

多个客户端与服务端通信

我们的目的是要多个客户端同时访问服务器然后进行聊天，这里就需要涉及到线程的概念。

客户端还是保持不变，需要在服务端上加入多线程。

多线程

与Socket一样，我同样也没有接触过多线程，上网查询做了一个小的实验。

```
import threading

def print_work(cnt):
    for i in range(cnt):
        print('new thread print:', i)

t = threading.Thread(target=print_work, args=(10,))
t.start();
sum = 0
for i in range(100):
    sum = sum + i
print('Sum =', str(sum))
```

运行结果

```
new thread print: 0
Sum = 4950
new thread print: 1
new thread print: 2
new thread print: 3
new thread print: 4
new thread print: 5
new thread print: 6
new thread print: 7
new thread print: 8
new thread print: 9
```

于是就可以基本了解多线程是如何工作的了，开始着手服务端的开发

服务端

每次接受一个客户端的连接，都新建一个线程，只需要把上次的代码套一个线程就好了。

```
# -*- coding: utf-8 -*-
import socket
import threading

HOST = "127.0.0.1"
PORT = 23333

def tcplink(sock, addr):
    print('Accept new connection from %s:%s...' % addr)
    sock.send('Welcome!'.encode('utf-8'))
    data = sock.recv(1024)
    sock.send(('Hello, %s!' % data.decode('utf-8')).encode('utf-8'))
    sock.close()
    print('Connection from %s:%s closed.' % addr)

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 True:
    sock, addr = s.accept() # 接受一个新连接:
    t = threading.Thread(target=tcplink, args=(sock, addr)) # 创建新线程来处理TCP连接
    t.start()
```