**COMP3015 Data Communications and Networking**

**Course Project Design Report**

SUN Jing xuan

14252392

2017/3/29

**1. General description of functions implemented**

**1.1 Basic functions**

• Password function:

For the Client, on inputting the correct password set by the Server, the Client get the access to the share folder and its sub folders.

If the input password is wrong after checking by the Server, the Client has the chance to input again until the password is right.

For the Server, he is able to set the password before connecting to the Client. That is, setting password function is private to the Server.

If the Server starts the connection with the Client without setting the password, the program will use a default password.

• File browsing function (file names and file sizes):

On sending the inquire, the Client will get a list of information of both the files and the directories under the default share folder. The information includes file name/directory name (remark: or absolute path) and the file/directory size in bytes. The list of information will be displayed on the console.

• Single file downloading function:

On inputting the legal demanded file name, the application will download the file to a default folder of the Client.

If the Client inputs a directory instead of a file name, he will be asked to input again.

If the input file name is illegal, that is there is not such a file in the Server’s share folder, the Client will be asked to input the file name again until he enters a legal one.

If the demanded file already exists in the default folder, the program will ask the Client if he still wants to download and the Client will have a chance to think and choose again.

**1.2 Additional functions**

• Directory downloading function:

On inputting the legal source and destination directory path, the Client will receive in the input destination the whole source folder he wants. Received folder structure would be the same as the source one. That is, if the downloaded folder has sub folders, the files in the sub folders will also be recursively downloaded.

If the input path is illegal, that is, there is no such file in the Server side share folder, the Client will be asked to input again until the path becomes legal.

If the destination folder does not exist, the program will help the Client to create the download folder automatically.

• Multiple users access:

The Multithread Server supports different Clients to access the share folder as well as carry out the above functions in the same time.

• Multiple files concurrent downloading function: on going

**2. Detailed application protocol**

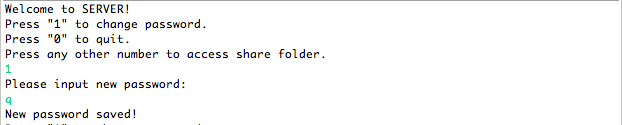
**2.1 User commands**

**Server side:**

Before setting up the connection between Server and Client, the Server is able to choose whether to set a new password or directly establish the connection. The program offers 3 choices including quit program. The Server should input the corresponding function number.

This is the only part requiring Server to input command. The Server user should key in an integer to make choice.

When input “1”, the Server user should key in the new password string.



When input “0”, the Server program will quit automatically.

**Client side:**

The whole program will establish the connection between the Server and the Client in first place, and only check password afterwards. The connection process is completed in the constructor function of FileServer and FileClient programs, including Server’s creating server socket and accept client socket. When running the Client program, the connection is automatically set up.



The Client will be ordered to input password. The Client user should key in a string.

If successful, the Client will be ordered to input user name. The Client user should key in a string.

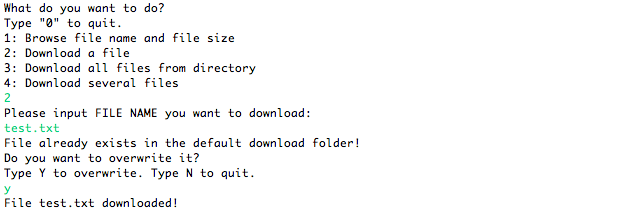


The Client program will give 4 choices of functions. The Client user should key in an integer to make choice.

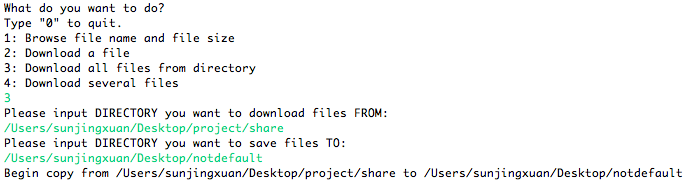
When input “1”, the information will be returned automatically.



When input “2”, the Client user should key in a string of file name. If file exists, he should input “y/Y” or “n/N” to choose if still download.



When input “3”, the Client user should key in a string of source folder path. If successful, he should key in a string of destination folder path.



When input “0”, the Client program will quit automatically.

**2.2 Mapping between user commands and internal commands**

Here we discuss the Client side in detail, because the Server side does not involve so much mapping.

There is a command string for each user command. The command structure is: function number + separator + input information. The separator symbol is “,”. On inputting the choice of functions, that is, the Client user inputs an integer, the scanner simply scans the integer and accepts it as the function number. The input information refers to the file name or source and destination directory path.

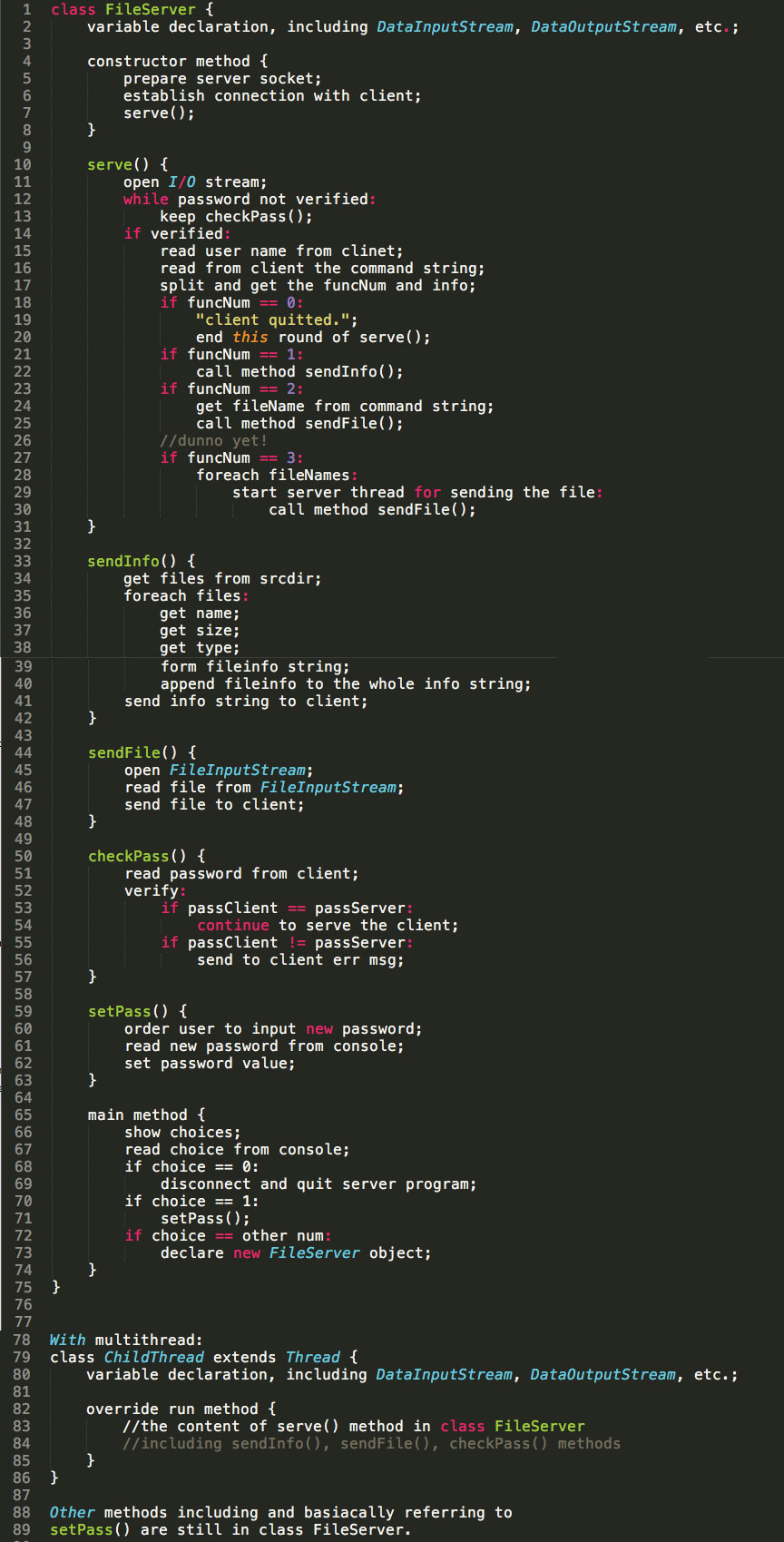
After formation of the command string within the Client side program, the string will be sent to Server through socket.

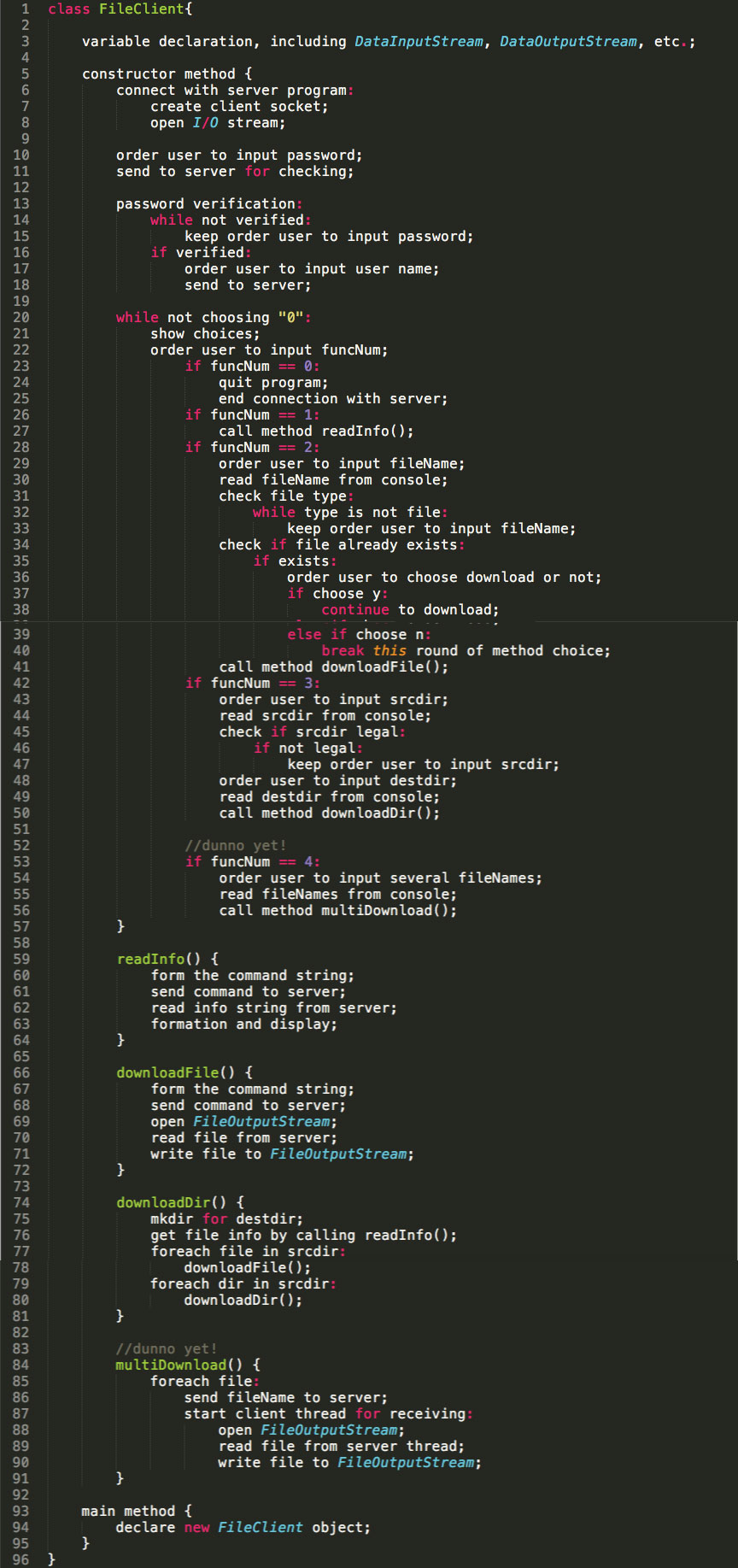
After getting the command string through socket, the Server program will again separate the whole string into function number and input information by chopping the string with the separator. The function number will be used to choose a certain function and the input information will become the parameter list of the chosen function.

Here is the mapping:

|  |  |  |
| --- | --- | --- |
| User command | Client program | Server program |
| 0 | funcNum = 0: quit |  |
| 1 | funcNum = 1: readInfo | funcNum = 1: sendInfo |
| 2 + file name | funcNum = 2: downloadFile | funcNum = 2: snedFile |
| 3 + src dir, dest dir | funcNum = 3: downloadDir, calling func 1 and 2 | funcNum = 1 and 2 |

**3. Flow chart and pseudo code**

**3.1 Server program**

**3.2 Client program**