COMP3331 ASSIGNMENT

MINGLANG XIE z5228006

The goal of this assignment is to implement my own version of an online discussion forum application. The application is based on a client server model consisting of one server and multiple clients communicate either sequentially. The client and server are communicating using TCP.

Server.py:

Two parameters are required to start the server, server port and admin password. Server post is the port number which the serve will use to communicate with the clients, and admin password is required to shut down the server (when client use SHT command).

After the server is initialized and started a new file **credentials.txt**, which is containing the username and the password, will be created to store the login information of all user.

A new thread will be created for each client. The server then creates a Client object for each thread and store it in a global variable **clients**. Each client object contains a username, pwd (password), conn (server connection with the client), request (request send from the client), logged (whether the user have logged in), executeComm (different request type from the client).

After the client connected with the server, it will first send login information of the user to the server. Then, the server responses differently for different login status, as the table show below:

Client

Login status	Request
Send a username to the server	login-N
Send a successful login flag to the server	login-S
Send a password to the server	login-P
Send a new password to the server (as a sign up)	login-NP

Server

Received login status	Response
The username does not exist in credentials.txt (login-N)	NU
The username is valid, and the user have not login (login-N)	PWD
The username is valid, but the user logged in (login-N)	LD
The password is correct for the given username (login-P)	Y
The password is not correct (login-P)	N
The password is correct for new user	YN

Figure 1, login/registration stage

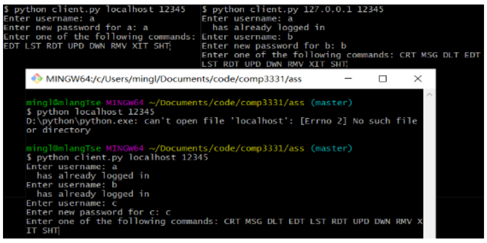
After the client send ‘login-S’ to the server, the server will output ‘successfully logged in’ for new user, and ‘successful login’ for the other user.

client.py:

Server IP address (the IP address of the machine on which the server is running) and Server port (the port number being used by the server) are needed when a client wants to communicate with the server.

The client first set up the TCP connection then process the login/registration stage (as shown in Figure 1). Then, the client will open two threads, one thread will handle the message sent from the server, and the other one thread will process the command entered by the user and send the command information to the server.

Testing functionality & Feature description



```
$ python client.py localhost 12345
Enter username: a
Enter new password for a: a
Enter one of the following commands:
EDT LST RDT UPD DWN RMV XIT SHT
```

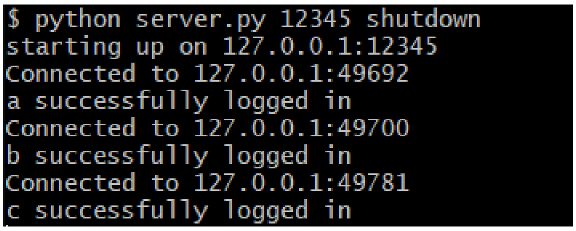
```
$ python client.py 127.0.0.1 12345
Enter username: a
  has already logged in
Enter username: b
Enter new password for b: b
Enter one of the following commands: CRT MSG DLT EDT
LST RDT UPD DWN RMV XIT SHT
```

MINGW64:/c/Users/mingl/Documents/code/comp3331/ass

```
mingl@mlangTse MINGW64 ~/Documents/code/comp3331/ass (master)
$ python localhost 12345
D:\python\python.exe: can't open file 'localhost': [Errno 2] No such file
or directory

mingl@mlangTse MINGW64 ~/Documents/code/comp3331/ass (master)
$ python client.py localhost 12345
Enter username: a
  has already logged in
Enter username: b
  has already logged in
Enter username: c
Enter new password for c: c
Enter one of the following commands: CRT MSG DLT EDT LST RDT UPD DWN RMV X
IT SHT
```

Multiple clients try to login



```
$ python server.py 12345 shutdown
starting up on 127.0.0.1:12345
Connected to 127.0.0.1:49692
a successfully logged in
Connected to 127.0.0.1:49700
b successfully logged in
Connected to 127.0.0.1:49781
c successfully logged in
```

Server message: multiple client login

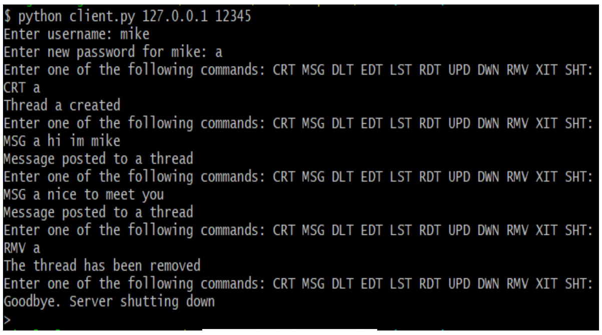
Command	Server’s response
CRT	• ‘Thread <thread name> exists’ • ‘Thread <thread name> created’ if success
MSG	• ‘Thread <thread name> not exists’ if the thread does not exist • ‘Message posted to <thread name> thread’ if success
DLT	• ‘Thread <thread name> not exists’ • ‘Invalid message number’ • ‘The message belongs to another user and cannot be delete’ • ‘Message posted to <thread name> thread’ if success
EDT	• ‘Thread <thread name> not exists’ • ‘Invalid message number’ • ‘The message belongs to another user and cannot be edited • ‘Message has been edited’ if success
LST	List of all existing threads
RDT	All messages in a specific thread
UPD	• ‘Thread <thread name> not exists’ • ‘<filename> already exist in Thread <thread name>’ • If success, receive the corresponding file from the client. Finally, response ‘<username> uploaded file <filename> to

	<thread name> thread’
DWN	● ‘Thread <thread name> not exists’ ● ‘File does not exist in Thread <thread name>’ ● If success, send the corresponding file to the client. Finally, response ‘<filename> successfully download’
RMV	● ‘Thread <thread name> not exists’ ● ‘The thread was created by another user and cannot be removed’ ● ‘The thread has been removed’ if success
XIT	● The client login out, response ‘Goodbye’
SHT	● ‘Incorrect password’ ● Send ‘Goodbye. Server shutting down\n>’ to all client, then close the server.

Problem

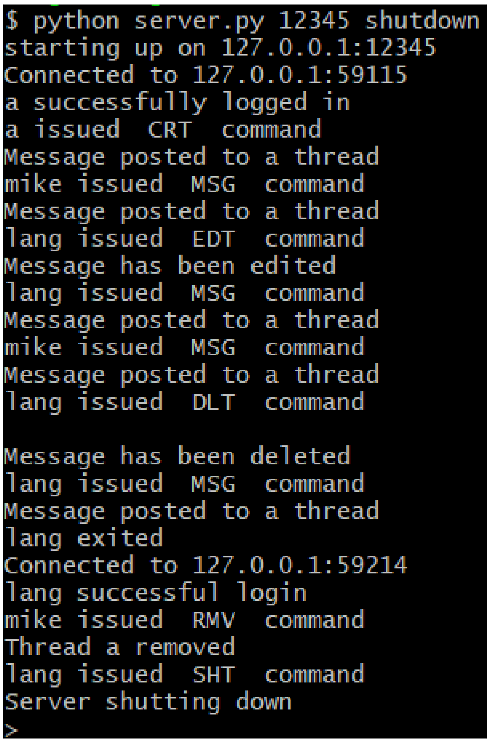
If the client uses ctrl + C to terminate the program, the client remains logged in on the server. Thus, the client cannot log in with the same username again.

Sample output:



```
$ python client.py 127.0.0.1 12345
Enter username: mike
Enter new password for mike: a
Enter one of the following commands: CRT MSG DLT EDT LST RDT UPD DWN RMV XIT SHT:
CRT a
Thread a created
Enter one of the following commands: CRT MSG DLT EDT LST RDT UPD DWN RMV XIT SHT:
MSG a hi im mike
Message posted to a thread
Enter one of the following commands: CRT MSG DLT EDT LST RDT UPD DWN RMV XIT SHT:
MSG a nice to meet you
Message posted to a thread
Enter one of the following commands: CRT MSG DLT EDT LST RDT UPD DWN RMV XIT SHT:
RMV a
The thread has been removed
Enter one of the following commands: CRT MSG DLT EDT LST RDT UPD DWN RMV XIT SHT:
Goodbye. Server shutting down
>
```

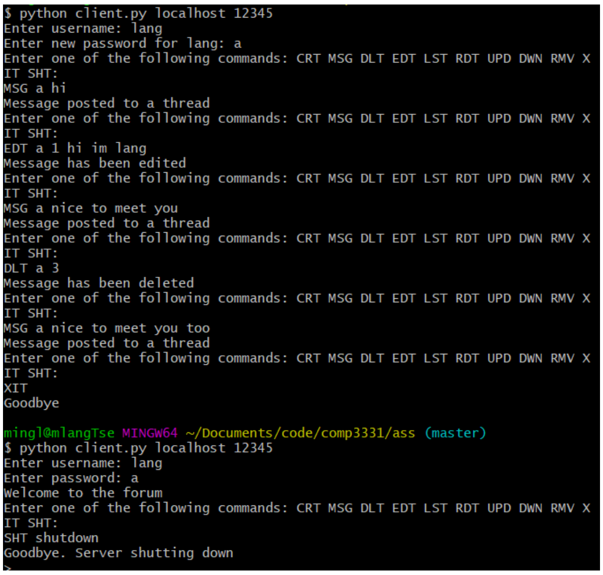
Client one



```
$ python server.py 12345 shutdown
starting up on 127.0.0.1:12345
Connected to 127.0.0.1:59115
a successfully logged in
a issued  CRT  command
Message posted to a thread
mike issued  MSG  command
Message posted to a thread
lang issued  EDT  command
Message has been edited
lang issued  MSG  command
Message posted to a thread
mike issued  MSG  command
Message posted to a thread
lang issued  DLT  command

Message has been deleted
lang issued  MSG  command
Message posted to a thread
lang exited
Connected to 127.0.0.1:59214
lang successful login
mike issued  RMV  command
Thread a removed
lang issued  SHT  command
Server shutting down
>
```

Server message



```
$ python client.py localhost 12345
Enter username: lang
Enter new password for lang: a
Enter one of the following commands: CRT MSG DLT EDT LST RDT UPD DWN RMV X
IT SHT:
MSG a hi
Message posted to a thread
Enter one of the following commands: CRT MSG DLT EDT LST RDT UPD DWN RMV X
IT SHT:
EDT a 1 hi im lang
Message has been edited
Enter one of the following commands: CRT MSG DLT EDT LST RDT UPD DWN RMV X
IT SHT:
MSG a nice to meet you
Message posted to a thread
Enter one of the following commands: CRT MSG DLT EDT LST RDT UPD DWN RMV X
IT SHT:
DLT a 3
Message has been deleted
Enter one of the following commands: CRT MSG DLT EDT LST RDT UPD DWN RMV X
IT SHT:
MSG a nice to meet you too
Message posted to a thread
Enter one of the following commands: CRT MSG DLT EDT LST RDT UPD DWN RMV X
IT SHT:
XIT
Goodbye

mingl@mlangTse MINGW64 ~/Documents/code/comp3331/ass (master)
$ python client.py localhost 12345
Enter username: lang
Enter password: a
Welcome to the forum
Enter one of the following commands: CRT MSG DLT EDT LST RDT UPD DWN RMV X
IT SHT:
SHT shutdown
Goodbye. Server shutting down
>
```

Client two