Project 1 - Mobile FTP Client / Server

Dates:

- Assigned: 2021-10-14 09:00:00(UTF+8)
- Deadline: 2021-11-28 23:59:59(UTC-12)

TAs:

- Nan Hua (huan19@fudan.edu.cn)
- Sen Yang (senyang19@fudan.edu.cn)

Introduction

The purpose of this project is to give you experience in developing **File Transfer Protocol**(FTP). You will write an Android client and server using a subset of the File Transfer Protocol (FTP) —RFC 959 (链接到外部网站。). At the end of this project, you will finish an amazing file transfer app!

Why are we doing this? As we know, software school has a tradition, that is, we have to submit our labs and projects to our school FTP server. So it's a good way to commemorate FTP, as it is gradually drowning in history(Just a joke ^_^). Actually, FTP plays an important role in the early stage of the Internet because of its convenience for transmission. Implementing such a protocol strictly following the RFC can make you have a deeper understanding of the Internet.

That's it. We're going for it! You will have a cool mobile FTP client and server :-) But be prepared, this is a multi-people project and it has a lot of depth—your skills will be exercised perhaps to their limit! So start early and feel more than welcome to ask questions. This is a hard project, so both TAs and the teacher will be more than welcome to help you debug, design, and provide hints in working on this project.

Project Outline

During the course of this project, you will do the following:

- Implement an Android FTP client/server to download/upload files.
- Design and implement some smart(?) strategies to accelerate file transmission and enhance robustness.

Project Requirements

- This is a group project done by **1-3 students**. You must find exactly one partner for this assignment. In case there is an odd number of people in the class and you are left out, please contact the TAs.
- And make sure at least one of your group has an **Android device**, otherwise, you have to use an emulator, which might be more difficult but less efficient.
- All submissions are due no later than 11:59 PM on the due dates. Turn in your submission by creating an archive containing all the applicable code and documentation. The archive name should be `ID1-ID2`, where `ID1` and `ID2` are the student IDs of the group members.
- Checkpoint 1: implement the Android client application and the client is able to upload or download files of different sizes.
- Checkpoint 2: implement the Android server application and the server can handle different requests for file uploading and downloading.
- Checkpoint 3: design and implement the strategies to improve the transmission.

Overview

The File Transfer Protocol (FTP) is a standard communication protocol used for the transmission of computer files from a server to a client through a computer network. FTP is built on a client-server model architecture using separate control and data connections between the client and the server.

THE FTP MODEL

With the above definitions in mind, the following model (shown in Figure 1) may be diagrammed for an FTP service.

```
                                      -------------
                                      |/---------\|
                                      ||   User  ||    --------
                                      ||Interface|<--->| User |
                                      |\----^----/|    --------
            ----------                |     |     |
            |/------\|  FTP Commands  |/----V----\|
            ||Server|<---------------->|   User  ||
            ||  PI  ||   FTP Replies  ||    PI   ||
            |\--^---/|                |\----^----/|
            |   |    |                |     |     |
--------    |/--V---\|      Data      |/----V----\|    --------
| File |<--->|Server|<---------------->|  User   |<--->| File |
|System|    || DTP  ||   Connection   ||   DTP   ||    |System|
--------    |\------/|                |\---------/|    --------
            ----------                -------------

            Server-FTP                   USER-FTP

NOTES: 1. The data connection may be used in either direction.
       2. The data connection need not exist all of the time.
```

Tiny FTP Client / Server

You don't have to implement all FTP commands. To make this project easier, the following minimum implementation is required for your server:

- USER - Authentication username.
- PASS - Authentication password.
- PASV - Enter passive mode.
- QUIT - Disconnect.
- PORT - Specifies an address and port to which the server should connect.
- TYPE - Sets the transfer mode (ASCII (链接到外部网站。)/Binary (链接到外部网

站。)).
- MODE - Sets the transfer mode (Stream, Block, or Compressed).
- STRU - Set file transfer structure.
- RETR - Retrieve a copy of the file.
- STOR - Accept the data and to store the data as a file at the server site.
- NOOP - No operation (dummy packet; used mostly on keepalives).

Grading

This information will give you a high-level view of how points will be allocated when grading this project.

Checkpoint 1: 35 points

The grade in this section is intended to test the ability to write a simple FTP client on Android.

- FTP Passvie Mode: 5 points
- File Transfer: 10 points
    - upload a specific file: 5 points
    - download a specific file: 5 points
- Folder Transfer: 10 points
    - upload a folder and internal files: 5 points
    - download a folder and internal files: 5 points
- Robustness: 10 points
    - handle the illegal operation of user
    - display error/warning messages

Checkpoint 2: 35 points

The grade in this section is intended to test the ability to write a simple FTP server on Android.

- Authorization: 5 points
    - anonymous / legal / illegal account

- File Transfer: 10 points
    - handle the upload of a specific file: 5 points
    - handle the download request of a specific file: 5 points
- Folder Transfer: 10 points
    - handle the upload request of a folder and internal files: 5 points
    - handle the download request of a folder and internal files: 5 points
- Robustness: 10 points
    - handle the illegal requests from a client:
    - display error/warning messages

Checkpoint 3: 15 points

The grade in this section is intended to evaluate your optimization strategies. The metric is **Transmission Time**. In other words, you need to compare the transmission time before and after applying your optimization.

- You need to design an experiment to evaluate your optimization strategies and make a specific analysis of your experiment results.
- To make it simple, you only need to optimize the transmission of **either** big files **or** small files.
- You can open at most **two** connections between client and server to accelerate the transmission.
- You can get the grade if both your strategies and your evaluation experiment make sense.

Usability: 5 points

As a mobile app, good interaction between users and app is important. You can get all points if your app can be easily operated and display the correct information.

Style: 10 points

Poor design, documentation, or code structure will probably reduce your grade by making it hard for you to produce a working program and hard for the grader to understand it. And you have to submit a **video** to introduce your great work, including a brief introduction of your code, how your client and server works, the process of file transfer, evaluation of your performance optimization, etc.

- Document: 4 points
- Video: 4 points
- Code structure: 2 points

For three-people group: 50 points

Additional commands you need implement, each command is 5 points:

- CDUP
- CWD
- DELE
- LIST
- MKD
- RMD
- RNFR
- RNTO

Another two test cases, each test case is 5 points:

1. Anonymous user can not delete any files or folders of the user *test*
2. Anonymous user can not change any files or folders of the user *test*

Test Cases

Here is the description of test cases:

- Anonymous account
- Default account: `test` / `test`
- Illegal account: `pikachu` / `pikachu`
- Folder1
    - 10 big files(each size is larger than 500MB)
- Folder2

- 10000 small files(each size between 10KB and 50KB)

Handin

Your submissions should contain the following files:

- `client.apk` / `server.apk` - FTP client and FTP server you implement. You can integrate two apps into one app.
- `src` - Folder of your source code: Files ending with .java, .c, .h, etc. only.
- `README.md` - File containing a brief description of the design of your current implementation, and precise instructions on how to run your app.
- `demo.mp4` - The video about your work

Getting Started

This section gives suggestions on how to approach the project. Naturally, other approaches are possible, and you are free to use them—at your own peril.

Start early! We cannot stress how important it is to start early in a project. It will give you more time to think about the problems, discuss with your classmates, and ask questions.

Read the RFCs selectively. RFCs are written in a style that you may find unfamiliar. However, it is wise for you to become familiar with it, as it is similar to the styles of many standards organizations. We don't expect you to read every page of the RFC, especially since you are only implementing a small subset of the full protocol, but you may well need to re-read critical sections a few times for the meaning to sink in.

Follow the standard. Be liberal in what you accept, and conservative in what you send. Following the guiding principle of Internet design will help ensure your server works properly with many different and unexpected client behaviors.

Code quality is important. Make your code modular and extensible where possible. Split the problem into different modules. Tackle one problem at a time and build on functionality only once it is completely solid and tested. This reduces the number of places you have to search to find the source of a bug. Define the interfaces between the modules also helps you and your partner make progress in parallel.

References

- FTP RFC (链接到外部网站。)
- URL RFC (链接到外部网站。)
- Android Developer Document (链接到外部网站。)

评分标准

评分标准

标准	等级	得分
此标准已链接至学习结果单人组队单人完成PJ，要求同二人组队一样，需完成所有功能。PJ满分为100分，评分时要求会适当降低。	100 得分满分 0 得分 无分数	100 分
此标准已链接至学习结果两人组队二人组队完成PJ，PJ总分为100分。小组成员的贡献占比总和 = 100% * 2。最终个人PJ得分 = 小组PJ得分 * 个人贡献占比，且不超过个人PJ满分100分。	100 得分满分 0 得分 无分数	100 分
此标准已链接至学习结果三人组队三人组队完成PJ，会在二人组队的基础上增加50分的功能需求，PJ总分为150分。小组成员的贡献占比总和 = 100% * 3。最终个人PJ得分 = 小组PJ得分 * 个人贡献占比 * 2/3，且不超过个人PJ满分100分。	100 得分满分 0 得分 无分数	100 分
总得分： 300 ， 满分 300		