

基于 Socket 的网络通信原理与软件开发

张恒，桂任舟

（同济大学，电子信息与工程学院 上海市， 201804）

摘要：生活在信息时代，绝大多数人们对于网络聊天不会感到陌生，比较常见的如微信，QQ 等社交软件。网络通信有着即时性、可靠性、便捷性等优点。本文首先介绍了基于 socket 的网络通信原理，及其相关的技术发展。在此基础上，本文设计开发了一款即时通讯软件：TJQQ，是基于 UDP 协议的 socket 通信技术的聊天软件，其 UI 界面在 Qt 开发平台实现。软件实现的代码即将在 GitHub 上开源。

关键词：socket、计算机网络、Qt

Network communication and software development based on socket

Abstract: In the information age, the vast majority of people will not feel strange to online chat, some common social software such as wechat, QQ, etc. Network communication has the advantages of immediacy, reliability and convenience. Firstly, this paper introduces the principle of network communication based on socket and its related technology development. On this basis, we design and develop an instant messaging software: TJQQ, which is a chat software based on UDP socket communication technology. Its UI interface is implemented in Qt. The software implementation code will be open-source on GitHub.

Keywords: socket, computer network, Qt

1. 引言

随着计算机和通信技术的飞速发展，人们的生活发生了翻天覆地的变化。过去人们的通信主要靠书信，书信的通信方式，即时性太差，后来的通信方式又有了电报、电话等，但都有着各种各样的弊端，比如建设成本高、通信不稳定等。直到计算机技术和通讯技术的来临，各种通讯软件，如雨后春笋般的被开发出来，极大地改变了人们的信息交流方式和生活方式。只需要下载某个软件，动动手指，就可以和远方的朋友进行即时通讯。所有的网络终端计算机、智能手机、iPAD 等，如果能够正常进行网络通信，首先要遵循可靠的网络协议，其中，比较基础和常用的就是基于 socket 的通讯方式。

基于此，本文介绍了 socket 网络通信的基本原理，并在此基础上，在 Qt 软件上进行了开发，设计了一个多人即时聊天室。

2. Socket 网络通信相关原理

2.1 网络模型及 socket 概念

计算机网络是指由通信线路互相连接的许多自主工作的计算机构成的集合体，各个部件之间以何种规则进行通信，就是网络模型研究的问题。网络模型一般是指 OSI 七层参考模型和 TCP/IP 四层参考模型。这两个模型在网络中应用最为广泛。其基本 结构见图 1。

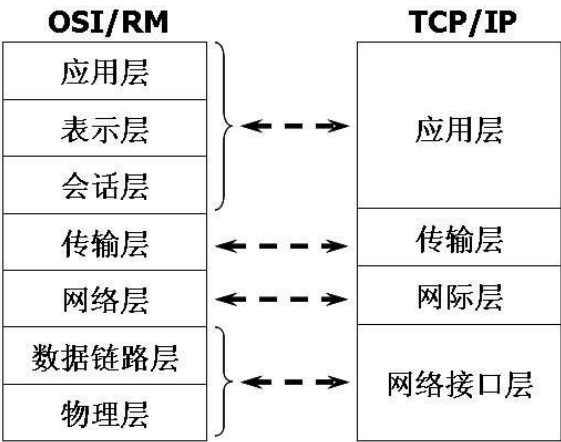


图 1. 计算机网络模型

Socket 起源于 UNIX，在 Unix 一切皆文件的思想下，进程间通信就被冠名为文件描述符（file descriptor），Socket 是一种“打开—读/写—关闭”模式的实现，服务器和客户端各自维护一个“文件”，在建立连接打开后，可以向文件写入内容供对方读取或者读取对方内容，通讯结束时关闭文件。Socket 是在应用层和传输层之间的一个抽象层，它把 TCP/IP 层复杂的操作抽象为几个简单的接口，供应用层调用实现进程在网络中的通信。在网络模型中，Socket 所在位置如图 2 所示。

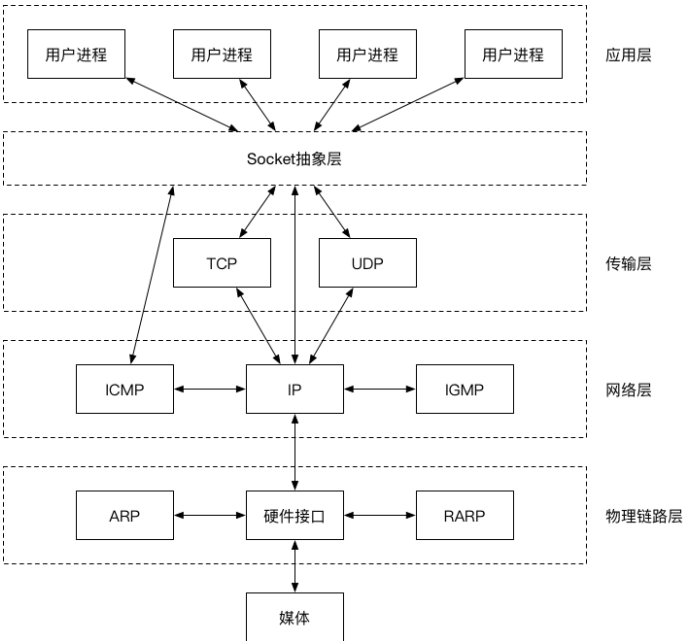


图 2. Socket 在网络模型中的简图

2.2 Socket 的分类

为了满足不同的通信程序对通信质量和性能的要求，一般的网络系统提供了三

种不同类型的套接字，以供用户在设计网络应用程序时根据不同的要求来选择。这三种套接为流式套接字（SOCK-STREAM）、数据报套接字（SOCK-DGRAM）和原始套接字（SOCK-RAW）。^[1]

（1）流式套接字。它提供了一种可靠的、面向连接的双向数据传输服务，实现了数据无差错、无重复的发送。流式套接字内设流量控制，被传输的数据看作是无记录边界的字节流。在 TCP/IP 协议簇中，TCP 协议实现字节流的传输，当用户想要发送大批量的数据或者对数据传输有较高的要求时，可以使用流式套接字。^[2]

（2）数据报套接字。它提供了一种无连接、不可靠的双向数据传输服务。数据包以独立的形式被发送，并且保留记录边界，不提供可靠性保证。数据在传输过程中可能会丢失或重复，不能保证在接收端按发送顺序接收数据。在 TCP/IP 协议簇中，使用 UDP 协议来实现数据报套接字。在出现差错的可能性较小或允许部分传输出错的应用场合，可以使用数据报套接字进行数据传输，这样通信的效率较高。^[2]

（3）原始套接字。该套接字允许对较低层协议（如 IP 或 ICMP）进行直接访问，常用于网络协议分析，检验新的网络协议实现，也可用于测试新配置或安装的网络设备。

2.3 Socket 的 UDP 通信原理

Socket 是应用层和传输层之间的一层抽象协议，可以用来进行进程间通信，一般有 UDP 和 TCP 两种通信方式，前者速度稍快，稳定性不好，无法丢包重传。后者速度稍慢一点，但稳定性很好，可以丢包重传。相比于 TCP 协议，UDP 编程步骤要简单许多。

Socket 函数对应于普通文件的打开操作。普通文件的打开操作返回一个文件描述字，而 socket() 用于创建一个 socket 描述符（socket descriptor），它唯一标识一个 socket。这个 socket 描述字跟文件描述字一样，后续的操作都有用到它，把它作为参数，通过它来进行一些读写操作。正如可以给 fopen 的传入不同参数值，以打开不同的文件。创建 socket 的时候，也可以指定不同的参数创建不同的 socket 描述符，socket 函数的三个参数分别为：

domain: 即协议域，又称为协议族（family）。常用的协议族有，AF_INET、AF_INET6、AF_LOCAL（或称 AF_UNIX，Unix 域 socket）、AF_ROUTE 等等。协议族决定了 socket 的地址类型，在通信中必须采用对应的地址，如 AF_INET 决定了要用 ipv4 地址（32 位的）与端口号（16 位的）的组合、AF_UNIX 决定了要用一个绝对路径名作为地址。

type: 指定 socket 类型。常用的 socket 类型有，SOCK_STREAM、SOCK_DGRAM、SOCK_RAW、SOCK_PACKET、SOCK_SEQPACKET 等等（socket 的类型有哪些？）。

protocol: 顾名思义，就是指定协议。常用的协议有，IPPROTO_TCP、IPPROTO_UDP、IPPROTO_SCTP、IPPROTO_TIPC 等，它们分别对应 TCP 传输协议、UDP 传输协议、SCTP 传输协议、TIPC 传输协议（这个协议我将会单独开篇讨论！）。

当我们调用 socket 创建一个 socket 时，返回的 socket 描述字它存在于协议族（address family，AF_XXX）空间中，但没有一个具体的地址。如果想要给它赋值一个地址，就必须调用 bind() 函数，否则就当调用 connect()、listen() 时系统会自动随机分配一个端口。

一些主要函数包括：

（1）bind() 函数

正如上面所说 bind() 函数把一个地址族中的特定地址赋给 socket。例如对应 AF_INET、AF_INET6 就是把一个 ipv4 或 ipv6 地址和端口号组合赋给 socket。

```
int bind(int sockfd, const struct sockaddr *addr, socklen_t addrlen);
```

(2) listen()、connect()函数

如果作为一个服务器，在调用 socket()、bind()之后就会调用 listen()来监听这个 socket，如果客户端这时调用 connect()发出连接请求，服务器端就会接收到这个请求。listen 函数的第一个参数即为要监听的 socket 描述字，第二个参数为相应 socket 可以排队的最大连接个数。socket()函数创建的 socket 默认是一个主动类型的，listen 函数将 socket 变为被动类型的，等待客户的连接请求。connect 函数的第一个参数即为客户端的 socket 描述字，第二参数为服务器的 socket 地址，第三个参数为 socket 地址的长度。客户端通过调用 connect 函数来建立与 TCP 服务器的连接。

```
int listen(int sockfd, int backlog);
```

```
int connect(int sockfd, const struct sockaddr *addr, socklen_t addrlen);
```

(3) accept()函数

TCP 服务器端依次调用 socket()、bind()、listen()之后，就会监听指定的 socket 地址了。TCP 客户端依次调用 socket()、connect()之后就想 TCP 服务器发送了一个连接请求。TCP 服务器监听到这个请求之后，就会调用 accept()函数取接收请求，这样连接就建立好了。之后就可以开始网络 I/O 操作了，即类同于普通文件的读写 I/O 操作。

```
int accept(int sockfd, struct sockaddr *addr, socklen_t *addrlen);
```

(4) read()、write()等函数

开发语言不同可能读写函数也就不同，只要把自己想要发送的消息，以字节流的方式写入 Socket 或者从 Socket 读出来即可实现网络的 I/O 操作。

(5) close()函数

在服务器与客户端建立连接之后，会进行一些读写操作，完成了读写操作就要关闭相应的 socket 描述字，好比操作完打开的文件要调用 fclose 关闭打开的文件。

以上是一些常见的 socket 编程的介绍，其中基于 UDP 协议的 socket 编程与 TCP 协议有些许的不同，使用 Socket 进行 UDP 通信的流程如下：

- (1) 创建一个 socket 对象
- (2) 准备好目标主机 IP 和目标端口
- (3) 绑定本机的端口（不必须）
- (4) 发送/接收数据
- (5) 关闭套接字

创建套接字对象，要进行 Socket 通信，我们需要使用 socket 模块，首先需要创建一个 Socket 对象。

```
# 使用 UDP 进行通信
```

```
udpSocket = socket.socket(socket.AF_INET, socket.SOCK_DGRAM)
```

3. 基于 Qt 的软件实现

3.1 开发平台及运行环境

操作系统：Windows10_64bit

开发平台：Qt5.9

编译器：MSVC2017 64bit (MSVC15.0)

3.2 通信部分的设计框架

3.2.1 实现基本聊天功能

在 widget 中, 定义枚举 `enum MsgType {Msg,UsrEnter,UsrLeft}` 分别代表: 聊天信息、新用户加入、用户退出。

3.2.2 声明聊天的方法

在 widge.h 中

```
public:
void sndMsg(MsgType type); //广播 UDP 消息
void usrEnter(QString username); //处理新用户加入
void usrLeft(QString username,QString time); //处理用户离开
QString getUsr(); //获取用户名
QString getMsg(); //获取聊天信息
private:
QUdpSocket * udpSocket; //udp 套接字
qint16 port; //端口
QString uName; //用户名
void ReceiveMessage(); //接受 UDP 消息
```

3.2.3 widget.cpp 中实现

构造函数中:

```
this->uName = username; //获取用户名
udpSocket = new QUdpSocket(this);
udpSocket->bind(port,QUdpSocket::ShareAddress |QUdpSocket::ReuseAddressHint);
发送消息函数 sndMsg
```

```
void Widget::sndMsg(MsgType type)
{.....
.....
udpSocket->writeDatagram(data,data.length(),QHostAddress::Broadcast,port);
}
```

接受消息 ReceiveMessage

```
void Widget::ReceiveMessage();
```

获取聊天信息

```
//获取聊天信息
QString Widget::getMsg();
```

3.3 基本开发步骤

3.3.1 创建新项目

第一步打开 Qt Creator, 点击新建 NewProject, 如图 3 所示。

Application -> Qt Widgets Application -> choose

创建项目名称例如: TJQQ, 路径选择。

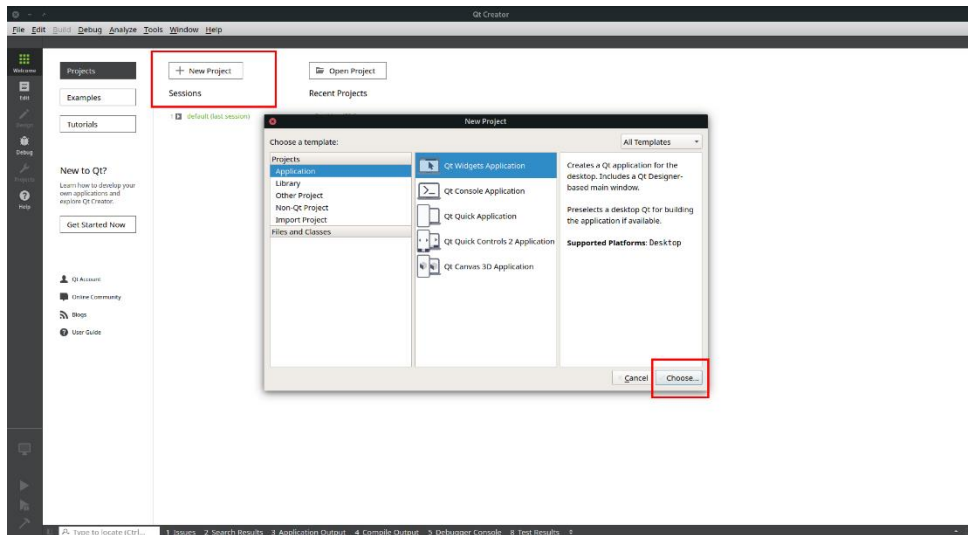


图 3. TJQQ 项目创建

选择套件，点击下一步。选择基类 `QWidget`，然后点击下一步，会自动会生成相应的.h 文件和.cpp 文件，如图 4 所示。然后点击完成，至此项目创建完毕。

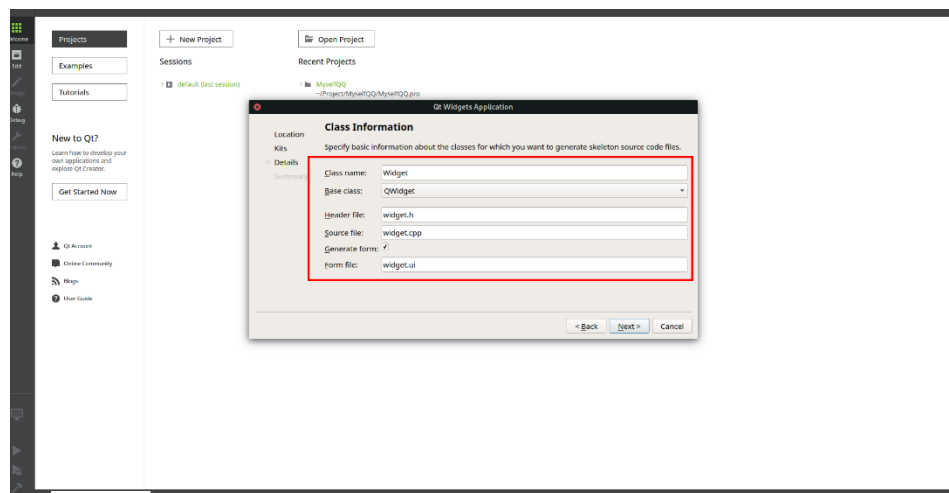


图 4. Widget 文件的创建

3.3.2 创建对话框列表

添加新文件，对话框列表类 `DialogList`，右击项目名，在弹出的快捷菜单中选择“添加新文件...”菜单项，在弹出的对话框中选择“Qt”选项。选择 Qt 设计师界面类，单击“Choose...”按钮；

界面模板选择 `Widget`，点击下一步，类名填写 `DialogList`，点击下一步。在汇总中单击“完成”按钮，系统会为我们添加“`dialoglist.h`”头文件和“`dialoglist.cpp`”源文件以及 `dialoglist.ui` 设计文件。

3.3.3 设计对话框列表 UI

设计对话框的 UI，如图 5 所示。该窗口的大小为 `250*700`，其中的主要控件是 `QToolBox`，修改该控件的 `currentItemText` 为“成员”，`QToolBox` 默认生成的第二页删除掉。在成员里我们放入一个 `Widget` 做布局操作，可以先利用一些测试控件放入到其中，然后做垂直布局，然后把测试的控件删除掉，这时该 `Widget` 中就有了一个 `layout` 布局。

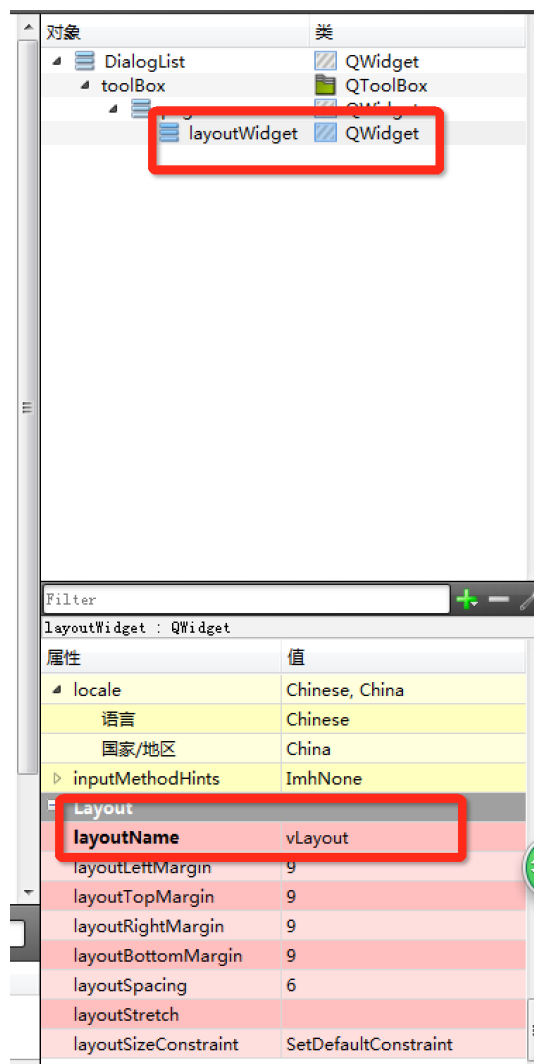


图 5. 对话框设计布局

3.3.4 资源导入

向项目中导入资源，对应按钮需要的图片作为头像图标使用，我选择的是同济大学校徽，以及自己的微信头像，大小重新调整在 100*100 左右)。如图 6，添加新文件 - Qt - Qt Resource 点击 choose 名称 res 下一步，点击完成，生成 res.qrc 文件。右击 res.qrc，点击 open in Editor，添加前缀 /。

添加文件 - 将准备好的文件选中，点击打开，添加成功。添加完前缀后，就是添加文件，此时要提前把相应的文件放到 TJQQ 项目的目录下，否则无法添加。

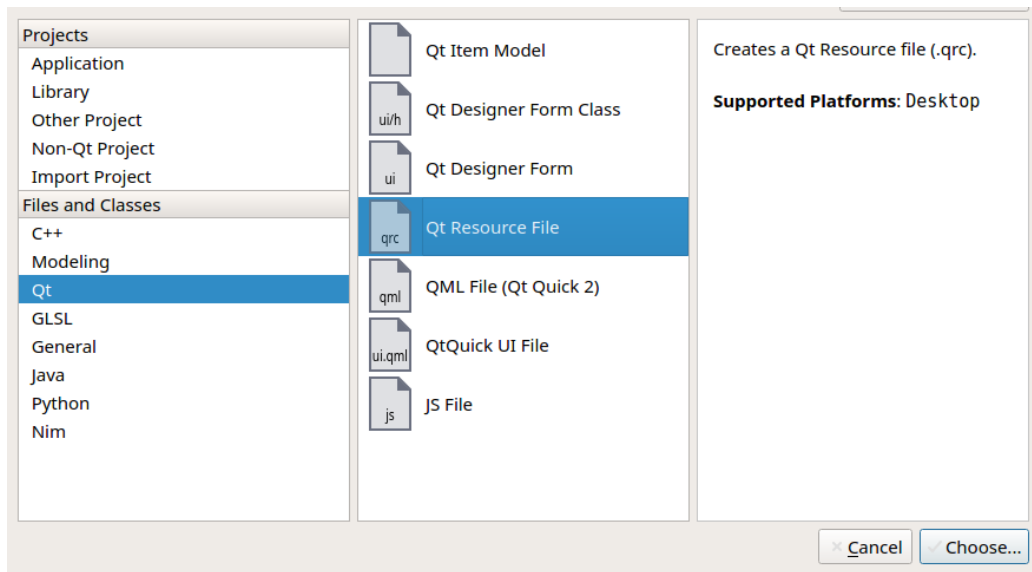


图 6. 资源导入操作步骤

3.3.5 设计聊天窗口

(1) 界面

双击 `widget.ui` 文件进入设计模式，界面宽度属性分别设置为 730 和 450，向界面中拖入部件并且进行设置。对各个控件设置在 Qt 里对他们的属性设置以及控件类型

(2) 字体大小下拉框设置

如图 7，界面上 5 号控件设置字体大小，设置区间为 8~22（与腾讯 QQ 软件完全相同），双击该部件，点击 + 号按钮添加新项目如图。`currentIndex` 属性设置为 4，即默认为 12 号字

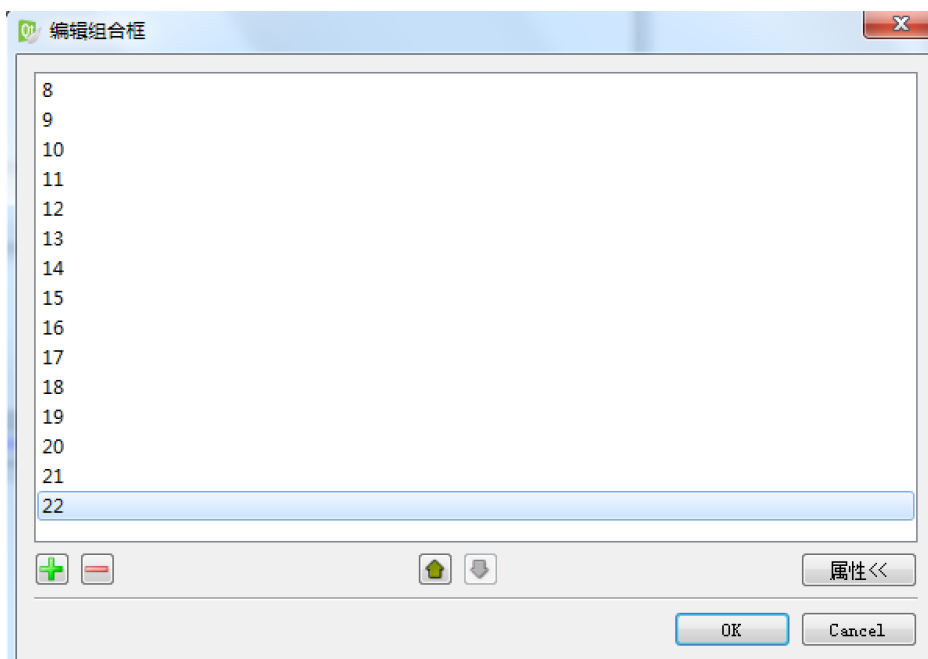


图 7. 聊天窗口字体设计

(3) TableWidget 设置

显示用户列表的 `TableWidget` 控件，将 `selectionModel` 属性选择为 `SingleSelection`（带

有选中效果)，将 `selectBehavior` 选择为 `SelectRows`（选中整行），取消选中的 `showGrid`（表格显示）双击 `TableWidget` 部件，添加“用户名”列。至此，所有弹出的聊天信息窗口全部设计完毕。

3.3.6 关联按钮与聊天窗口

添加按钮点击事件，下一步就是要关联起聊天的列表窗口和具体的聊天信息窗口了，也就是点击按钮弹出窗口。

```
//添加点击事件
for(int i = 0 ; i < vToolBtn.size();i++)
{
    connect(vToolBtn[i],&QToolButton::clicked,this,[=]()
    {
        Widget *chatWidget = new Widget(0,vToolBtn[i]->text());
        chatWidget->setWindowTitle(vToolBtn[i]->text());
        chatWidget->setWindowIcon(vToolBtn[i]->icon());
        chatWidget->show();
    });
}
```

3.3.7 辅助功能

除了以上基本功能之外，还添加了一些辅助功能，增加了良好的用户体验。这些辅助功能包括：字体设置、字号设置、加粗、倾斜、下划线、设置文本颜色 保存聊天记、清空聊天记录

3.4 结果演示

所有开发工作完成，开始调试，不断改进，最终，本文设计开发的 TJQQ 可以实现多人聊天。在测试环节，分别以：“同济大学校长”、“桂老师”“1941712 张恒”等 3 人，进行了即时消息交流。经测试，程序运行良好，实现了基本功能。演示效果，分别如图 8，图 9，图 10，图 11 所示：

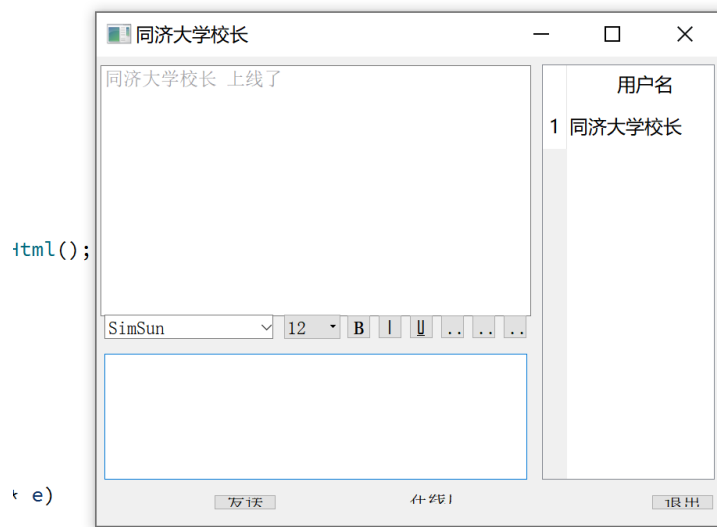


图 8. 用户上线演示图

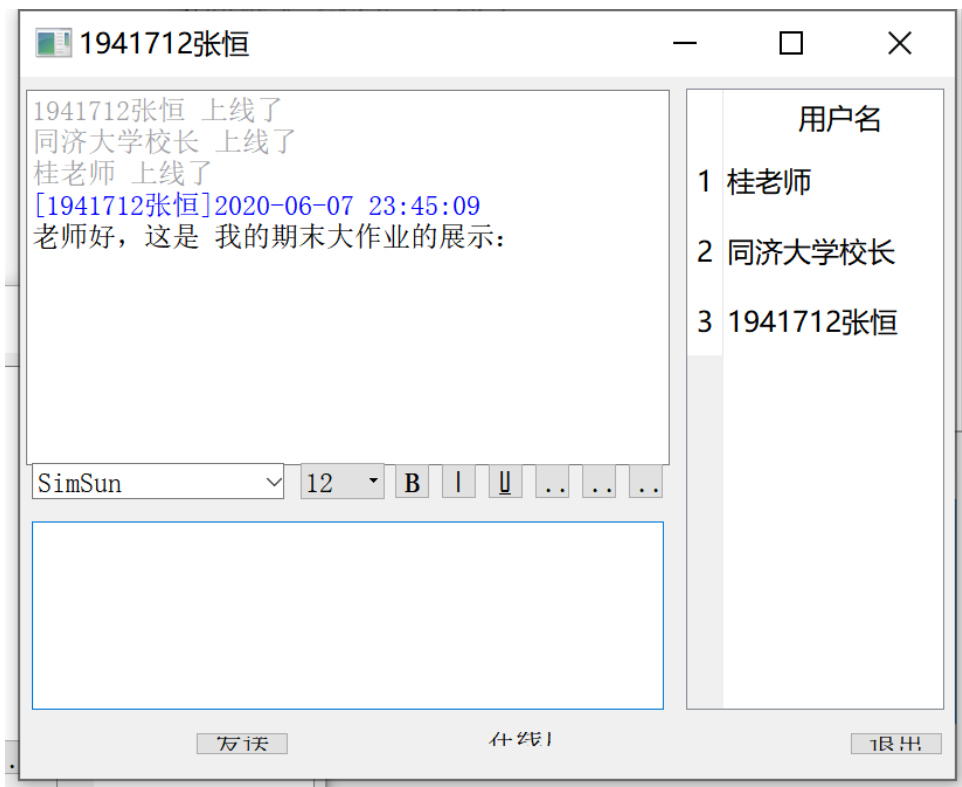


图 9. 用户“张恒”发送消息演示图

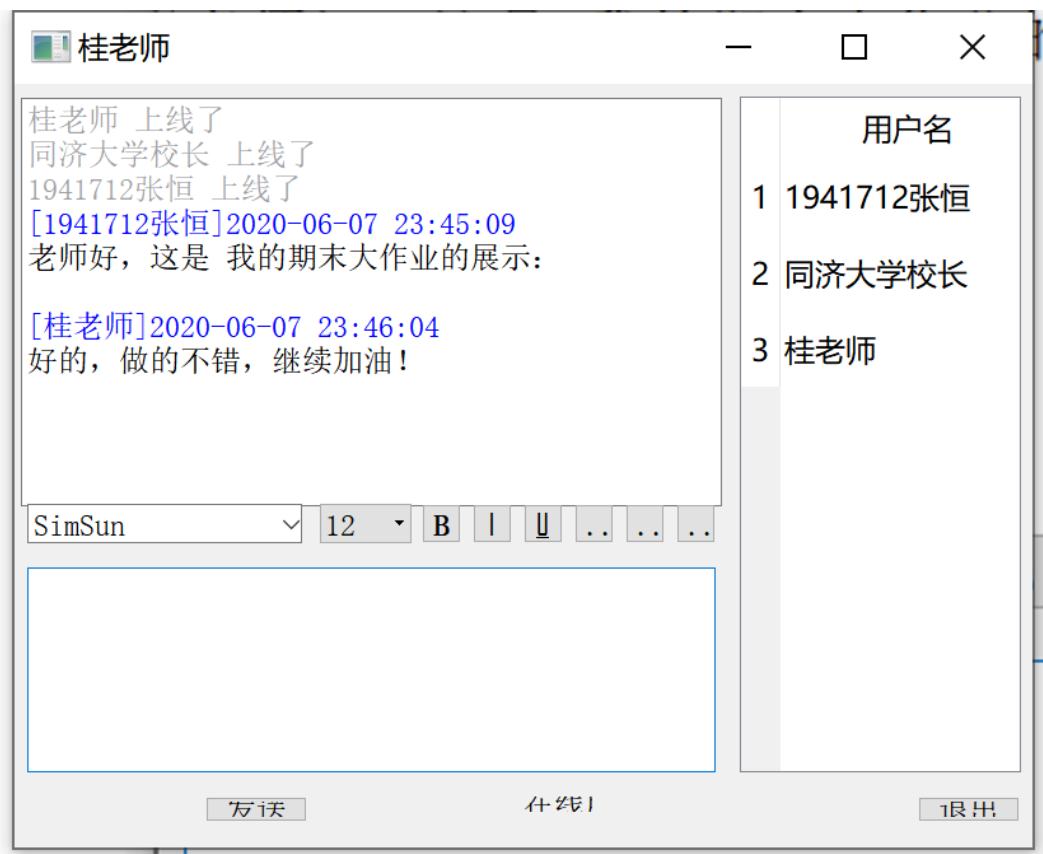


图 10. 用户“桂老师”窗口演示图



图 11. 收到用户“校长”的消息 演示图

4. 结束语

在熟练掌握 socket 网络通信基本原理的基础上，本文设计开发了一款名为 TJQQ 的即时通信软件，对 socket 编程进行了良好实践，实现了基本功能。后续工作是继续对 QT 界面进行优化，当前版本有些辅助功能未能较好的实现，比如用户头像未能显示，且 UI 界面的美观性还需进一步提高。

参考文献

- [1] Shenzhen Anhaoruihe Electronics Co. Ltd.; Patent Issued for Rotary Socket (USPTO 10,608,395)[J]. Electronics Newsweekly,2020
- [2] Lanka Mahesh,Jose Luis Calvo Guirado,Sagrika Shukla,Varun Raj Kumar,Yuvika Raj Kumar. Clinical and radiographic findings without the use of bone substitute materials in extraction sockets and delayed implant placement- A case series[J]. Journal of Oral Biology and Craniofacial Research,2020,10(2)
- [3]谢希仁,计算机网络第五版,电子工业出版社