

# Assignment 1 - FTP프로토콜 기본 기능 구현

---

## 이름

2016032897 산업공학과 김기범

## 기본 기능 종류

- **CD**(Change Directory)  
서버의 현재 directory 위치 변경
- **LIST**  
서버의 현재 directory의 목록 보여주기
- **GET**  
서버의 파일을 Client가 실행시킨 프로그램 위치에 불러오기(저장)
- **PUT**  
Client의 파일을 서버의 **현재 Directory**위치에 저장시키기

## 프로그램 구조 및 설계

### Status Code(Server To Client)

401 - 서버에 해당하는 파일이 존재하지 않을 때 보내는 code.

E.X) 401 Failed No such file exists

200 - 서버의 현재 directory 위치를 성공적으로 바꿨다는 code.

E.X) 200 Moved to C:\Users\rlqja\eclipse-workspace\Network\src

201 - List Method의 요청으로 오는 Response Message의 코드다. 서버에서 directory에는 entry의 총 갯수를 알려줄 때의 code다. 그리고 서버에서 오는 File list 목록 Client에게 보여준다.

E.X) 201 Comprising 5 entries

501 - Directory 이름이 존재하지 않을 때 보내는 Code.

E.X) 501 Failed Directory name is invalid 203 - **Put** method로 Request 했을 때 서버에서 파일을 받을 준비가 되었다는 Code. 그리고 Client에게 Server로 새로운 Data Channel을 위한 TCP Connection을 지시한다.

E.X) 203 Ready to receive 300 - **Get** method로 Request 했을 때 서버에서 파일의 총 **byte** 크기를 알려주고 새로운 Data Channel을 열라는 Code.

E.X) 300 Containing 7000 bytes in total

502 - Server에서 알 수 없는 이유로 파일을 받을 수 없다는 Code

E.X) 502 Failed for Unknown reason

### CD

**CD**으로 서버의 현재 Directory 위치를 변경시킬 때는 절대경로, 상대경로를 모두 지원한다. "CD" Path를 서버에게 보낼 때 "(빈 공간)"과 "."은 서버에서 현재 위치를, ".."은 현 Directory의 Parent Directory의 절대경로를, "\home\src\ftp"등은 마지막 디렉토리의 **절대 경로**를 Client에게 보내도록 했다. 서버에서 Client가 요청한 Directory가 **존재하면** 현재 서버의 Directory 위치를 변경하고 **200** Code를 보내고, 존재하지 않으면 **501** Code를 보낸다.

### LIST

**LIST**는 요청한 Path에 있는 파일 목록을 보여달라는 Method다. **CD**와 마찬가지로 „,„상대경로, 절대경로 모두를 지원한다. 서버에게 LIST (Path)를 요청하여 Path가 존재하면 201코드를 먼저 보낸다. 201코드를 통해 Client에게 해당 Directory가 존재하고 곧 list를 보내준다는 것을 알려준다. 그 후, 실제로 file list를 Client에게 보낸다. 만약 directory가 존재하지 않으면 501 Code를 Client에게 보내서 존재하지 않다는 것을 알려준다.

## GET

**GET**은 요청한 파일이 있으면 현재 Client directory 위치에 파일을 가져오는 함수다. 먼저 Client가 GET (파일명)을 보내면 서버에서는 파일이 존재하는지 확인한다. 존재하면 **300**코드를 Client에게 보내서 파일이 존재한다는 것을 알리고 **TCP Connection**을 요청한다. 이 사이에 서버는 Data Channel를 위해 새로운 포트 번호로 **ServerSocket**을 연다. Client는 새로운 Socket을 만들고 Data Port 번호로 서버에게 TCP Connection을 요청한다. Server에서는 accept()를 통해 TCP Connection이 정상적으로 이루어지면 요청한 파일을 Client에게 전송한다. 이 때, **DataOutputStream**을 통해 file를 **byte**단위로 전송한다. Client은 서버로부터 받은 byte 배열을 이용하여 **FileOutputStream**을 통해 file로 저장한다. 파일이 존재하지 않을 경우 **401**번의 status code를 보내 존재하지 않다는 것을 알리고 종료시킨다.

Server에서 Client에게 파일 데이터를 보낼 때는 "{SeqNo(1byte), CHKsum(2bytes), Size(1byte), 데이터청크(1000bytes)}"의 포맷을 위해 크기가 1005인 **byte 배열**을 형성시킨다. Index로 0에는 Seq번호, 1~2에는 **Checksum**, 3~4에는 Chunk size, 5~에는 byte로 이루어진 데이터 chunk가 담겨있다.

Server에서 Client에게 데이터를 보낼 때마다 Client는 **ack**메세지를 Server에게 보낸다. "{SeqNo(1byte), CHKsum(2bytes)}"의 형식을 맞추기 위해 크기가 3인 byte 배열을 형성했다. 0에는 Seq번호, 1~2에는 checksum번호가 들어가있다. Assignment1에서는 **packet loss** 등의 문제가 발생하지 않으므로 data chunk를 하나 보내고 **ack message**를 받기를 기다린다. ack message를 받은 후(이 때 "#"출력), 다음 **data chunk**를 보내도록 한다.

## PUT

**PUT**은 서버의 현재 Directory 위치에 해당 파일을 보낸다. Client에서는 PUT (파일명)에서 파일이 실제로 존재하는지 확인한 후, 서버에게 요청한다. 파일이 존재하지 않으면 **TCP Connection**을 하지 않고 종료시킨다. 서버에서는 잘 받았는 뜻으로 **203** code를 Client에게 보내고 Data Channel를 위해 새로운 TCP Connection을 요청한다. 이 사이에 서버는 새로운 포트 번호로 **ServerSocket**을 만들어놓고 기다린다. Client가 203 code를 받으면 서버에게 새로운 TCP Connection을 만든다. 보낼 파일을 byte로 변환 후, 서버에게 보낸다. 서버는 파일을 byte 배열로 받는데, 이를 **FileOutputStream**을 이용하여 파일로 저장시킨다.

## Server

서버를 열 때 매개변수를 2개, 1개, 0개를 받는다. 2개를 받을 때는 순서대로 Commend Port 번호, Data Channel를 위한 Port 번호를 지정한다. 1개를 받을 때는 Commend Port 번호만 지정한다. 아무것도 받지 않으면 기본 포트 번호로 Commend는 2020, Data Channel의 포트 번호는 2121로 지정한다. 서버를 열 때 Main 파일명은 **ServerMain**이다. 자세한 구현 방법은 이후에 설명할 것이다.

## Client

Client가 서버에게 Request할 때 프로그램을 실행시키는데, 매개변수로 3개 또는 0개를 받는다. 3개를 받을 때는 순서대로 **Sever ip**, Commend Channel를 위한 포트 번호, Data Channel를 위한 포트 번호를 지정한다. 아무것도 매개변수로 지정하지 않으면 Default 값으로 ip, commend port 번호, Data Channel port 번호 순서대로 127.0.0.1, 2020, 2121로 지정된다. 사용자가 프로그램을 종료시키고 싶으면 **Quit**를 입력하면 종료된다. Client 프로그램을 실행시킬 때 Main 파일명은 **ClientMain**이다.

## 동작 절차

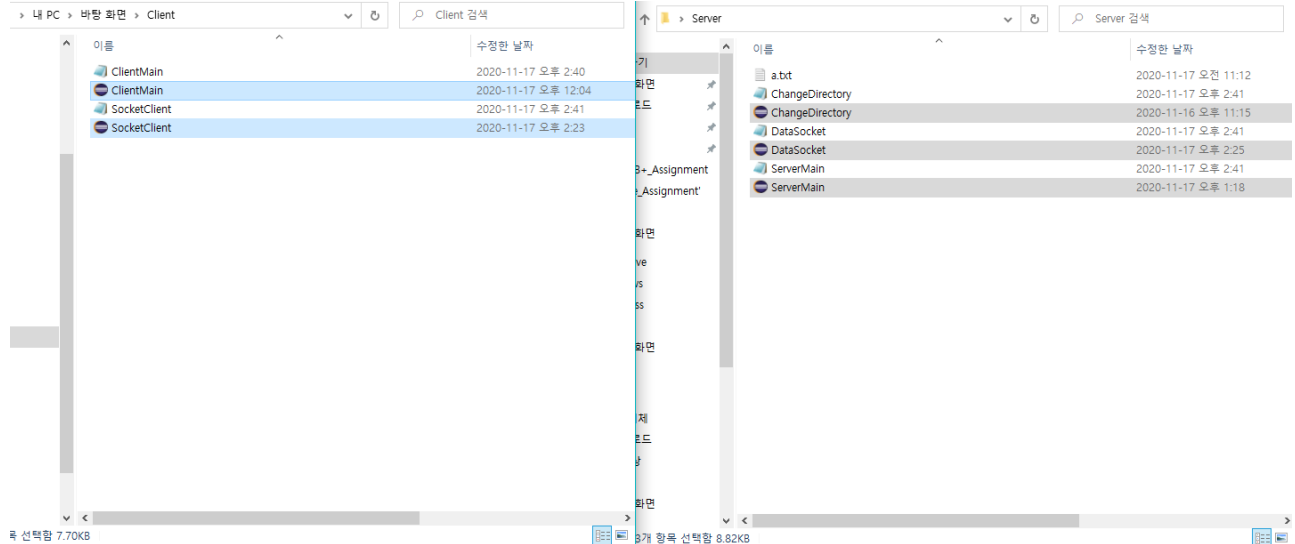
## • 실행 전

Server와 관련된 파일들(ServerMain.java, DataSocket.java, ChangeDirectory.java)을 같은 폴더에 놓는다. Client와 관련된 파일들(ClientMain.java, SocketClient.java)를 같은 폴더에 놓는다. 각 폴더에서 모든 java 파일을 compile시켜 **class** 파일을 형성시켜 놓는다. 실행은 Windows10에서 CMD에서 실행시켰다. 컴파일 시 CMD에서 java파일이 있는 곳으로 Directory를 이동시킨 후, **javac 파일명.java**으로 컴파일했다.

컴파일 예시

```
C:\Users\wrlqja\Desktop>javac ClientMain.java
```

컴파일 다 된 후의 모습(a.txt는 get을 위해 만들어놓은 파일)



## • 실행

1. ServerMain 클래스 파일을 실행시킨다.(컴파일이 다 된 후) ServerMain 클래스 파일이 있는 Directory로 이동한 후, **java ServerMain** or **java ServerMain 20 21**(포트 번호 지정) 실행

```
C:\Users\wrlqja\Desktop\Server>java ServerMain
Server Open
```

Or

```
C:\Users\wrlqja\Desktop\Server>java ServerMain 20 21
Server Open
```

2. ClientMain 클래스 파일을 실행시킨다. ClientMain 클래스 파일이 있는 Directory로 이동한 후, **java ClientMain** or **java ClientMain 127.0.0.1 20 21**(서버 ip, 포트번호 지정) 실행한다.

```
C:\Users\wrlqja\Desktop\Client>java ClientMain
=====
Enter the Quit if you want to quit this program.
=====
```

Or

```
C:\Users\wrlqja\Desktop\Client>java ClientMain 127.0.0.1 20 21
=====
Enter the Quit if you want to quit this program.
=====
```

3. 실행시킨 ClientMain에서 Method를 입력하여 서버에게 요청한다. 이 때 **PUT**할 때는 ClientMain이 있는 폴더 내에 보낼 파일이 존재해야 한다. **GET**할 때는 ClientMain 폴더 내에 파일이 저장된다. cd, get, put, list는 대소문자 구별하지 않아도 된다.

여기서 파일 이름이 한글말일 때는 **encoding**이 안되어 나타난다. 아래 결과에서 각 Method를 입력 한 후에는 "Enter the Quit if you want to quit this program"의 문구가 나타난다.

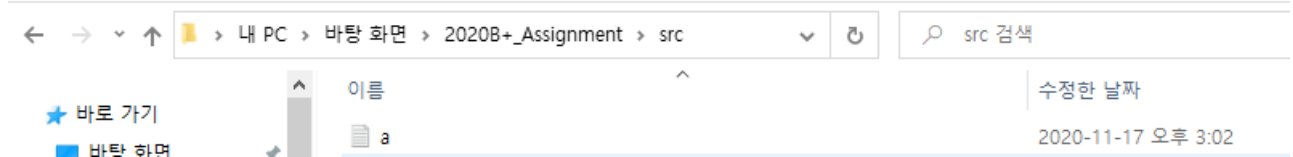
```

C:\Users\rlqja\Desktop\Client>java ClientMain 127.0.0.1 20 21
Enter the Quit if you want to quit this program.
cd
C:\Users\rlqja\Desktop\Server>
Enter the Quit if you want to quit this program.
list
ChangeDirectory.class,1048
ChangeDirectory.java,1405
DataSocket.class,3135
DataSocket.java,3552
ServerMain.class,4685
ServerMain.java,4076
Enter the Quit if you want to quit this program.
cd ..
C:\Users\rlqja\Desktop>
Enter the Quit if you want to quit this program.
cd #2020B+_Assignment\src
C:\Users\rlqja\Desktop\2020B+_Assignment\src>
Enter the Quit if you want to quit this program.
get bptree.java
Received bptree.java / 7918bytes
##### Completed...
Enter the Quit if you want to quit this program.
put a.txt
a.txt transferred / 8 bytes
##### Completed...
Enter the Quit if you want to quit this program.
quit
Client is Closed.

C:\Users\rlqja\Desktop\Server>java ServerMain 20 21
Server Open
Request: cd .
Response :200 Moved to C:\Users\rlqja\Desktop\Server
Request: list
Response :201 Comprising 6 entries
Request: cd ..
Response :200 Moved to C:\Users\rlqja\Desktop
Request: cd #2020B+_Assignment\src
Response :200 Moved to C:\Users\rlqja\Desktop\2020B+_Assignment\src
Request: get bptree.java
Response: 300 Containing 7918bytes in total
Request: put a.txt
Request :8
Response :203 Ready to receive
C:\Users\rlqja\Desktop\Server>

```

- PUT으로 서버 directory에 저장



- GET으로 Client 폴더에 저장

