Network Project

Implementation of FTP protocol

Semester A

2018

16.01.18

Loay Naser, 204436042, Software Engineering
Ahmad Igbarea, 305051492, Software Engineering

Table of Contents

0.Proposal

FTP protocol implementation

Our goal is to demonstrate the implementation of the ftp protocol, we will implement

A part of a ″backup-server″ such as： upload method, the server log and multi-client upload….

The file that will be uploaded to the server will be divided to n packet, in our project we chose to show every packet at the console with the number of it for demonstration purposes.

The overall project will contain：

0. Proposal

1. Background and theory

2. Module design

3. Implementation

4. User guide

5. References

0. Background and Theory

1.1.Abstract

The world most magnificent invention, the INTERNET, the internet is about a worldwide interconnected computer networks that uses the TCP/IP protocol to communicate with each other.

In the early beginning of the internet, it only was for the US military to make it easier to link scientists and university professors around the world.

In the meantime the internet is about a network of networks, serve a millions of users of all the needed reasons, it may be for academic reasons, business or just for fun and other uncountable reasons.

A network of networks, today, the Internet serves as a global data communications system that links millions of private, public, academic and business networks via an international telecommunications backbone that consists of various electronic and optical networking technologies.

"Decentralized by design, no one owns the Internet and it has no central governing authority. As a creation of the Defense Department for sharing research data" [1]

1.2 FTP Protocol

Ftp stands for file transfer protocol, the ftp is a standard network communication that used for transferring files across the networks over the tcp/ip, it's based on the same concept of the server-client architecture, in order to transfer a file, you need the sender that is a client and a server that receive the file.

The implementation of the ftp protocol dates from 1971, described in RFC 141, when a file transfer system between MIT machines was developed. Since the 1971 many improvements has been done to the original ftp protocol, in this days the current ftp protocol defined by the RFC 959.

The ftp protocol become one of our important protocol that we use it in our daily tasks without even knowing that, it may be in file sharing between remote machines or independence between client and server machine system file.

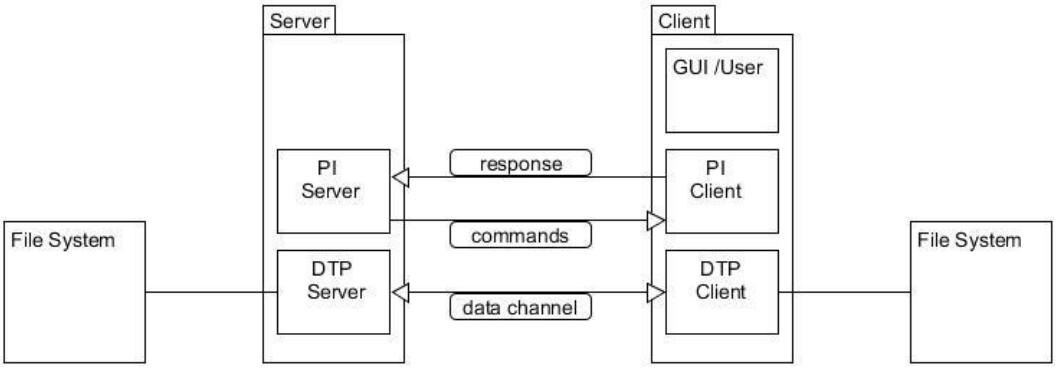


Figure 1: server-client architecture

- **DTP** (*Data Transfer Process*) is the process in charge of establishing the connection and managing the data channel
- **PI** (*Protocol Interpreter*) interprets the protocol allowing the DTP to be controlled using commands received over the control channel.
    - The SERVER-PI is responsible for listening to the commands coming from a Client-PI over the control channel on a port, establishing the connection for the control channel, receiving FTP commands from the Client-PI over this, responding to them and running the SERVER-DTP.
    - The Client-PI is responsible for establishing the connection with the FTP server, sending FTP commands, receiving responses from the SERVER-PI and controlling the Client-DTP.

1.3 FTP Store

The ftp store command it used to upload a file to the server and save it.

The command store asks the server for permission to accept the file, the fie transferred as a sequence of bytes, after receiving the accept response, the server store the file by the giving name from the client. In case that the file does not exist the server create a new file by the name, otherwise it overwrite it. [2]

1.4 FTP LogFile

In each FTP server there is an element called logFile, the log file is usually used for configuring the logging options for an ftp connection.

In the log file can be stored parameters as:

- Date
- Client IP address
- Size of response in bytes
- The mode that was used to transfer the file <asci/binary>
- Direction the file was either it was sent or received
- Service name that is usually ftp

The purpose of it is to keep tracking the ftp communication and in case if something gone wrong, to be able to track it.

1.5 FTP Connection

The FTP protocol employs a pair of connections between the client and server; in the ftp service, usually we have a server that always listing on the port 21, waiting for a client to connect and passes a command to the server. After that a second data channel are established for transferring the file, called "the data channel", it usually established by the server at port 20 to a specified client port. [3]

1.6 The Project Purpose

For our project in network class we have chosen the FTP protocol, the reason that make us take this protocol to implement and research it, is that we have been using it for a long time in our daily basics task without even think of it. Like all PC's users this protocol is so much essential for us, how many times a day you send a picture to a friend, make a backup copy of your work to cloud servers or downloading a game for fun.

The FTP protocol first arrival to the networks was in the 1971, the fact that the ftp protocol is still in use until these days says a lot about how important this protocol is, for sure the protocol have been throw dozens of improvement since that but still the same concept, you may look at the history of RFC's in the ftp web page. [4]

So, to learn the concept of the protocol and to feel it in a closer way, we decided to implement a small part of the protocol, such like the connection establishing, handling clients, accepting connections and files transferring, saving the sent file and the server log. For that we build something close to the backup servers, our server only designed to receive files from clients and make a copy of them, its practical and useable in a way that can be used for personal purposes such like sending pictures or documents for my pc to my laptop in a way that make us the programmers feel like we have done something useful for us and for our learning acknowledge.

1.7 Module Background:

In the module background we are going to explain some of our project modules and classes that we implement.

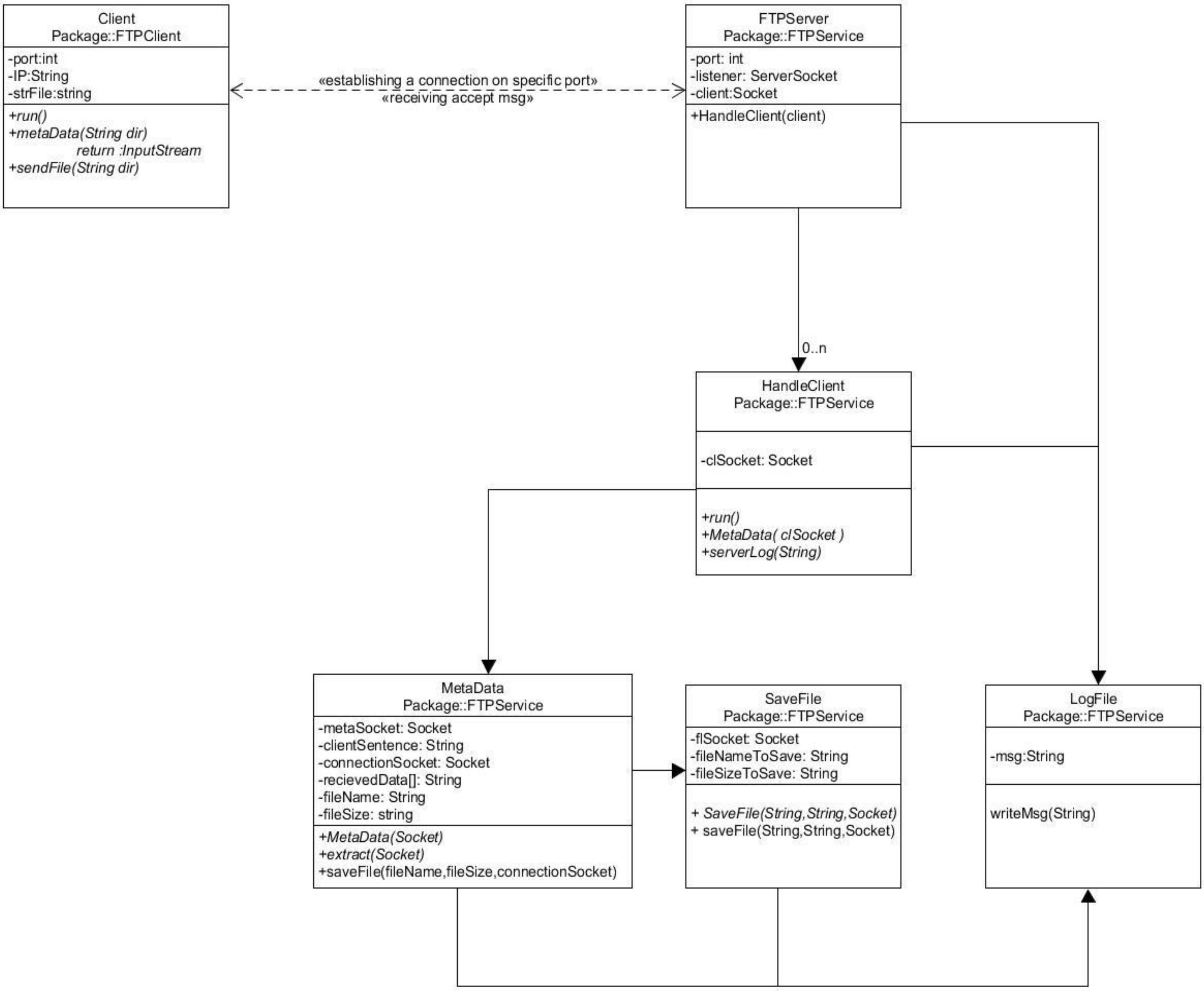


Figure 2: System Class diagram

1.7.1 FTPServer class

The main purpose of the FTPServer class is only to listen to the new clients on the localhost, waiting for a client to establish a connection to the server, via a specified port that built in the code and a server socket as a listener.

When a client connect to the server, the server passes the client socket to the HandleClient class.

1.7.2 MetaData class

The MetaData class is responsible for extracting the name, type and the size of the file that the client want to send. As in the real ftp protocol a two channel is required to transfer a file the first one for commands and the second for the data, we make it similar in our project.

In our project, we divided the connection to two type messages the first for command "BufferReader" message and the second is for data "DataInputStream".

The MetaData receive a socket from the HandleClient class and extract from it the Meta data needed to save the file, after that it send them to the saveFile class

1.7.3 SaveFile class

The constructor of saveFile class demand a string that represent the name of the file, also string that contain the file size and the client socket to receive the data from it.

The saving process is simple, it divides the packets into bytes of 4096, and run a loop until all the bytes received and written in the server directory. The loop depends on the file size that received from the Meta data class and in every iteration; it decrease the amount of bits written from the original file size.

1.7.4 ServerLog class

The ServerLog class designed to be as the actual ftp log file, it contains all the data and metadata of the connection, also the file name, size and type, the time of every function accord from the beginning of the ftp server session until it shutdown.

So, if any error happen or if the manager want to keep tracking the sessions, the ServerLog is the wright reference for him.

1. Module design

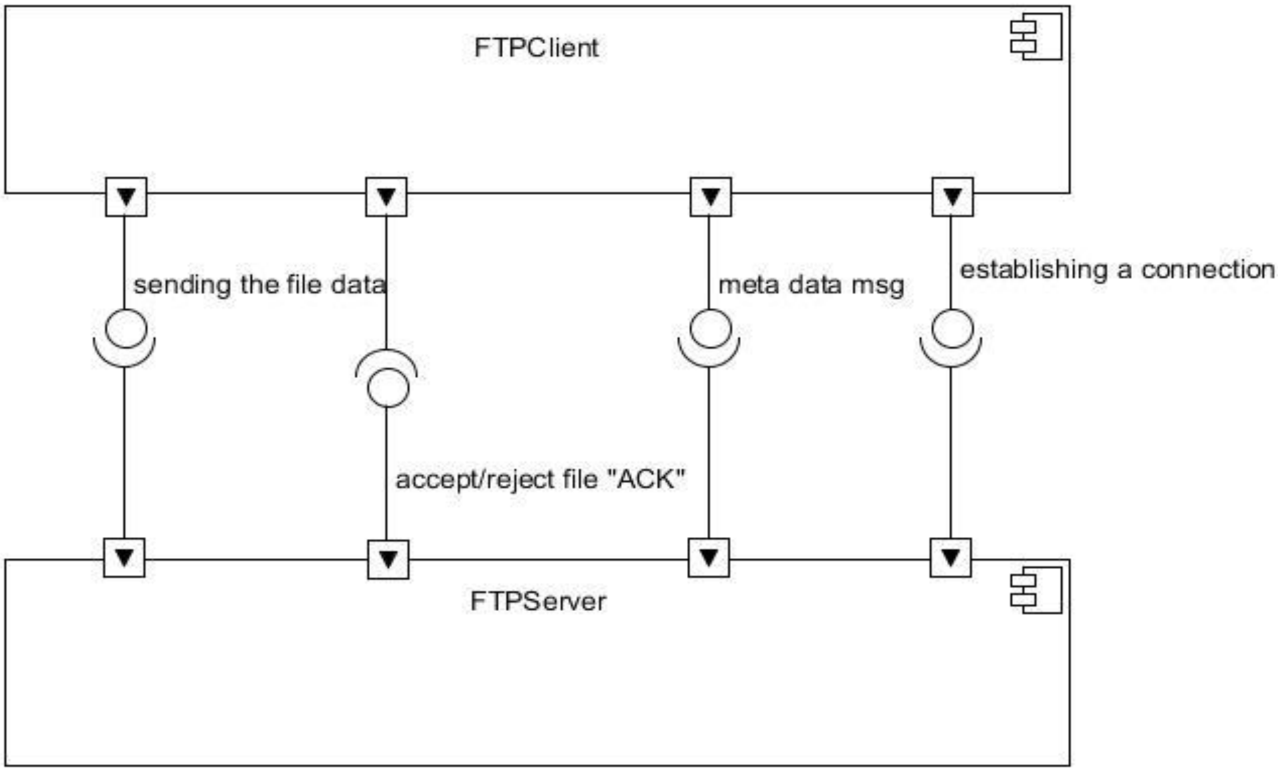


Figure 3: system component diagram

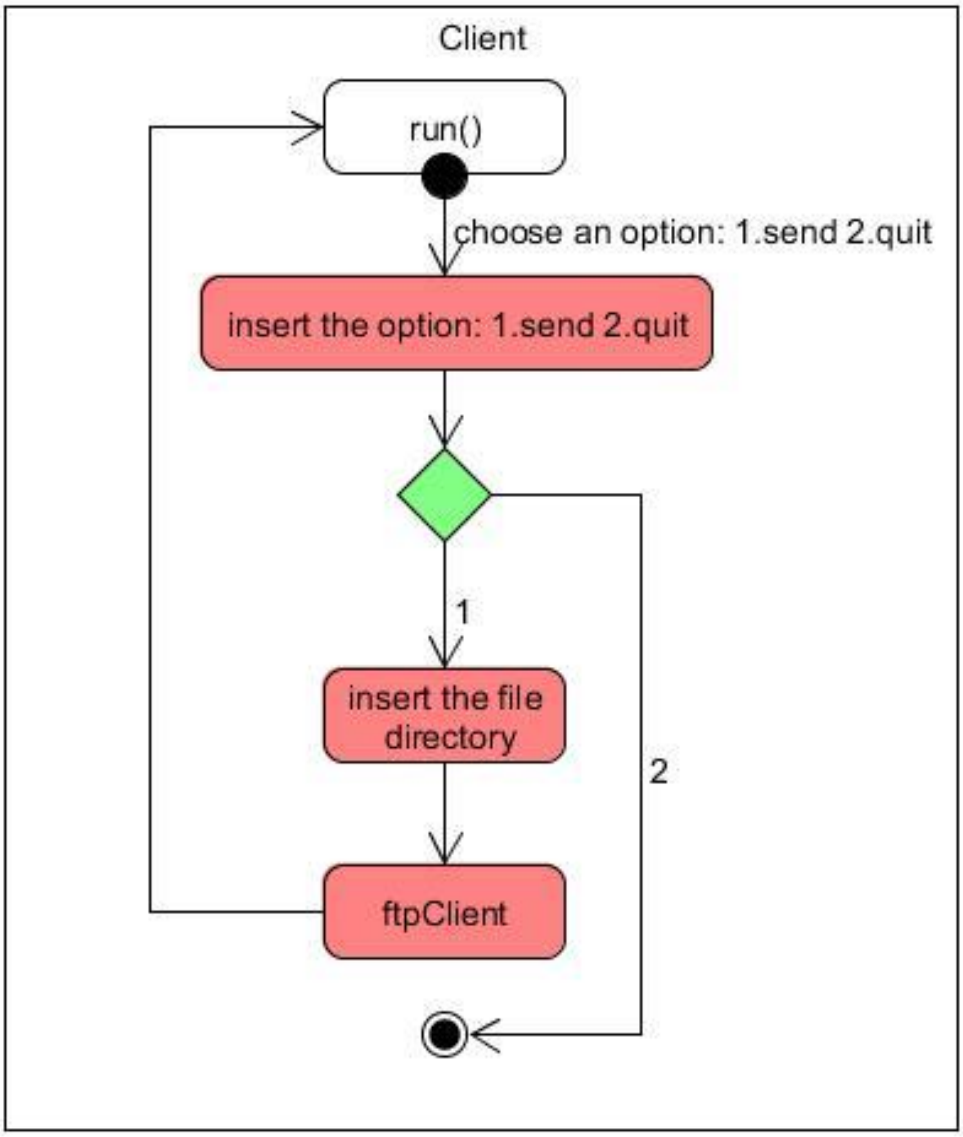

Client
run()
choose an option: 1.send 2.quit
insert the option: 1.send 2.quit
1
2
insert the file directory
ftpClient

Figure 4: FTPClient Activity diagram

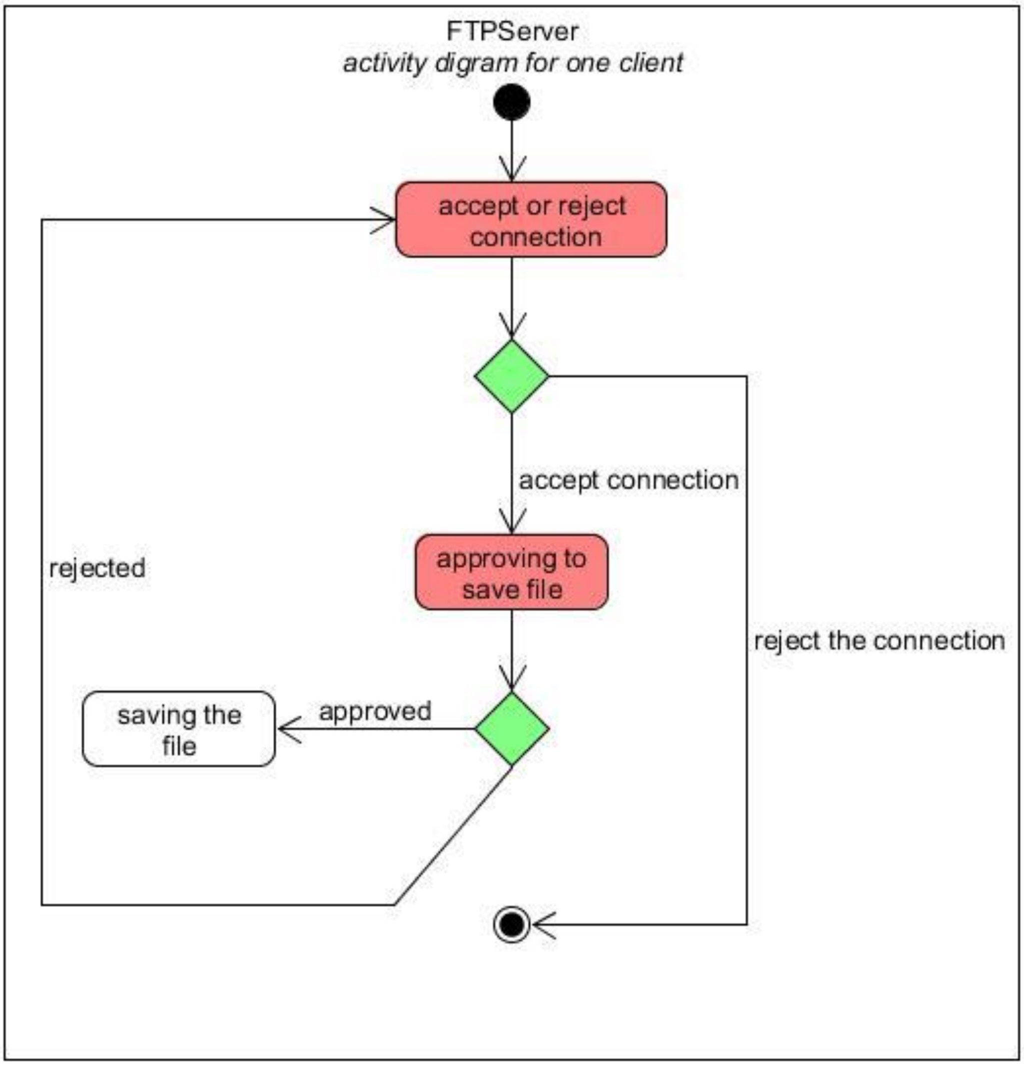


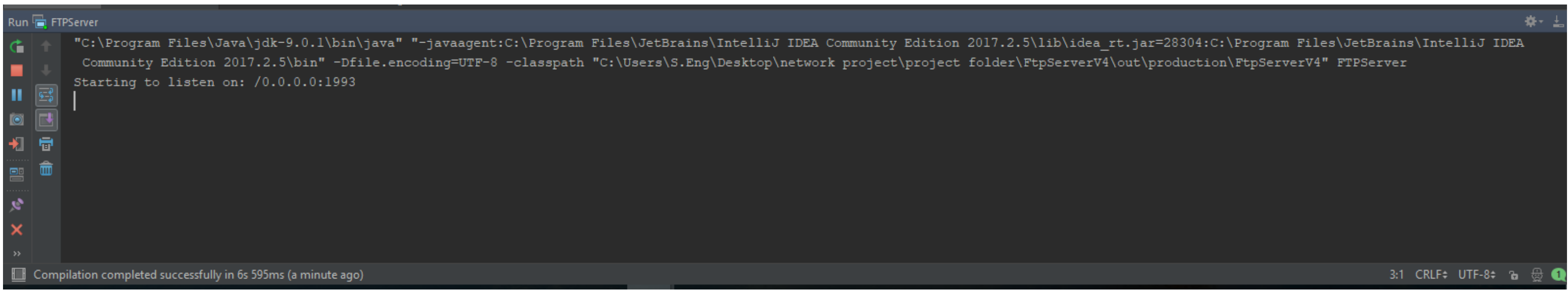
Figure 5: FTPServer Activity diagram

2. Implementation

https://github.com/loayN/FTP-protocol-Implementation-

3. User guide

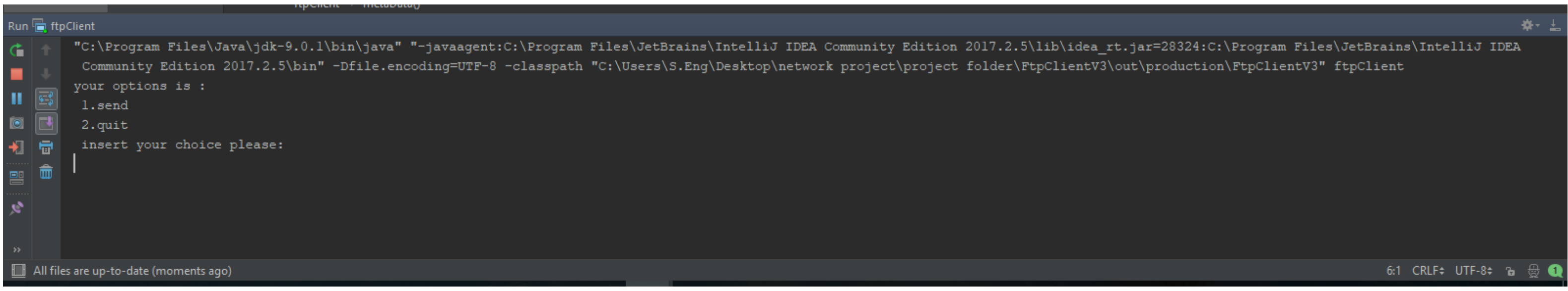
First: run the server program, it will start listening on the local host at port 1993.



```
"C:\Program Files\Java\jdk-9.0.1\bin\java" "-javaagent:C:\Program Files\JetBrains\IntelliJ IDEA Community Edition 2017.2.5\lib\idea_rt.jar=28304:C:\Program Files\JetBrains\IntelliJ IDEA
 Community Edition 2017.2.5\bin" -Dfile.encoding=UTF-8 -classpath "C:\Users\S.Eng\Desktop\network project\project folder\FtpServerV4\out\production\FtpServerV4" FTPServer
Starting to listen on: /0.0.0.0:1993
```

Compilation completed successfully in 6s 595ms (a minute ago)

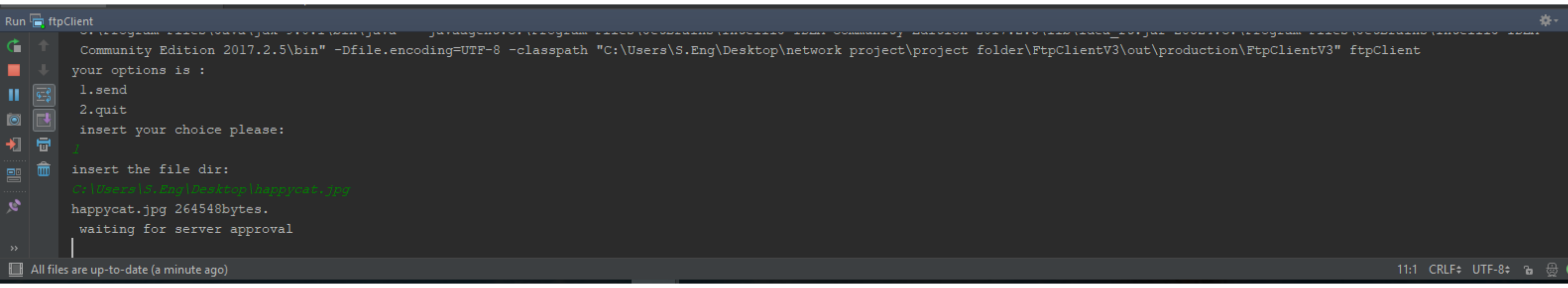
On the client side you have to choose one of the options to send or to quit.



```
"C:\Program Files\Java\jdk-9.0.1\bin\java" "-javaagent:C:\Program Files\JetBrains\IntelliJ IDEA Community Edition 2017.2.5\lib\idea_rt.jar=28324:C:\Program Files\JetBrains\IntelliJ IDEA
 Community Edition 2017.2.5\bin" -Dfile.encoding=UTF-8 -classpath "C:\Users\S.Eng\Desktop\network project\project folder\FtpClientV3\out\production\FtpClientV3" ftpClient
your options is :
 1.send
 2.quit
 insert your choice please:
```

All files are up-to-date (moments ago)

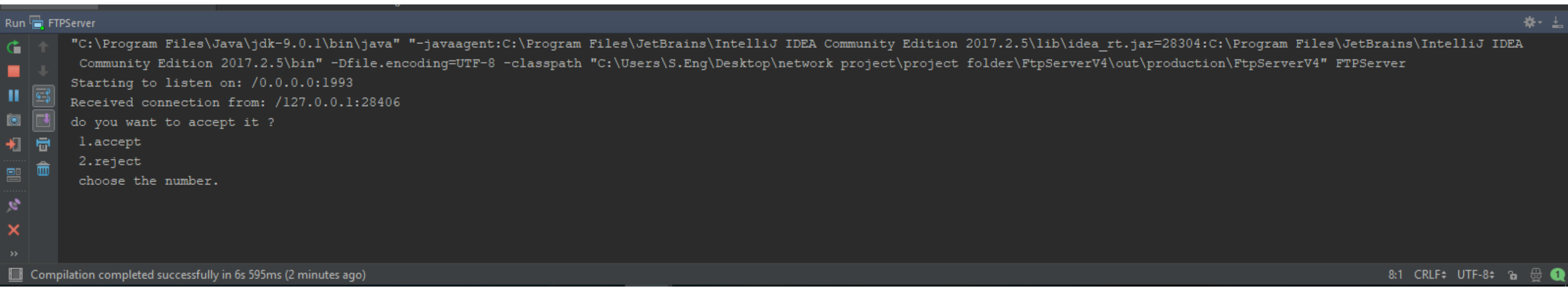
In case if you choose to send, you will be asked to enter the file directory, and wait to server approval.



```
 Community Edition 2017.2.5\bin" -Dfile.encoding=UTF-8 -classpath "C:\Users\S.Eng\Desktop\network project\project folder\FtpClientV3\out\production\FtpClientV3" ftpClient
your options is :
 1.send
 2.quit
 insert your choice please:
1
insert the file dir:
C:\Users\S.Eng\Desktop\happycat.jpg
happycat.jpg 264548bytes.
 waiting for server approval
```

All files are up-to-date (a minute ago)

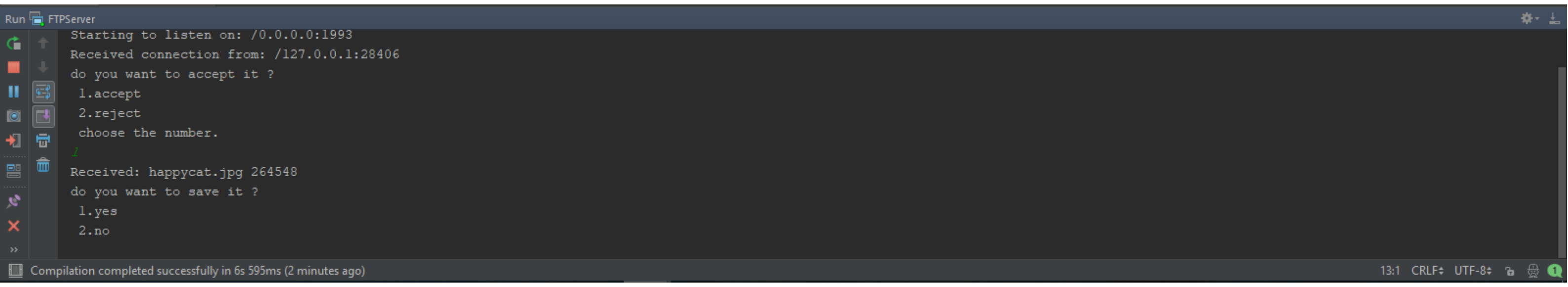
In the server side you will see that a client want to connect, you may accept it or not.



```
"C:\Program Files\Java\jdk-9.0.1\bin\java" "-javaagent:C:\Program Files\JetBrains\IntelliJ IDEA Community Edition 2017.2.5\lib\idea_rt.jar=28304:C:\Program Files\JetBrains\IntelliJ IDEA
 Community Edition 2017.2.5\bin" -Dfile.encoding=UTF-8 -classpath "C:\Users\S.Eng\Desktop\network project\project folder\FtpServerV4\out\production\FtpServerV4" FTPServer
Starting to listen on: /0.0.0.0:1993
Received connection from: /127.0.0.1:28406
do you want to accept it ?
 1.accept
 2.reject
 choose the number.
```

Compilation completed successfully in 6s 595ms (2 minutes ago)

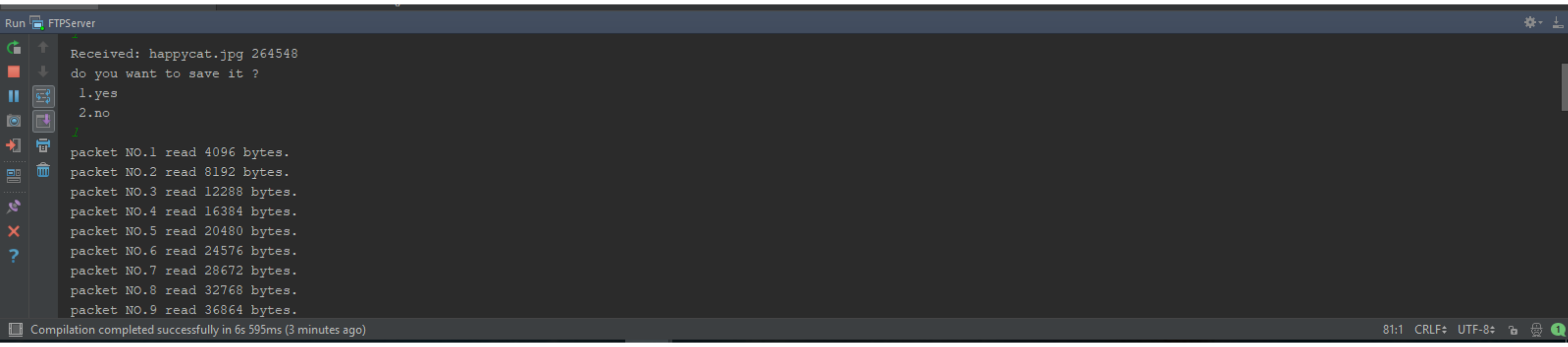
In case you accept it, you will receive the file name and size and asks you if you want to save it.



```
Starting to listen on: /0.0.0.0:1993
Received connection from: /127.0.0.1:28406
do you want to accept it ?
 1.accept
 2.reject
 choose the number.
1
Received: happycat.jpg 264548
do you want to save it ?
 1.yes
 2.no
```

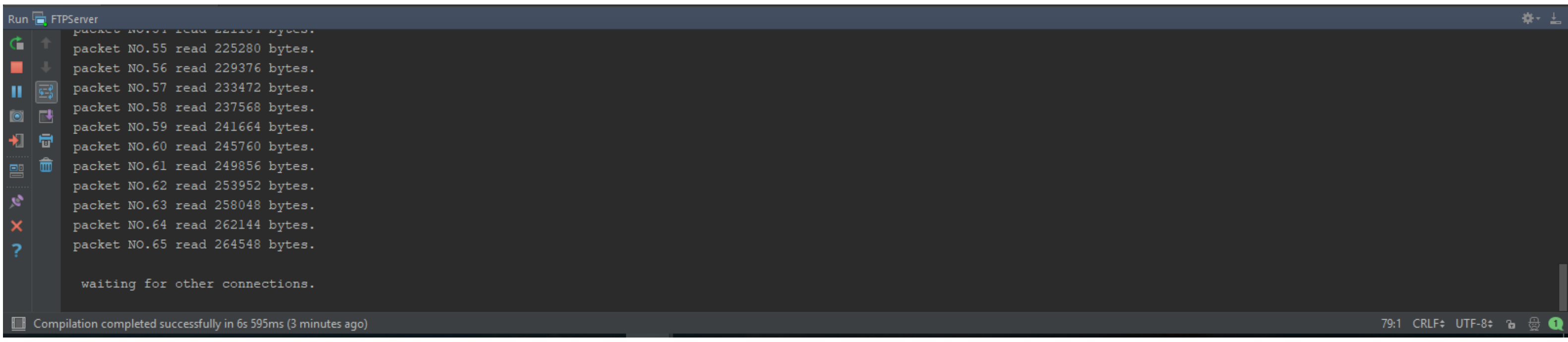
Compilation completed successfully in 6s 595ms (2 minutes ago)

If you accept to save it it will show you the packet that been sent and aessage that the server will continue to listen to other connections.



```
Received: happycat.jpg 264548
do you want to save it ?
 1.yes
 2.no
1
packet NO.1 read 4096 bytes.
packet NO.2 read 8192 bytes.
packet NO.3 read 12288 bytes.
packet NO.4 read 16384 bytes.
packet NO.5 read 20480 bytes.
packet NO.6 read 24576 bytes.
packet NO.7 read 28672 bytes.
packet NO.8 read 32768 bytes.
packet NO.9 read 36864 bytes.
```

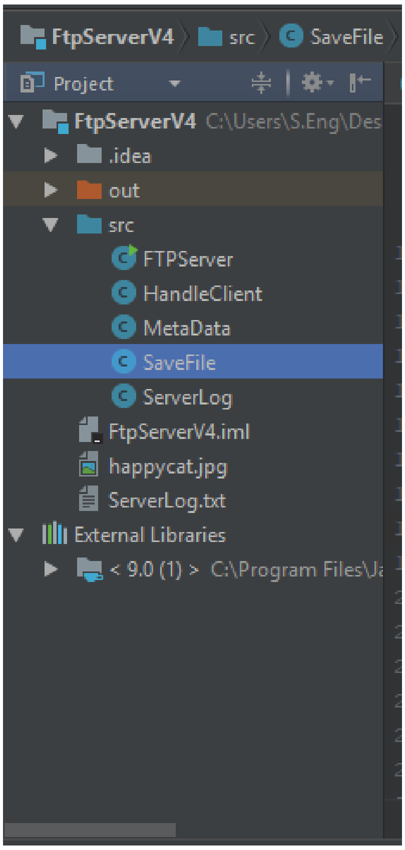
Compilation completed successfully in 6s 595ms (3 minutes ago)



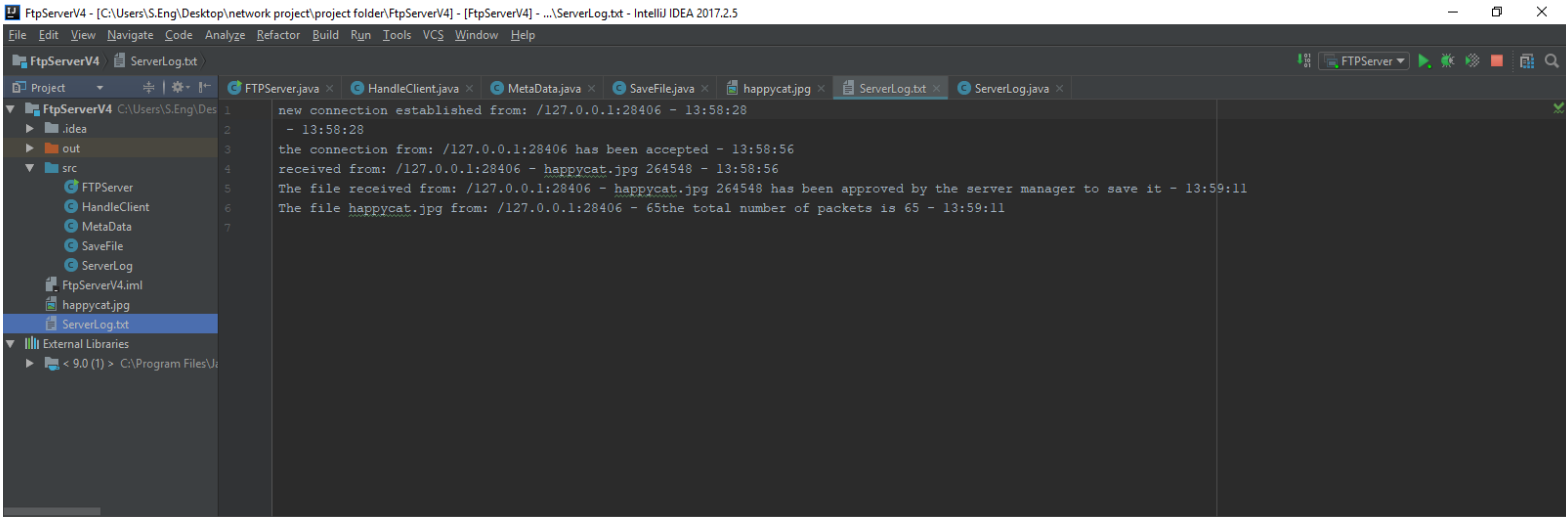
```
packet NO.54 read 221184 bytes.
packet NO.55 read 225280 bytes.
packet NO.56 read 229376 bytes.
packet NO.57 read 233472 bytes.
packet NO.58 read 237568 bytes.
packet NO.59 read 241664 bytes.
packet NO.60 read 245760 bytes.
packet NO.61 read 249856 bytes.
packet NO.62 read 253952 bytes.
packet NO.63 read 258048 bytes.
packet NO.64 read 262144 bytes.
packet NO.65 read 264548 bytes.

 waiting for other connections.
```

Compilation completed successfully in 6s 595ms (3 minutes ago)

The files that the server received will be saved at the same directory of the project.

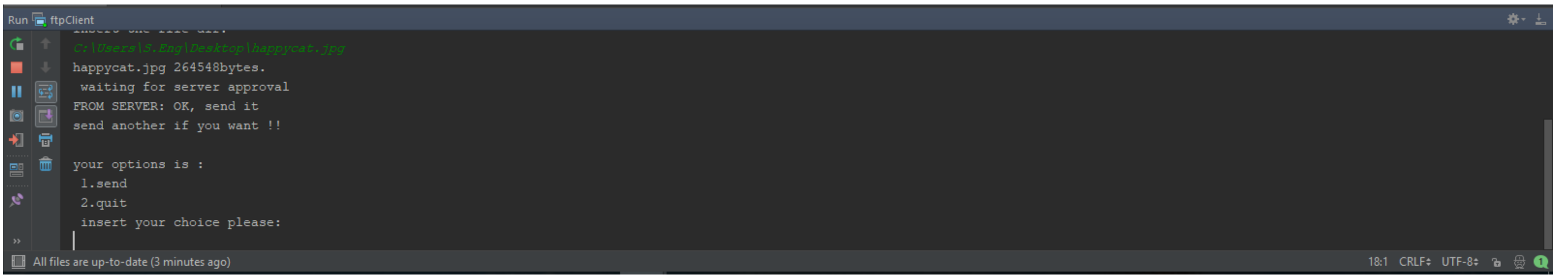


The server log also saved at the same project directory as shown below


FtpServerV4 - [C:\Users\S.Eng\Desktop\network project\project folder\FtpServerV4] - [FtpServerV4] - ...\ServerLog.txt - IntelliJ IDEA 2017.2.5
File Edit View Navigate Code Analyze Refactor Build Run Tools VCS Window Help
FtpServerV4 〉 ServerLog.txt
FTPServer
Project
FTPServer.java
HandleClient.java
MetaData.java
SaveFile.java
happycat.jpg
ServerLog.txt
ServerLog.java
FtpServerV4 C:\Users\S.Eng\Des
.idea
out
src
FTPServer
HandleClient
MetaData
SaveFile
ServerLog
FtpServerV4.iml
happycat.jpg
ServerLog.txt
External Libraries
< 9.0 (1) > C:\Program Files\J

```
new connection established from: /127.0.0.1:28406 - 13:58:28
 - 13:58:28
the connection from: /127.0.0.1:28406 has been accepted - 13:58:56
received from: /127.0.0.1:28406 - happycat.jpg 264548 - 13:58:56
The file received from: /127.0.0.1:28406 - happycat.jpg 264548 has been approved by the server manager to save it - 13:59:11
The file happycat.jpg from: /127.0.0.1:28406 - 65the total number of packets is 65 - 13:59:11
```

After the approval of the server to save the file, the client receive acception massage ″OK, send it″, and then let you choose to send another or quit.



```
C:\Users\S.Eng\Desktop\happycat.jpg
happycat.jpg 264548bytes.
 waiting for server approval
FROM SERVER: OK, send it
send another if you want !!

your options is :
 1.send
 2.quit
 insert your choice please:
```

4. Reference

[1] A Brief Guide to the History of the Internet, Investintech.com Inc. - https://www.investintech.com/content/historyinternet/

[2] FTP protocol (File Transfer Protocol), January 2018 -

http://ccm.net/contents/272-ftp-protocol-file-transfer-protocol

[3] Steve CORPORATION, Steve Gibson, - https://www.grc.com/port_20.htm

[4] File Transfer Protocol, J. Postel, J. Reynolds, October 1985 - https://tools.ietf.org/html/rfc959