

# Java 程序设计实验报告

## Project3

朱理真 3190101094

计算机科学与技术

2021-12-17

## 目的和要求

实现多客户端的纯文本聊天服务器。

1. 能同时接受多个客户端的连接
2. 将一个客户端发送的文本向所有客户端转发（包括发送方）

## 设计结构

### 1. 服务器Server

主要运行run函数。运行过程如下：

1. 根据给定的端口，创建服务器套接字
2. 根据给定的最大同时连接的客户端个数，运行相同个数的Read Task（Read Task创建时接收Server的内部类ServerListener）。
3. 运行一个Send Task
4. 进入accept循环，将接收的客户端套接字放入未处理套接字队列FreeList中并唤醒队列。

### 2. 读取任务Read Task

主要运行run函数。运行过程如下：

1. 如果running变量非真，结束运行
2. 从未处理套接字队列FreeList中取出一个套接字，若队列为空，则等待
3. 运行Listener的read函数
4. 回到1

### 3. 发送任务Send Task

主要运行run函数。运行过程如下：

1. 如果running变量非真，结束运行
2. 从MessageQueue中取出一个Message，以及发送对象的套接字（若Queue为空，等待至被唤醒）
3. 发送Message至该套接字

### 4. 服务器事件处理器Server Listener

主要运行read函数，运行过程如下：

1. 根据传入的客户端套接字，读取一个TalkMessage
2. 若读取中途抛出异常，表示套接字断开，则删除该套接字，结束此函数
3. 将TalkMessage放入MessageQueue中，发送对象是所有的客户端
4. MessageQueue唤醒等待者
5. 回到1

### 5. 消息对象 TalkMessage

继承自HttpMessage(原创，但未使用)，具有与Http包相似的格式

头部字段在内存中以哈希表形式储存，因此可以有可扩展的字段

具有函数read，运行过程如下：

1. 从传入的输入流中不断读取头部至缓冲中，直到读到一个空行
2. 将Http字段加入到Message中
3. 读取content

具有函数send，运行过程如下：

1. 将所有字段输出到传入的套接字中
2. 输出空行
3. 输出content

#### 6. 客户端 Client

构造时，根据传入的IP地址和端口号，尝试连接至服务器

具有函数run，运行过程如下：

1. 运行ReadTask（传入了内部类Client Listener）
2. 进入等待输入循环，若输入为Q，则断开套接字并退出
3. 否则发送TalkMessage

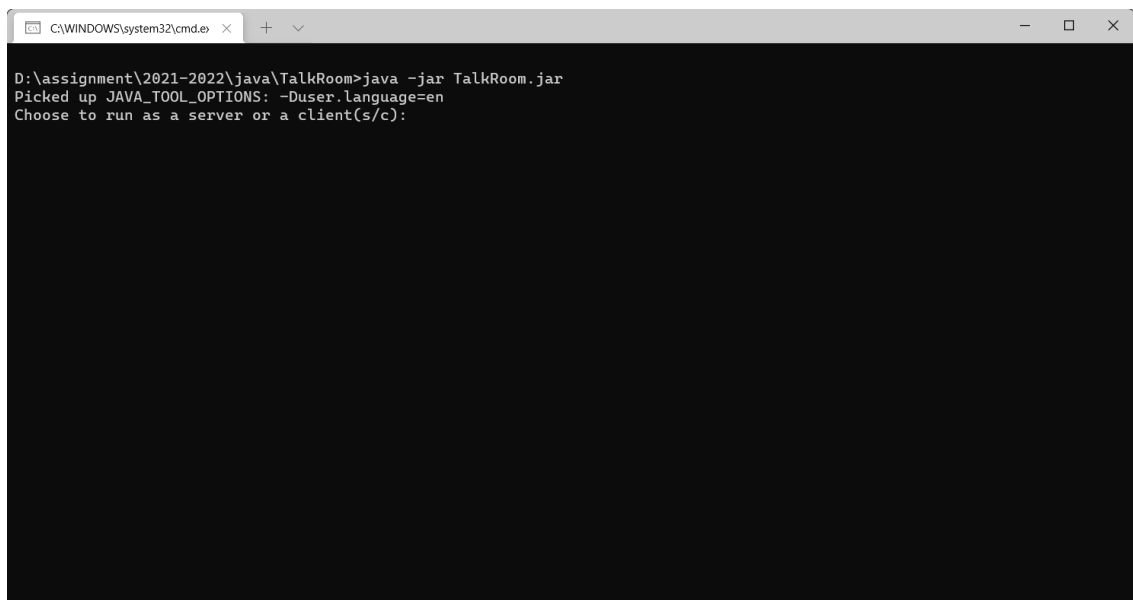
#### 7. 客户端事件处理器 Client Listener

主要具有函数read，运行过程如下：

1. 根据传入的客户端套接字，读取一个TalkMessage
2. 若读取中途抛出异常，表示套接字断开，则删除该套接字，结束此函数
3. 打印TalkMessage的内容到屏幕上
4. 回到1

## 运行展示

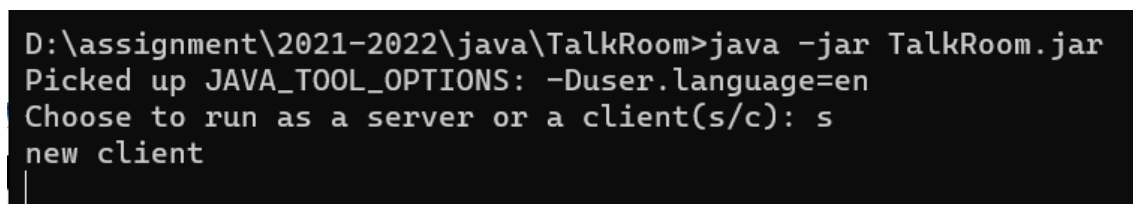
- 开始界面



```
C:\WINDOWS\system32\cmd.exe
D:\assignment\2021-2022\java\TalkRoom>java -jar TalkRoom.jar
Picked up JAVA_TOOL_OPTIONS: -Duser.language=en
Choose to run as a server or a client(s/c):
```

可以根据需要选择运行服务端还是客户端

- 服务端



```
D:\assignment\2021-2022\java\TalkRoom>java -jar TalkRoom.jar
Picked up JAVA_TOOL_OPTIONS: -Duser.language=en
Choose to run as a server or a client(s/c): s
new client
```

选择服务端后即可运行，监听端口默认为1094（学号后4位）

```
D:\assignment\2021-2022\java\TalkRoom>java -jar TalkRoom.jar
Picked up JAVA_TOOL_OPTIONS: -Duser.language=en
Choose to run as a server or a client(s/c): s
new client
socket quit
new client
```

会将套接字的建立和端口信息打在屏幕上

- 客户端

```
D:\assignment\2021-2022\java\TalkRoom>java -jar TalkRoom.jar
Picked up JAVA_TOOL_OPTIONS: -Duser.language=en
Choose to run as a server or a client(s/c): c
put the server IPv4 address: 127.0.0.1
connected
put your words and enter.
enter Q to quit.
```

选择客户端，还需输入IP地址才可连接。

```
hello, world!
$msg: from id=2047329716
$msg: hello, world!
|
```

输入文本后，在本客户端可以看到自己发送的消息，和自己的id

How are you?	\$msg: from id=648129364
\$msg: from id=648129364	\$msg: How are you?
\$msg: How are you?	I'm fine. Thank you.
\$msg: from id=2047329716	\$msg: from id=2047329716
\$msg: I'm fine. Thank you.	\$msg: I'm fine. Thank you.

在多人连接状态下，每个人都可以看到自己和别人发出的消息，及其id号。

```
q
socket closed
```

按下q即可退出聊天

## 问题与总结

- 待解决的问题

- 消息流中含有杂质

在做运行测试时，发现转化的字符串是残缺的。调试发现，服务端接收到客户端的消息发生了变化，其中一个变化表现在content内容前总会有两个多余的字节，且第2个字节总是等于content的长度。仔细检查代码，自己并没有设计发送设计这些信息。虽然知道了出错的规律可以规避，但是无法根除这个问题。

- 待增加的功能

- 升级客户端，使其更具交互性

有以下几种设想：

1. 利用Swing设计窗口客户端程序
2. 设计Http网页

由于已经实现了简单Http后端（这里未给出），第二种方案更具吸引力

- 遇到但已解决的问题

- Client无法退出

开始，Client退出时，并没有主动关闭套接字，导致Read Task仍在另一个线程运行，因此无法结束进程。主动断开套接字并注意唤醒等待的队列即可

- 无法接收两个连续发送消息

调试发现，每个消息在读取时，都会创建一个BufferedInputStream，因此在读取套接字信息时，可能将下一个紧跟的消息也缓存了，但是并不处理。之后别的消息再读取套接字时，由于之前的消息已经被读取，也就接收不到消息了。处理同一个套接字，统一建立一个BufferedInputStream，即可解决问题。

- 总结

Project3有一定的工程量和复杂度，对我来说是一个挑战，也是一个有趣的作业。在实践的过程中，我更加熟悉了Socket库的各种常用类的使用，同时也学习了有关设计模式、消息机制的理念，为日后我参与编写服务器后端项目（如果有）打下了一定的基础。总之，这次Project很有意义。