#udpserver.py

import socket

import os

# 创建一个UDP socket对象， AF\_INET表示使用IPv4地址，SOCK\_DGRAM表示使用UDP协议

server\_socket = socket.socket(socket.AF\_INET, socket.SOCK\_DGRAM)

# 绑定端口号，假设端口号为12345，""表示接受任意IP地址的连接

server\_port = 12345

server\_socket.bind(("", server\_port))

# 打印服务器启动信息，告知用户服务器正在监听的端口号

print(f"UDP Server is running on port {server\_port}...")

# 无限循环以接收数据

while True:

   # 接收客户端发送的数据，recvfrom方法返回数据和发送者的地址

    data, addr = server\_socket.recvfrom(1024)  # 缓冲区大小为1024字节

    print(f"Received {len(data)} bytes from {addr}")

    # 将接收到的数据写入文件，'ab'模式表示以二进制格式追加数据到文件末尾

    with open("received\_file", "ab") as file:  # 'ab'模式表示追加到文件末尾

        file.write(data)

    # 打印信息，告知用户数据已经被写入文件

    print("Data has been written to 'received\_file'")

#udpclient.py

import socket

import sys

# 检查命令行参数数量

if len(sys.argv) != 3:

print("Usage: python udpclient.py <ServerIP> <ServerPort>")

sys.exit(1)

# sys.argv是一个列表，其中包含了命令行调用时传递的参数

# 如果参数数量不等于3（脚本名+两个参数），则打印使用方法并退出

# 从命令行参数获取服务器的IP地址和端口号

server\_ip = sys.argv[1]

# sys.argv[0]是脚本名称，sys.argv[1]是第一个参数，即服务器IP地址

server\_port = int(sys.argv[2])

# sys.argv[2]是第二个参数，即服务器端口号，转换成整数类型

# 创建UDP socketp AF\_INET表示使用IPv4地址，SOCK\_DGRAM表示使用UDP协议

client\_socket = socket.socket(socket.AF\_INET, socket.SOCK\_DGRAM)

# 打开要发送的文件

file\_path = "file" # 这里假设要发送的文件名为file

try:

with open(file\_path, "rb") as file:

# 读取文件并发送

while True:

data = file.read(1024) # 每次读取1024字节

if not data:

break # 如果读取的数据为空，说明文件已经读取完毕，跳出循环

client\_socket.sendto(data, (server\_ip, server\_port)) # sendto方法发送数据到指定的IP地址和端口号

print(f"File '{file\_path}' has been sent to {server\_ip}:{server\_port}")# 打印文件发送成功的信息

except FileNotFoundError:

print(f"The file '{file\_path}' does not exist.")

# 关闭socket连接，释放资源[pd]

client\_socket.close()

#输入python3 udpclient.py 192.168.244.21(目标地址) 12345(目标端口号与udpserver.py中的端口号保持一致)