

10-基本的音视频同步

难点

步骤

这阶段的作业

腾讯课堂《FFmpeg/WebRTC/RTMP音视频流媒体高级开发》<https://ke.qq.com/course/468797?tuin=137bb271>

难点

- 音视频同步采用以音频为基准的方式。
- 音频维系时间轴
- 视频时间戳怎么去对比音频时钟？
- **NaN** (Not a Number, 非数) 是计算机科学中数值数据类型的一类值，表示未定义或不可表示的值。常在浮点数运算中使用。

步骤

1. 时钟结构设计和API设计

```

1 // 这里讲的系统时钟 是通过av_gettime_relative()获取到的时钟, 单位为微妙
2 typedef struct Clock {
3     double pts; // 时钟基础, 当前帧(待播放)显示时间戳, 播放后, 当前帧
    变成上一帧
4     // 当前pts与当前系统时钟的差值, audio、video对于该值是独立的
5     double pts_drift; // clock base minus time at which we updated
    the clock
6     // 当前时钟(如视频时钟)最后一次更新时间, 也可称当前时钟时间
7     double last_updated; // 最后一次更新的系统时钟
8 } Clock;

```

2. stream_open中初始化时钟init_clock(&vidclk); init_clock(&audclk);
3. 设置audio的时钟, 在audio_decode_frame设置audio_clock, 然后在sdl_audio_callback设置audio时钟set_clock(&is->audclk, is->audio_clock);
4. video时间戳比对audio时钟, 在video_refresh

```

1 // 对比audio的时间戳
2 double diff = vp->pts - get_master_clock();
3
4 std::cout << __FUNCTION__ << "vp->pts:" << vp->pts << " - af->pts:" <<
    get_master_clock() << ", diff:" << diff << std::endl;
5
6 if(diff > 0) {
7     *remaining_time = FFMIN(*remaining_time, diff);
8     return;
9 }

```

这阶段的作业

- 实现控制文件的大小

- 实现暂停功能
- 实现seek功能