飞鸽传书——Qt版的QQ

[飞鸽传书——Qt版的QQ 1](#_Toc62409619)

[1 前言 1](#_Toc62409620)

[2 PART01 ---- 简单的服务器和客户端的登陆界面 1](#_Toc62409621)

[2.1 Linux下（纯C代码） 2](#_Toc62409622)

[2.1.1 任务目标：实现最简单的服务器功能 2](#_Toc62409623)

[2.1.2 任务实现 2](#_Toc62409624)

[2.2 Windows下（Qt代码） 10](#_Toc62409625)

[2.2.1 任务目标：制作登录窗口 10](#_Toc62409626)

[2.2.2 任务实现： 10](#_Toc62409627)

[2.2.2.1 创建Login类 10](#_Toc62409628)

[2.2.2.2 类的前置声明与预编译指令 10](#_Toc62409629)

[2.2.2.3 声明并初始化需要的窗口插件 11](#_Toc62409630)

[2.2.2.4 使用栅格布局管理器布局 13](#_Toc62409631)

[2.2.2.5 设置窗口信息 14](#_Toc62409632)

[2.2.2.6 设置按钮相关的信号与槽 16](#_Toc62409633)

[2.2.2.7 编辑main.cpp，实现从登陆界面到主界面的跳转 17](#_Toc62409634)

[2.3 实现效果 18](#_Toc62409635)

[2.3.1 Server 18](#_Toc62409636)

[2.3.2 Client 19](#_Toc62409637)

# 前言

**平台：Windows、Linux**

**编译器：Qt Create 4.11.0**

# PART01 ---- 简单的服务器和客户端的登陆界面

+ 新增内容

  + Linux下的纯C客户端(仅实现了最简单的应用)

  + 登陆窗口（未绑定数据库）

## Linux下（纯C代码）

### 任务目标：实现最简单的服务器功能

### 任务实现

**文件：**QQ-version-of-Qt\PART01\Code\_in\_Ubuntu\server.c

#include <stdio.h>

#include <arpa/inet.h>

#include <sys/socket.h>

#include <stdlib.h>

#include <string.h> *//memset*

#include <unistd.h>

#include <netdb.h>

#include <sys/types.h>

#define INET\_ADDRSTRLEN 16

**int** main(**int** argc,**char** \*argv**[]**)

{

*/\**

*\**

*\*argc ---- 指示程序启动时命令行参数的个数*

*\**

*\*argv ---- 则包含具体的参数字符串*

*\*/*

*//预备工作：检查参数是否正确*

    if (argc !=2)

    {

        printf("请输入正确的参数\n如：./server 5005\n\n");

        return -1;*//非成功结束*

    }

*//第一步：创建插座*

*/\**

*\*插座特点*

*1.系统不自动分配端口*

*2.初始化插座是主动的，需要手动调节成被动（监听）模式*

*\*功能：创建一个用于网络通信的I/O描述符（插座）*

*\*int socket(int family, int type, int protocol)*

*\*int socket(协议族    ,插座类型,协议类别)*

*\*/*

**int** serverfd;   *//服务器的插座*

    if((serverfd = socket(AF\_INET,SOCK\_STREAM,0)) == -1)

    {

        perror("socket");

        return -1;

    }               *//插座创建成功*

*//第二步 把服务端用于通信的地址和端口绑定到socket上*

*/\**

*\**

*\*int bind(int sockfd,const struct sockaddr \*myaddr,socklen\_t addrlen)*

*\*int bind(socket插座,指向特定于协议的地址结构指针,该地址结构的长度)*

*\*功能 将本地协议地址与服务器插座文件符(serverfd)绑定*

*\*sockaddr\_in ：在IPv4因特网域(AF\_INET)中,套接字地址结构用sockaddr\_in命名*

*\*/*

**struct** sockaddr\_in servaddr;    *//定义服务端地址信息的数据结构*

    memset(&servaddr,0,sizeof(servaddr));*//初始化servaddr*

    servaddr.sin\_family = AF\_INET;

*//协议族，在socket编程中只能是AF\_INET*

    servaddr.sin\_addr.s\_addr = htonl(INADDR\_ANY);

*//任意ip地址*

*//inet\_addr("192.168.190.134"); // 指定ip地址*

    servaddr.sin\_port = htons(atoi(argv[1]));*//指定通信端口*

    if(bind(serverfd,(**struct** sockaddr \*)&servaddr,sizeof(servaddr)) != 0)

    {

        perror("bind");

        close(serverfd);

        return -1;

    }                               *//开始绑定*

*//第3步：把socket设置为监听模式*

*/\**

*\*int listen(int sockfd, int backlog)*

*\*int listen(socket监听插座，连接队列的长度)*

*\*功能 将插座由主动修改为被动*

*\*    使os为该插座设置一个连接队列，用来记录所有连接到该插座的连接*

*\*/*

    if(listen(serverfd,10) != 0)

    {

        perror("listen");

        close(serverfd);

        return -1;

    }

*//第4步：接受客户端的连接*

*/\**

*\*int accept(int sockfd,struct sockaddr \*cliaddr,socklen\_t \*addrlen);*

*\*int accept(socket监听插座,用于存放客户端插座地址结构,插座地址结构体长度);*

*\*功能 从已连接队列中取出一个已经建立的连接，如果没有任何连接可用，则进入睡眠等待*

*\*accept函数返回的是一个已连接套接字，这个套接字代表当前这个连接*

*\*/*

**int** clientfd;       *//声明客户端的插座*

**int** socklen = sizeof(**struct** sockaddr\_in);  *//获得插座地址结构struct sockaddr\_in的大小*

**struct** sockaddr\_in clientaddr;      *//声明客户端地址数据结构*

*//接受连接*

    clientfd = accept(serverfd,(**struct** sockaddr \*)&clientaddr,(socklen\_t\*)&socklen);

    printf("客户端（%s）已连接。\n",inet\_ntoa(clientaddr.sin\_addr));

*//第5步：与客户端通信，接收客户端发过来的报文后，回复ok。*

*/\**

*\*ssize\_t recv(int sockfd, void \*buf,size\_t nbytes, int flags);*

*\*ssize\_t recv(客户端插座, 指向接收网络数据的缓冲区,*

*接收缓冲区的大小(以字节为单位), 套接字标志(常为0));*

*\*返回值：成功接收到的数据字节数*

*\*ssize\_t send(int sockfd, const void\* buf, size\_t nbytes, int flags);*

*\*ssize\_t send(服务端插座, 待发送数据缓存区的地址,*

*发送缓存区大小(以字节为单位), 套接字标志(常为0));*

*\*功能：用于发送数据 注意：不能用TCP协议发送0长度的数据包*

*\*返回值：成功发送的字节数*

*\*/*

**char** buffer[1024];      *//声明缓冲区*

    while (1)

    {

**int** iret;

        memset(buffer,0,sizeof(buffer));*//初始化缓冲区*

*//接收客户端的请求报文*

        if((iret = recv(clientfd,buffer,sizeof(buffer),0)) <= 0)

        {

            printf("iret=%d\n",iret);

            break;

        }

        printf("接受：%s\n",buffer);*//显示接受到的消息*

*//向客户端发送响应结果*

        if ( (iret=send(clientfd,buffer,strlen(buffer),0))<=0)

        {

            perror("send");

            break;

        }

        printf("发送：%s\n",buffer);*//显示发送出去的消息*

    }

*// 第6步：关闭socket，释放资源*

    close(serverfd);

    close(clientfd);

}

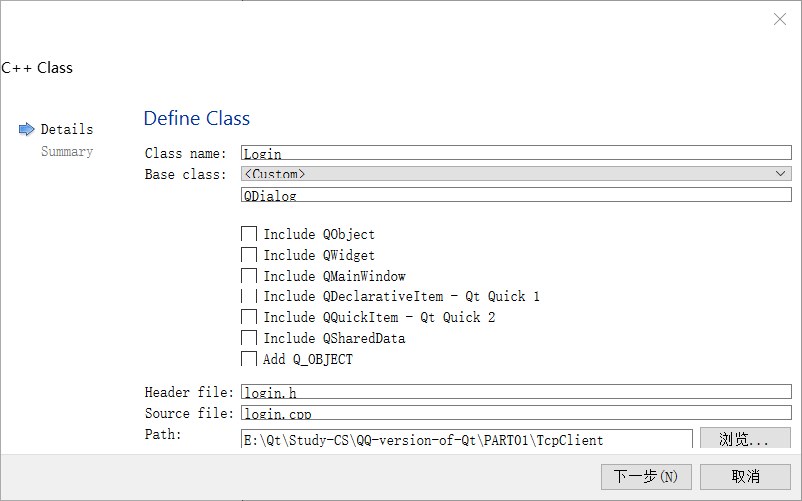
## Windows下（Qt代码）

### 任务目标：制作登录窗口

### 任务实现：

#### 创建Login类

右击项目名 🡪 add new 🡪 C++ 🡪 C++ Class 🡪如下图填写选择



#### 类的前置声明与预编译指令

文件：QQ-version-of-Qt\PART01\TcpClient\login.h

*//需要包含的类*

#include <QWidget>      *//窗口类*

#include <QDialog>      *//对话框类*

#include <QLabel>       *//标签类*

#include <QPushButton>  *//按钮类*

#include <QLineEdit>    *//输入行类*

#include <QMessageBox>  *//消息类*

#include <QGridLayout>  *//栅格布局管理器类*

*//类的前置声明*

**class** QLabel;

**class** QLineEdit;

**class** QPushButton;

#### 声明并初始化需要的窗口插件

文件：QQ-version-of-Qt\PART01\TcpClient\login.h

类的声明Class Login中：

**private:**

    QLabel \*usrLabel;*//文本：账号*

    QLabel \*pwdLabel;*//文本：密码*

    QLabel \*logoLabel;*//文本：logo*

    QLabel \*logup;*//文本：注册账号（用以日后连接数据库使用）*

    QLabel \*pswfind;*//文本：找回密码（用以日后连接数据库使用）*

    QLineEdit \*usrLineEdit;*//输入框：账号*

    QLineEdit \*pwdLineEdit;*//输入框：密码*

    QPushButton \*loginBtn;*//按钮：登陆*

    QPushButton \*exitBtn;*//按钮：退出*

文件：QQ-version-of-Qt\PART01\TcpClient\login.cpp

Login类中：

*//初始化窗口插件*

    logoLabel = new QLabel(this);

    usrLabel = new QLabel(this);

    pwdLabel = new QLabel(this);

    logup = new QLabel(this);

    pswfind = new QLabel(this);

    usrLineEdit = new QLineEdit(this);

    pwdLineEdit = new QLineEdit(this);

    loginBtn = new QPushButton(this);

    exitBtn = new QPushButton(this);

*//设置文本*

    logoLabel->setText("飞鸽传书");

    usrLabel->setText("用户名:");

    pwdLabel->setText("密码:");

    logup->setText("注册账户");

    pswfind->setText("找回密码");

    loginBtn->setText("登陆");

    exitBtn->setText("退出");

*//设置logo文本对齐方式（居中）*

    logoLabel->setAlignment(Qt::AlignHCenter);

#### 使用栅格布局管理器布局

文件：QQ-version-of-Qt\PART01\TcpClient\login.cpp

Login类中：

*//栅格布局管理器*

*//定义栅格布局管理器*

    QGridLayout \*loginLayout = new QGridLayout;

*//添加插件*

    loginLayout->addWidget(logoLabel,0,1);

    loginLayout->addWidget(usrLabel,2,0);

    loginLayout->addWidget(usrLineEdit,2,1);

    loginLayout->addWidget(logup,2,2);

    loginLayout->addWidget(pwdLabel,4,0);

    loginLayout->addWidget(pwdLineEdit,4,1);

    loginLayout->addWidget(pswfind,4,2);

    loginLayout->addWidget(loginBtn,6,0);

    loginLayout->addWidget(exitBtn,6,2);

*//设置管理器间距*

    loginLayout->setHorizontalSpacing(10);*//两列之间的间距*

    loginLayout->setVerticalSpacing(10);*//两行之间的间距*

    loginLayout->setContentsMargins(10,10,10,10);*//设置左侧、顶部、右侧和底部边距*

*//设置窗口大小*

    setMinimumSize(400,150);*//窗口最大大小*

    setMaximumSize(400,150);*//窗口最小大小*

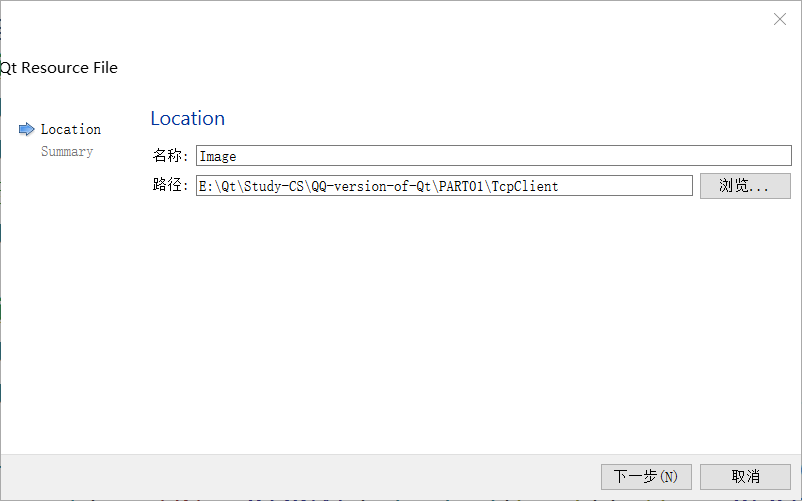
*//安装布局管理器*

    setLayout(loginLayout);

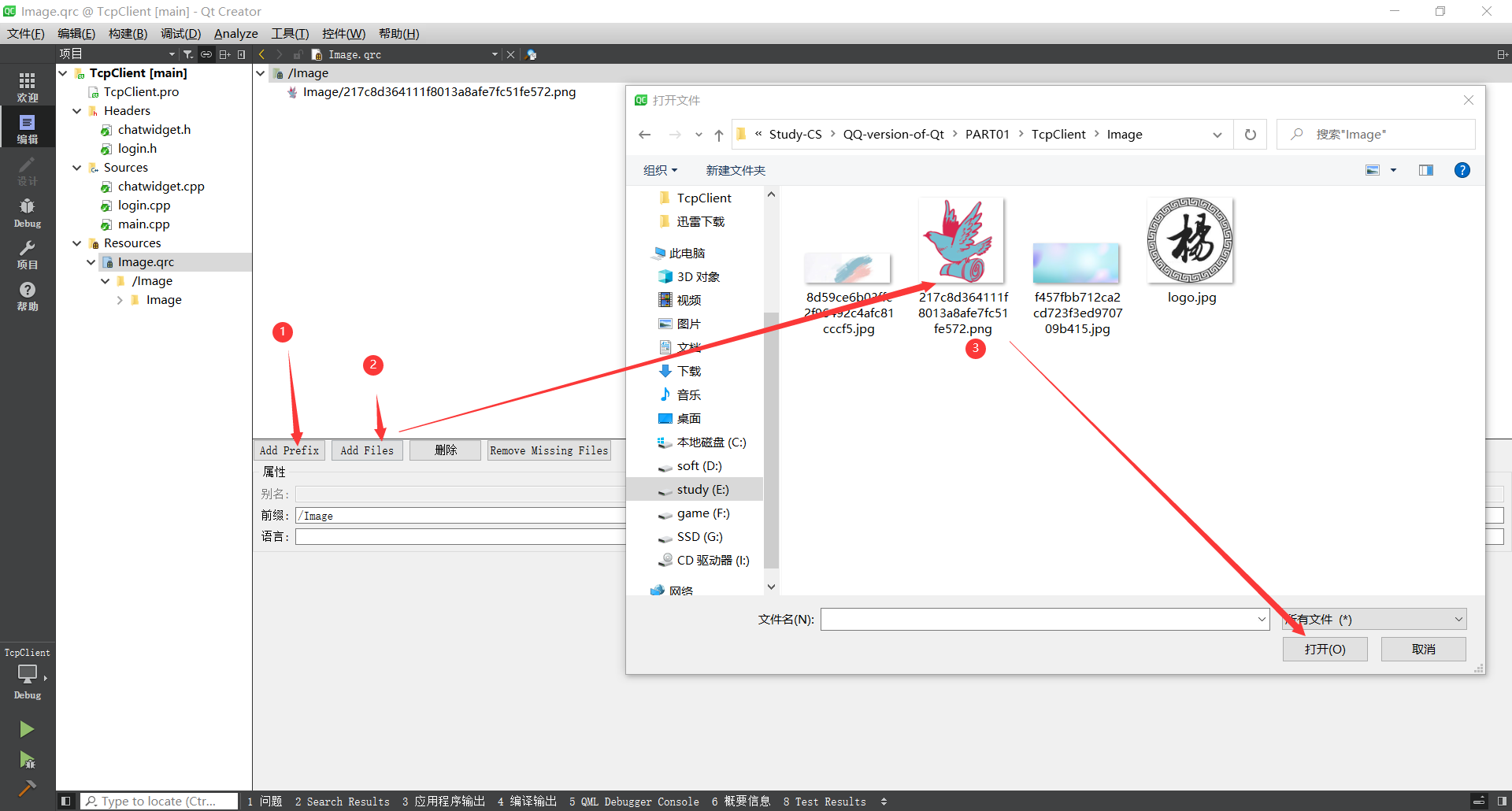
#### 设置窗口信息

文件：QQ-version-of-Qt\PART01\TcpClient\login.cpp

右击项目名 🡪 add new 🡪 Qt 🡪 Qt Resource File



右击资源文件Image.qrc 🡪Open in Editor



Login类中：

*//设置窗口名和窗口图标*

    setWindowTitle("飞鸽传书    登陆");

    setWindowIcon(QIcon(":/Image/Image/217c8d364111f8013a8afe7fc51fe572.png"));

#### 设置按钮相关的信号与槽

文件：QQ-version-of-Qt\PART01\TcpClient\login.h

类的声明Class Login中：

private slots:

**void** login();*//登陆进聊天室界面*

文件：QQ-version-of-Qt\PART01\TcpClient\login.cpp

Login类中：

*//设置信号与槽*

    connect(loginBtn,SIGNAL(clicked(**bool**)),this,SLOT(login()));

    connect(exitBtn,SIGNAL(clicked(**bool**)),this,SLOT(close()));

编辑login()函数

**void** Login::login()

{

    if(usrLineEdit->text().trimmed() == "yyz"

            && pwdLineEdit->text().trimmed() == "123456")

*//if(1) --- 为了便于测试，可以先设置if（1）*

    {

        accept();

    }

    else

    {

        QMessageBox::warning(this,"警告！","用户名或密码错误！",QMessageBox::Yes);

        usrLineEdit->clear();

        pwdLabel->clear();

        usrLineEdit->setFocus();

    }

}

#### 编辑main.cpp，实现从登陆界面到主界面的跳转

#include "chatwidget.h"

#include "login.h"

#include <QApplication>

**int** main(**int** argc, **char** **\***argv[])

{

    QApplication a(argc, argv);

    ChatWidget w;

    Login l;

    if(l.exec() == QDialog::Accepted)

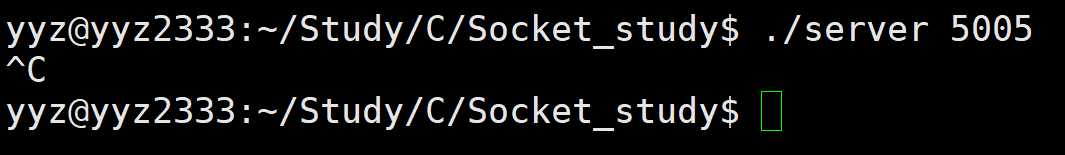
    w.show();

    return a.exec();

}

## 实现效果

### Server



### Client

