# Bilibili开源项目ijkPlayer编译及使用注意

**一:关于ijkplayer基本介绍**

ijkPlayer用于实现在线视频直播以及本地视频多种格式的播放，本地视频音频包含主流的格式，而主要功能在线直播部分封装了rtmp，hls协议用于直播视频流，开源项目地址为：*<https://github.com/Bilibili/ijkplayer>*

**二：使用背景**

本文档创建基于在线课堂项目，叙述内容为android端如何编译FFmpeg与ijkPlayer结合作为so架包并用于视频直播项目。

**三：前期准备**

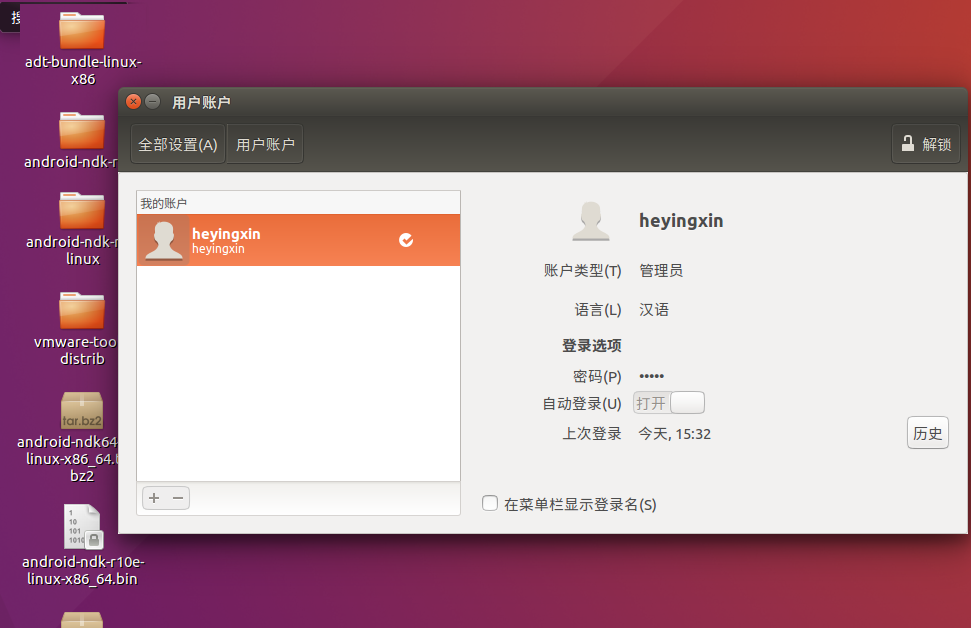
1：开发工具为 android studio 1.5

2：PC为 mac os10或者Ubuntu（mac没有尝试，用的Ubuntu乌班图虚拟机，还有一种使用windows的cygwin但是不建议使用）

**四：详细步骤**

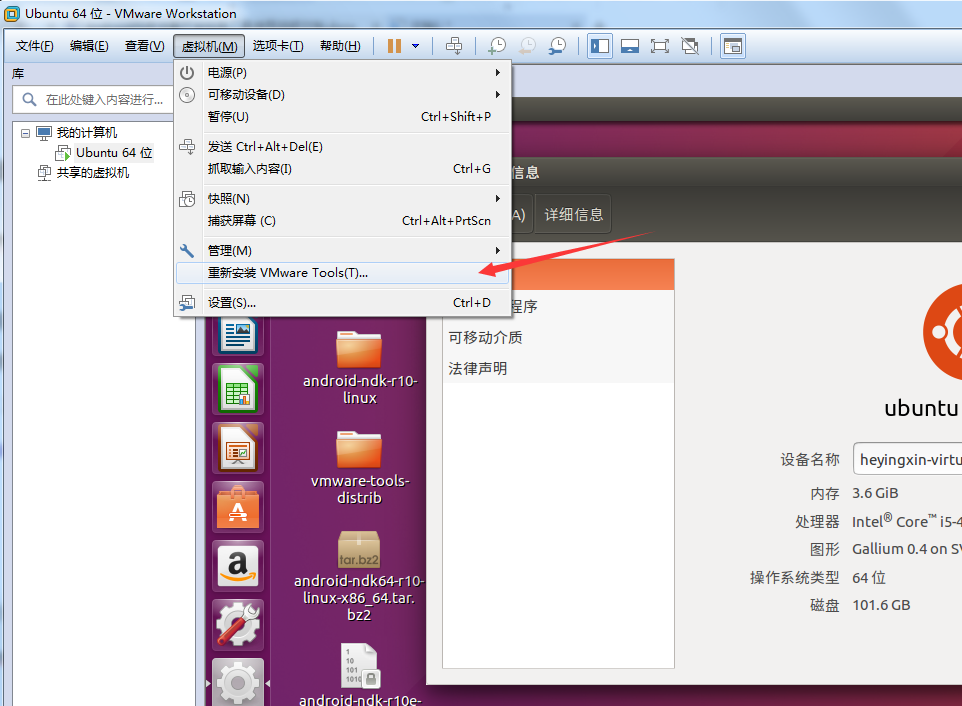
1：下载[VMware Workstation](http://www.baidu.com/link?url=0TwmVrpXygAshL0nnAd5ui5YY0qw-EhYDzRpfmi7rLtHdwjGdyewH6_A3Ca7Eisg5SdPO_EhIxgZppAltEtteq&wd=&eqid=c877ae940001106f0000000357679c6b" \t "http://www.baidu.com/_blank)虚拟机，并安装乌班图系统（镜像下载地址可以使用http://192.168.1.119/iso 注意下deskTop可视化界面的），没有就自己通过其他渠道下载了，下载完成后安装系统并注册系统User。



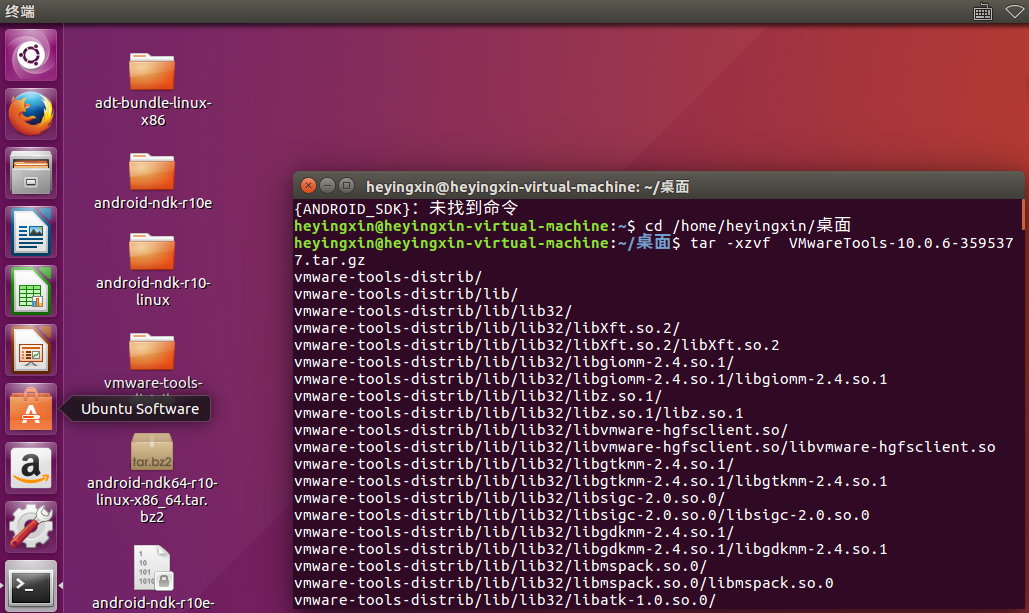


我这里使用的是乌班图16.04 64位；

2：安装工具方便windows与Ubuntu交互。



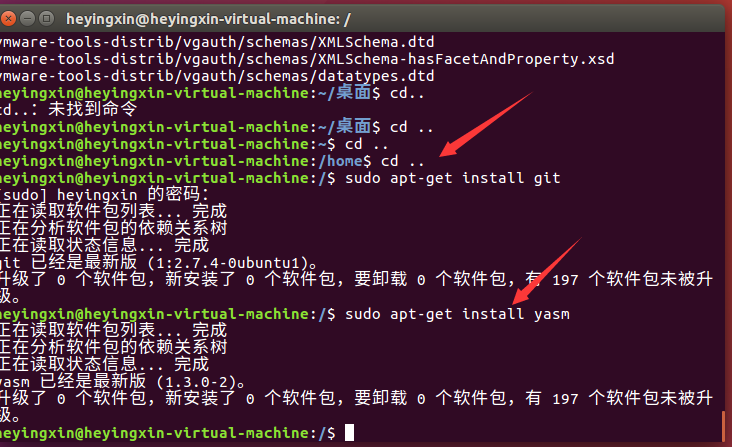
下载完成后会在Ubuntu下生成一个VMwareTools压缩包，右键桌面打开终端通过cd命令到压缩包的目录下：



输入命令：tar -xzvf  VMwareTools-10.0.6-3595377.tar.gz

下面会跑一堆日志即安装完成，完成之后即可在win与Ubuntu直接互相拷贝数据等。

3：安装git与yasm工具

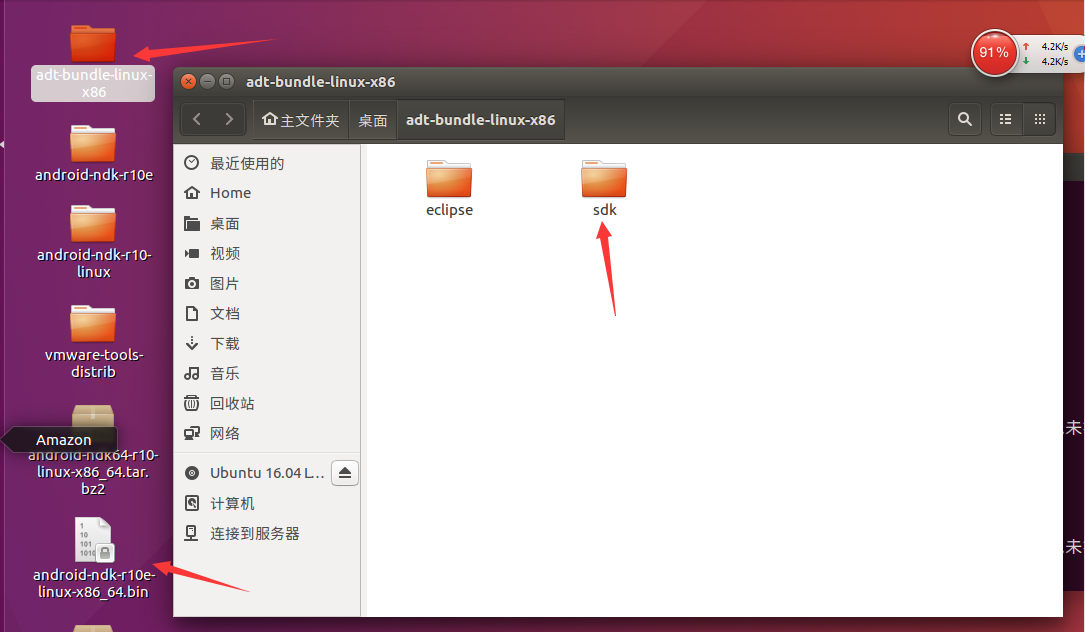


Cd到跟目录直接执行管理员命令安装git与yasm，命令分别是：

sudo apt-get install git

sudo apt-get install yasm

4：拷贝sdk与ndk到乌班图桌面，直接从windows下拷



这里注意两点

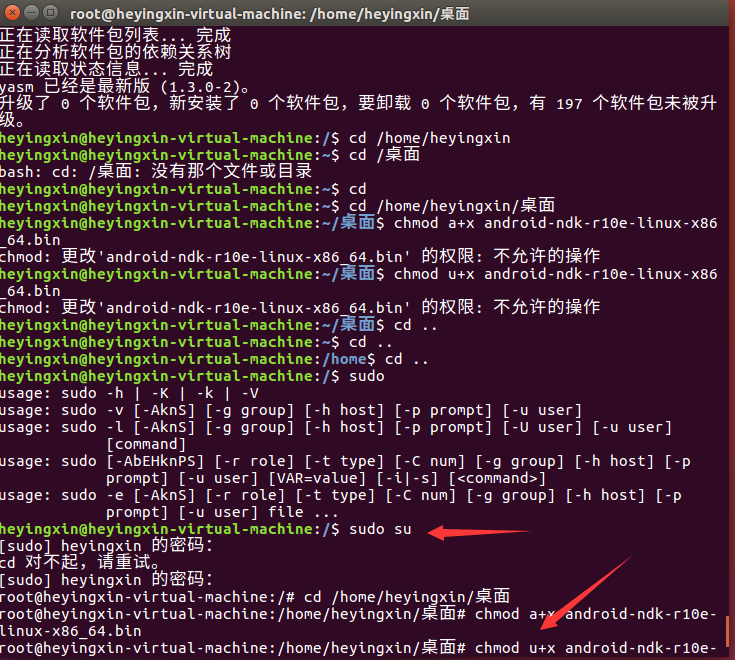
1：必须是linux版的，sdk与ndk都是。

2：ndk是一个bin包，sdk是文件，sdk很大完全版的设置36个G，这里我的sdk只保留了api18，并没有拷贝所有的否则很耽误时间。

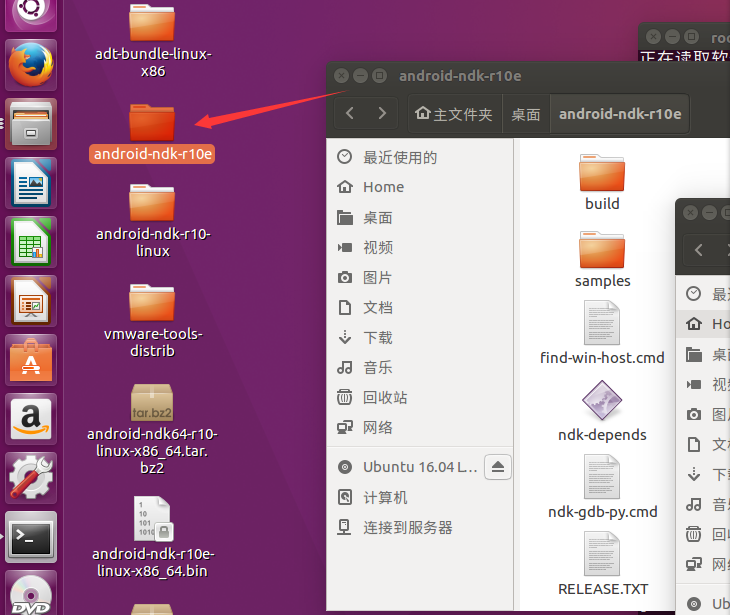
1. ndk下载地址：<http://wear.techbrood.com/tools/sdk/ndk/>

这里我下载的是linux 64位 re10版本

1. sdk下载地址：<http://dl.google.com/android/adt/adt-bundle-linux-x86_64-20131030.zip>
2. Ndk安装注意下载下来的bin文件无法直接打开也是命令安装



首先调用 sudo su进入root，否则会报上面的操作不允许，然后进入ndk的下载目录，进入命令chmod a+x （后面是ndk地址），下面会跑一截日志，会在桌面生成ndk的目录。



5：对sdk与ndk进行配置，与windows的环境变量一样，乌班图需要配置自己的环境变量，进入文档Home，按ctrl+h显示影藏文件，这时会出现一个名为.bashrc的文件，我们需要修改这个文件，加上配置，代码如下：

ANDROID\_NDK=/home/heyingxin/桌面/android-ndk-r10e

export ANDROID\_NDK

ANDROID\_SDK=/home/heyingxin/桌面/adt-bundle-linux-x86/sdk

export ANDROID\_SDK

NDK=/home/heyingxin/桌面/android-ndk-r10e

export NDK

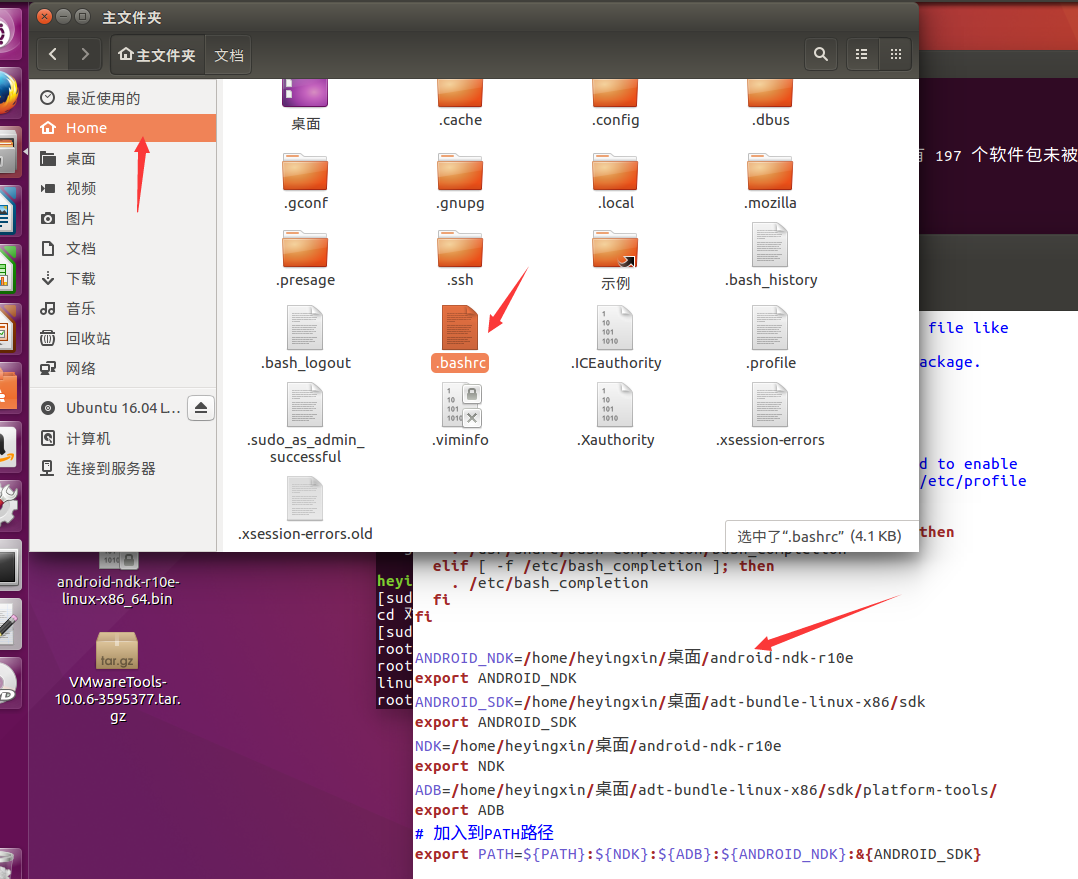
ADB=/home/heyingxin/桌面/adt-bundle-linux-x86/sdk/platform-tools/

export ADB

# 加入到PATH路径

export PATH=${PATH}:${NDK}:${ADB}:${ANDROID\_NDK}:&{ANDROID\_SDK}

\*路径名与文件名以自己的为准，修改完保存并重启即可。



开始编译步骤：

1：进入github地址下拉找到android build命令（由于版本更新，build会出现变化，最好注意看下githug的步骤，我的也是以他的为准，这里额外加上步骤意义）

# 直接拉取源代码到本地

1：git clone https://github.com/Bilibili/ijkplayer.git ijkplayer-android

2：cd ijkplayer-android

# 检查更新代码

3：git checkout -B latest k0.5.1

# 初始化，会把ffmpeg的代码拉取到本地等等操作

4：./init-android.sh

5：cd android/contrib

6：./compile-ffmpeg.sh clean

# 编译ffmpeg软解码库

7：./compile-ffmpeg.sh all

8：cd ..

# 会生成各种版本的so文件

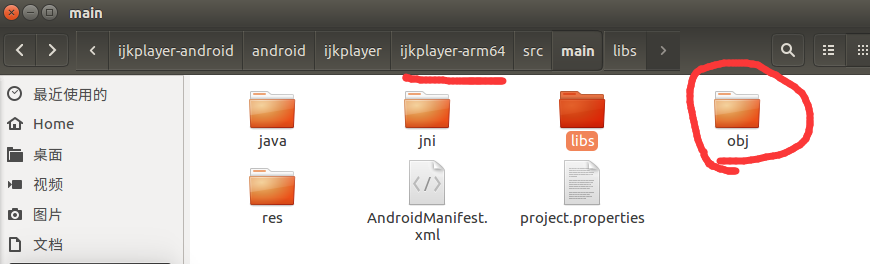
9：./compile-ijk.sh all

根据我的编译经验，第1步按照你的网速时间可能比较长，第7步编译Ffmpeg根据你机器速度一般时间也比较长（甚至很长）

而我所遇到的问题只有一个：



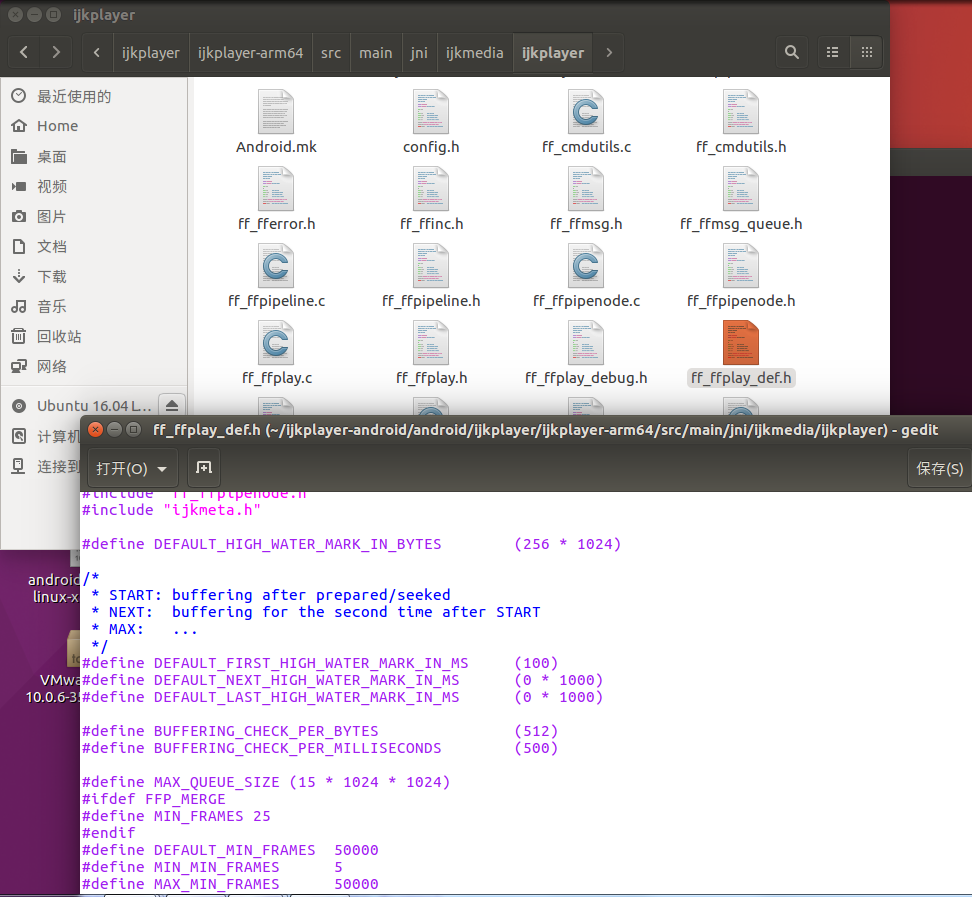
我在github提问了相关问题，结论是FFmpeg本身没有编译成功导致的，如果sdk和ndk配置成功的状态下，或是重启机器注销乌班图重进等方式继续运行编译步骤的代码，最终会成功，成功之后生成so文件，这里会生成不同cpu的不同so分别在不同目录下：



**五：编译完成后的修改注意**

1：在关于ijkPlayer的延迟问题上折腾了很久，由于FFmpeg的c层或者ijk的java层几乎都没有注释，而且意义不明，github也没有相关介绍回答，这里根据经验入列一下修改位置，避免延迟问题：

1：根据视图可以看到我的目录详细地址，在地址中寻找到ff\_ffplay\_def.h这个文件，我们首先需要修改这个c文件



1：首先DEFAULT\_LAST\_HIGH\_WATER\_MARK\_IN\_MS 这个值，原来是5x1000，这里我改成0x1000，尝试改成1x100时报错，这里根据名称来看应该是最大的允许缓冲时间

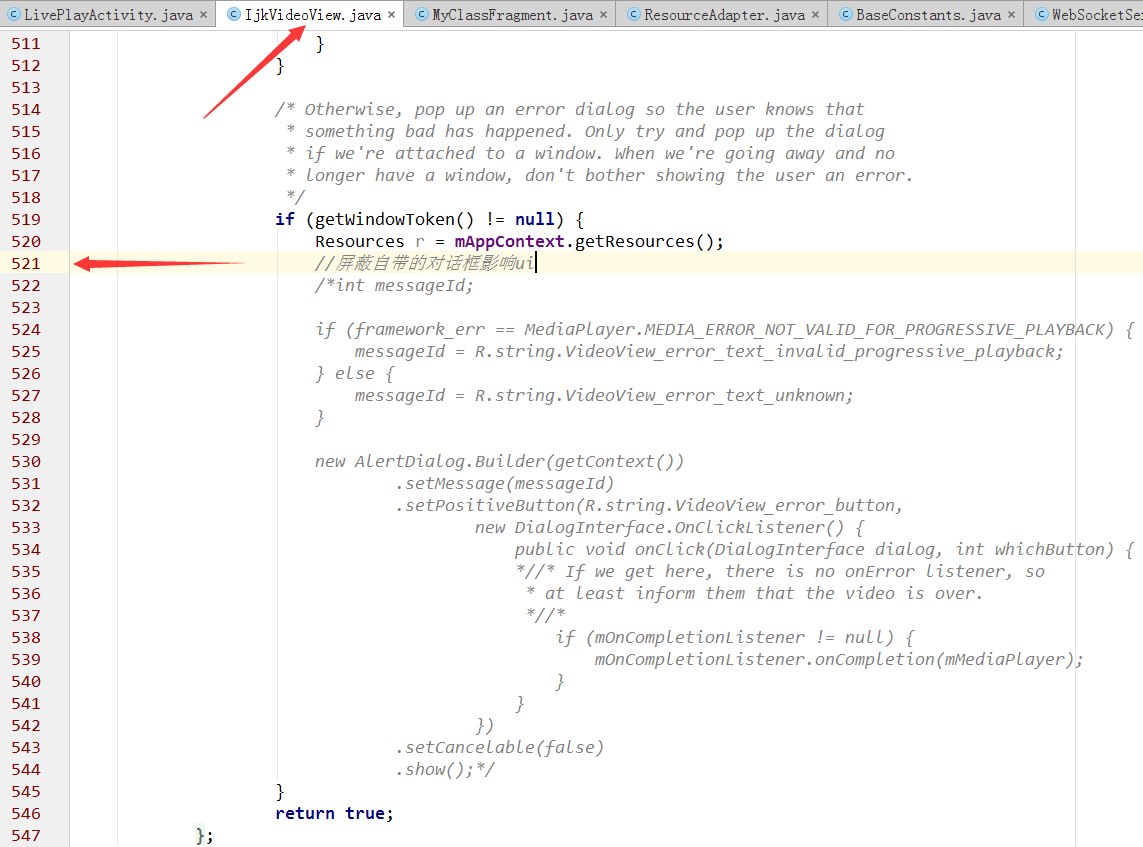
2：BUFFERING\_CHECK\_PER\_BYTES这个值，可能是一个缓冲区的缓冲大小，改下会造成延迟，改大超过512讲无效，最优值就是512

3：MAX\_QUEUE\_SIZE (15 \* 1024 \* 1024)最大缓冲队列，这个值android我这边修改几乎没有影响，但是据ios那边描述改小可以提速

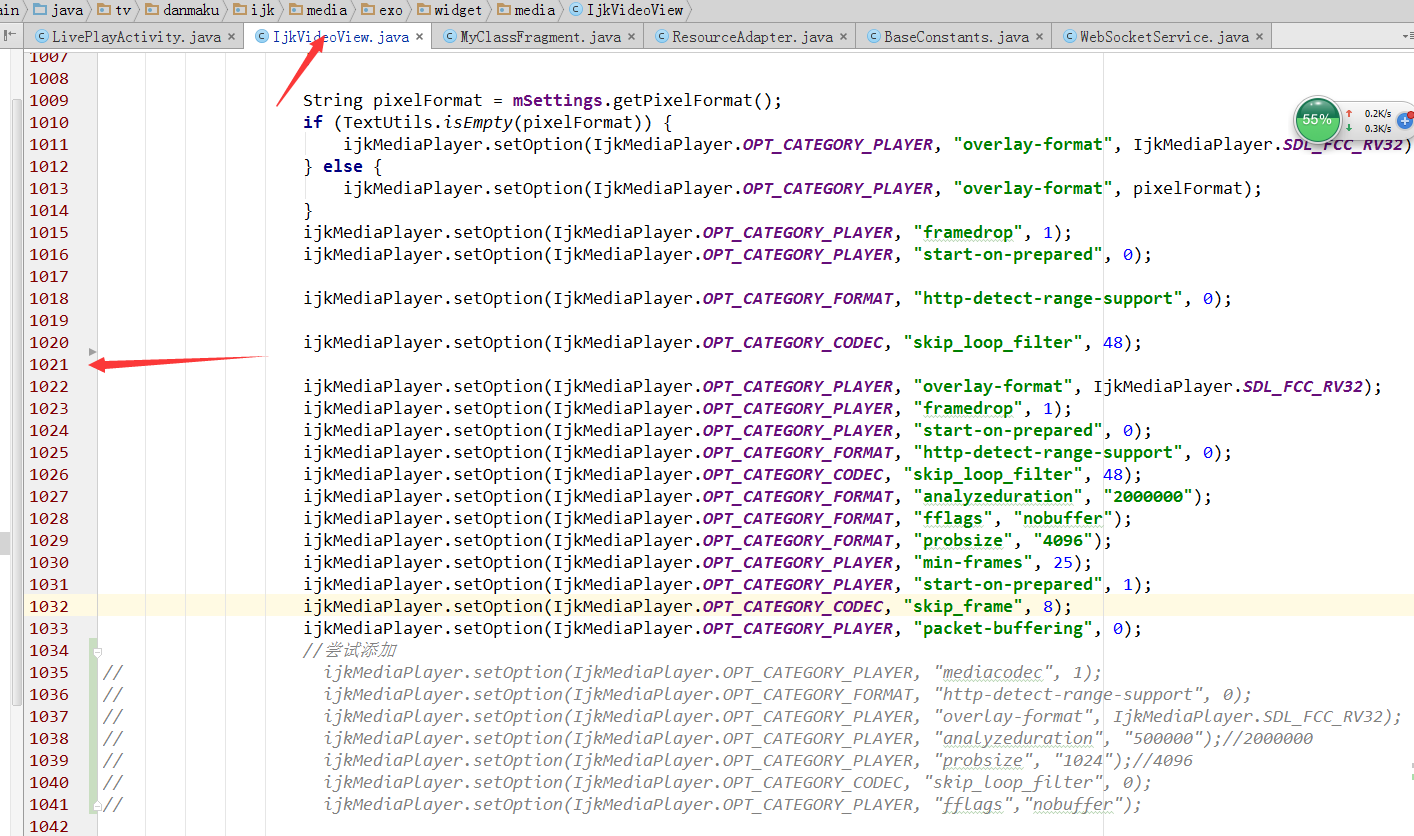
4：MIN\_FRAMES 25 这个帧数建议就调25，我调大时出现过卡很久的情况出现

二：android的java文件修改部分：

1. ：去除自带的警告对话框，这个对话框弹出难看，可以自己改一个，这里我选择屏蔽



（2）：在IjkVideoView的createPlayer方法中，下面添加了如下的配置，执行配置setOption调用的native方法，通过跨平台改变底层C文件配置，通过这些配置可以做到将延迟降到最低，目前来看这个配置是最优的。



1. ：关于视频大小的问题，由于ijk内部写了根据视频流计算mediaPlay并封装了视频高宽，但是实际需要的ui如果根据这个计算可能会很难看，这里我是将它屏蔽掉的：



**六：一些基本方法**

在Demo中他有个*VideoActivity的界面，这里有关于基本的使用方法，其中关于AndroidMediaController这个控制进度的方法，如果是直播或者Ui不允许可以去掉不用，精炼下来的代码主要以下几步：*

**private** IjkVideoView **mVideoView**

mVideoView=(IjkVideoView)findViewById(R.id.***video\_view1***)

**mVideoView1**.setVideoPath(**mVideoPath1**)//string地址

mVideoView1.start();//开始播放

//如何重启播放器

**mVideoView1**.stopPlayback();  
**mVideoView1**.release(**true**);  
**mVideoView1**.stopBackgroundPlay();  
**mVideoView1**.resume();  
**mVideoView1**.start();

最后关于ijkplayer的视频流监听，由于本身项目开源，封装的东西并不完全，关于监听视频流是否在播放避免界面卡死，建议两种方案：

1：**mVideoView1**.isPlaying()方法直接判断，mediaPlayer是否在播放；

2：**mVideoView1**.setOnErrorListener();实现一个错误监听

以上为在编译使用ijkPlayer项目中遇到的问题和处理经验，另附ijkplayer开发交流群（490805051）。