

口罩预约系统设计文档

语言 python3.8

通信协议设计

通信所用数据格式为 json，可以方便的转成字典使用。

客户端发出的数据类型：

客户端数据都以{'opration':op, 'dist':dist, 'name':name, ...}的格式发出，根据 op 操作数的不同服务端会解析后面变量，而 dist 和 name 变量用于确定具体的用户。

- op=0, 表示用户注册，还需要的参数是用户第一次设定的密码
- op=1, 表示用户登陆，同样需要用户密码用作服务端验证
- op=2, 表示退出登陆，不需要其他参数
- op=3, 用户请求所有地区的情况，不需要其他参数
- op=4, 用于请求某个地方的具体预约情况，参数表示搜索目标地址
- op=5, 用户预定口罩，参数为请求的口罩数量

服务端数据以{'opInfo':op, ...}的格式发出，op 表示服务端的操作或客户端操作的结果，参数根据 op 确定。

- op=0, 表示返回显示在首页的信息，参数为用户未接受的消息列表或搜索的某地区的预约情况，或所有地区的整体情况
 - op=1, 表示用户注册成功
 - op=2, 表示用户登陆成功
 - op=3, 表示密码错误
 - op=4, 表示用户账户不存在
 - op=5, 表示用户未登陆
 - op=6, 表示用户成功退出登陆状态
 - op=7, 表示用户预定口罩成功
 - op=8, 表示当前地区不开放预约或口罩已预约完
 - op=9, 表示用户请求预约的地区不存在
 - op=10, 表示用户的操作数错误
 - op=11, 表示返回配送口罩的信息
 - op=12, 表示返回显示在消息界面的信息，参数是各种来源的字符串或口罩的取件码
- 本协议由于使用了 json 数据，通过数字代表操作，具有较好的扩展性。

服务端设计：

为了保证服务端的运行效率，服务端采用命令行进行操作。服务端的运行启动两个子线程，一个用作 shell 解析和执行命令，一个用作接收通信端的消息，接收消息线程通过针对不同的访问开子线程处理事务，提高了服务端的效率，后续 shell 根据需要再开子线程实现通信功能。

服务端维护 8 个全局变量，分别是在线用户字典，各地信息字典，各地区用户信息字典，以及不在线用户未接收消息的字典，以及四个变量的线程锁，共 8 个全局变量。我们通过一个("地址","用户名")元组来确定一个通信实体，并一次来记录在线用户的端口和地址，以及为接收用户的信息列表，注意，由于未接收用户消息都是在用户上线后发出，所以不用记录用户的套接字，各地用户详细信息为了提高查找效率，先用地区将用户分开，再在各地名维护的字典下按照用户名确定密码和订阅口罩的数量，地区信息字典用于记录每个地区的预约信息，每个地址键确定了一个数组，数组记录了预约状态，口罩数量，已被预约口罩数

量，以及参加预约者的集合。下面给出几个数据结构的样例：

在线用户字典：

```
{("dist","usrName"):(("addr",port), ...)}
```

各地信息字典：

```
{"地区名": [0(是否在预约), 0(口罩数量), 0(已被预约数量), {}(参加预约人集合)], ...}
```

各地区用户信息字典：

```
{"dist": { "usrName": [usrId,0(预约数量),00000(用户取口罩码)], ...}, ...}
```

不在线用户未接收消息的字典：

```
{("dist","usrName"):[ "msg",...], ...}
```

同时，由于多个线程读写这些文件，我们在每次读写这些变量时加上线程锁，使用完后再释放。

服务端的 shell 线程通过轮询标准输入解析命令，命令以空格分割以确定命令和参数，服务端的功能主要是管理变量和和用户数据，具体功能将在使用手册部分说明。

服务端的通信线程类有四个，分别实现的是正常接收用户请求，向在线用户发送消息，向口罩预约者发送取件码，向某个列表中的用户发送消息。具体处理了通信协议中的各种请求和发出各种数据。

客户端设计

客户端通过 ui 界面获取用户的额操作，具有有一定的用户友好性。ui 界面作为主线程，解析和识别用户操作，然后根据用户需求发出消息，并且接收和解析服务端的数据，然后为用户展示。

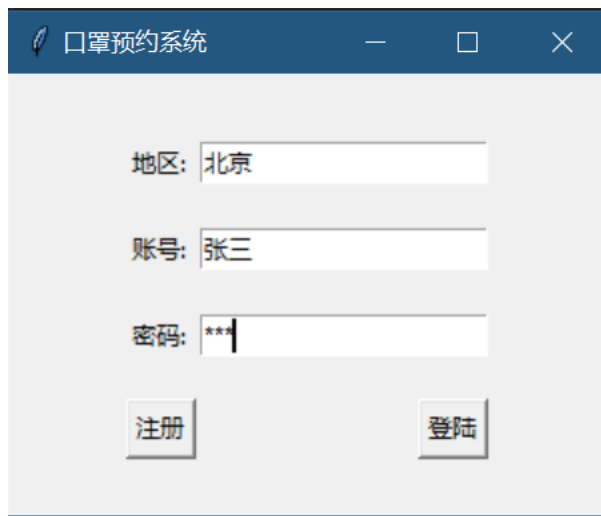
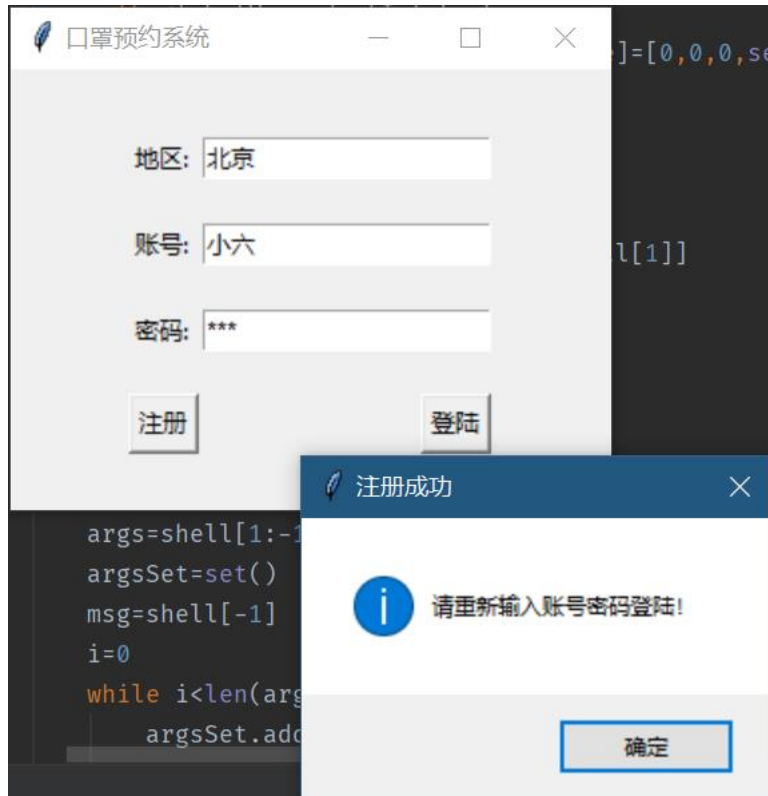
ui 界面使用了 tkinter 标准库，一共有两个主要的 page，第一个 page 用于验证登陆和注册，第二个 page 用于实现程序主要的功能，第二个 page 通过菜单可以切换操作界面，即首页，预约界面和消息界面，分别实现查询所有地区情况、查询某地具体情况、预约口罩、接收其他消息。发送消息的逻辑于 ui 界面是串行的，不主动接收数据，而是通过接收数据的另一线程读取数据，由于加锁会导致接收数据的线程在获得锁后始终拥有锁，并且用 threading.event() 临时阻塞线程没有成功，同时考虑到 ui 界面接收的数据都是一来一回的形式，所以认为可以不加锁。

客户端开启了一个子线程用于接收任何经过绑定套接字的数据，这些数据分为主动请求的和被动接收的，所以可以将这些数据分开存储在全局变量中，而这也是客户端维护的所有全局变量。客户端的消息处理界面同通过在刷新按钮后读取未处理消息的列表，将消息显示在界面上。

用户使用手册

客户端使用

登陆和注册：在三个框中分别填入地区，姓名和密码，然后点击登陆或注册按钮



主页用于查询信息，如果直接点击查询按钮不输入地址，将会获得所有地区的情况，如果输入具体的地址将会获得该地的详细情况。

口罩预约系统

主页 参加预定 消息 设置

查询地区:

北京 : 正在预约

口罩预约情况: 8/10 预定人数:3

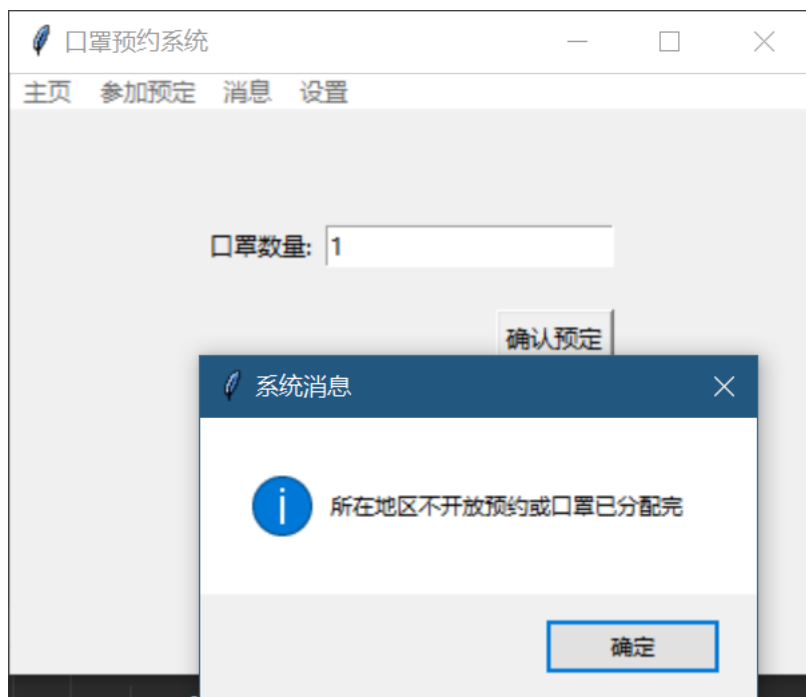
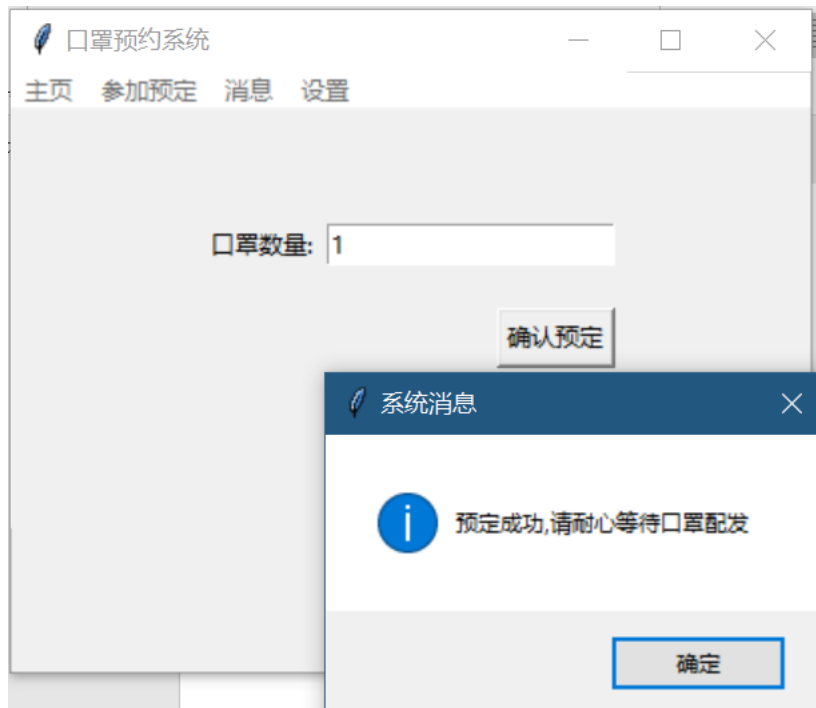
口罩预约系统

主页 参加预定 消息 设置

查询地区:

北京 : 不在预约

口罩预约：输入所需口罩数量，客户端会做基本判断输入是否合理，然后返回服务端的预约成功与否的信息。



消息接收：刷新后获得响应的口罩消息或服务端发送的消息





通过直接关闭界面，然后捕获信号之后向服务器发送退出登陆的信息。

服务端使用

- /msg [(用户列表)] 信息：群发消息
- /dist：获得所有地区的整体情况
- /append (地区)：增加地区
- /erase (地区)：删除地区
- /exit：退出命令行
- /enter (地区)：进入某地区
- /openNewround (口罩数量)：开启新一轮预约
- /list 列出本地区的详细情况
- /kickout (用户名)：强制用户下线
- /ban (用户名)：禁封用户账号
- /handout：结束本轮预约，并向每一个用户发送取件码，不在线用户在上线后自动发送
- /leave：退出行政区管理