
Tcp/udp socket forum



Introduction



First, I would like to introduce the results of my homework. I did this assignment using Python3. According to the homework manual, we need to create a multi-threaded online discussion forum. The server and client should communicate using TCP/UDP.

I need to implement the corresponding functions according to the requirements, and can upload and download files.

requirements

Language: python3.8

Libraries that come with python:

socket

threading

sys, select

os

collections

Run steps:(for example)

```
python3 TCP_UDP_Server3.py 10005
```

```
python3 TCP_UDP_Client3.py 127.0.0.1 10005
```

Why I use this tech-stack

Because I am more familiar with python, maybe I like it better, because it is easy to learn: Python language is a relatively easy to learn programming language compared to other programming languages, it focuses on how to solve problems rather than programming language syntax and structure. Graceful: The Python language strives to be concise and beautiful in code. In the Python language, indentation is used to identify code blocks, and the readability of the code is significantly improved by reducing useless braces and removing visual noise such as semicolons at the end of statements. Reading a good Python program feels like reading English, and it allows you to focus on solving the problem without getting too caught up in the syntax of the programming language itself. Rich and powerful library: The Python language is called Battery Included. The Python language class library is very comprehensive, including class libraries to solve various problems. No matter what functionality is implemented, there are ready-made class libraries that can be used. If a function is special and the standard library does not provide corresponding support, then there is a high probability that a corresponding open source project will provide similar functions. Reasonable use of Python class libraries and open source projects can quickly implement functions and meet business needs.

Project design consideration

I am basically passing commands from client to server. Then make a judgment in the Server. In terms of time, the implementation of DLT and EDT is laborious. For MSG, I created an auxiliary related function and structure using a while loop. The first string of each line is a number, the message number plus 1 step. For DLT and EDT, I think the difficulty lies in matching the CRUD of the corresponding message number.

Function realization:

Most of the features on the list are implemented(CRT, MSG, DLT, EDT, LST, RDT, UPD, DWN, RMV, XIT), except 2 features:

Successful authentication of multiple concurrent existing and new users including all error handing. This will take more time to tune up, so I'll put it on hold for now.

Implementation of mechanisms to recover from occasional loss of 1 1 UDP segments , I am not very familiar with UDP segments, I think I may not done it this time, I 'll put it on hold for now.