文件testH264VideoStreamer.cpp

运行testH264VideoStreamer，vlc播放无法出图像

运行./testOnDemandRTSPServer，vlc可出图像

"h264ESVideoTest" stream, from the file "test.264"

Play this stream using the URL "rtsp://192.168.111.135:8554/h264ESVideoTest"

* ServerMediaSession\* createNewSMS 这是一个静态接口，创建了H264VideoFileServerMediaSubsession对象。
* 根据不同的文件名称创建不同的subsession，以下都以264为例，构建的对象是H264VideoFileServerMediaSubsession，H264VideoFileServerMediaSubsession继承于FileServerMediaSubsession，FileServerMediaSubsession继承于OnDemandServerMediaSubsession，OnDemandServerMediaSubsession继承于ServerMediaSubsession。
* RTSPServer::RTSPClientSession::handleCmd\_SETUP中调用了subsession->getStreamParameters。
* 在OnDemandServerMediaSubsession的getStreamParameters中调用了createNewStreamSource（该接口为纯虚函数），实际为调用264VideoFileServerMediaSubsession::createNewStreamSource获得FramedSource，接着将FramedSource作为入参调用createNewRTPSink。返回一个流的token。这个streamToken就是StreamState对象，setup的时候创建的rtspsink和filesource会长期存在，sink从source获取数据流以rtp形式发送到客户端，streamstate数组来维护一个rtsp流的多路，数组的每个元素代表一路流，内部通过streamtoken表示。Streamstate将sink和source关联起来。
* FramedSource\* H264VideoFileServerMediaSubsession::createNewStreamSource
  + 构造 ByteStreamFileSource对象，打开一个要播放的文件
  + ByteStreamFileSource继承于FramedSource，通过传入FramedSource参数构造 H264or5VideoStreamFramer对象
    - H264or5VideoStreamFramer继承于MPEGVideoStreamFramer，构造父类MPEGVideoStreamFramer
    - MPEGVideoStreamFramer继承于FramedFilter，构造父类FramedFilter，FramedFilter成员对象fInputSource（FramedSource）即为第一步的ByteStreamFileSource对象
  + H264or5VideoStreamFramer有成员对象MPEGVideoStreamParser，通过传入FramedSource参数，对H264or5VideoStreamParser构造
    - H264or5VideoStreamParser继承于MPEGVideoStreamParser，构造父类MPEGVideoStreamParser
    - MPEGVideoStreamParser继承于StreamParser，构造父类StreamParser
    - StreamParser有成员对象fInputSource（FramedSource）即为第一步的ByteStreamFileSource对象
* 返回H264VideoStreamFramer，H264VideoStreamFramer继承于H264or5VideoStreamFramer，继承于MPEGVideoStreamFramer，继承于FramedFilter，继承于FramedSource

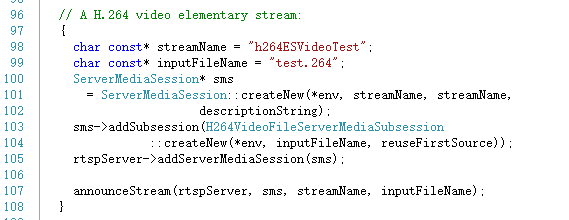
StreamParser中那些getBytes()或getBits()最终会导致读文件的操作。从文件读取一次数据后，StreamParser::afterGettingBytes1（）被调用，StreamParser::afterGettingBytes1（）中做一点简单的工作后便调用fClientContinueFunc这个回调函数，fClientContinueFunc可能指向Frame的函数体也可能是指向RtpSink的函数体。

* MPEGVideoStreamParser继承于StreamParser，StreamParser中的回调fClientContinueFunc函数为静态函数：&MPEGVideoStreamFramer::continueReadProcessing。回调中通过入参clientData转换为MPEGVideoStreamFramer，再调用动态接口continueReadProcessing，该接口中调用FramedSource的afterGetting，该接口中又嵌入一个回调afterGettingFunc\* fAfterGettingFunc，该回调在FramedSource::getNextFrame中赋值。
* 在FramedSource::getNextFrame中调用doGetNextFrame（这是一个纯虚函数），实际调用的是H264or5Fragmenter::doGetNextFrame() 和 ByteStreamFileSource::doReadFromFile()，后者是真正读文件。
* H264or5VideoRTPSink中FramedFilter\* fOurFragmenter 实际为H264or5Fragmenter。H264or5Fragmenter::doGetNextFrame()

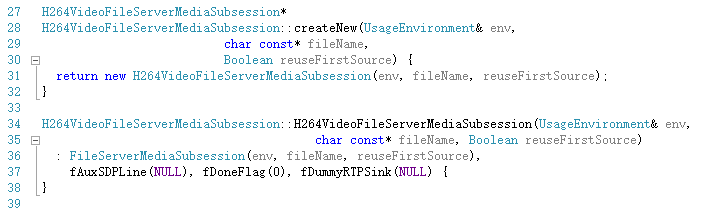
加入fs，参考testOnDemandRTSPServer.cpp，以h264为例

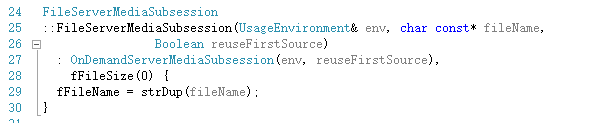
步骤：

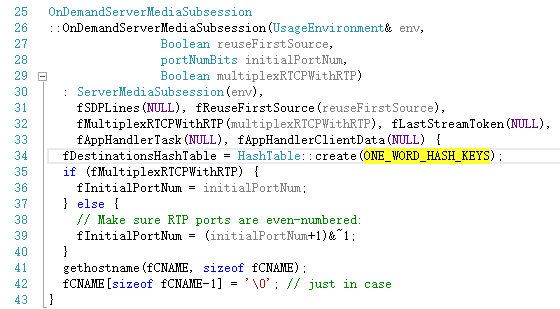
1. 先创建一个rtspserver，RTSPServer\* rtspServer = RTSPServer::createNew(\*env, 8554, authDB);
2. 创建一个session容器，再创建一个subsession，将H264VideoFileServerMediaSubsession添加到该容器中

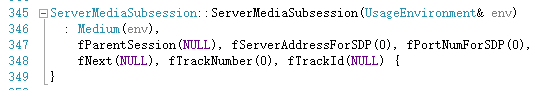


1. 创建H264VideoFileServerMediaSubsession。这个时候并没有对应的打开文件









1. handleCmd\_SETUP中调用了getStreamParameters

