**南京航空航天大学金城学院**

**《C语言程序设计》课程设计**

**实验报告**

**题目: 企业员工信息管理系统**

**组号: 第三十五组**

**学号: 2018023417**

**姓名: 苗壮**

**完成日期: 2019 年 2 月 22 日**

# 目 录

*一、需求分析****…………………………………………………………3***

*二、程序的主要功能****…………………………………………………3***

*三、程序运行平台****……………………………………………………3***

*四、系统总框架图****……………………………………………………5***

*五、程序类的说明****……………………………………………………5***

*六、模块分析****…………………………………………………………7***

*七、比较有特色的函数****………………………………………………7***

*八、存在的不足与编程体会****…………………………………………10***

*九、参考文献****…………………………………………………………10***

*十、程序源代码****………………………………………………………10***

**一、需求分析**

为方便企业对员工信息的管理和自动化，特设计此系统。由于c语言程序的可移植性和可维护性较强，且数据比较安全，所以采用c语言进行设计。

1. **程序的主要功能**

1、信息录入功能：

将员工的个人信息（姓名、性别、电话、工号、所属部门、职位、工号、入职年份、薪水）以链表形式录入系统并保存。

2、信息编辑功能：

将员工的信息进行修改以及直接删除员工的所有记录。

3、信息显示功能：

在屏幕上显示所有员工的所有信息。

4、查询功能：

通过输入某项信息的数值，查找所有符合条件的员工并显示这些员工所有项目的信息。

5、排序功能：

通过选定信息的某个项目，将所有员工的信息按照该项目数值的升序进行排序。

**三、程序运行平台**

**Visual Studio 2019 preview**

运行程序“企业员工信息管理系统”，界面如下：

◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆

欢迎使用企业员工信息管理系统

制作者：苗壮

◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆

请输入数字选择功能：

1. 员工信息录入
2. 保存员工信息
3. 显示员工信息
4. 编辑员工信息
5. 删除员工信息
6. 员工信息排序
7. 查找员工信息
8. 退出系统

◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆

1. **选择1进入员工信息录入**

请输入员工姓名：

请输入员工性别：

请输入员工手机号码：

请输入员工工号：

请输入员工所属部门：

请输入员工职位：

请输入员工入职年份：

请输入员工薪水

1. **选择2保存员工信息**

保存员工信息到本地TXT文件

1. **选择3显示员工信息**

姓名 性别 手机号码 工号 所属部门 职位 入职年份 薪水

XX XX XXXXXX XX XXXXXX XX XXXXXX XX

1. **选择4编辑员工信息**

请选择查找被编辑员工信息的方式：

1. 按姓名查找
2. 按工号查找
3. 按电话号码查找
4. 取消编辑返回上一级菜单
5. **选择5删除员工信息**

1)请选择查找被删除员工信息的方式：

2)输入工号选择删除条目

3)输入姓名选择删除条目

4)取消删除返回上一级菜单

1. **选择6员工信息排序**

请选择排序的依据：

1. 按工号排序
2. 按薪水排序
3. 取消排序返回上一级菜单
4. **选择7查找员工信息**

请输入数字选择查找方式

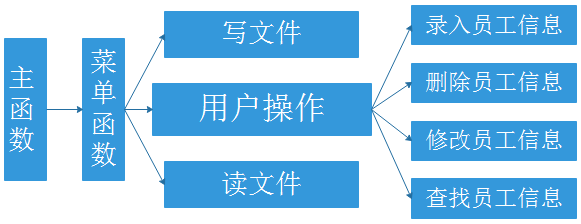
1. 姓名查询
2. 手机号码查询
3. 工号查询
4. 取消查询返回上一级菜单
5. **选择8退出程序**

请输入数字选择是否退出

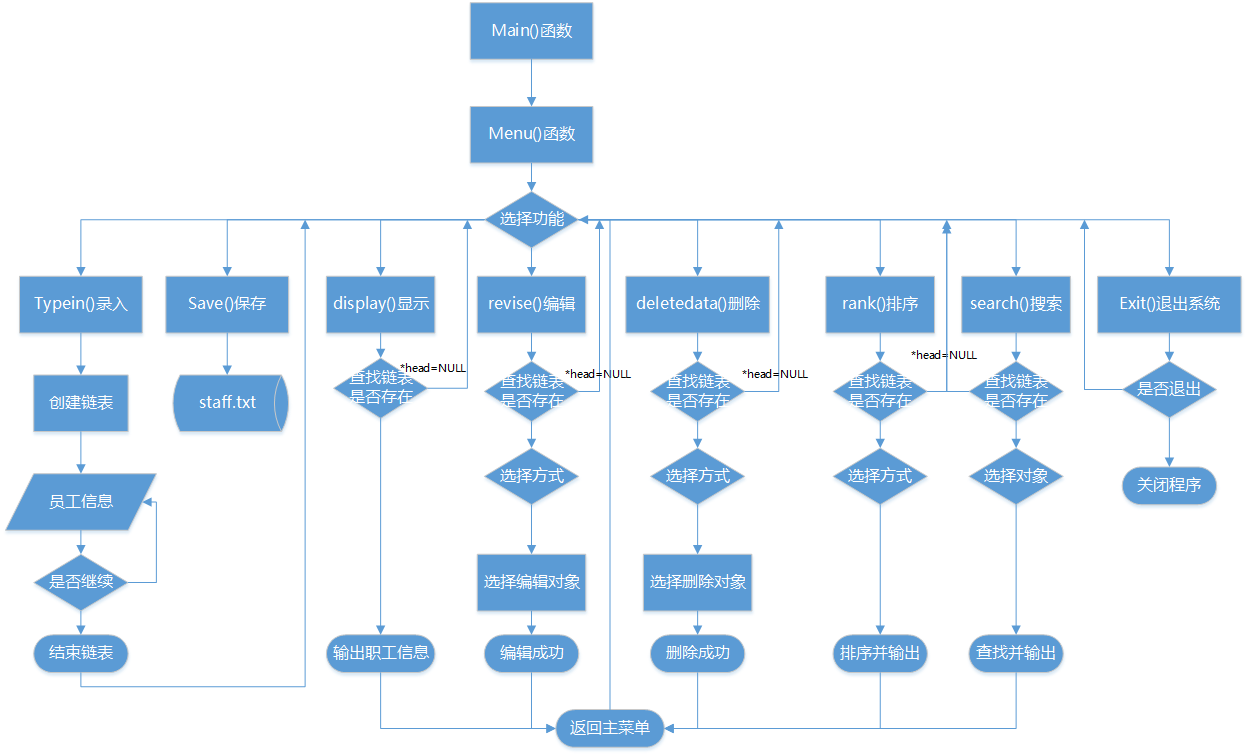
1. 确认退出
2. 返回上一级菜单

**四、系统总框架图**

4.1程序总框架图

****

4.2程序数据流程图

**五、程序类的说明**

#include<stdio.h>

#include<string.h>

#include<stdlib.h>

#define SIZE\_W 30//定义文字型char数组最大长度

#define SIZE\_N 20//定义数字型char数组最大长度

typedef struct staff //员工的基本信息

{

char name[SIZE\_W];//员工姓名

char sex[SIZE\_N];//员工性别

char phone[SIZE\_W];//员工手机号码

char number[SIZE\_W];//员工工号

char department[SIZE\_W];//员工部门

char position[SIZE\_W];//员工任职情况

int year;//员工入职年份

int payment;//员工薪水

struct staff \*next;

}STAFF;

STAFF \*head = NULL;

STAFF \*p, \*q, \*current;

struct staff \* creat\_link()//创建链表

{ char test[20];

head = (STAFF\*)malloc(sizeof(STAFF));//head指向头结点

head->next = NULL;//空链表

p = head;//head赋给临时指针变量

q = (STAFF\*)malloc(sizeof(STAFF));//q指向第一个结点，之后赋值

square();

printf("请输入员工信息：\n");

typein(q);//传递第一个结点地址进行数据录入

strcpy(test, creat\_link\_test());//把确认信息暂存到test中

while (strcmp(test, "y") == 0)//确认是否继续输入

{

creat\_link\_next();//链表-检验是否继续输入

strcpy(test, creat\_link\_test());//再次确认是否继续并赋给test暂存

}

p->next = q;//把q指的结点链接到上一个尾节点

p = q; // 把p指向q指的结点

p->next = NULL;//结尾赋值NULL

return head;

}

**六、模块分析**

1.添加模块

用户录入新的学生信息并保存。

2.显示模块

将用户录入的学生信息显示在屏幕上。

3.编辑模块

用户可根据实际情况修改已经录入额信息。

4.删除模块

用户可根据实际情况删除一个或多个学生的所有信息。

5.查找模块

用户可根据某项目特定的数据查找所有符合条件的学生并显示符

合条件学生的所有信息。

**七、比较有特色的函数**

struct staff \* creat\_link()//创建链表并且协同输入

{

char test[20];

head = (STAFF\*)malloc(sizeof(STAFF));//head指向头结点

head->next = NULL;//空链表

p = head;//head赋给临时指针变量

q = (STAFF\*)malloc(sizeof(STAFF));//q指向第一个结点，之后赋值

square();

printf("请输入员工信息：\n");

typein(q);//传递第一个结点地址进行数据录入

strcpy(test, creat\_link\_test());//把

while (strcmp(test, "y") == 0)

{

creat\_link\_next();//链表-检验是否继续输入

strcpy(test, creat\_link\_test());

}

p->next = q;//把q指的结点链接到上一个尾节点

p = q; //分析：少指向一次 把p指向q指的结点

p->next = NULL;//结尾赋值NULL

return head;

}

void creat\_link\_next(void)//链表-检验是否继续输入

{

p->next = q;

p = q;

q = (STAFF\*)malloc(sizeof(STAFF));

printf("请继续输入\n");

typein(q);

}

char \*creat\_link\_test(void)//链表-子菜单

{

char \*p = (char \*)malloc(sizeof(char));

square();

printf(" 是否要继续输入？\n");

printf(" 是请按'y' 否请按'n'\n");

square();

printf("请输入选择是否继续");

scanf("%s", p);

if ((strcmp(p, "n") != 0) && (strcmp(p, "y") != 0))

{

printf("输入字符错误!请重新输入\n");

creat\_link\_test();

}

else

return p;

}

说明：建立链表，并使用malloc函数灵活使用内存。链表数据的输入以创建输入函数typein的方式进行。在输入一组数据后用户可根据实际情况调用char\*creat\_link\_test(void)函数，以输入y或n的方式选择继续输入下一组数据或者终止输入。

void color(short x) //自定义函根据参数改变颜色

{

if (x >= 0 && x <= 15)//参数在0-15的范围颜色

SetConsoleTextAttribute(GetStdHandle(STD\_OUTPUT\_HANDLE), x); //只有一个参数，改变字体颜色

else//默认的颜色白色

SetConsoleTextAttribute(GetStdHandle(STD\_OUTPUT\_HANDLE), 7);

void square(void)

{

int i = 0;

for (i = 0; i < line; i++)

{

color(3);

printf("%2c", 4);

color(15);

}

printf("\n");

}

}

说明：通过使用SetConsoleTextAttribute颜色函数用来改变方块的颜色为蓝色（文件定义包含在Windows.h和stdlib.h中），又由于该函数改变的是整个画面，不能应用在局部字体，所以在输出方块函数中先改变方块颜色，在输出方块完成后再次调用该函数改变字体颜色为比默认颜色略亮的亮白色，使得整体函数更加的美观。

附：各个数字对应的颜色

0 = 黑色 1 = 蓝色 2 = 绿色 3 = 湖蓝色 4 = 红色 5 = 紫色

6 = 黄色 7 = 白色 8 = 灰色 9 = 淡蓝色 A = 淡绿色 B = 淡浅绿色

C = 淡红色 D = 淡紫色 E = 淡黄色 F = 亮白

* + 1. **过程中存在的不足、对策、与编程体会**

1、由于员工信息占据的字符有很大的不同，经过调试过后依旧没有找到最好的方法对输出的数据进行对齐处理。

2、为了避免需要输入数字、字符情况输入非数字、字符导致程序运行发生错误或无法运行，在用于数字输入的scanf函数后加入getchar（）命令防止程序无限死循环。

3、在Debug 模式下， VS 会把未初始化的栈内存全部填成0xcc，当字符串看就是 烫烫烫烫…… 会把未初始化的堆内存全部填成0xcd，当字符串看就是 屯屯屯屯…… 可以让我们方便地看出那些内存没初始化 未初始化的变量会被系统赋初值为0xCC,超过了ASCII码0-127这个范围，因此这个“字符串”被系统当成了宽字符组成的字符串， 即两个字节数据组成一个字符，而0xCCCC表示的宽字符正好是乱码中的那个“烫”字。 烫烫烫烫烫烫烫烫烫烫烫烫烫烫烫烫烫烫烫烫烫烫烫烫烫烫烫是debug中未初始化的栈变量 屯屯屯屯屯屯屯屯屯屯屯屯屯屯屯屯屯屯屯屯屯屯屯屯屯屯屯是debug中未初始化的堆变量。

**九、参考文献**

人民邮电出版社 《C Primer Plus（第六版）》

上海交通大学出版社 《C 语言程序设计》

* 1. **程序源代码**

#define \_CRT\_SECURE\_NO\_DEPRECATE

#include<stdio.h>

#include<string.h>

#include<stdlib.h>

#include<Windows.h>

#define SIZE\_W 30//定义文字型char数组最大长度

#define SIZE\_N 20//定义数字型char数组最大长度

#define line 17 //定义方块数量

void color(short x);//颜色改变函数

void square(void);//打印line个方块

void menu(void);//菜单

void exit\_system(void);//退出系统

struct staff \* creat\_link(void);//创建链表

char \*creat\_link\_test(void);//链表-子菜单

void creat\_link\_next(void);//链表-检验是否继续输入

void typein(struct staff\*temp);//输入

void display(struct staff \*head);//读取

void save(struct staff \*head);//保存

void save2(struct staff \*head);//即时保存函数

void delete\_data(void);//删除

void delete\_data\_number(void);//删除-工号查找

void delete\_data\_name(void);//删除-姓名查找

void search(void);//查找

void search\_number(void);//查找-按工号

void search\_name(void);//查找-按姓名

void search\_phone(void);//查找-按手机号码

void revise(void);//编辑

void revise\_number();//编辑-查找-按工号

void revise\_name(void);//编辑-查找-按姓名

void revise\_phone();//编辑-查找-按手机号码

//void sort(void);//排序

//void sort\_number(struct staff \*head);//排序-工号

//void sort\_payment(struct staff \*head);//排序-薪水

void color(short x)//自定义函根据参数改变颜色

{

if (x >= 0 && x <= 15)//参数在0-15的范围颜色

SetConsoleTextAttribute(GetStdHandle(STD\_OUTPUT\_HANDLE), x);//只有一个参数，改变字体颜色

else

SetConsoleTextAttribute(GetStdHandle(STD\_OUTPUT\_HANDLE), 7);//改为默认的颜色白色

}

void square(void)//打印line个方块

{

int i = 0;

for (i = 0; i < line; i++)

{

color(3);

printf("%2c", 4);

color(15);

}

printf("\n");

}

typedef struct staff//定义结构体

{

char name[SIZE\_W];//员工姓名

char sex[SIZE\_N];//员工性别

char phone[SIZE\_W];//员工手机号码

char number[SIZE\_W];//员工工号

char department[SIZE\_W];//员工部门

char position[SIZE\_W];//员工任职情况

int year;//员工入职年份

int payment;//员工薪水

struct staff \*next;

}STAFF;

STAFF \*head = NULL;

STAFF \*p, \*q, \*current;

int main()

{

square();

printf(" 欢迎使用企业员工信息管理系统\n");

printf(" 制作人：苗壮\n");

square();

printf("\t");

system("pause");

system("CLS");

printf("\n");

menu();

return 0;

}

void menu(void)

{

int choice;

square();

printf(" 请输入数字选择功能：\n");

printf(" 1)员工信息录入\n");

printf(" 2)保存员工信息\n");//完成

printf(" 3)显示员工信息\n");//完成

printf(" 4)编辑员工信息\n");

printf(" 5)删除员工信息\n");

printf(" 6)查找员工信息\n");

printf(" 7)退出系统\n");

square();

printf("请输入数字选择");

scanf("%d", &choice);

getchar();//防止输入错误导致程序死循环

switch (choice)

{

case 1:

head = creat\_link();

save2(head);

menu();

break;

case 2:

save(head);

menu();

break;

case 3:

display(head);

menu();

break;

case 4:

revise();

menu();

break;

case 5:

delete\_data();

menu();

break;

case 6:

search();

menu();

break;

case 7:

exit\_system();

break;

default:

printf("输入错误！请重新输入\n");

menu();

}

}

void typein(struct staff\*temp)//输入

{

printf("请输入员工姓名:");

scanf("%s", temp->name);//char类型不需要通过getchar除错

printf("请输入员工性别:");

scanf("%s", temp->sex);

printf("请输入员工手机号码:");

scanf("%s", temp->phone);

printf("请输入员工工号:");

scanf("%s", temp->number);

printf("请输入员工所属部门:");

scanf("%s", temp->department);

printf("请输入员工所在岗位:");

scanf("%s", temp->position);

printf("请输入员工入职年份:");

scanf("%d", &temp->year);

getchar();

printf("请输入员工薪水:");//防止错误输入导致程序崩溃

scanf("%d", &temp->payment);

getchar();

}

struct staff \* creat\_link()//创建链表

{

char test[20];

head = (STAFF\*)malloc(sizeof(STAFF));//head指向头结点

head->next = NULL;//空链表

p = head;//head赋给临时指针变量

q = (STAFF\*)malloc(sizeof(STAFF));//q指向第一个结点，之后赋值

square();

printf("请输入员工信息：\n");

typein(q);//传递第一个结点地址进行数据录入

strcpy(test, creat\_link\_test());//把确认信息暂存到test中

while (strcmp(test, "y") == 0)//确认是否继续输入

{

creat\_link\_next();//链表-检验是否继续输入

strcpy(test, creat\_link\_test());//再次确认是否继续并赋给test暂存

}

p->next = q;//把q指的结点链接到上一个尾节点

p = q; //分析：少指向一次 把p指向q指的结点

p->next = NULL;//结尾赋值NULL

return head;

}

void creat\_link\_next(void)//链表-检验是否继续输入

{

p->next = q;

p = q;

q = (STAFF\*)malloc(sizeof(STAFF));

printf("请继续输入\n");

typein(q);

}

char \*creat\_link\_test(void)//链表-子菜单

{

char \*p = (char \*)malloc(sizeof(char));

square();

printf(" 是否要继续输入？\n");

printf(" 是请按'y' 否请按'n'\n");

square();

printf("请输入选择是否继续");

scanf("%s", p);

if ((strcmp(p, "n") != 0) && (strcmp(p, "y") != 0))

{

printf("输入字符错误!请重新输入\n");

creat\_link\_test();

}

else

return p;

}

void display(struct staff \*head)//读取

{

/\*if (head == NULL)

{

printf("系统中没有信息！\n");

menu();

}

STAFF \*a;

printf("员工信息：\n");

printf("%s%s\n", STARS, STARS);

printf("员工姓名 员工性别 员工手机号码\t 员工工号 员工所属部门 员工职位 员工入职年份 员工薪水\n");

for (a = head->next; a != NULL; a = a->next)

{

printf("%s\t", a->name);

printf("%s\t", a->sex);

printf("%s\t", a->phone);

printf("%s\t", a->number);

printf("%s\t", a->department);

printf("%s\t", a->position);

printf("%d\t", a->year);

printf("%d\n", a->payment);

}

menu();\*/

int a[10];

char ch;

FILE \*fpRead = fopen("D://staff.txt", "r");

if (fpRead == NULL)

{

printf("error");

;

}

ch = fgetc(fpRead);

printf("员工姓名 员工性别 员工手机号码 员工工号 员工所属部门 员工职位 员工入职年份 员工薪水\n");

while (ch != EOF)

{

putchar(ch);

ch = fgetc(fpRead);

}

getchar();

menu();

}

void save(struct staff \*head)//保存函数

{

if (head != NULL)

{

FILE \*fp=NULL;

STAFF \*a;

fp = fopen("D:\\staff.txt", "w+");

for (a = head->next; a != NULL; a = a->next)

{

fprintf(fp,"%s\t", a->name);//输出链表节点数据到文件staff.txt

fprintf(fp, "%s\t", a->sex);

fprintf(fp, "%s\t", a->phone);

fprintf(fp, "%s\t", a->number);

fprintf(fp, "%s\t", a->department);

fprintf(fp, "%s\t", a->position);

fprintf(fp, "%d\t", a->year);

fprintf(fp, "%d\n", a->payment);

}

fclose(fp);

printf("保存完毕！\n");

}

else

{

printf("链表为空，保存失败！\n");

menu();

}

}

void save2(struct staff \*head)//即时保存函数

{

if (head != NULL)

{

FILE \*fp=NULL;

STAFF \*a;

fp = fopen("D:\\staff.txt", "w+");

for (a = head->next; a != NULL; a = a->next)

{

fprintf(fp, "%s\t", a->name);//输出链表节点数据到文件staff.txt

fprintf(fp, "%s\t", a->sex);

fprintf(fp, "%s\t", a->phone);

fprintf(fp, "%s\t", a->number);

fprintf(fp, "%s\t", a->department);

fprintf(fp, "%s\t", a->position);

fprintf(fp, "%d\t", a->year);

fprintf(fp, "%d\n", a->payment);

}

fclose(fp);

}

else

{

printf("链表为空，保存失败！\n");

menu();

}

}

void delete\_data(void)//删除菜单

{

if (head == NULL)

{

printf("链表为空，删除失败！\n");

menu();

}

square();

printf(" 输入数字选择功能\n");

printf(" 1)输入工号选择删除条目\n");

printf(" 2)输入姓名选择删除条目\n");

printf(" 3)取消删除返回上一级菜单\n");

square();

printf("请输入数字选择：");

int t;

scanf("%d", &t);

getchar();

switch (t)

{

case 1:

delete\_data\_number();

save2(head);

break;

case 2:

delete\_data\_name();

save2(head);

break;

case 3:

menu();

break;

default:

printf("输入错误！请重新输入\n");

delete\_data();

}

}

void delete\_data\_number(void)//按工号删除

{

char number[SIZE\_W];

q = head;

p = head->next;

printf("请输入员工工号:");

scanf("%s", number);

while ((p != NULL) && strcmp(p->number, number) != 0)

{

q = p;

p->next;

}

if (p == NULL)

printf("无法查找到匹配的员工工号\n");

else

{

q->next = p->next;

printf("删除成功！\n");

}

save2(head);

}

void delete\_data\_name(void)//按姓名删除

{

char name[SIZE\_W];

q = head;

p = head->next;

printf("请输入员工姓名:");

scanf("%s", name);

while ((p != NULL) && strcmp(p->name, name) != 0)

{

q = p;

p->next;

}

if (p == NULL)

printf("无法查找到匹配的员工姓名\n");

else

{

q->next = p->next;

printf("删除成功！\n");

}

save2(head);

}

void search(void)//搜索菜单

{

if (head == NULL)

{

printf("链表为空，查找失败！\n");

menu();

}

square();

printf(" 请输入数字选择查询方式：\n");

printf(" 1)姓名查询\n");

printf(" 2)手机号码查询\n");

printf(" 3)工号查询\n");

printf(" 4)返回上一级菜单\n");

square();

printf("请输入数字选择");

int n;

scanf("%d", &n);

getchar();

switch (n)

{

case 1:

search\_name();

break;

case 2:

search\_phone();

break;

case 3:

search\_number();

break;

case 4:

break;

default:

printf("输入错误！请重新输入\n");

search();

}

}

void search\_number(void)//按工号查找

{

char number[SIZE\_W];

int test = 0;

p = head;

printf("请输入要查找的员工的姓名:");

scanf("%s", number);

while (p != NULL)

{

if (strcmp((p->number), number) == 0)

{

printf("员工信息：\n");

square();

printf("员工姓名 员工性别 员工手机号码\t 员工工号 员工所属部门 员工职位 员工入职年份 员工薪水\n");

printf("%s\t", p->name);

printf("%s\t", p->sex);

printf("%s\t", p->phone);

printf("%s\t", p->number);

printf("%s\t", p->department);

printf("%s\t", p->position);

printf("%d\t", p->year);

printf("%d\n", p->payment);

test++;

break;

}

p = p->next;

}

if (test == 0)

printf("不存在该信息!");

}

void search\_name(void)//按姓名搜索

{

char name[SIZE\_W];

int test = 0;

p = head;

printf("请输入要查找的员工的姓名:");

scanf("%s", name);

while (p != NULL)

{

if (strcmp((p->name), name) == 0)

{

printf("员工信息：\n");

square();

printf("员工姓名 员工性别 员工手机号码\t 员工工号 员工所属部门 员工职位 员工入职年份 员工薪水\n");

printf("%s\t", p->name);

printf("%s\t", p->sex);

printf("%s\t", p->phone);

printf("%s\t", p->number);

printf("%s\t", p->department);

printf("%s\t", p->position);

printf("%d\t", p->year);

printf("%d\n", p->payment);

test++;

break;

}

p = p->next;

}

if (test == 0)

printf("不存在该信息!");

}

void revise(void)//编辑菜单

{

if (head == NULL)

{

printf("链表为空，编辑失败！\n");

menu();

}

int choice;

square();

printf(" 请输入数字选择功能\n");

printf(" 1)输入工号查找编辑对象\n");

printf(" 2)输入姓名查找编辑对象\n");

printf(" 3)输入手机号码查找编辑对象\n");

printf(" 4)返回上一级菜单\n");

square();

printf("请输入数字选择");

scanf("%d", &choice);

getchar();

switch (choice)

{

case 1:

revise\_number();

save2(head);

break;

case 2:

revise\_name();

save2(head);

break;

case 3:

revise\_phone();

break;

case 4:

menu();

break;

default:

printf("输入错误！请重新输入\n");

revise();

}

}

void revise\_number()//按工号编辑

{

STAFF \*q, b;

int i = 0;

q = head->next;

char number[20];

printf("请输入查询员工工号：");

scanf("%s", &number);

while (q != NULL)

{

if (strcmp((p->number), number) == 0)

{

square();

printf("请输入更改的员工姓名:");

scanf("%s", &b.name);

printf("请输入更改的员工性别:");

scanf("%s", &b.sex);

printf("请输入更改的员工手机号码:");

scanf("%s", &b.phone);

printf("请输入更改的员工工号:");

scanf("%s", &b.number);

printf("请输入更改的员工所属部门:");

scanf("%s", &b.department);

printf("请输入更改的员工所在岗位:");

scanf("%s", &b.position);

printf("请输入更改的员工入职年份:");

scanf("%d", &b.year);

getchar();

printf("请输入更改的员工薪水:");

scanf("%d", &b.payment);

getchar();

strcpy(q->name, b.name);

strcpy(q->sex, b.sex);

strcpy(q->phone, b.phone);

strcpy(q->number, b.number);

strcpy(q->department, b.department);

strcpy(q->position, b.position);

q->year = b.year;

q->payment = b.payment;

square();

printf("\n修改成功！\n");

i++;

break;

}

q = q->next;

}

if (i == 0)

printf("无法查找到输入的员工信息!\n");

}

void revise\_name()//按姓名编辑

{

STAFF \*q, b;

int i = 0;

q = head->next;

char name\_test[20];

printf("请输入查询员工姓名：");

scanf("%s", &name\_test);

while (q != NULL)

{

if (strcmp((p->name), name\_test) == 0)

{

square();

printf("请输入更改的员工姓名:");

scanf("%s", &b.name);

printf("请输入更改的员工性别:");

scanf("%s", &b.sex);

printf("请输入更改的员工手机号码:");

scanf("%s", &b.phone);

printf("请输入更改的员工工号:");

scanf("%s", &b.number);

printf("请输入更改的员工所属部门:");

scanf("%s", &b.department);

printf("请输入更改的员工所在岗位:");

scanf("%s", &b.position);

printf("请输入更改的员工入职年份:");

scanf("%d", &b.year);

getchar();

printf("请输入更改的员工薪水:");

scanf("%d", &b.payment);

getchar();

strcpy(q->name, b.name);

strcpy(q->sex, b.sex);

strcpy(q->phone, b.phone);

strcpy(q->number, b.number);

strcpy(q->department, b.department);

strcpy(q->position, b.position);

q->year = b.year;

q->payment = b.payment;

square();

printf("\n修改成功！\n");

i++;

break;

}

q = q->next;

}

if (i == 0)

printf("无法查找到输入的员工信息!\n");

}

void revise\_phone()//按手机编辑

{

STAFF \*q, b;

int i = 0;

q = head->next;

char phone[20];

printf("请输入查询员工手机号码：");

scanf("%s", &phone);

while (q != NULL)

{

if (strcmp((p->phone), phone) == 0)

{

square();

printf("请输入更改的员工姓名:");

scanf("%s", &b.name);

printf("请输入更改的员工性别:");

scanf("%s", &b.sex);

printf("请输入更改的员工手机号码:");

scanf("%s", &b.phone);

printf("请输入更改的员工工号:");

scanf("%s", &b.number);

printf("请输入更改的员工所属部门:");

scanf("%s", &b.department);

printf("请输入更改的员工所在岗位:");

scanf("%s", &b.position);

printf("请输入更改的员工入职年份:");

scanf("%d", &b.year);

getchar();

printf("请输入更改的员工薪水:");

scanf("%d", &b.payment);

getchar();

strcpy(q->name, b.name);

strcpy(q->sex, b.sex);

strcpy(q->phone, b.phone);

strcpy(q->number, b.number);

strcpy(q->department, b.department);

strcpy(q->position, b.position);

q->year = b.year;

q->payment = b.payment;

square();

printf("\n修改成功！\n");

i++;

break;

}

q = q->next;

}

if (i == 0)

printf("无法查找到输入的员工信息!\n");

}

/\*

void sort(void)

{

if (head == NULL)

{

printf("链表为空，排序失败！\n");

menu();

}

int choice;

square();

printf(" 请输入数字选择功能\n");

printf(" 1)根据工号进行排序\n");

printf(" 2)根据薪水进行排序\n");

printf(" 3)返回上一级菜单\n");

square();

printf("请输入数字选择");

scanf("%d", &choice);

getchar();

switch (choice)

{

case 1:

sort\_number(head);

break;

case 2:

sort\_payment(head);

break;

case 3:

menu();

break;

default:

printf("输入错误！请重新输入\n");

sort();

}

}

void sort\_number(struct staff \*head)

{

STAFF \*p1, \*p2;

char name[20], sex[20], phone[20], number[20], department[20], position[20];

int year, payment;

for (p1 = head->next; p1 != NULL; p1 = p1->next)

{

for (p2 = p1->next; p2 != NULL; p2 = p2->next)

{

if (strcmp(p1->number, p2->number) == 1)

{

strcpy(name, p1->name);

strcpy(sex, p1->sex);

strcpy(phone, p1->phone);

strcpy(number, p1->number);

strcpy(department, p1->department);

strcpy(position, p1->position);

year = p1->year;

payment = p1->payment;//

strcpy(p1->name, p2->name);

strcpy(p1->sex, p2->sex);

strcpy(p1->phone, p2->phone);

strcpy(p1->number, p2->number);

strcpy(p1->department, p2->department);

strcpy(p1->position, p2->position);

p1->year = p2->year;

p1->payment = p2->payment;//

strcpy(p2->name, name);

strcpy(p2->sex, sex);

strcpy(p2->phone, phone);

strcpy(p2->number, number);

strcpy(p2->department, department);

strcpy(p2->position, position);

p2->year = year;

p2->payment = payment;//

}

}

}

display(head);

}

void sort\_payment(struct staff \*head)

{

STAFF \*p1, \*p2;

char name[20], sex[20], phone[20], number[20], department[20], position[20];

int year, payment;

for (p1 = head->next; p1 != NULL; p1 = p1->next)

{

for (p2 = p1->next; p2 != NULL; p2 = p2->next)

{

if (p1->payment > p2->payment)

{

strcpy(name, p1->name);

strcpy(sex, p1->sex);

strcpy(phone, p1->phone);

strcpy(number, p1->number);

strcpy(department, p1->department);

strcpy(position, p1->position);

year = p1->year;

payment = p1->payment;//

strcpy(p1->name, p2->name);

strcpy(p1->sex, p2->sex);

strcpy(p1->phone, p2->phone);

strcpy(p1->number, p2->number);

strcpy(p1->department, p2->department);

strcpy(p1->position, p2->position);

p1->year = p2->year;

p1->payment = p2->payment;//

strcpy(p2->name, name);

strcpy(p2->sex, sex);

strcpy(p2->phone, phone);

strcpy(p2->number, number);

strcpy(p2->department, department);

strcpy(p2->position, position);

p2->year = year;

p2->payment = payment;//

}

}

}

display(head);

}

\*/

void search\_phone(void)//退出函数

{

char phone[SIZE\_W];

int test = 0;

p = head;

printf("请输入要查找的员工的姓名:");

scanf("%s", phone);

while (p != NULL)

{

if (strcmp((p->phone), phone) == 0)

{

printf("员工信息：\n");

square();

printf("员工姓名 员工性别 员工手机号码\t 员工工号 员工所属部门 员工职位 员工入职年份 员工薪水\n");

printf("%s\t", p->name);

printf("%s\t", p->sex);

printf("%s\t", p->phone);

printf("%s\t", p->number);

printf("%s\t", p->department);

printf("%s\t", p->position);

printf("%d\t", p->year);

printf("%d\n", p->payment);

test++;

break;

}

p = p->next;

}

if (test == 0)

printf("不存在该信息!");

}

void exit\_system(void)

{

int choice;

square();

printf(" 确定要退出本系统么？\n");

printf(" 请输入数字决定是否退出\n");

printf(" 1)确认退出\n");

printf(" 2)返回上一级菜单\n");

square();

printf("请输入数字选择");

scanf("%d", &choice);

getchar();

switch (choice)

{

case 1:

printf("感谢您的使用！再见！\n");

save2(head);

system("pause");

exit(EXIT\_FAILURE);

case 2:

menu();

break;

default:

printf("输入错误！请重新输入\n");

exit\_system();

}

}