

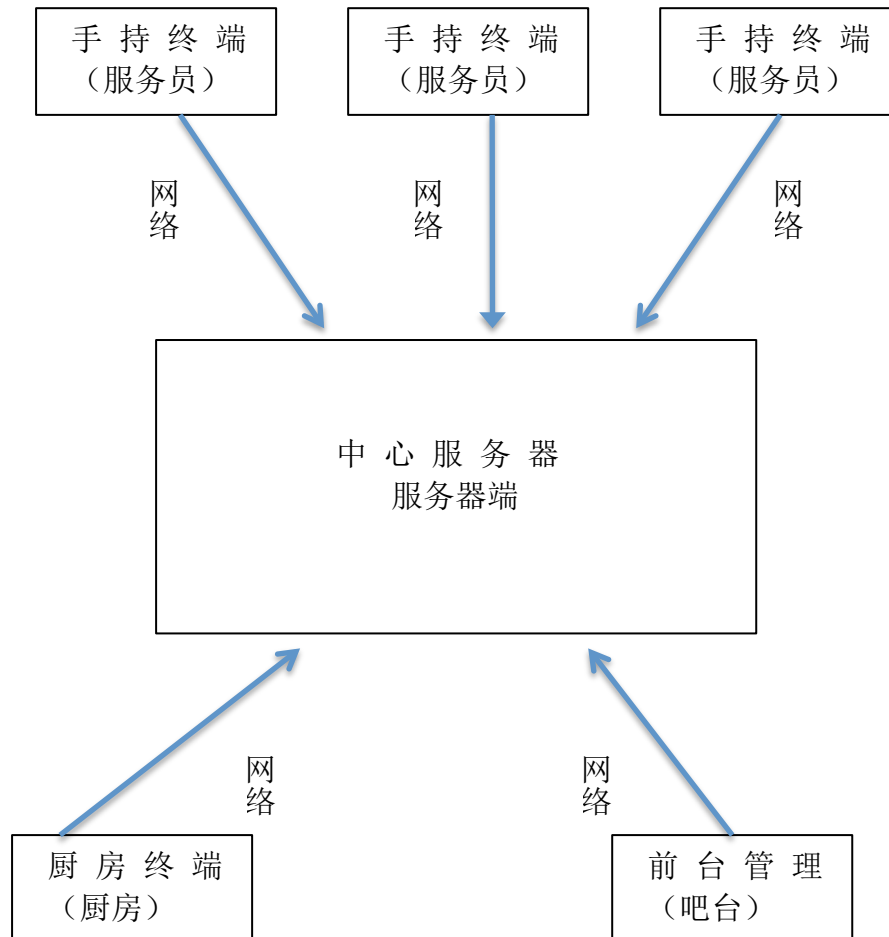
酒店综合管理系统

1， 平台开发和环境简介

linux + gcc + gdb + vim

2， 功能描述

实现酒店的订餐自动化管理，包含服务器端，前台管理客户端，手持终端，厨房客户端。中心服务器负责存储菜品的原材料、酒水库存，库存不足时，及时提醒；职员考勤、薪水等福利数据；客户会员资料，历史消费，点餐菜单数据的保存，及时将手持终端所点菜单交付给厨房客户端，前台管理客户端完成客户消费完毕后的买单清算，会员办理等，最终完成订餐的自动化管理；



手持终端：通过 TCP 连接到中心服务器，然后通过账号密码的形式登陆到中心服务器，登陆成功后，可以直接进行每一桌客户的点单，点餐完成后，将每桌菜单发送给服务器。

厨房终端：通过 TCP 连接到中心服务器，账号密码的形式登陆到服务器，登

陆后，等待服务器发送的消费者每一桌所点菜单，并打印出所点菜单，每桌消费者菜单有三种状态，等待、准备中、完成。并可查看所有菜品点单。

前台客户端：通过默认的超级管理员的账号密码，登陆服务器后，可以添加、查看员工的登陆账号、密码、员工资料、薪水、福利待遇等功能，添加、查看菜品的原材料库存，添加菜品。添加、查看会员卡信息，结算消费者的消费金额。

服务器：服务器启动之后，完成员工的登陆验证，菜品原材料、菜品的存贮；消费者会员卡信息的存贮；有消费者点单之后，应将消费者所在的桌号及对应的点菜传送给厨房端；消费者买单时，根据会员积分享有折扣，形成消费账单、对应积分的存贮。服务器安装了数据库，数据库采用 **MYSQL**，所有数据存储在 **MYSQL** 中。

3， 设计实施

一、配置文件读取

- 1，服务端程序启动后读取配置文件，配置文件中包含要监听使用的 IP、端口等信息。
- 2，配置文件格式可以参考/etc/目录下的各种服务的配置文件。
- 3，配置文件一般规则：
 - 1) 在配置文件中每一行是作为一条配置；
 - 2)、每条配置有两个项：配置项名称（配置变量名）、配置项值；
 - 3)、 “#” 之后的内容为注释；
 - 4)、配置文件命名：xxx.conf

二、Sock 编程

根据配置文件信息启动服务端程序，监听端口，等待客户端连接。完成客户端于服务端简单的 **tcp** 连接。使用 I/O 复用机制完成客户端与服务端之间的一对多的连接。服务端记录每个客户端的基本信息：每个客户端的 IP、端口、账号、套接字等基本信息。使用链表记录保存这些信息。

所有客户端发送数据采用协议的形式，协议包含两部分：包头+数据；
包头格式可参考如下：

```
struct head
{
    char version[20]; //软件版本
    char protocol[20]; //协议版本
    short type; //数据类型
    short dataopt; //数据操作
    int length; //数据长度
    char data[0];
};
```

数据类型可分为：会员卡；员工；菜品；原材料；消费；
可参考如下：

```
#define MEMBER 1 //会员
#define EMPLOYEE 2 //员工
#define FOOD 3 //菜品
#define MATERIAL 4 //原材料
#define CUNSUME 5 //消费
```

数据操作可分为：增加、删除、查看、修改，保存；
可参考如下：

```
#define ADD 10 //增加
#define DEL 20 //删除
#define SHOW 30 //查看
#define MODIFY 40 //修改
#define SAVE 50 //保存
```

注意：当数据类型为消费时，数据操作只有增加、删除、查看、保存，当消费完成，买单后的数据操作为保存。

客户端送法数据时，在数据前面加上包头，进行封装，一次发送给服务器，服务器分为两次读取数据，第一次读取根据包头的数据类型，选择对应的数据结构进行第二次读取数据；数据读取完毕之后根据数据的操作类型来对服务器上的数据进行增加，删除，查看，修改等；

三、功能要求

1，手持终端（模拟）

1)，完成账号密码的验证；

2)，实现菜品的分类，比如：凉菜，素菜，干锅，铁板，招牌菜，汤类，酒水....等等；

3)，完成点菜，所点的每一个菜形成节点建立链表，发送给服务器，其结构可参考如下：

```
struct selectfood
{
    short tablenum; //桌号
    short type; //菜的类型
    char name[50]; //菜的名称
    short number; //菜的编号
    short count; //单个菜的数量
    float price; //菜的价格
    float discount; //折扣
    .....
};
```

4), 可查看所点菜单, 总消费金额

2, 厨房客户端 (pc 机器)

- 1) ,将每桌客户所点菜单按桌号打印
- 2), 并设置每桌订单的状态,分三个状态, 初始状态为等待, 其次正在准备中, 第三个已经完成;

3, 吧台客户端 (pc 机器)

- 1) 会员卡模块: 办理会员卡, 会员卡享有一定的消费折扣, 并进行积分;
- 2) 员工模块: 可增加, 删除, 查找, 修改员工信息, 建立员工的薪酬福利待遇管理;
- 3) 菜单模块: 菜品的分类, 添加, 删除, 查找, 修改每一种菜的信息;
- 4) 原材料模块: 可增加, 删除, 查找, 修改可长期存贮的原材料种类, 数量, 价格的数据信息;
- 5) 消费模块: 查看每桌订单, 完成消费金额的结算, 查看、删除历史消费清单;

4, 服务器端

- 1) 实现员工的登陆验证
- 2) 将手持终端所点菜单转发给厨房客户端
- 3) 会员卡, 员工, 菜品, 原材料, 消费清单等数据的保存;

四、数据同步

原材料库存, 会员卡信息, 菜单, 消费者所点订单, 消费情况等数据, 应让服务器与所有客户端随时保持同步, 及时更新。

五、心跳机制

客户端每隔一定时间向服务端发送一个数据包, 该数据包用途主要使用证明客户端存在, 网络没有断开, 客户端程序本身也没有出现任何问题。就像证明自己活着一样。如果客户端每隔 10 秒发送一个心跳包给服务端。如果服务端超过 10 秒钟没有接收到客户端的心跳包, 则认为该客户端已经死亡 (网络断开或者客户端程序出现问题)。

提示：可以在客户端使用 I/O 复用机制，使用 I/O 复用中的超时机制进行定时。

六、项目要求

- 1、采用 C 语言完成代码的编写。
- 2、编写 makefile 管理整个项目。
- 3、编写项目设计书。
- 4、以模块化编写项目代码，按照不同模块组织 .h/.c 文件。
- 5、规范代码格式并添加注释。
- 6、编写测试报告，包括单模块测试，模块间测试。
- 7、编写项目总结，包括项目设计说明、项目中采用的知识点列举、项目中遇到的问题及解决方法等。