基于mqtt的ftp实现

# 一、设计说明

## 结构原理

1. MQTT Broker作为FTP的服务器
2. MQTT Client作为FTP的客户端
3. 主题“command”模拟FTP的命令端口
4. 主题“dataTransfer”模拟FTP的数据端口
5. 订阅用于：client在“dataTransfer”接收服务器的文件数据，在“command”接收命令返回结果
6. 发布用于：client在“dataTransfer”向服务器传送文件数据，在“command”发送请求

Broker(作为FTP服务器)

Client(作为FTP客户端)

Publish（上传文件/发送指令）

Subscribe(下载文件/接收指令返回结果)

## 框架选择

1. Broker端

选用Aedes → <https://www.npmjs.com/package/aedes#client> ，Mosca作者的新库

1. Client端

选用QT的官方的MQTT库 → [https://doc.qt.io/QtMQTT/qtmqtt-index.html#](https://doc.qt.io/QtMQTT/qtmqtt-index.html)

1. 使用MQTT3.1版本，消息均采用QoS2

## 消息格式

由于QT的MQTT不支持MQTT5，从而无法使用MQTT5的消息属性特性，自定义了消息格式：

msgNo--data

1. msgNo：消息编号，每一条消息都有自己的编号，通过其值段区分不同的命令，当客户端接受到消息并返回结果时，在结果前附加上相同的编号以供客户端识别。
2. data数据段

# 二、实现

## 环境

1. Broker：windows 8.1 x64，nodejs 10.3.0， Aedes 0.39.0
2. Client：windows 10 x64，C++ mingw32(QT5.12 32位）

## 命令说明

1. ls [path]：列出服务器文件夹下的文件
2. get files：取得服务器文件下的文件
3. put：上传文件

# 三、文件目录

Mqtt-ftp

﹂ ftp-broker：broker源码

﹂ ftp-main：ftp服务器的默认文件夹

﹂ ftp-client：client源码

﹂ ftp-client-release：client打包发布文件

\* client的本机文件夹被定义为%USERPROFILE%（存放接收的文件）

# 四、运行

## 运行前

设置客户端的hosts文件，在末尾添加一行：

|  |
| --- |
| 服务器在局域网的ip 自定义的服务器的主机名 |

如：192.168.1.12 tsyshiu.cn

## 服务器端运行

|  |
| --- |
| cd mqtt-ftp/ftp-broker  npm install  node broker.js |

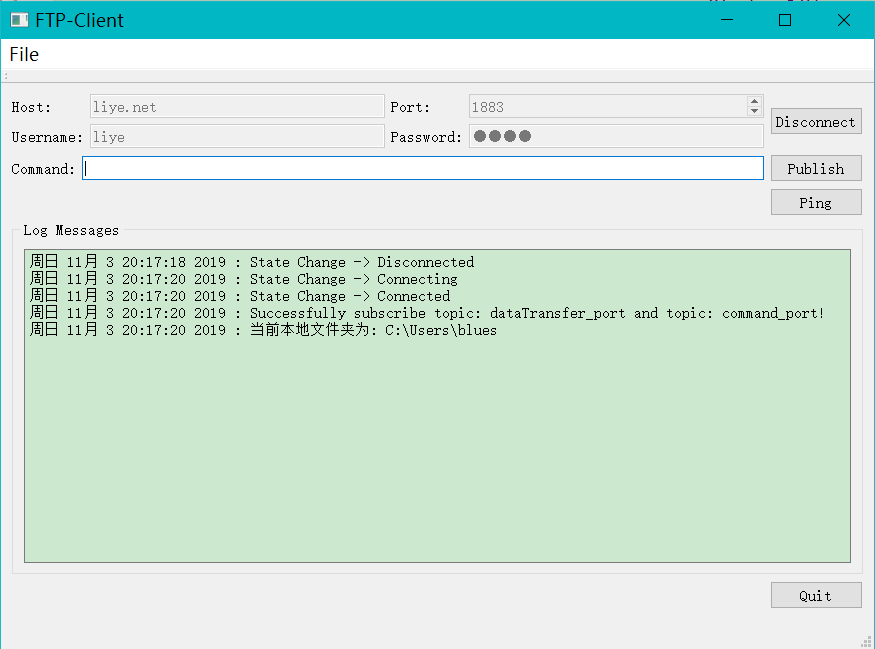
## 客户端运行

源码自行用qt编译

或者直接执行ftp-client-release/ftp-client.exe

# 五、展示

## 连接及订阅



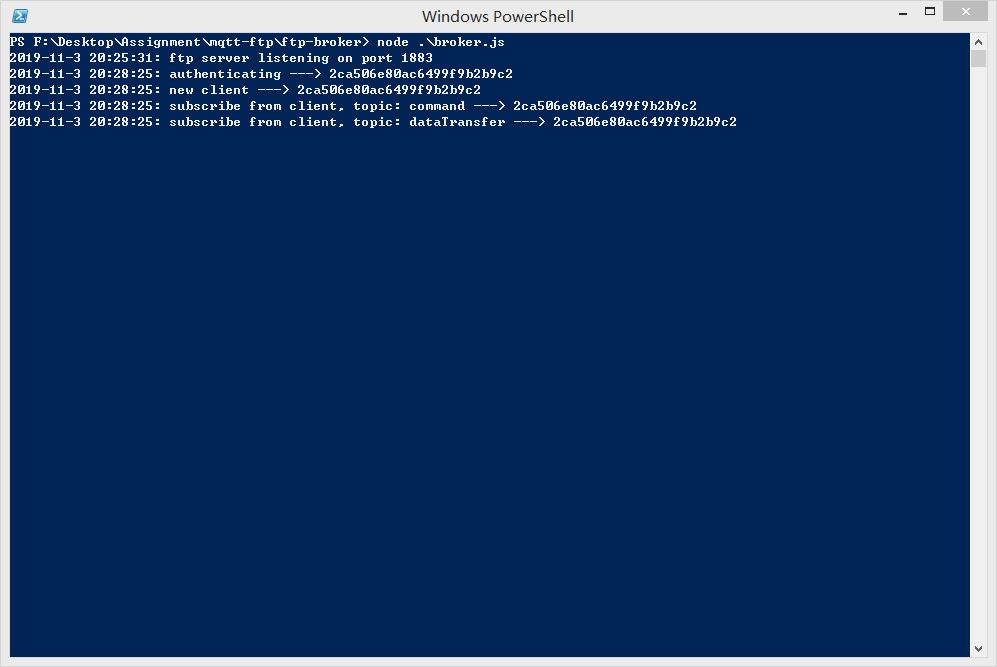
Host：MQTT Broker的主机名

Port：MQTT Broker的端口号

Username、Password：账号密码

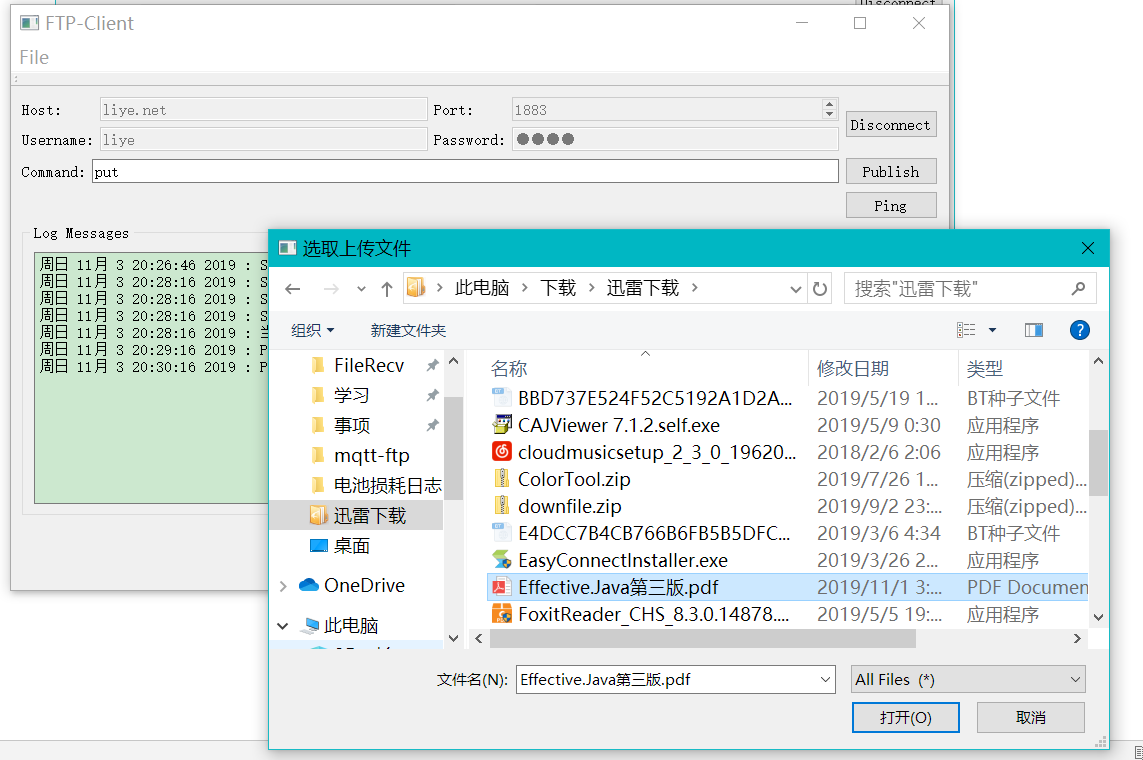
操作流程：Connect🡪输入命令🡪publish🡪Disconnect

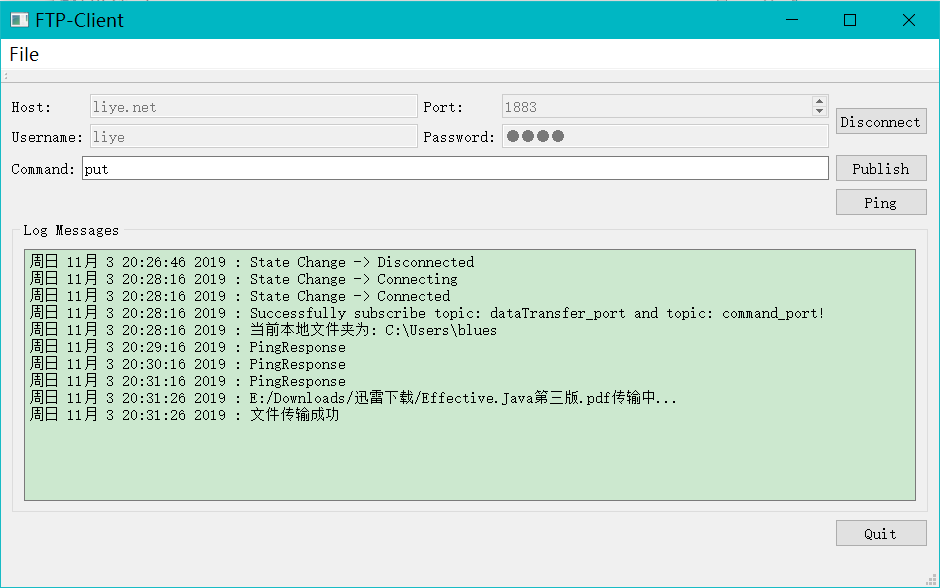
服务器端显示：



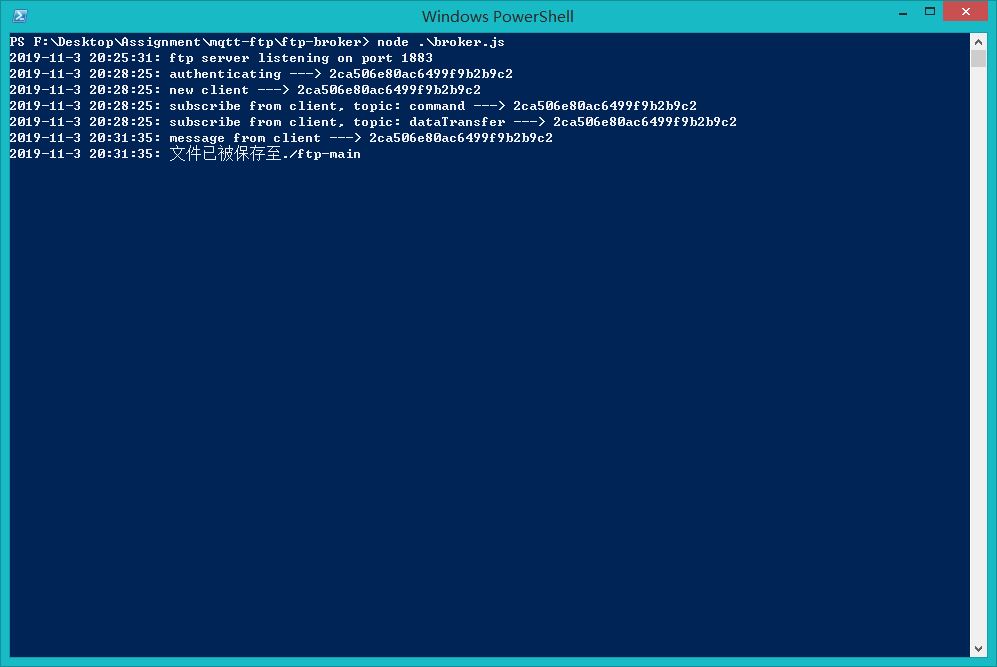
## 上传本地文件（上传一个pdf示例）

输入put命令，publish，选取文件pdf：

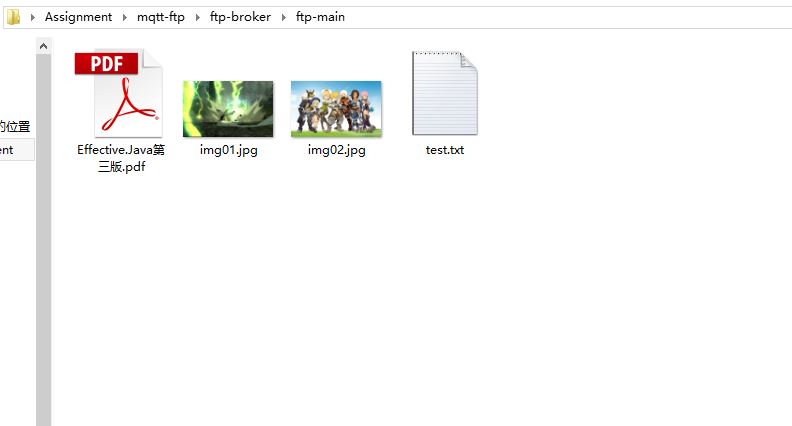




服务器端显示：

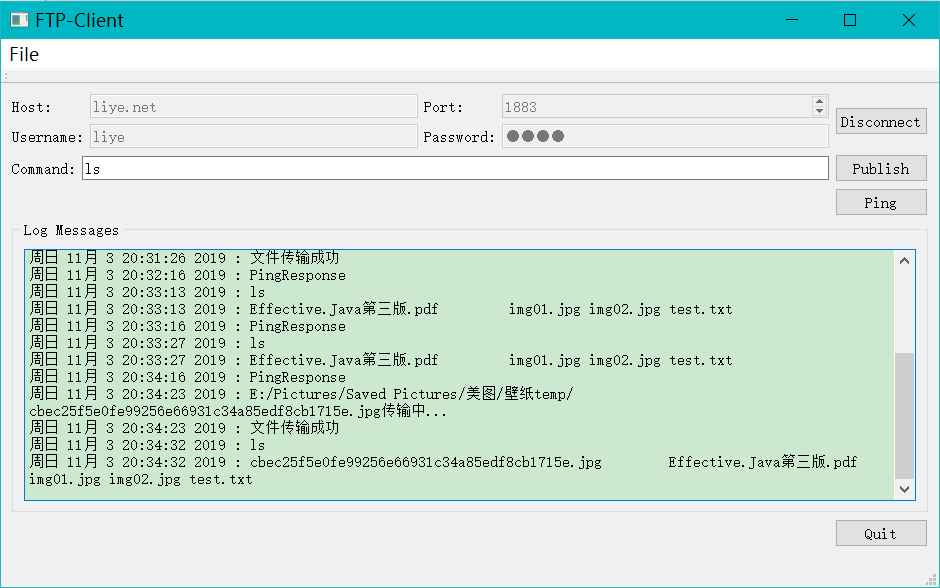


服务器文件夹传输后为：

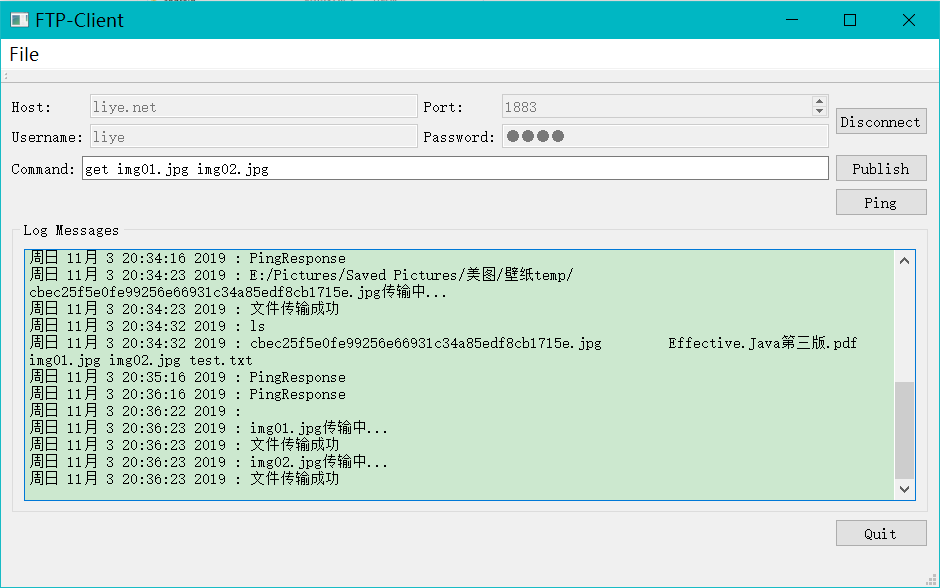


## 下载服务器文件（下载两个img示例）

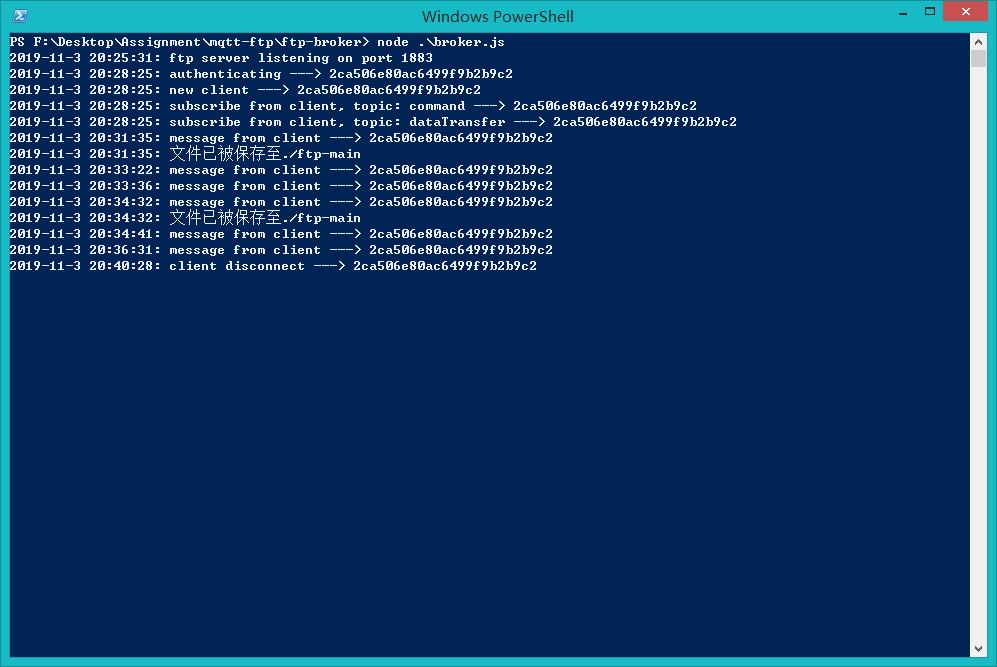
先用ls查看服务器下文件：



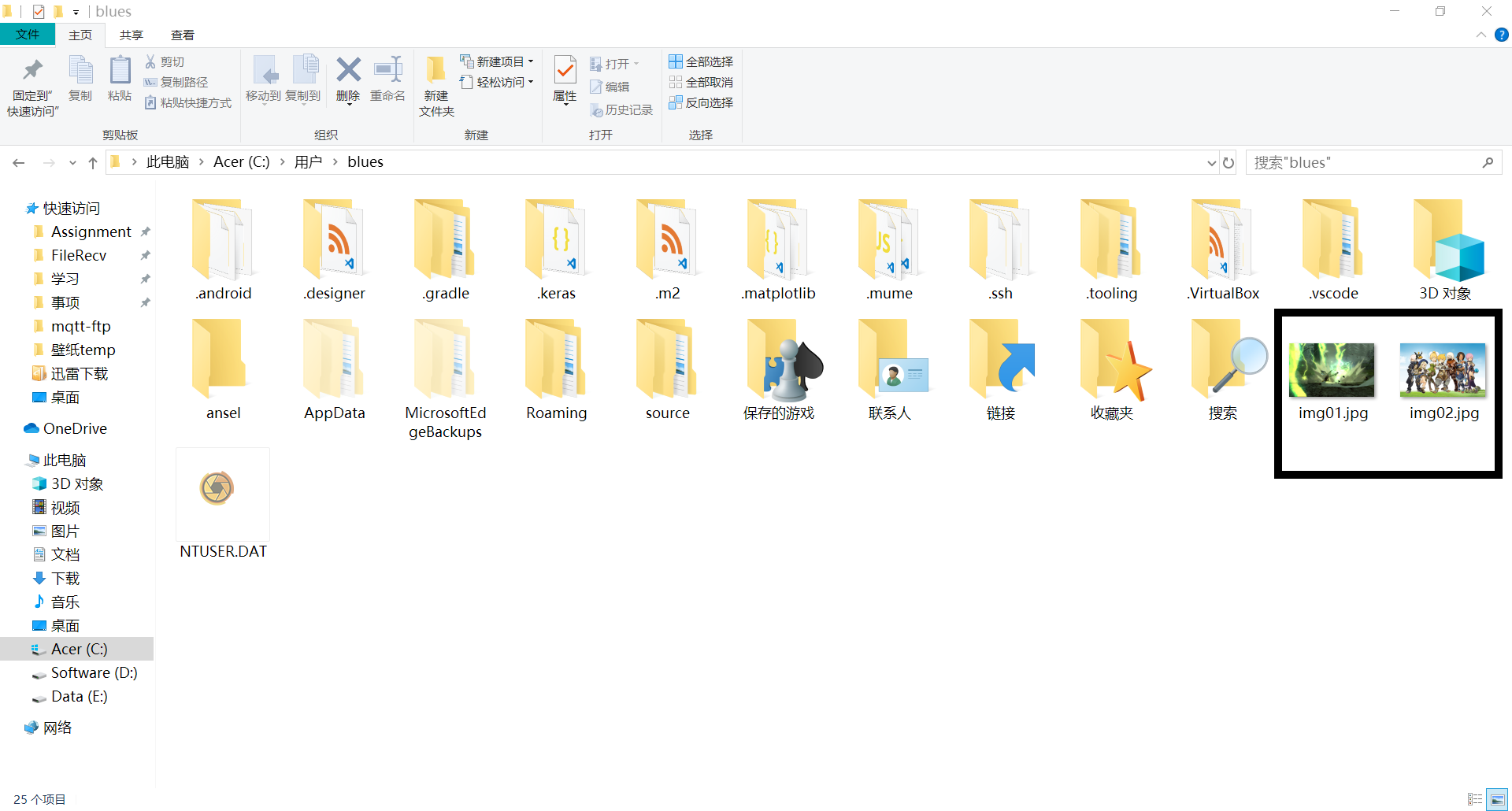
输入get命令，下载img01和img02：



服务器端显示：



客户端最后结果：



## 断开连接

