

# 20377199 赵芮箐 第13周作业

**作业内容：** 题目一：利用socket和多线程，实现支持多人对话的聊天室。具体地，实现Manager和Chatter 两个类，Chatter只需和Manager之间建立一对一联系，而Manager则负责广播或转发所有用户的消息。请在实际中找个场景运用。

相关要求如下：

1. 实现Manager类, 服务器, 管理成员进入和离开聊天室, 接收成员消息并广播
2. 实现Chatter类, 用户, 向管理员发送加入和退出请求, 发送和接收消息
3. Manager类使用多线程服务多个用户
4. Chatter用户发送和接收消息需要依赖不同线程进行
5. Manager类具备定向转发功能, 比如Chatter可以在消息中通过@指定特定用户, 这样Manager将仅转发给被指定用户。
6. Chatter在离开时, 自动保存聊天记录到硬盘 (包括时间、发信人, 信息)。
7. Manager也应保存所有聊天室记录到硬盘。

题目二：根据udp的演示例子，实现一个简单的“无声监控”服务程序，并在实际中找个场景运用。

1. 服务端要支持多线程，客户端退出时要停止相应子线程。
2. 测试时要实现多个接入测试，即有多个客户端获取视频流。
3. 客户端按一定时长存储视频流 (比如每10分钟存储一个文件)。
4. 服务端记录客户端接入的日志 (哪个客户端, 何时接入, 何时离开) 文件。
5. 要用类进行封装。
6. (附加) 感兴趣的同学了解一下组播，并尝试用组播来减轻服务端压力。

## 题目二思路：

- 服务端的主线程用tcp一直监听，客户端进行接入
- 获取客户端的ip和端口后，开启子线程用udp传输监控视频数据
- 客户端退出时会通过udp反馈信息，子线程随即结束

maybe tcp&udp一起用的写法有点笨蛋，但是是改了多次后妥协的结果。

只用udp实现全部的话，客户端退出后主线程的网络也会断掉，其他的客户端就没办法接入了。

所以最后选择用tcp实现接入确认的这部分，不得不说tcp三次握手is可靠许多。

anyway, 至少是tcp和udp都练习到了就是说。

## Part1: 服务端

```
import cv2
import sys
import numpy as np
from socket import *
from threading import Thread
from datetime import datetime

class Server():
    def __init__(self, socket, addr, capture):
        self._socket = socket
        self._addr = addr
        self._cap = capture

    def __get_cam(self):
        while True:
            ret, frame = self._cap.read()
            yield frame

    def post_cam(self):
        for frame in self.__get_cam():
```

```

        img_encode = cv2.imencode('.jpg', frame)[1]
        data_encode = np.array(img_encode)
        data = data_encode.tobytes()
        self._socket.sendto(data, self._addr)
        reply, addr = self._socket.recvfrom(1024)
        # 收到客户端消息若为退出, 则结束线程
        if reply.decode('utf-8') == 'end':
            break
    with open('week13/ServerLog.txt', 'a', encoding='utf-8') as f:
        f.write(f'{datetime.now()} 客户端{self._addr} 退出\n')

def accpet_client():
    # tcp通讯用于获取接入的ip
    tcp_socket = socket(AF_INET, SOCK_STREAM)
    tcp_socket.bind(('0.0.0.0', 9999))
    tcp_socket.listen(5)
    # udp通讯用于传输视频
    udp_socket = socket(AF_INET, SOCK_DGRAM)
    capture = cv2.VideoCapture(0)
    capture.set(3, 160) # 故意修改小的, 不然mac上容易超过udp发送数据的长度限制
    capture.set(4, 90) # windows上好像不用这么小
    while True:
        print('新的客户端正在接入中...')
        conn, addr = tcp_socket.accept()
        client = (addr[0], int(conn.recv(1024).decode('utf-8')))
        # 记录Log日志
        with open('week13/ServerLog.txt', 'a', encoding='utf-8') as f:
            f.write(f'{datetime.now()} 客户端{client} 接入\n')
        # 开启子线程, 传输监控视频
        s = Server(udp_socket, client, capture)
        t = Thread(target=Server.post_cam, args=(s,))
        t.start()

if __name__ == "__main__":
    a = Thread(target=accpet_client)
    a.start()
    a.join()

```

## Part2: 客户端

```

import cv2
import sys
import time
import numpy as np
from socket import *
from datetime import datetime

rsize = 400000
fourcc = cv2.VideoWriter_fourcc(*'XVID')

class Receiver:
    def __init__(self, socket, addr, savetime=15,
savepath='D:/code/mp2022/week13/camera'):
        self._socket = socket
        self._addr = addr

```

```

self._time = savetime
self._path = savepath

def __save_cam(self, frame):
    # 每一段时间储存一次视频
    if time.time() - self._start > self._time:
        self._start = time.time()
        path = self._path + (datetime.now().strftime('%H-%M-%S')) + '.avi'
        self._out = cv2.VideoWriter(path, fourcc, 10.0, (160,120))
        self._out.write(frame)

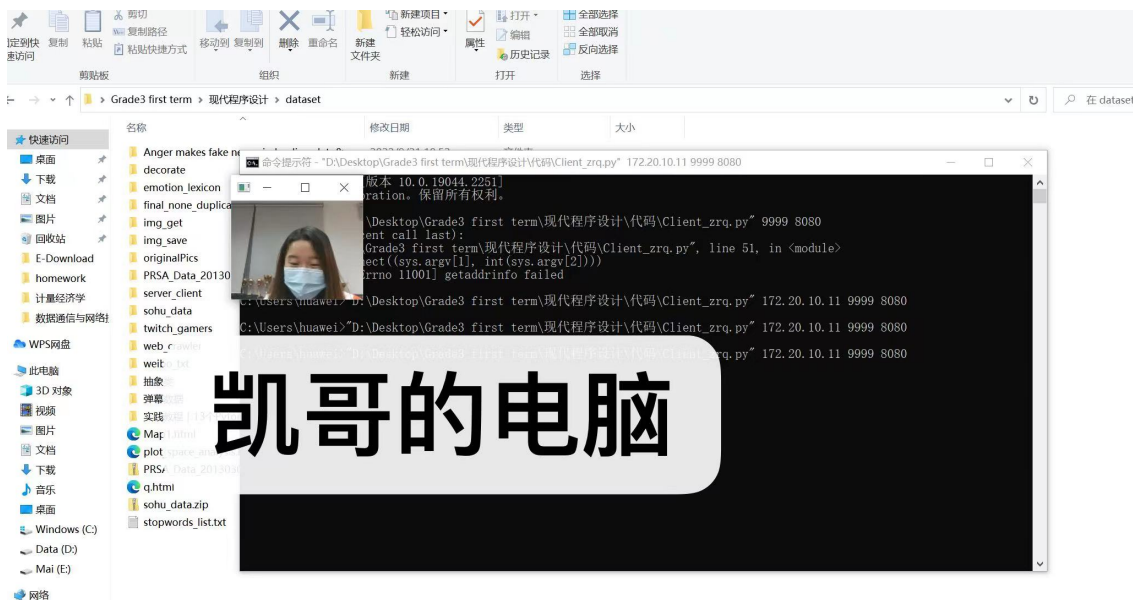
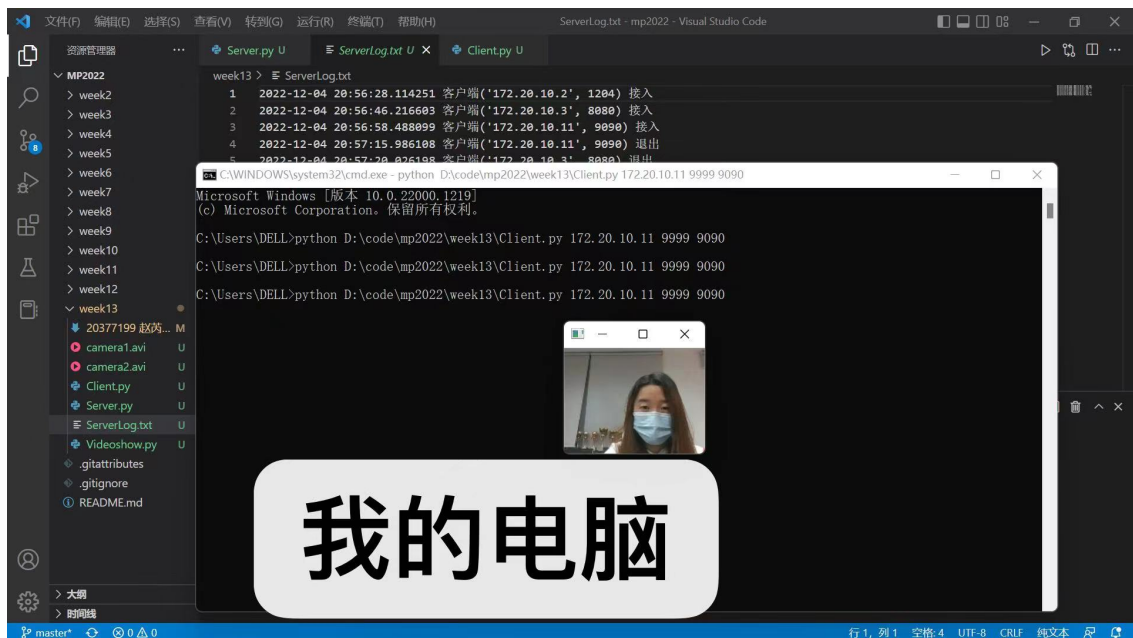
def receive_cam(self):
    self._socket.bind(self._addr)
    self._start = time.time()
    path = self._path + (datetime.now().strftime('%H-%M-%S')) + '.avi'
    self._out = cv2.VideoWriter(path, fourcc, 10.0, (160,120))
    # fourcc 指定编码器; fps 要保存的视频的帧率
    while True:
        data, addr = self._socket.recvfrom(rsize)
        nparr = np.frombuffer(data, np.uint8)
        img_decode = cv2.imdecode(nparr, cv2.IMREAD_COLOR)
        cv2.imshow('receive', img_decode)
        self.__save_cam(img_decode)      # 储存每一帧
        c = cv2.waitKey(50)
        udp_socket.sendto('continue'.encode('utf-8'), addr)
        if c == 27:                      # 按了esc后可以退出
            cv2.destroyAllWindows()
            udp_socket.sendto('end'.encode('utf-8'), addr)
            break

if __name__ == "__main__":
    # 和服务器连接, 将自己ip和port发过去
    # sys.argv[1], sys.argv[2] 是服务器的ip和port
    # sys.argv[3] 是客户端用于接收视频的端口号
    tcp_socket = socket(AF_INET, SOCK_STREAM)
    tcp_socket.connect((sys.argv[1], int(sys.argv[2])))
    tcp_socket.send((sys.argv[3]).encode('utf-8'))
    udp_socket = socket(AF_INET, SOCK_DGRAM)
    r = Receiver(udp_socket, ('0.0.0.0', int(sys.argv[3])))
    r.receive_cam()

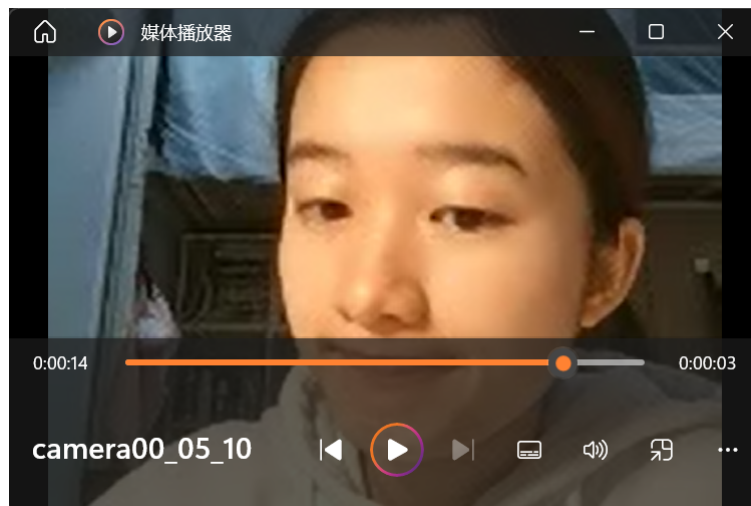
```

## 测试结果：

- 多台客户端链接获取视频：



- 保存的视频:



可爱而又模糊的我

- Log日志:

```
Server.py U  ServerLog.txt U X  Client.py U  Videoshow.py U
week13 >  ServerLog.txt
1  2022-12-04 20:56:28.114251 客户端('172.20.10.2', 1204) 接入
2  2022-12-04 20:56:46.216603 客户端('172.20.10.3', 8080) 接入
3  2022-12-04 20:56:58.488099 客户端('172.20.10.11', 9090) 接入
4  2022-12-04 20:57:15.986108 客户端('172.20.10.11', 9090) 退出
5  2022-12-04 20:57:20.026198 客户端('172.20.10.3', 8080) 退出
6  2022-12-04 20:57:22.718187 客户端('172.20.10.2', 1204) 退出
7  2022-12-04 20:57:31.680824 客户端('172.20.10.2', 1204) 接入
8  2022-12-04 20:57:35.505605 客户端('172.20.10.2', 1204) 退出
9  2022-12-04 20:57:47.944899 客户端('172.20.10.11', 9090) 接入
10 2022-12-04 20:58:24.651136 客户端('172.20.10.11', 9090) 退出
11 2022-12-04 21:00:01.929524 客户端('172.20.10.3', 8080) 接入
12 2022-12-04 21:00:52.283854 客户端('172.20.10.3', 8080) 退出
13 2022-12-04 21:02:38.223976 客户端('172.20.10.3', 8080) 接入
14 2022-12-04 21:02:52.004572 客户端('172.20.10.3', 8080) 退出
15 2022-12-04 21:03:54.952529 客户端('172.20.10.3', 8080) 接入
16 2022-12-04 21:04:17.753745 客户端('172.20.10.3', 8080) 退出
17 2022-12-04 21:05:43.596142 客户端('172.20.10.3', 8080) 接入
18 2022-12-04 21:06:08.751459 客户端('172.20.10.3', 8080) 退出
19
```

可以多台客户端同时接入, 退出去之后也可以再进来

## 代码:

[https://github.com/rachhhhing/mp2022\\_python/blob/master/week13](https://github.com/rachhhhing/mp2022_python/blob/master/week13)

## Ref:

- UDP通信: [https://blog.csdn.net/qq\\_19446965/article/details/110678592](https://blog.csdn.net/qq_19446965/article/details/110678592)
- opencv: <https://zhuanlan.zhihu.com/p/44255577>

最后, 本周作业特别鸣谢 ljsx 和 jzk 同学, 在一起分装物资的晚上专门带电脑出门帮我测试

特别批评 dxi 同学, 明明也要来却以电脑太重为由拒绝了我 🙄