

BUG修复：多条消息合并问题

平时在简单对用户的命令做出响应时没有问题，但是当用户输入 `lst` 来列出所有的主题时却出现了比较严重的问题。

因为主题内容可能很长，而socket每次只能传输1k，所以我在上一版的程序里将主题一条一条的发过去，但这样出现造成连续发送的两条消息被合并，以致于主题显示不全（只能显示前两行）。

初步考虑的解决方案是在需要连续发送的消息前加上这个数据有多少字节，这样就可以一条一条的按照设置好的长度进行接收，既不会消息连在一起，也不会超过1k而发送失败。

修改后的代码：

server.py

```

# -*- coding: utf-8 -*-
import time
import User
import Topic
import socket
import threading

HOST = "127.0.0.1"
PORT = 23333

# global variable
is_normal = True
userlist = []
topiclist = []
topicid = 0

def hand_user_con(usr):
    global is_normal, topicid, userlist, topiclist
    while True:
        data = usr.skt.recv(1024).decode('utf-8').split('|')
        exist = False
        for user in userlist:
            if data[1] == user.username:
                usr.send_msg('UserExist')
                exist = True
                break
        if exist: continue
        else:
            usr.send_msg('LoginSuccess')
            break
    usr.username = data[1]
    userlist.append(usr)
    usr.send_msg('Status%d!' % len(userlist) + str(len(topiclist)))
    while True:
        data = usr.skt.recv(1024).decode('utf-8')
        time.sleep(1)
        print(usr.username, data)
        msg = data.split('|')
        if msg[0] == 'exit':
            usr.logout()
            userlist.remove(usr)
            break
        elif msg[0] == 'lsu':
            send_data = ['lsu']
            for user in userlist:
                send_data.append(user.username)
                send_data.append(user.addr)
            usr.send_msg('|'.join(send_data))
        elif msg[0] == 'lst' and topiclist != []:
            send_datas = []
            for topic in topiclist:
                send_datas.append('%s%s%s%s%s' % (str(topic.id),
                                                    topic.user.username, topic.user.addr, topic.date, topic.content))
            send_data_len = [str(len(x)) for x in send_datas]

```

```

        usr.send_msg('lstl' + ('l'.join(send_data_len)))
        data = usr.skt.recv(4).decode('utf-8')
        for send_data in send_datas:
            usr.send_msg(send_data)

    elif msg[0] == 'ntpc':
        topicid += 1
        topic = Topic.Topic(topicid, usr, msg[1])
        topiclist.append(topic)
    elif msg[0] == 'dtpc':
        tid = int(msg[1])
        success = False
        for topic in topiclist:
            if tid == topic.id and usr.username == topic.user.username:
                success = True
                break
        if success:
            topiclist.remove(topic)
            usr.send_msg('dtpclsuccess')
        else:
            usr.send_msg('dtpclfail')
    elif msg[0] == 'ch':
        success = False
        for user in userlist:
            if user.username == msg[1]:
                success = True
                user.send_msg('chl%s%s' % (usr.username, msg[2]))
        if success:
            usr.send_msg('sendlsuccess')
        else:
            usr.send_msg('sendlfail')

s = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
s.bind((HOST, PORT))
s.listen(5) # 监听端口
print('Waiting for connection...')
while is_normal:
    sock, addr = s.accept() # 接受一个新连接
    usr = User.User(sock, addr[0])
    t = threading.Thread(target=hand_user_con, args=(usr, )) # 创建新线程来处理TCP连接
    t.start()
s.close()

```

client.py

```

# -*- coding: utf-8 -*-

import time
import socket
import threading

HOST = "127.0.0.1"
PORT = 23333
username = ""

# global variable
is_normal = True

def valid_username(username):
    for c in username:
        if not (c.isalnum() or c == '_'):
            return False
    return True

def list_cmd():
    print('exit: 登出并退出')
    print('help: 打开帮助菜单')
    print('lsu: 列出当前在线的人')
    print('lst: 列出当前所有主题')
    print('ntpc <主题内容>: 创建一个主题')
    print('dtpc <主题序号>: 删除你创建的主题')
    print('ch <用户名> <聊天内容>: 向指定用户发送一条消息')

def recieve_msg(username, s):
    global is_normal
    while is_normal:
        data = s.recv(1024).decode('utf-8').split('|')
        if data[0] == 'lsu':
            print()
            for i in range(1, len(data), 2):
                print('%s(%s)' % (data[i], data[i + 1]))
        elif data[0] == 'lst':
            s.send('recv'.encode('utf-8'))
            print(data)
            for i in range(1, len(data)):
                length = int(data[i])
                topic = s.recv(length).decode('utf-8').split('|')
                print("\n%s. %s(%s) 创建于 %s" % (topic[0], topic[1], topic[2], topic[3]))
                print("----%s" % topic[4])
        elif data[0] == 'dtpc':
            print()
            if data[1] == 'success':
                print('删除成功')
            else:
                print('删除失败')
        elif data[0] == 'send':
            if data[1] == 'fail':
                print('\n发送失败')
        elif data[0] == 'ch':

```

```

        print('\n[%s]: %s' % (data[1], data[2]))
    else: s.close()

def handle_msg(msg, s):
    global is_normal
    msg = msg.split()
    if msg[0] == 'exit':
        s.send('exit'.encode('utf-8'))
        is_normal = False
    elif msg[0] == 'help':
        if len(msg) > 1:
            print('参数过多')
        else:
            list_cmd()
    elif msg[0] == 'lsu' or msg[0] == 'lst':
        if len(msg) == 1:
            s.send(msg[0].encode('utf-8'))
        else:
            print('参数过多')
    elif msg[0] == 'ntpc':
        if len(msg) == 1:
            print('参数过少')
        else:
            temp_str = ' '.join(msg[1:])
            s.send(('ntpc|%s' % temp_str).encode('utf-8'))
    elif msg[0] == 'dtpc':
        if len(msg) < 2:
            print('参数过少')
        elif len(msg) > 2:
            print('参数过多')
        else:
            s.send(('dtpc|%s' % msg[1]).encode('utf-8'))
    elif msg[0] == 'ch':
        if len(msg) < 3:
            print('参数过少')
        else:
            temp_str = ' '.join(msg[2:])
            s.send(('ch|%s|%s' % (msg[1], temp_str)).encode('utf-8'))
    else:
        print('找不到此命令, 请键入help打开帮助菜单')

try:
    s = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
    s.connect((HOST, PORT))
except:
    print('与服务端连接失败')
    exit()

while True:
    username = input('请输入你的用户名:')
    if not username: continue
    if not valid_username(username):
        print('用户名不能包含特殊字符')
        continue

```

```
s.send(('login|s' % username).encode('utf-8'))
data = s.recv(1024).decode('utf-8')
if data == 'UserExist':
    print('用户名已存在')
    continue
elif data == 'LoginSuccess':
    print('登录成功')
    break

data = s.recv(1024).decode('utf-8').split('|')
if data[0] == 'Status':
    print('在线人数: %s' % data[1])
    print('主题数: %s' % data[2])
list_cmd()

t = threading.Thread(target=recieve_msg, args=(username, s))
t.start()

while is_normal:
    msg = input('%s>' % username)
    if not msg: continue
    handle_msg(msg, s)
```