ID2010 Programming of Interactive Systems

LAB1 Chat-Programs

Ziteng Zhang <ztzhang@kth.se>

27/02/2023

1. Chosen task

In this lab, I tackled the first optional tasks in order to familiarise myself with the software and source code.

- **When the client announces added and removed servers, the output is different from what the user sees in the .list command. Make the announcements present the same text as the .list command**.

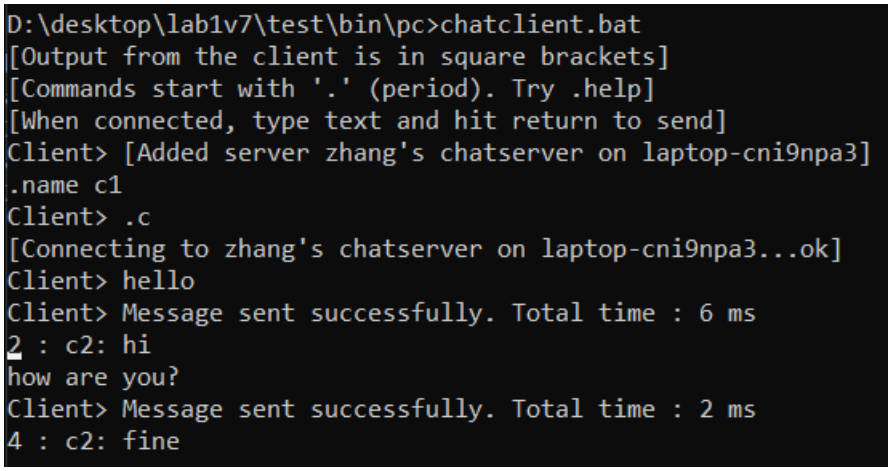
```
D:\desktop\lab1v7\test\bin\pc>chatclient.bat
[Output from the client is in square brackets]
[Commands start with '.' (period). Try .help]
[When connected, type text and hit return to send]
Client> [Added server net.jini.core.lookup.ServiceItem@38ed58d9]
[Added server net.jini.core.lookup.ServiceItem@29f9ce6f]
.list
[zhang's chatserver on laptop-cni9npa3]
[zhang's chatserver on laptop-cni9npa3]
Client>
```

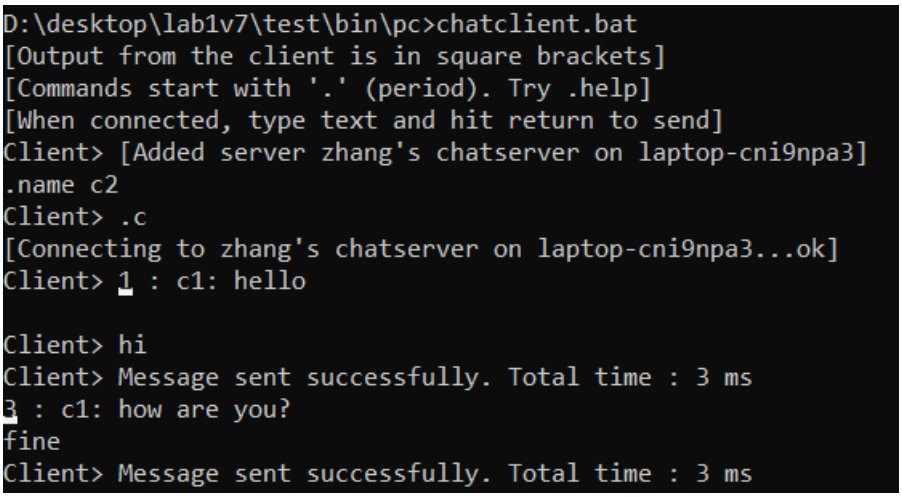
```
D:\desktop\lab1v7\test\bin\pc>chatclient.bat
[Output from the client is in square brackets]
[Commands start with '.' (period). Try .help]
[When connected, type text and hit return to send]
Client> [Added server zhang's chatserver on laptop-cni9npa3]
.list
[zhang's chatserver on laptop-cni9npa3]
Client> [Removed server zhang's chatserver on laptop-cni9npa3]
```

And then, I handled the mandatory task(1).

- **Prevent the client from displaying the text it has sent. The server can do this filtering easily, but consider that the return of a text to the client is a valuable acknowledgement that the server actually delivered the text, and a diagnostic on how long it took to do so. Hint: you cannot rely on the client's user name string, as this can be set to anything on any client.**

```
D:\desktop\lab1v7\test\bin\pc>chatserver.bat
Server zhang's chatserver on laptop-cni9npa3 started.
Server> Registered as a Jini service 26996f8a-c3f1-483b-b1f8-de7bc2f9c833
Added client : Proxy[RemoteEventListener,RemoteObjectInvocationHandler[UnicastRef [liveRef: [endpoint:[169.254.53.124:53
278](remote),objID:[-39e73541:185ff4e3a9d:-7fff, -8365164646776521166]]]]]
Added client : Proxy[RemoteEventListener,RemoteObjectInvocationHandler[UnicastRef [liveRef: [endpoint:[169.254.53.124:53
292](remote),objID:[6fb0a3eb:185ff4e63df:-7fff, -8056690128861082550]]]]]
MSG#1:c1: hello
MSG#2:c2: hi
MSG#3:c1: how are you?
MSG#4:c2: fine
```





2. Design and Implementation

- **For the first optional task, I changed the method serviceAdded and serviceRemoved. When the client announces added and removed servers, it doesn't use the String of ServiceItem but remoteName, which is used for .list. In order to do this, I added the following code.**

```
public void serviceAdded (ServiceDiscoveryEvent e) {
  ServiceItem sit = e.getPostEventServiceItem ();
  if (sit.service instanceof ChatServerInterface) {
    servers.add (sit);
    serverIDs.put (sit.serviceID, sit);
    ChatServerInterface chat = (ChatServerInterface) sit.service;
    String remoteName = null;
      try {
          remoteName = chat.getName ();
          System.out.println ("[Added server " + remoteName + "]");
      } catch (RemoteException rex) {
          System.out.println ("[Added server " + sit.toString() + "]");
      }

  }
}
public void serviceRemoved (ServiceDiscoveryEvent e) {
    ServiceItem sit = e.getPreEventServiceItem ();
    if (sit.service instanceof ChatServerInterface) {
      servers.remove (serverIDs.get (sit.serviceID));
      ChatServerInterface chat = (ChatServerInterface) sit.service;
```

```
        String remoteName = null;
        try {
            remoteName = chat.getName ();
            System.out.println ("[Removed server " + remoteName + "]");
        } catch (RemoteException rex) {
            System.out.println ("[Removed server " + sit.toString() + "]");
        }
      }
    }
```

- **For the mandatory task(1), firstly, I achieved preventing the client from displaying the text it has sent.**

  I added a list to store msgSender in the class ChatServer.

```
/**
 * Incoming message senders are placed on the message sender queue.
 * The distribution thread consumes the queue by sending message to
 * message sender. Class LinkedList is not thread-safe, so access to
 * it must be synchronized.
 */
protected LinkedList<String> msgClients = new LinkedList<String> ();
```

And I changed the method say in class ChatServer. (Use rel to record msgSender)

```
public void say (String msg,  RemoteEventListener rel) throws RemoteException
{
  if (msg != null) {
    synchronized (msgClients){
        msgClients.addLast(rel.toString());
    }
    addMessage (msg);
  }
}
```

Then the server delivered the text to clients seperately. (The way the client is handled will also be adjusted)

```
public void run () {

  while (runDelivery) {

  String msg = getNextMessage ();
  if (msg != null) {
  // Prepare notifications
  ChatNotification note = new ChatNotification ( source: this, msg, msgCount, id: 0);
  ChatNotification note0 = new ChatNotification ( source: this, msg, msgCount, id: 1);
  // Send them to all registered listeners.
  synchronized (clients) {
    try {
      String msgClient="";
      synchronized (msgClients){
        msgClient=msgClients.removeFirst();
      }
      for (RemoteEventListener rel : clients){
          if(rel.toString().equals(msgClient)){
              rel.notify(note0);
          }
          else{
              rel.notify (note);
          }
      }

    }
```

Last, I recorded the time when the client sent the message, and I also recorded the time when it received the message returned by the server, so that the running time was obtained.

```
    System.out.println ("[" + verb + ": unknown command]");
  }
    }
    else if (0 < arg.length ()) {
    synchronized (time) {
        Long startTime = System.currentTimeMillis();
        time.addLast(startTime);
    }
    sendToChat ( text: myName + ": " + arg);
    }
```

```
        chat.getText ());
  else{
      synchronized (time){
          Long startTime=time.removeFirst();
          Long endTime=System.currentTimeMillis();
          System.out.println("Message sent successfully. Total time : "+ (endTime-startTime)+" ms");
      }
  }
}
```