# Assignment report

1.Including in the tar.file

client.py

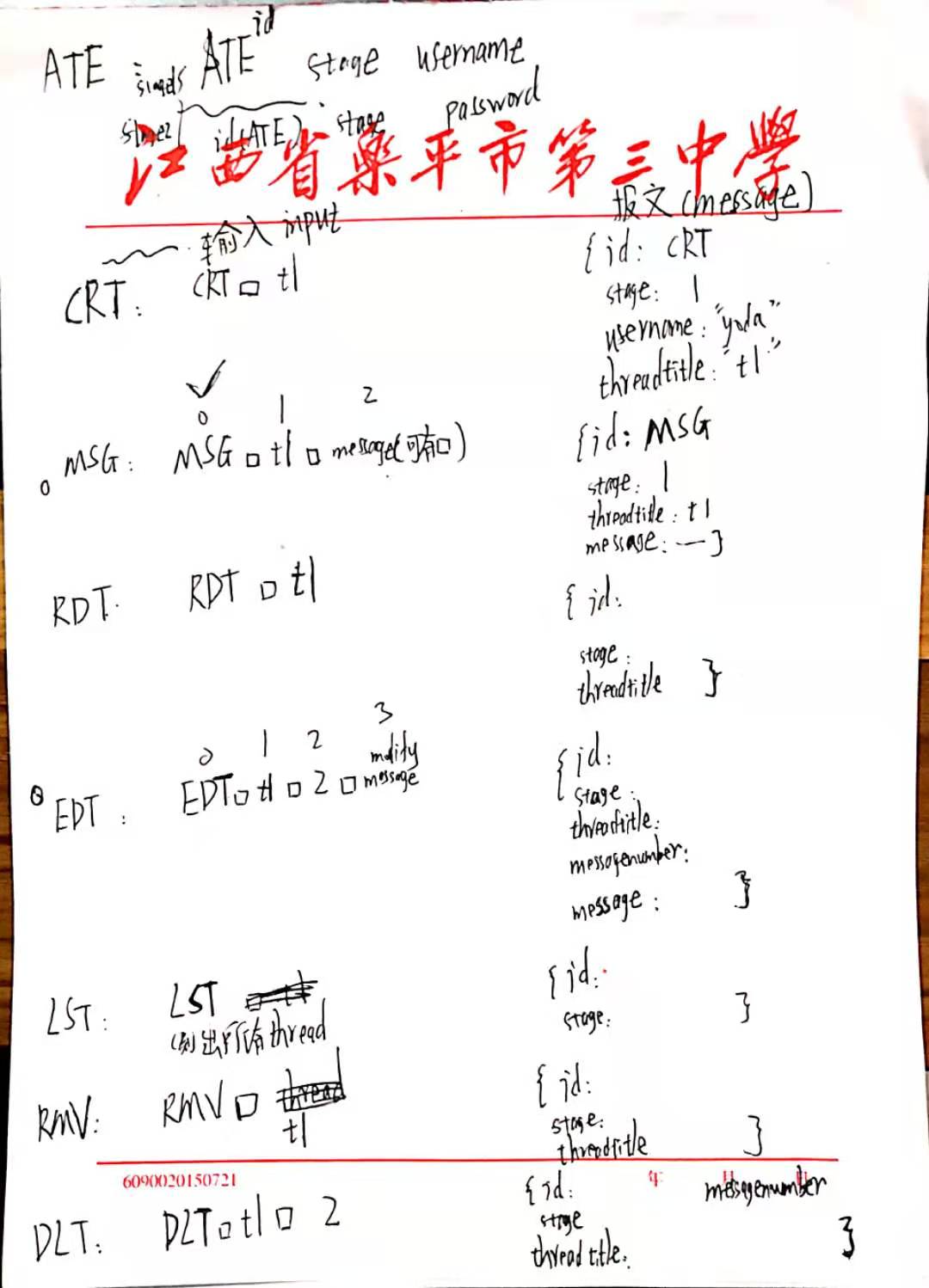
server.py

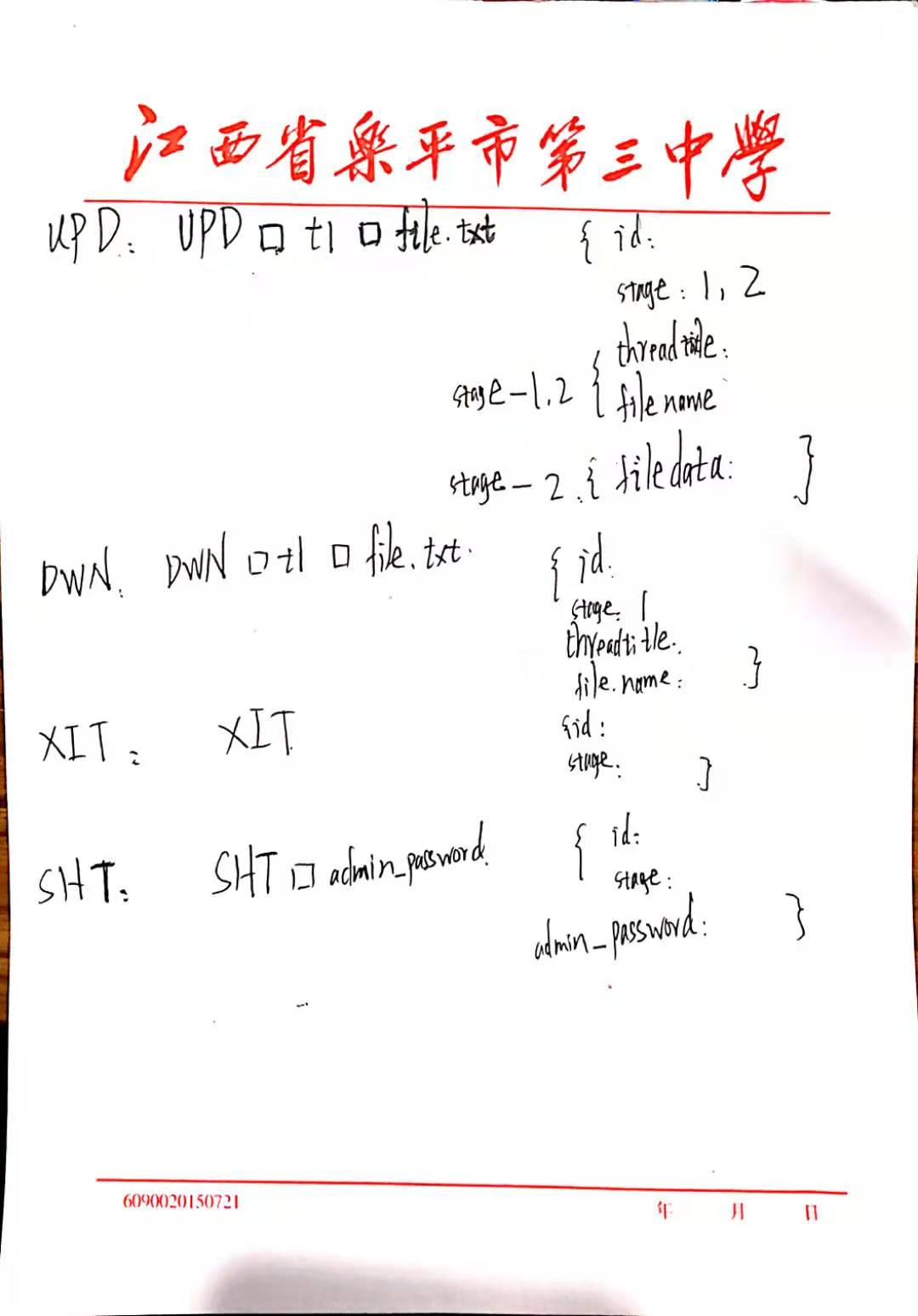
manager.py

report.pdf

2.My design thoughts

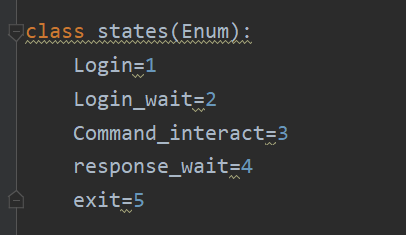
Firstly, according to the assignment description file, I recorded each command’s input format and application layer message format.





Then I use Object-oriented programming method, set class Server,Client,Message,User,File,UserManager,Thread,ThreadManager.

Then I set there exists five state in the whole sending and receiving loop:

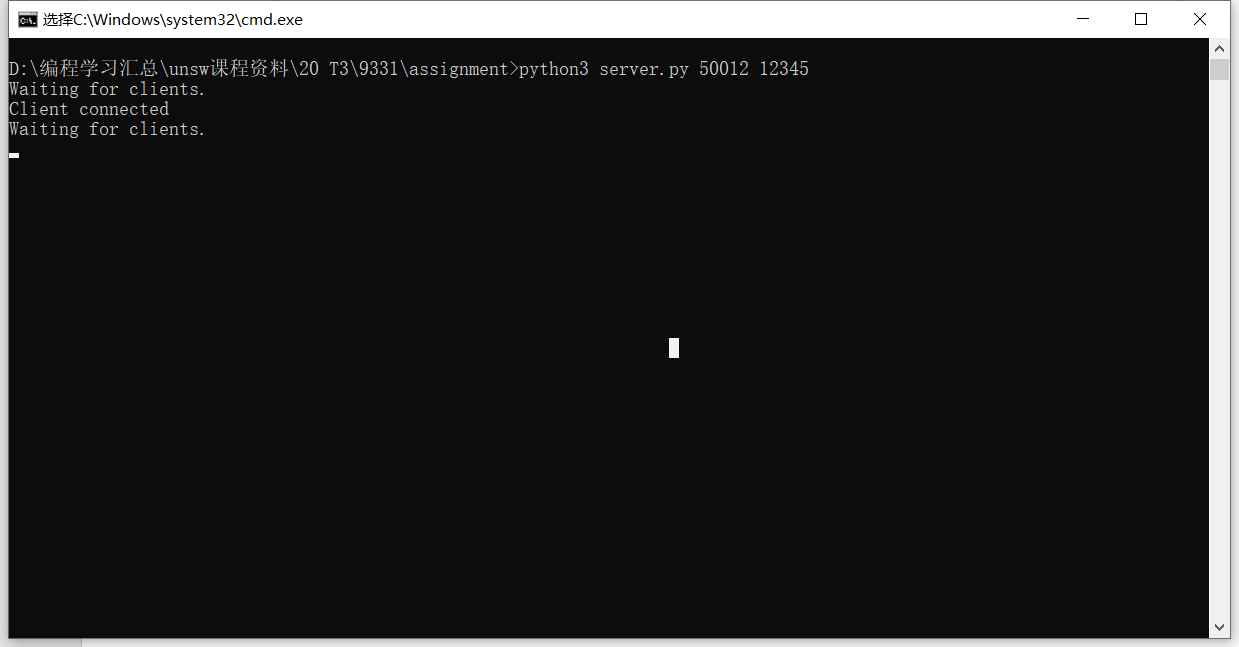


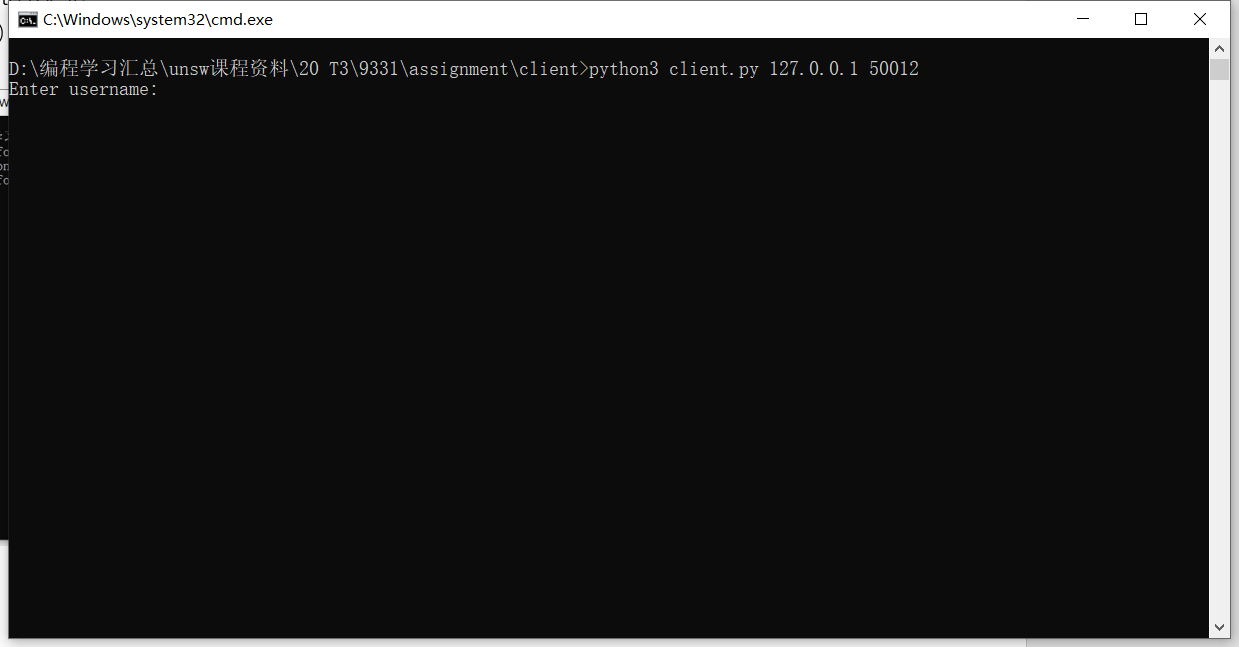
Login,Login\_wait,Command\_interact,response\_wait and exit to simulate the first and second stage which message exchanging between client and server.

Finally I use the function set in each .py file to implement every command’s function and got the effect I want.

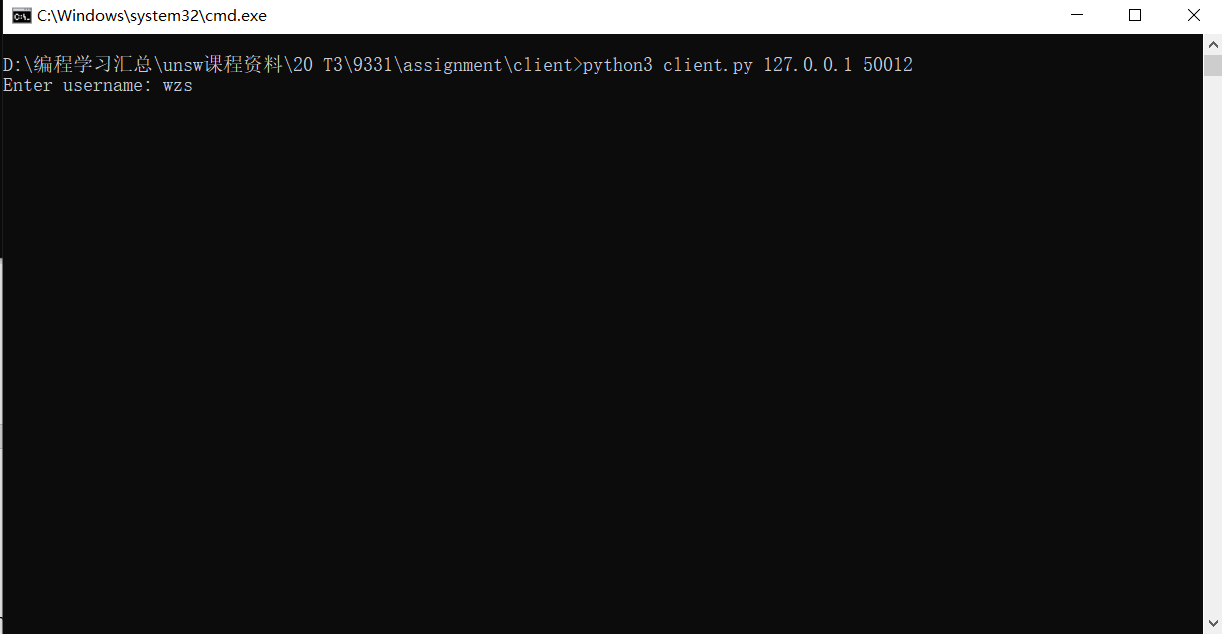
3.How to use it

(1) The client and server login interface

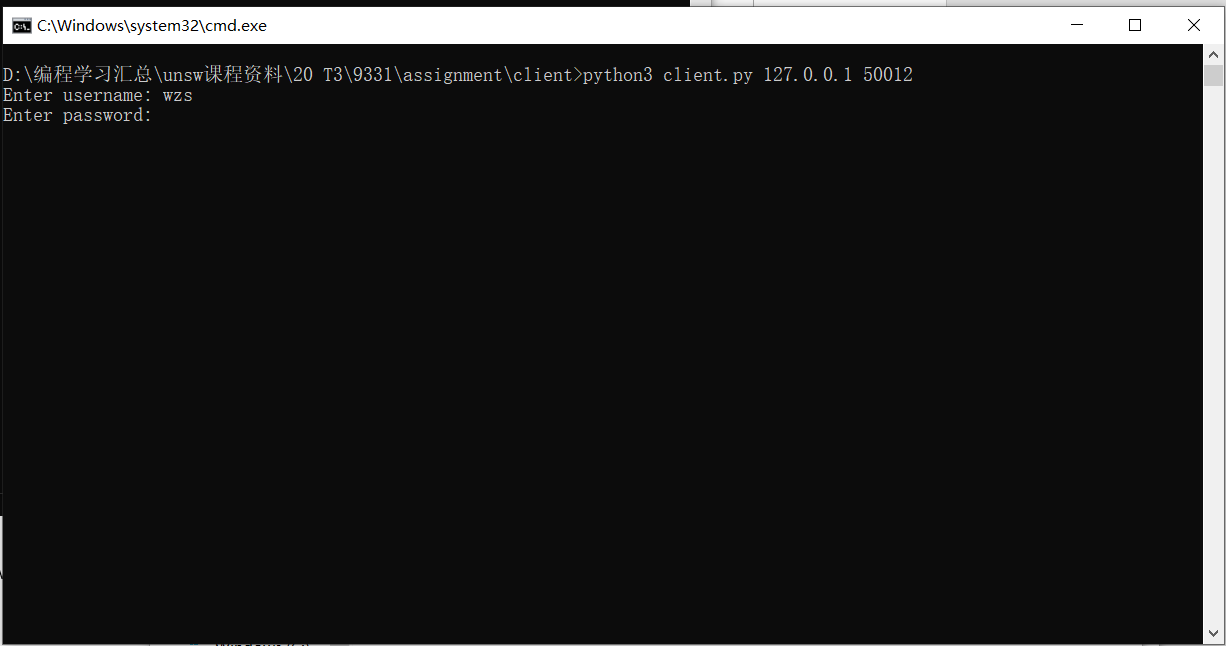




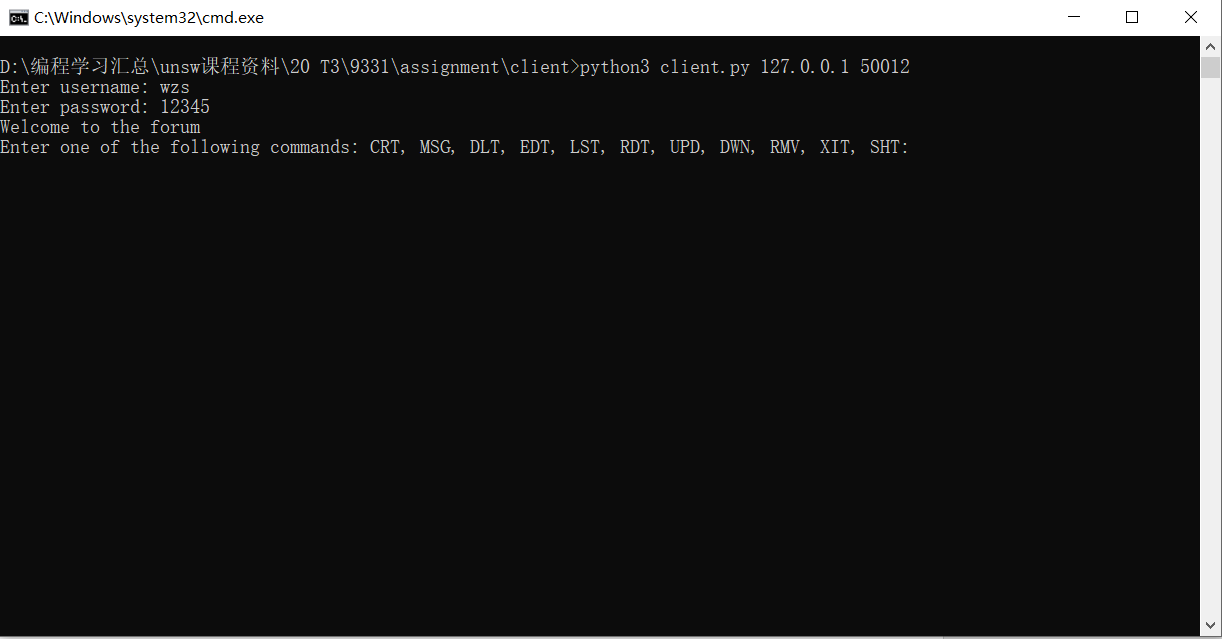
Then enter the username in the client (login stage 1):

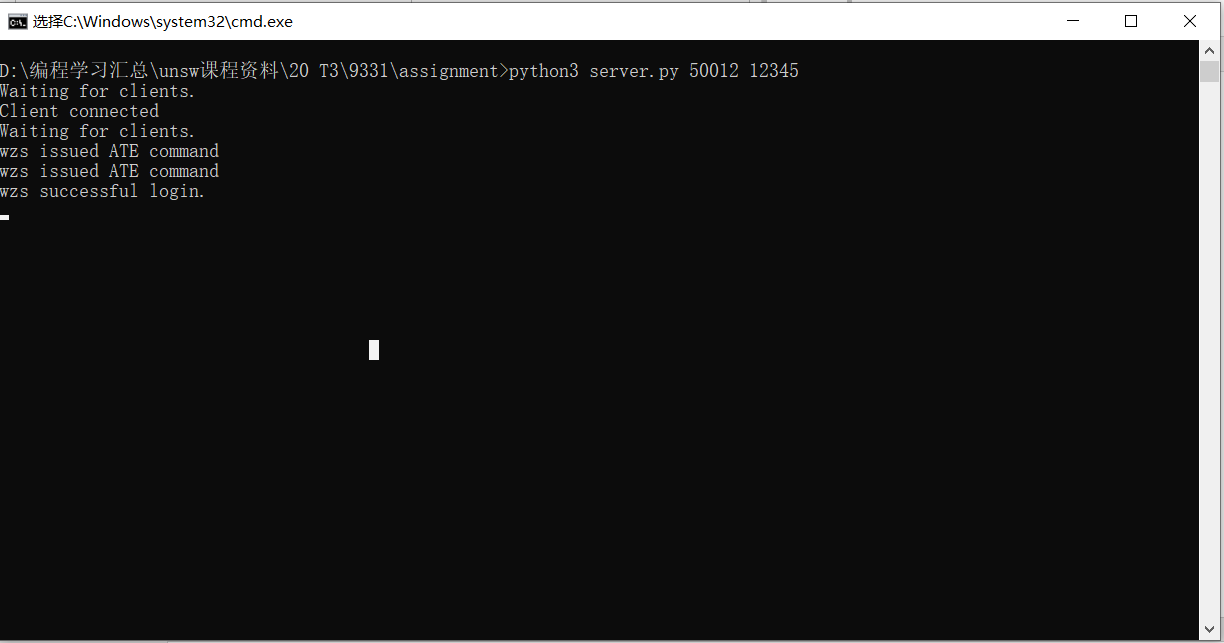


Enter the password (login stage 2)

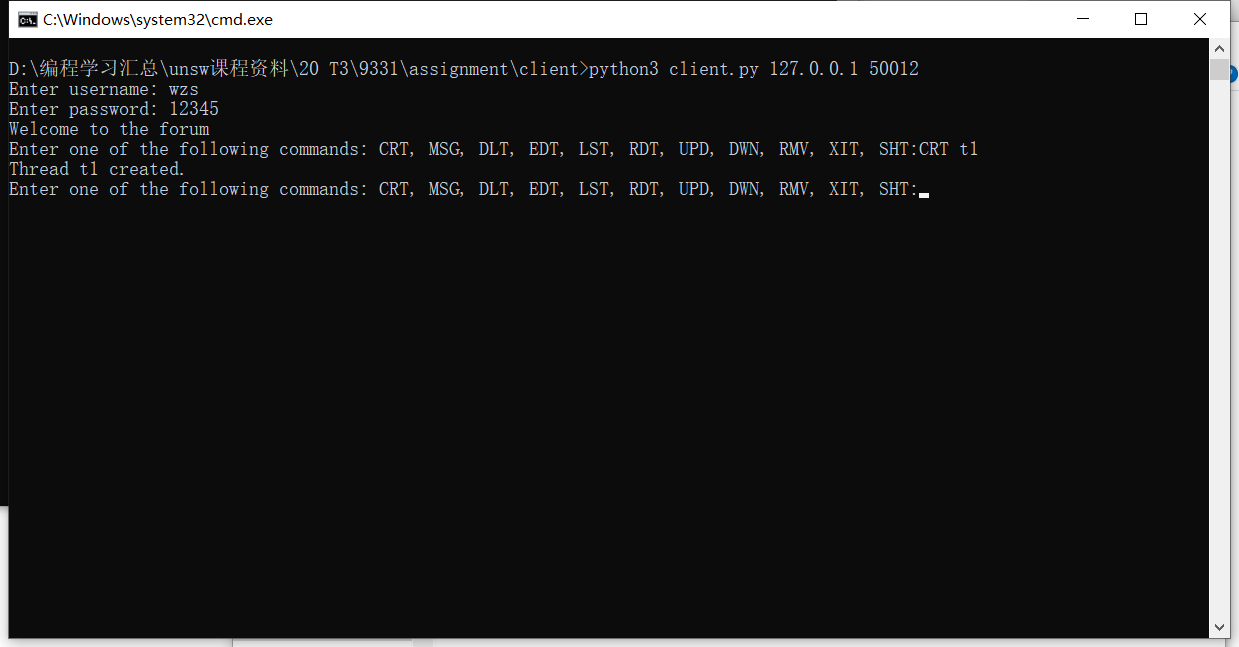


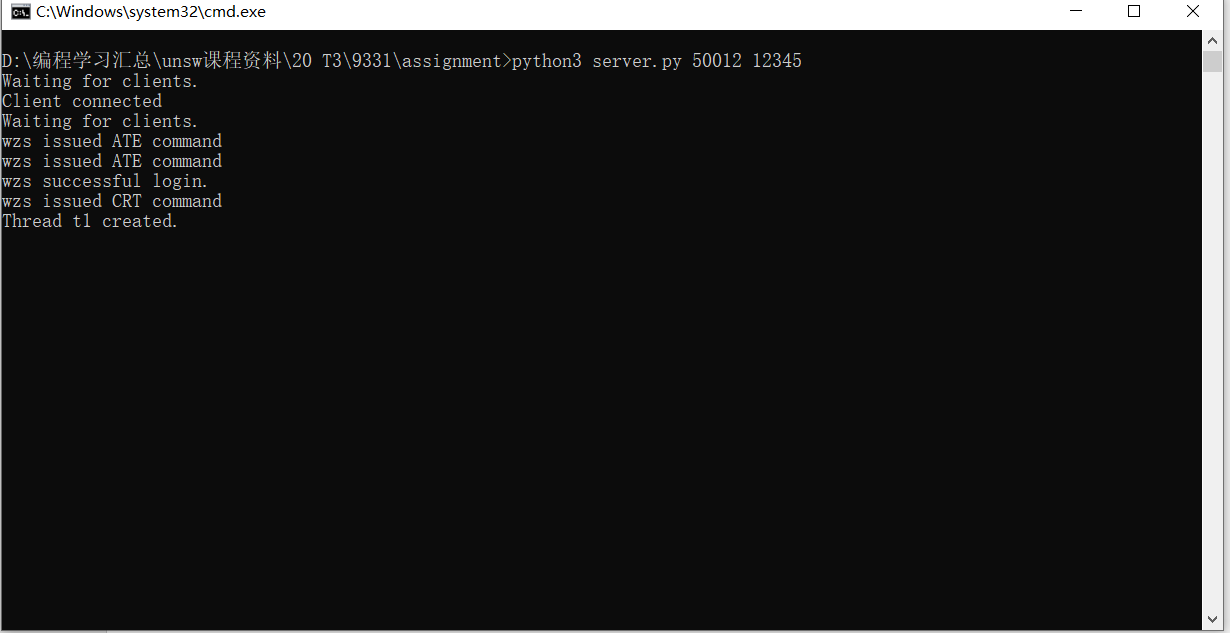
Finally login successfully：



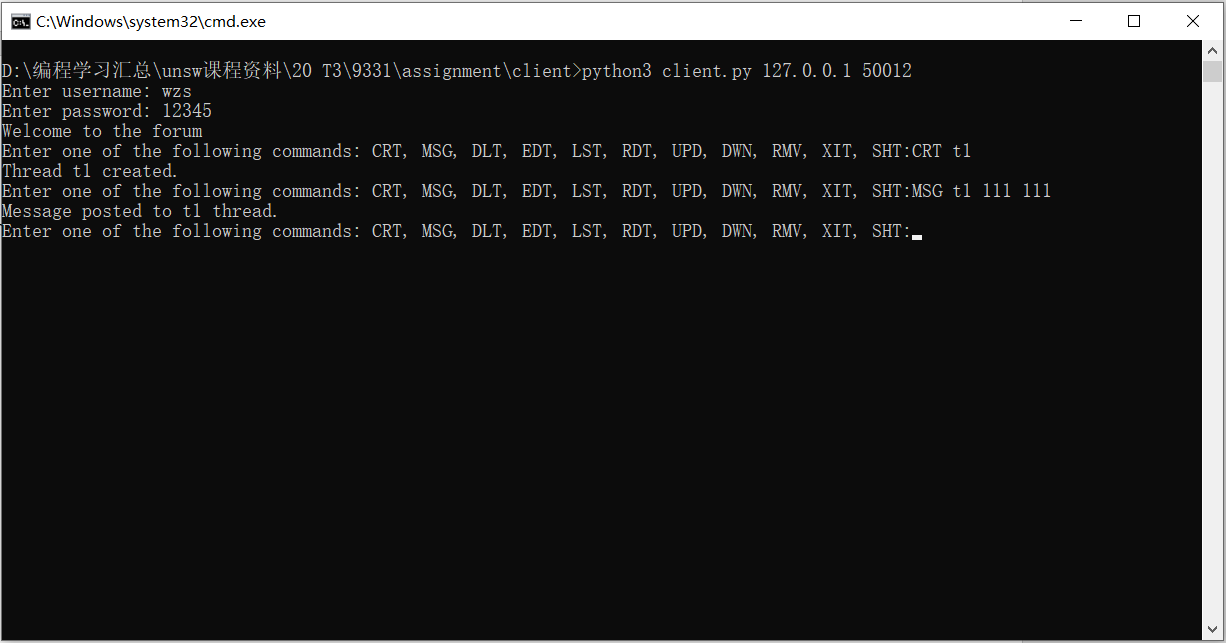


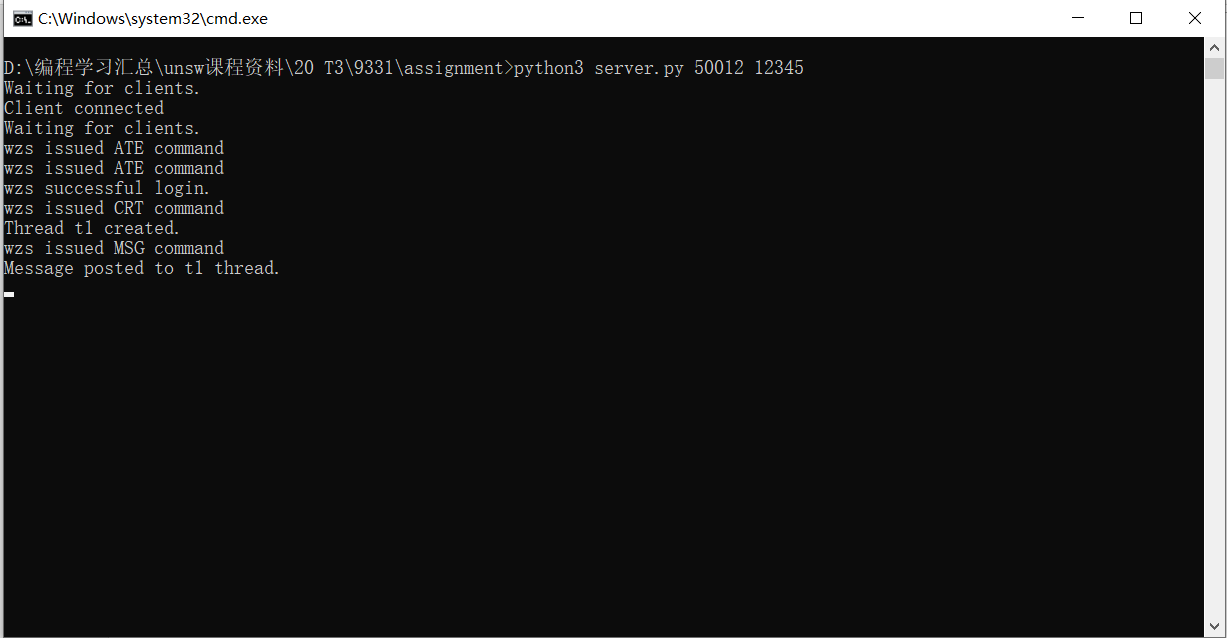
Enter the command CRT t1



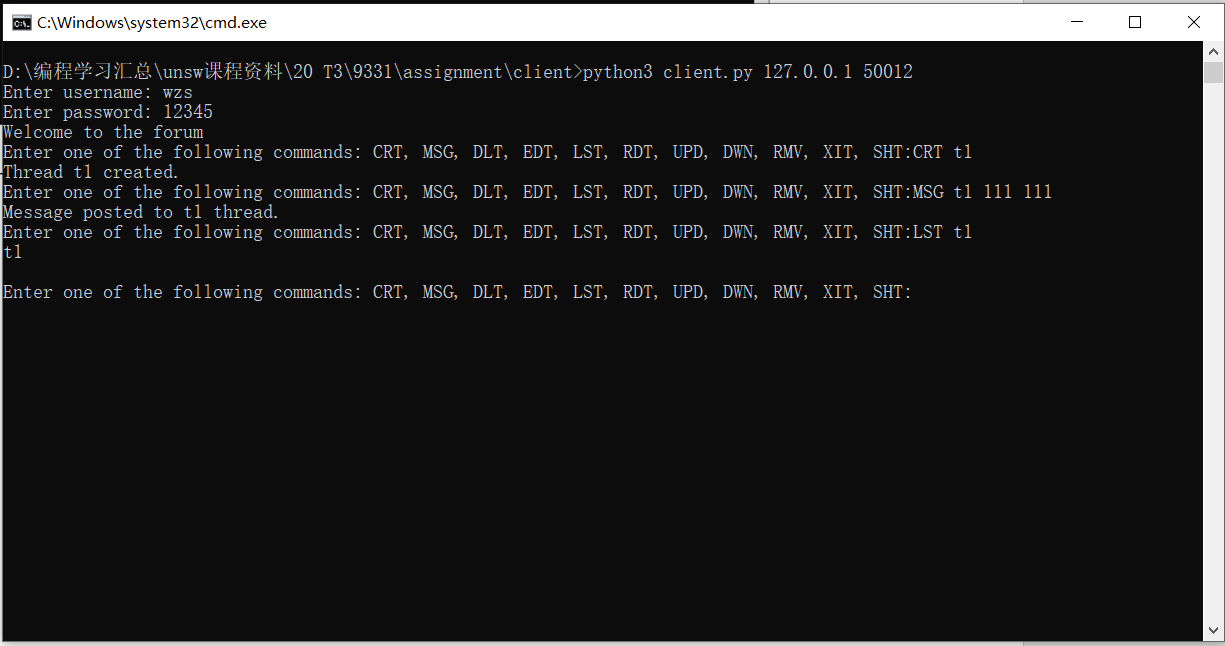


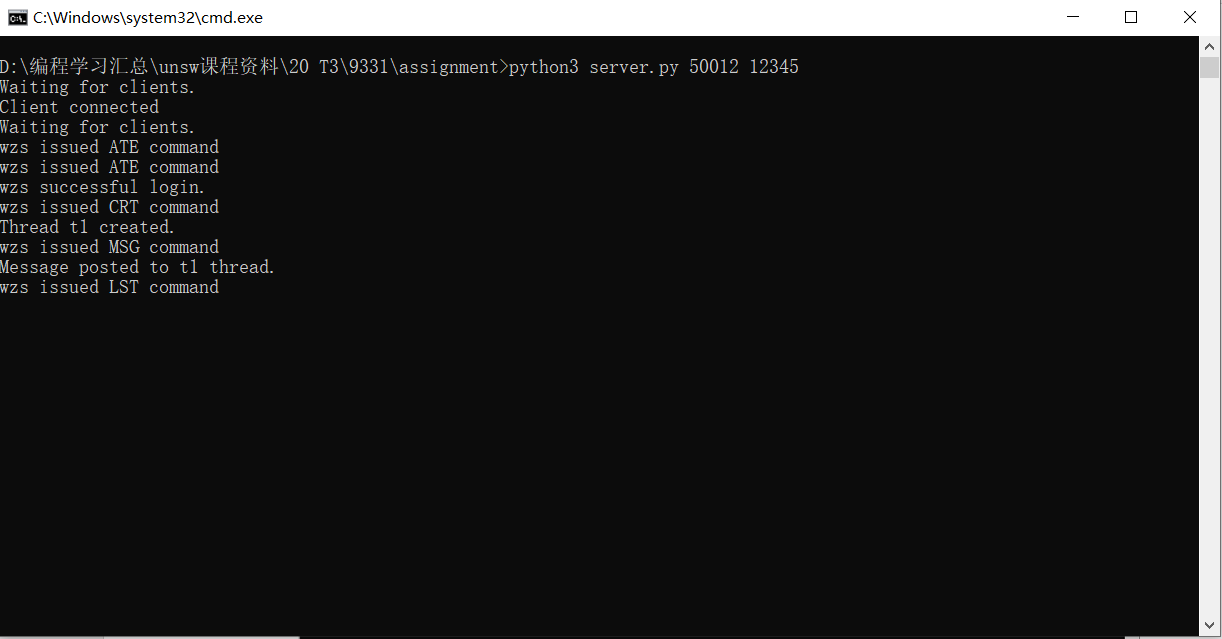
Enter the command MSG t1 111 111



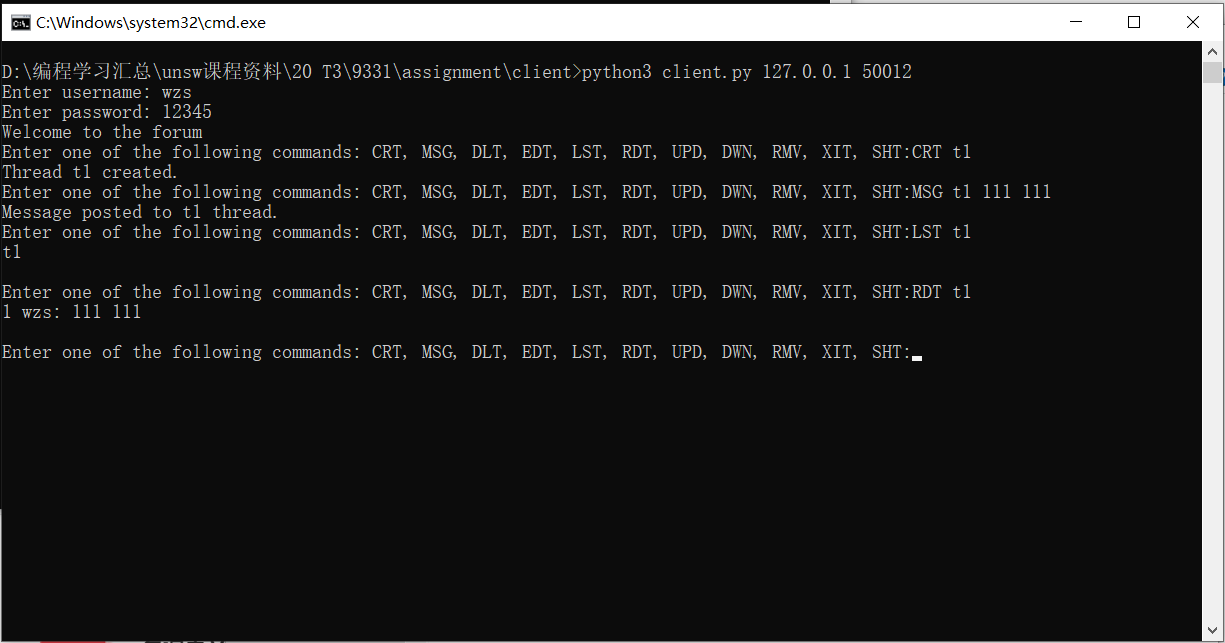


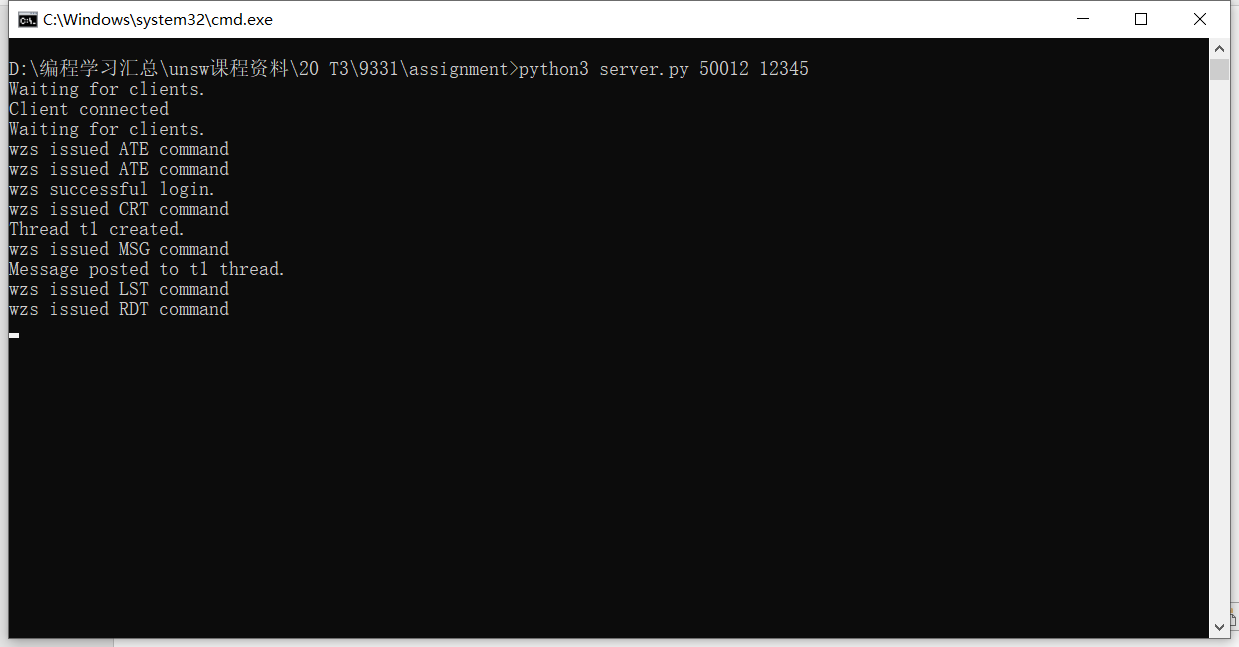
Enter the command LST t1：



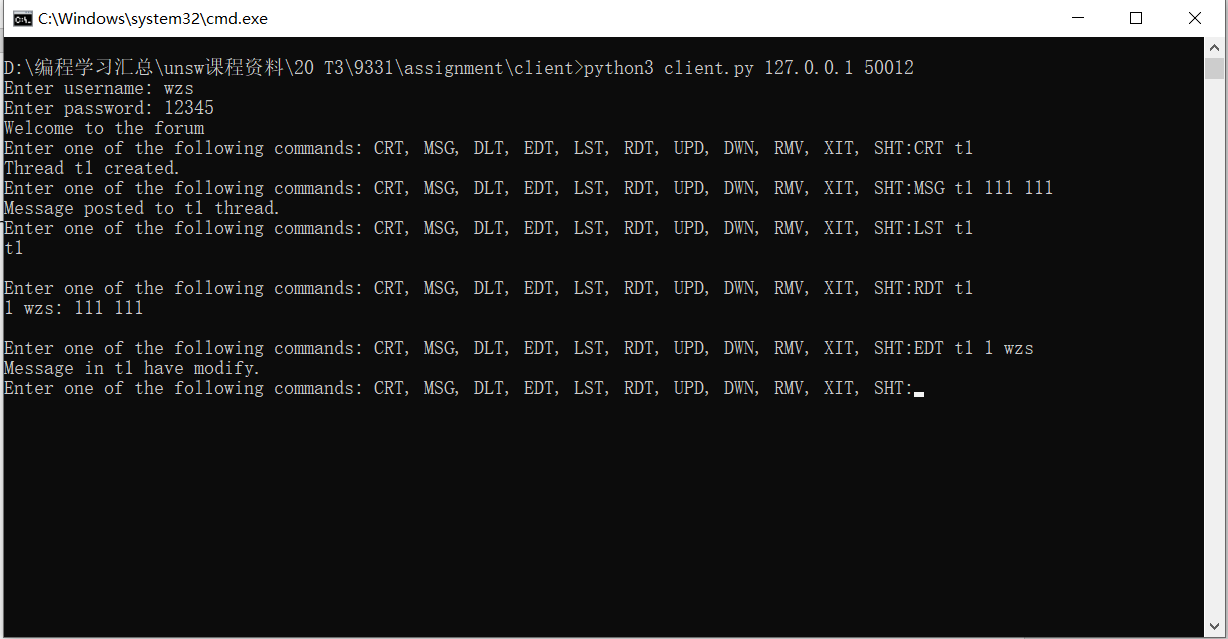


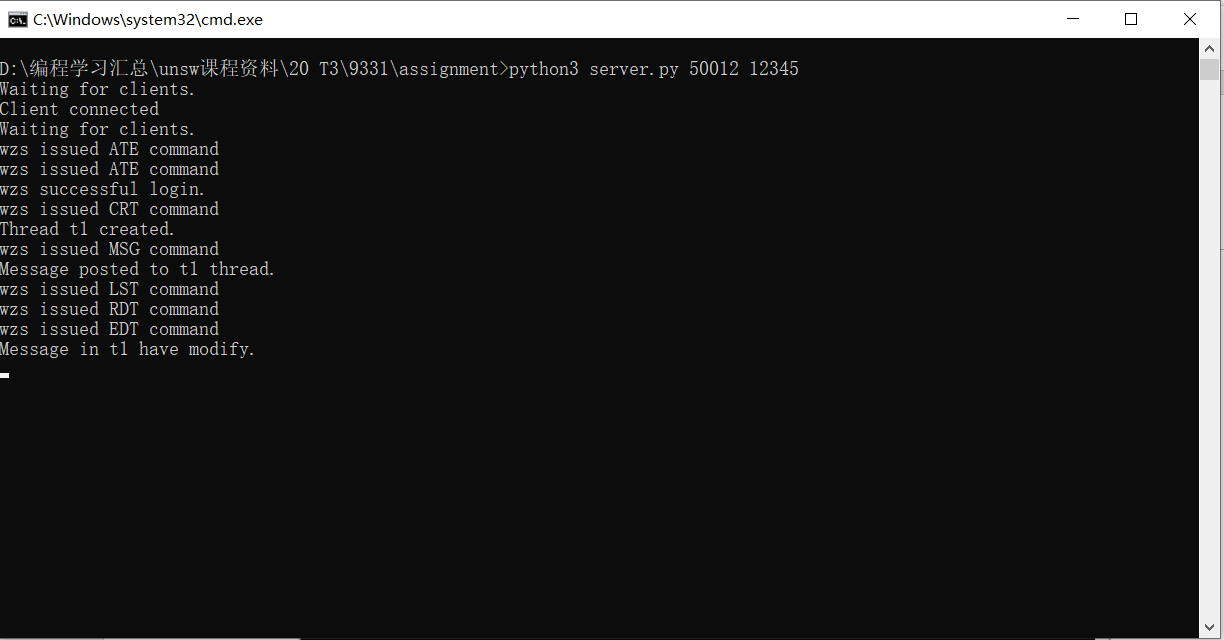
Enter the command RDT t1：



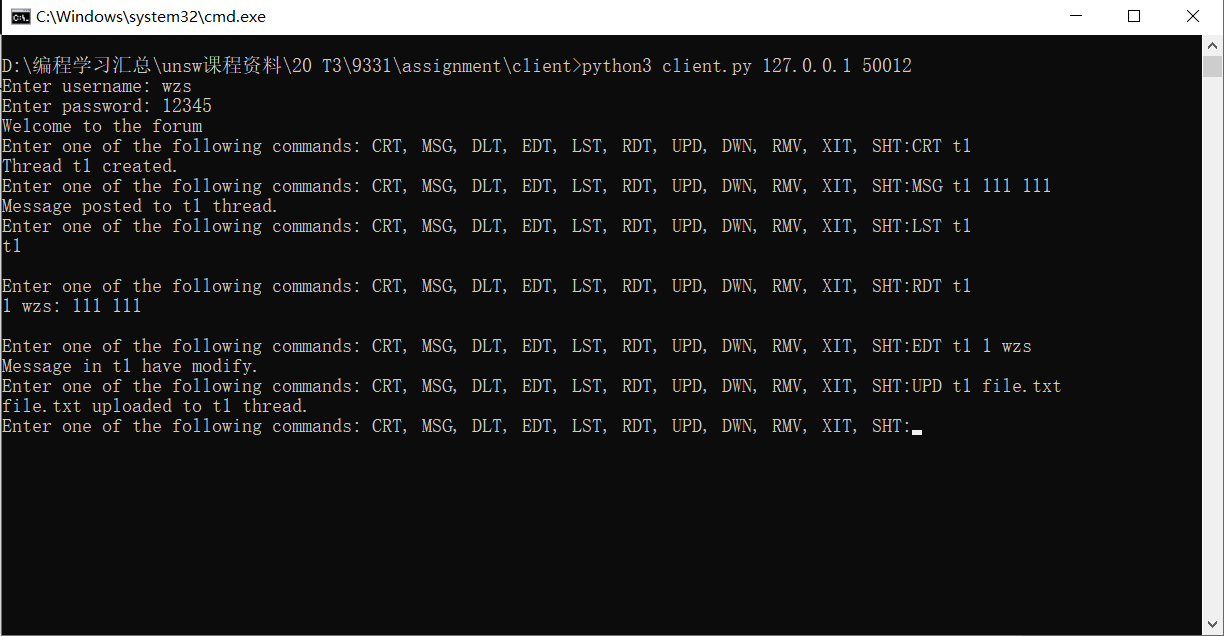


Enter the command EDT t1 1 wzs：

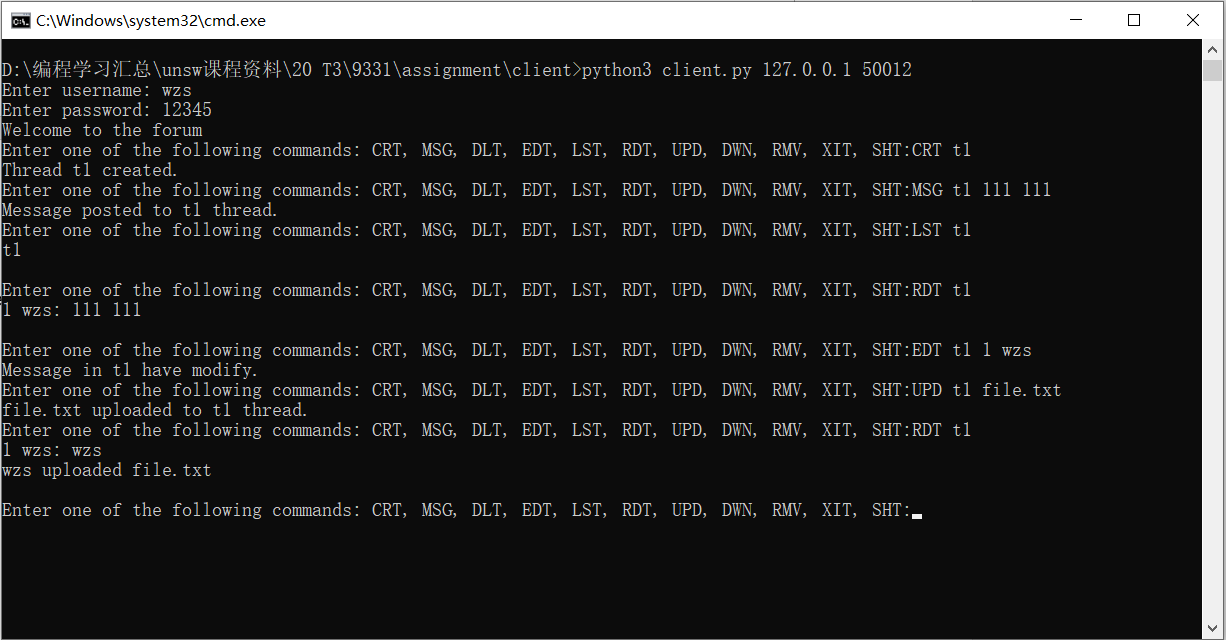




Enter the command UPD t1 file.txt:



Enter the command RDT t1:



Enter the command RMV t1:

