



http://ticktick.blog.51cto.com 【复制】 【订阅】

博客 | 图库 | 写博文 | 帮助

首页 | 嵌入式 | 软件编程 | Linux学习与应用 | Android开发 | 视频相关技术 | 工具及调试笔记 | 生活随笔 | 其他

tickTick 的BLOG

相关视频课程

更多



写留言
发消息
去学院学习
加好友链接
进家园 加好友



2013大数据全球技术峰会现场实录视频(共1929人学习)



iOS开发视频教程-iOS数据持久化【企业级中242人学习】



Linux艰辛之路—双机热备与负载均衡视频教学1422人学习

博主的更多文章>>

博客统计信息

51CTO推荐博客

用户名: tickTick
文章数: 92
评论数: 299
访问量: 599228
无忧币: 2187
博客积分: 2810
博客等级: 7
注册日期: 2009-06-13

热门专题

更多>>



Exchange Server 2013 服务器配置详解
阅读量: 1240



从零开始学高德地图JS API
阅读量: 1602



Exchange 2013SP1和O365混合部署详解
阅读量: 872

热门文章

c++构造函数详解

undefined reference问题..

c++异常处理机制示例及讲解

Doxygen详细介绍 (三) (..

C++串口编程实例

入门视频采集与处理 (学..

谈谈RTP传输中的负载类型..

ortp库使用入门

原创 ortp编程示例代码

2011-11-03 13:13:08

标签: ortp 编程 示例 发送接收 代码

原创作品，允许转载，转载时请务必以超链接形式标明文章 [原始出处](#)、作者信息和和本声明。否则将追究法律责任。
<http://ticktick.blog.51cto.com/823160/704891>

鉴于很多网友找我要ortp的示例代码，因此，今天抽空把相关资料整理了一下，写了一个windows版的ortp示例程序，发布在这里供网友们参考吧。

编译及运行环境：VS2008，windows

编程语言：c/c++，ortp库为c语言封装，我用c++对其进行了进一步封装，如果需要c语言的封装接口，只需要把类中相关函数提取出来即可使用。

ortp库：ortp-0.9.1（由于是以前写的代码，故用的ortp库比较老，但不影响使用和学习，我附件中的工程中已经把ortp-0.9.1库文件添加进去了）

整个测试代码在工程的附件中，大家下载后直接编译后，在Debug目录下打开2个本程序，一个选择Client，一个选择Server，即可看到测试效果。

下面，我的相关代码发布如下（附件中有完整的工程）。

一、ORTP接收端封装类

```
01. //////////////////////////////////////////////////////////////////////////////////////////////////////////////////
02. // COPYRIGHT NOTICE
03. // Copyright (c) 2011, 华中科技大学 卢俊（版权声明）
04. // All rights reserved.
05. //
06. /// @file CortpClient.h
07. /// @brief ortp客户端类声明文件
08. ///
09. /// 实现和提供ortp的客户端应用接口
10. ///
11. /// @version 1.0
12. /// @author 卢俊
13. /// @date 2011/11/03
14. //
15. //
16. // 修订说明:
17. //////////////////////////////////////////////////////////////////////////////////////////////////////////////////
18.
19. #ifndef CORTPCLIENT_H_
20. #define CORTPCLIENT_H_
21.
22. #include <ortp/ortp.h>
23. #include <string>
24.
25. /**
26. * CortpClient ortp客户端管理类
27. *
28. * 负责封装和提供ortp相关接口
29. */
30. class CortpClient
31. {
32. public:
33.
34. /** 构造函数/析构函数
35. *
36. * 在创建/销毁该类对象时自动调用
```

搜索BLOG文章

搜索

最近访客



Bill_Hoo



卧听潇雨



ruiqi..



王者鹤兮



xhb29



OTExc..



benji..



ddl0601



睦文峰



majestli



8207531



草蜢

最新评论

niceheart: 回复 tickTick: 楼主可否给我一个..

tickTick: 回复 niceheart: 这个代码只是用..

niceheart: 请问楼主在项目中是如何去用使用? ..

kangear: 你这个似乎是NDK编程的, Android系..

kkdluf: 太感谢楼主了

51CTO推荐博文

更多>>

【云图】如何创建云图（云存储）

性能分析工具-PerfView

RMI原理揭秘之安全通信

使用“忽略授权表”参数登录多实..

ASP.NET MVC学前篇之扩展方法、链..

【web开发】之常用上传下载代码

MHA高可用部署及测试

修改MFC OCX的GUID

基于Python的HTTPS协议模拟登陆+..

[android底层]jni中获取自定义的..

Eclipse常用的一些设置

友情链接

锦哉琉璃版氮锦氢燃

肖骁的博客

Dian团队官方主页

```
37. */
38. COrtpClient();
39. ~COrtpClient();
40.
41. /** ORTP模块的初始化
42. *
43. * 在整个系统最开始调用, 负责ORTP库的初始化
44. * @return: bool 是否成功
45. * @note:
46. * @see:
47. */
48. static bool init();
49.
50. /** ORTP模块的逆初始化
51. *
52. * 在系统退出前调用, 负责ORTP库的释放
53. * @return: bool 是否成功
54. * @note:
55. * @see:
56. */
57. static bool deInit();
58.
59. /** 创建RTP接收会话
60. *
61. * 负责产生RTP接收端会话, 监听服务器端的数据
62. * @param: const char * localip 本地ip地址
63. * @param: int localport 本地监听端口
64. * @return: bool 是否成功
65. * @note:
66. * @see:
67. */
68. bool create(const char * localip, int localport );
69.
70. /** 获取接收到的rtp包
71. *
72. * 将接收到的rtp数据包取出
73. * @param: char * pBuffer
74. * @param: int & len
75. * @return: bool 是否成功
76. * @note:
77. * @see:
78. */
79. bool get_rcv_data( char *pBuffer, int &len );
80.
81. private:
82.
83. RtpSession *m_pSession; /** rtp会话句柄 */
84.
85. long m_curTimeStamp; /** 当前时间戳 */
86. int m_timeStampInc; /** 时间戳增量 */
87.
88. };
89.
90. #endif // COrtpClient_H_
91.
92. ///////////////////////////////////////////////////////////////////
93. // COPYRIGHT NOTICE
94. // Copyright (c) 2011, 华中科技大学 卢俊 (版权声明)
95. // All rights reserved.
96. //
97. /// @file COrtpClient.cpp
98. /// @brief ortp客户端实现文件
99. ///
100. /// 实现和提供ortp的客户端应用接口
101. ///
102. /// @version 1.0
103. /// @author lujun
104. /// @date 2011/11/03
105. //
106. // 修订说明:
107. //
108. ///////////////////////////////////////////////////////////////////
109.
110. #include "COrtpClient.h"
111.
112. /* the payload type define */
113. #define PAYLOAD_TYPE_VIDEO 34
114.
115. /* RTP video Send time stamp increase */
116. #define VIDEO_TIME_STAMP_INC 3600
117.
118. /** 从rtp接收缓冲区一次读取的字节数 */
119. #define READ_RECV_PER_TIME 1024
120.
121. COrtpClient::COrtpClient()
122. {
123.     m_pSession = NULL;
124.     m_timeStampInc = 0;
125.     m_curTimeStamp = 0;
126. }
127.
128. COrtpClient::~COrtpClient()
129. {
130.     if (!m_pSession)
131.     {
132.         rtp_session_destroy(m_pSession);
133.     }
134. }
```

http://ticktick.blog.51cto.com/823160/704891

2/8

```
134. }
135.
136. bool COrtpClient::init()
137. {
138.     int ret;
139.     WSADATA wsaData;
140.
141.     /** 初始化winsocket */
142.     if ( WSASStartup(MAKEWORD(2,2), &wsaData) != 0)
143.     {
144.         return false;
145.     }
146.
147.     ortp_init();
148.     ortp_scheduler_init();
149.
150.     return true;
151. }
152.
153. bool COrtpClient::deInit()
154. {
155.     ortp_exit();
156.
157.     if (WSACleanup() == SOCKET_ERROR)
158.     {
159.         return false;
160.     }
161.
162.     return true;
163. }
164.
165. bool COrtpClient::create( const char * localip, int localport )
166. {
167.     if ( m_pSession != NULL)
168.     {
169.         return false;
170.     }
171.
172.     /** 创建新会话 */
173.     m_pSession = rtp_session_new(RTP_SESSION_RECVONLY);
174.     if ( !m_pSession)
175.     {
176.         return false;
177.     }
178.
179.     /** 配置相关参数 */
180.     rtp_session_set_scheduling_mode(m_pSession,1);
181.     rtp_session_set_blocking_mode(m_pSession,1);
182.     rtp_session_set_local_addr(m_pSession,localip,localport);
183.     rtp_session_enable_adaptive_jitter_compensation(m_pSession,1);
184.     rtp_session_set_jitter_compensation(m_pSession,40);
185.
186.     rtp_session_set_payload_type(m_pSession,PAYLOAD_TYPE_VIDEO);
187.     m_timeStampInc = VIDEO_TIME_STAMP_INC;
188.
189.     return true;
190. }
191.
192. bool COrtpClient::get_rcv_data( char *pBuffer, int &len )
193. {
194.     int rcvBytes = 0;
195.     int totalBytes = 0;
196.     int have_more = 1;
197.
198.     while(have_more)
199.     {
200.         if ( totalBytes+READ_RECV_PER_TIME > len )
201.         {
202.             /** 缓冲区大小不够 */
203.             return false;
204.         }
205.         rcvBytes = rtp_session_rcv_with_ts(m_pSession,pBuffer+totalBytes,READ_RECV_PER_TIME,
206. m_curTimeStamp,&have_more);
207.         if (rcvBytes <= 0)
208.         {
209.             break;
210.         }
211.         totalBytes += rcvBytes;
212.     }
213.
214.     /** 判断是否读取到数据 */
215.     if (totalBytes == 0)
216.     {
217.         return false;
218.     }
219.
220.     /** 记录有效字节数 */
221.     len = totalBytes;
222.
223.     /** 时间戳增加 */
224.     m_curTimeStamp += m_timeStampInc;
225.
226.     return true;
227. }
```

二、ORTP发送端封装类

```

01.  //////////////////////////////////////
02.  //  COPYRIGHT NOTICE
03.  //  Copyright (c) 2011, 华中科技大学 卢俊 (版权声明)
04.  //  All rights reserved.
05.  //
06.  /// @file    COrtpServer.h
07.  /// @brief   ortp服务器类声明文件
08.  ///
09.  /// 实现和提供ortp的服务器端应用接口
10.  ///
11.  /// @version 1.0
12.  /// @author  卢俊
13.  /// @date    2011/11/03
14.  //
15.  //
16.  //  修订说明:
17.  //////////////////////////////////////
18.
19.  #ifndef CORTPSERVER_H_
20.  #define CORTPSERVER_H_
21.
22.  #include <ortp/ortp.h>
23.
24.  /**
25.   *   COrtpServer RTP发送类
26.   *
27.   *   负责使用RTP协议进行数据的发送
28.   */
29.  class COrtpServer
30.  {
31.  public:
32.
33.      /** 构造函数
34.       *
35.       *   该函数为该类的构造函数, 在创建该类对象时自动调用
36.       */
37.      COrtpServer();
38.
39.      /** 析构函数
40.       *
41.       *   该函数执行析构操作, 由系统自动调用
42.       */
43.      ~COrtpServer();
44.
45.      /** ORTP模块的初始化
46.       *
47.       *   在整个系统最开始调用, 负责ORTP库的初始化
48.       *   @return: bool  是否成功
49.       *   @note:
50.       *   @see:
51.       */
52.      static bool init();
53.
54.      /** ORTP模块的逆初始化
55.       *
56.       *   在系统退出前调用, 负责ORTP库的释放
57.       *   @return: bool  是否成功
58.       *   @note:
59.       *   @see:
60.       */
61.      static bool deInit();
62.
63.      /** 创建RTP接收会话
64.       *
65.       *   负责产生RTP接收端会话, 监听服务器端的数据
66.       *   @param: const char * destIP  目的地址的IP
67.       *   @param: int destport  目的地址的监听端口号
68.       *   @return: bool  是否成功
69.       *   @note:
70.       *   @see:
71.       */
72.      bool create(const char * destIP, int destport );
73.
74.      /** 发送RTP数据
75.       *
76.       *   将指定的buffer中的数据发送到客户端
77.       *   @param: unsigned char * buffer  需要发送的数据
78.       *   @param: int len  有效字节数
79.       *   @return: int  实际发送的字节数
80.       *   @note:
81.       *   @see:
82.       */
83.      int send_data( unsigned char *buffer, int len );
84.
85.  private:
86.
87.      RtpSession *m_pSession;    /** rtp会话句柄 */
88.
89.      long        m_curTimeStamp; /** 当前时间戳 */
90.      int         m_timeStampInc; /** 时间戳增量 */
91.
92.      char        *m_ssrc;       /** 数据源标识 */
93.  };
94.

```

```

95.  #endif // COrtpServer_H_
96.
97.  //////////////////////////////////////
98.  //  COPYRIGHT NOTICE
99.  //  Copyright (c) 2011, 华中科技大学 卢俊 (版权声明)
100. //  All rights reserved.
101. //
102. /// @file    COrtpServer.cpp
103. /// @brief   ortp服务器类实现文件
104. ///
105. /// 实现和提供ortp的服务器端应用接口
106. ///
107. /// @version 1.0
108. /// @author  lujun
109. /// @date    2011/11/03
110. ///
111. ///
112. /// 修订说明:
113. //////////////////////////////////////
114.
115. #include "COrtpServer.h"
116.
117. /* the payload type define */
118. #define PAYLOAD_TYPE_VIDEO 34
119.
120. /* RTP video Send time stamp increase */
121. #define VIDEO_TIME_STAMP_INC 3600
122.
123. COrtpServer::COrtpServer()
124. {
125.     m_ssrc = NULL;
126.     m_pSession = NULL;
127.     m_timeStampInc = 0;
128.     m_curTimeStamp = 0;
129. }
130.
131. COrtpServer::~COrtpServer()
132. {
133.     if (!m_pSession)
134.     {
135.         rtp_session_destroy(m_pSession);
136.     }
137. }
138.
139. bool COrtpServer::init()
140. {
141.     int ret;
142.     WSADATA wsaData;
143.
144.     /** 初始化winsocket */
145.     if (WSAStartup(MAKEWORD(2,2), &wsaData) != 0)
146.     {
147.         return false;
148.     }
149.
150.     ortp_init();
151.     ortp_scheduler_init();
152.
153.     return true;
154. }
155.
156. bool COrtpServer::deInit()
157. {
158.     ortp_exit();
159.
160.     if (WSACleanup() == SOCKET_ERROR)
161.     {
162.         return false;
163.     }
164.
165.     return true;
166. }
167.
168. bool COrtpServer::create( const char * destIP, int destport )
169. {
170.     m_ssrc = getenv("SSRC");
171.
172.     m_pSession = rtp_session_new(RTP_SESSION_SENDBYTES);
173.
174.     rtp_session_set_scheduling_mode(m_pSession,1);
175.     rtp_session_set_blocking_mode(m_pSession,1);
176.     rtp_session_set_remote_addr(m_pSession,destIP,destport);
177.
178.     if(m_ssrc != NULL)
179.     {
180.         rtp_session_set_ssrc(m_pSession,atoi(m_ssrc));
181.     }
182.
183.     rtp_session_set_payload_type(m_pSession,PAYLOAD_TYPE_VIDEO);
184.     m_timeStampInc = VIDEO_TIME_STAMP_INC;
185.
186.     return true;
187. }
188.
189. int COrtpServer::send_data( unsigned char *buffer, int len )
190. {
191.     int sendBytes = 0;

```

```

192.
193.     /** 强转 */
194.     const char *sendBuffer = (const char*)buffer;
195.
196.     sendBytes = rtp_session_send_with_ts(m_pSession, sendBuffer, len, m_curTimeStamp);
197.
198.     if ( sendBytes > 0)
199.     {
200.         m_curTimeStamp += m_timeStampInc; /** 增加时间戳 */
201.     }
202.
203.     return sendBytes;
204. }

```

三、测试程序

```

01.  //////////////////////////////////////
02.  //  COPYRIGHT NOTICE
03.  //  Copyright (c) 2011, 华中科技大学 卢俊 ( 版权声明 )
04.  //  All rights reserved.
05.  //
06.  /// @file    main.cpp
07.  /// @brief   ortp测试文件
08.  ///
09.  /// 测试ortp发送结构体
10.  ///
11.  /// @version 1.0
12.  /// @author   卢俊
13.  /// @e-mail   lujun.hust@gmail.com
14.  /// @date    2011/10/19
15.  //
16.  //
17.  //  修订说明:
18.  //////////////////////////////////////
19.
20. #include <iostream>
21.
22. #include "COrtpClient.h"
23. #include "COrtpServer.h"
24.
25. /** 本地IP地址 */
26. const char * LOCAL_IP_ADDR = "127.0.0.1";
27.
28. /** 本地监听端口 */
29. const int LOCAL_RTP_PORT = 8000;
30.
31. /** 目的监听端口 */
32. const int DEST_RTP_PORT = 8000;
33.
34. /** 目的IP地址 */
35. const char * DEST_IP_ADDR = "127.0.0.1";
36.
37. /** 一次发送的数据长度 */
38. const int SEND_LEN_PER_TIME = 8*1024;
39.
40. /** 接收缓冲区的总大小 */
41. const int RECV_BUFFER_LEN = 10*1024;
42.
43. /** 一次接收的数据长度 */
44. const int RECV_LEN_PER_TIME = 1024;
45.
46. bool ortpServer()
47. {
48.     COrtpServer ortpServer;
49.
50.     COrtpServer::init();
51.
52.     if (!ortpServer.create(DEST_IP_ADDR, DEST_RTP_PORT))
53.     {
54.         std::cout << "ortpServer.create fail!\n";
55.         getchar();
56.         getchar();
57.         return false;
58.     }
59.
60.     unsigned char * buffer = new unsigned char[SEND_LEN_PER_TIME];
61.
62.     while (1)
63.     {
64.         if ( ortpServer.send_data(buffer, SEND_LEN_PER_TIME) <= 0)
65.         {
66.             std::cout << "send fail!\n";
67.         }
68.
69.         Sleep(100);
70.         std::cout << "send bytes\n";
71.     }
72.
73.     delete [] buffer;
74.
75.     COrtpClient::deInit();
76.
77.     return true;
78. }
79.
80. bool ortpClient()

```

```
81. {
82.     COrtpClient ortpClient;
83.
84.     COrtpClient::init();
85.
86.     if (!ortpClient.create(LOCAL_IP_ADDR, LOCAL_RTP_PORT))
87.     {
88.         std::cout << "ortpClient.create fail!\n";
89.         getchar();
90.         getchar();
91.         return false;
92.     }
93.
94.     char *buffer = new char[RECV_BUFFER_LEN];
95.
96.     while(1)
97.     {
98.         int len = RECV_BUFFER_LEN;
99.         if (!ortpClient.get_recv_data(buffer, len))
100.        {
101.            Sleep(10);
102.            continue;
103.        }
104.
105.        std::cout << "successful recv,data len =" << len << std::endl;
106.    }
107.
108.    COrtpClient::deInit();
109.
110.    delete [] buffer;
111.
112.    return true;
113. }
114.
115. void main()
116. {
117.     std::cout << "enter num,1 ->client, 2->server! \n";
118.
119.     int num;
120.
121.     std::cin >> num;
122.
123.     while(1)
124.     {
125.         if (num == 1)
126.         {
127.             ortpClient();
128.             break;
129.         }
130.         else if (num == 2)
131.         {
132.             ortpServer();
133.             break;
134.         }
135.         else
136.         {
137.             std::cout << "please input 1 or 2 !\n";
138.         }
139.     }
140.
141.     getchar();
142.     getchar();
143. }
```

有关ORTP的介绍、RTP的介绍、RTP的负载类型和时间戳的含义等理论性的东西，都可以在我博客中的其他文章中找到，以上就是整个工程的代码，注释不是很多，因为有些地方我也不是特别清楚，比如jitter、scheduling什么的，如果有什么其他疑问欢迎留言或者E-mail来信交流。

本文出自“对影成三人”博客，请务必保留此出处<http://ticktick.blog.51cto.com/823160/704891>

分享至:

1

收藏 +

璇玑CTO、zhunavaro、kangear 3人

了这篇文章

附件下载:

 ORTP示例代码

类别: 软件编程 | 阅读(4002) | 评论(5) | [返回博主首页](#) | [返回博客首页](#)

[上一篇 软件系统设计思路](#) [下一篇 养成良好的项目文件上传习惯](#)



相关文章



关注51CTO博客微信
获得每日精选推荐文章
微信号: blog51cto



关注51CTO官方微信
我们提供不一样的东西
微信号: weixin51cto

- [a programmer should know...](#)
- [字符串数组定义](#)
- [基于Linux的socket编程模板](#)
- [拷贝网页内容增加版权信息的 JavaScript 代..](#)
- [javascript 中的eval\(\)](#)
- [如何编写代码才能使得效率高](#)
- [PHP+MySQL事务处理示例代码](#)
- [网页制作几个小技巧\(含代码示例\)](#)
- [php小偷实例教程](#)

文章评论

- [1楼]

璇玑CTO

回复

2011-11-05 09:20:09

mark一下
- [2楼]

zhunavaro

回复

2012-10-11 17:01:10

楼主，谢谢你的例子！
我有点不明白的是，通过：ortpServer.send_data(buffer,SEND_LEN_PER_TIME)，一次发送8k字节的数据，不会超过MTU的值吗？
ortp库内部会自动处理吗？
- [3楼]

zhunavaro

回复

2012-10-12 11:45:22

楼主，在上面的例子中，一次发送8k的数据，然后时间戳就增加3600，按照25fps考虑的话，是不是这里假设一帧数据就是8k？
- [4楼]

kangear

回复

2013-04-02 13:21:32

怎么还有接收端程序，vlc播放器也可以直接播放rtp流吧。
- [5楼]

kangear

回复

2013-04-02 14:15:38

我是一个初学linux的，vs也没有用过，要是linux下makefile工程就好太好了。

发表评论 [2014 WOT全球软件技术峰会【火热抢票中】](#)

昵 称：

[登录](#) [快速注册](#)

验证码：

请点击后输入验证码 [博客过2级，无需填写验证码](#)

内 容：