

## 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：20734

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：rtsp://239.255.42.42:1234，即可收看直播的视频。

```
[cpp]
1. // 网络直播系统.cpp：定义控制台应用程序的入口点。
2. // 雷霄骅
3. // 中国传媒大学/数字电视技术
4. // leixiaohua1020@126.com
5.
6. #include "stdafx.h"
7.
8. #include "liveMedia.hh"
9. #include "BasicUsageEnvironment.hh"
10. #include "GroupsockHelper.hh"
11.
12. // #define IMPLEMENT_RTSP_SERVER
13. // #define USE_SSM 1
14. #ifndef USE_SSM
15. Boolean const isSSM = True;
16. #else
17. Boolean const isSSM = False;
18. #endif
19.
20.
21.
22. #define TRANSPORT_PACKET_SIZE 188
23. #define TRANSPORT_PACKETS_PER_NETWORK_PACKET 7
24.
25.
26. UsageEnvironment* env;
27. char const* inputFileName = "test.ts";
28. FramedSource* videoSource;
29. RTPSink* videoSink;
30.
31. void play(); // forward
32.
33. int main(int argc, char** argv) {
34.     // 首先建立使用环境：
35.     TaskScheduler* scheduler = BasicTaskScheduler::createNew();
36.     env = BasicUsageEnvironment::createNew(*scheduler);
37.
38.     // 创建 'groupsocks' for RTP and RTCP:
39.     char const* destinationAddressStr
40. #ifdef USE_SSM
41.         = "232.255.42.42";
42. #else
43.         = "239.255.42.42";
44.     // 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
45. #endif
46.     const unsigned short rtpPortNum = 1234;
47.     const unsigned short rtcpPortNum = rtpPortNum+1;
48.     const unsigned char ttl = 7; //
49.
50.
51.     struct in_addr destinationAddress;
52.     destinationAddress.s_addr = our_inet_addr(destinationAddressStr);
53.     const Port rtpPort(rtpPortNum);
54.     const Port rtcpPort(rtcpPortNum);
55.
56.     Groupsock rtpGroupsock(*env, destinationAddress, rtpPort, ttl);
57.     Groupsock rtcpGroupsock(*env, destinationAddress, rtcpPort, ttl);
58. #ifdef USE_SSM
59.     rtpGroupsock.multicastSendOnly();
60.     rtcpGroupsock.multicastSendOnly();
61. #endif
62.
63.     // 创建一个适当的“RTPSink”：
64.
65.     videoSink =
66.         SimpleRTPSink::createNew(*env, &rtpGroupsock, 33, 90000, "video", "mp2t",
67.             1, True, False /*no 'M' bit*/);
68.
69.
70.     const unsigned estimatedSessionBandwidth = 5000; // in kbps; for RTCP b/w share
71.     const unsigned maxNAMELen = 100;
```

```

71.     const unsigned maxCNAMElen = 100,
72.     unsigned char CNAME[maxCNAMElen+1];
73.     gethostname((char*)CNAME, maxCNAMElen);
74.     CNAME[maxCNAMElen] = '\0';
75. #ifdef IMPLEMENT_RTSP_SERVER
76.     RTCPInstance* rtcp =
77. #endif
78.     RTCPInstance::createNew(*env, &rtcpGroupsock,
79.         estimatedSessionBandwidth, CNAME,
80.         videoSink, NULL /* we're a server */, isSSM);
81.     // 开始自动运行的媒体
82.
83. #ifdef IMPLEMENT_RTSP_SERVER
84.     RTSPServer* rtspServer = RTSPServer::createNew(*env);
85.
86.     if (rtspServer == NULL) {
87.         *env << "Failed to create RTSP server: " << env->getResultMsg() << "\n";
88.         exit(1);
89.     }
90.     ServerMediaSession* sms
91.     = ServerMediaSession::createNew(*env, "testStream", inputFileName,
92.         "Session streamed by \"testMPEG2TransportStreamer\"",
93.         isSSM);
94.     sms->addSubSession(PassiveServerMediaSubsession::createNew(*videoSink, rtcp));
95.     rtspServer->addServerMediaSession(sms);
96.
97.     char* url = rtspServer->rtspURL(sms);
98.     *env << "Play this stream using the URL \"" << url << "\"\n";
99.     delete[] url;
100. #endif
101.
102.
103.     *env << "开始发送流媒体...\n";
104.     play();
105.
106.     env->taskScheduler().doEventLoop();
107.
108.     return 0; // 只是为了防止编译器警告
109.
110. }
111.
112. void afterPlaying(void* /*clientData*/) {
113.     *env << "...从文件中读取完毕\n";
114.
115.     Medium::close(videoSource);
116.     // 将关闭从源读取的输入文件
117.
118.     play();
119. }
120.
121. void play() {
122.     unsigned const inputDataChunkSize
123.     = TRANSPORT_PACKETS_PER_NETWORK_PACKET*TRANSPORT_PACKET_SIZE;
124.
125.     // 打开输入文件作为一个"ByteStreamFileSource":
126.
127.     ByteStreamFileSource* fileSource
128.     = ByteStreamFileSource::createNew(*env, inputFileName, inputDataChunkSize);
129.     if (fileSource == NULL) {
130.         *env << "无法打开文件 \"" << inputFileName
131.         << "\" 作为 file source\n";
132.         exit(1);
133.     }
134.
135.
136.     videoSource = MPEG2TransportStreamFramer::createNew(*env, fileSource);
137.
138.
139.     *env << "Beginning to read from file...\n";
140.     videoSink->startPlaying(*videoSource, afterPlaying, videoSink);
141. }

```

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

[cpp]  

1.

版权声明：本文为博主原创文章，未经博主允许不得转载。 <https://blog.csdn.net/leixiaohua1020/article/details/11696449>

文章标签：[Live555](#) [直播](#)

个人分类：[Live555](#)

