东软睿道

项目实训总结报告

项 目 名 称: ColorMail简易邮箱系统

学 校 名 称: 北京理工大学

项 目 讲 师: 卢春江

姓 名: 蒋泽林

学 号: 1120182394

实 训 日 期: 2020年9月7日-9月17日

**1 引言**

1.1项目概述

本项目是以Qt为平台，利用QTcpSocket与QTcpServer来实现客户端和服务端之间的连接，以此来做一个简易的局域网内能互发邮件的邮件系统。本系统利用了许多Qt中类似于QTcpSocket、QJsonArray，QListWidegt，QWidegt，QLabel，QDateTime等原有的库文件，并且可以通过手动增加页面控件可视化的设计ui界面，最方便的是加入了槽函数和信号的机制，使得编写代码更加简单与方便，如此给我们的开发提供了很大的方便。

1.2 需求分析

在现实生活中，邮件系统无处不在，很多时候都是通过发邮件通知许多重要的事，这给我们的生活提供了极大的便利，因此通过实现简易的邮件系统，可以联网互相发送邮件，这是非常重要的一次体验。

1.3 运行环境

I5-8300H,1060

Ubuntu 20.04.1LTS 64bit

**2 项目设计**

2.1 设计思路

在本次项目中，我主要负责前端，不仅需要构思界面的主体内容，需要包含哪些部件，更需要将对应的页面构建所需要的功能实现，我主要是应用信号和槽函数去实现对应按键的功能，应用QDateTime实现实时时间显示，QJsonArray来接受服务器所发送的内容，QTcpSocket连接服务器等功能。

2.2 模块功能介绍

接受服务器所返回的内容和信号，并且连接服务器，向服务器发送界面上对应的内容，主要是两个部分，并且，还需要实现个界面对应按键的功能，利用槽函数和信号，在一个界面通过信号发送另一个界面所需要的内容，并在另一个界面中通过public slot函数去接受发送的内容，达到页面间的值传递。

2.3 模块结构图

QTcpSocket

SIGNAL

服务器

界面

界面

QJsonArray

SLOT

2.4 程序流程图

发送内容：QString 内容——> QTcpSocket连接服务器——>append发送给服务器

接受服务器内容：QJsonArray服务器内容——>toString转化为字符串——>QString接受

界面传值：Connect——>SIGNAL——>emit发送 ——>SLOT——>接受

2.5 功能设计分工

由于我负责前端，因此主要是接受服务器数据，连接服务器，最主要的是实现界面上对应按键的功能

**3 详细设计**

在前端的主要是依靠信号和槽函数机制来实现功能，并且需要配合后端的工作，如果后端需要什么样的数据，就将相应格式的内容通过服务器连接QTcpSocket发送给后端，再者就是接受服务器发送的收据，将其转换为字符形式，如果需要在多个界面之前传递值或者实现某功能，那都是在一个界面利用SIGNAL将对应的内容通过emit发送出去，然后在需要接受的界面添加public SLOT槽函数来接收信号发出的内容，通过connect来连接，即可实现对应功能。

**4 实现方案**

1. 主界面与添加好友界面之间的跳转，在点击添加好友后回跳转到加好友的界面，当添加好友成功后会显示添加成功得字样，此时关闭窗口回退到主界面，点击联系人一栏可以看到

代码：

void MainWindow::**on\_pushButton\_send\_clicked**(bool checked) //显示窗口

{

emit sendname(name);

addpeople->show();

}

void addfriend::**on\_pushButton\_clicked**() //点击添加按钮

{

tcpClient = new QTcpSocket(this);

tcpClient->abort();

tcpClient->*connectToHost*("192.168.43.188", 6666); //连接服务器

QString ID = ui->lineEdit->text();

QString n\_name = ui->lineEdit\_2->text();

if( ID==NULL )

{

ui->label\_notice->setText("无效操作");

return;

}

QJsonArray add\_array;

add\_array.append(2);

add\_array.append(name);

add\_array.append(ID);

add\_array.append(n\_name);

QJsonDocument add\_doc;

add\_doc.setArray(add\_array);

QByteArray add\_byte = add\_doc.toJson(QJsonDocument::Compact);

tcpClient->write(add\_byte);

connect(tcpClient, SIGNAL(readyRead()), this, SLOT(add\_GetTorF()));

}

void addfriend::**add\_GetTorF**()

{

QByteArray add\_qba = tcpClient->readAll();

QJsonParseError json\_error;

QJsonDocument add\_getdoc(QJsonDocument::fromJson(add\_qba, &json\_error));

if(json\_error.error != QJsonParseError::NoError)

{

return;

}

QJsonArray ar\_add = add\_getdoc.array();

int add\_flag = ar\_add[0].toInt();

if(add\_flag == 0)//添加成功之后显示

{

ui->label\_notice->setText("添加成功!");

QString userID = ui->lineEdit->text();

QString username = ui->lineEdit\_2->text();

emit successaddfriend(userID , username); //向email发送用户信息

}

else if(add\_flag == 1)//添加不成功显示

{

ui->label\_notice->setText("已添加该用户！");

}

else

{

ui->label\_notice->setText("用户不存在！");

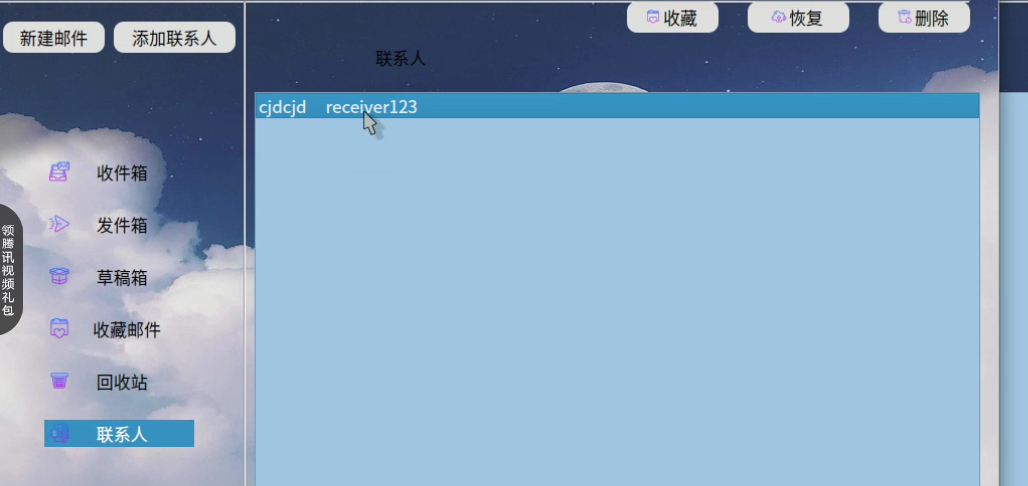
}

}

图片：







1. 主界面与发送邮件之间的连接，点击新建邮件或者双击联系人都能跳转到发送邮件得界面，并还双击联系人还可以将联系人自动填入到发件人一栏

代码：

void MainWindow::**on\_pushButton\_newcreat\_clicked**()//新建发送窗口

{

if( newcreatflag )

{

emit sentusermessage(ui->listWidget\_friend); //将listWidget\_friend的指针传递给sentemail

newcreatflag = 0; //关闭接受主页面的联系人传值

}

sentemail->show();

emit sendname(name);

}

void MainWindow::**on\_listWidget\_friend\_itemDoubleClicked**(QListWidgetItem \*item) //双击联系人转到sentemail界面

{

QObject::connect(this,SIGNAL(sentreceivermessage(QString)),sentemail,SLOT(receivermessage(QString)));//连接email与sentemail窗口

QString receiver = item->text();

QString receivername = NULL;

for(int i=0 ; i<receiver.size();i++)

{

if( receiver[i]==' ' and receiver[i+1]!=' ' )

{

for( int j = i+1 ; j < receiver.size() ; j++ )

{

receivername+=receiver[j];

}

break;

}

}

if( newcreatflag )

{

emit sentusermessage(ui->listWidget\_friend); //将listWidget\_friend的指针传递给sentemail

newcreatflag = 0; //关闭接受主页面的联系人传值

}

emit sentreceivermessage(receivername);

sentemail->show();

emit sendname(name);

}

void sentEmail::**receivermessage**(QString data)//接受联系人信息

{

ui->lineEdit\_receiver->setText(data);

}

void sentEmail::**receiveusermessage**(QListWidget \* listwidget\_friend)

{

for(int i=0 ; i<listwidget\_friend->count() ; i++)

{

ui->listWidget\_contacts->addItem(listwidget\_friend->item(i)->text()); //将email中的信息加入到sentemail的联系人中

}

}

图片：



1. 双击发件箱，收件箱，草稿箱，收藏夹，回收站，都能查看邮件内容，并且有附件的可以发送附件，草稿箱打开可编辑

代码：

void MainWindow::**on\_listWidget\_receive\_itemDoubleClicked**(QListWidgetItem \*item) //双击收件箱信件查看信件内容 ???????信件内容+附件

{

QString information = item->text();

emailinformation = item;

RdretcpSocket = new QTcpSocket(this);

RdretcpSocket->abort();

RdretcpSocket->*connectToHost*("192.168.43.188", 6666); //连接服务器

connect(RdretcpSocket, SIGNAL(readyRead()), this, SLOT(Rev\_Rev()));

QJsonArray Rdre\_ar;

Rdre\_ar.append(5);

Rdre\_ar.append(1);

QString username=NULL;

QString theme=NULL;

QString time=NULL;

int i =0;

for(i=0 ; i<information.size() ;i++) //得到用户名

{

if( information[i]!=' ' )

{

username+=information[i];

}

else

{

break;

}

}

for(;i<information.size(); i++)

{

if( information[i]!=' ' )

{

break;

}

}

for(; i <information.size();i++) //得到主题

{

if( information[i]!=' ' )

{

theme+=information[i];

}

else

{

break;

}

}

for(;i<information.size(); i++)

{

if( information[i]!=' ' )

{

break;

}

}

for(;i<information.size();i++) //得到时间

{

time+=information[i];

}

Rdre\_ar.append(username);

Rdre\_ar.append(theme);//??????????????????需要邮件的主题、时间

Rdre\_ar.append(time);

QJsonDocument Rdre\_doc;

Rdre\_doc.setArray(Rdre\_ar);

QByteArray Rdre\_bt = Rdre\_doc.toJson(QJsonDocument::Compact);

RdretcpSocket->write(Rdre\_bt);

//RdretcpSocket->readyRead();

int row = ui->listWidget\_receive->currentRow();

ui->listWidget\_receive->item(row)->setBackground(QColor("white")); //已读邮件

}

void MainWindow::**Rev\_Rev**()

{

QByteArray Rdre\_qba = RdretcpSocket->readAll();

QJsonParseError json\_error;

QJsonDocument Rdre\_getdoc(QJsonDocument::fromJson(Rdre\_qba, &json\_error));

if(json\_error.error != QJsonParseError::NoError) //发生错误

{

return;

}

QJsonArray Rdre\_array = Rdre\_getdoc.array();

//内容是Rdre\_array[0],是否有附件是Rdre\_array[1],有附件则值为1

QString text = Rdre\_array[0].toString();

int file = Rdre\_array[1].toInt();

qDebug() << file ;

connect(this,SIGNAL(look\_receiveemail(QString,QString,QString,QString,int)),look\_email,SLOT(receive\_receiveemail(QString,QString,QString,QString,int)));

lookemail\_information(emailinformation , 0 ,text , file);

}

void MainWindow::**on\_listWidget\_send\_itemDoubleClicked**(QListWidgetItem \*item) //双击发件箱信件查看信件内容 ????????信件内容+附件

{

emailinformation = item;

RdsdtcpSocket = new QTcpSocket(this);

RdsdtcpSocket->abort();

RdsdtcpSocket->*connectToHost*("192.168.43.188", 6666); //连接服务器

connect(RdsdtcpSocket, SIGNAL(readyRead()), this, SLOT(Rev\_Send()));

QJsonArray Rdsd\_ar;

Rdsd\_ar.append(5);

Rdsd\_ar.append(0);

QString username;

QString theme;

QString time;

QString information = item->text();

int i =0;

for(i=0 ; i<information.size() ;i++) //得到用户名

{

if( information[i]!=' ' )

{

username+=information[i];

}

else

{

break;

}

}

for(;i<information.size(); i++)

{

if( information[i]!=' ' )

{

break;

}

}

for(; i <information.size();i++) //得到主题

{

if( information[i]!=' ' )

{

theme+=information[i];

}

else

{

break;

}

}

for(;i<information.size(); i++)

{

if( information[i]!=' ' )

{

break;

}

}

for(;i<information.size();i++) //得到时间

{

time+=information[i];

}

Rdsd\_ar.append(username);

Rdsd\_ar.append(theme);//??????????????????需要邮件的主题、时间

Rdsd\_ar.append(time);

QJsonDocument Rdsd\_doc;

Rdsd\_doc.setArray(Rdsd\_ar);

QByteArray Rdsd\_bt = Rdsd\_doc.toJson(QJsonDocument::Compact);

RdsdtcpSocket->write(Rdsd\_bt);

}

void MainWindow::**Rev\_Send**()

{

QByteArray Rdsd\_qba = RdsdtcpSocket->readAll();

QJsonParseError json\_error;

QJsonDocument Rdsd\_getdoc(QJsonDocument::fromJson(Rdsd\_qba, &json\_error));

if(json\_error.error != QJsonParseError::NoError) //发生错误

{

return;

}

QJsonArray Rdsd\_array = Rdsd\_getdoc.array();

//内容是Rdsd\_array[0],是否有附件是Rdsd\_array[1],有附件则值为1

QString text = Rdsd\_array[0].toString();

int file = Rdsd\_array[1].toInt();

connect(this,SIGNAL(look\_sendemail(QString,QString,QString,QString,int)),look\_email,SLOT(receive\_sendemail(QString,QString,QString,QString,int)));

lookemail\_information(emailinformation , 1 ,text , file);

}

void MainWindow::**on\_listWidget\_draft\_itemDoubleClicked**(QListWidgetItem \*item) //双击草稿箱查看信件内容

{

emailinformation = item;

RdDrtcpSocket = new QTcpSocket(this);

RdDrtcpSocket->abort();

RdDrtcpSocket->*connectToHost*("192.168.43.188", 6666); //连接服务器

connect(RdDrtcpSocket, SIGNAL(readyRead()), this, SLOT(Rev\_Drf()));

QJsonArray RdDr\_ar;

RdDr\_ar.append(5);

RdDr\_ar.append(0);

QString username;

QString theme;

QString time;

QString information = item->text();

int i =0;

for(i=0 ; i<information.size() ;i++) //得到用户名

{

if( information[i]!=' ' )

{

username+=information[i];

}

else

{

break;

}

}

for(;i<information.size(); i++)

{

if( information[i]!=' ' )

{

break;

}

}

for(; i <information.size();i++) //得到主题

{

if( information[i]!=' ' )

{

theme+=information[i];

}

else

{

break;

}

}

for(;i<information.size(); i++)

{

if( information[i]!=' ' )

{

break;

}

}

for(;i<information.size();i++) //得到时间

{

time+=information[i];

}

RdDr\_ar.append(username);

RdDr\_ar.append(theme);//??????????????????需要邮件的主题、时间

RdDr\_ar.append(time);

QJsonDocument RdDr\_doc;

RdDr\_doc.setArray(RdDr\_ar);

QByteArray RdDr\_bt = RdDr\_doc.toJson(QJsonDocument::Compact);

RdDrtcpSocket->write(RdDr\_bt);

}

void MainWindow::**Rev\_Drf**()

{

QByteArray RdDr\_qba = RdDrtcpSocket->readAll();

QJsonParseError json\_error;

QJsonDocument RdDr\_getdoc(QJsonDocument::fromJson(RdDr\_qba, &json\_error));

if(json\_error.error != QJsonParseError::NoError) //发生错误

{

return;

}

QJsonArray RdDr\_array = RdDr\_getdoc.array();

//内容是RdDr\_array[0],是否有附件是RdDr\_array[1],有附件则值为1

QString text = RdDr\_array[0].toString();

int file = RdDr\_array[1].toInt();

connect(this,SIGNAL(look\_draftemail(QString,QString,QString,QString,int)),look\_email,SLOT(receive\_draftemail(QString,QString,QString,QString,int)));

lookemail\_information(emailinformation , 2 ,text , file);

}

void MainWindow::**on\_listWidget\_collect\_itemDoubleClicked**(QListWidgetItem \*item) //双击收藏箱查看信件内容

{

emailinformation = item;

RdCotcpSocket = new QTcpSocket(this);

RdCotcpSocket->abort();

RdCotcpSocket->*connectToHost*("192.168.43.188", 6666); //连接服务器

connect(RdCotcpSocket, SIGNAL(readyRead()), this, SLOT(Rev\_Col()));

QJsonArray RdCo\_ar;

RdCo\_ar.append(5);

RdCo\_ar.append(0);

QString username;

QString theme;

QString time;

QString information = item->text();

int i =0;

for(i=0 ; i<information.size() ;i++) //得到用户名

{

if( information[i]!=' ' )

{

username+=information[i];

}

else

{

break;

}

}

for(;i<information.size(); i++)

{

if( information[i]!=' ' )

{

break;

}

}

for(; i <information.size();i++) //得到主题

{

if( information[i]!=' ' )

{

theme+=information[i];

}

else

{

break;

}

}

for(;i<information.size(); i++)

{

if( information[i]!=' ' )

{

break;

}

}

for(;i<information.size();i++) //得到时间

{

time+=information[i];

}

RdCo\_ar.append(username);

RdCo\_ar.append(theme);//??????????????????需要邮件的主题、时间

RdCo\_ar.append(time);

QJsonDocument RdCo\_doc;

RdCo\_doc.setArray(RdCo\_ar);

QByteArray RdCo\_bt = RdCo\_doc.toJson(QJsonDocument::Compact);

RdCotcpSocket->write(RdCo\_bt);

}

void MainWindow::**Rev\_Col**()

{

QByteArray RdCo\_qba = RdCotcpSocket->readAll();

QJsonParseError json\_error;

QJsonDocument RdCo\_getdoc(QJsonDocument::fromJson(RdCo\_qba, &json\_error));

if(json\_error.error != QJsonParseError::NoError) //发生错误

{

return;

}

QJsonArray RdCo\_array = RdCo\_getdoc.array();

//内容是Rdre\_array[0],是否有附件是Rdre\_array[1],有附件则值为1

QString text = RdCo\_array[0].toString();

int file = RdCo\_array[1].toInt();

connect(this,SIGNAL(look\_collectemail(QString,QString,QString,QString,int)),look\_email,SLOT(receive\_collectemail(QString,QString,QString,QString,int)));

lookemail\_information(emailinformation ,3 , text,file);

}

void MainWindow::**on\_listWidget\_recycle\_itemDoubleClicked**(QListWidgetItem \*item) //双击回收站查看信件内容

{

emailinformation = item;

RdDetcpSocket = new QTcpSocket(this);

RdDetcpSocket->abort();

RdDetcpSocket->*connectToHost*("192.168.43.188", 6666); //连接服务器

connect(RdDetcpSocket, SIGNAL(readyRead()), this, SLOT(Rev\_Del()));

QJsonArray RdDe\_ar;

RdDe\_ar.append(5);

RdDe\_ar.append(0);

QString username;

QString theme;

QString time;

QString information = item->text();

int i =0;

for(i=0 ; i<information.size() ;i++) //得到用户名

{

if( information[i]!=' ' )

{

username+=information[i];

}

else

{

break;

}

}

for(;i<information.size(); i++)

{

if( information[i]!=' ' )

{

break;

}

}

for(; i <information.size();i++) //得到主题

{

if( information[i]!=' ' )

{

theme+=information[i];

}

else

{

break;

}

}

for(;i<information.size(); i++)

{

if( information[i]!=' ' )

{

break;

}

}

for(;i<information.size();i++) //得到时间

{

time+=information[i];

}

RdDe\_ar.append(username);

RdDe\_ar.append(theme);//??????????????????需要邮件的主题、时间

RdDe\_ar.append(time);

QJsonDocument RdDe\_doc;

RdDe\_doc.setArray(RdDe\_ar);

QByteArray RdDe\_bt = RdDe\_doc.toJson(QJsonDocument::Compact);

RdDetcpSocket->write(RdDe\_bt);

}

void MainWindow::**Rev\_Del**()

{

QByteArray RdDe\_qba = RdDetcpSocket->readAll();

QJsonParseError json\_error;

QJsonDocument RdDe\_getdoc(QJsonDocument::fromJson(RdDe\_qba, &json\_error));

if(json\_error.error != QJsonParseError::NoError) //发生错误

{

return;

}

QJsonArray RdDe\_array = RdDe\_getdoc.array();

//内容是Rdre\_array[0],是否有附件是Rdre\_array[1],有附件则值为1

QString text = RdDe\_array[0].toString();

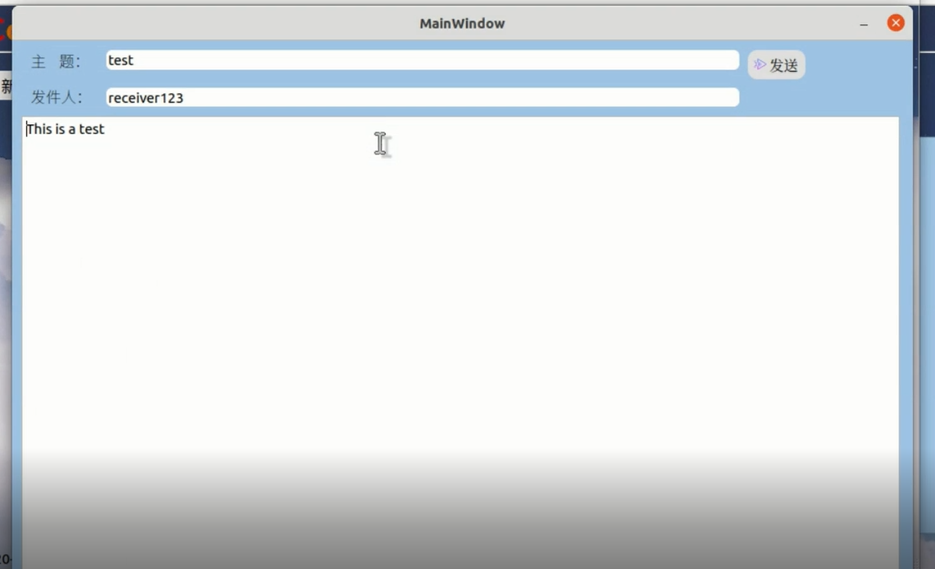
int file = RdDe\_array[1].toInt();

connect(this,SIGNAL(look\_deleteemail(QString,QString,QString,QString,int)),look\_email,SLOT(receive\_deleteemail(QString,QString,QString,QString,int)));

lookemail\_information(emailinformation ,4 , text,file);

}

图片：



4、还有很多其他功能，实现方法都是基于信号和槽函数，因此不再过多展示，在前端中，除了注册和登陆界面不由我负责，后面的所有界面全部由我负责编写相应功能的实现。

**5 系统测试**

针对前端需要接受后端传送的内容，所以我很多时候需要和后端的同学一起debug，因为每个传送过来的文件我都特殊命名，因此我在客户端deBug的时候可以及时地看到每次传输过来的内容是否正确，从而从中找取Bug。用这种方法可以较快地找出是前端还是后端（数据库）的问题。

在窗口之间传值，我一般都会用qDebug来输出内容，查找对应函数是否执行，检查信号函数和槽函数，以及将其是否连接在一起，因为一开始编写代码的不规范，因此很多时候后面的bug都是因为指针没有delete的原因，导致明明关闭窗口，但是再以打开以前的内容被没有被删掉，最后就是很多代码冗余过高，导致很难去查找问题所在，每次调试的时间都特别久，这确实对我们的编码规范有一定要求。

**6 心得体会**

在本次项目实训中，我们小组四位同学的基础都很一般，对Qt操作和Linux系统都不是很熟悉，更不要说实际的编程和写软件了。因此在一周左右的高强度开发中，我们彼此配合，每天都一起商量好安排接下来的学习内容和任务。这是让我体会的最深的一点，即同组同学之间的配合与默契在合作开发中特别重要。

其次，我负责的是前端的界面优化和功能实现，因为需要接受后端的内容，在deBug的时候同时需要找前后端两边的问题，各个界面之间的，因此也让我对deBug的过程更加印象深刻，对断点以及实时显示数据值的利用更加熟练。

很感谢这一次的专业实习，让我第一次真正上手写一个可以真正使用的软件，让我们小组成员有了巨大的成就感，也让我们真正地体会到了做一个合格的程序员的不简单。这一次印象深刻的实习让我体会到了编程的快乐和默契配合的爽快，也让我坚定信心将这一门学科学好，为当一个合格的程序员做好准备！