**Computer Network - Programming Assignment 2**

컴퓨터소프트웨어학부 2017029925 양문영

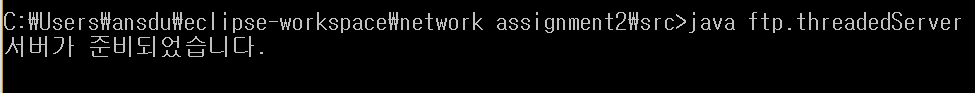
Programming Assignment 1에서 구현한 FTP 프로그램은 한 client가 서버와 연결되어 수행 중인 경우 다른 client는 연결요청에서 block되어 대기하다가 먼저 연결된 client가 수행을 종료하면 바로 ftp session이 시작될 수 있도록 한다. 하지만 이런 식으로 하면 여러 명의 client가 연결될 때 먼저 연결된 client가 연결을 끊을 때까지 대기해야 하는 문제가 생긴다. 이 문제를 해결하기 위해 Programming Assignment 2에서 서버는 각각의 client별로 전담 thread를 배정하여 복수의 client를 동시에 service를 할 수 있도록 한다.

1. compile

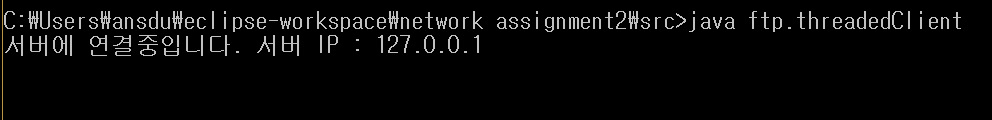


1. 실행 방법

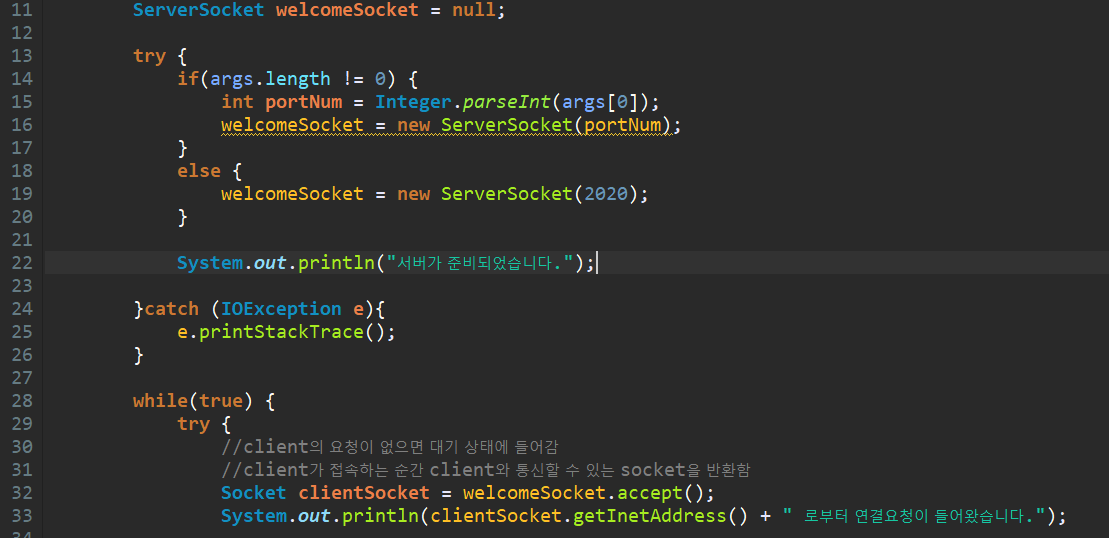
먼저, server를 실행시킨다.



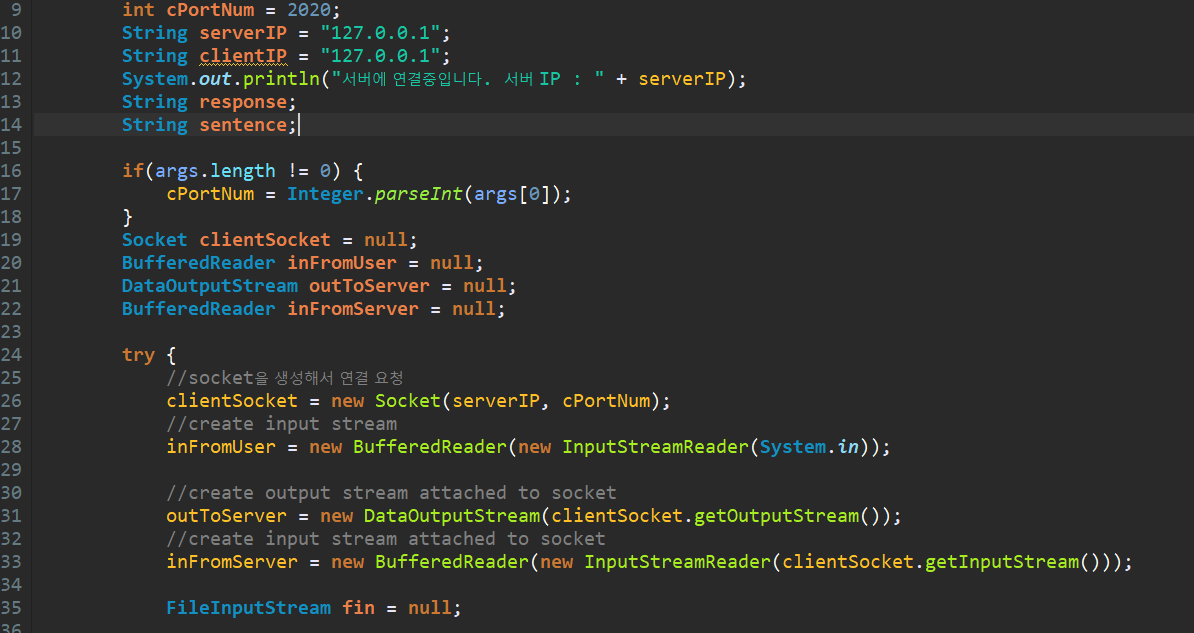
그리고 client를 실행시켜준다.



1. client-server 동작 절차



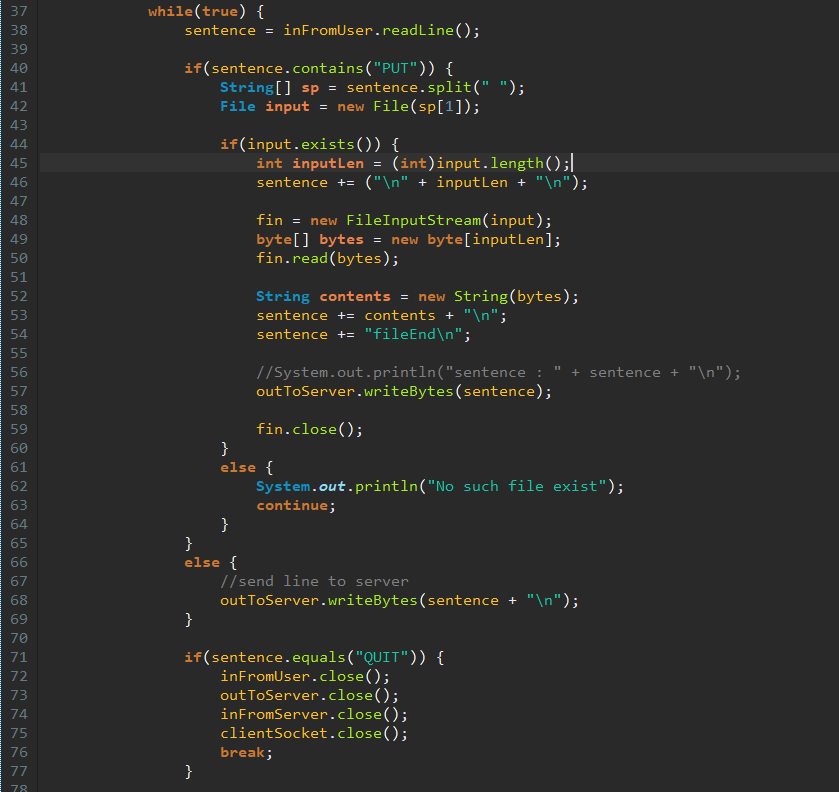
server를 실행시키면 port number를 가지고 welcomSocket을 만들어서 서버를 열고 client가 connection되기를 accept()에서 기다린다.



client를 실행시키면 clientSocket을 만들어서 server와 연결요청을 하고 user가 입력하는 command와 서버에게 보내기 위한 output stream, 서버가 보낸 메시지를 읽기 위한 input stream을 준비한다. 그리고 파일을 읽기 위한 FileInputStream도 준비한다.



server에서 client가 연결되면 35번째 줄을 통해 client별로 thread를 만들어줘서 ClientHandler class를 통해 connection을 한다.



client에서 38번째 줄을 통해 user가 보낸 command를 읽는다. command로 ‘QUIT’을 입력하면 server와의 connection이 종료된다.



ClientHandler class를 실행시키면 (threadedServer.java의 37번째 줄에서의 t.start()를 통해) 79번째 줄을 통해 client가 보낸 메시지를 읽고 각 command에 맞는 처리는 processResponse() method를 통해 처리해주고 그 method를 통해 나온 message를 93번째 줄을 통해 client에게 보내준다.



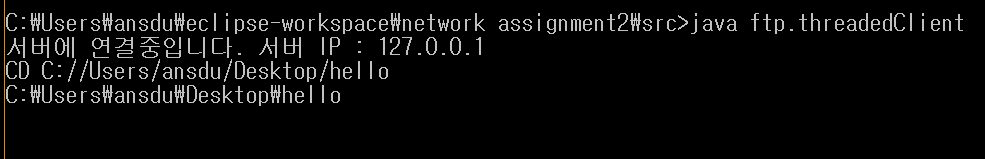
threadedClient.java에서 80번째 줄을 통해 server에게서 온 메시지를 처리해준다. switch문을 통해 state code별로 처리를 해준다. 메시지에 맞는 print를 user에게 해준 후 while loop를 통해 다른 command를 다시 받는다.

1. command 처리
   1. CD
      1. CD (절대경로)

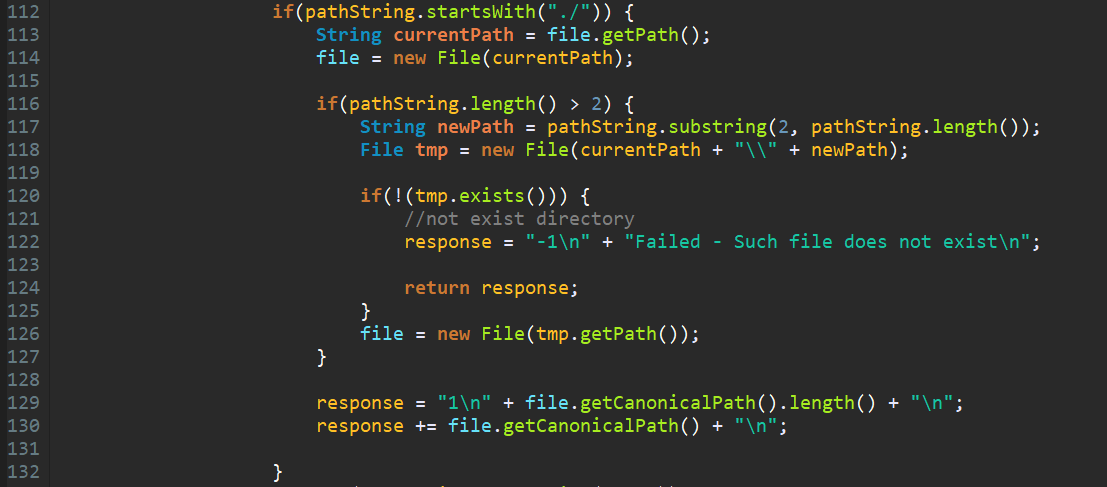


server의 현재 디렉토리 경로를 입력한 경로로 바꾸고 state code 1과 보낼 메시지의 길이, 현재 server의 현재 디렉토리를 메세지로 보낸다. 입력한 경로에 해당하는 디렉토리가 없다면 state code -1과 오류 메시지를 보낸다.

* 실행화면

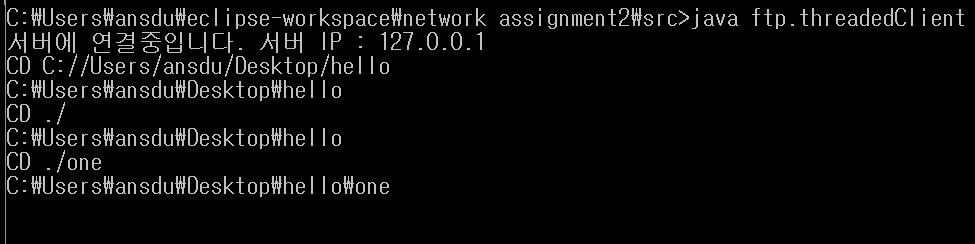


* + 1. CD ./

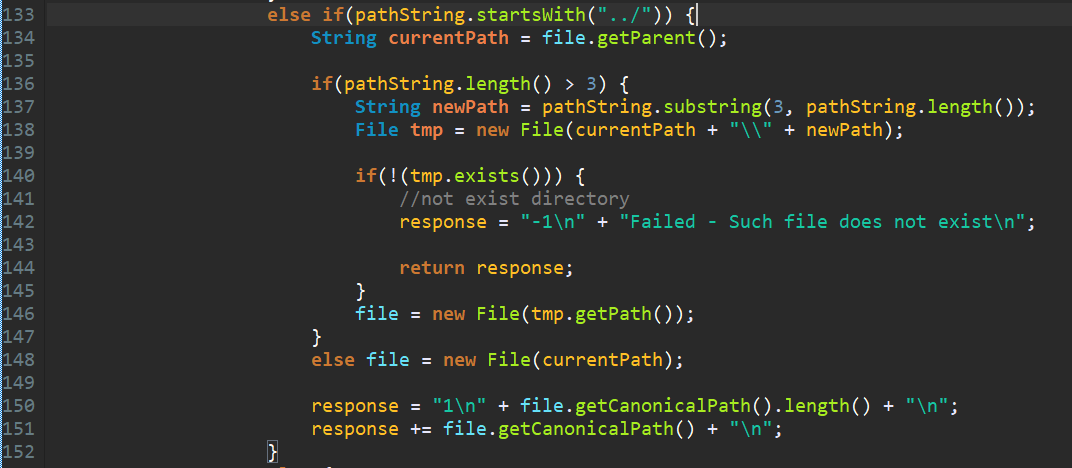


./ 뒤에 상대경로가 더 있다면 (116번째 줄의 if문) 상대경로를 가져와서 server의 현재 디렉토리를 바꿔주고 없는 경로라면(120번째 줄의 if문) state code -1과 에러 메시지를 client에게 보낸다. ./ 뒤에 상대경로가 없다면 바로 state code 1과 보낼 메시지의 길이, 현재 server의 디렉토리를 client에게 보낸다.

* 실행화면

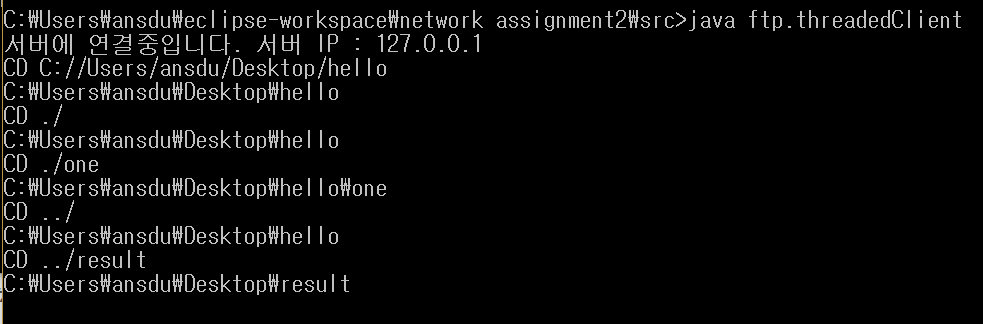


* + 1. CD ../

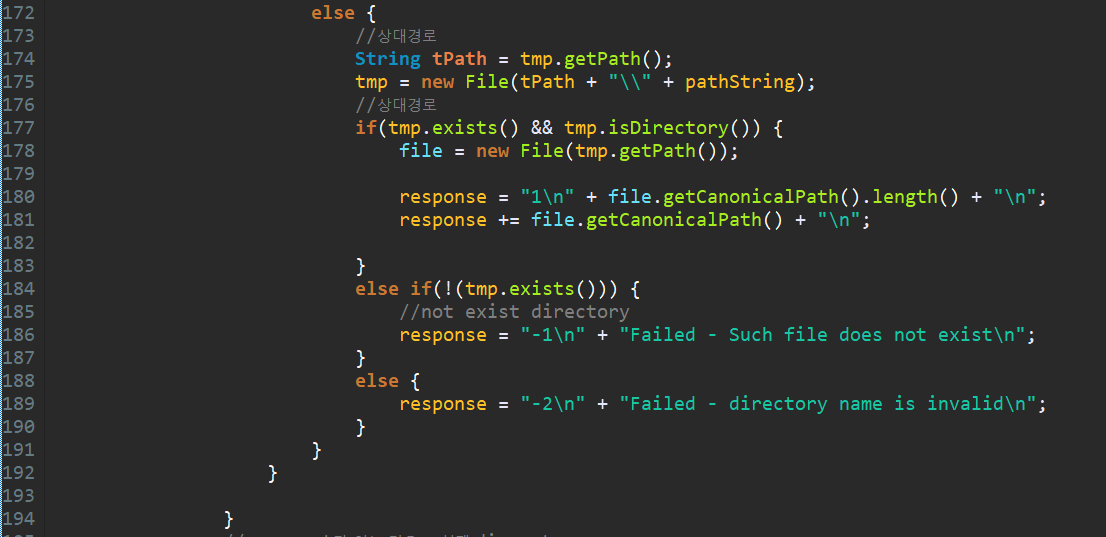


../ 뒤에 상대경로가 더 있다면 (137번째 줄의 if문) 상대경로를 가져와서 server의 현재 디렉토리를 바꿔주고 없는 경로라면(140번째 줄의 if문) state code -1과 에러 메시지를 client에게 보낸다. ../ 뒤에 상대경로가 없다면 바로 state code 1과 보낼 메시지의 길이, 현재 server의 디렉토리를 client에게 보낸다.

* 실행화면



* + 1. CD (상대경로)

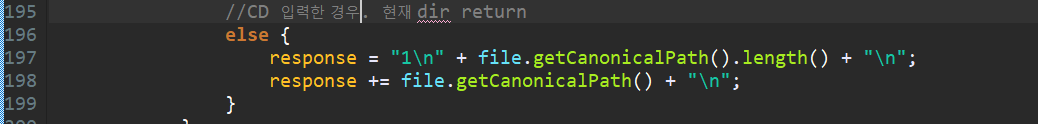


입력한 상대경로가 존재하고 디렉토리라면 server의 디렉토리를 바꿔주고 state code 1과 보낼 메시지의 길이, server의 현재 디렉토리를 client에게 보낸다. 존재하지 않다면 state code -1과 에러 메시지를 client에게 보낸다.

* 실행화면



* + 1. CD



state code 1과 보낼 메시지의 길이, server의 현재 디렉토리를 client에게 보낸다.

* 실행화면



그 외의 메시지를 client가 보냈다면 state code -2와 에러 메시지를 client에게 보낸다. (threadedServer.java에서 189번째 줄)

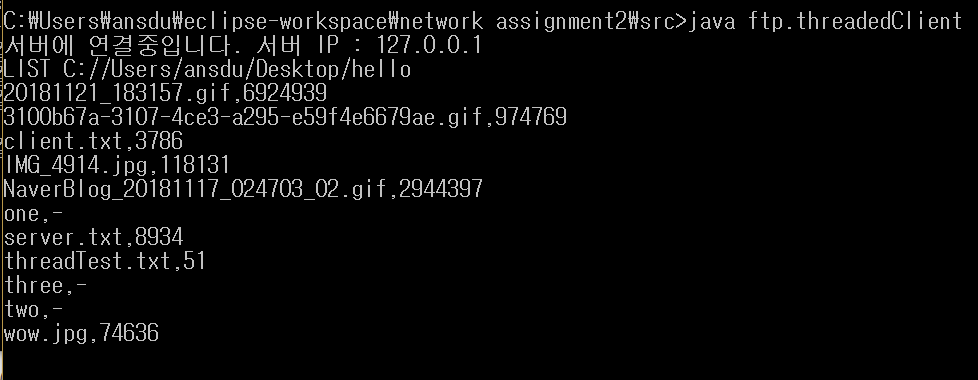
threadedClient.java에서 137번째 줄을 통해 state code가 1이라면 server에게서 받은 메시지를 print한다. state code -1이라면 143번째 줄을 통해 server에게서 받은 메시지를 print한다. state code -2이라면 149번째 줄을 통해 server에게서 받은 메시지를 print한다.

* 1. LIST
     1. LIST (절대경로)



절대경로에 해당하는 디렉토리가 존재한다면 state code 1과 보낼 메시지의 길이, 그 디렉토리에 있는 파일과 폴더들의 이름과 각각의 크기를 client에게 보내준다. (server의 현재 디렉토리는 바뀌지 않는다.) 디렉토리가 존재하지 않는다면 state code -1과 에러 메시지를 client에게 보내준다.

* 실행화면

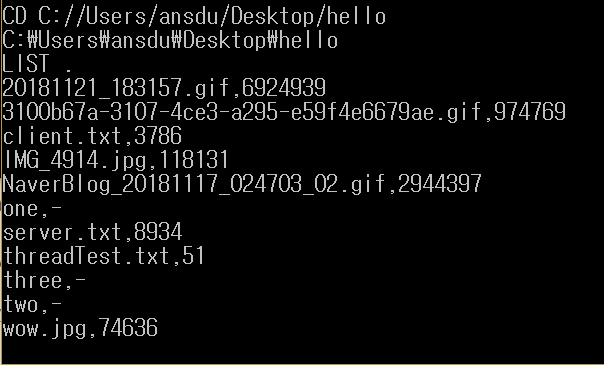


* + 1. LIST .

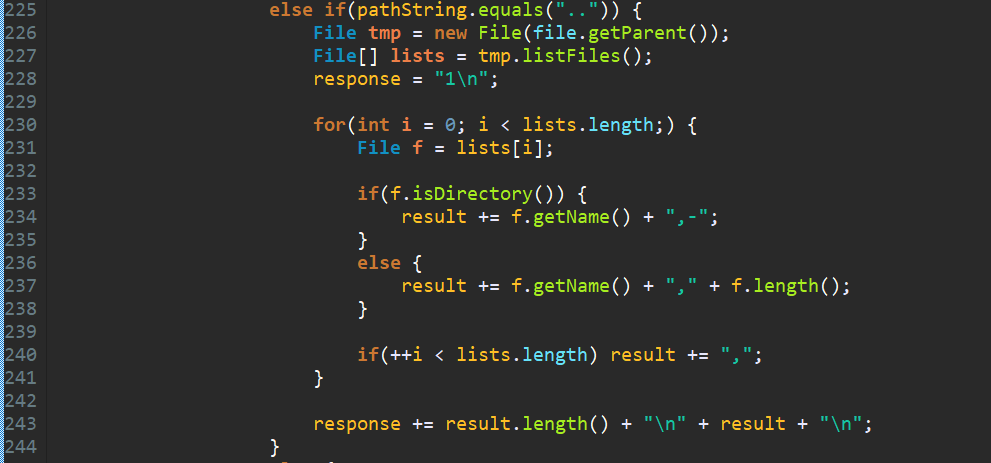


state code 1과 보낼 메시지의 길이, server의 현재 디렉토리에 있는 파일과 폴더들의 이름과 각각의 크기를 client에게 보내준다.

* 실행화면

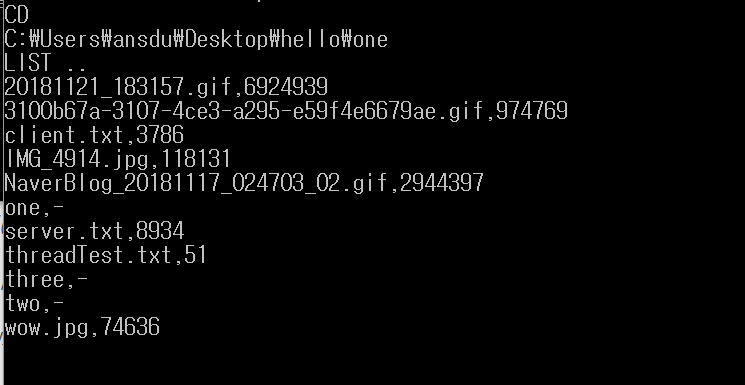


* + 1. LIST ..



state code 1과 보낼 메시지의 길이, server의 현재 디렉토리의 상위 디렉토리에 있는 파일과 폴더들의 이름과 각각의 크기를 client에게 보내준다. (server의 현재 디렉토리는 바뀌지 않는다.)

* 실행화면

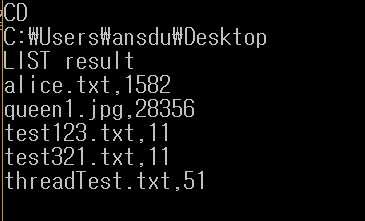


* + 1. LIST (상대경로)

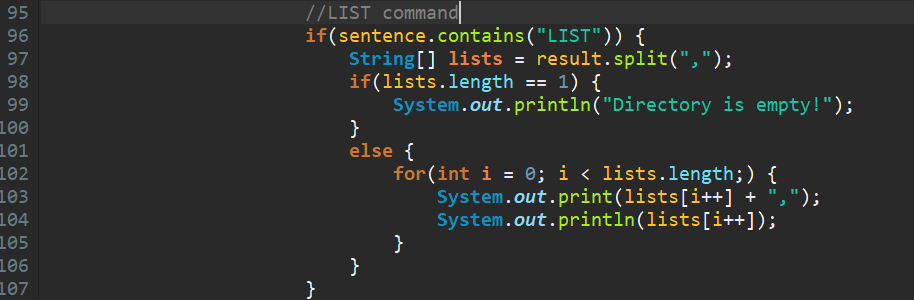


입력한 상대경로가 존재하고 디렉토리라면 state code 1과 보낼 메시지의 길이, 입력한 경로에 있는 파일과 폴더들의 이름과 각각의 크기를 client에게 보낸다. (server의 현재 디렉토리는 바뀌지 않는다.) 존재하지 않다면 state code -1과 에러 메시지를 client에게 보낸다.

* 실행화면



그 외의 메시지를 client가 보냈다면 state code -2와 에러 메시지를 client에게 보낸다. (threadedServer.java에서 305번째 줄)



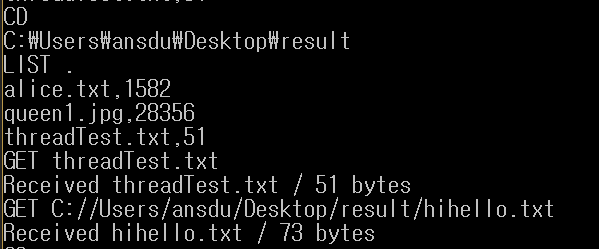
threadedClient.java에서 state code가 1이라면 server에게서 받은 메시지를 출력해준다. state code -1이라면 143번째 줄을 통해 server에게서 받은 메시지를 print한다. state code -2이라면 149번째 줄을 통해 server에게서 받은 메시지를 print한다.

* 1. GET (절대경로) / GET (파일)



client가 보낸 경로가 절대경로라면 그 경로에 해당하는 파일을 File class object로 선언해주고(321번째 줄, server의 현재 디렉토리는 바뀌지 않는다) 그냥 파일 이름을 보냈다면 File class object로 선언해준다(324번째 줄). 해당 파일이 존재한다면 state code 1과 파일 길이, 파일의 이름, 파일 data, 파일의 끝을 표시하는 ‘fileEnd’를 client에게 보내준다. 파일이 존재하지 않다면 state code -1과 에러 메시지를 client에게 보내준다.

* 실행화면

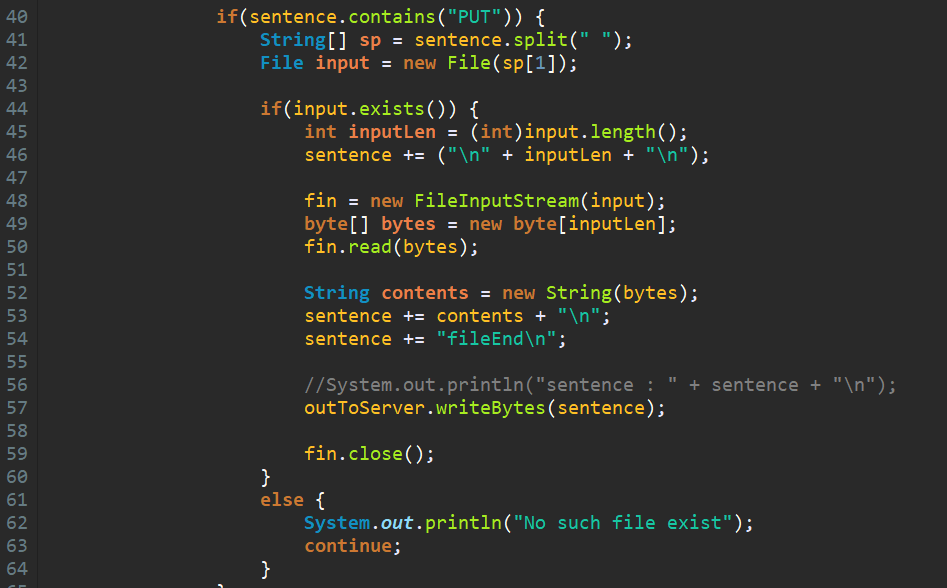


그 외의 메시지를 client가 보냈다면 state code -2와 에러 메시지를 client에게 보낸다. (threadedServer.java에서 351번째 줄)



threadedClient.java에서 state code가 1이라면 file의 이름을 filename 변수에 담고(110번째 줄) file의 길이를 length 변수에 담고(91번째 줄) server가 보낸 파일 data를 한 줄씩 읽고 파일을 써준다. 쓴 파일의 길이가 보낸 파일의 길이(length)와 같다면 성공적으로 받은 것이므로 성공 메시지를 출력하고 그렇지 않다면 쓰고 있던 파일을 지우고 에러 메시지를 출력한다. state code -1이라면 143번째 줄을 통해 server에게서 받은 메시지를 print한다.

* 1. PUT (파일) (client -> server)

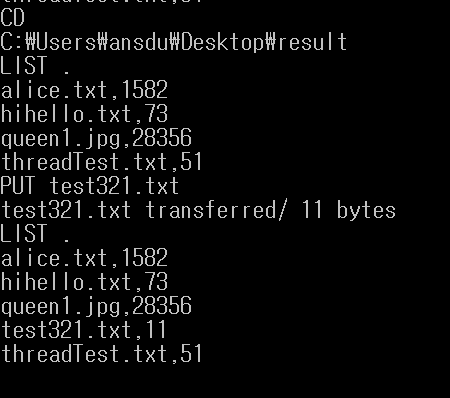


threadedClient.java에서 해당 파일이 있다면 파일의 길이와 파일 data, 파일의 끝을 표시하는 ‘fileEnd’를 server에게 보내준다. 파일이 존재하지 않다면 에러 메시지를 출력하고 다음 command를 기다린다.



threadedServer.java에서는 thread 동기화 문제를 해결하기 위해 synchronized block 내에서 PUT command를 처리해주도록 하였다. client가 보낸 파일의 길이를 fileLen 변수에 담고 (360번째 줄) server의 현재 디렉토리에 파일을 쓴다. 쓴 파일이 보낸 파일의 길이와 같다면 state code 1과 성공적으로 받았다는 메시지를 client에게 보내준다. 그렇지 않다면 쓴 파일을 지우고 state code -10과 에러 메시지를 client에게 보내준다.

* 실행화면



threadedClient.java에서 137번째 줄을 통해 state code가 1이라면 server에게서 받은 메시지를 print한다

command error라면 server에서 state code -20과 에러 메시지를 보낸다. threadedClient.java에서 161번째 줄을 통해 server에게서 받은 메시지를 출력한다.