

# 简易聊天软件

---

## 作业一

---

### 实现功能

1. 客户端和服务端互发消息

### 不足之处

1. 如果直接关闭会话窗口的话会使另一端陷入死循环。
2. 界面不是非常友好。
3. 虽然可以有多个客户端同时给服务器端发消息，但是服务器端只能给一个客户端发消息。

### 程序设计

#### 服务器端

1. 有个线程数组，存放连接的接收消息线程和发送消息线程。
2. 接收消息线程和发送消息线程同时工作。

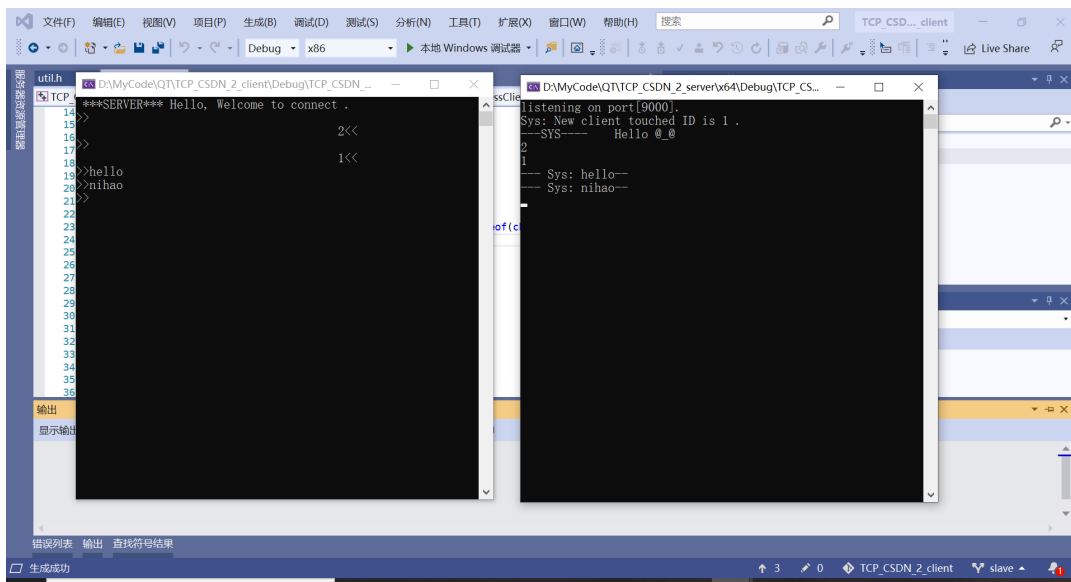
#### 客户端

1. 接收消息线程和发送消息线程同时工作。

### 学习经验

1. 要有两个线程，一个管理接收消息，一个管理发送消息。

### 运行截图



## 作业二

### 实现功能

1. 用昵称和密码注册，昵称可以相同，用id和密码登入。
2. 如果通信的id在线的话，会直接显示给对方，否则插入数据库，等待对方上线后发送给对方。
3. 可以昵称相同，但是要用id登入。

### 不足之处

1. 消息中不能有空格，有空格的话会按多个消息传送
2. 1给2发消息时，2必须在选择和2通信之前登入才算在线，在选择之后登入不算在线。（也就是做不到在线人数和状态时时刷新）。
3. 服务器端最多可以在线人数由MAXCLIENTS控制，但是客户端退出之后，服务器端依然算作在线。

### 程序设计

#### 服务器端

1. MySQLManage.h文件是封装了和数据库通信的一些原语。
2. util.h文件是一些处理格式转换的函数封装，比如：
  1. 将字符串分割为vector再封装成map。
  2. 得到当前时间 YY-MM-DD HH:MM:SS形式以方便插入数据库。
  3. char \* 和 string的相互转换

4. 字符串前后加单引号（数据库更新插入时字符串要被单引号包裹）
3. 服务器端**只有接受消息线程**。
4. 客户端和服务端的消息分为三种：
  1. init\_mess=1，初始化消息，用于标识登入注册和正常发送消息，而登入和注册的分别是用关键字nick\_name，没有nick\_name的消息是登陆，否则是注册。
  2. init\_mess=2，通信消息，正常通信，通信内容有：from\_id, to\_id, time, content。
  3. init\_mess=3，表示要接受接受的离线消息。

## 客户端

1. /与客户端-服务端不同（发送消息线程和接受消息线程同时开启），客户端-客户端通信的接受消息线程在控制信息初始化之后再开启。其实也可以用信号量，当初始化完成后增加一个信号量，之前的话接受消息线程wait即可。
2. 先选择登入还是注册，注册完成后也要登入。
3. 登入完成后输入通信的to\_id。之后先接受离线消息，再进行在线通信。
4. 客户端有两个线程，一个发送消息线程，一个接收消息线程。

## 学习经验

## 运行截图

