F0903701 李自勉 5090379027

大作业最终

【目的】

1.利用所学的知识进行一个中型系统级程序的设计，进一步理解和掌握基本数据结构和算法。
2.通过程序中涉及到的排序、查找、树、Hash 等操作加深对算法、程序设计思路、常用程序设计技巧的理解与掌握，逐步培养系统级的程序开发能力。 强调对模板库的复用和程序性能的分析。

【内容】

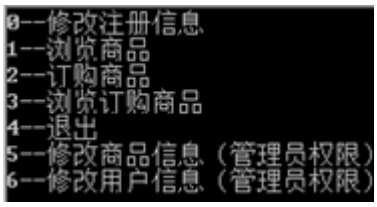
设计一个掌上交易平台，普通用户通过客户端进行商品浏览、订购、退订、评分，具有管理员权限的用户可以增加/删除/修改商品信息。用户属性至少包含如下特性： 用户名，密码，姓名，性别，年龄，地址，电话号码，兴趣爱好等信息。 商品属性至少包含如下特性： 商品名，商品类别，产地，型号，厂家，生产日期，单价，库存数量等信息。 订购信息至少包含如下信息： 用户名，商品名，单价，数量，订购日期，评分。

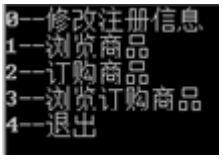
【前端设计】
1.概述

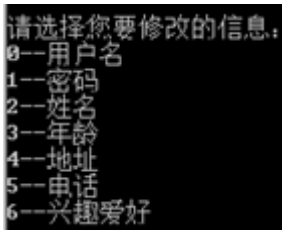
平台内的用户分为管理员和用户两类。管理员不能通过注册获得，只能由管理员来添加，系统内置一个管理员账号作为初始的账号。用户则可以随便注册。两种账户的区别在于操作权限上，管理员可以实现用户所有的操作，并且单独拥有对商品以及用户的增删以及修改操作的权限。

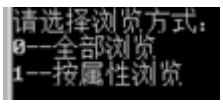
用户、商品、订购商品分别拥有自己的属性。用户属性包含用户名，密码，姓名，性别，年龄，地址，电话号码，兴趣爱好。 商品属性包含商品名，商品类别，产地，型号，厂家，生产日期，单价，库存数量。 订购信息包含用户名，商品名，单价，数量，订购日期，评分。

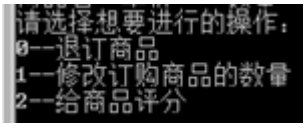
2.菜单设计

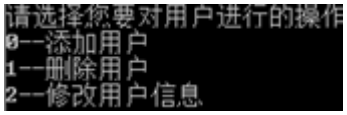












3.数据结构

第一阶段时为了方便前端设计，数据使用 list 的方式存在内存中的，通过 list 指针进行搜索查找。

【数据存储】

1.数据类型

Hash 表

```
class HashTable
{
public:
    enum KindofCell {Active,Empty,Delete};
    //标志Hash表中单元的状态。Active：保存结点。Empty：单元为空。Deleted：保存的结点已被删除
    HashTable();//构造
    ~HashTable(){delete[] Arr;}//析构
    HashTable & operator = ( HashTable & R);
    int Insert ( string &X);//插入X
    int Remove( string &X);//删除X
    int Find ( string &X);
    //查找成功，返回结点数据值。可用函数WasFound确定查找是否成功
    int IsFound ( string &X) ;//测试X是否存在，存在返回，否则为
    int WasFound () ;//最近一次查找成功则返回
    int IsEmpty () ; //Hash表为空，返回，否则为
    void MakeEmpty ();//清空Hash表
    unsigned int FindPos (string &X);
    //查找值为X的结点在散列表中的单元地址
private:
    struct HashCell
    {
        string Element;//保存结点数据值
        KindofCell info;//标志单元状态
        HashCell ():info(Empty){}
        HashCell (string &E,KindofCell i = Empty):Element(E),info(i){}
    };
    enum { DefaultSize = 1000000};
    unsigned int ArrSize;//散列表用的所在的单元个数
    int CurrentSize;//散列表中当前保存的结点个数
    int LastFindOk;//最后一次查找成功则为
    HashCell *Arr;//保存散列表用的数组
    void AllocateArr ();//创建一个数组
    };
```

2.存储文件

硬盘存储文件分为 idx 和 dat 两种，前者为索引文件，后者为数据文件。
索引文件内索引的位置按照 Hash 表中的位置进行存储，中间空闲位置由空格补充，
存储结构为 关键字：数据偏移量：数据长度。
数据文件则按照每种不同的数据按不同的结构存储，但总体上是一行一个数据。

3.算法描述

所有操作均建立在 Hash 表基础上。
插入操作，就是在索引文件里按照 Hash 的散列表添加一条索引，再在数据文件的文件末添加一条数据。
修改操作，首先是根据索引文件找到数据的位置，再找到数据文件内的位置直接覆盖。
删除操作，直接删除掉索引，对数据不做操作。

3.性能测试

对插入操作进行测试

数据条数	数据大小	操作时间
2000	5MB	1.342s
20000	5MB	9.903s
200000	20MB	113.974s

有数据可见，当插入数据条数达到 20000 条，数据大小大约为 20MB 时，操作时间已达近 2 分钟，耗时较长，所以没进一步测试。

【设计失误】

1.对于 Hash 表的实现上采用了线性 Hash 表，而没采用链式，所以对于关键词相同的项进行查找时只能进行遍历，效率十分低下。
2.索引文件占用空间较大，在数据量较小时，索引文件比数据文件大很多。