



Chita

基于JMF及RTP协议的局域网通讯工具

[11211017 陈扬羽] << 2

# 简介

---

- ▶ Chita是一个实现在局域网内音频及视频通讯的程序，当然它很弱小所以算不上软件。它的名字源于北印度语，意为猎豹，当然这是因为我希望它能很迅猛。不过后面才发现它的速度只能算是Cat（我可没有说Java很慢~~），反正你用了就知道了……
  - ▶ 它是基于RTP(Real-time Transport Protocol)的，采用了JMF框架，能够实现局域网内的点对点通讯以及多对多通讯（这里用到了组播技术）
  - ▶ 由于JMF是一个很古老的框架了，所以实际上这个项目的实用价值不高，因为它必须在32位的JRE下使用，而且必须先安装JMF。总的来说，这只是一个玩具，你要是在玩游戏的时候想用它和队友交流的话还是可以的。
  - ▶ 虽然如此，但是编写它还是让我付出了艰辛的劳动，我们等等再说。
-

## 文件列表 1/2

---

- ▶ AVReceiver.java
- ▶ AVTransmit.java
- ▶ BroadcastMode.java
- ▶ **Chita.java** \*主程序入口
- ▶ MeetingMode.java
- ▶ PointToPointMode.java
- ▶ TestMode.java
- ▶ 总计7个文件，位于base包中



## 文件列表 2/2

---

- ▶ CaptureConfigPanel.java
- ▶ CaptureDeviceDetector.java
- ▶ DetectInternet.java
- ▶ EditTargetPanel.java
- ▶ GUIPrintStream.java
- ▶ JAudioFormatChooser.java
- ▶ JVideoFormatChooser.java
- ▶ ModeSelectPanel.java
- ▶ RTPExport.java
- ▶ RTPSocketAdapter.java
- ▶ 总计10个文件，位于helper包中



# 技术要点

---

- ▶ Chita的核心部分是JMF编程，它使用了JMF提供的API，用来管理硬件设备，获取信息源，创建处理器以及播放器等。通过各种接口的调用，我就能获取硬件设备的信息并且调用这些设备，这比用C/C++来完成简便地多。在JMF的基础上，再实现一个网络通信的功能即可。
  - ▶ 在网络编程这部分我选择了RTP协议，它是基于UDP的，没有TCP繁琐的三次握手，牺牲了安全性换来了实时性，（这个申老师讲过了）。而且，JMF正好天然支持RTP。在网络编程这部分Java有C/C++不具有的优势。
  - ▶ 在GUI方面采用的是卡片式布局，并且使用了SwingWorker以及LAF。
  - ▶ 在网络通讯的部分使用了多线程
  - ▶ 在一些小地方使用了流的重定向
-

# 困难清单

---

## 一. 古老的JMF早已停止维护

虽然我曾经考虑过是否要更换框架，但是Java对于多媒体框架的开发似乎一直没有新进展。

## 二. 有关JMF编程及Java RTP的资料及其稀少

通过学校提供的论文搜索系统我找到了很多珍贵的资料（我发现有人居然用JMF做学位申请论文 o(╯□╰)o）

## 三. 大量的英文文档需要阅读

由于JMF在02年之后就没有更新过，在Oracle的网站上差点就找不到JMF的文档了。不过没办法，这些文档是最重要的参考。令人惊喜的是我找到了一些实例代码，通过复用及改写那些代码我完善了一些功能。

## 四. 需要多台计算机用于测试

这个是没办法用虚拟机来解决的，因为硬件设备只有一套。我不得不麻烦舍友和同学借我电脑用……

## 五. 不在统一环境下编写代码带来的字符集问题

这个纯粹是自己手贱了……最后只好修改编码格式然后重新写一些代码。

---



# 总结

---

- ▶ 好了，到这里差不多就结束了。Chita是一个不成熟的程序，而且我不再打算继续维护下去了，JMF已经被淘汰了，Chita一开始就注定只能是一个Demo。不过在写Chita的过程中我才意识到我真正地在学习Java。课堂上的知识没有运用都会被遗忘，真正动手去做才让我看见未知的领域是多么辽阔，而且充满了发现的乐趣。
- ▶ 最后，我要说明一下小组分工。我想你已经猜到了，所有的工作都是我一个人完成的……如果实在要四个人的话，就权且当做搜索资料的我、查看文献的我、编写代码的我、测试检查的我四个人吧……  
o(╰□╯)o

