Xi'an Jiaotong-Liverpool University

西交利物浦大學

Report of Networking Project

Author Xu Zhao

ID 1927631

Module CAN201 Introduction to Networking

Teacher Fei Cheng

Date 29^{th} / Nov / 2021

Abstract

This is a report about a file sharing application of CAN201 assignment. In this report, there is an introduction about this coursework to introduce the background of the file sharing background and the coursework. Next, details about the personal solution to the coursework are followed. Methodology of protocols, functions and ideas is explained in the form of diagrams. How I implement the program is also provided. Finally, the testing results are given.

Introduction

Requirement

The project requires to design a file sharing program using Python Socket network programming. Firstly, the program can be applied to files with any format and folders with a max size of 500MB. Secondly, the transfer speed should be as faster as the program can. Meanwhile, the new files/folders and changed files/folders can be synchronized automatically. Thirdly, the program should reload files/folders if they are disconnected, which means to resume from interruption. Finally, the files should not have any errors. The project should use TCP in transport layer protocol.

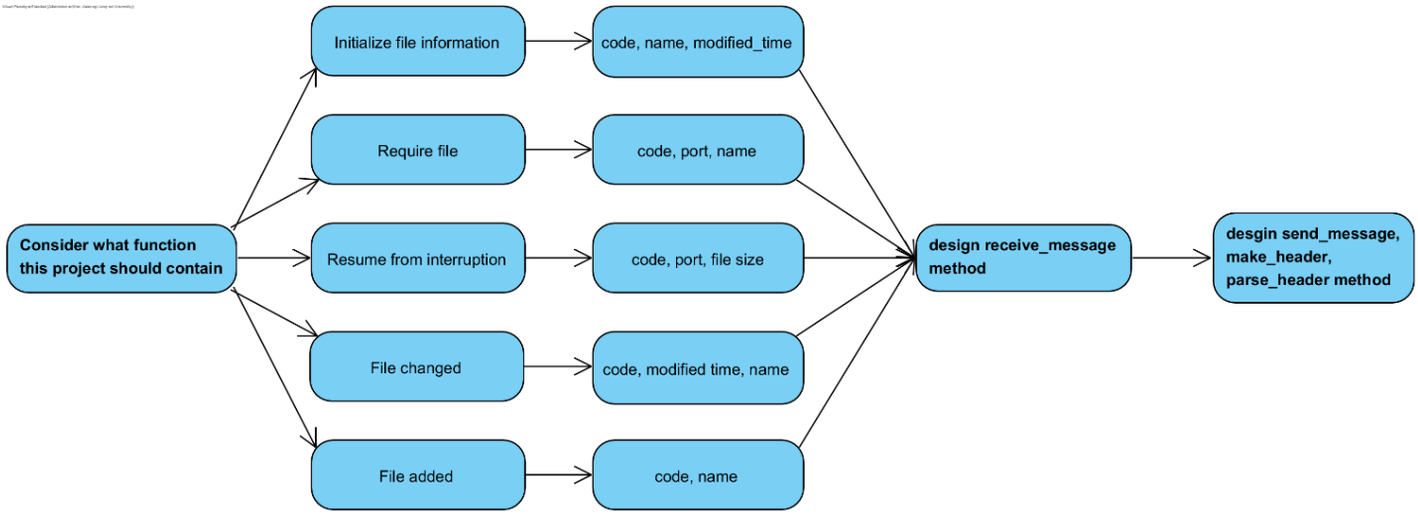
Background/literature review

File sharing function are widely used in many applications today[1]. Many applications provide this service professionally, such as BaiduNetDisk, iCloud, Google Drive and Mega. Besides this function is necessary to almost any applications. Although wechat is a social application, it needs this function because people sometimes share small files while chatting.

In this coursework, I design a program which satisfy all the requirements above after many researches and tests and write a report to introduce it.

Methodology

Protocol:



Proposed functions:

receive_message():

This function is the most important part of the protocol. The instruction code is connected to the function of the program.

function	Instruction_code	Necessary paramaters
start	10	Name, port, mtime, positon
Request send file	0	Name, port, position
Modify file	1	Name, port, mtime, position
Add file	2	Name, port, mtime, position
Send folder	3	Name, port, position
Add folder	4	Name, port, mtime, position
Reload file	5	Name, port, mtime, position
Reload folder	6	Name, port, mtime, position
Continue receive file based on received size	7	Name, port, mtime, positon
Continue receive folder based on received size	8	Name, port, mtime, positon

The below two functions take an important part in identification and operation.

Scan_reload(): It scans share folder and pack or send messages according to protocol

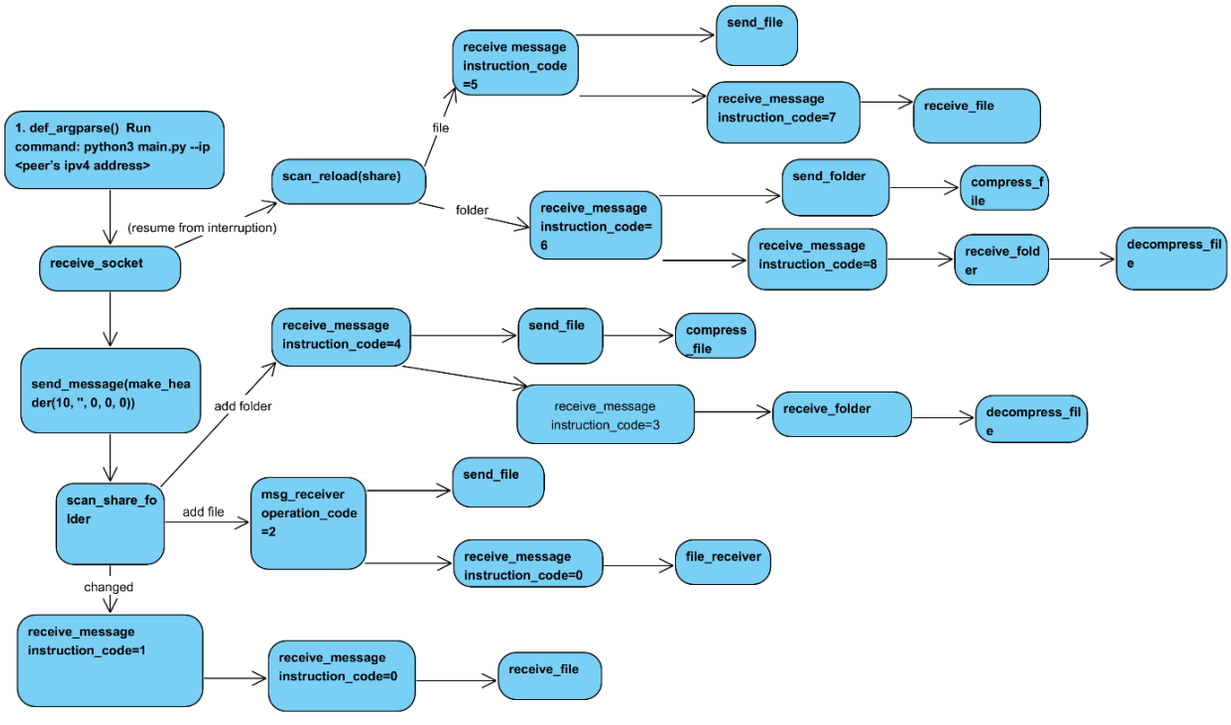
Scan_share_folder(): it keeps scanning share folder and check if there is update in this folder.

Implementation

Implementation

First, according to lab3, I try to write a program which can send and receive a picture using knowledge about open and close a file. Next, improve the program to make it manage to send and receive a big file by dividing it into many packages. After analyzing the function of the application, design a protocol. According to the coursework requirement, add many functions to satisfy such as resume from interruption and avoid package loss and package error.

program flow charts



programming skills

oop: I write different functions to satisfy the requirement and call them when necessary.

For example, in send_folder() I call compress_file() to transform it into a file and call send_file() and in receive_folder() I call receive_file and decompress_file() to transform it back to folder.

```
def send_folder(folder_path, port, position):
    compress_file(folder_path)
    name = folder_path + '.zip'
    send_file(name, port, position)
```

```
def receive_folder(folder_name, receive_socket, mtime, position):
    receive_file(folder_name + '.zip', receive_socket, mtime, position)
    decompress_file(folder_name, mtime)
```

Difficulties

In this coursework, I meet many problems which cannot use knowledge in class to solve. So, I go to Baidu and csdn to find the solution.

I find how to compress and decompress a file on csdn[2].

```
import os,sys
import zipfile

def zip_dir(dirname, zipfilename):
    filelist = []
    #Check input ...
    fulldirname = os.path.abspath(dirname)
    fullzipfilename = os.path.abspath(zipfilename)
    print "Start to zip %s to %s ..." % (fulldirname, fullzipfilename)
    if not os.path.exists(fulldirname):
        print "Dir/File %s is not exist, Press any key to quit..." % fulldirname
        inputStr = raw_input()
        return
    if os.path.isdir(fullzipfilename):
        tmpbasename = os.path.basename(dirname)
        fullzipfilename = os.path.normpath(os.path.join(fullzipfilename, tmpbasename))
    if os.path.exists(fullzipfilename):
        print "%s has already exist, are you sure to modify it ? [Y/N]" % fullzipfilename
        while 1:
            inputStr = raw_input()
            if inputStr == "N" or inputStr == "n" :
                return
            else:
                if inputStr == "Y" or inputStr == "y" :
```

I also find how to design protocol on csdn[3].

Testing and results

Test environment

2 PC that emulates the Linux kernel with Virtual box.

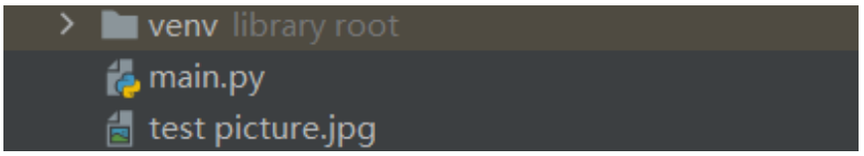
Test plan

Before my code runs in the virtual machine, I test some functions.

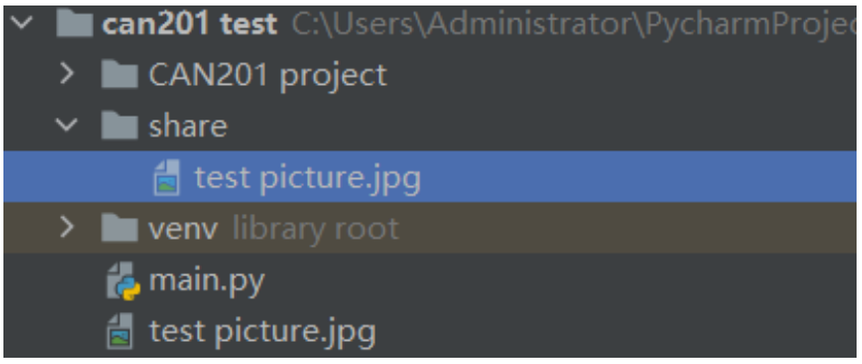
```
#test
#name = 'test picture.jpg'
#mtime = os.path.getmtime(name)
#p = threading.Thread(target=receive_file(), args=('test picture.jpg', receive_socket, mtime, 0))
#p.start()
#send_file('test picture.jpg', file_port, 0)

#compress_file('CAN201 project')
#decompress_file('CAN201 project')
```

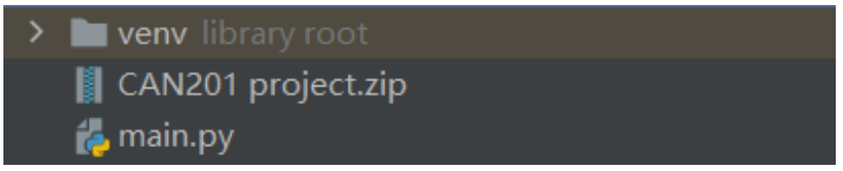
I put a picture in the same directory with main file and transfer it.



Run the code. The picture is transferred in share folder



Put a folder CAN201 project in the directory and run compress_file()



The compressed file will appear.

Test result

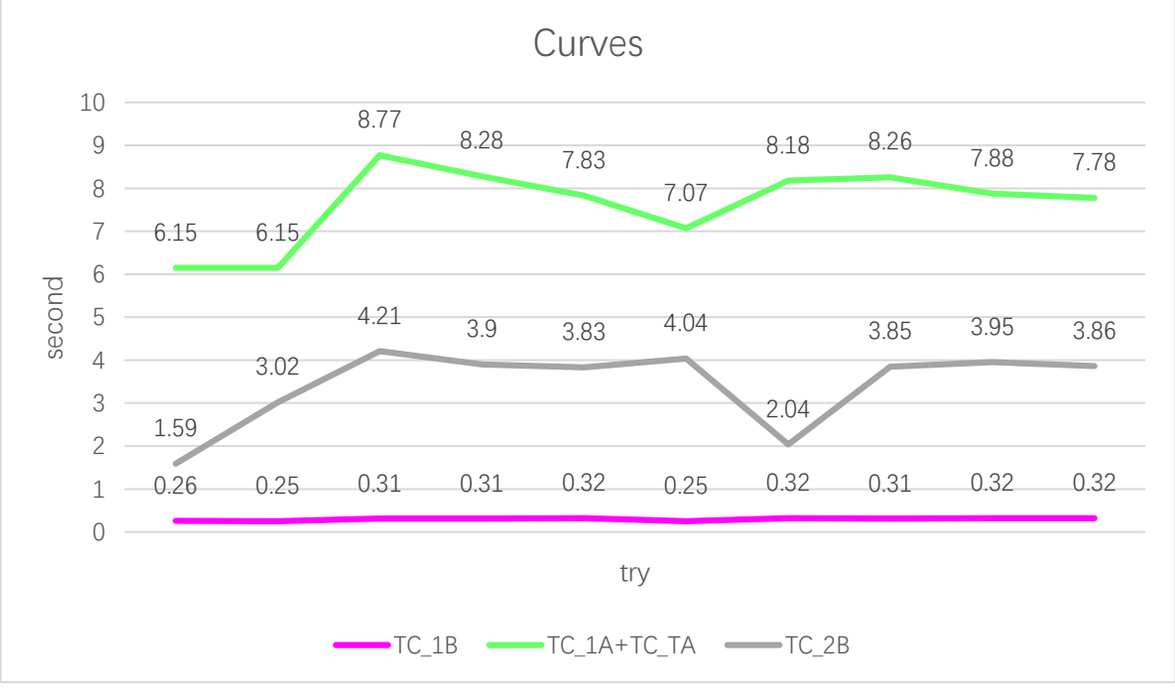
After the teacher release the test environment. The result is below.

```
Start receiving  folders.zipReceived  3 folders 1638177403.5866733 0 27500  from  192.168.56.105

Send  (2, 'file2.ppt', 0.0, 0, 0)  to  192.168.56.106
Start compressing  folders
folders  compressed
Finish sending
Received  2 file2.ppt 0.0 0 0  from  192.168.56.105
Writing finished
folders.zip  received
Start decompressing  folders
folders  decompressed
Not exist, create new file
Send  (4, 'folders', 1638177403.5866733, 0, 0)  to  192.168.56.106
Received  4 folders 1638177403.5866733 0 0  from  192.168.56.105
MD5_2A: PASS
MD5_FA: PASS
New update for  file2.ppt
Send  (1, 'file2.ppt', 1638177413.3033407, 0, 0)  to  192.168.56.106
Received  1 file2.ppt 1638177413.3033407 0 0  from  192.168.56.105
Send  (0, 'file2.ppt', 1638177413.3033407, 0, 27500)  to  192.168.56.105
Start receiving  share/file2.ppt
Received  0 file2.ppt 1638177413.3033407 0 27500  from  192.168.56.106
Writing finished
share/file2.ppt  received
Finish sending
MD5_2B: PASS
Result: {'RUN_A': True, 'RUN_B': True, 'MD5_1B': True, 'TC_1B': 0.24917912483215332, 'MD5_2A': True, 'TC_2A+TC_FA': 6.138761043548584, 'MD5_FA': True, 'MD5_2B': 1, 'TC_2B': 2.9854

Process finished with exit code 0
```

Curves



Conclusion

In this coursework, I learn many things by myself. In network programming, I learn how to design a protocol and many python method because this project needs many after-class knowledge. I also learn how to complete a big work gradually. Finish the main task and do high requirement work step by step. It is difficult to work when you want to satisfy all the requirements in the beginning. I also know the importance of self-study. Without it, the project is impossible to finish. In the future, I will keep self-study and

stage-work as my habits.

Reference

[1] Caraway and B. Robot, “Survey of File-Sharing Culture.” *International Journal of Communication (Online),* pp. 564, Apr2012

[2] J.Ning, “Python learning(26)-Compressing and decompressing files” CSDN.com. https://blog.csdn.net/linda1000/article/details/10432133 (accessed Nov. 29, 2021)

[3]Q.hu, “Python network programming, TCP/IP, Socket, C/S and so on” CSDN.com. https://blog.csdn.net/qwe5810658/article/details/79440389(accessed Nov. 29, 2021)