

计算机与信息学院

《数据结构课程设计》

报 告

开课学期： 2024 年 秋季 学期

网选班号： 03 班

姓 名： 林睦昊

学 号： 202310120503

单链表操作演示软件

1. 课题描述

**题目12： 简单的职工管理系统（\*\*）**

**任务：**对单位的职工进行管理，包括插入、删除、查找、排序等功能。

**要求：**

　　职工对象包括姓名、性别、出生年月、工作年月、学历、职务、住址、电话等信息。

（1）新增一名职工：将新增职工对象按姓名以字典方式职工管理文件中。

（2）删除一名职工：从职工管理文件中删除一名职工对象。

（3）查询：从职工管理文件中查询符合某些条件的职工。

（4）修改：检索某个职工对象，对其某些属性进行修改。

（5）排序：按某种需要对职工对象文件进行排序。

**实现提示：**

　　职工对象数不必很多，便于一次读入内存，所有操作不经过内外存交换。

（1）由键盘输入职工对象，以文件方式保存。程序执行时先将文件读入内存。

（2）对职工对象中的"姓名"按字典顺序进行排序。

（3）对排序后的职工对象进行增、删、查询、修改等操作。

2、需求分析

（根据老师提出的要求，你具体准备实现哪些功能，这些功能的输入输出是什么）

本题目总共设计了三种不同的账号类型：

分别为：管理员账户；普通职工账户；政府部门或其他统计部门需要查看本公司员工信息的有关部门账户

不同的账户类型有着不同的操作权限

1. 管理员账户：拥有着系统的最高权限，可以查找员工信息，删除某一个员工信息，添加一个员工信息，更改员工信息（无法更改名字），对员工信息进行排序等
2. 普通员工账户：可以查询自己的信息，也可以更改自己的信息（但需要管理员授权）
3. 其他账户：仅可以查询员工信息但是需要向管理员索要到密钥才可查询

查询功能可以分为：使用名字进行查询，使用工号进行查询；

2.1 约束

（系统硬件平台、软件平台等）。

系统硬件平台为联想台式机XXX型号，其中CPU为YY, 内存为8G。

系统使用C/C++语言，在Windows 7操作系统下实现。采用命令行的界面方式完成系统人机交互。链表节点为整数数据。

2.2 功能规定

（系统实现的功能规范性描述）

（1）由键盘输入职工对象，以文件方式保存。程序执行时先将文件读入内存。

（2）对职工对象中的"姓名"按字典顺序进行排序。

（3）对排序后的职工对象进行增、删除、查询、修改等操作。

2.2.1 角色划分

管理员，普通员工，政府部门或其他需要查询公司员工信息的有关统计审查部门账户

2.2.2 系统功能

（分管理员，普通员工不同的功能）

管理员：拥有着系统的最高权限，可以查找员工信息，删除某一个员工信息，添加一个员工信息，更改员工信息（无法更改名字），对员工信息进行排序等

普通员工：可以查询自己的信息，也可以更改自己的信息（但需要管理员授权）

政府部门或其他需要查询公司员工信息的有关统计审查部门账户：仅可以查询员工信息但是需要向管理员索要到密钥才可查询

2.3 性能规定

（时间和空间复杂性要求）

3、概要设计

（设计实验，验证所采用的方法能符合系统性能要求。描述软件的每个模块(函数)的参数和功能，模块间的关系）

删除函数：Delete\_同下查询对应

查询函数：Findemployee()

添加函数：addyg()

修改函数：changeemployee()

排序函数：sortemployee()

删除函数：Deletemployee()

判断用户是否存在：getemployeename(char \* name)

功能菜单函数用于加载主功能菜单：menu()

程序的管理员可以对数据进行删除员工数据、新舔员工数据、查询员工数据、修改员工资料、对员工进行排序等操作

普通员工账户仅仅可以查询和修改数据

登录注册功能

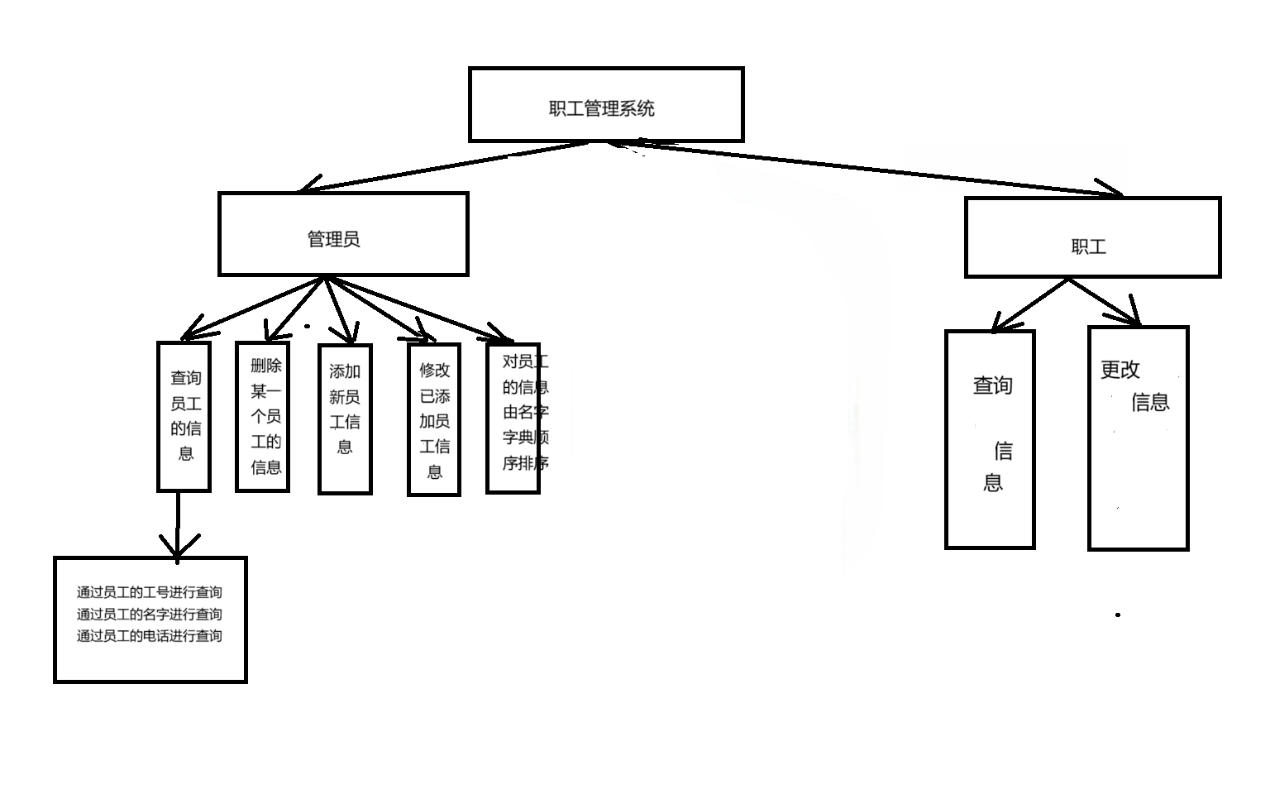
读取数据readwenjian()

保存数据savewenjian()

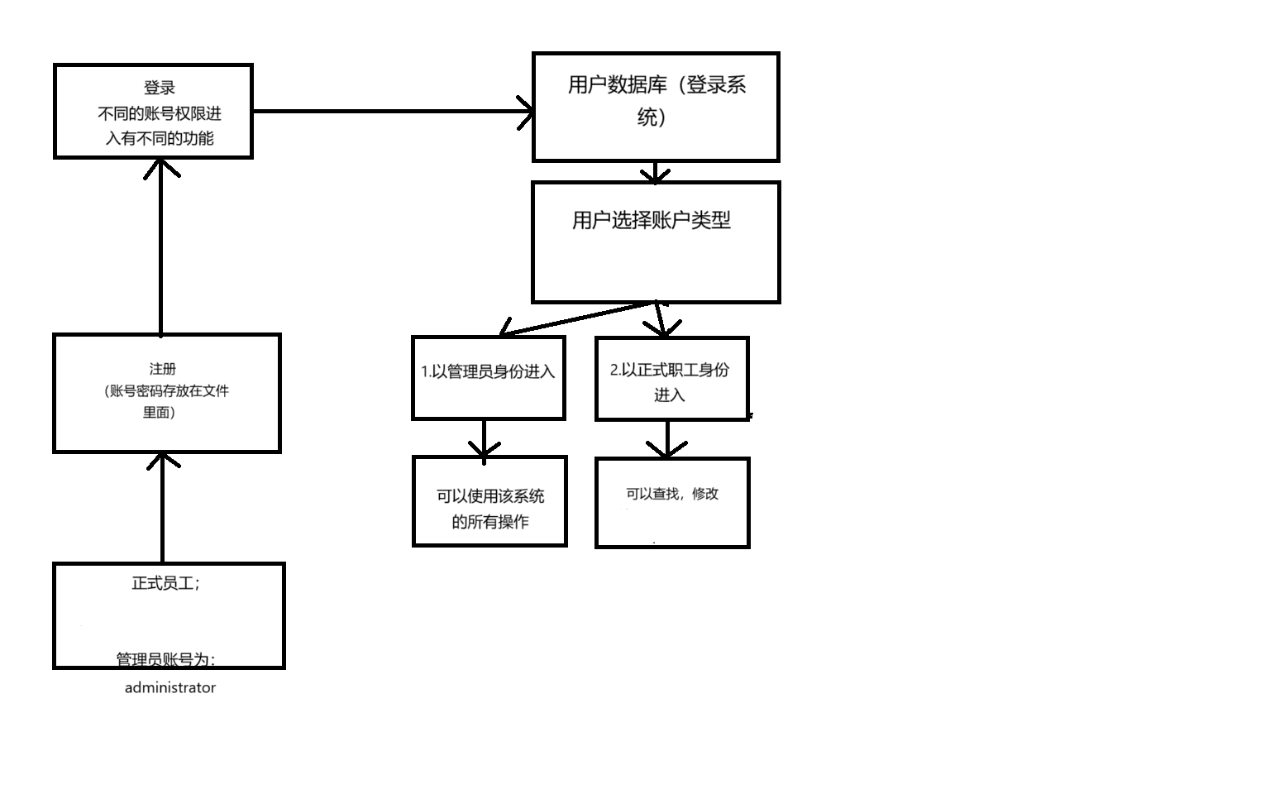
注册int registerusers()返回值用于判断是否登录注册成功

登录int userlogin()返回值同上

主要功能设计图



主要登录功能设计图



3.1 实验与分析

（设计时间和空间的实验，并进行复杂性分析，得出正确的结论）

硬件平台内存为8G，系统软件使用1G,空余内存为可以满足性能规定1.链表查询的理论时间复杂性是T=O(n), 跟链表长度有关系，但具体的计时时间，受硬件平台限制，因此需通过实验进行预先分析。

（1）实验设计与结果

为进一步测试系统性能，换一台计算机，硬件设备为XXX,重复以上实验，并将平均时间列入表1。

表1 在不同的硬件平台上的平均查找时间对比

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 机器 | 查找 | 2K | 3K | 5K | 7K |
| 联想台式机 | 不成功查找 | 15 | 50 | 135 | 265 |
| 成功查找 | 10 | 25 | 75 | 120 |
| 华为笔记本 | 不成功查找 | 38 | 75 | 205 | 306 |
| 成功查找 | 20 | 46 | 98 | 156 |

……

（2）分析与结论

（**分析非常重要，分析是这部分重点，分数集中在这里**）

struct Employee{

    char name[50];//姓名

    char gender[10];//性别

    char phone[11];//电话11位

    double money;//工资

    int age;//年龄

    char location[20];//地址

};

为数据结构

采用//员工结构体数组

struct Employee employeesz[100];

来存储信息，测试数量为7000个生成的数据模型，所需时间非常短几乎使用微秒来计算，因此不必采用其他的方式。

查找采用的是遍历的查找方式，遍历员工结构体数组，来进行查找，最坏的情况就是需要遍历整个数组才能查询到员工信息，所以时间复杂度为O(n);

查找有好几种查找模式，可以按照姓名进行查找，也可以按照性别，工资和年龄进行查找。

按照工资和年龄进行查找使用的是模糊查找方法，及需要输入一个区间值，系统会查找出在这个区间范围内的所有的员工的信息。

经过测试在1000000以内该系统的查找算法都不会耗费过多的时间，生成数据也不会耗时过多的时间，这可能与电脑硬件性能有关，测试的三台机器均使用的是i9-13900HX的移动平台的处理器，相比于机房的9代或者是10代i5，性能方便有较大提升，但是对查找时间的影响依旧十分小，误差仅有几十微秒，可能只有当数据超过一定的数量级才可能会出现较长的查找时间。

3.2模块功能与调用关系

本系统的数据结构设计（逻辑结构、存储结构）

（……）

本程序包含12个函数：

(1)void main():主函数包含了用户与系统交互的基本逻辑，用了两个while循环分别来运行管理员和普通员工的操作权限，其中while循环中的循环条件为Flag为全局变量给定当Flag为特定某值时会被系统判定为管理员权限

registerusers函数与loginusers函数的返回值来判断是否注册或者登录成功，如果成功才能进入到while循环中，在循环中使用了switch来与用户进行交互。

(2)int registerusers():这个函数为注册函数其中声明了一个结构体来当作用户的注册账号和密码还有用户名的储存结构，将用户输入的账号和密码储存在文件中方便下次读取，用户输入密码时会自动被系统替换成’\*’以此来达到加密的效果，若用户输入相同账号时会被系统判定为重复注册需要更改，若再次确认密码时出错会由两次修改机会（修改次数可以通过更改registerusers函数中的time变量进行变更，默认为2次）。

(3)int loginusers():为登录函数密码错误会退出程序，需要重新运行，其他类似于上述注册函数。

(4)void menu():菜单函数，这个函数会加载管理员和普通员工的功能菜单。

(5)void getemployeename(char \*name):为判断员工信息是否重复，以员工的姓名为判定条件，若重复返回值>-1，不重复返回值-1;用于下列其他函数的判断。

(6)void addyg():添加员工函数首先系统会判断添加的新名字是否已经存在，若存在会退出添加程序，随后依次添加员工的信息，当一个员工录入完成后会提示是否继续录入，继续会输入1，终止输入2，当输入2时addyg函数中的while循环会自动退出从而结束员工信息录入操作。

(7)void Findemployee():查询函数，在这个函数中会给出可以通过什么什么查询的条件，其次在工资，年龄查询中是按照范围进行查询的，同时在本函数中还拥有输出模块，及程序会根据你所选择的查询方法来对Employee.txt文件中已经用有的员工数据进行查询和输出，同时还会计算工人的平均年龄，平均工资和总查询的人数。

(8)void savewenjian():这是一个保存文件函数，当调用这个函数时系统会将已经储存在内存中的数据同意保存到存在于电脑硬盘储存中的txt文件里面方便下次启动程序的时候进行读取（在main函数中每次调用功能函数后都会默认调用一次savewenjian函数来自动保存员工数据以防电脑死机或者其他不可抗原因导致的数据丢失），同时在用户主动退出程序时也会默认最后一次保存数据。

(9)void readwenjian():为读取函数，会在程序开始时就掉用此函数，来获得储存在文件中的员工的各项数据。

(10)void changeemployee():为修改函数，默认是通过姓名进行修改，在用户输入姓名时系统会事先判定输入的姓名是否已经被添加，如果未被添加，系统会弹出” **未查询到此员工信息请选择添加员工操作**”的提示字样，若名字输入合法，系统会让用户依次修改此员工的信息，修改完成后会自动储存此用户信息到文件之中。

(11)void Deletemployee():删除员工函数，同修改函数一样，系统会事先判断输入名字是否合法，若要删除员工则只需输入员工姓名即可删除。

(12)void sortemployee():这个函数是排序函数，主要采用了一个冒泡排序，对其条件采用正向排序。排序结束后也会保存文件。

4、详细设计

（重点算法的流程图和实现，关键代码处要有注释。）

#include <stdio.h>

#include <stdlib.h>

#include <string.h>

//员工的数据

struct Employee{

    char name[50];//姓名

    char gender[10];//性别

    char phone[11];//电话11位

    double money;//工资

    int age;//年龄

    char location[20];//地址

};

//员工结构体数组

struct Employee employeesz[100];

//员工个数

int employeegs = 0;

//以下是登录系统的制作

//密码与账户的数据储存结构

typedef struct users{

    char name[20];

    char zh[20];

    char mm[20];

}A;

int i=0;

//判断是否为管理员进入系统

int Flag=0;

//注册界面(注册成功返回1，未成功返回0)

int registerusers(){

    A b,a;

    char c[20];

    FILE \*fp;

    printf("=========================================================\n");

    printf("=                     注        册                      =\n");

    printf("=                                                       =\n");

    printf("=========================================================\n");

    //读取文件Users.txt储存在与代码同目录下

    fp = fopen("Users.txt","r");

    fread(&b, sizeof(A), 1, fp);

    printf("请输入账号\n");

    scanf("%s",&a.zh);

    //验证账号是否已经被注册

    while(1)

    {

        if (strcmp(a.zh, b.zh))

        {

            if (!feof(fp))

            {

                fread(&b, sizeof(A), 1, fp);

            }

            else

            {

                break;

            }

        }

        else

        {

            printf("该账号已经被注册\n");

            fclose(fp);

            return 0;

        }

    }

    //time为最大错误密码输入次数（注册）最大可以输入错误两次

    int time = 2;

    //设置密码环节，需要用户输入两次密码，第一次输入密码第二次确定密码

    do

    {

        printf("请设置密码:\n");

        i = 0;

        for (i = 0; i < 20; i++)

        {

            //\_getch不可查看密码

            a.mm[i] = \_getch();

            if (a.mm[i] == '\r')

            {

                break;

            }

            else

            {

                printf("\*");

            }

        }

        a.mm[i] = '\0';

        printf("\n");

        printf("请确认密码:\n");

        i = 0;

        for (i = 0; i < 20; i++)

        {

            c[i] = \_getch();

            if (c[i] == '\r')

            {

                break;

            }

            else

            {

                printf("\*");

            }

        }

        c[i] = '\0';

        printf("\n");

        if (!strcmp( a.mm,c) )

        {

            printf("请输入用户名：");

            scanf("%s",&a.name);

            fp = fopen("Users.txt", "a");

            fwrite(&a,sizeof(A),1,fp);

            printf("%s,注册成功！\n", a.name);

            fclose(fp);

            return 1;

        }else

        {

            if (time != 0)

            {

                printf("密码错误！\n请重新输入密码\n您还有%d次机会\n", time);

                time--;

                continue;

            }

            else

            {

                printf("多次输入错误，即将退出！\n");

                fclose(fp);

                return 0;

            }

        }

        //若机会耗尽退出程序

    }while(time >=0);

}

//登录系统(成功登录返回1密码错误返回0)

int userslogin(){

    A a,b;

    FILE \*fp;

    printf("=========================================================\n");

    printf("=                     登        录                      =\n");

    printf("=                                                       =\n");

    printf("=========================================================\n");

    printf("=输入您的账号：\n");

    scanf("%s",&b.zh);

    printf("=输入您的密码：\n");

    i = 0;

    for (i = 0; i < 20; i++)

    {

        b.mm[i] = \_getch();

        if (b.mm[i] == '\r')

        {

            break;

        }

        else

        {

            printf("\*");

        }

    }

    b.mm[i] = '\0';

    printf("\n");

    fp = fopen("Users.txt", "r");

    while (1)

    {

        fread(&a, sizeof(A), 1, fp);

        if (strcmp(b.zh,a.zh)==0)

        {

            break;

        }

        else

        {

            if (!feof(fp))

            {

                fread(&a, sizeof(A), 1, fp);

            }

            else

            {

                printf("账号或密码错误\n");

                fclose(fp);

                return 0;

            }

        }

    }

    if (strcmp(b.mm, a.mm)==0)

    {

        //发出提示“用户名”欢迎回来

        if(strcmp(a.name,"administrator")==0){

            Flag = 9;

        }

        printf("%s，欢迎回来\n",a.name);

        fclose(fp);

        return 1;

    }

    else

    {

        printf("密码错误！\n");

        fclose(fp);

        return 0;

    }

}

//功能菜单函数

void menu(){

    if(Flag == 9){

        printf("================您以管理员身份进入本系统=================\n");

        printf("=1.添加新员工                                           =\n");

        printf("=2.删除员工                                             =\n");

        printf("=3.进入查询页面                                         =\n");

        printf("=4.对员工进行排序                                       =\n");

        printf("=5.修改员工信息                                         =\n");

        printf("=6.退出本系统                                           =\n");

        printf("=请输入相应的操作序号来操作系统                         =\n");

        printf("=========================================================\n");

    }else if(Flag!=9){

        printf("================您以员工的身份进入本系统=================\n");

        printf("=1.进入查询页面                                         =\n");

        printf("=2.修改你的信息                                         =\n");

        printf("=3.退出本系统                                           =\n");

        printf("=请输入相应的操作序号来操作系统                         =\n");

        printf("=========================================================\n");

    }

}

//判断员工信息是否重复

int  getemployeename(char \*name){

    int i=0;

    int flag=-1;

    //不重复返回flag=-1；

    //重复返回flag>-1；

    //默认不重复

    for(i=0;i<employeegs;i++){

        if(strcmp(name,employeesz[i].name)==0){

            //重复使用姓名,同时返回是哪一个员工的名字与输入新名字重复

            flag = i;

            break;

        }

    }

    return flag;

}

//添加员工

void addyg(){

   int yonflag = 1;

    while(yonflag == 1){

        //输入员工信息

        printf("请输入需要添加的员工姓名\n");

        scanf("%s",employeesz[employeegs].name);

        int nameflag = getemployeename(employeesz[employeegs].name);

        if(nameflag > -1){

            printf("\*\*\*\*名字已经存在\*\*\*\*\n");

            break;

        }

        printf("=--请输入新员工性别\n");

        scanf("%s",employeesz[employeegs].gender);

        printf("=--请输入新员工电话\n");

        scanf("%s",employeesz[employeegs].phone);

        printf("=--请输入新员工的工资\n");

        scanf("%lf",&employeesz[employeegs].money);

        printf("=--请输入新员工的地址\n");

        scanf("%s",employeesz[employeegs].location);

        printf("=--请输入新员工的年龄\n");

        scanf("%d",&employeesz[employeegs].age);

        employeegs++;

        getchar();

        printf("===是否继续录入(继续输入1，终止输入2)：");

        scanf("%d",&yonflag);

    }

}

//查询员工

void Findemployee(){

    printf("================管理员身份进入了查询系统=================\n");

    printf("=1.按照姓名查询                                         =\n");

    printf("=2.按照性别查询                                         =\n");

    printf("=3.按照工资查询                                         =\n");

    printf("=4.按照年龄查询                                         =\n");

    printf("=5.展现全部信息                                         =\n");

    printf("=========================================================\n");

    printf("=请输入您的查询方式:");

    int choice=0;

    //查询条件变量声明

    char cxtj[20];

    double minmoneytj = 0,maxmoneytj = 1;

    int  minage = 0,maxage = 1;

    //以下是计算总人数，平均工资，平均年龄的变量声明

    int count=0;

    double allmoney=0;

    int allage=0;

    //选择查询方式

    scanf("%d",&choice);

    switch (choice) {

    case 1:

        printf("=请输入需要查询员工的姓名:");

        scanf("%s",cxtj);

        break;

    case 2:

        printf("=请输入需要查询员工的性别:");

        scanf("%s",cxtj);

        break;

    case 3:

        printf("=请输入需要查询员工的工资(输入一个范围)\n");

        printf("=MAX:");

        scanf("%lf",&maxmoneytj);

        printf("=MIN:");

        scanf("%lf",&minmoneytj);

        break;

    case 4:

        printf("=请输入需要查询员工的年龄(输入一个范围)\n");

        printf("=MAX:");

        scanf("%d",&maxage);

        printf("=MIN:");

        scanf("%d",&minage);

        break;

    case 5:

        printf("=为您展现了所有员工信息:)\n");

        break;

    default:

        printf("=您所选择的查询方式尚未开发:(\n");

        break;

    }

    //输出信息

    int i = 0;

    for(i=0;i<employeegs;i++){

        if(choice==5

            ||

            (choice ==1&&strstr(employeesz[i].name,cxtj))

            ||

            (choice==2&&strstr(employeesz[i].gender,cxtj))

            ||

            (choice==3&&employeesz[i].money>=minmoneytj&&employeesz[i].money<=maxmoneytj)

            ||

            (choice==4&&employeesz[i].age>=minage&&employeesz[i].age<=maxage)){

            printf("========================================\n");

            printf("=\t姓名：%s\n",employeesz[i].name);

            printf("=\t性别：%s\n",employeesz[i].gender);

            printf("=\t年龄：%d\n",employeesz[i].age);

            printf("=\t工资：%lf\n",employeesz[i].money);

            printf("=\t电话：%s\n",employeesz[i].phone);

            printf("=\t地址：%s\n",employeesz[i].location);

            //计算总人数和总工资与总年龄方便后续计算平均数

            count++;

            allmoney+=employeesz[i].money;

            allage+=employeesz[i].age;

        }

    }

    //计算平均数和输出总人数

    printf("======总人数为：%d\n",count);

    if(count == 0){

        printf("由于总人数为0，不可计算平均数据\n");

    }else{

        double argmoney = allmoney/count;

        printf("======平均工资为：%.1lf\n",argmoney);

        double argage = (double)allage/count;

        printf("======平均年龄为：%.1lf\n",argage);

    }

}

//员工信息文件读写与储存

//员工信息的储存

void savewenjian(){

    FILE \*fp;

    fp = fopen("Employee.txt","w+");

    int i;

    for(i=0;i<employeegs;i++){

        fprintf(fp,"%s %s %s %lf %d %s\n",employeesz[i].name,employeesz[i].gender,employeesz[i].phone,employeesz[i].money,employeesz[i].age,employeesz[i].location);

    }

    fclose(fp);

    printf(":) 已经为您自动保存员工信息！\n");

}

//员工信息的读取

void readwenjian(){

    FILE \*fp;

    fp = fopen("Employee.txt","r");

    if(NULL==(fp = fopen("Employee.txt","r"))){

        printf("打开文件失败请检查程序文件是否完整\n");

        fclose(fp);

        return;

    }

    int i;

    while(

        fscanf(fp,"%s%s%s%lf%d%s\n",employeesz[i].name,employeesz[i].gender,employeesz[i].phone,&employeesz[i].money,&employeesz[i].age,employeesz[i].location) !=EOF

        ){

        i++;

    }

    employeegs = i;

    fclose(fp);

}

//修改员工信息

void changeemployee(){

    //检查输入姓名是否合法

    printf("请输入要修改员工的姓名\n");

    char c[20];

    scanf("%s",c);

    if(getemployeename(c)==-1){

        printf("未查询到此员工信息请选择添加员工操作\n");

        return ;

    }

    int i=0,count=0;

    for(i=0;i<employeegs;i++){

        if(strstr(employeesz[i].name,c)){

        count = i;

        }

    }

    printf("请输入新的性别：");

    scanf("%s",employeesz[count].gender);

    printf("请输入新的电话：");

    scanf("%s",employeesz[count].phone);

    printf("请输入新的工资：");

    scanf("%lf",&employeesz[count].money);

    printf("请输入新的年龄：");

    scanf("%d",&employeesz[count].age);

    printf("请输入新的地址：");

    scanf("%s",employeesz[count].location);

    printf("恭喜修改成功\n");

    savewenjian();

}

//删除员工

void Deletempolyee(){

     printf("=请输入要删除员工的姓名\n");

    char c[20];

    scanf("%s",c);

    if(getemployeename(c)==-1){

        printf("=未查询到此员工信息请选择添加员工操作\n");

        return ;

    }

    int i=0,count=0;

    for(i=0;i<employeegs;i++){

        if(strstr(employeesz[i].name,c)){

            count = i;

        }

    }

    int nums=0;

    for(nums = count;nums<employeegs-1;nums++){

        employeesz[nums]=employeesz[nums+1];

    }

    employeegs--;

    savewenjian();

    printf("已经删除了该员工\n\n");

}

//排序员工

void sortemployee(){

    int choice=0;

    printf("================管理员身份进入了排序系统=================\n");

    printf("=1.按照姓名排序                                         =\n");

    printf("=2.按照性别排序                                         =\n");

    printf("=3.按照工资排序                                         =\n");

    printf("=4.按照年龄排序                                         =\n");

    printf("=5.按照电话排序                                         =\n");

    printf("=========================================================\n");

    printf("=请输入您的操作\n\n");

    scanf("%d",&choice);

    int j=0,i=0;

    for(i=employeegs-1;i>0;i--){

        for(j=0;j<i;j++){

            if(choice==1&&strcmp(employeesz[j].name,employeesz[j+1].name)>0||choice==2&&strcmp(employeesz[j].gender,employeesz[j+1].gender)>0||choice==3&&employeesz[i].money>employeesz[j+1].money||choice==4&&employeesz[j].age>employeesz[j+1].age||choice==5&&strcmp(employeesz[j].phone,employeesz[j+1].phone)>0){

                struct Employee EM = employeesz[j];

                employeesz[j]=employeesz[j+1];

                employeesz[j+1]=EM;

            }

        }

    }

    printf("=排序成功，若想查看排序结果可以进入查询系统\n\n");

    savewenjian();

}

//主函数模块

int main(){

    //开始就读取文件

    readwenjian();

    int flag = 0;

    //主菜单界面

    printf("=========================================================\n");

    printf("=                欢迎来到职工管理系统                   =\n");

    printf("=                   请选择账户登录                      =\n");

    printf("=                  账号输入格式如下                     =\n");

    printf("=             管理员账号：administrator                 =\n");

    printf("=   普通用户请使用自己的名字的英文大写首字母进行登录    =\n");

    printf("=                  新员工请进行注册                     =\n");

    printf("=              登录请输入1，注册请输入2                 =\n");

    printf("=========================================================\n");

    printf("=请输入您的操作方式：");

    scanf("%d",&flag);

    printf("\n");

    int loflag=0;

    int reflag=0;

    if(flag == 1){

        //登录

        loflag = userslogin();

    }else if(flag == 2){

        //注册

        reflag = registerusers();

    }else{

        printf("=您输入的不是1或者2，操作非法自动退出程序\n");

        return 1;

    }

    //功能菜单界面

    //与用户的交互逻辑

    //管理员权限

    if(loflag==1||reflag==1){

    while(Flag==9){

        int choice=0;

        menu();

        scanf("%d",&choice);

        switch (choice) {

        case 1:

            addyg();

            savewenjian();

            break;

        case 2:

            Deletempolyee();

            break;

        case 3:

            Findemployee();

            break;

        case 4:

            sortemployee();

            break;

        case 5:

            changeemployee();

            break;

        case 6:

            printf("==已经为您最后一次保存员工数据==  :)\n");

            savewenjian();

            printf("==退出本程序！\n");

            return 1;

            break;

        default:

            printf("非法操作应当禁止   :(\n");

            break;

        }

    }

    //普通用户权限

    while(Flag!=9){

        int choice=0;

        menu();

        scanf("%d",&choice);

        switch (choice) {

        case 1:

            Findemployee();

            break;

        case 2:

            changeemployee();

            break;

        case 3:

            printf("==已经为您最后一次保存员工数据:)\n");

            savewenjian();

            printf("==退出本程序！\n");

                getchar();

            return 1;

            break;

        default:

            printf("==非法操作应当禁止:(\n");

            break;

        }

    }

    }else if(loflag==0||reflag==0){

        return 0;

    }

    int ma=0;

    scanf("%d",&ma);

    return 0;

}

5、代码或者功能测试结果

（你用什么样的数据进行测试，包括输入，预计输出，实际输出。如果实际输出和预计输出不一致，说明有错，**分析错误的原因。**）（注：**分析非常重要，只有输出，没有分析则成绩会受影响**）输入：正确、不正确

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 测试方案编号 | 输入 | 预计输出 | 实际输出 | 原因分析与修改，最后通过与否 |
| 1 | 错误的密码 | 密码错误并且退出程序 | 再次输入一个数，退出程序 | 主函数中的管理登录的逻辑出错未设置当登录函数返回0时的操作 |
| 2 | 输入未知的操作方法 | 操作方法错误，并且重新输入 | 空 | 管理操作的函数出问题，增加了输入未知操作时的反应 |
| 3 | 输入未知的用户名 | 用户未注册 | 空 | 注册函数问题，增加了功能 |

6、本人在团队中承担的工作说明及课程设计体会

工作说明：本人在此次课设中完成了部分代码的编写，编写了几个关键的功能函数，完成了代码的测试环节。

设计体会：让我更好地了解到了课设的意义，以及团队合作的意义。