项目详细

首先是登录对话框



该窗口对象为

QDialogLogin 继承于QDialog 类

并将其 用 setWindowFlags ()设置为无边框和登录界面模式

密码输入框为QLineEdit 类 将其用 setEchoMode 设置为Password 模式

默认密码为123456 ,会用MD5加密进行比对

退出时, 对 init.dat 文件 重新写入 先写入保存的用户名个数也就是QcomboBox 项的个数，再写入用户名

登录时,从init.dat文件读出 用户名个数 .用户名.并显示

由于设置为了无边框模式所以需要

重写下列事件

void *mousePressEvent*(QMouseEvent\*event);*//鼠标摁住*

void *mouseMoveEvent*(QMouseEvent\*event);*//鼠标移动*

void *mouseReleaseEvent*(QMouseEvent\*event);*//鼠标松开*

来实现对话框的拖动

在鼠标按住时

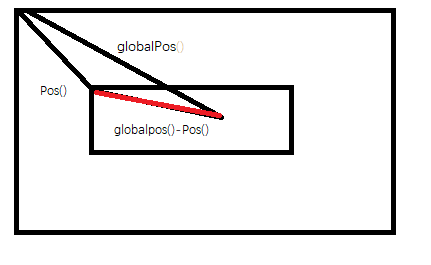
*mousePressEvent*(QMouseEvent\*event)内部

使用globalPos()函数来

*获取鼠标以电脑屏幕左上角为原点的向量*

*Pos()函数获取以对话框左顶点以电脑屏幕左上角为原点的向量*

m\_lastPos=globalPos()-*Pos() 算出 鼠标以对话框左顶点为原点的向量，这个向量在拖动过程中不会变，如下图*

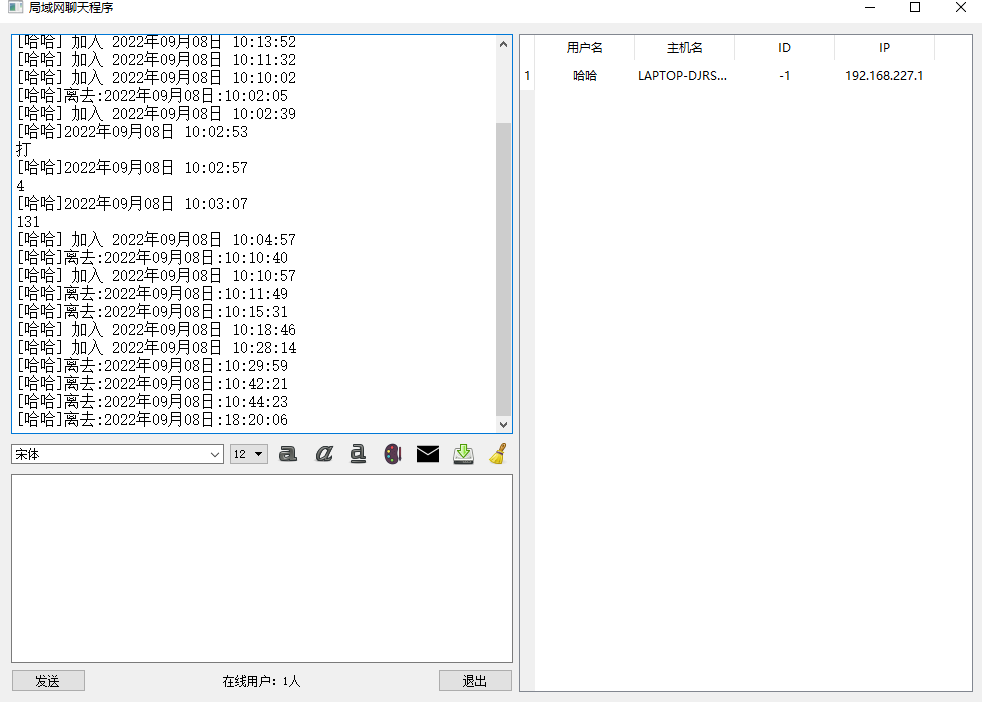


*然后当松开鼠标时用*m\_lastPos-globalPos()算出

移动后*对话框左顶点以电脑屏幕左上角为原点的向量*

*然后用move函数移动*

*聊天窗口*



*聊天记录框为 QTextBrowser类*

*发送框为QTextEdit*

*右边为QTableWidget*

*用户的登陆，下线，文件接收提醒，都使用udp广播实现*

*定义*

*enum* MessageType

{

*Message*,*//通信消息*

*NewUser*,*//新的用户*

*UserLeft*,*//用户离去*

*FileName*,*//文件名*

*RefuseFile*,*//拒绝接收文件*

};

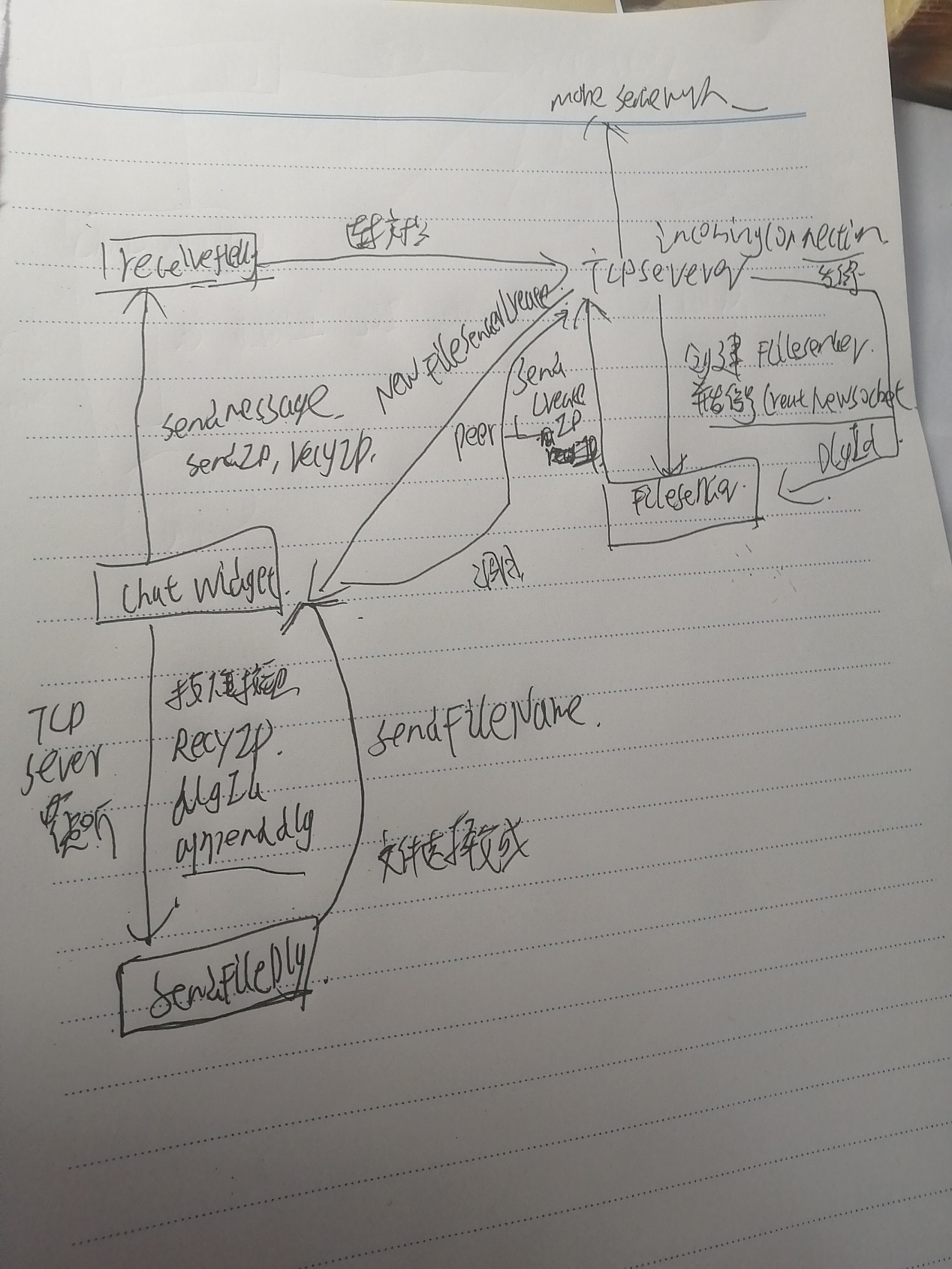
*来区分不同的消息*

*每次发送udp报文最先写入消息类型，接收端根据不同的消息的类型来做相应的处理*

*每个新用户登录首先都会广播一个udp登录类型的报文，当旧用户收到新用户的报文它也会广播一个udp登录类型的报文使得新用户能知道它也登录了.*

*文件传输*

*的逻辑有点复杂*

**

*因为必须要在线程内部创建Qtcpsocket所以只能通过发信号的方式控制socket和线程*

*也就是MyTCpServer类负责控制，发送线程和发送对话框也不在一个线程，所以也要发信号和建立信号来通知线程服务于哪个对话框，*

*文件开时传输*

*首先发送一个文件头*

*把要发送文件的大小 文件名大小，文件名*

使用QDataStream 数据流依次写入字节数组

发送文件头的函数会进行连接

connect(Clientsocket,*SIGNAL*(bytesWritten(qint64 )),

*this*,*SLOT*(updataSendProgess(qint64)));

不知道为啥有时候会发不出去

要调用一下

Clientsocket->*waitForBytesWritten*(100);

类里会记录以下变量

m\_TotalBytes=0;*//要发送数据的总大小*

m\_bytesWritten=0;*//已经发送数据的大小*

m\_bytesToWrite=0;*//剩余数据的大小*

q m\_payloadSize=0;*//每次发送数据的大小*

发送端每次发64kb

接收端同样用QDataStream 数据流先读出文件头

创建文件缓存，

然后接收到字节数组然后写入文件.