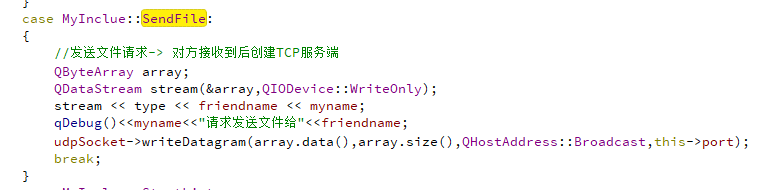


**文件发送**

文件发送采用TCP实现文件可靠传输，

文件发送的数据格式：为总数据长度 文件名长度 文件名 文件内容

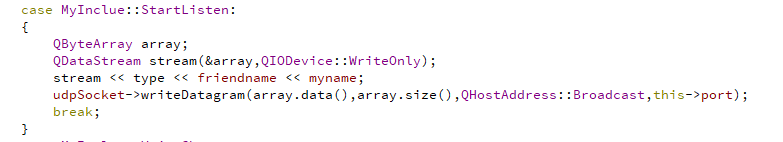
1.发送方UDP发送 请求发送文件的请求



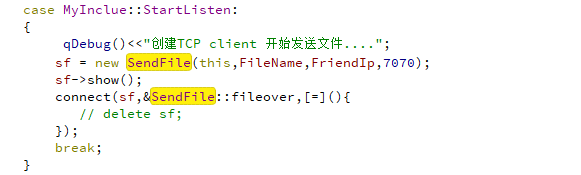
1. 接收方接收到对方发过来的发送文件请求



1. 接收方创建TCP服务端，通过UDP告知刚才的发送方，已经准备好接收文件



4.发送方，得知接收方准备好接收了就创建tcp客户端



5.发送方，在成功连接上接收方创建TCP服务端后开始发送文件



1. 文件传输进度更新

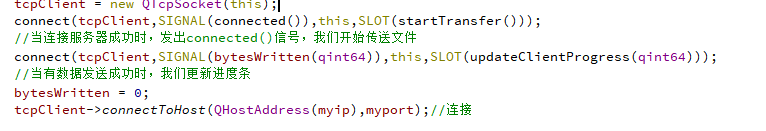
在发送前先读取并保存文件大小，数据发送发每次会发送loadSize = 4\*1024;

大小的数据，每接收到一次就用总数减掉当前接收的大小，然后实时更新文件传输进度，



**1.TCP文件传输实现原理**

**在Qt中，要使用TCP通信必须要在.pro文件中加入network模块。在客户端部分，主要是用到QTcpsocket类创建socket对象去连接服务器端口，连接成功后即可正常传输数据。在判断连接状态时，主要会用到connected()、disconnected()、stateChanged()等相关信号 ；在数据读写的过程中，主要会用到类QIODevice中的bytesWritten()、readyRead()等信号。在创建socket对象后， 可以绑定信号后直接连接服务器，其代码示例如下**



**2. 自己封装数据进行传输**

**在TCP传输的过程中，可以自己指定对应的数据包进行传输。下面用代码示例一般数据的传输。**



3. 用TCP传输文件

在传送文件的过程中，要用到几个变量用来统计传送的进度，包括：

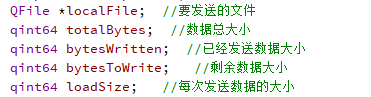
传送的总大小、

文件名、

已发送数据的大小

待发送数据的大小

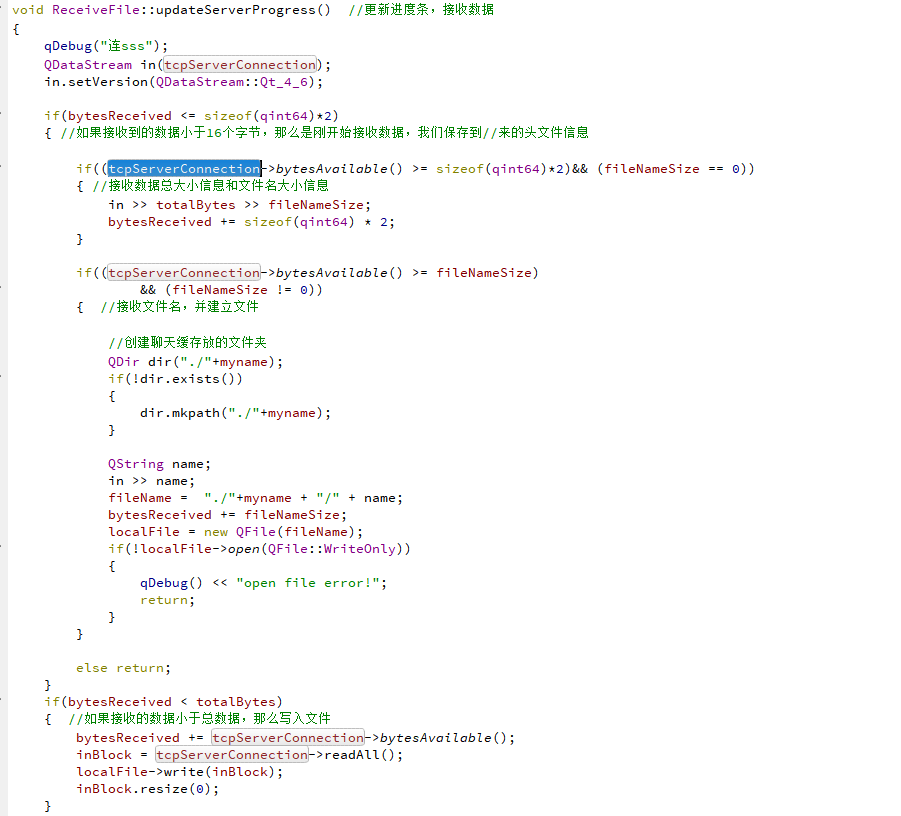
每次发送的最大数据大小等



下面的代码示例出了文件的传输过程，在传输过程中可以设置文件的进度条。



4. 文件的接收

在TCP数据传输文件时，如何封装发送的，接收时就如何解包，数据格式一定要一致，否则就会出错。在TCP通信过程中，传输数据一定要定义好格式，以什么样的格式发送就以什么样的格式接收，否则就会出现数据错误。

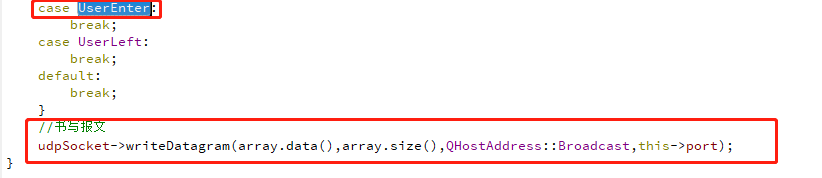
UDP解析

基础的聊天功能是基于UDP协议的，也就是面向无连接的，这样收发消息很方便，但是这样的话对用户登录和下线的处理比较麻烦

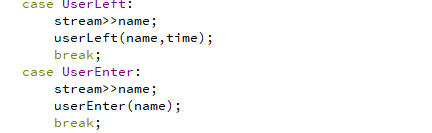
1. 登陆和退出

用户上线时主动发送上线通知，通知当前在线的其他用户

报文内容：上线通知 UserEnter 上线用户的ID



其他用户收到通知后，都知道当前有新用户上线会把他添加到自己的在线列表。



但是新上线的用户无法获知在他之前上线的用户所以当用户在获取到其他人上线的信息时也会调用sendMsg(UserEnter);发送自己也在先的信息，相当于所有人都发送进入，这样子所有用户都能知道当前所有在线的用户

用户下线时主动发送下线通知，通知当前在线的其他用户

报文内容：下线通知 UserLeftr 下线用户的ID

1. 发送消息模块

这里就是将输入的消息传送给sendMsg(QString msg)这个函数，指定需要发送的客户端的地址，封装好后，发送给服务端。

数据类型为：

类型（消息）

发送者name (对于接收者来说这个就是自己的ID)

接收者name (对于接收者来说这个就是发送者ID)



（3）接收消息的模块

这里接收到消息之后解析出

信息类型mytpe

接收者名字 name

发送者名字 fname

因为当前消息模式为点对点私聊所以接收者应判断接收者的name和自己的是否一直，不一致则不是发给自己的消息



QFontComboBox()主要是用来选择字体。实现聊天字体选择

1.设置和获取当前的字体

QFontComboBox.setCurrentFont(self, f: QtGui.QFont) #设置字体

QFontComboBox.currentFont() #获取字体

其他设置为下面的代码

