**程序设计与实践**

**实验报告**

**实验二：个人资金账户管理**

**实验时间：2022.03.14 ~ 2022.03.22**

**一：实验名称：个人资金账户管理**

**二：实验环境: 一台安装有C语言编译器的电脑。**

**三：实验目的**

（1）培养和锻炼对具有一定复杂度和规模的问题的分析与求解能力。

（2）掌握程序设计的综合方法，能综合应用各种数据类型实现较复杂数据的存储。

（3）培养良好的C程序设计风格与代码规范意识。

**四：实验内容**

（1） 资金账户的信息统一放在随机文件中，该随机文件包括的数据项有记录ID、发生日期、发生事件、发生金额（正的表示收入，负的表示支出）和金额。每发生一笔收支，文件要增加一条记录，并计算一次余额。

（2） 程序实现3个功能，包括：(1)，能创建资金账户文件并添加收入或支出信息记录；(2)，能显示所有记录，得知资金账户的收支流水账；(3)，能查询最后一条记录，获知账户最后的余额。账户文件名命名为cashbox.dat，文件部分内容如下;

（3） 增加**修改资金账户**的功能。输入一个记录ID，如果文件中已存在该记录，则输入新的记录信息并更新资金账户文件中相应记录的信息。要求定蔷义和调用函数UpdateLog()，其功能是修改资金账户记录。

LongID CreateDateNote Charge Balance

1 2006-06-01 alimony 500.00 500.00

2 2006-06-08 shopping -300.00 200.00

3 2006-06-15 shopping -60.00 140.00

4 2006-06-20 workingpay 200.00 340.00

5 2006-08-01 scholarship 1000.00 1340.00

**五：实验步骤**

1. 创建资金账户文件并添加收入或支出信息记录

struct LogData log,lastlog;

long logcount;

printf("Input logdate(format:2022-03-14):");

scanf("%s",log.logdate);

printf("Input lognote:");

scanf("%s",log.lognote);

printf("Input Charge:Income+and expend-:");

scanf("%lf",&log.charge);

logcount=getLogcount(cfptr); /\*获取记录数\*/

if(logcount>0){

fseek(cfptr,size\*(logcount-1),SEEK\_SET);

fread(&lastlog,size,1,cfptr); /\*读入最后记录\*/

log.logid=lastlog.logid+1; /\*记录号按顺序是上次的+1\*/

log.balance=log.charge+lastlog.balance;

}

else{ /\*如果文件是初始状态，记录数为0\*/

log.logid=1;

log.balance=log.charge;

}

rewind(cfptr);

printf("logid=%ld\n",log.logid);

fwrite(&log,sizeof(struct LogData),1,cfptr); /\*写入记录\*/

fclose(cfptr);

}



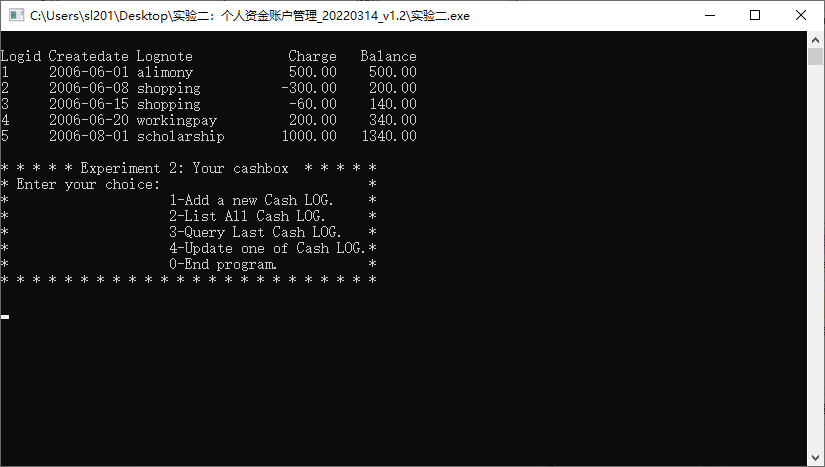
1. 显示所有记录，得知资金账户的收支流水账

struct LogData log;

fseek(cfptr,0L,SEEK\_SET); /\*定位指针到文件开始位置\*/

fread(&log,size,1,cfptr);

printf("\nLogid Createdate Lognote Charge Balance\n");



1. 查询最后一条记录，获知账户最后的余额

struct LogData log;

long logcount;

logcount=getLogcount(cfptr);

if(logcount>0) /\*表示有记录存在\*/

{

fseek(cfptr,size\*(logcount-1),SEEK\_SET); /\*定位最后记录\*/

fread(&log,size,1,cfptr); /\*读取最后记录\*/

printf("\nThe last log is:\n");

printf("\nlogid:%-6ld\nlogdate:%-11s\nlognote:%-15s\n",

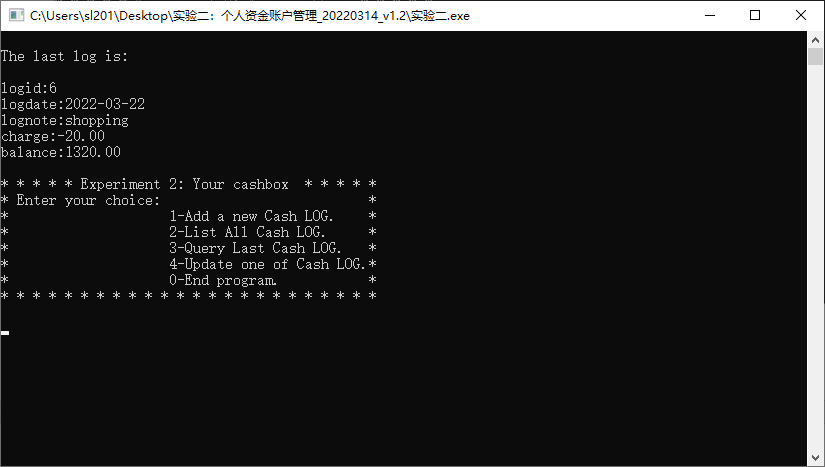
log.logid,log.logdate,log.lognote);

printf("charge:%-10.2lf\nbalance:%-10.2lf\n",

log.charge,log.balance); /\*显示最后记录内容\*/

}

else printf("no logs in file!\n");



1. 输入新的记录信息并更新资金账户文件中相应记录的信息

使用结构体数组进行一一对应，更新数据，重点部分详细如下。

logcount=getLogcount(cfptr); /\*获取记录数\*/

rewind(cfptr);

fread(&logx+1,size,logcount,cfptr); //定位指针需要+1，否则log[i].logid会从2开始

for(i=0;i<=logcount;i++){

if(id==log[i].logid){ //找到id

……

flag=i; //使用标记flag记录位置i

break;

}

}

if(flag!=-1){ //没有找到flag=-1

……(输入数据)

strcpy(log[flag].logdate,update.logdate); //直接复制输入的数据

strcpy(log[flag].lognote,update.lognote); //直接复制输入的数据

if(log[flag].charge!=update.charge){ //对比一下收支是否有变化，没用变化可以跳过计算和循环

update.balance=log[flag].balance-log[flag].charge+update.charge;

//新的余额=原余额-原收支+新收支

log[flag].charge=update.charge; //更新收支

log[flag].balance=update.balance; //更新余额

for(i=flag+1;i<=logcount;i++){ //循环从flag后一个开始

log[i].balance=log[i-1].balance+log[i].charge; //重新计算余额

}

}

fwrite(&logx+1,size,logcount,cfptr); //写入cashbox.bat

……(输出数据)

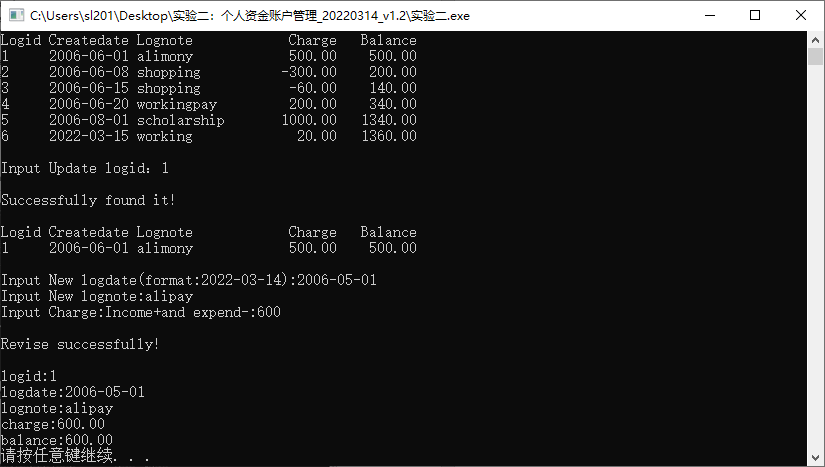
}else{

rewind(cfptr);

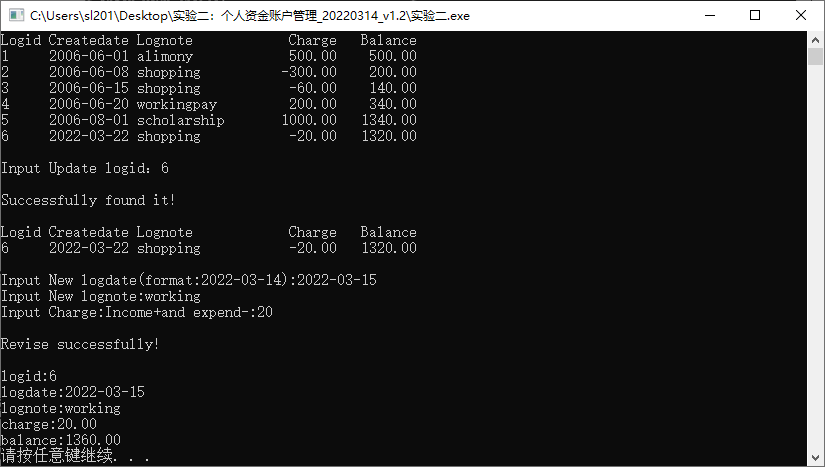
printf("\nNot Found! Back to the menu…");

}

i.需要更新的id=1



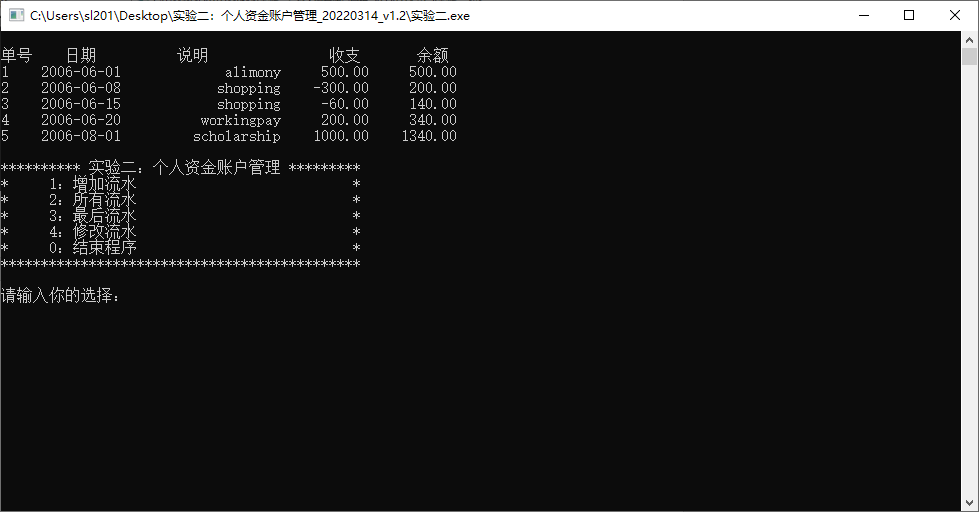
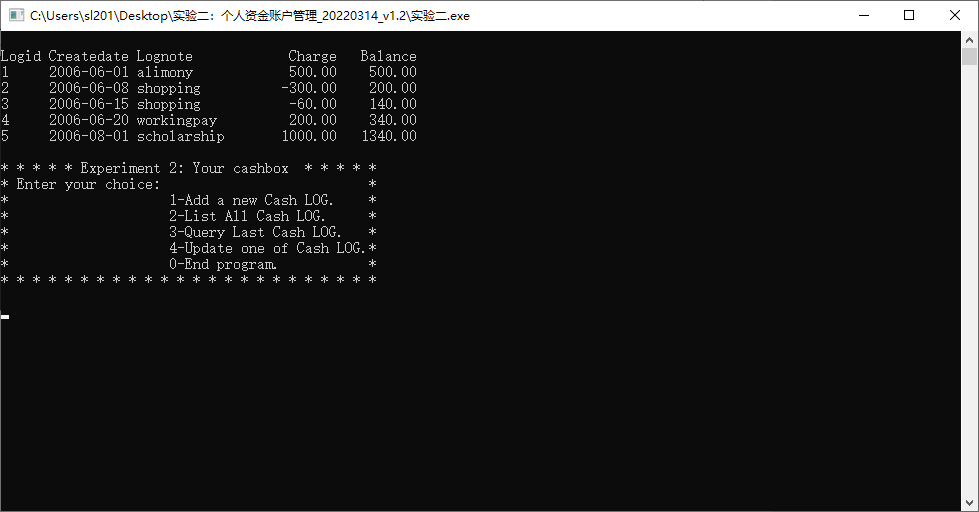
ii.需要更新的id>1



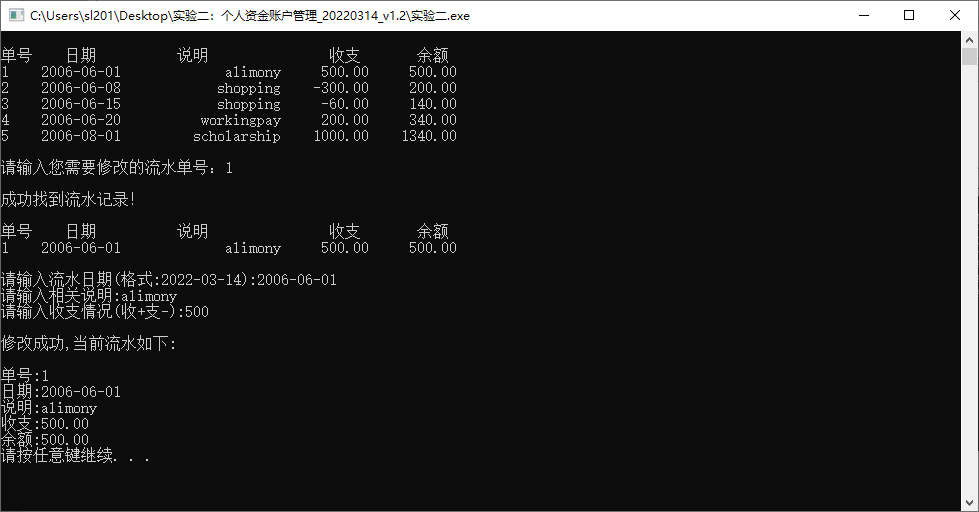
1. 菜单与界面

方便使用，保留原有程序中的英文界面，增加和更新中文翻译界面，可自由选择语言。



其他中文界面（输入数据仅为演示界面使用）



char language[20];

int mychoice,la=1;

printf("Please choose the language for your cashbox: EN or CN(default)\n");

printf("请选择您需要的语言界面：简体中文(默认) 或 英语\n");

scanf("%s",language);

do{

if(strcmp(language,"EN")==0||strcmp(language,"英语")==0||strcmp(language,"英文")==0||strcmp (language,"English")==0){

la=0;

……

}else{

……

}

**六：实验小结**

完成情况：基本完成实验内容，新增Updatelog去对已有数据进行更新替换。

重点：fread(&logx+1,size,logcount,cfptr); //定位指针需要+1，否则log[i].logid会从2开始

难点：对于结构体的数据处理，注意结构体中的拼写、输入是否有错。

有待改进的地方：去尝试增加删除已输入的数据，更全面完善功能。

收获：复习和学习文件的运用和相关函数，进一步加深对指针的使用和印象，加强自我对结构体+数组的思维拓展。

体会：面对不熟悉知识花了较长的时间完成实验，希望下次能够再早一点完成。

**七：参考资料**：C语言程序设计（第4版），何钦铭，颜晖主编。