

# 浙江大学

## 本科实验报告

课程名称: Java 应用技术

姓 名: XX

学 院: 计算机科学与技术学院

系: 计算机科学与技术系

专 业: 计算机科学与技术学院

学 号: 3140102441

指导教师: 翁恺

2016 年 12 月 30 日

# 浙江大学实验报告

课程名称: java 应用技术 实验类型: 综合

实验项目名称: CSCW Markdown Editor

学生姓名: XX 专业: 计算机科学与技术 学号: XXX

同组学生姓名: 张寒城 指导老师: 翁恺

## 一、 实验目的和要求

在已经实现的 markdown editor 的基础上, 实现两个 markdown editor 之间的通讯, 即实现共同编辑, 两边显示的内容能够时刻保持一致。

## 二、 实验原理

使用 socket 实现两台计算机之间的通信。其中一台计算机作为服务器, 开启一个线程, 新建一个 server 类, 接受数据, 另外一台计算机作为客户端, 向服务器发起连接, 这里的话, 如果要跟别人的电脑连接, 就需要知道对方的 IP 地址, 如果跟自己连接, 那么只需要向 localhost 发起连接就好了。

服务器和客户端连接好了之后, 两边的线程都在时刻接受数据。每当任意一方的用户对输入进行改变之后, 就会把 markdown 的内容发给给另一方, 包括光标的位置, 然后另一方接受到数据之后, 就会更新 markdown 的内容, 以及设置光标的位置, 从而实现通讯。

### 三、 实验过程

服务器线程:

```
class ServerThread extends Thread{
    public void run(){
        try{
            server = new ServerSocket(PORT);
            socket = server.accept();
```

客户端线程:

```
class ClientThread extends Thread{
    public void run(){
        try{
            IsConnect = true;
            //addr = InetAddress.getByName("10.111.230.183");
            addr = InetAddress.getLocalHost() ;
            socket = new Socket(addr, PORT);
```

接受数据:

```
while(true){
    String line = "";
    String result = "";
    while (!(line = in.readLine()).equals("pos"))
    {
        result += (line+"\n");
    }
}
```

发送数据:

```
textarea_edit.addKeyListener(new KeyListener() {

    @Override
    public void keyTyped(KeyEvent arg0) {
        // TODO Auto-generated method stub
    }

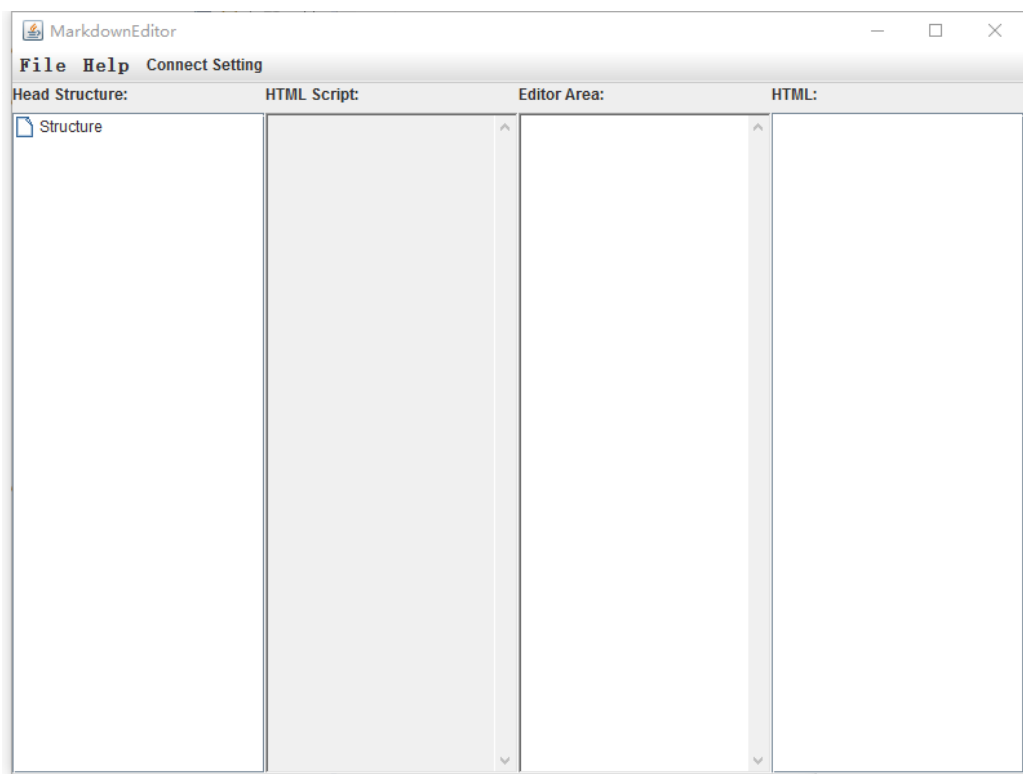
    @Override
    public void keyReleased(KeyEvent arg0) {
        // TODO Auto-generated method stub
    }
});
```

```
        if(IsConnect){  
            Integer pos =  
textarea_edit.getCaretPosition();  
            String string =  
textarea_edit.getText().toString();  
            string += "\npos";  
            string += ("\n" + pos.toString());  
            System.out.println(string);  
            out.println(string);  
        }  
    }  
}
```

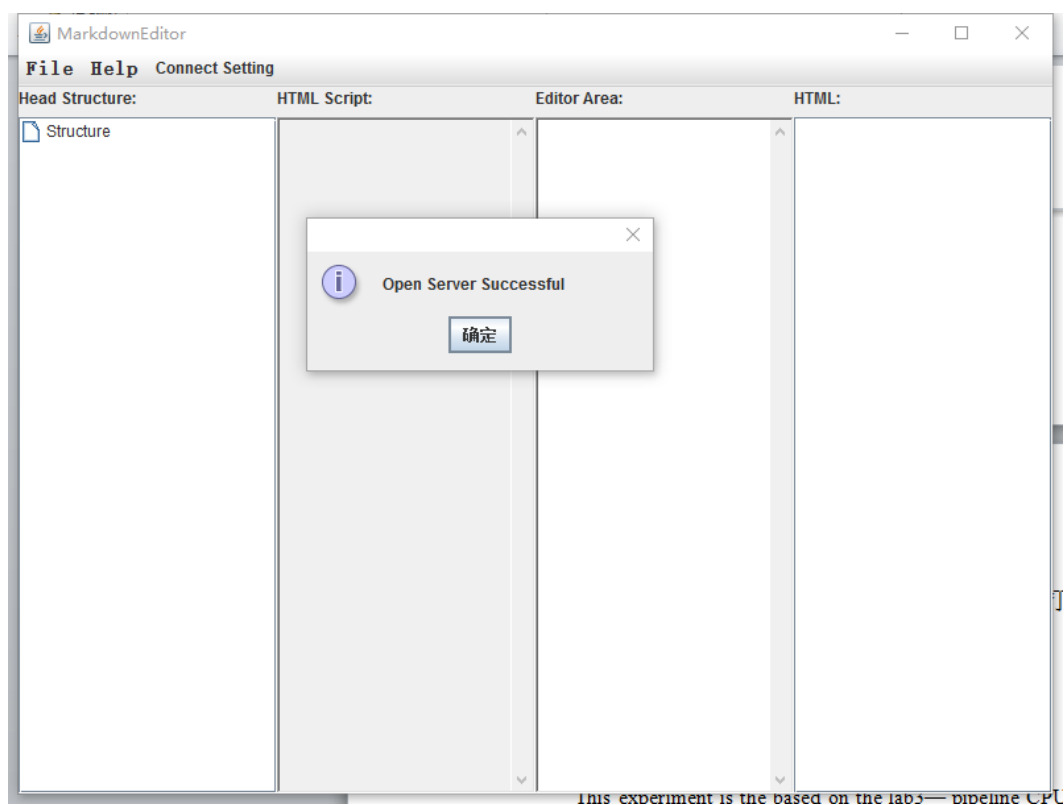
#### 四、 实验结果分析

与自己的电脑连接：

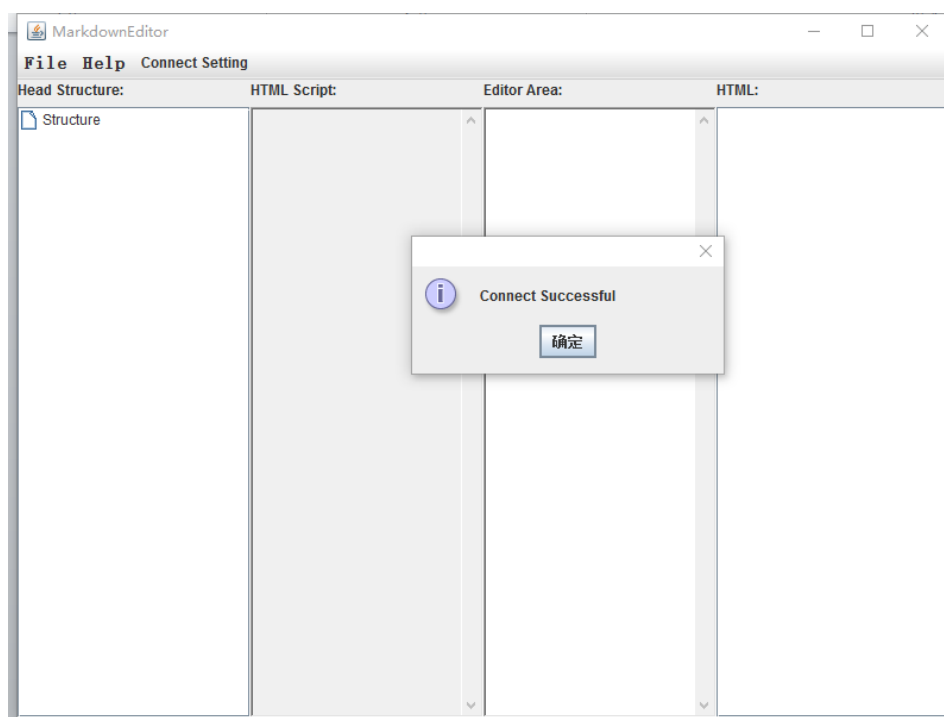
打开作为服务器的 markdown， 初始 GUI



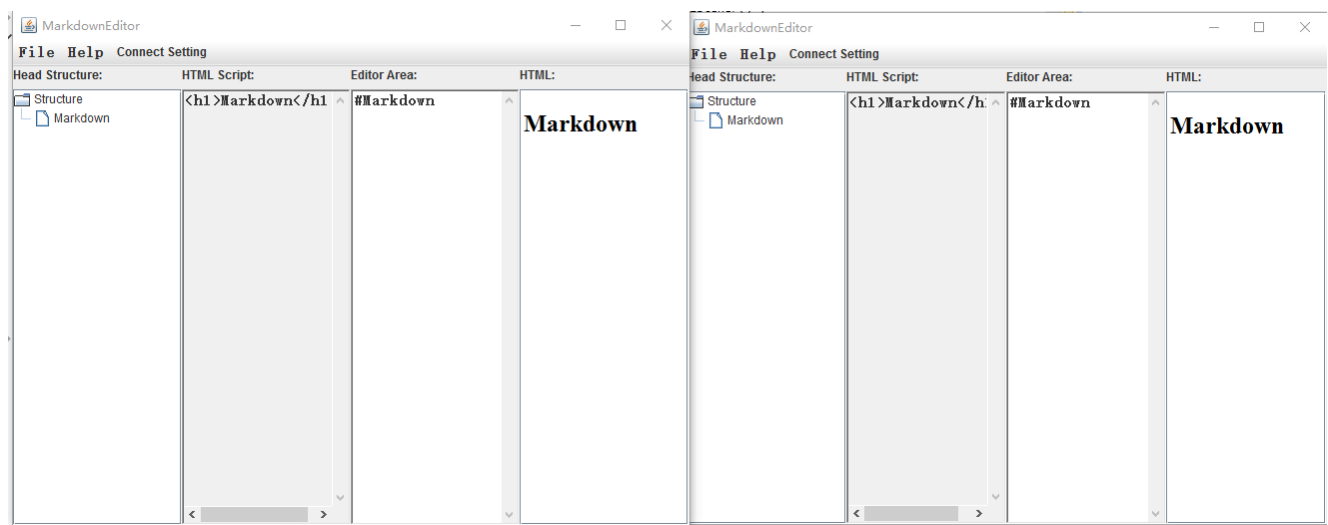
点击 connect setting 里面的 open server，打开服务器连接



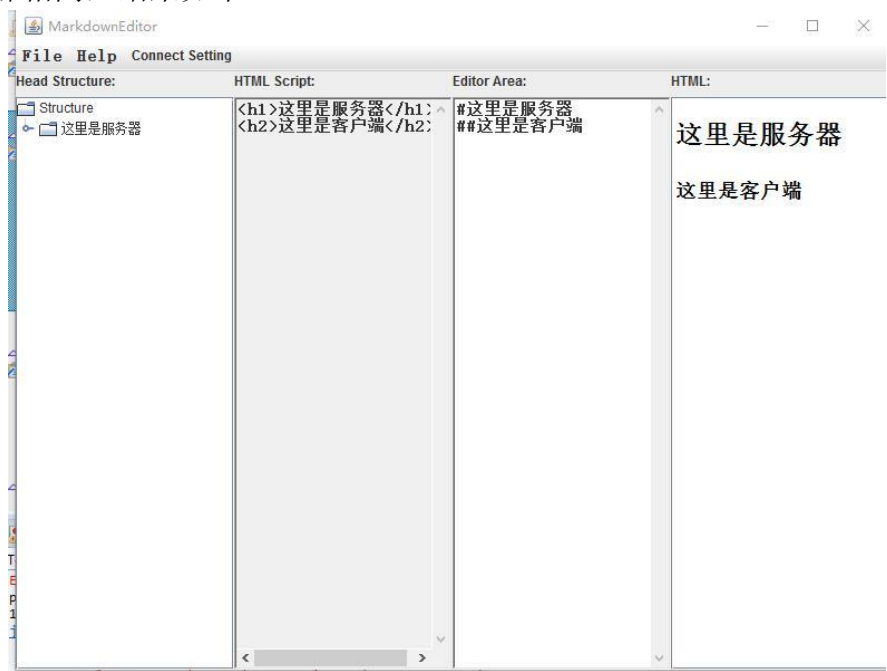
再打开测试的客户端 markdown，同样点击 connect setting 里面的 connect。



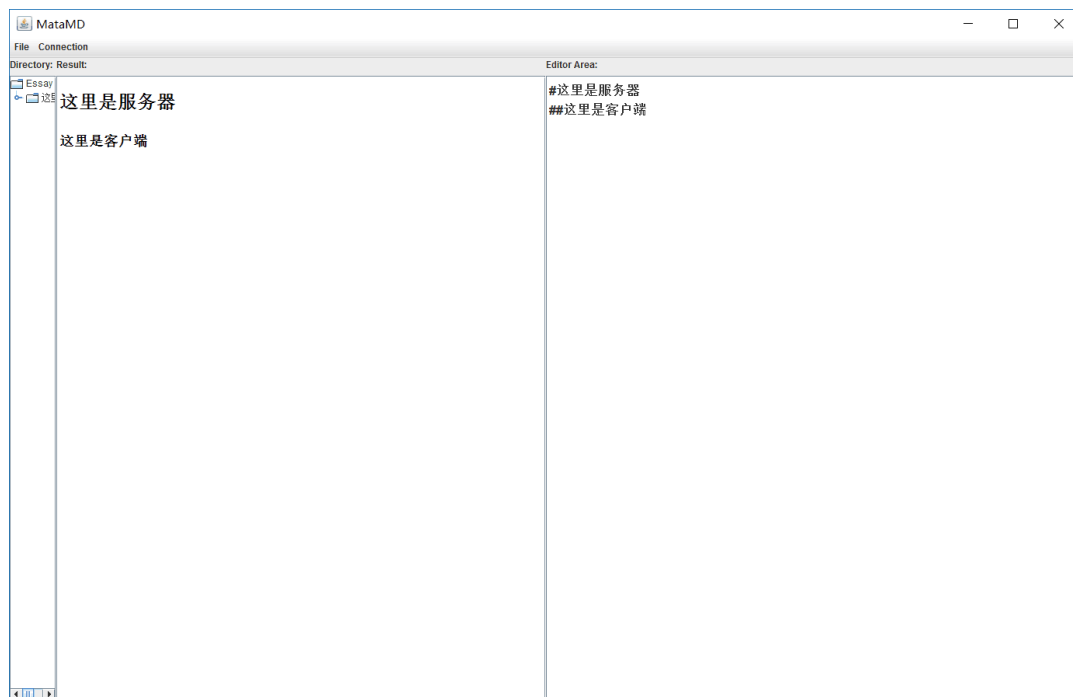
这样，两个 markdown 就实现了连接，这时候，在任意一方改变输入的内容，都能在另一方得到更新，实现实时同步。



与组员电脑连接，只要把连接的 IP 地址改为对方的 IP 地址即可，其余操作步骤相同，结果如下：



我的 markdown



队友的 markdown

## 五、 讨论与心得

这个作业，由于是在上个作业的基础上,只要完成通讯功能即可，因此，在工作量上并不是很大，最主要的是我们要理解 TCP 协议以及 socket 到底是如何实现数据的接受与发送的。另外，就是要搞清楚服务器和客户端之间的关系就好。在做作业的过程中，我遇到的最大的问题就是我一开始是没有给一个独立的线程去创建 socket 的，这样导致的结果就是 GUI 的使用，导致逻辑紊乱，无法达到通讯的效果，在开启了独立线程之后，这一问题也就迎刃而解了。总的来说，通过这次作业，让我对 socket 编程有了更深刻的理解，收获很大。