

两个音频文件混音

- 1 avformat_open_input 打开输入文件
- 1 avformat_find_stream_info寻找输入文件的音频流
- 1 创建编解码器上下文 AVCodecContext
- 1 寻找编解码器avcodec_find_decoder
- 1 打开编解码器avcodec_open2
- 1 av_dump_format打印

- 2 avcodec_find_encoder寻找编码器
- 2 avcodec_alloc_context3 创建编码上下文
- 2 avcodec_open2打开编码器
- 2 avformat_alloc_output_context2初始化一个用于输出的AVFormatContext结构体
- 2 avformat_new_stream在结构体初始化一个音频流
- 2 avcodec_parameters_from_context将编码上下文信息的数据拷贝到输出上下文的condcpar中
- 2 avio_open打开输出的IO
- 2 avformat_write_header写入文件的头部信息
- 2 av_audio_fifo_alloc创建缓存队列

- 3 avfilter_get_by_name获取要使用的过滤器
- 3 avfilter_inout_alloc分配输入输出流
- 3 avfilter_graph_alloc用于管理所有过滤器
- 3 avfilter_graph_create_filter创建过滤器，将其添加到过滤图中
- 3 av_opt_set_bin设置输出配置
- 3 avfilter_graph_parse_ptr将字符串描述的过滤图添加到过滤图中
- 3 avfilter_graph_config检查图管理graph的配置，并使其生效
- 3 avfilter_graph_dump打印信息



缓存管理fifo

- av_audio_fifo_alloc创建缓存队列
- av_audio_fifo_space会返回缓存队列中可供写入的内存
- av_audio_fifo_write将数据写入缓存队列
- av_audio_fifo_size返回fifo中当前存储的采样数量
- av_audio_fifo_read从缓冲队列中读取数据

- 1 打开两个输入音频文件的编解码器
- 2 打开输出文件的io流，并创建两个缓存队列fifo
- 3 配置过滤器规则avfilter
- 4 定义两个线程及两个方法，在两个子线程分别对两个输入的音频文件的数据进行解码，并添加缓存队列fifo
- 5 主线程main方法里，通过方法av_audio_fifo_read从两个缓存队列fifo里面分别取出解码后的数据
- 6 通过方法：av_buffersrc_add_frame，将缓存队列中取出来的数据交给配置好的滤镜进行处理
- 7 通过方法：av_buffersink_get_frame_flags，取出滤镜混合之后样本数据
- 8 对数据进行编码写入io
- 9 释放所有资源，关闭IO.写入成功