Project Milstone #2

--- Dongning Li & Haoge Lin

Introduction

This project mimics the iterative server. This server will receive three kinds of messages "Peers?", "PEER:John:PORT=2356:IP=163.118.239.68%", and other messages except "%". This server will consider other messages as the gossip message.

Technology

This project bases on following major technologies:
- JAVA: to build the server
- SQLite: to save the data
- Make: to compile the file
- bash: to run the program and send command

Architecture

Directory Architecture

- root: contains the compile.sh, makefile, read.me, and database.
- lib: contains the external libraries for compile the program
- scripts: contains the .sh file
- src: contains the .java file

Java Architecture

In src directory, there are couples of java file:
- com.cse.np.sc: contains the file coding for main function
  - ClientSocket.java: programs the function on client end
  - MainFunction.java: Mainly control the thread and database
  - TCPServerSocket.java: programs the function about TCP server
  - UDPServerSocket.java: programs the function about UDP server

- com.cse.np.util: contains the file could be used by other file often
  - Constant.java: contains the constant used in the project
  - Database.java: contains the function of database
  - GetOpts.java: contains the method to process the command line
  - ModifyCommand.java: contains the method to process the input message

Database Architecture

- Database Name: NPMil2.db
- Table Name: - MessageReceived
            - PeersCatalog
- MessageReceived: - encryption
                - date
                - message
- PeersCatalog: - name
            - portNumber
            - IPAddress

User Manual

- Unzip the tar.bz2 file
- Open the terminal and input following command:
  - 1. "cd <project directory path>"
  - 2. "make build"
  - 3. "chmod +x compile.sh"
  - 4. "./compile.sh <port number>"
- Open a new terminal and input the following command:
  - 1 For tcp connection : "sock :<port number>"]
    - 1.1 After inputing, input enter to end this command line
  - 2 For udp connection : "sock -u "<port number>"
    - 2.1 continue inputing till system gets 1024 bytes

Result Expect

After compling with makefile:

```
Dongnings-MacBook-Pro:Dongning_Li_Haoge_Li eastaia$ make
javac  -cp .:lib/sqlite-jdbc-3.7.2.jar:lib/java-getopt-1.0.13.jar:bin -d bin -encoding UTF-8 src/com/cse/np/util/Constant.java
javac  -cp .:lib/sqlite-jdbc-3.7.2.jar:lib/java-getopt-1.0.13.jar:bin -d bin -encoding UTF-8 src/com/cse/np/util/DatabaseUtl.java
javac  -cp .:lib/sqlite-jdbc-3.7.2.jar:lib/java-getopt-1.0.13.jar:bin -d bin -encoding UTF-8 src/com/cse/np/util/GetOpts.java
javac  -cp .:lib/sqlite-jdbc-3.7.2.jar:lib/java-getopt-1.0.13.jar:bin -d bin -encoding UTF-8 src/com/cse/np/util/ModifyCommand.java
javac  -cp .:lib/sqlite-jdbc-3.7.2.jar:lib/java-getopt-1.0.13.jar:bin -d bin -encoding UTF-8 src/com/cse/np/sc/ClientSocket.java
javac  -cp .:lib/sqlite-jdbc-3.7.2.jar:lib/java-getopt-1.0.13.jar:bin -d bin -encoding UTF-8 src/com/cse/np/sc/TCPServerSocket.java
javac  -cp .:lib/sqlite-jdbc-3.7.2.jar:lib/java-getopt-1.0.13.jar:bin -d bin -encoding UTF-8 src/com/cse/np/sc/UDPServerSocket.java
javac  -cp .:lib/sqlite-jdbc-3.7.2.jar:lib/java-getopt-1.0.13.jar:bin -d bin -encoding UTF-8 src/com/cse/np/sc/MainFunction.java
```

Make compile.sh executable and run the program:

```
[Dongnings-MacBook-Pro:Dongning_Li_Haoge_Li eastaia$ chmod +x compile.sh
[Dongnings-MacBook-Pro:Dongning_Li_Haoge_Li eastaia$ ./compile.sh 2345
 TCP Server Started!
 UDP Server Started!
```

Open a new terminal and connect it with TCP

```
[Dongnings-MacBook-Pro:~ eastaia$ tools/sock :2345
 Connected to the Server
```

Open a new terminal and connect it with UDP

```
[Dongnings-MacBook-Pro:~ eastaia$ tools/sock -u :2345
```