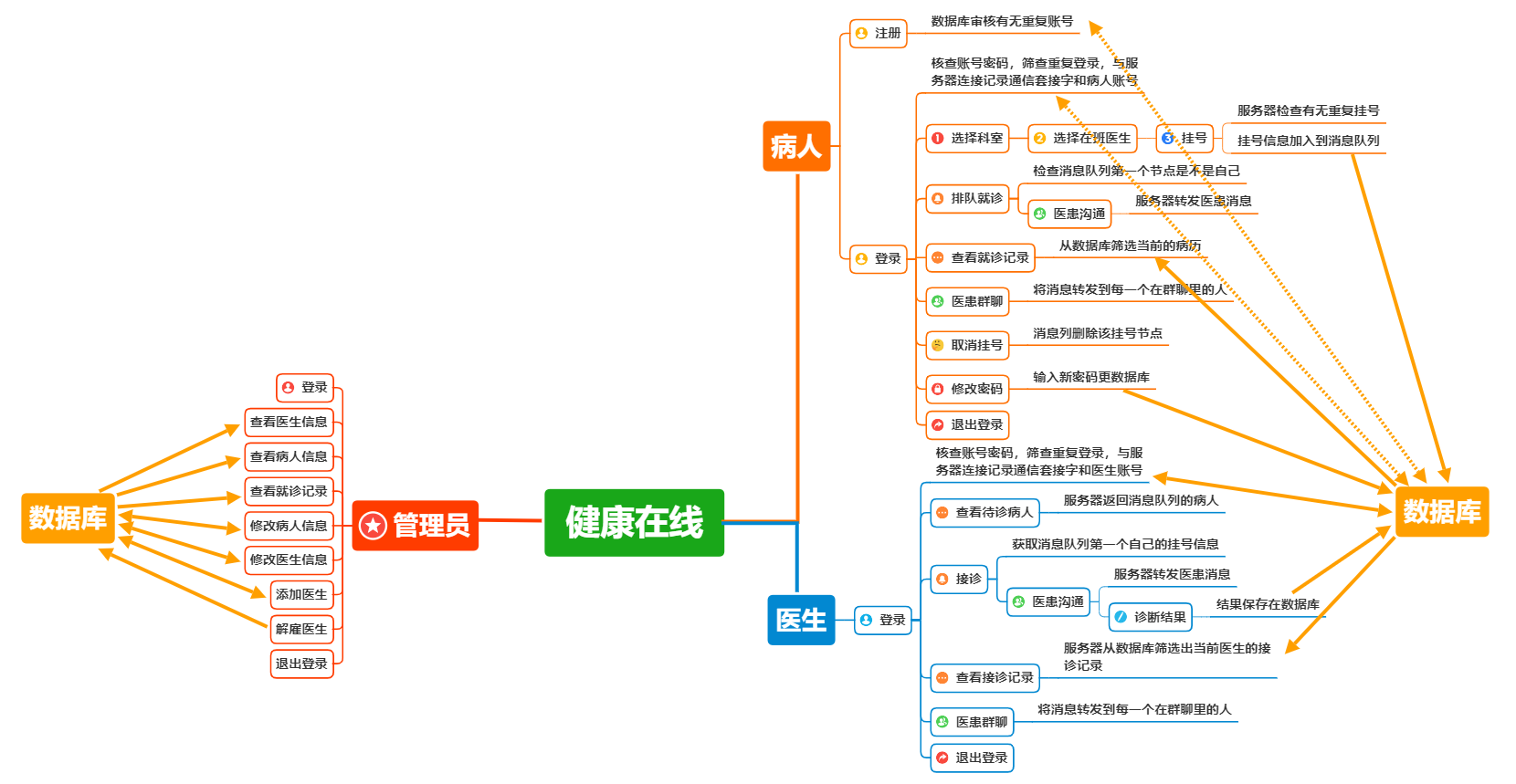
健康在线演示文档

1. 引言

**本开发文档旨在详细描述健康在线系统的开发过程，包括系统架构、功能模块、技术栈、运行过程等。该系统主要用于病人挂号、医生接诊、病人与医生之间的通信等功能。**

**思维导图**

****

2. 系统架构

**2.1 客户端-服务器架构**

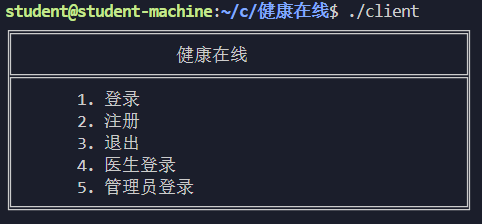
**系统采用经典的客户端-服务器架构，客户端负责用户交互，服务器负责处理业务逻辑和数据存储。**

**2.2 模块划分**

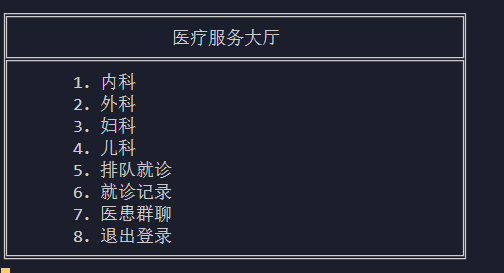
* **用户界面模块**

**将界面数据存放在文件中，通过函数调用输出文件显示出来**

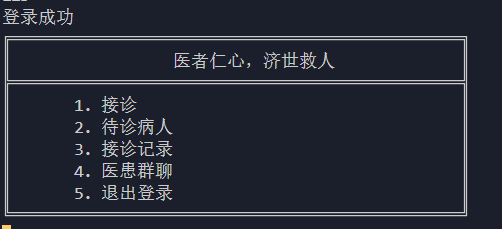
**1.主菜单**



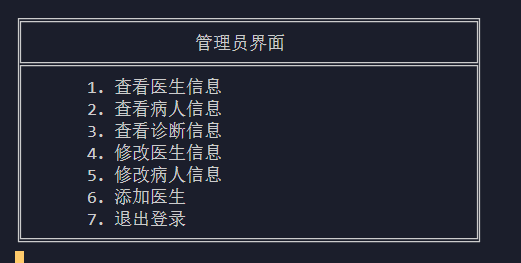
**2.病人登录后菜单**

****

**3.医生登陆成功后菜单**

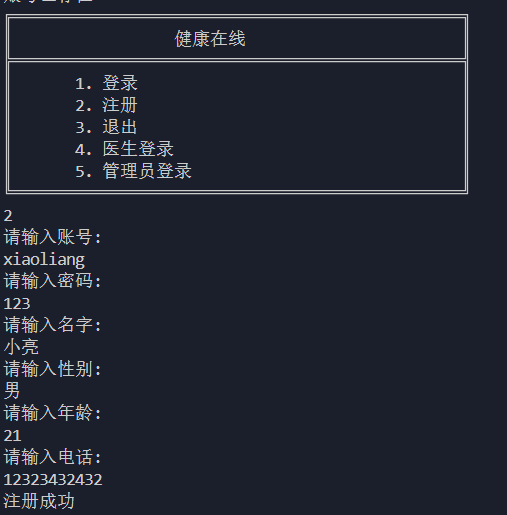
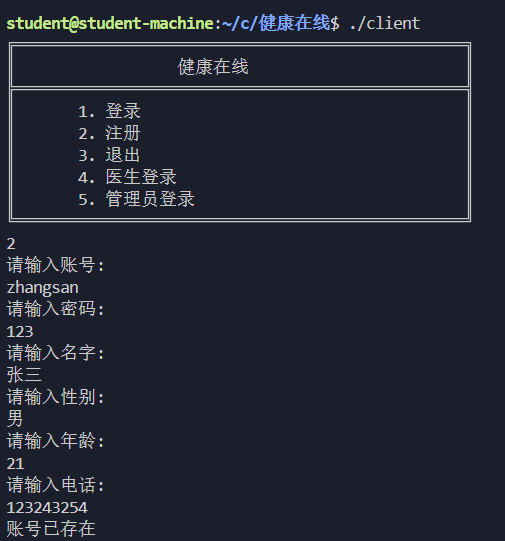
****

**4.管理员登录成功后菜单**

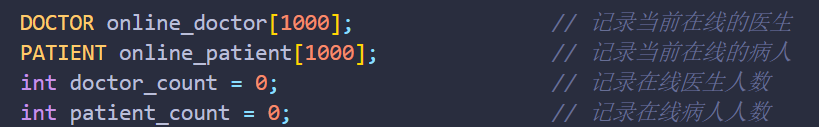
****

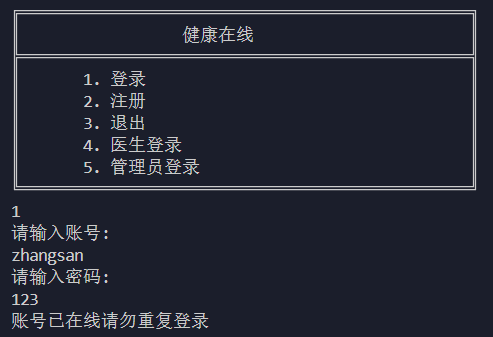
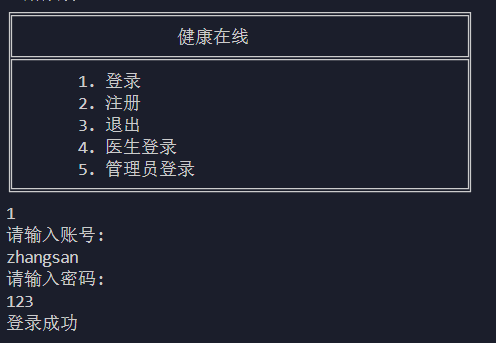
* **注册登录模块**

**1.病人注册，服务器审核账号是否存在存在则返回错误信号**

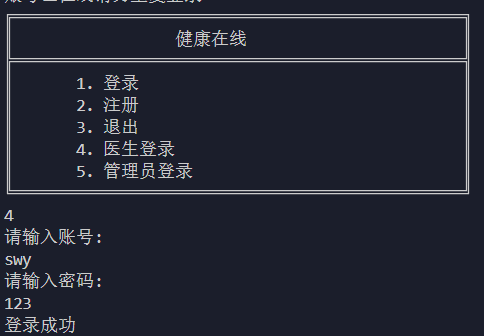
****

**2.病人登录，客户端输入账号密码发送到服务器，服务器审核账号密码是否正确，该账号是否已经在线，登陆成功后记录登录信息**

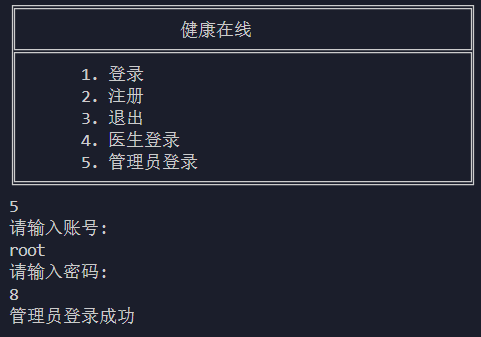
****

****

**3.医生登录，同病人登录一样，医生账号由管理员管理**

****

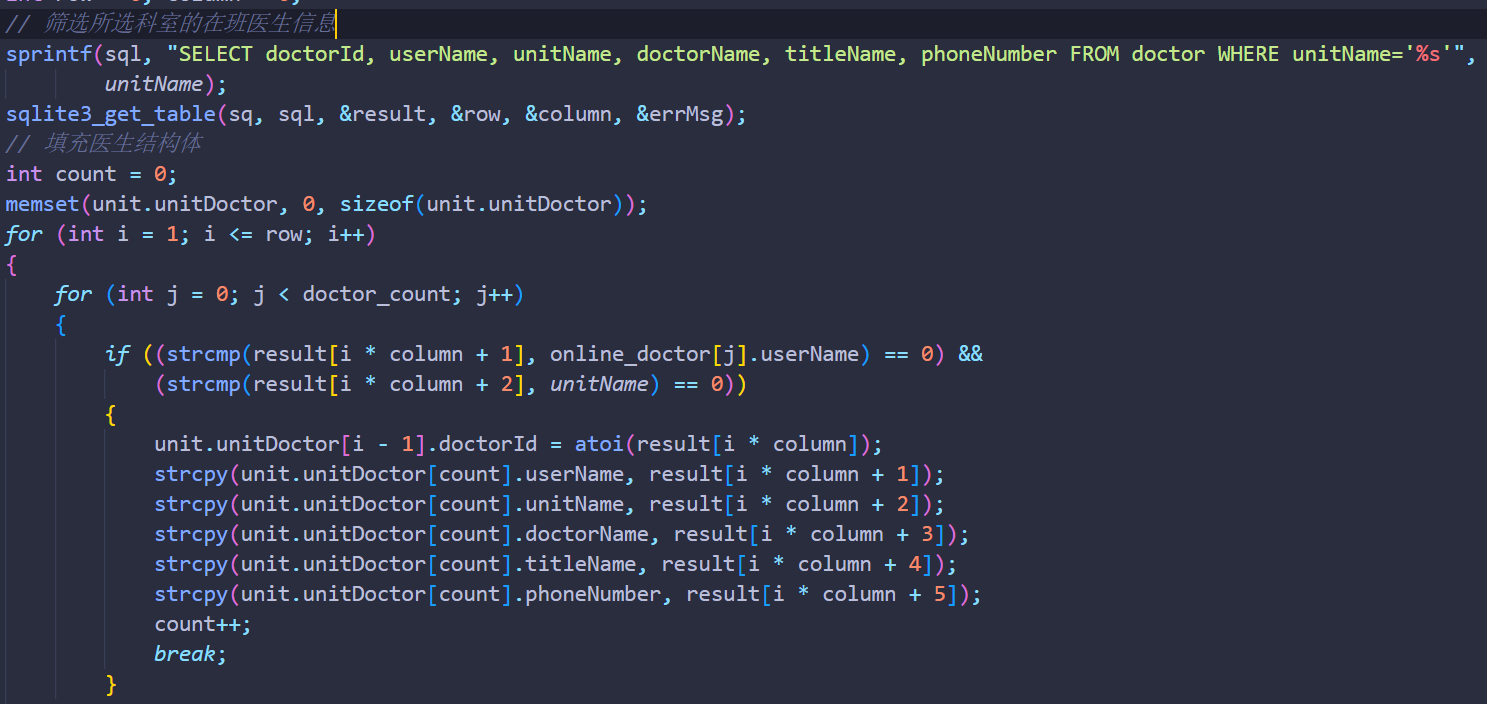
**5.管理员登录，管理员内置账号密码**

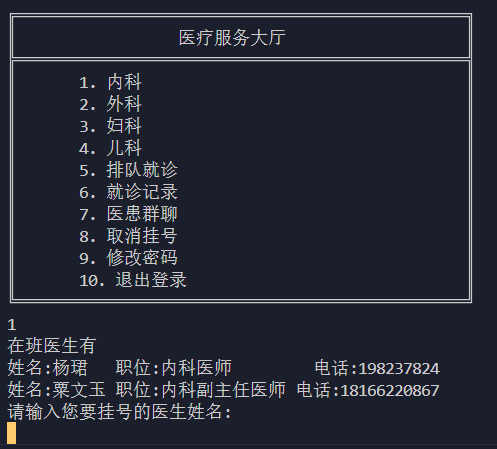
****

* **挂号模块**

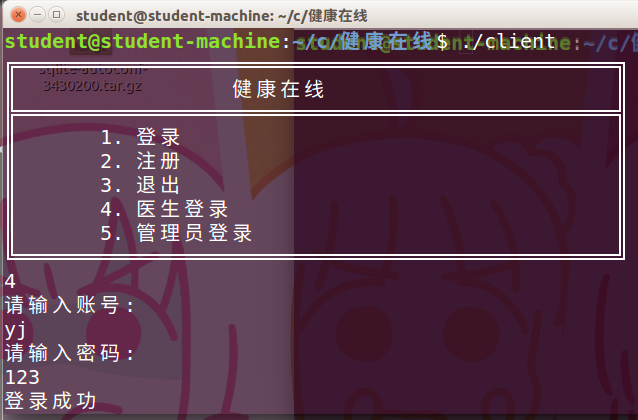
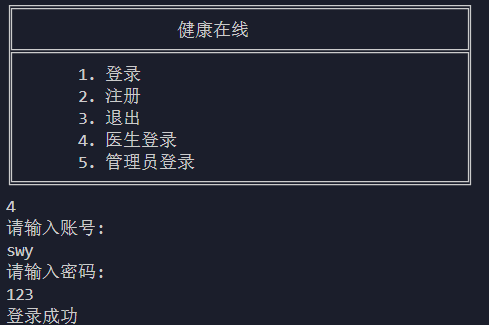
**1.病人登陆成功后输入科室选项，选择科室**

**2. 服务器返回该科室在线的医生**

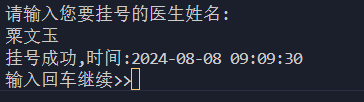
****

****

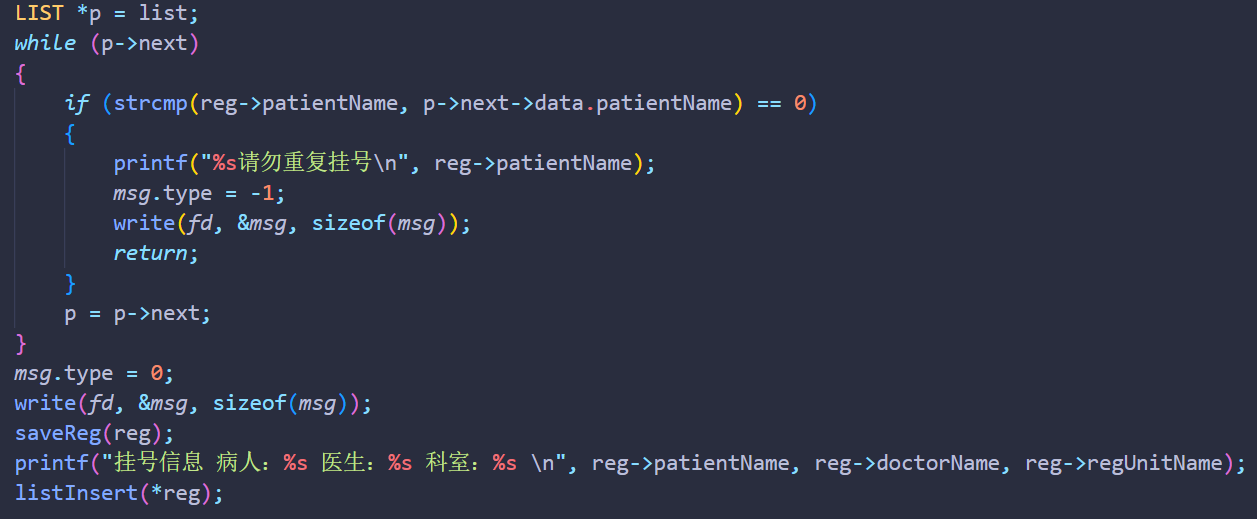
下面是两个在线的医生

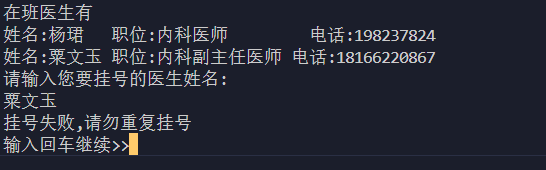
****

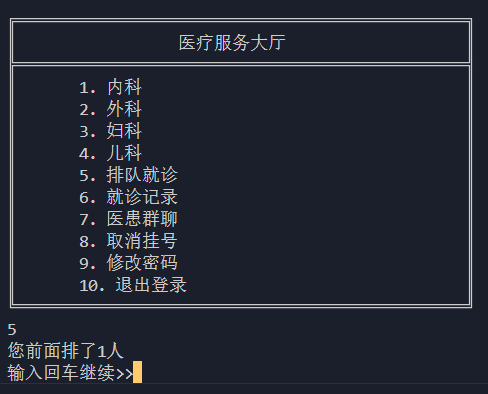
**3. 病人选择医生挂号**

****

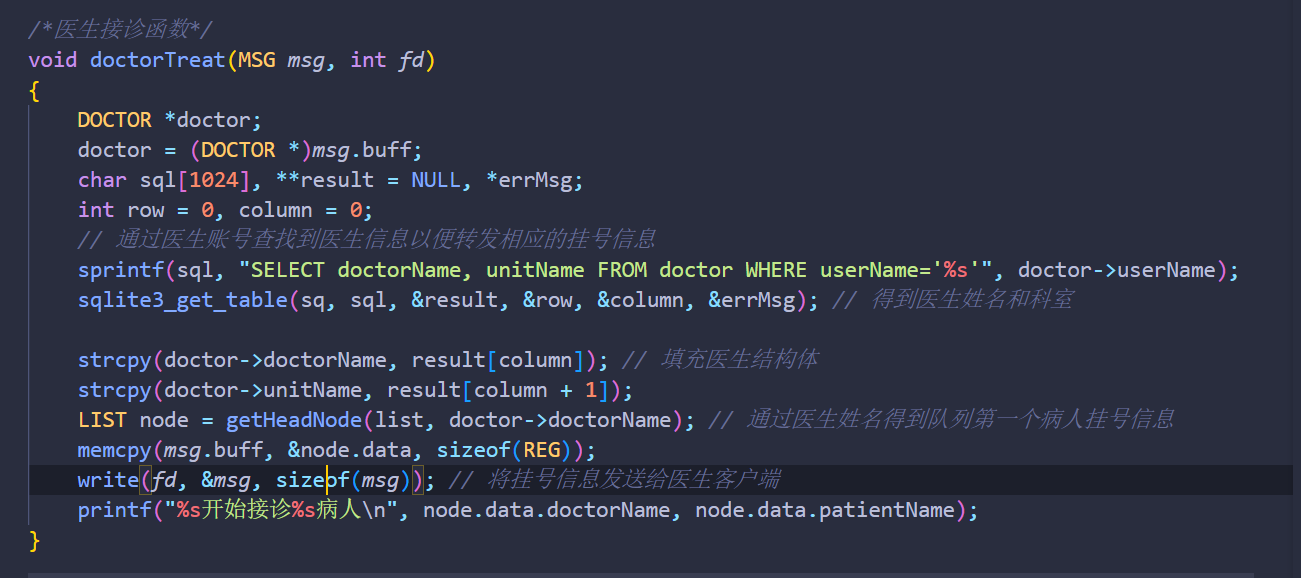
**4. 服务器判断消息是否重复，不是则将挂号信息插入消息队列，返回一个确实信息msg.type = 0,至此病人可以开始排队就诊或做其他事情**

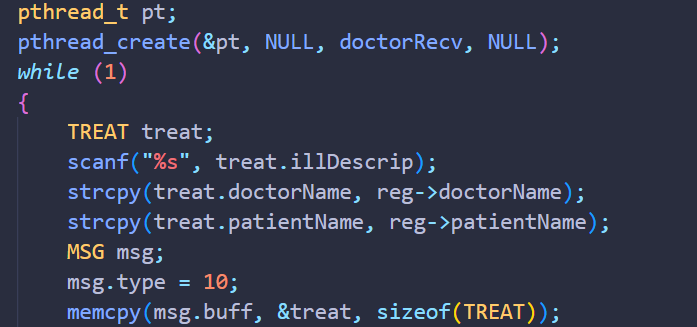
****

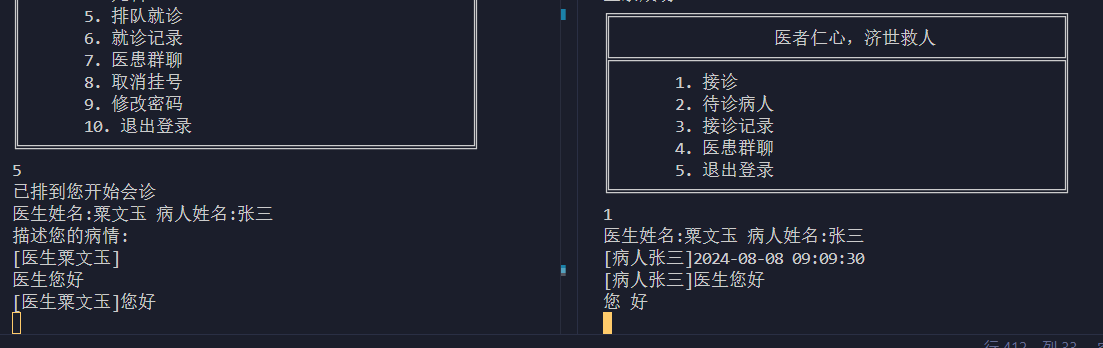
****

****

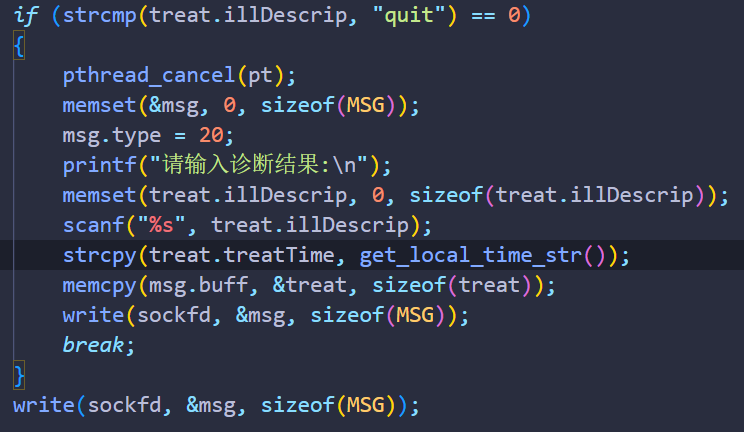
* **接诊模块**
  1. **医生登录选择接诊，服务器返回一个挂号信息**

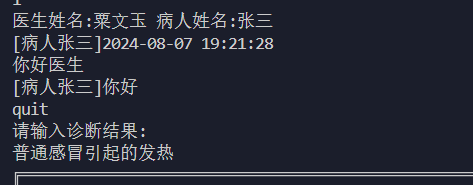
**2.收到挂号信息，与排队病人问诊，创建接收消息线程**

****

****

**3.结束会诊,输入quit结束会诊，输入诊断结果，诊断结果发送给服务器，服务器保存到数据库**

****

****

**4.病人也可输入quit结束会诊，在查看诊断记录选项中可以显示自己的诊断结果，由服务器从数据库筛出，采用循环接收，直到收到结束信号”over”**

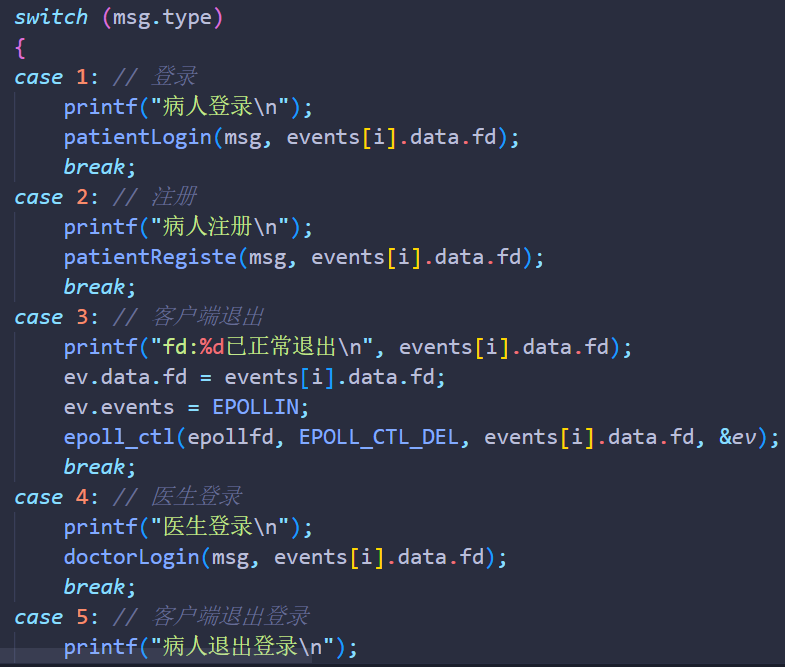
****

****

* **通信模块**

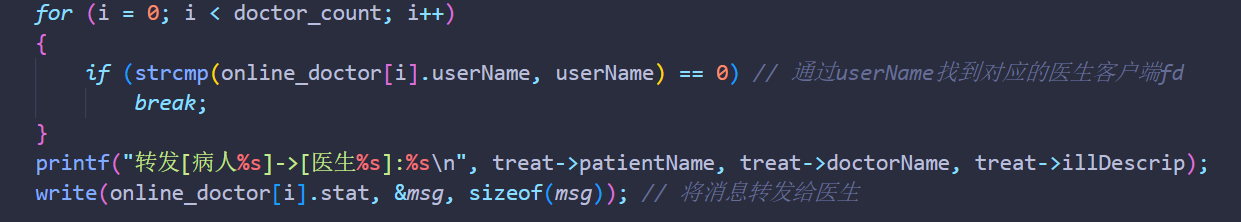
**1.服务器接收信号**

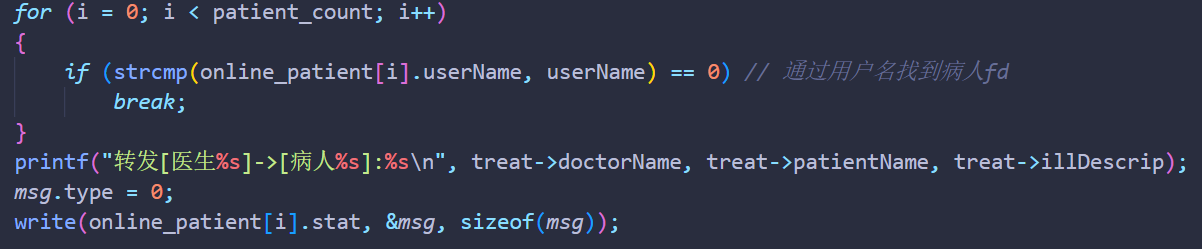
**采用以数字编号对应某一事件的形式来告知服务器客户端要干什么。**

****

**1.医患在就诊时的私聊**

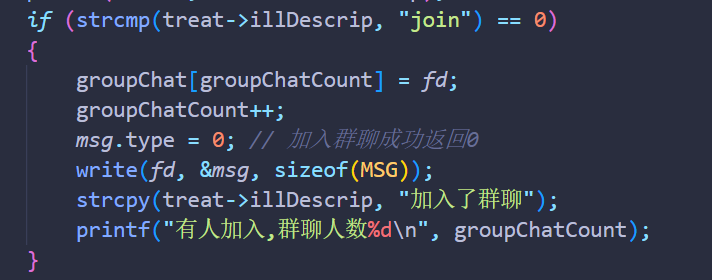
**利用登录时记录在数组的医患账号和通信套接字，在私发时通过医患名字做索引即可找到指定医患通信套接字，从而发送给指定的医患**

****

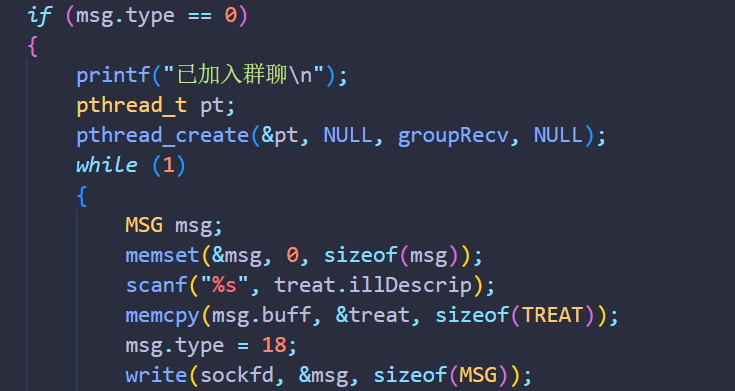
****

**2.医患群聊(医生和患者的大群聊)**

**当医生或患者选择医患群聊，向服务器发送一个“join”信号服务器收到信息将fd记录在群聊数组**

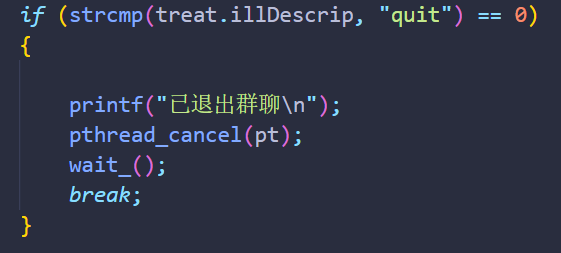
****

**收到加群成功的信号msg.type = 0后，客户端创建线程进入循环开始聊天**

****

****

**输入”quit”即可退出聊天**

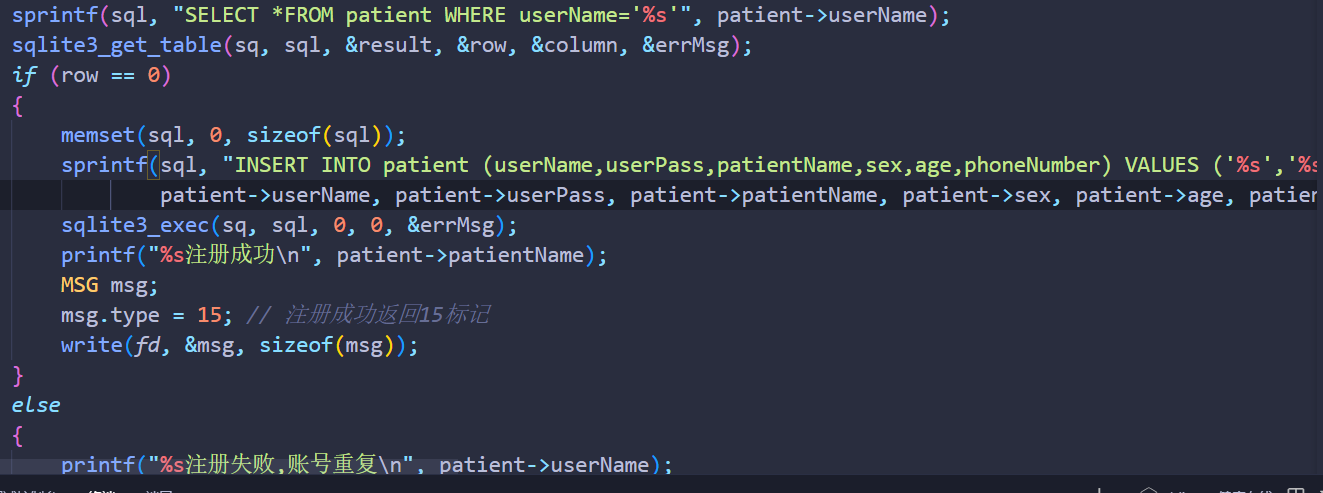
****

**将退出信号发送服务器，服务器清除fd**

****

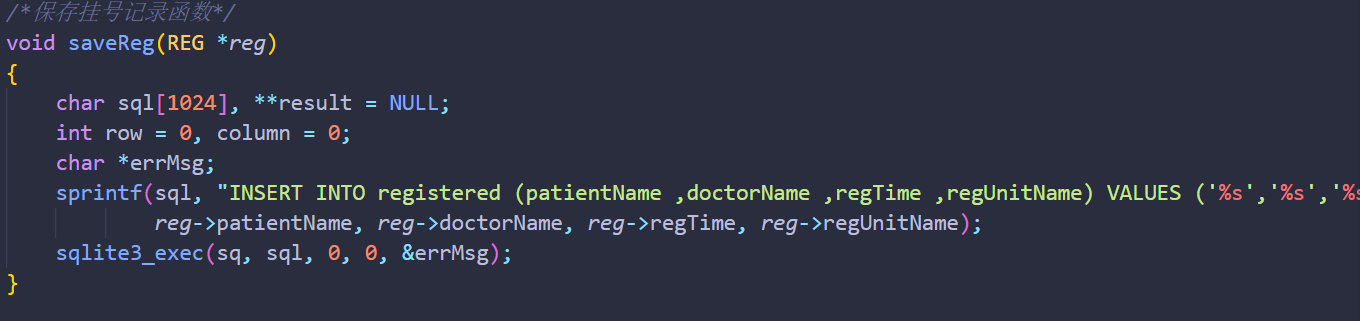
* **数据存储模块**

**1.注册时对病人信息进行存储**

****

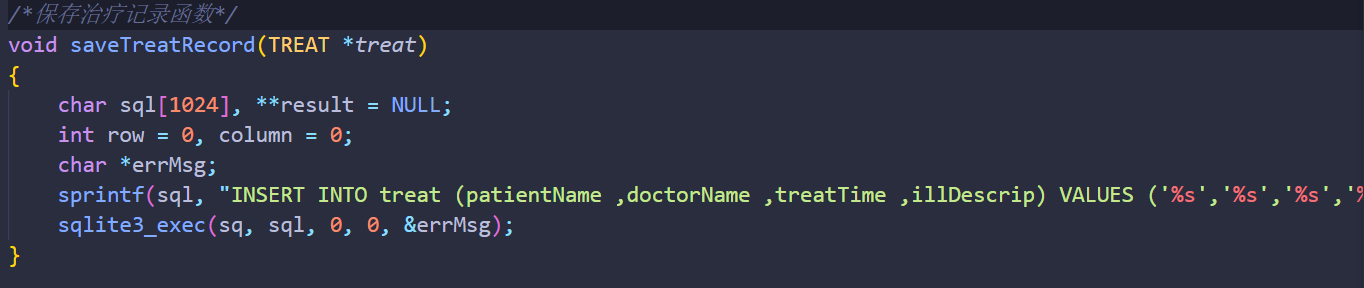
****

**2.挂号信息存储**

****

****

**3.诊断信息存储**

****

****

* **管理员模块**

**1.查看病人信息**

**2.查看医生信息**

**3.查看就诊记录**

**4.修改病人信息**

**5.修改医生信息**

**6.添加医生**

**7.解雇医生**

3. 技术栈

**3.1 编程语言**

* **C语言**

**3.2 网络协议**

* **TCP/IP协议**

**3.3 并发模型**

* **epoll模型**

**3.4 系统编程**

* **进程、线程管理**
* **信号处理**

**3.5 网络编程**

* **Socket编程**

**3.6 数据结构**

* **链队**

**3.7 文件IO**

* **文件读写**

4. 运行过程

4.1 程序启动

1. **初始化服务器和客户端连接**
   * **客户端启动，创建Socket连接到服务器。**
   * **服务器启动，监听客户端连接请求。**

4.2 病人注册

1. **病人输入个人信息**
   * **病人输入姓名、性别、年龄、联系方式等。**
2. **服务器验证信息并保存到数据库**
   * **服务器验证输入信息，将病人信息保存到数据库。**

4.3 病人登录

1. **病人输入账号密码**
   * **病人输入账号和密码。**
2. **服务器验证账号密码，返回登录结果**
   * **服务器验证账号密码，返回登录成功或失败信息。**

4.4 医生端登录

1. **医生输入账号密码**
   * **医生输入账号和密码。**
2. **服务器验证账号密码，返回登录结果**
   * **服务器验证账号密码，返回登录成功或失败信息。**

4.5 病人选择科室

1. **病人选择科室类型**
   * **病人选择内科、外科、妇科、儿科等。**
2. **向服务器发送选择科室的信号**
   * **客户端向服务器发送选择科室的请求。**
3. **服务器返回当前在线的医生信息**
   * **服务器返回当前在线的医生列表。**

4.6 选择医生挂号

1. **病人选择一个医生**
   * **病人从医生列表中选择一个医生。**
2. **向服务器发送挂号请求**
   * **客户端向服务器发送挂号请求。**
3. **服务器将挂号信息加入消息队列**
   * **服务器将挂号信息加入消息队列，等待医生接诊。**

4.7 开始排队

1. **病人进入排队状态**
   * **病人进入排队状态，等待医生接诊。**
2. **等待医生接诊**
   * **病人等待医生从消息队列中取出挂号信息。**

4.8 医生开始接诊

1. **医生从消息队列中取出挂号信息**
   * **医生从消息队列中取出挂号信息。**
2. **开始问诊**
   * **医生开始问诊，与病人进行通信。**

4.9 两端通过服务器进行私聊

1. **病人发送消息**
   * **病人发送消息给医生。**
2. **医生接收消息**
   * **医生接收病人发送的消息。**
3. **医生发送消息**
   * **医生发送消息给病人。**
4. **病人接收消息**
   * **病人接收医生发送的消息。**

4.10 聊天结束

1. **医生输入诊断结果**
   * **医生输入诊断结果。**
2. **将诊断结果保存在数据库**
   * **服务器将诊断结果保存在数据库中。**

4.11 医生查看接诊记录

1. **从数据库读取接诊记录**
   * **医生从数据库中读取接诊记录。**
2. **显示接诊记录**
   * **服务器将接诊记录返回给医生，显示在客户端。**

4.12 医生查看待诊病人

1. **从消息队列读取待诊病人信息**
   * **医生从消息队列中读取待诊病人信息。**
2. **显示待诊病人信息**
   * **服务器将待诊病人信息返回给医生，显示在客户端。**

4.13 病人查看自己诊断记录

1. **向服务器发送请求**
   * **病人向服务器发送查看诊断记录的请求。**
2. **服务器返回诊断记录**
   * **服务器从数据库中读取病人的诊断记录，返回给病人。**