# 原 使用Live555类库实现的网络直播系统

2013年09月15日 00:22:08　阅读数：20709

Live555主要有四个类库：

libUsageEnvironment.lib；libliveMedia.lib；libgroupsock.lib；libBasicUsageEnvironment.lib

将这四个类库以及相关的头文件导入VC++2010之后，可以轻松实现网络直播系统。

在这里直接贴上完整代码，粘贴到VC里面就可以运行。

注：程序运行后，使用播放器软件（VLC Media Player，FFplay等），打开URL：rtp://239.255.42.42:1234，即可收看直播的视频。

[cpp]

```cpp
// 网络直播系统.cpp : 定义控制台应用程序的入口点。
// 雷霄骅
// 中国传媒大学/数字电视技术
// leixiaohua1020@126.com

#include "stdafx.h"

#include "liveMedia.hh"
#include "BasicUsageEnvironment.hh"
#include "GroupsockHelper.hh"

//#define IMPLEMENT_RTSP_SERVER
//#define USE_SSM 1
#ifdef USE_SSM
Boolean const isSSM = True;
#else
Boolean const isSSM = False;
#endif



#define TRANSPORT_PACKET_SIZE 188
#define TRANSPORT_PACKETS_PER_NETWORK_PACKET 7


UsageEnvironment* env;
char const* inputFileName = "test.ts";
FramedSource* videoSource;
RTPSink* videoSink;

void play(); // forward

int main(int argc, char** argv) {
  // 首先建立使用环境：
  TaskScheduler* scheduler = BasicTaskScheduler::createNew();
  env = BasicUsageEnvironment::createNew(*scheduler);

  // 创建 'groupsocks' for RTP and RTCP:
  char const* destinationAddressStr
#ifdef USE_SSM
    = "232.255.42.42";
#else
    = "239.255.42.42";
  // Note: 这是一个多播地址。如果你希望流使用单播地址,然后替换这个字符串与单播地址
#endif
  const unsigned short rtpPortNum = 1234;
  const unsigned short rtcpPortNum = rtpPortNum+1;
  const unsigned char ttl = 7; //


  struct in_addr destinationAddress;
  destinationAddress.s_addr = our_inet_addr(destinationAddressStr);
  const Port rtpPort(rtpPortNum);
  const Port rtcpPort(rtcpPortNum);

  Groupsock rtpGroupsock(*env, destinationAddress, rtpPort, ttl);
  Groupsock rtcpGroupsock(*env, destinationAddress, rtcpPort, ttl);
#ifdef USE_SSM
  rtpGroupsock.multicastSendOnly();
  rtcpGroupsock.multicastSendOnly();
#endif

  // 创建一个适当的“RTPSink”:

  videoSink =
    SimpleRTPSink::createNew(*env, &rtpGroupsock, 33, 90000, "video", "mp2t",
                 1, True, False /*no 'M' bit*/);


  const unsigned estimatedSessionBandwidth = 5000; // in kbps; for RTCP b/w share
  const unsigned maxCNAMElen = 100;
```

```
  const unsigned maxCNAMElen = 100;
  unsigned char CNAME[maxCNAMElen+1];
  gethostname((char*)CNAME, maxCNAMElen);
  CNAME[maxCNAMElen] = '\0';
#ifdef IMPLEMENT_RTSP_SERVER
  RTCPInstance* rtcp =
#endif
    RTCPInstance::createNew(*env, &rtcpGroupsock,
                estimatedSessionBandwidth, CNAME,
                videoSink, NULL /* we're a server */, isSSM);
  // 开始自动运行的媒体

#ifdef IMPLEMENT_RTSP_SERVER
  RTSPServer* rtspServer = RTSPServer::createNew(*env);

  if (rtspServer == NULL) {
    *env << "Failed to create RTSP server: " << env->getResultMsg() << "\n";
    exit(1);
  }
  ServerMediaSession* sms
    = ServerMediaSession::createNew(*env, "testStream", inputFileName,
           "Session streamed by \"testMPEG2TransportStreamer\"",
                     isSSM);
  sms->addSubsession(PassiveServerMediaSubsession::createNew(*videoSink, rtcp));
  rtspServer->addServerMediaSession(sms);

  char* url = rtspServer->rtspURL(sms);
  *env << "Play this stream using the URL \"" << url << "\"\n";
  delete[] url;
#endif


  *env << "开始发送流媒体...\n";
  play();

  env->taskScheduler().doEventLoop();

  return 0; // 只是为了防止编译器警告

}

void afterPlaying(void* /*clientData*/) {
  *env << "...从文件中读取完毕\n";

  Medium::close(videoSource);
  // 将关闭从源读取的输入文件

  play();
}

void play() {
  unsigned const inputDataChunkSize
    = TRANSPORT_PACKETS_PER_NETWORK_PACKET*TRANSPORT_PACKET_SIZE;

  // 打开输入文件作为一个“ByteStreamFileSource":

  ByteStreamFileSource* fileSource
    = ByteStreamFileSource::createNew(*env, inputFileName, inputDataChunkSize);
  if (fileSource == NULL) {
    *env << "无法打开文件 \"" << inputFileName
     << "\" 作为 file source\n";
    exit(1);
  }


  videoSource = MPEG2TransportStreamFramer::createNew(*env, fileSource);


  *env << "Beginning to read from file...\n";
  videoSink->startPlaying(*videoSource, afterPlaying, videoSink);
}
```

完整工程下载地址： http://download.csdn.net/detail/leixiaohua1020/6272839

[cpp]

```

```



文章标签： Live555 直播

个人分类： Live555