大家好，我是北小菜。

今天讲一个实战项目《从零开始编写一个RTSP服务器》

本次视频为系列视频第四讲，讲解内容是：**实现一个基于TCP的RTP同时传输h264和aac的RTSP服务器，并能够拉流播放**

1. 本次实战项目系列视频，分为如下5个：
2. 从零开始编写一个RTSP服务器（1）RTSP协议讲解及代码实现
3. 从零开始编写一个RTSP服务器 （2） 实现一个基于UDP的RTP传输h264的RTSP服务器，并能够进行rtsp拉流播放
4. 从零开始编写一个RTSP服务器（3）实现一个基于UDP的RTP传输aac的RTSP服务器，并能够进行rtsp拉流播放
5. 从零开始编写一个RTSP服务器（4）实现一个基于TCP的RTP同时传输h264和aac的RTSP服务器，并能够进行rtsp拉流播放
6. 从零开始编写一个RTSP服务器（5） 基于我的开源项目BXC\_RtspServer进行详细的源码讲解，这是一个完整可用，支持多线程，基于socket编写的IO多路复用的RTSP流媒体服务器

本系列的前三期视频，我们已经了解和实现了RTSP协议，以及实现了基于UDP的RTP分别传输h264视频流和aac音频流。

本期视频主要讲一讲基于TCP的RTP，如何同时传输h264视频流和aac音频流。

在前三期视频的学习基础上，本期其实比较简单，相对于之前实现的功能，变化如下

1. 客户端请求RTSP的Describe请求时，RTSP服务器返回的SDP协议，需要同时包含音频流和视频流的信息。
2. 客户端请求RTSP的Setup请求时，RTSP服务器不需要再对应创建RTP和RTCP的UDP连接通道，因为TCP版的RTP传输，客户端与服务器交互时，无论是RTSP信令还是RTP数据包或者是RTCP数据包，都是使用同一个tcp连接通道。只不过这个tcp连接通道在发送rtp数据包或者rtcp数据包时，需要加一些分隔字节。
3. 客户端请求RTSP的Play请求时，RTSP服务器在对Play请求回复以后，还需要源源不断的同时向客户端发送音频流和视频流的RTP数据包。
4. 有几点注意，在这个案例项目中，使用的h264视频文件，对应的fps需要是25。另外由于Nalu的数量并不等于视频帧数量的原因，该案例的音视频并不能同步。

//ffmpeg命令行 从mp4视频文件提取h264 码流文件

ffmpeg -i test.mp4 -an -vcodec copy -f h264 test.h264

//ffmpeg命令行 从mp4视频文件提取aac 音频文件

ffmpeg -i test.mp4 -vn -acodec aac test.aac

备注：-i 表示输入文件

-vm disable video / 丢掉视频