

Programming Assignment – FTP Client and Server Program

1 개요

이 과제는 FTP 프로토콜이 동작하는 방식을 이해하고 이를 바탕으로 한 간단한 파일 전송 프로그램을 구현하는 것이다. 아래 그림에서 알 수 있듯이 FTP 프로그램은 기본적으로 명령어 채널과 데이터 채널을 별도로 사용한다. 명령 및 응답은 항상 명령어 채널을 사용하며, 파일 송/수신 시 데이터 전송을 위해서는 데이터 채널을 사용한다.

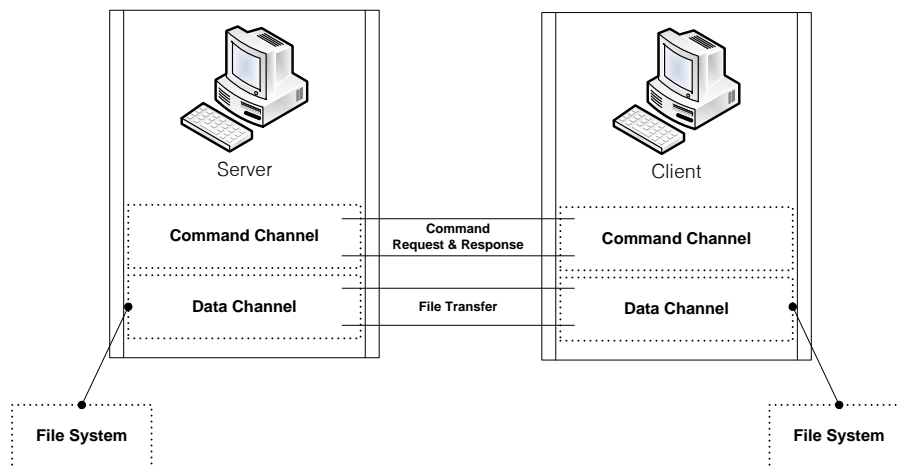


그림 1. FTP (File Transfer Protocol) 프로그램

- **Assignment #1** – 아래의 FTP 프로토콜 서버 및 클라이언트 프로그램을 Java blocking IO 와 thread를 이용하여 구현한다. 즉, 서버 측에서 각각의 command, data channel에 별도의 전담 thread로 할당되도록 설계한다.
- **Assignment #2** – 동일한 기능을 Java NIO를 이용하여 구현한다.

2 세부 사항

- Command channel: 클라이언트가 파일 송수신을 위해서 "LIST", "GET", "PUT", "CD"와 같은 명령어와 파라미터를 서버로 전송하고 결과 문자열을 받을 때 사용하는 채널이다.
- Data channel: 서버와 클라이언트 사이에서 PUT/GET 명령에 의해 파일 전송이 실제로 일어나는 채널이다.
- Command channel 과 data channel 은 서버-클라이언트 연결마다 1 개씩 생성되며, 한 번 생성된 channel 은 서버-클라이언트 연결 해제 전까지는 없어지지 않는다.
- 파일 전송 프로그램이 지원하게 될 명령어는 아래와 같이 동작하되, 파일 또는 디렉토리는 다음 네 가지 종류를 지원해야 한다.
 - (절대경로)
 - (상대경로)

- "." (마침표 - 현재 위치)
- ".." (마침표 2 개 - 상위 디렉토리)

| | | |
|------------|---|------------------------------------|
| CD (디렉토리명) | <ul style="list-style-type: none">• 서버에서의 '현재 디렉토리 위치'를 변경할 때 사용되는 명령어이다.• 이에 대해 서버는 "OK - 새로운 디렉토리 위치\n"로 응답한다. 단, 응답은 절대 경로이다. 이 응답은 클라이언트에 출력되어야 한다.• 만일 파라미터가 디렉토리가 아닌 경우 에러 메시지를 출력하도록 한다. | |
| 참고사항 | <ul style="list-style-type: none">• File 클래스의 exists(), isDirectory(), getCanonicalPath() 메소드• Path 클래스의 isAbsolute() 메소드• PrintWriter, BufferedReader 클래스 등 | |
| | | |
| | 윈도우 | 리눅스 |
| 명령 | CD C:\ftp\ | CD /home/username/ftp/ |
| 결과 | OK - C:\ftp\ | OK - /home/username/ftp/ |
| | | |
| | 윈도우 | 리눅스 |
| 명령 | CD subdir1 | CD subdir1 |
| 결과 | OK - C:\ftp\subdir1\ | OK - /home/username/ftp/subdir1/ |
| | | |
| | 윈도우 | 리눅스 |
| 명령 | CD .. | CD .. |
| 결과 | OK - C:\ftp\ | OK - /home/username/ftp/ |
| | | |
| | 윈도우 | 리눅스 |
| 명령 | CD C:\nodir\ | CD /nodir/ |
| 결과 | FAILED - Directory name is invalid | FAILED - Directory name is invalid |

| | |
|--------------|--|
| LIST (디렉토리명) | <ul style="list-style-type: none">• 서버가 유지하는 파일 리스트를 받아 올 때 사용한다.• 서버는 해당 디렉토리 내의 결과를 "(파일명),(파일크기)\n"로 응답한다. 이 응답은 클라이언트에 출력되어야 한다.• 디렉토리의 크기는 '-'로 표기한다.• 파일의 크기는 byte 단위로 표기한다.• 만일 파라미터가 디렉토리가 아닌 경우 에러 메시지를 출력하도록 한다. |
| 참고사항 | <ul style="list-style-type: none">• File 클래스의 listFiles() 메소드 |
| | |
| | 리눅스 (이하 윈도우 생략) |
| 명령 | LIST /home/username/ftp/ |

| | |
|----|--|
| 결과 | subdir1,- subdir2,- subdir3,- test.txt,13 test.jpg,71514 |
| 명령 | LIST . |
| 결과 | (현재 디렉토리 출력, 생략) |
| 명령 | LIST .. |
| 결과 | (상위 디렉토리 출력, 생략) |
| 명령 | LIST subdir1 |
| 결과 | a.txt,13 b.jpg,71514 c.txt,18 d.jpg,25584 |
| 명령 | LIST subdir2 |
| 결과 | Empty Directory |
| 명령 | LIST nodir |
| 결과 | FAILED - Directory name is invalid |

| | |
|-----------|--|
| GET (파일명) | <ul style="list-style-type: none"> • 서버로부터 지정된 파일을 수신하고자 할 때 사용한다. • 파일은 클라이언트의 현재 디렉토리에 저장된다. • 클라이언트가 서버로 Command channel 을 통해 'GET (파일명)'을 전송하면, 서버는 해당 파일의 크기를 응답한다. • 클라이언트는 해당 파일의 크기만큼 Data channel 을 통해 파일의 데이터를 받는다. (파일 크기를 이용하여 파일의 끝을 계산해서 다운로드 완료 처리). • 만일 서버에 해당 파일이 없거나 경로가 정확하지 않은 경우 에러 메시지를 출력하도록 한다. • 만일 클라이언트에 해당 파일이 이미 존재하는 경우 에러 메시지를 출력하도록 한다. |
| 참고사항 | <ul style="list-style-type: none"> • File 클래스의 length() 메소드 • FileInputStream, FileOutputStream, InputStream, OutputStream 등 |
| 명령 | GET /home/username/ftp/subdir1/a.txt |

| | |
|----|---------------------------------------|
| 결과 | Received a.txt / 13 byte(s) |
| 명령 | GET subdir1/b.jpg . |
| 결과 | Received b.jpg / 71514 byte(s) |
| 명령 | GET subdir1/a.txt |
| 결과 | FAILED - File already exists (Client) |
| 명령 | GET nofile.txt |
| 결과 | FAILED - File does not exist (Server) |

| | |
|-----------|--|
| PUT (파일명) | <ul style="list-style-type: none"> • 서버로 지정된 파일을 송신하고자 할 때 사용한다. • 파일은 서버의 현재 디렉토리에 저장된다. • 클라이언트가 보낸 파일 크기를 알아낸 후, 서버로 Command channel 을 통해 'PUT (파일명) (파일 크기)'를 전송하면, 서버는 준비가 되었음을 알린다. • 서버는 해당 파일의 크기만큼 Data channel 을 통해 파일의 데이터를 받는다. (파일 크기를 이용하여 파일의 끝을 계산해서 다운로드 완료 처리). • 만일 클라이언트에 해당 파일이 없거나 경로가 정확하지 않은 경우 에러 메시지를 출력하도록 한다. • 만일 서버에 해당 파일이 이미 존재하는 경우 에러 메시지를 출력하도록 한다. |
| 명령 | PUT a.txt |
| 결과 | Sent a.txt / 13 byte(s) |
| 명령 | PUT b.jpg . |
| 결과 | Sent b.jpg / 71514 byte(s) |
| 명령 | PUT a.txt |
| 결과 | FAILED - File already exists (Server) |
| 명령 | PUT nofile.txt |
| 결과 | FAILED - File does not exist (Client) |

3 유의 사항

- 사용 OS 는 리눅스 또는 윈도우로 한다. (원하는 OS 택일한 후 문서에 기술)
- 프로그래밍 언어는 Java 만 허용한다.
- 제출 소스 코드에서의 서버 및 클라이언트의 IP 주소 기본값은 "127.0.0.1", Command channel 및 Data channel 의 기본 포트 번호는 각각 "2121", "2020"으로 한다.
- 단, 프로그램 실행 시 포트 번호를 파라미터로 받는 경우, 그 값으로 한다.
 - Java FTPServer <Command Port> <Data Port>
 - Java FTPClient <Command Port> <Data Port>

4 소스 코드 제출

- 소스 및 설계 문서 파일('6. 구현 문서 제출' 참고)을 포함하여 jar/zip 형태로 압축하여 파일명을 "ftp-이름-assignment#jar/zip"으로 하여 제출한다. (#은 1 또는 2)
- 이 압축 파일에는 컴파일 된 class file 이나 executable 파일이 포함되지 않도록 유의한다.
- 과제 구현에 대한 문의는 담당 조교 (이한영: esteir4216@hanyang.ac.kr)에게 이메일로 한다.

5 제출 기한

- 1 차 과제에 대한 소스 코드 제출 시한은 **11 월 20 일 월요일 23:59** 까지이다.
- 21 일 화요일 23:59 이내(24 시간 이내)에 제출한 경우 50%를, 22 일 수요일 23:59 까지(48 시간 이내) 제출한 경우 25%를 인정한다. 이후 제출은 인정하지 않는다.
- 2 차 과제에 대한 제출 시한은 **12 월 4 일 월요일 23:59** 까지이다.
- 5 일 화요일 23:59 이내(24 시간 이내)에 제출한 경우 50%를, 6 일 수요일 23:59 까지(48 시간 이내) 제출한 경우 25%를 인정한다. 이후 제출은 인정하지 않는다.

6 설계 및 구현 문서 제출

- 1 차 과제에 대한 문서의 인쇄본을 **11 월 21 일 화요일 수업 시작 전에** 제출한다.
- 2 차 과제에 대한 문서의 인쇄본을 **12 월 5 일 화요일 수업 시작 전에** 제출한다.
- 이 문서에서는 프로그램의 설계와 구조의 기술 외에, 제출된 프로그램을 컴파일 및 실행하는 방법과 클라이언트-서버 동작 절차도 단계적으로 정확하게 기술하여야 한다.
- 프로그램 test 시 기술된 절차를 그대로 따랐음에도 불구하고 프로그램이 수행되지 않는 경우에는 제출 프로그램이 동작하지 않는 것으로 간주한다.