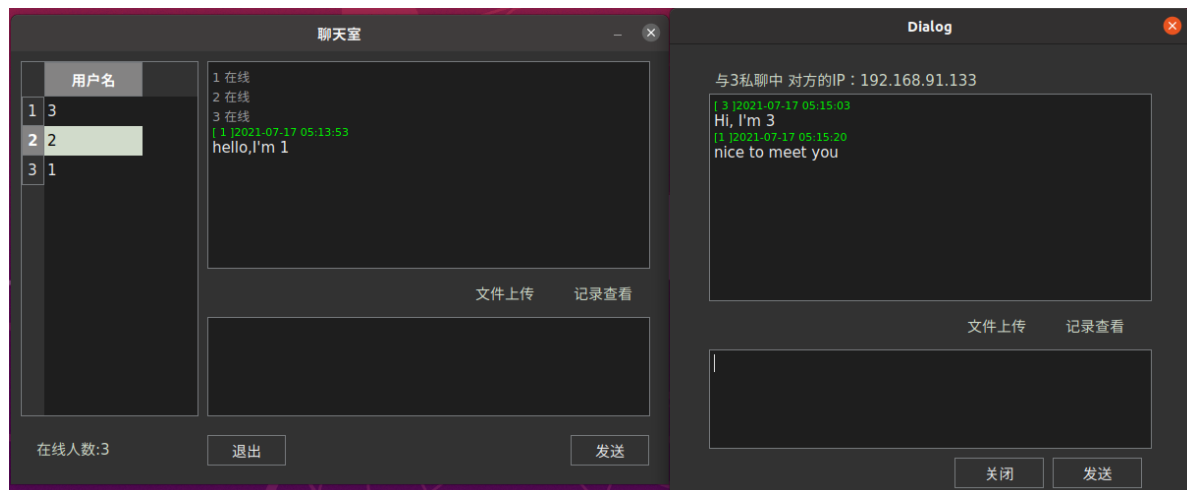


PChat - Linux即时通讯聊天室

宋尚儒/1120180717

概述

该程序实现了在局域网范围内的简易聊天室，程序使用qt实现，为无中心服务器的P2P模式，可以实现包括群聊、私聊、查看历史记录（自动保存在本地文件中）、发送文件功能。

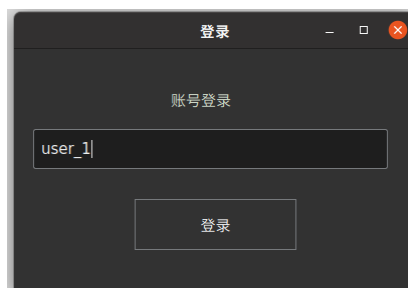


环境

- 操作系统: vmware16 ubuntu-20.04
- QT 5.90 GCC 64bit
- 使用端口号:
 - UDP: 8717 8718
 - TCP: 8719

基本功能展示

登录



输入用户名完成登录，进入群聊界面

群聊



在群聊界面的消息框中输入消息，点击发送即可在群聊中发送消息。

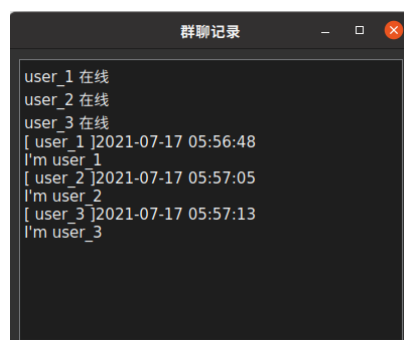
私聊



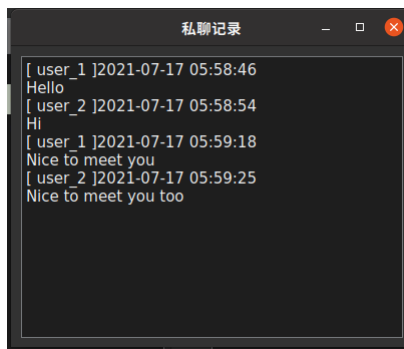
在群聊人员列表中双击某用户，即可进入私聊界面，当别的用户发起私聊时也会自动进入私聊界面，在私聊界面的消息框中输入消息，点击发送即可发送消息。

记录查看

可以在相应聊天界面点击记录查看来看聊天记录

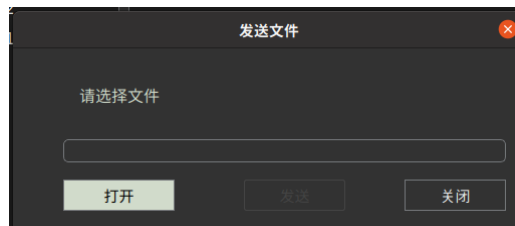


群聊记录如上所示

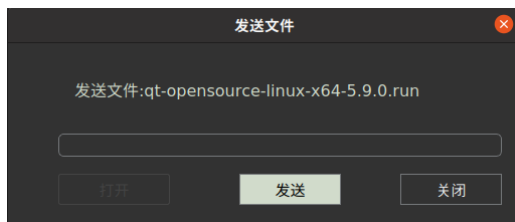


私聊记录如上所示

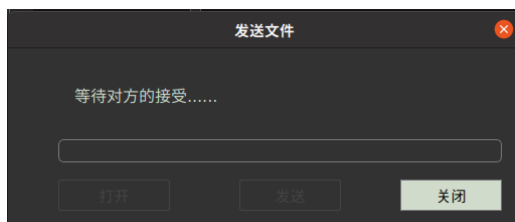
文件传输



在群聊界面选中某用户或者在私聊界面中，点击文件上传按钮后会弹出如上窗口



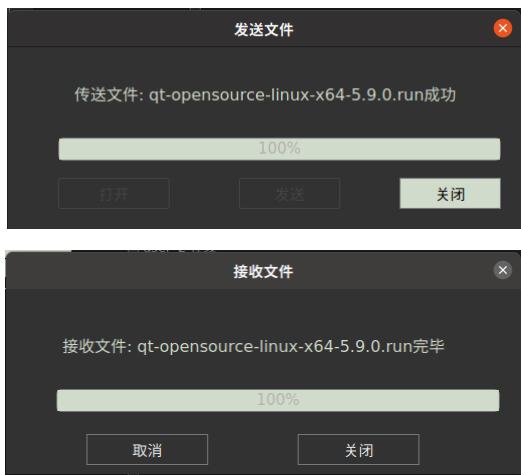
点击打开后弹出文件选择对话框，选择文件后如上所示



点击发送后需要等待对方确认



选择yes后弹出文件选择对话框，选择文件后开始传送，发送和接收窗口如上所示



传输完成后如上所示，接收端成功收到文件

功能设计

该程序是无中心的局域网聊天软件，不存在中心服务器对消息进行存储转发，采用自定义用户名、主机名、IP地址来标志一个客户端。对于消息的传递，采用UDP来实现；对于文件传输，由于需要更高的可靠性，采用TCP传输。

文件组成

```
main.cpp           //主函数
lib.h              //统一保存库，定义消息类型
mainwindow.cpp \ mainwindow.h //主窗口，群聊界面
login.cpp \ login.h   //登录界面
chat.cpp \ chat.h     //私聊界面
tcpserver.cpp \ tcpserver.h //文件发送机制
tcpclient.cpp \ tcpclient.h //文件接收机制
history.cpp \ history.h //历史记录
```

消息类型

通过UDP传输的消息需要负责在所有客户端之间维持网络的一致性，会在整个局域网进行广播发送，目标端口号为8717，可分为以下几种类型

- Message
普通消息，即聊天消息，格式为{用户名,主机名,本地IP地址,消息内容}
- NewParticipant
新加入用户，当用户新加入后需要广播该消息，收到该消息的用户会回复同类型消息
格式为{用户名,主机名,源IP地址}
- LeftParticipant
用户退出，告知用户离开
格式为{用户名,主机名}
- RefuseFile
拒绝接收文件，作为对发送文件消息的拒绝响应，在群聊中广播，在私聊中单播

格式为{用户名,主机名,源IP地址}

- SendFileName

发送文件，在群聊中广播，在私聊中单播

格式为{用户名,主机名,源IP地址,目标IP地址,文件名}

- AskChat

要求私信，仅存在单播

格式为{用户名,主机名,源IP地址}

用户登录

一个客户端通过自定义用户名、主机名、IP地址来标志，其中用户名由用户在登录界面输入。登陆界面由 `Login` 类进行实现，通过QT的信号槽机制，在main函数中连接登录信号和主窗口自定义的槽函数，实现用户名的传递。

群聊功能

完成登录后即会进入群聊界面，该界面主要通过 `Mainwindow` 类进行实现

群聊界面除了窗口的几个组件外，最重要的功能还是与其他客户端继续网络通信，因此分为两个主要功能，并可进一步细分

- 接收消息并处理

将UDP Socket绑定在固定端口，采用信号槽机制监听消息

- NewParticipant消息，收到后如果不在自己已知的用户中，广播NewParticipant告知新加入者自身存在，在自己群聊界面中添加对应用户，更新在线人数。
- LeftParticipant消息，收到后从自己群聊界面中删去对应用户，更新在线人数。
- Message消息，根据消息内容在聊天窗口显示聊天信息。
- SendFileName消息，收到后启动文件接收机制，具体会在之后文件传输功能中说明。
- RefuseFile消息，收到后表明对方拒绝接收文件，关闭文件发送机制。
- AskChat消息，收到后表明对方要求发起私聊，初始化私聊对象 `privateChatFrom`

- 接收本地操作并广播发送消息

- 登陆完成后，立刻发送NewParticipant消息，告知他人自己加入聊天室
- 在所有关闭主窗口的事件后，立刻发送LeftParticipant消息，告知他人自己离开聊天室。
- 点击发送按钮后，获取发送消息框内的消息，并且打包发送Message消息，在聊天室中发布消息。
- 通过信号槽机制接收到发送文件机制的选择完成信号（带有选择好的文件名）后，会将文件名打包为SendFileName消息并发送给待接收方。
- 当用户启动文件接收机制后，会首先启动是否接受的窗口，若选择接收，则会正式开始文件传输，否则发送RefuseFile消息。

私聊功能

私聊功能通过Chat类实现，该功能的启动可以分为两方，发起方和接收方

发起方

发起方通过在群聊界面用户列表双击目标接收方来启动一次私聊，启动过程需要完成如下工作

- 传入接收方的用户名、IP地址、自己的用户名、IP地址来初始化私聊对象`privateChatTo`，初始化需要绑定了本地IP地址和私聊的专用UDP端口8718，同时设置了信号槽机制及时处理在该端口接收到的消息。
- 向接收方发送AskChat消息，表示需要发起一次私聊

- 显示私聊窗口，开始私聊过程

私聊可以发送文件和消息，文件传输机制和群聊类似，会在之后详细说明。普通消息发送同样类似与群聊，包括Message、LeftParticipant等，需要注意的是这里均使用UDP单播发送。此外，完成消息发送后需要在私聊显示框内显示相应的消息。

此外通过信号槽机制，在私聊界面关闭时发送信号，使得主窗口即时释放对象。

接收方

接收方在客户端主窗口接收到AskChat消息后，根据消息中的用户名、IP地址启动一次私聊，启动过程需要完成如下工作

- 传入发起方的用户名、IP地址、自己的用户名、IP地址来初始化私聊对象privateChatFrom，初始化过程与发起方相同。
- 显示私聊窗口，开始私聊过程

私聊消息和文件的发送和接收的处理过程与发起方相同。同样通过信号槽机制即时释放对象。

历史记录功能

在群聊和私聊界面均可通过记录查看按钮查看对应的历史记录。历史记录以本地文件的形式进行保存，保存过程自动化进行，可分为群聊记录和私聊记录

- 群聊文件名为“RECORD_GROUP”
- 私聊分文件保存，通过用户名+IP地址进行标志，文件名格式为“RECORD_[username]_[IP]”。

当退出群聊界面或私聊界面的事件发生时（比如按下退出按钮），会根据界面后台对象对应的信息保存到对应的文件中。需要注意的是，文件采用追加方式进行写操作。

记录的查看使用History类进行实现，当窗口中按下记录查看按钮后，会读取相应的记录文件中的内容并且显示在History类的文本显示框内。

文件传输功能

文件传输功能需要实现文件的发送与接收，这通过tcpserver（发送端）类和tcpclient（接收端）类进行实现。因为文件传输需要比较高的可靠性，所以采用TCP来实现文件传输，绑定端口8719。

当群聊或私聊界面需要发送文件时，通过点击文件上传按钮（群聊界面还需要在用户列表单击目标用户）进行文件传输机制，需要完成如下工作

- 初始化server对象，通过信号槽机制将新tcp连接信号与真正执行文件发送的函数进行绑定。显示发送文件窗口
- 点击打开，在文件选择窗口中选择要发送的文件，然后点击发送按钮，之后需要等待接收端进行确认操作，一旦接收端确认，将会建立tcp连接，发送信号，触发真正的文件发送操作。
- 文件发送通过tcp进行发送，每向socket中写入一个缓冲区，便会触发一次更新界面操作，计算发送的文件量、发送速度、预估时间等信息，当发送文件量等于文件大小时，显示发送完成。

接收端接收到SendFileName消息后知道需要接收文件，如果选择接收，则需要进行如下工作

- 初始化client对象，通过信号槽机制将tcp的readyRead信号与接收文件流并保存到本地文件的函数进行绑定。显示接收文件窗口，选择文件保存地址。
- 文件接收通过tcp进行接收，会循环从socket中读入一个缓冲区并保存到本地文件中，触发更新界面操作，计算接收的文件量、发送速度、预估时间等信息，当接收文件量等于文件大小时，显示接收完成。

群聊和私聊窗口均可进行文件传输，通过以上机制可以实现文件的可靠传输。

总结

本次实验的设计思路一开始就是去中心化，包括代码构建都是在这一基础上进行的，但这一去中心系统的安全性无法得到保证，目前的完善思路应该是包括一个注册服务器和公私钥以及哈希认证机制来确保安全性，但受限于时间和能力无法进一步实现，构建过程中还接触并熟悉了linux下的部分系统调用机制。