

Programming Assignment 1: Implementing an FTP Client

A) First you need to install an FTP server on your local machine. Follow the steps below to do so:

Microsoft Windows:

- 1) Download FileZilla Server from here: <http://filezilla-project.org/download.php?type=server>
- 2) Click on the downloaded executable and follow the on-screen instructions to install the FTP Server.
- 3) Create an FTP user name and password (Edit → Users)
- 4) During the previous step you will be asked to add a local directory to be your ftp directory, create a new directory somewhere on your machine and then within FileZilla select that directory to be your FTP home directory. Make sure you grant write permissions to be able to upload files later on.

Ubuntu:

- 1) Install vsFTPD Server
> sudo apt-get install vsftpd
- 2) Once the FTP server is installed, a user 'ftp' is created with the default home FTP directory '/srv/ftp'

- You are done setting up your FTP Server on your local machine. Now test your server by connecting to it from the command line using the following command:

> ftp localhost

- Enter the user name and password. (**Windows:** use the username and password you created in step 3.

Ubuntu: Enter the default user name and leave the password blank.

- Try some of the ftp '**command line**' commands. For all available commands, type:

> help

For a quick help on each command, type:

> help 'command_name'

(Remember, these commands are **NOT** the same as the native FTP protocol commands used within the FTP request message structure; for a list of native FTP commands, consult the FTP section in the book or the RFC 959).

- Create a simple text file 'ftp_test.txt' on your local machine and upload it to the FTP server using the 'put' command over the command line (Make sure this file is located at the same directory where you issued the 'ftp localhost' command:

> put ftp_test.txt

B) Once your FTP server is installed successfully. Move on to implementing the FTP client:

To make things easier for you, I am providing an incomplete FtpClient.java class. All you need to do is fill up the missing snippets of code.

For native FTP commands and reply codes please refer to RFC 959 (<http://www.ietf.org/rfc/rfc959.txt>).

NOTE 1: If you are using ubuntu, make sure to unset the FILEZILLA flag.

NOTE 2: To enable debugging messages set the DEBUG flag to true

NOTE 3: There are two data transfer modes in FTP, active and passive, please consult RFC 959 for further details on how each mode works. We will use the passive mode in this assignment. For the passive mode to work, you need to disable IPv6 feature in your OS if it is enabled. Use this link (<http://support.microsoft.com/kb/929852>) to do so; simply download the 'Disable IPv6' fix it msi, install the fix and restart your PC.

NOTE 4: Make sure you turn off your firewall if your FtpClient is refusing to connect to FTP server. To test your FtpClient separately, you can create a test class with a main function. Try to instantiate an FtpClient, connect using your FTP user name and password, and download a file that you already uploaded to your FTP home directory:

```
package webserver;
public class FtpTest {
    public static void main(String args[]) {
        ...
    }
}
```

C) Extend your web server to act as an FTP Client when requesting text files only (.txt). Mainly, when you request a text file (.txt) from within your web browser, the web server will not have a copy of this file. It will instantiate an FtpClient, retrieve the text file from your local FTP server and then send it back to your web browser as an HTTP response.

More specifically, you will need to modify the following snippets of code:

```
// Construct the response message.
String statusLine = null;
String contentTypeLine = null;
String entityBody = null;
if (fileExists) {
    statusLine = "?";
    contentTypeLine = "Content-type: " +
        contentType( fileName ) + CRLF;
} else {
    // if the file requested is any type other than a text (.txt) file, report
    // error to the web client
    if (!contentType(fileName).equalsIgnoreCase("")) {
        statusLine = "?";
        contentTypeLine = "?";
        entityBody = "<HTML>" +
            "<HEAD><TITLE>Not Found</TITLE></HEAD>" +
            "<BODY>Not Found</BODY></HTML>";
    } else { // else retrieve the text (.txt) file from your local FTP server
        statusLine = "?";
        contentTypeLine = "?";

        // create an instance of ftp client
        ?

        // connect to the ftp server
        ?

        // retrieve the file from the ftp server, remember you need to
        // first upload this file to the ftp server under your user
```

```

        // ftp directory
        ?

        // disconnect from ftp server
        ?
        // assign input stream to read the recently ftp-downloaded file
        fis = new FileInputStream(fileName);
    }

}

// Send the entity body.
if (fileExists) {
    sendBytes(fis, os);
    fis.close();
} else {
    if (!contentType(fileName).equalsIgnoreCase(?)) {
        os.writeBytes(?);
    } else {
        sendBytes(fis, os);
    }
}
}

```