- 服务端

```python
from socket import *

class Server:
    def __init__(self):
        self.__udp_socket = socket(AF_INET, SOCK_DGRAM)
        self.__list_name = []
        self.__addr_list = []
        self.__udp_socket.bind(("0.0.0.0", 8888))  # 自动创建套接字

    def __login(self, data, addr):
        """
        进行登录验证
        :param data:
        :param addr:
        :return:
        """
        if data.decode() in self.__list_name:
            self.__udp_socket.sendto("失败".encode(), addr)
        else:
            self.__list_name.append(data.decode())
            self.__addr_list.append(addr)
            self.__udp_socket.sendto("1".encode(), addr)

    #  封装发送消息给所有人的函数
    def __sent_all(self, data):
        for addr in self.__addr_list:
            self.__udp_socket.sendto(data, addr)

    def main(self):
        while True:
            data, addr = self.__udp_socket.recvfrom(1024)
            if addr in self.__addr_list:
                self.__sent_all(data)
            else:
                self.__login(data, addr)


server = Server()
server.main()
```

- 客户端

```python
from socket import *
from multiprocessing import *
import sys


class Client:
    ADDR = ("127.0.0.1", 8888)

    def __init__(self):
        self.__udp_socket = socket(AF_INET, SOCK_DGRAM)
        self.name = ""

    def __login(self):
        while True:
            self.name = input("请输入昵称（不能重复）")
            self.__udp_socket.sendto(self.name.encode(), Client.ADDR)
            data, addr = self.__udp_socket.recvfrom(1024)
            if data.decode() == "失败":
                continue
            else:
                break
        self.__udp_socket.sendto(f"{self.name}进入聊天室".encode(), Client.ADDR)

    # 一直循环，回车控制退出
    def __send_message(self):
        while True:
            content = input(">>")
            msg = self.name + ": " + content
            if not content:
                self.__client_quit()
                break  # 使用回车退出
            self.__udp_socket.sendto(msg.encode(), Client.ADDR)

    def __client_quit(self):
        self.__udp_socket.sendto(f"{self.name}退出聊天室".encode(), Client.ADDR)
        self.__udp_socket.close()
        sys.exit()

    # 接受消息作为子进程
    # 一直循环，父进程退出才退出
    def __receive_message(self):
        while True:
            data, addr = self.__udp_socket.recvfrom(1024)
            print(data.decode())

    def main(self):
        self.__login()
        p = Process(target=self.__receive_message, daemon=True)
        p.start()
        self.__send_message()


client = Client()
client.main()
```