

一、UDP:

1. 代码:

UDP:服务器

```
s.py - E:\Filesave\PythonCode\network\Cs双向\s.py (3.7.2)
File Edit Format Run Options Window Help
from socket import *
ADDR = ('', 21567)
udpSerSock = socket(AF_INET, SOCK_DGRAM)
udpSerSock.bind(ADDR)
while True:
    data, ADDR = udpSerSock.recvfrom(1024)
    print(data.decode(), ADDR)
    data = input('>').encode()
    udpSerSock.sendto(data, ADDR)
udpSerSock.close()
```

UDP:客户端，其中的 ip 是寝室通讯时自己的 ip

```
from socket import *
ADDR = ('172.29.131.189', 21567)
udpCliSock = socket(AF_INET, SOCK_DGRAM)
while True:
    data = input('>').encode()
    udpCliSock.sendto(data, ADDR)
    data, ADDR = udpCliSock.recvfrom(1024)
    print(data.decode(), ADDR)
udpCliSock.close()
```

2. 存在的问题：只能你发一条我接受一条然后在回你，比较死板

二、TCP:

1.代码:

TCP:服务器

```
from socket import *
import threading

ADDR = ('', 21567)
tcpSerSock = socket(AF_INET, SOCK_STREAM)
tcpSerSock.bind(ADDR)
tcpSerSock.listen(5)

def handle_c(tcpCliSock):
    while True:
        data = tcpCliSock.recv(1024)
        print(data.decode(), addr)

while True:
    tcpCliSock, addr = tcpSerSock.accept()
    #tcpCliSock.setblocking(0)
    t=threading.Thread(target=handle_c, args=(tcpCliSock,))
    t.start()
    while True:
        data = input('> ')
        tcpCliSock.send(data.encode())

tcpSerSock.close()
```

TCP: 客户端

```
c.py - E:\Filesave\PythonCode\network\cs\c.py (3.7.2)
File Edit Format Run Options Window Help

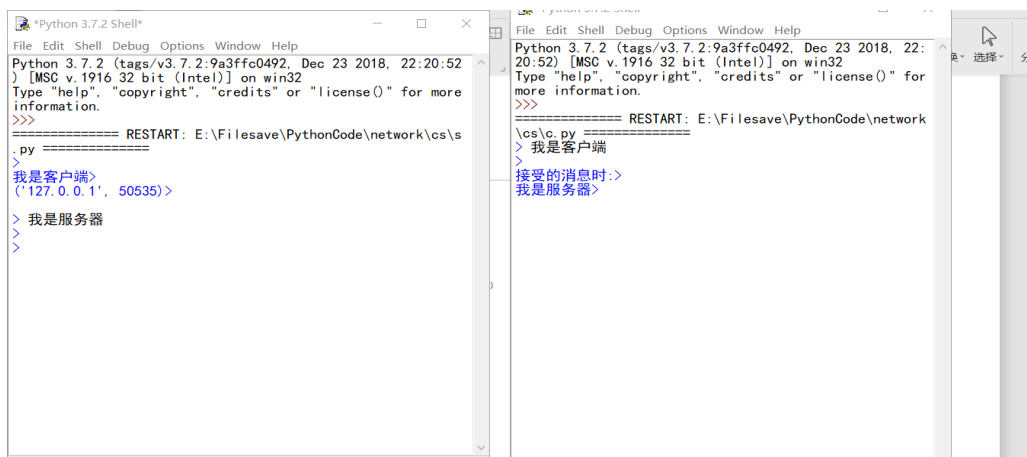
from socket import *
import threading

ADDR = ('localhost', 21567)
tcpCliSock = socket(AF_INET, SOCK_STREAM)
tcpCliSock.connect(ADDR)

def ReceiveInfo():
    while True:
        data=tcpCliSock.recv(1024)
        print("接受的消息时:", data.decode())

t=threading.Thread(target=ReceiveInfo)
t.start()
while True:
    data = input('> ')
    tcpCliSock.send(data.encode())
tcpCliSock.close()
```

2. 运行结果



```
*Python 3.7.2 Shell*
File Edit Shell Debug Options Window Help
Python 3.7.2 (tags/v3.7.2:9a3ffc0492, Dec 23 2018, 22:20:52) [MSC v.1916 32 bit (Intel)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
===== RESTART: E:\Filesave\PythonCode\network\cs\s
.py =====
> 我是客户端
('127.0.0.1', 50535)
> 我是服务器
>
```

```
Python 3.7.2 (tags/v3.7.2:9a3ffc0492, Dec 23 2018, 22:20:52) [MSC v.1916 32 bit (Intel)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
===== RESTART: E:\Filesave\PythonCode\network
\cs\c.py =====
> 我是客户端
> 接受的消息时:>
我是服务器>
```

3. 存在问题:

虽然可以实现相互交流自由,但是需要自己回车键已达到接受消息,这个是非常影响体验感的,还需要改进

三、TCP: 带界面版本

1. 代码: (截图接不下了只能复制粘贴了)

服务器:

'''

Created on 2019 年2 月28 日

@author: zhouning

'''

import tkinter as tk

from socket import *

import threading

'''

设置服务器方面的东西

'''

ADDR = ('', 21566)

tcpSerSock = socket(AF_INET, SOCK_STREAM)

tcpSerSock.bind(ADDR)

tcpSerSock.listen(5)

#线程用来处理客户端, 相当于小弟

def handle_c(tcpCliSock):

while True:

try:

data = tcpCliSock.recv(1024)

print(data.decode(),addr)

trecive.insert('end', addr)

trecive.insert('end', data.decode())

except Exception as e:

print(e)

tcpCliSock.close()

break

tcpCliSock, addr =tcpSerSock.accept()

thread=threading.Thread(target=handle_c ,args=(tcpCliSock,))

```

thread.start()
'''
开始设置界面
'''

window = tk.Tk()          #主窗口
window.title('服务器')    #窗口标题
window.geometry('400x500') #窗口尺寸

#下面写一些界面
l = tk.Label(window,
    text='消息',          # 标签的文字
    width=15, height=2    # 标签长宽
)
l.pack()                  # 固定窗口位置

trecive = tk.Text(window,height=10) #创建文本框，用户可输入内容
trecive.height=10
trecive.pack(padx=10,pady=10)

l2 = tk.Label(window,
    text='准备发送的消息', # 标签的文字
    width=15, height=2    # 标签长宽
)
l2.pack()                  # 固定窗口位置
t = tk.Text(window,height=10) #创建文本框，用户可输入内容
t.pack(padx=10,pady=10)

def sendInfo():            #用于按钮的发送消息
    var=t.get(0.0, 'end')
    if var!='':
        trecive.insert('end', var)
        tcpCliSock.send(var.encode())
        t.delete(1.0, 'end')
#下面的按钮用来发送消息
b = tk.Button(window,
    text='发送',          # 显示按钮上的文字
    width=15, height=2,
    command= sendInfo
)
b.pack()                  # 按钮位置
window.mainloop()         #循环消息，让窗口活

tcpSerSock.close()

```

```
print("我是服务器，结束了")
```

客户端:

'''

Created on 2019 年2 月28 日

@author: zhouning

'''

```
import tkinter as tk
```

```
from socket import *
```

```
import threading
```

```
ADDR = ('Localhost', 21566)
```

```
tcpCliSock = socket(AF_INET, SOCK_STREAM)
```

```
window = tk.Tk()           #主窗口
```

```
window.title('客户端')     #窗口标题
```

```
window.geometry('400x500') #窗口尺寸
```

#下面写一些界面

```
l = tk.Label(window,
```

```
    text='消息',          # 标签的文字
```

```
    width=15, height=2      # 标签长宽
```

```
)
```

```
l.pack()                  # 固定窗口位置
```

```
trecive = tk.Text(window,height=10) #创建文本框，用户可输入内容
```

```
trecive.height=10
```

```
trecive.pack(padx=10,pady=10)
```

```
l2 = tk.Label(window,
```

```
    text='准备发送的消息',    # 标签的文字
```

```
    width=15, height=2        # 标签长宽
```

```
)
```

```
l2.pack()                 # 固定窗口位置
```

```
t = tk.Text(window,height=10) #创建文本框，用户可输入内容
```

```
t.pack(padx=10,pady=10)
```

```
def sendInfo():
```

```
    var=t.get(0.0,'end')
```

```
    if var!='':
```

```
        trecive.insert('end', var)
```

```

        tcpCliSock.send(var.encode())
        t.delete(1.0, 'end')
        if var=='88':
            print("执行了退出")
            tcpCliSock.close()

#下面的按钮用来发送消息
b = tk.Button(window,
               text='发送',      # 显示按钮上的文字
               width=15, height=2,
               command= sendInfo
               )      # 点击按钮执行的命令
b.pack()              # 按钮位置

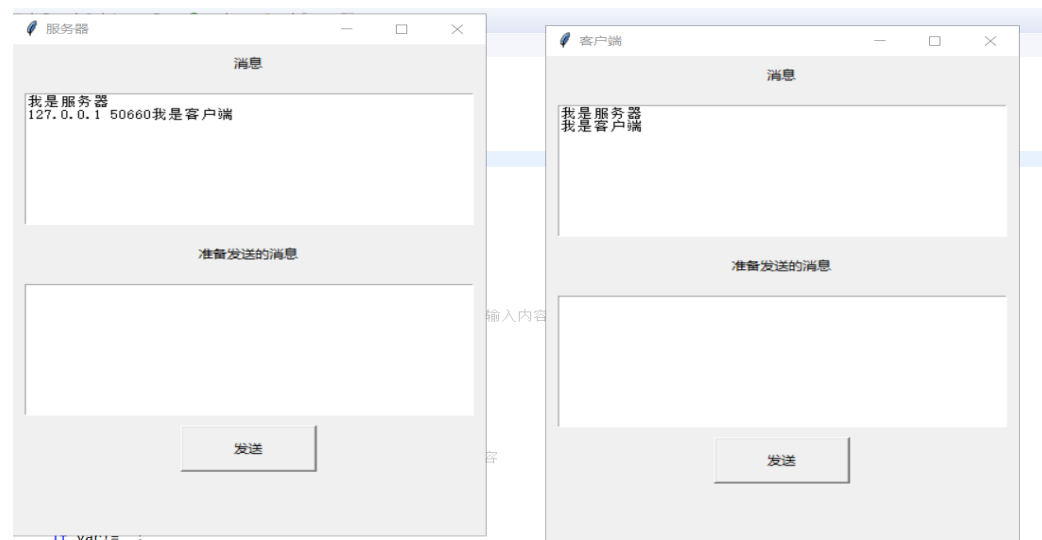
def ReceiveInfo():
    while True:
        data=tcpCliSock.recv(1024)
        print("接受的消息时:",data.decode())
        trecv.insert('end', data.decode())

tcpCliSock.connect(ADDR)
thread=threading.Thread(target=ReceiveInfo)
thread.setDaemon(True)
thread.start()

window.mainloop()      #循环消息，让窗口活起来
tcpCliSock.close()
print("我是客户端，结束了")

```

2. 运行结果:



3. 存在的问题:虽然解决了开始 tcp 不带界面版本当中的一些线程的问题,以及显示的问题,但是这个界面版本还是有问题,它只能 1 对 1,无法实现一个服务器对多个客户端,具体的就在于它服务器开线程是为了显示,而不是为了解决和客户端的通讯。所以在这个方面还需要改进,需要自己查找资料来解决这个问题。