

“Linux Chat”项目报告

李海龙、吴杰、于浩淼、张登凯

2019 年 7 月 7 日

项目地址: <https://github.com/bitlinux/qtchat>

目录

“LINUX CHAT”项目报告	1
一、实验环境与分工	3
1.实验环境	3
2.成员分工	3
二、设计方案	4
1.服务端模块	5
2.客户端模块	5
3.服务端与客户端通信	6
三、特色与创新	6
四、问题与解决方案	7
五、功能介绍	9
5.1.注册功能	9
5.2.忘记密码及重置密码	10
5.3.登录	10
5.4.添加好友	11
5.5.群聊天	13
5.6.修改昵称	15
5.7.私聊	15
5.8.BBS	17
六、附录	20
6.1.数据库表	20
6.2.接口定义表	23

一、实验环境与分工

1.实验环境

软件平台：Linux

用户界面：QT

编程语言：C++

数据库：MySQL

2.成员分工

吴杰

- 27/5/19
需求分析和任务分配
- 3/6/19
实现主界面基本图形框架(好友列表待实现)
- 8/6/19
实现好友列表中的frienditem，能够循环加入item到列表中（待实现从服务端获取数据和打开聊天窗口的界面）
- 9/6/19
修复bug，新增修改昵称dialog、上传头像文件选取、item到窗口的链接
- 3/7/19
实现好友列表、在线状态、好友请求列表实时刷新，完善私聊窗口文件传送
- 4/7/19
增加BBS模块界面，编写bbsitem，新建帖子界面 待实现：帖子详情界面，在帖子详情界面实现回复和删除帖子（通过发帖用户id与自己id进行比较，来确定是否显示删除按钮），需要新增回复窗口或者使用点击用户昵称传递参数的功能；BBS的文件上传和下载模块的连接
- 5/7/19
实现BBS帖子详情界面，实现帖子回复，帖子删除，BBS中帖子item的实时刷新功能，点击查看帖子详情的时候从服务器加载帖子下的所有回复
- 6/7/19
界面美化：FriendItems和BBSItems之间绘制1px灰色分割线，Login界面添加Logo，文字布局/颜色调整；软件功能测试
- 7/7/19
功能测试，项目文件清理，部分文档撰写

李海龙

- 27/5/19
Qt5 GUI开发框架使用
- 1/7/19
设计实现BBS的文件分享模块的界面及功能，包括文件的上传，下载，显示 BBS文件分享服务器搭建、维护
- 3/7/19
设计实现BBS新建帖子操作的界面、功能
- 7/7/19
功能测试，部分项目文档撰写

于浩淼

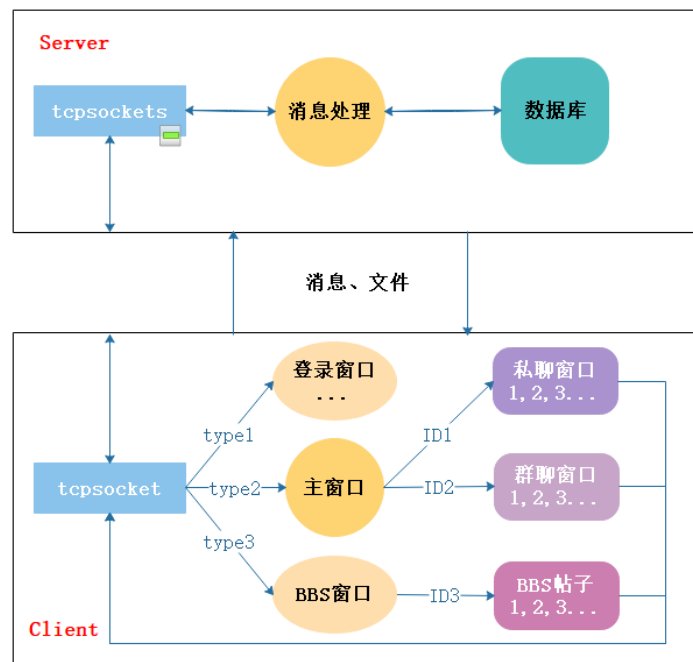
- 27/5/19
数据库表设计
- 3/6/19
设计实现私聊、群聊 GUI
- 5/6/19
设计实现登录、注册、忘记密码、添加好友、创建群聊GUI
- 10/6/19
设计服务端与客户端、客户端与客户端的通信接口
- 20/6/19
实现GUI界面的跳转，客户端与服务端的连接，实现注册和忘记密码的功能
- 25/6/19
实现登录、添加好友、创建群聊等功能
- 1/7/19
完善了客户端的通信架构，实现了私聊、私聊消息记录功能
- 3/7/19
实现了群聊、添加群成员、修改群名等功能
- 7/7/19
部分项目文档编写

张登凯

- 27/5/19
Mysql数据库使用方法调查
- 25/6/19
重新构建服务器，建立数据库，并且使服务器与数据库成功连接
- 26/6/19
实现服务器的登录和注册功能，解决网络环境下的通信问题
- 1/7/19
实现服务器与客户端之间的主界面通信，搭建完成通信架构，实现私聊，看历史消息，建立群聊，更改群名，添加好友，更改昵称的功能
- 2/7/19
完成了发送文件，刷新页面的功能
- 3/7/19
完成了BBS，发帖回帖删帖的功能
- 5/7/19
完成BBS文件发送的功能，辅助客户端进行测试
- 6/7/19
完成不同操作系统下服务器的移植
- 7/7/19
部分文档的编写

二、设计方案

采用的设计方法为模块设计法。程序主要分为服务端和客户端两个大模块，模块之间使用 TCP 协议进行通信，具体的程序架构图如下：



1.服务端模块

服务端接收客户端的连接，每一个客户端对应一个 `tcpsocket`。服务端通过维护一个 `tcpsocket` 列表，来记录所有在线的用户。

`Tcpsocket` 负责接收和发送消息。当 `tcpsocket` 收到消息后，服务端将根据消息类型进行相应处理，增删改查数据库，并向用户发送消息处理的结果。

2.客户端模块

每个客户端都创建专属于自己的 `tcpsocket`，用于发送和接收服务端发来的消息。客户端接收消息的过程可以分为三层，第一层为 `tcpsocket`，第二层为主界面、BBS 界面以及登录等界面，第三层为群聊、私聊、帖子等界面。

第一层向第二层发送消息：当 `tcpsocket` 收到消息后，将首先读入消息的类型 `type`，并将 `type` 作为信号发射出去。接收 `type` 信号的槽函数存在于登录窗口、注册窗口、主窗口以及 BBS 等窗口之中，当 `type` 信号被发射，这些窗口的槽函数便会依次执行，每个槽函数都会通过 `switch` 语句来判断接收的信号 `type` 类型是否是自己要处理的类型，如果不是，则不执行任何操作；如果是，则进行相应的处理。

第二层向第三层发送消息：一个主窗口对应多个私聊和群聊窗口，每个聊天窗口需要只显示发送给它自己的消息。因此，在主窗口接收 `type` 信号的槽函数中，需要再从 `tcpsocket` 中读入私聊对方 ID 或者群聊 ID，然后再将 ID 值作为信号发射出去。接收 ID 信号的槽函数存在于私聊和群聊窗口中，当 ID 信号被发射，这些窗口的槽函数也会被依次执行。每个槽函数在函数开头时判断接收的 ID

值是否是自己的，如果不是，则不执行任何操作；如果是，则读入消息，将消息显示在聊天窗口上。由 BBS 窗口向帖子传递消息的方式与此一致。

在客户端，所有窗口都可以直接向 `tcpsocket` 中写入消息，并发送至服务端。

3.服务端与客户端通信

服务端与客户端按照提前约定好的协议进行通信。协议规定了可以处理的消息类型，以及处理该类型消息所需的参数。发送或接收的消息都必须以 `type` 类型作为开头，便于两端进行相应的处理。

三、特色与创新

特色：

1. 安全性

目前市场上的大多数聊天软件都是采用 UDP 进行消息的传输，而我们采用的是 TCP 协议，相对而言更加安全，比较适合小型的即时通信软件。

2. 实用性

我们所设计的软件实现了注册、登录、忘记密码等功能，便于用户管理自身账号；也提供了添加好友、私人聊天、创建群聊、多人聊天、查看聊天记录等功能，不仅支持普通消息的收发，也支持文件的发送与接收，便于用户群体的沟通交流；修改昵称、修改群名以及设置头像等实用的小功能，使用户能够充分展示自身个性；程序还实现了 BBS 板块，用户可以自由发帖回帖，在文件共享区上传下载文件，享受 BBS 社区的丰富和便捷。

3. 跨平台性

本程序由 Qt 开发，界面简洁大方，具备跨平台性。一次编写，多次编译，各种系统，均可运行。因此，程序的开发更为方便，程序的可扩展性较高，后续的改进完善较为省力。

创新：

客户端的三层消息接收架构

由于服务端需要通过 `tcpsocket` 是否连接来判断用户的登录状态，因此，用户在登录之后，不能再断开 `tcpsocket` 的连接。所以，用户在登录之后，所有的消息，都将由同一个 `tcpsocket` 接收和发送。

在 Qt 中，使用信号与槽的方式来发送和接收消息。一个信号可以绑定多个槽，当接收到一个信号后，绑定的多个槽会顺序执行。这也就意味着，当收到数据流到达的信号时，即使数据流中只有一个 `type` 值，如果每个槽函数中都从数

据流中读取数据作为自己的 `type` 值，那么只有第一个执行的槽函数读取的值是正确的，接下来执行的槽函数读取的值都是错误的。所以，这就需要我们设计一种客户端消息发送的机制，避免多个槽函数错误的多次读取同一类型数值的情况。

于是，我们根据自己对于 Qt 信号与槽机制的理解。设计了客户端的三层消息发送架构：

第一层是 `tcpsocket`，从数据流中读取 `type` 值，并通过 `send_type` 信号发送；第二层包括登录及登录之前的窗口（注册窗口、忘记密码窗口），以及主窗口和 BBS 窗口。这些窗口都接收 `send_type` 信号，读取 `type` 值，通过 `switch` 语句判断该类型是否由自己处理，如果不是，则不执行任何操作；如果是，则进行处理。同时，主窗口和 BBS 窗口还会从数据流中读入私聊或群聊或帖子的 ID 号，并通过 `send_ID` 等信号发送；

第三层包括私聊、群聊、帖子窗口。这些窗口都接收 `send_ID` 信号，读入 ID 值，通过 `if` 语句来判断此 ID 是否与自己的 ID 一致。如果不一致，则不执行任何操作；如果一致，则继续读入消息，并将其显示在聊天窗口上。

客户端的三层消息接收架构妥善的解决了多个槽函数顺序读取带来的问题，是我们思考讨论的成果，使得我们对于 Qt 中信号与槽的机制有了更深刻的认识。

四、问题与解决方案

1. 同时打开多个聊天窗口，只有位于好友列表第一个的窗口能够正确的接收消息。

原因：绑定同一个信号的多个槽函数会顺序执行。假设要接收消息的聊天窗口位于好友列表中第二的位置，消息的格式为发送者 ID、发送时间、发送消息内容。当消息到来时，位于好友列表第一的消息窗口将会首先读入发送者 ID，判断发送者 ID 与自己的 ID 是否一致，发现不一致后返回。接下来，位于好友列表第二的消息窗口也想要读入发送者 ID，但是发送者 ID 已经被取出了，它实际读入的内容是发送时间，发送时间肯定与该消息窗口的 ID 值不一致，因此该窗口直接返回，无法显示消息。

解决：将客户端的接收流程改为上文提到的三层接收架构。

2. 信号与槽的关联问题，信号由主窗口发送，槽函数在私聊窗口中，应当如何将信号与槽函数关联起来？

解决：主窗口中包含着多个 `friend_item`，每个 `friend_item` 对应着一个私聊窗口。在 `friend_item` 建立时，主窗口可以通过构造函数，将自己的指针作为 `parent` 参数传递给 `friend_item`。同时 `friend_item` 可以建立一个私聊窗口指针对象，并为它分配内存。这样，在 `friend_item` 的构造函数中，可以获得主窗口和私聊窗口的指针，就可以将主窗口发射的信号与私聊窗口中的槽函数成功关联起来。

3. 在通信时，如何使用 QDataStream 传递自己设计的数据结构？

解决：通过重载操作符，可以使用 QDataStream 传递自己设计的数据结构和相应的 QVector 变量。如果是数据结构，在重载 out 操作符时，需要加上 const；如果是 QVector，则都不需要加上 const。

4. 删除帖子后，再新建帖子，数据库报错重复主键。

原因：每个帖子都作为一条记录存在数据表中，记录的主键就是帖子建立的序号，由 count(*) 函数生成。如果目前数据表中有 10 条记录，那么下一个插入的记录的主键就是 11。这种主键的生成方法在删除记录后就会出现。假如之前一共 11 条记录，序号分别为 1 到 11，在删除其中某一条后，数据表中剩下了 10 条记录，此时插入新的记录，通过 count(*) 生成的主键值为 10+1=11，与当前表中的第 10 条序号为 11 的记录发送冲突。

解决：使用 max() 函数代替 count() 函数，对于上述情况，max 函数返回的值为 11，因此新记录的主键将被设置为 11+1=12，不会发生冲突。

5. 当关闭子窗口的时候，主窗口也同时被关闭。

原因：子窗口的构造函数中存在父窗口作为 parent 组件参数，导致在析构函数调用的时候将父窗口也析构了。

解决：将指向父窗口的指针置为空指针，在析构子窗口的时候就不会析构父窗口。

6. 各种实时刷新逻辑，比如加好友之后对方无法实时显示好友请求的通知；好友的在线状态更新，BBS 帖子的实时刷新。

原因：一开始编码过程当中，仅仅在初始化界面的过程当中对这些数据进行了请求，没有对数据进行实时更新。

解决：好友请求发出、好友上线、帖子的删除和新增，服务器不仅写入数据库，还转发消息给客户端，客户端监听对应的消息类型进行处理，从而实现实时更新。

7. 聊天消息和文件传送功能的延迟，当传送文件比较大的时候，接收方的延迟比较大。

原因：项目采用了 TCP Socket 进行通信，同时，消息和文件都使用了服务器转发机制，文件和消息都先发送给服务器并存储，再通过服务器转发给接收者。

解决：

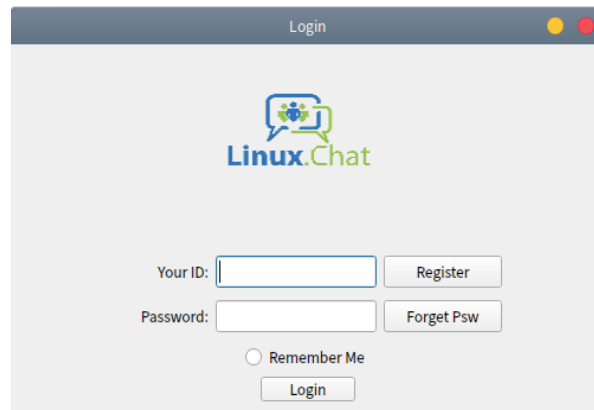
a. 考虑采用 UDP 进行消息通信，文件传送采用 TCP 直接互传，不通过服务器转发，服务器仅仅存储文件传送的记录。

b. 客户端之间的通信使用点对点 TCP，发送消息给好友的同时也发送消息给服务器，便于服务器记录消息历史记录，同时也减少了延迟。

五、功能介绍

Linux Chat 包含即时通讯和 BBS 两个部分，即时通讯部分实现了注册、登录、忘记密码、修改昵称、群聊和私聊、添加好友、添加群、通讯内容备份、文件传输等功能，BBS 部分实现了发帖、回帖、删帖、上传下载文件等功能。

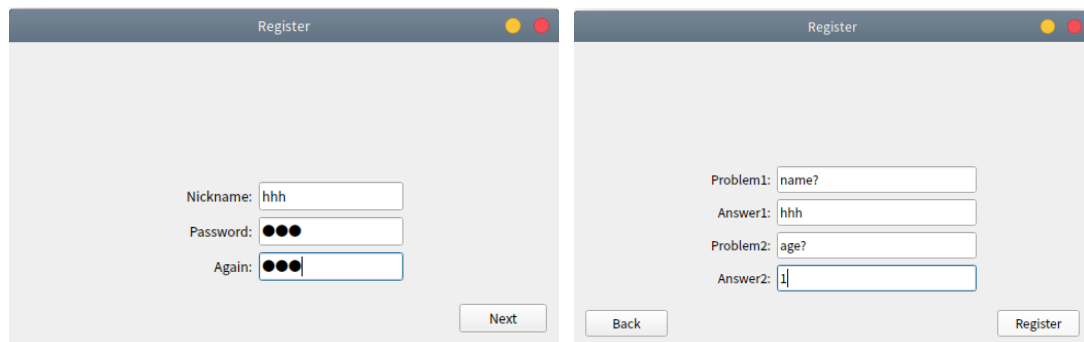
接下来将结合程序不同界面，介绍功能。首先，运行程序，进入登录界面：



The image shows the 'Login' window of the Linux Chat application. It features the 'Linux.Chat' logo at the top. Below the logo, there are two input fields: 'Your ID:' and 'Password:'. To the right of the 'Your ID:' field is a 'Register' button. To the right of the 'Password:' field is a 'Forget Psw' button. Below these fields, there is a radio button labeled 'Remember Me' and a 'Login' button at the bottom.

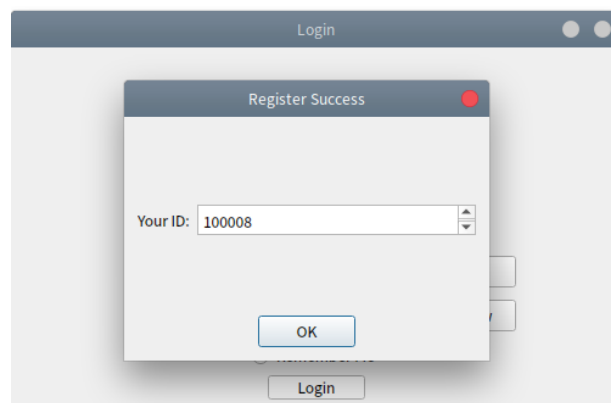
5.1.注册功能

点击 Register 进入注册界面，输入昵称，密码，确认密码，点击 Next，进入密保信息填写界面：



The image shows two side-by-side screenshots of the 'Register' window. The left screenshot shows the initial registration form with fields for 'Nickname:' (containing 'hhh'), 'Password:' (masked with dots), and 'Again:' (masked with dots). There is a 'Next' button at the bottom right. The right screenshot shows the security question setup screen with fields for 'Problem1:' (containing 'name?'), 'Answer1:' (containing 'hhh'), 'Problem2:' (containing 'age?'), and 'Answer2:' (containing '1'). There are 'Back' and 'Register' buttons at the bottom.

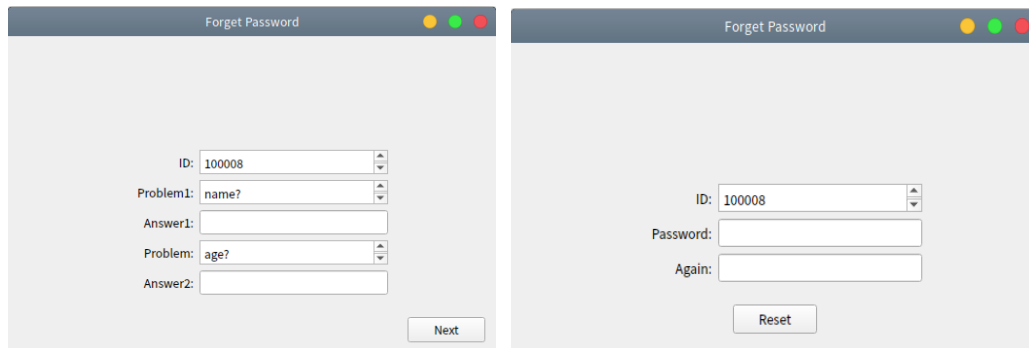
注册成功，弹出注册成功对话框，返回注册成功的用户 ID：



The image shows the 'Login' window with a 'Register Success' dialog box overlaid on top. The dialog box has a title bar that says 'Register Success'. Inside the dialog, there is a 'Your ID:' label followed by a text box containing the value '100008'. At the bottom of the dialog is an 'OK' button. The 'Login' button from the background window is still visible at the bottom.

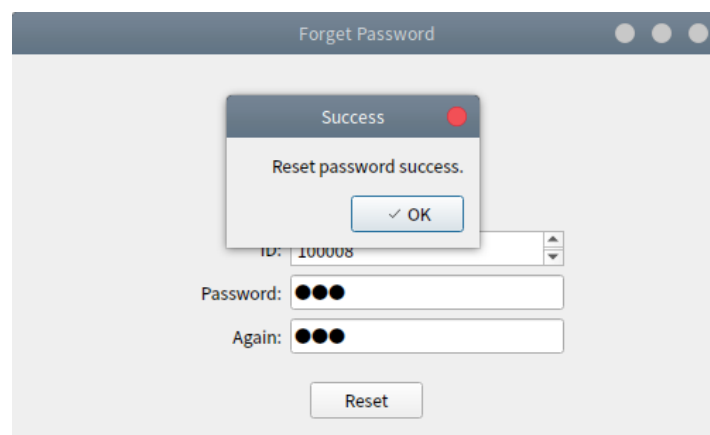
5.2.忘记密码及重置密码

在登录界面点击 **Forget Password**，进入密保信息验证界面，填写密保信息后，点击 **Next**，若密保问题回答正确，进入密码重置页面，若错误则弹出错误提示框：



The image shows two screenshots of a 'Forget Password' window. The left screenshot displays the security question verification interface. It includes a dropdown for 'ID' with the value '100008', a dropdown for 'Problem1' with the value 'name?', a text input for 'Answer1', a dropdown for 'Problem2' with the value 'age?', and a text input for 'Answer2'. A 'Next' button is at the bottom right. The right screenshot displays the password reset interface. It includes a dropdown for 'ID' with the value '100008', a text input for 'Password', and a text input for 'Again'. A 'Reset' button is at the bottom right.

密码重置界面，输入用户 ID 的新密码和确认密码。点击 **Reset** 重置密码，密码重置成功，弹出密码重置成功提示框：



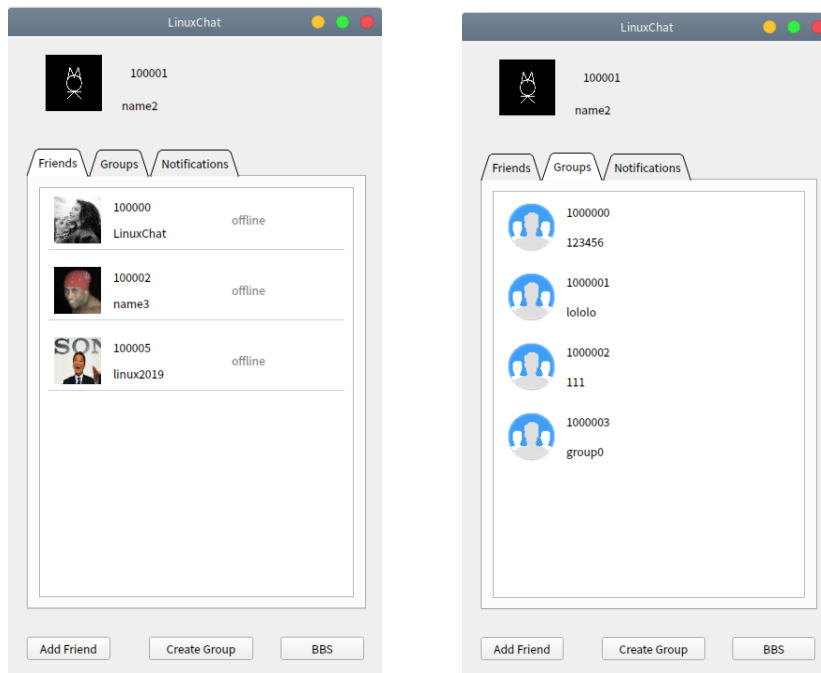
The image shows a screenshot of the 'Forget Password' window with a success dialog box overlaid. The dialog box has a title 'Success' and a message 'Reset password success.' with an 'OK' button. The background window shows the ID (100008), Password, and Again fields, with the Password and Again fields masked with dots. A 'Reset' button is at the bottom.

5.3.登录

在登录界面，输入用户 ID 和密码，点击 **Login**，若信息正确，进入聊天软件主界面，若错误则弹出错误提示框，若连接超时，则提示连接超时无法连接至服务器。主界面包含好友列表，群组列表，消息通知共三个部分。

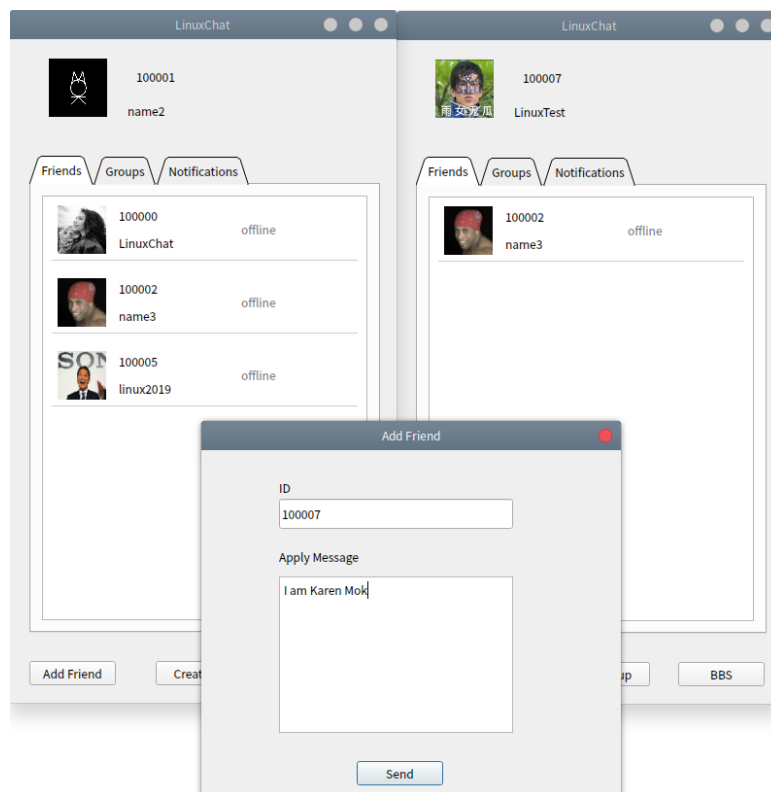
- 3.1 好友列表：好友在线状态 **online/offline** 实时刷新。
- 3.2 群组列表：显示当前所在的所有群聊 ID 和群名称。
- 3.3 通知列表：显示好友请求。

以下分别为好友列表和群聊列表的界面：

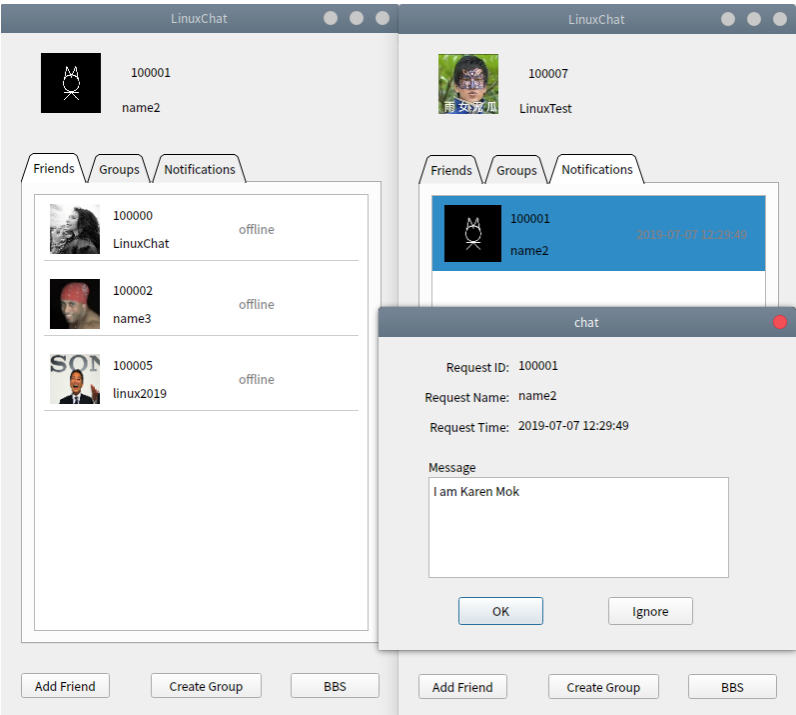


5.4.添加好友

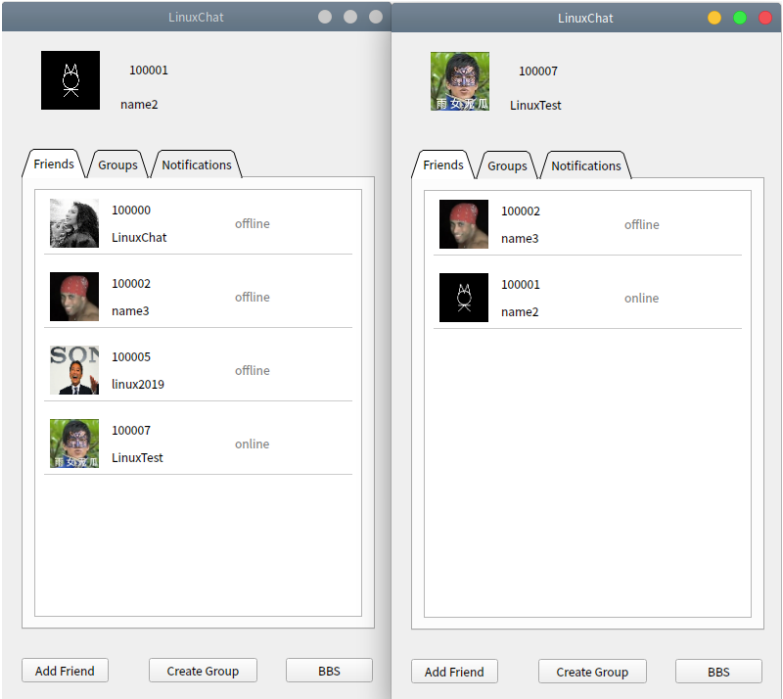
点击 Add Friend 按钮进入好友申请界面，输入要添加的好友的 ID 和验证消息，点击 Send 发送申请：



接收方的通知列表中实时显示新好友请求，点击即可查看好友请求详情，包括请求时间，对方的 ID 和昵称，以及验证消息：



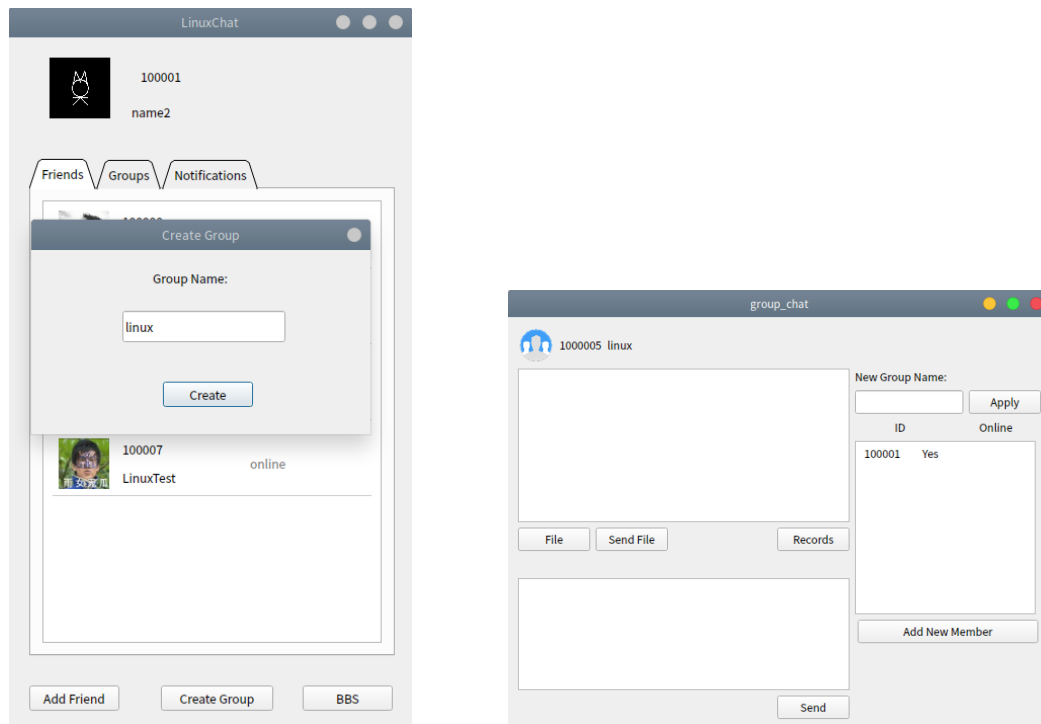
点击 Ignore 忽略好友请求，点击 OK 同意好友请求，添加好友成功，好友被添加到各自的好友列表中：



5.5.群聊天

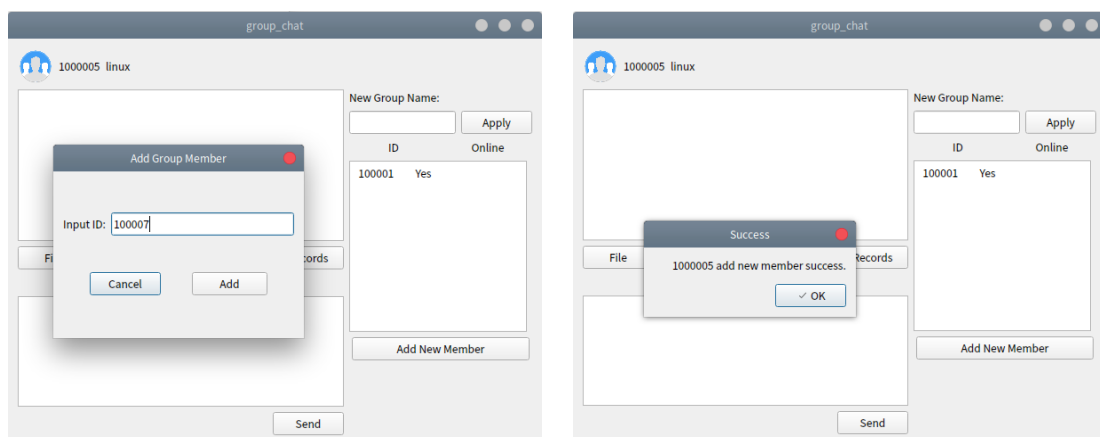
5.5.1 创建群聊

从主界面点击 **Create Group**，弹出创建群聊窗口，输入群聊名称，即可创建群聊(下左图)：

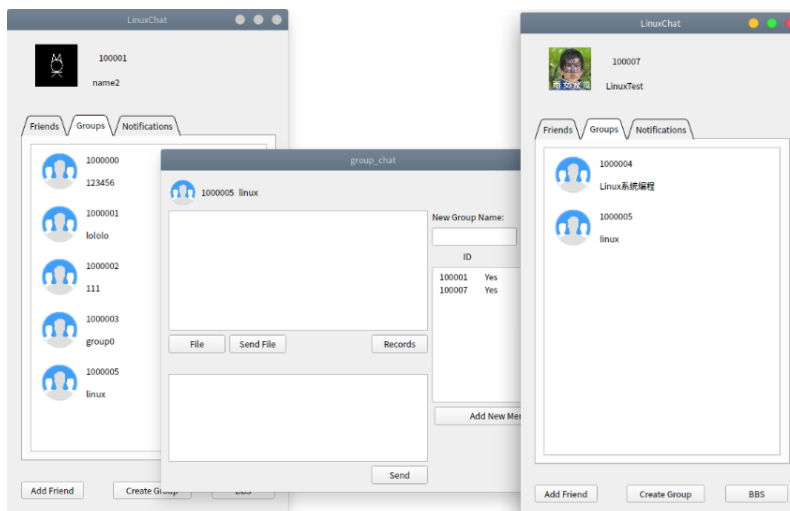


5.5.2 添加群成员

从群聊列表中点击新创建的群聊，进入群聊窗口，此时只有创建者一个用户，点击 **Add New Member**，提示输入要加入的群成员 ID，输入之后，添加新成员进入群聊：

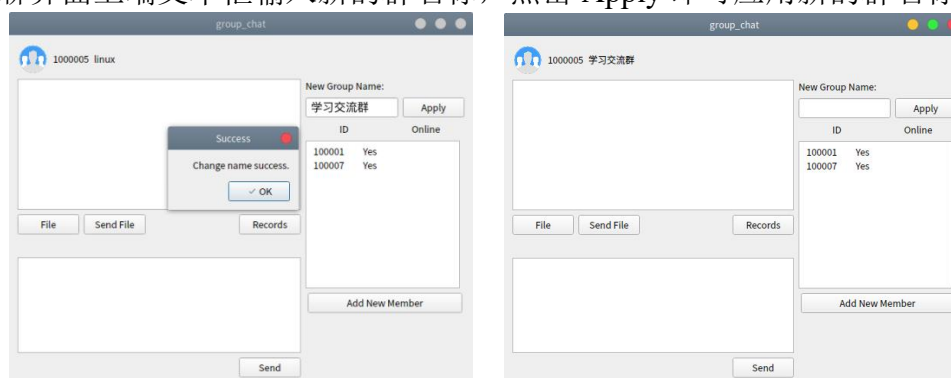


点击 Add，等待服务器返回成功消息，即添加成功，添加成功后，可以看到两个人都进入了群聊，同时群聊窗口的群成员列表刷新：



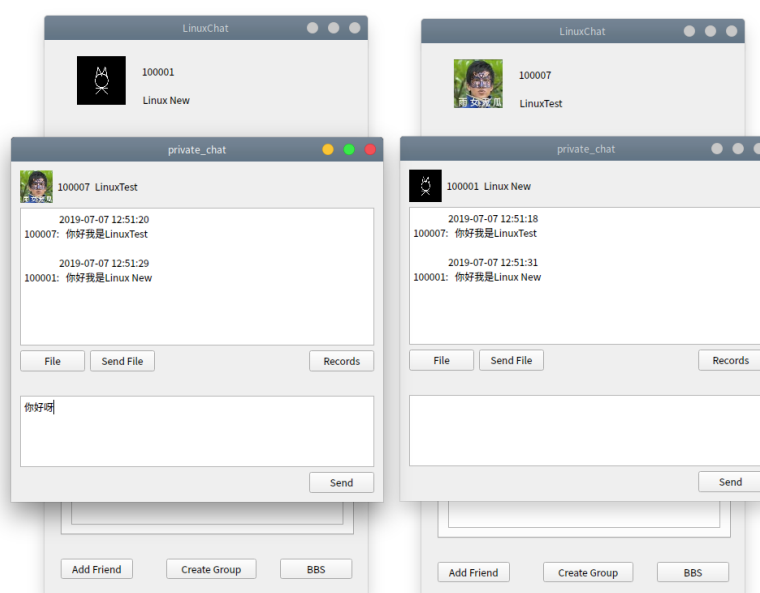
5.5.3 更改群名称

从群聊界面上端文本框输入新的群名称，点击 Apply 即可应用新的群名称：



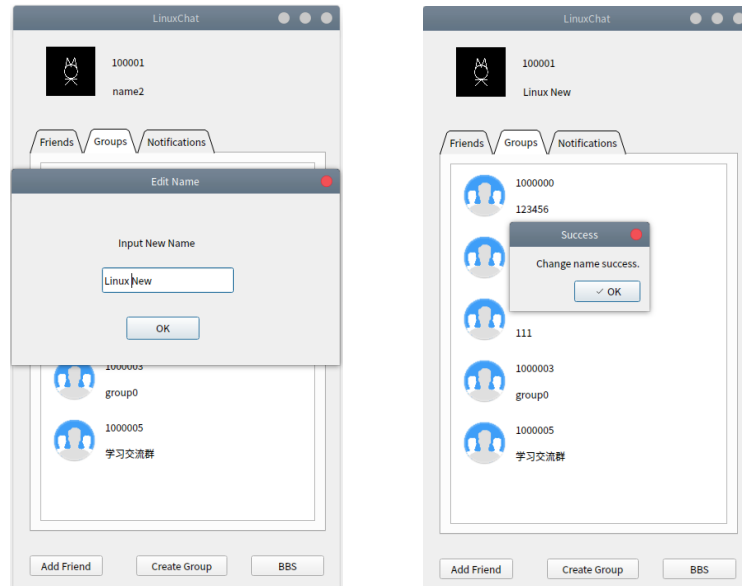
5.5.4 群聊天

在编辑文本框编辑消息点击 Send 即可发送一个群消息：



5.6.修改昵称

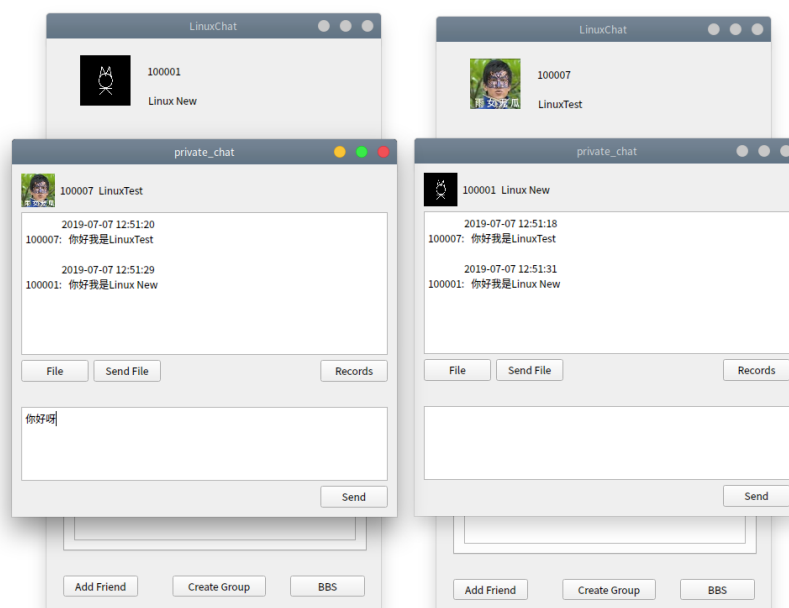
点击主窗口的昵称部分，弹出修改昵称提示框，输入新的昵称，点击 OK 即可修改昵称，修改成功后弹出成功提示框：



5.7.私聊

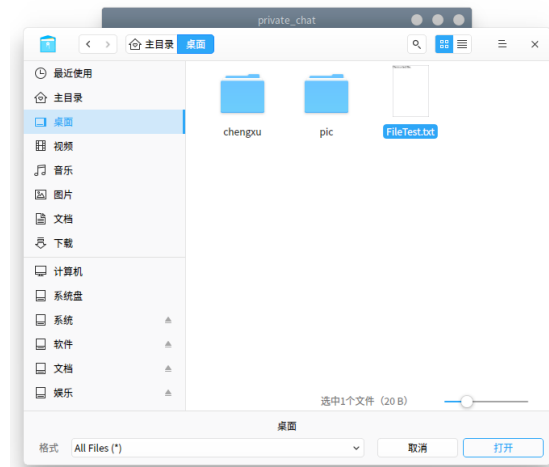
5.7.1 即时通讯

在好友列表点击头像进入私聊窗口，私聊窗口会显示对方的 ID 和昵称，编辑消息后点击 Send 即可发送消息，在进入窗口，会加载最近 5 条消息历史记录，点击 Records 按钮可以查看所有历史消息：

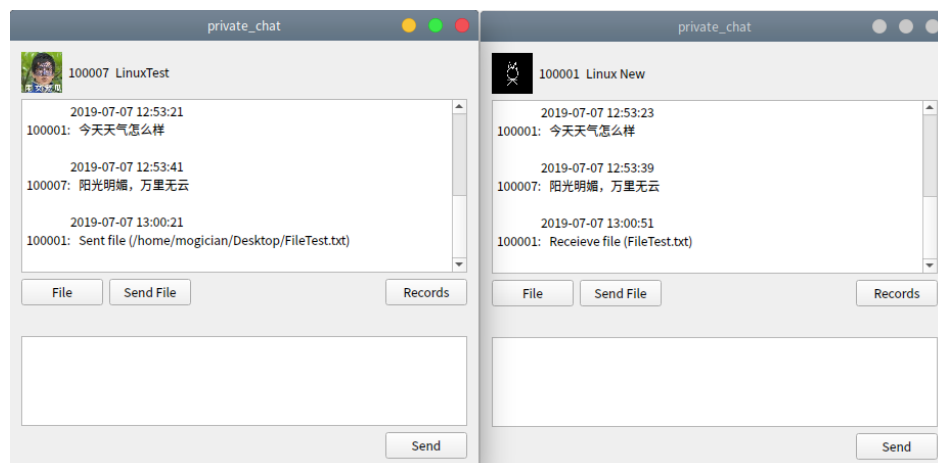


5.7.2 文件传输

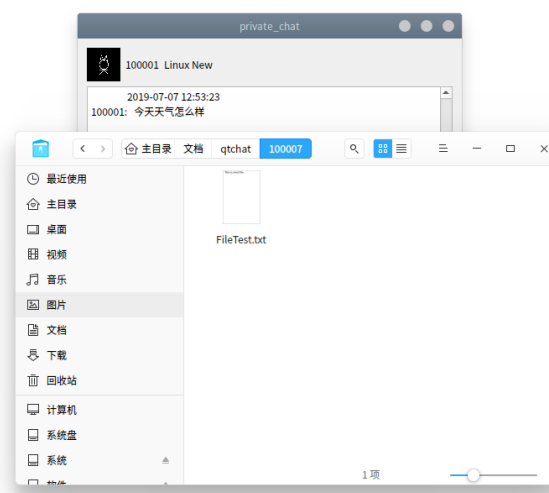
私聊窗口点击 File，弹出文件选择界面，选择要发送的文件，选择完成后点击 Send File 发送文件：



文件发送方显示发送成功提示，接收方显示接收消息的信息和接收文件名，并将接收文件存储在本地：

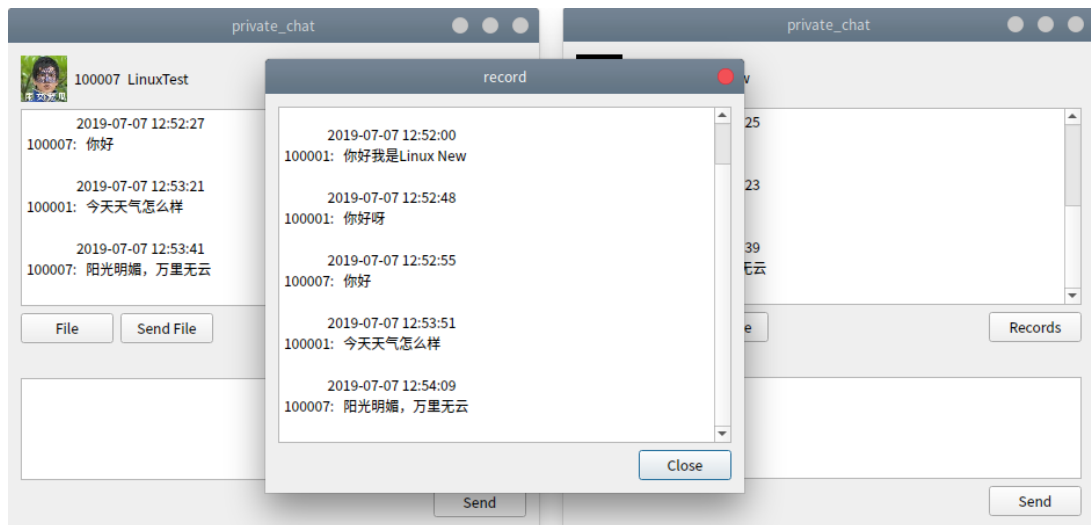


打开本地对应用户 ID 的文件夹，可以看到已经成功接收文件：



5.7.3 消息历史记录

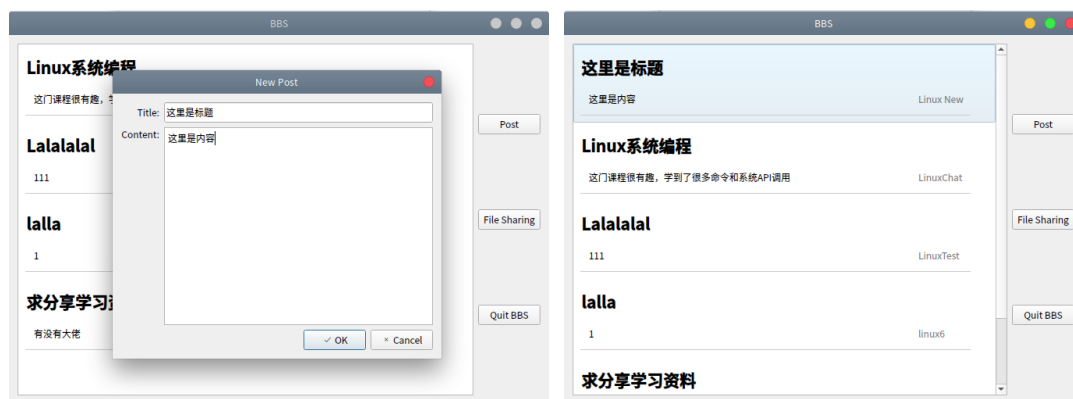
点击私聊窗口的 **Records** 按钮，即可查看所有的消息记录，消息记录存储在服务器端，点击的时候请求所有与对方的聊天记录。



5.8.BBS

5.8.1 发送帖子

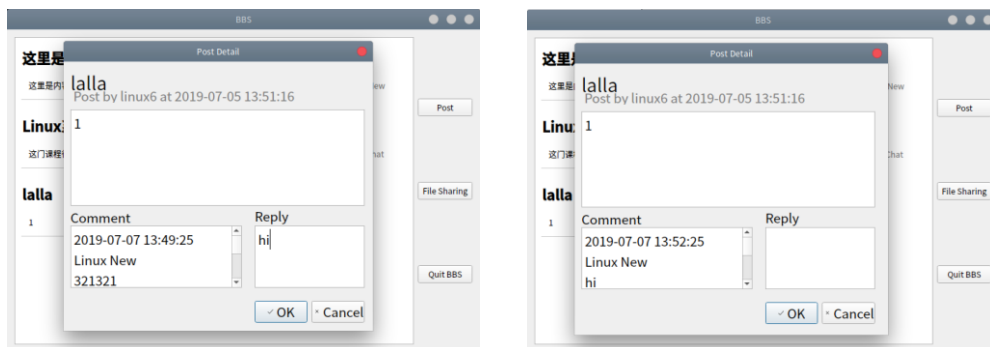
从主页面点击 **BBS** 按钮即可进入 BBS 的主页面，主页面显示所有用户发的帖子，点击 **Post** 即可新建帖子(下左图)，帖子列表将显示帖子的简要信息，比如帖子的标题，帖子的内容，发帖人的昵称：



点击 **OK** 之后将会向服务器发新建帖子请求，服务器将帖子的相关信息(发帖人、发帖时间、帖子标题、帖子内容、帖子的 ID)保存下来，成功之后客户端将刷新帖子列表(上右图)。

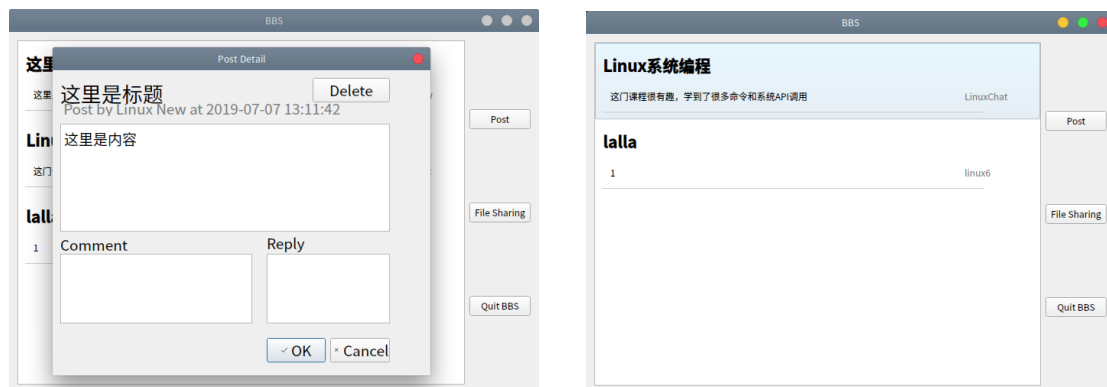
5.8.2 帖子评论

点击帖子列表中的帖子，即可打开帖子详情窗口，将加载该帖子下的所有回复，同时在右侧文本框，你也可以发送一个帖子回复，点击 OK 确认评论：



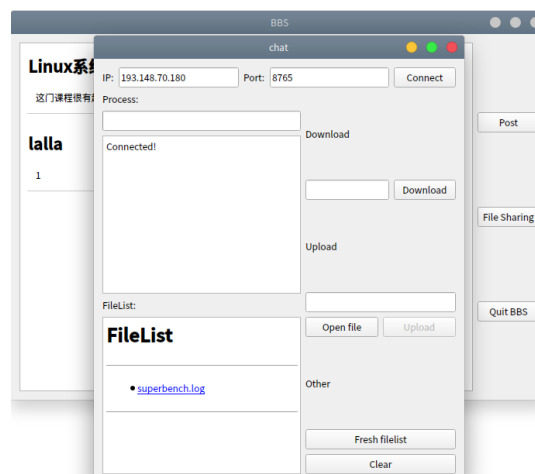
5.8.3 帖子删除

只有发帖人才有权限删除帖子，因此如果你是发帖人，将会在右上角显示删除按钮，点击删除即可删除帖子：

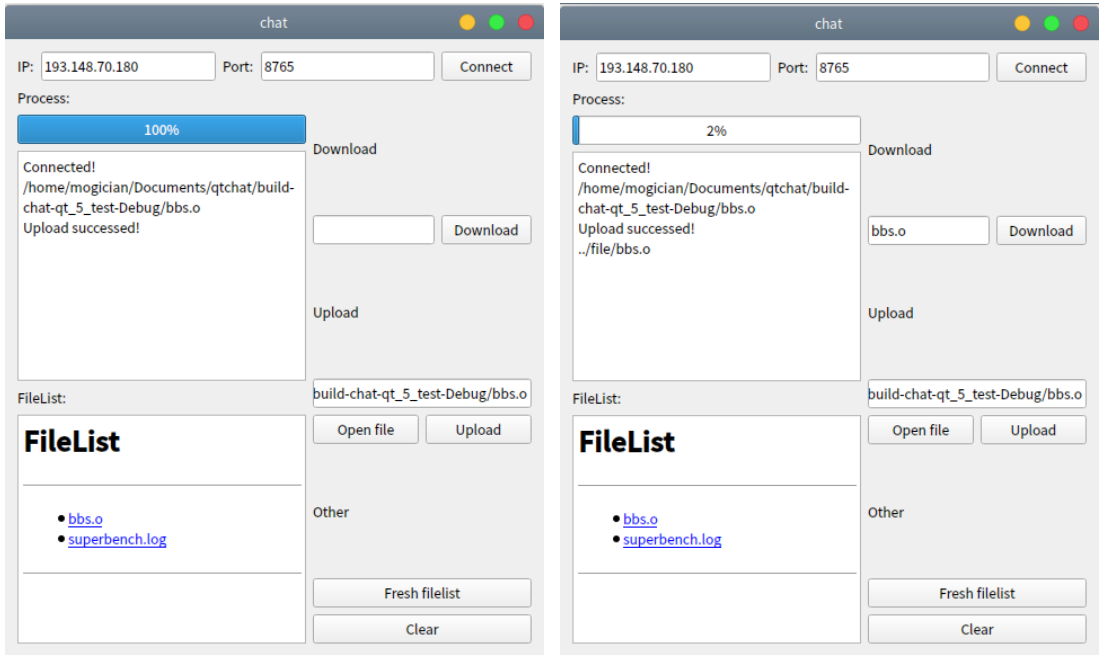


5.8.4 BBS 文件共享区

在 BBS 的主页面点击 File Sharing 进入文件共享界面。界面中 IP 框，Port 框分别填入文件共享服务器的 IP 地址和运行端口，客户端与服务器端通过 TCP 协议进行文件传输，点击 Connect 建立 TCP 连接，状态框输出“Connected!”消息。点击 Fresh file list 刷新文件共享服务器的文件列表，显示在左侧 file list 框中。点击 clear 可清除状态框的消息内容：



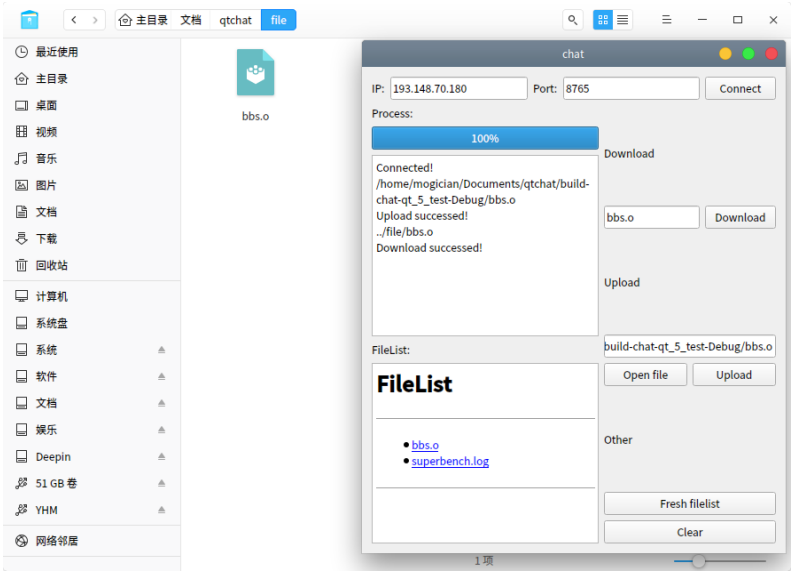
点击 **Open file** 打开本地文件，选中文件后，在 **Open file** 上方显示选中的文件的路径和文件名，点击 **Upload**，上传文件。进度条显示上传进度，状态框输出上传文件的路径和名称：



上传成功时，状态框输出“Upload succeed!”(上左图)。点击 **Fresh file list**，刷新文件列表内容，显示上传成功的文件。

输入文件共享服务器端的文件名，点击 **Download**，开始下载。进度条显示下载进度，状态框输出文件下载到的目录和文件名(上右图)。

下载成功时，状态框输出“Download succeed!”，文件被下载到 **file** 文件夹：



六、附录

6.1.数据库表

用户个人信息数据表

属性名称	类型	长度	是否非空	是否为主键	备注
用户 ID 号	int	10	是	是	
用户密码	varchar	50	是		加密存储
用户密保问题	varchar	50	是		
用户密保答案	varchar	50	是		加密存储
用户昵称	varchar	50	是		设置默认值为：新用户
用户注册时间	time		是		注册时自动生成
用户是否在线	boolean		是		
用户性别	char	1			
其它用户个人资料					

注：一个用户对应一条记录

用户好友数据表

属性名称	类型	长度	是否非空	是否为主键	备注
用户 1 ID 号	int	10	是	是	
用户 2 ID 号	int	10	是	是	

注：一个用户对应多条记录，(ID1, ID2) 与 (ID2, ID1) 为一条记录

群信息数据表

属性名称	类型	长度	是否非空	是否为主键	备注
群 ID 号	int	15	是	是	
群主 ID 号	int	10	是		
群名称	varchar	50	是		默认值为群主的昵称+创建的群
用户数量	int		是		默认值为 1，小于 100
群建立时间	time		是		建立时自动生成
其它群信息					

注：一个群对应一条记录

群成员数据表

属性名称	类型	长度	是否非空	是否为主键	备注
群 ID 号	int	15	是	是	
用户 ID 号	int	10	是	是	
其它信息： 是否为管理员？	boolean				

申请好友数据表

属性名称	类型	长度	是否非空	是否为主键	备注
发出申请用户 ID	int	10	是	是	
接收申请用户 ID	int	10	是	是	
申请信息	text	100			
申请日期	time		是		申请时自动生成
申请是否通过	boolean		是		

用户聊天数据表

属性名称	类型	长度	是否非空	是否为主键	备注
聊天记录 id 号	int	20	是	是	
发送者 ID 号	int	10	是		
接收者 ID 号	int	10	是		
消息内容 1	text				
消息内容 2	blob				2 种消息内容不能都为空
消息类型	boolean		是		
发送时间	time		是		发送时自动生成
是否发送成功	boolean		是		
是否已读	boolean		是		

群聊天数据表

属性名称	类型	长度	是否非空	是否为主键	备注
群聊天记录 id 号	int	20	是	是	
群 ID 号	int	15	是		
发送消息用户 ID 号	int	10	是		
消息内容	text	100	是		
发送时间	time		是		发送时自动生成

用户群聊天数据表

属性名称	类型	长度	是否非空	是否为主键	备注
群聊天记录 id 号	int	20	是	是	
用户 ID 号	int	10	是	是	
是否为发送者	boolean		是		
是否已读	boolean		是		

6.2.接口定义表

界面表

界面名称	界面功能	界面内容
login	登录界面	填写 ID、password 注册、忘记密码、记住密码、登录
reg_one	注册界面 1	填写昵称、两次输入密码 继续进行注册
reg_two	注册界面 2	填写两个密保问题及答案 返回上一注册界面、注册成功
reg_success	注册成功界面	显示注册成功的 ID
forget_one	忘记密码界面	填写两个密保问题答案 继续进行忘记密码设置
forget_two	重设密码界面	两次输入密码 完成重设密码
mainwindow	主界面	修改昵称、修改头像、 打开私聊界面、打开群聊界面、打开通知界面、 添加好友、创建群聊、进入 BBS
editname	修改昵称界面	修改昵称
private_chat	私聊界面	发送接收消息、发送接收文件、查看所有聊天记录
group_chat	群聊界面	发送接收群消息、查看群聊天记录、修改群名称、增加群成员
create_group	创建群聊	输入群聊名称
private_chat_message	私聊消息记录界面	显示所有私聊消息
group_chat_message	群聊消息记录界面	显示所有群聊消息
add_group_member	添加群成员界面	输入新的群成员 ID
notice_dialog	通知界面	选择是否接受添加好友请求
addfriend	添加好友界面	输入添加 ID 和申请信息

界面传递给界面数据表

数据发送界面	数据接收界面	传递数据
reg_one	reg_two	Nickname, Password
reg_two	reg_success	ID
login	forget_one	ID, Problem1, Problem2
forget_one	forget_two	ID
login	mainwindow	ID, Nickname, Photo friends(ID, Nickname, Photo, isonline) groups(groupID, groupname, Photo) Notifcations(ID, Nickname, Photo, Message, time)
mainwindow	private_chat	ID, Nickname, otherID, otherNickname, otherPhoto, messages (time, sendID, receiveID, content)
private_chat	private_chat_message	messages (time, sendID, receiveID, content)
mainwindow	group_chat	ID, Nickname, groupID, groupname, photo members(ID, nickname, isonline) group_messages (time, sendID, content)
private_chat	private_chat_message	messages (time, sendID, receiveID, content)
group_chat	group_chat_message	group_messages (time, groupID, sendID, content)
group_chat	add_group_member	groupID
mainwindow	create_group	ID
mainwindow	editname	ID
mainwindow	notice_dialog	(myID, ID, Nickname, Photo, Message, time)
mainwindow	addfriend	myID

界面与服务器数据传递表

数据发送者	数据接收者	传递数据	功能	类型
reg_two	服务器	Nickname, Password, Problem1, Answer1, Problem 2, Answer2	请求注册	
服务器	reg_two	ID	返回注册 ID	
login	服务器	ID	请求密保问题	
服务器	login	Problem1, Problem2	回复密保问题	
forget_one	服务器	ID, Answer1, Answer2	请求验证密保答案	
服务器	forget_one	right or wrong	正确与否	
forget_two	服务器	ID, Password	请求重设密码	
服务器	forget_two	success or false	成功与否	
login	服务器	ID, Password	请求登录	
服务器	login	success or false Nickname, Photo friends(ID, Nickname, Photo, isonline) groups(groupID, groupname, Photo) Notifcations(ID, Nickname, Photo, Message, time)	登录是否成功 返回昵称、头像 好友列表 群聊列表 通知列表	
editname	服务器	ID, new Nickname	请求修改昵称	
服务器	editname	success or false	成功与否	
mainwindow	服务器	ID, new photo	请求修改头像	
服务器	mainwindow	success or false	成功与否	
mainwindow	服务器	ID, otherID	请求获得最近 10 条聊天记录	
服务器	mainwindow	messages (time, sendID, receiveID, content)	返回最近 10 条聊天记录	
mainwindow	服务器	ID, groupID	请求获得最近 10 条群聊记录	
服务器	mainwindow	messages	返回最近 10	

		(time, sendID, receiveID, content)	条群聊记录	
private_chat	服务器	messages (time, sendID, receiveID, content)	私聊发送消息	
服务器	private_chat	messages (time, sendID, receiveID, content)	私聊接收消息	
private_chat	服务器	messages (time, sendID, receiveID, content)	私聊发送文件	
服务器	private_chat	messages (time, sendID, receiveID, content)	私聊接收文件	
private_chat	服务器	ID, otherID	请求所有聊天记录	
服务器	private_chat	messages (time, sendID, receiveID, content)	返回所有聊天记录	
mainwindow	服务器	groupID	请求群成员和最近 10 条群聊天记录信息	
服务器	mainwindow	group_members (ID, nickname, isonline) group_messages (time, groupID, sendID, content)	返回群成员信息和最近 10 条群聊天记录信息	
group_chat	服务器	group_messages (time, groupID, sendID, content)	发送群消息	
服务器	group_chat	group_messages (time, groupID, sendID, content)	接收群消息	
group_chat	服务器	groupID, new groupname	修改群名称	
服务器	group_chat	succeed or false	成功或失败	
add_group_member	服务器	groupID, new memberID	添加群成员	
服务器	add_group_member	success or false	成功或失败	
group_chat	服务器	groupID	请求所有群聊消息	
服务器	group_chat	group_messages	返回所有群	

		(time, groupID, sendID, content)	聊消息	
notice_dialog	服务器	myID, otherID	请求接受添加好友	
服务器	notice_dialog	success or false	添加好友成功	
notice_dialog	服务器	myID, otherID	请求忽略添加好友	
服务器	notice_dialog	success or false	忽略成功	
addfriend	服务器	myID, otherID, message, time	请求添加好友	
服务器	addfriend	success or false	请求发送成功	
create_group	服务器	ID	请求创建群聊	
服务器	create_group	groupID	创建群聊成功	