**CSCW Markdown Editor报告**

概览

[1 服务器端设计 1](#_Toc471086300)

[2 客户端设计 1](#_Toc471086301)

[3 遇到的问题和解决方式 2](#_Toc471086302)

[（1）怎么解决更新编辑域的卡死问题 2](#_Toc471086303)

[（2）怎么解决循环刷新问题 2](#_Toc471086304)

[（3）怎么判断服务器传过来的是md文本更新还是css更新 2](#_Toc471086305)

# 1 服务器端设计

服务器端维护一个线程列表。主线程监听窗口，等待客户端连接。一旦有客人来了，就新起一个线程去处理客户端（在这个程序里面是一直等待接受客户端的消息），并将其加入线程列表。某一个子线程接受到客户端的消息以后，就向除了自己的其他所有线程转发消息。

# 2 客户端设计

客户端监测到文本域的更新，就将更新后的文本域内容发给服务器。

客户端导入自定义css菜单被点击，就将导入的自定义css内容发给服务器。

客户端另有一个子线程等待接收服务器发来的消息，服务器发来消息后就更新编辑域或者用新的css更新预览。

# 3 遇到的问题和解决方式

## （1）怎么解决更新编辑域的卡死问题

一开始我直接在客户端等待接收服务器消息的线程里刷新编辑域，结果卡死了。查了一下发现Java建议耗时工作应当有一个单独的线程去处理，比如刷新编辑域就是一个耗时工作。所以我利用Swing 提供的工具类SwingUtilities另起了一个线程刷新界面。

## （2）怎么解决循环刷新问题

设一个flag来区别三种状态

自己更新的状态：刷新并通知服务器

自己被自己触发更新的状态：不刷新也不通知服务器

自己被别人触发更新的状态：刷新但不通知服务器

## （3）怎么判断服务器传过来的是md文本更新还是css更新

如果传输的内容是css，在传输流前面加上“css”。接收方接受到就判断前三个字符是否是css，如果是，就是css更新（再把这个标记去掉再刷新预览），如果不是，就是md文本更新。

这样有一个弊端就是“css”字符串变成这个软件的保留字符了，设计一下转义字符就能解决，但我懒得了….