COMP 9331 Assignment

student ID : z5344573

Author : Tianyi Hou

This assignment is developed based on Python3.

code directory:
Server_UDP.py
Client_UDP.py
Credentials.txt

Program design ideas:

For the convenience of testing the code:
UPD: The name of the successfully uploaded file will be changed to **username_threadtitle_filename**
DWN: The successfully downloaded filename will become **download_filename**

1. Server_UDP:

When the server starts, a queue for storing messages, a buffer for storing user information, a user manager, a thread manager and a file transfer manager are generated, all of which are implemented by the dictionary type in Python.

Server message format

```
def packet(username, command, code, password='', step=0, message='', message_dict=None, thread_title='', file_name=''):
    if message_dict is None:
        message_dict = {}
    packet_message = {
        "username": username,
        "password": password,
        "step": step,
        "command": command,
        "code": code,
        "message": message,
        "thread_title": thread_title,
        "file_name": file_name,
        "message_dict":message_dict
    }
    return json.dumps(packet_message).encode()
```

1. Login:

   Username Verification Status : After the server receives the login request: the message contains username, step, and command,where command=“login”, step=0 (step=0 indicates that the user is currently in the login state of authentication).

   According to the username in the message, check whether the current user exists in the server database. If it exists in the database, check the login status of the current user.In the end, there will be two receipts.

   1. User is logged in. The user is logged in, which is represented by

code=2 in the message.Back to username verification state

2. It is a new user .The user is a new user, it is represented by code=1 in the message.Back to username verification state
3. Username verification succeeded and set that user status to logged in.Successful username verification is indicated by code=0 in the message.

Password verification status:The received message contains username, password, step=0, command="login".The server will compare the password of the corresponding user name and will return two kinds of receipts.

1. wrong password .Back to username verification state
2. password is correct.Enter the state of waiting for input command (step=1)

2. Command=CRT

The server receives the message (username, command=CRT, step=1, thread_title).The server checks whether the thread exists in the thread manager according to thread_title, and returns two results.

1. Thread already exists.It is displayed in the message with code=1 and returns to the state of waiting for command input
2. Thread created successfully and the thread is stored in the dictionary of the thread manager.It is displayed in the message with code=1 and returns to the state of waiting for command input

3. Command=MSG

The server receives the message (username, command=MSG, step=1, thread_title).The server checks whether the thread exists in the thread manager according to thread_title, and returns two results.

1. Thread does not exist .It is displayed in the message with code=1 and returns to the state of waiting for command input
2. Message added successfully.It is displayed in the message with code=0 and returns to the state of waiting for command input

4. Command=DLT

The server receives the message (username, command=DLT, step=1, thread_title,index).The server checks whether the thread exists in the thread manager according to thread_title and Check whether the message exists in the thread according to the index , and returns four results.

1. Thread does not exist .It is displayed in the message with code=2 and returns to the state of waiting for command input
2. Message is not present in this thread.It is displayed in the message with code=2 and returns to the state of waiting for command input

3. This message belongs to another user and has no right to modify it.It is displayed in the message with code=3and returns to the state of waiting for command input
4. Message deleted successfully.It is displayed in the message with code=0 and returns to the state of waiting for command input

5. Command=EDT

The server receives the message (username, command=EDT, step=1, thread_title,index,message).The server checks whether the thread exists in the thread manager according to thread_title and Check whether the message exists in the thread according to the index , and returns four results.

1. Thread does not exist .It is displayed in the message with code=2 and returns to the state of waiting for command input
2. Message is not present in this thread.It is displayed in the message with code=2 and returns to the state of waiting for command input
3. This message belongs to another user and has no right to modify it.It is displayed in the message with code=3and returns to the state of waiting for command input
4. Message edit successfully.It is displayed in the message with code=0 and returns to the state of waiting for command input

6. Command==LST

The server receives the message (username, command=LST, step=1).Service checks if thread exists in thread manager , and returns 2 results.

1. No thread exists.It is displayed in the message with code=1 and returns to the state of waiting for command input
2. Returns the dictionary holding the thread.It is displayed in the message with code=0 and returns to the state of waiting for command input

7. Command=RDT

The server receives the message (username, command=RDT, step=1,thread_title).The server checks whether the thread exists in the thread manager according to thread_title and Check whether the message exists in the thread according to the index , and returns 3 results.

1. No thread exists.It is displayed in the message with code=1 and returns to the state of waiting for command input
2. Store the message in the thread into a dictionary, return the dictionary.It is displayed in the message with code=0 and returns to the state of waiting for command input
3. No message in thread.It is displayed in the message with code=2 and returns to the state of waiting for command input

8. Command=UPD

The server receives the message (username, command=UPD, step=1,thread_title,file_name).The server checks whether the thread exists in the thread manager according to thread_title and Check whether the file exists in the thread according to the username+thread_title+file_name , and returns 3 results.

1. File already exists.It is displayed in the message with code=2 and returns to the state of waiting for command input
2. The thread dose not exist.It is displayed in the message with code=1 and returns to the state of waiting for command input
3. File upload successfully.It is displayed in the message with code=0 and returns to the state of waiting for command input

9. Command=DWN

The server receives the message (username, command=DWN, step=1,thread_title,file_name).The server checks whether the thread exists in the thread manager according to thread_title and Check whether the file exists in the thread according to the username+thread_title+file_name , and returns 3 results.

1. File does not exists.It is displayed in the message with code=2 and returns to the state of waiting for command input
2. The thread dose not exist.It is displayed in the message with code=1 and returns to the state of waiting for command input
3. File download successfully.It is displayed in the message with code=0 and returns to the state of waiting for command input

10. Command=RMV

The server receives the message (username, command=RMV, step=1, thread_title,index,message).The server checks whether the thread exists in the thread manager according to thread_title, and returns 3 results.

1. thread does not exist.It is displayed in the message with code=2 and returns to the state of waiting for command input
2. No permission to delete this thread.It is displayed in the message with code=1 and returns to the state of waiting for command input
3. Thread deleted successfully.It is displayed in the message with code=0 and returns to the state of waiting for command input

11. Command=XIT

The server receives the message (username, command=XIT, step=1).The server will close the user thread and set the user status to not logged in

2. Cient_UDP:

Client message format

```
def packet(username, command, password='', step=0, message='', thread_title='', file_name='', right_username=1,
                right_password=1, index='',have_file=0):
    packet_message = {
        "username": username,
        "password": password,
        "step": step,
        "command": command,
        "message": message,
        "thread_title": thread_title,
        "file_name": file_name,
        "index": index,
        "right_username": right_username,
        "right_password": right_password,
        "have_file":have_file
    }
    return json.dumps(packet_message).encode()
```

Client function:

1. Verify that username and password have no illegal characters and are not empty
2. Verify that the entered command line format is correct
   E.g EDT 9331
   E.g XIT yoda
3. Split the contents of the entered command line
   E.g EDT 9331 1 Networks is awesome =>
   (command=EDT,thread_title=9331,index=1,message=Networks is awesome)
4. Verify the code in the server's receipt message
   E. g if code=0 print("The message has been deleted")
   elif code=1 print("The message belongs to another user and cannot be edited")

Summarize

During the development process, I had a deeper understanding of UDP

communication and the interaction between the server and the client, and also improved my coding ability to a certain extent.