[socket 基于udp实现远程执行命令](https://www.cnblogs.com/dangrui0725/p/9483556.html)

server.py

[复制代码](javascript:void(0);)

# socket 基于udp实现远程执行命令

from socket import \*

import subprocess

ip\_port = ('127.0.0.1', 8080)

buffer\_size = 1024

udp\_server = socket(AF\_INET, SOCK\_DGRAM)

udp\_server.bind(ip\_port)

while True:

cmd, addr = udp\_server.recvfrom(buffer\_size)

cmd = cmd.decode('utf-8')

print('收到客户端命令', cmd)

res = subprocess.Popen(cmd, shell=True, # 第一个参数：命令字符串，第二个参数指定由shell处理

stderr=subprocess.PIPE, # 将基本的输入、输出及错误都放入管道

stdin=subprocess.PIPE, # 这些在管道里的信息都是字节形式,编码为utf-8

stdout=subprocess.PIPE

)

err = res.stderr.read() # 定义一个err变量接收基本的错误信息

if err: # 如果错误信息不为空

cmd\_res = err # 输出的结果为基本的错误信息

else:

cmd\_res = res.stdout.read() # 输出的结果为基本的输出信息

if not cmd\_res: # 有些命令无返回结果，需要进行判断

cmd\_res = '该命令没有返回结果'.encode('gbk')

udp\_server.sendto(cmd\_res, addr) # 向客户端发送执行的结果

udp\_server.close()

[复制代码](javascript:void(0);)

client.py

[复制代码](javascript:void(0);)

from socket import \*

ip\_port = ('127.0.0.1', 8080)

buffer\_size = 1024

udp\_client = socket(AF\_INET, SOCK\_DGRAM)

while True:

cmd = input('请输入命令').strip()

if not cmd: continue

if cmd == 'quit': break

cmd = cmd.encode('utf-8')

udp\_client.sendto(cmd, ip\_port)

cmd\_res, addr = udp\_client.recvfrom(buffer\_size) # 这里的值过小会接收不全服务端发来的信息

print('命令执行的结果是', cmd\_res.decode('gbk')) # windows系统默认编码为gbk

udp\_client.close()

[复制代码](javascript:void(0);)