错误发生的位置呢?

在命令行中输入：adb shell logcat | ndk-stack -sym obj/local/armeabi

你会发现合成虽然成功了，但应用程序仍会崩溃退出，因为 ffmpeg 原始的 main 函数最后有一个 exit 语句，只需注释掉即可。

成功合成后，你可能会发现再次调用时会出现”INVALID HEAP ADDRESS IN dlfree ffmpeg”的错误。这是由于 ffmpeg 内存泄漏，一些动态分配的空间没有被释放。因此，我们可以将 ffmpeg 的合成放在一个 service 中，合成完成后再杀掉该 service。

AnroidMainfest.xml：

在你的程序中注册一个 receiver：

这样，你就可以第一次调用 ffmpeg 生成一个没有声音的视频文件，播放它，让用户进行配音，然后再次调用，将用户的声音与视频合成。

在我的测试中，我发现，如果使用 AAC 编码器录制，虽然可以合成视频，但无法使用 MediaPlayer 播放 AAC 文件（这可能是小米 2s 独有的问题?)。而如果使用 ARMNB，则无法合成，还需要重新生成 libffmpeg.so，并在 configure 中启用 ARMNB。然而，在启用了 ARMNB 后，运行`./config.sh`编译 ARMNB 时会出错，提示找不到相应的文件。找到了文件后又提示找不到所包含的文件。同样地，启用 libmp3lame 也没有成功。

之所以希望能够进行播放，是因为合成一个 10 秒的 1280*720 视频和 10 秒的录音大约需要 1 分钟的时间。希望用户在决定是否合成之前可以先听听配音的效果。

观察配音秀的 dubbing 文件夹，你会发现配音秀录制了用户的声音，得到了 tmp.amr 和 tmp.pcm 文件，然后上传到服务器进行合成，并将合成的文件下载下来。

在配音时，原始视频、字幕和已经去除了需要配音片段的音频已经下载好了。因此，在配音时，只需播放这个音频，其中需要配音的部分是静音的。

在 Eclipse 中使用 ndk-build

实际上，我们不必在命令行中进行 `ndk build`。只需选择项目，右键选择”Android tools”，然后选择”Add native Support”即可。这样，每次在项目中运行时，都会自动进行 `ndk build`。

一键生成 JNI 函数头文件

每次手动编写这些头文件都很麻烦，其实我们可以通过一键生成来简化这个过程:

配置一个外部工具:

使用 `javah` 命令生成，并将其集成到 Eclipse 中。

在 Eclipse 中点击标题栏后，你就可以在 jni 目录中看到类似于 `com_lzw_iword_video_Myffmpeg.h` 的文件，就像这样:

然后，将这个头文件包含到你的 C 文件中，复制粘贴函数头即可。

如果你也在将 ffmpeg 移植到 Android 平台，或者在使用 ffmpeg 时遇到了问题，欢迎在评论中提问，进行交流讨论～