

```
SwrContext *swr_alloc_set_opts(struct SwrContext *s, int64_t out_ch_layout,
AVSampleFormat out_sample_fmt, int out_sample_rate

, int64_t in_ch_layout, AVSampleFormat in_sample_fmt, int in_sample_rate, int
log_offset, void *log_ctx

);
```

参数1: 重采样上下文

参数2: 输出的layout, 如: 5.1声道...

参数3: 输出的样本格式。Float, S16, S24

参数4: 输出的样本率。可以不变。

参数5: 输入的layout。

参数6: 输入的样本格式。

参数7: 输入的样本率。

参数8, 参数9, 日志, 不用管, 可直接传0

针对音频的播放速度, 可以通过样本率的改变而改变。

课间问题

能不能将解码中配置信息直接放置到AudioTrack中播放, 比如采样率和采样位数。不经过上面的转码?

可以 native 原因 实战 播放底层 没有太多的代码 java native opensles

大概讲讲音视频同步原理?

复杂

音视频 音视频同步

音视频同步 绝对同步 是 做不到

音频为准 太少 音频 1.1

假如前面画面卡顿了, 后面的视频帧还没解码出来, 音频在播放, 怎样同步呢

能否解释下 刚刚为何是在主线程解码?

音频解编码后文件大小没多大变化 为什么还需要这个过程

音频就是挤牙膏 袋子中 给你 pcm

牙膏 包装 aac

怎么重采样

重采样

有一个人 非常优秀

90 90 90 80

中学

另外一个中学

重新评估

89.9 90.1 90 90

听到的声音

32位 16位 损失 高清无损的声音

播放器 声音重采样

播放器

视频 yuv surface

就是原先可能可能不是44100都采样成44100?

48000 44100

41000 44100

重采样是不是只有从高采样率到低采样率才有意义?

意义 能够播放

视频 webrtc so

ffmpeg

$44100 * 4 * 60 / 1024 / 1024$ 一分钟都10M吗

webrtc 不一样

搭建服务 器