

Framework

The python version is python3.

Server

Create a multithreaded class to receive TCP transfer message from multiple clients. The class contains a main function for receiving and judging the content of the received information, and several auxiliary functions for implementing functions such as operations and maintenance logs.

Client

Set up a loop to continuously receive the instructions sent by the client, and process the received instructions in the format and send them to the server. Most of the client's instructions are transmitted through TCP, but the transmission of UVF instructions requires UDP.

Function

Login

When the client requests to connect to the server, it needs to enter 'login' at first. After the input is completed, the server prompts the client to enter the device name and password, and authenticates the user device. The authentication information is judged according to the credential.txt file on the server side. I set number of locks is 3, and the user cannot login for the next 10 seconds after the user inputs errors more than 3 times. After successful login, a welcome message will be displayed, and the user information and login timestamp will be added to the log file of the edge device in the server.

EDG

The input parameters of the EDG command include fileID and dataAmount, where fileID is an integer that uniquely identifies the file that generates data, and the dataAmount parameter is used to indicate the number of randomly generated data samples. If any of the two parameters is of the wrong type or is missing any parameter, a corresponding error will be

prompted and the data will not be generated successfully. After the data is successfully generated, it will be stored in the client, and a prompt will be given.

UED

My UED operation consists of two steps. The first is to send the fileID to the server. fileID should be an integer. If fileID is not an integer or has not been entered, an input error will be prompted. After successful input, the server returns the user name of the edge device. After that, the file which need to be uploaded on the client is searched in the client file according to the user name and fileID. At this time, it will search whether the file exists locally. If it does not exist, it will prompt that the file does not exist, and the UED request has an error. If it exists, the content in the file will be send to the server together with the user name and fileID as message. After the server reads the information, a file with the same name and the same content is generated on the server according to the message. After the upload is completed, the completed content is displayed, and a log file named uploadlog.txt is maintained on the server.

SCS

The SCS command contains two arguments: fileID and operation. fileID should be an integer. If the fileID is not an integer or the operation does not satisfied the set requirements or the two parameters are not fully input, an input error will be prompted. After successful input, the server returns message, which includes the operation, data source of the operation and the result of the operation. Client will output the result that from the message at last.

DTE

The fileID is the input parameter of the EDG command, where fileID is an integer. If the parameter is of the wrong type or missing parameters, a corresponding error will be prompted, and the data will not be deleted successfully. After the command is successfully entered, the server judges whether the file has been uploaded to the server before based on the fileID and user name. If it has not been uploaded to the server, it will prompt that the file does not exist. If the file exists, server read the file at first to get the file length, which need to be record in the log file. Then, the deleted file information is added to the deleted file log: deletion_log.txt. At last, server delete the file and send a message to the client to prove the file has been deleted.

AED

After receiving the AED command, the client sends the command to the server. After receiving

the command, the server accesses the log file of the edge device, reads the file line by line after excluding the user's information, and transmits the log to the client as message. After the client receives the message, it print it on the terminal.

OUT

After the client receives the OUT command, it sends the command to the server and exits the terminal. After the server receives the command, the active number in the record will be reduced by one and the online record of the user will be removed. Then server will update the edge device log file.

UVF

devicename and filename should be included in the input as parameters. Firstly, judge whether the input parameters are complete. If less parameters are input, it will prompt that the parameters are not completed, and the UVF file transfer task ends. After that, it will check whether the input filename file exists. If it does not exist, it will prompt that the file does not exist, and the UVF file transfer task ends. Afterwards, the receiver is turned on listening to prepare to receive the file, and then the sender reads the file in binary mode and sends it to the receiver in the form of message. The receiver receives the transmitted binary file and then generates the same content locally on the receiver. After the reception is completed, a prompt indicating that the download has been completed will be made. If a prompt is received, it proves that the transmission has been completed. The UVF transmission task is completed by multi-threading, and the receiver keeps listening and waiting for the sender to send. In order to deal with the situation of transferring a large file, I set the multi-threading to end 15 seconds after the file transfer starts, so as to ensure that the transfer will not fail because the file is too large.