使用说明

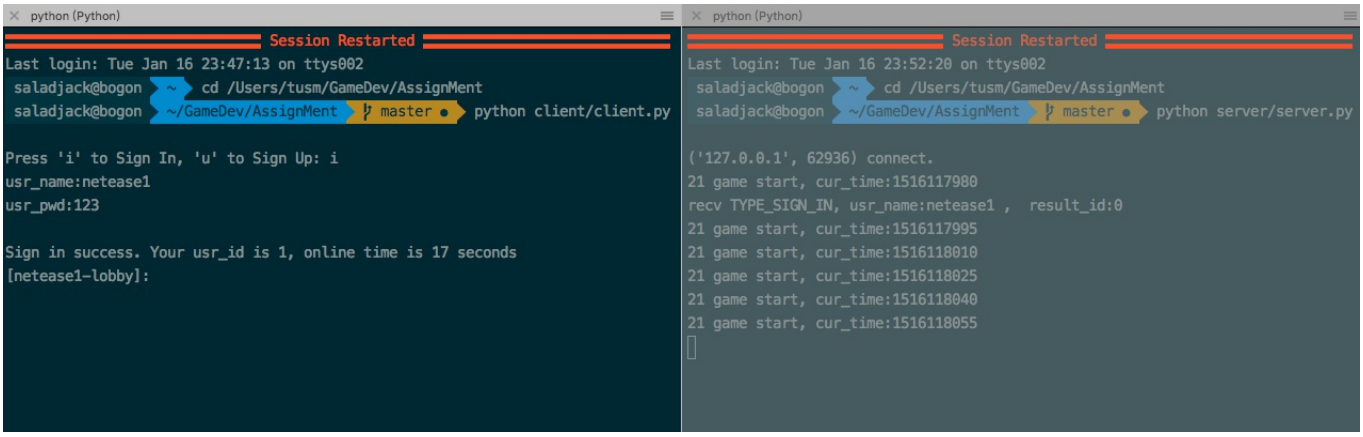
1.**必须**先运行服务端程序server.py

```
$ python server.py
```

2.之后运行client.py

```
$ python client.py
```

3.客户端输入i登录，输入u注册



4.登录成功后，用户会默认进入大厅(lobby)，用户可直接在大厅里聊天，也可以输入特定的**命令**来进行其它的操作，具体命令的操作方式如下：

```
$ [netease1-lobby]:/create room      # 创建并进入房间
$ [netease1-lobby]:/enter room 1     # 进入1号房间
$ [netease1-room-1]:/quit room       # 退出房间
$ [netease1-room-1]:/21game 4+5+6+6  # 参加21点游戏，提交的答案为：4+5+6+6
$ [netease1-lobby]:/chat to netease2 # 与用户netease2私聊
$ [netease1-private]:/chat quit      # 退出私聊
$ [netease1-lobby]:/sign out         # 注销
```

架构说明

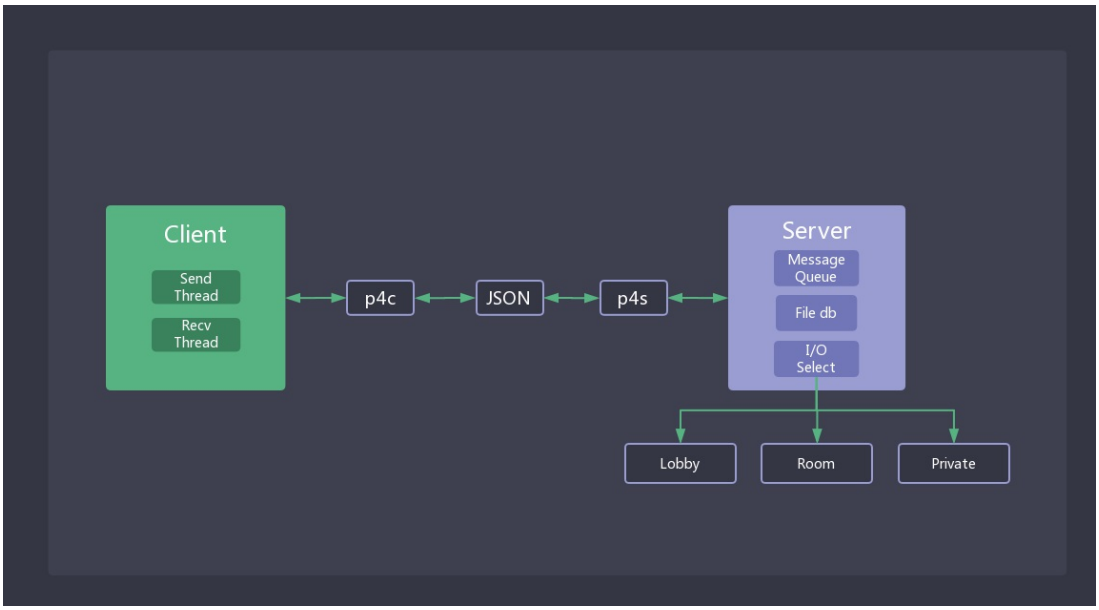
1.server/db存储了所有已有用户的信息，文件每一行的数据格式为

```
用户名 密码 用户ID 最近一次登录的时间戳(秒) 总登录时长(秒)
```

2.C/S端各有一份协议p4c/p4s，协议在传输时会序列化成JSON格式

3.考虑到在Python中，Windows的select()方法只能接收socket的输入流，而不像Linux还能接收sys.stdin的输入流。所以为了使sys.stdin不会阻塞接收信息，在Client端收、发逻辑要处于不同的线程

4.架构示意图



注意

1.为方便测试21点游戏，可运行服务端测试程序server-test.py进行测试。

2.游戏每30秒发起一次，答题时间限制为15秒，每次的游戏数字均为4 5 6 6

```
$ python server-test.py
```

```
$ [netease1-room-1]:Black Jack is ready,the numbers are 4 5 6 6
$ [netease1-room-1]:/21game 4+5+6+6
$ [netease1-room-1]:Your answer has sent to server, wait seconds...
$ [netease1-room-1]:Winner is netease1, his/her answer is 4+5+6+6=21
```