**Python 基于TCP实现远程执行命令**

**服务端**

from socket import \*

import subprocess

import struct

# address=('127.0.0.1',8000)

# back\_log=5

# buffer\_size=1024

tcp\_server=socket(AF\_INET,SOCK\_STREAM) ##定义tcp协议的socket接口

tcp\_server.setsockopt(SOL\_SOCKET,SO\_REUSEADDR,1) ##加入这条socket配置，重用IP和端口，还可以在操作系统上配置ip和端口重用

tcp\_server.bind(('127.0.0.1',8000)) ##为套接字绑定IP和端口，（IP和端口可以标示全网唯一）

tcp\_server.listen(5) ## 可以接收的接收的连接数，并且监听端口

while True:

conn,addr=tcp\_server.accept() ##建立链接，并堵塞端口

print(addr)

while True:

try: ## 异常处理，防止客户端不正常断开

data\_cmd = conn.recv(1024) ## 接收消息

if not data\_cmd:break ##防止接收的消息为空

print('收到客户端的命令')

## 通过subprocess.Popen执行操作系统命令，并且把返回值重定向到subprocess.PIPE的管道零时存放

res = subprocess.Popen(data\_cmd.decode('utf-8'), shell=True,stderr=subprocess.PIPE,\

stdout=subprocess.PIPE,stdin=subprocess.PIPE)

err = res.stderr.read() ##通过res.stderr.read()读取错误消息

if err:

res\_data=err

else:

res\_data=res.stdout.read() ##通过res.stdout.read()读取标准输出消息

if not res\_data:

res\_data = '执行成功'.encode('gbk') ##防止操作系统执行命令后，返回值为空时，为客户端返回此消息

length = len(res\_data) ## 计算返回值数据长度

data\_length = struct.pack('i', length) ## 通过struct.pack处理返回值数据长度，并且也只能占用4个字节

conn.send(data\_length) ## 发送返回值数据长度（4字节）作为返回给客户端消息的包头，因为是定长的，所\

# 以客户端在接收时，第一次接收4个字节，就知道服务端返回的消息长度，就可以到自己的内存按次长度获取数据

conn.send(res\_data) ## 发送数据

except Exception as e:

print(e) ##处理方式

break

**客户端**

from socket import \*

import subprocess

import struct

# ip\_port=('127.0.0.1',8000)

buff\_size=1024

tcp\_client=socket(AF\_INET,SOCK\_STREAM)

tcp\_client.connect(('127.0.0.1',8000)) ## 链接服务端

while True:

cmd\_data=input('请输入命令：') ## 输入命令

if not cmd\_data:continue ##判断是否输入空值，为空跳出本次循环，让用户重新输入

if cmd\_data == 'quit': break ## 当用户输入quit时，断开链接（相当于不正常断开）

tcp\_client.send(cmd\_data.encode('utf-8')) ## 发送命令

length\_data = tcp\_client.recv(4) ## 第一次接收4字节，就是服务端用struct.pack封装为4字节的包头

head\_data=struct.unpack('i',length\_data)[0] ## 街封装，结果是列表类型

# 处理数据

res\_length=0

res\_msg=b''

while res\_length < head\_data:

res\_msg += tcp\_client.recv(buff\_size)

res\_length += len(res\_msg)

print(res\_length)

print('命令的执行结果是 ', res\_msg.decode('gbk'))

tcp\_client.close()

**输出结果：**

## 服务端：

('127.0.0.1', 63745)

收到客户端的命令 dir

## 客户端：

请输入命令：dir

903

命令的执行结果是 驱动器 D 中的卷是 软件

卷的序列号是 65F3-3762

D:\python\study\socket01 的目录

2020/03/10 18:57 <DIR> .

2020/03/10 18:57 <DIR> ..

2020/03/08 19:55 231 client.py

2020/03/10 18:57 1,067 cmd\_client\_tcp.py

2020/03/10 18:57 2,290 cmd\_server\_tcp.py

2020/03/09 15:54 333 ntp-client.py

2020/03/09 16:25 468 ntp-server.py

2020/03/08 19:54 315 server.py

2020/03/09 15:54 333 udp-client.py

2020/03/09 15:54 320 udp-server.py

2020/03/08 19:34 0 \_\_init\_\_.py

2020/03/08 22:34 443 客户端.py

2020/03/08 22:33 445 客户端11.py

2020/03/08 22:33 720 服务端.py

2020/03/10 17:12 <DIR> 练习

12 个文件 6,965 字节

3 个目录 76,889,382,912 可用字节