

淮海工学院计算机工程学院 课程设计报告

设计名称: 数据结构课程设计

选题名称: 员工信息管理系统

姓名: 学号:

专业班级: 计算机科学与技术

系(院): 计算机工程学院

设计时间: 2018.12.25~2019.1.6

设计地点: 计算机实验室、教室

指导教师评语:

成绩:

签名:

年 月 日

1. 课程设计目的

- 1、训练学生灵活应用所学数据结构知识，独立完成问题分析，结合数据结构理论知识，编写程序求解指定问题。
2. 初步掌握软件开发过程的问题分析、系统设计、程序编码、测试等基本方法和技能；
3. 提高综合运用所学的理论知识和方法独立分析和解决问题的能力；
4. 训练用系统的观点和软件开发一般规范进行软件开发，巩固、深化学生的理论知识，提高编程水平，并在此过程中培养他们严谨的科学态度和良好的工作作风。

2. 课程设计任务与要求:

任务

根据教材《数据结构-C 语言描述》（耿国华主编）和参考书《数据结构题集（C 语言版）》（严蔚敏、吴伟民主编）选择课程设计题目，要求通过设计，在数据结构的逻辑特性和物理表示、数据结构的选择应用、算法的设计及其实现等方面加深对课程基本内容的理解和综合运用。

设计题目从任务书所列选题表中选取，每班每题不得超过 2 人。

学生自选课题。

学生原则上可以结合个人爱好自选课题，要求课题有一定的深度与难度，有一定的算法复杂性，能够巩固数据结构课程所学的知识。学生自选课题需在 18 周前报课程设计指导教师批准方可生效。

要求:

1、在处理每个题目时，要求从分析题目的需求入手，按设计抽象数据类型、构思算法、通过设计实现抽象数据类型、编制上机程序和上机调试等若干步骤完成题目，最终写出完整的分析报告。前期准备工作完备与否直接影响到后序上机调试工作的效率。在程序设计阶段应尽量利用已有的标准函数，加大代码的重用率。

2、设计的题目要求达到一定工作量（300 行以上代码），并具有一定的深度和难度。

3、程序设计语言推荐使用 C/C++，程序书写规范，源程序需加必要的注释；

4、每位同学需提交可独立运行的程序；

5、每位同学需独立提交设计报告书（每人一份），要求编排格式统一、规范、内容充实，不少于 10 页（代码不算）；

6、课程设计实践作为培养学生动手能力的一种手段，单独考核。

3. 课程设计说明书

一 需求分析

● 功能需求: 该程序是模拟员工管理系统, 实现输入浏览, 查找, 删除, 修改功能。此程序规定:

● 性能需求:

(1) 管理员能够向系统中输入每个员工的基本信息, 包括职工号、姓名、性别、年龄、学历、住址、电话等。

(2) 用户能按职工号、姓名、电话查询这个员工的基本信息;

(3) 管理员能够实现 **录入职工信息**(职工信息录入(职工信息用文件保存), 可以一次完成若干条记录的输入)、**浏览全部职工信息**(完成对全部职工信息的显示)、**查找职工信息**(①完成按职工的职工号查询职工的相关信息, 并显示。 ②完成按职工的学历查询职工的相关信息, 并显示。 ③完成按职工的电话号码查询职工的相关信息, 并显示。)**删除职工信息**(通过输入职工的姓名完成对该名职工的信息进行删除。**插入新职工信息**(完成添加新职工的信息的任务)修改某个职工信息(通过输入职工的姓名完成对该名职工的信息进行修改)。排序职工信息(可以按照职工号, 姓名, 电话号码进行排序, 默认递增排序);

● 数据需求:

程序建立一个线性表, 以数组的形式完成对员工信息(职工号、姓名、性别、年龄、学历、住址、电话)的保存

用户要输入的数据类型有:

1. 职工号, 年龄, 电话号码, 为整型;
2. 姓名, 学历, 住址, 为字符数组
3. 性别为字符 X, Y

二 概要设计

● 系统用到的抽象数据类型定义:

1、ADT LinearList{

数据元素: $D = \{a_i \mid a_i \in D_0, i=1, 2, \dots, n, n \geq 0, D_0 \text{ 为某一数据对象}\}$

关系: $S = \{\langle a_i, a_{i+1} \rangle \mid a_i, a_{i+1} \in D_0, i=1, 2, \dots, n-1\}$

基本操作:

(1) InitList(L)

(2) DestroyList(L)

(3) ClearList(L)

(4) EmptyList(L)

(5) ListLength(L)

(6) Locate(L, e)

(7) GetData(L, i)

```
(8) InsList(L, i, e)
```

```
(9) DelList(L, i, &e)
```

```
}ADT LinearList
```

● **系统中的子程序和功能说明:**

```
void menu();主菜单
```

```
void choose()选择要主菜单的操作;
```

```
void input();录入员工信息并且写到文件中
```

```
void menu2();查找菜单2
```

```
void search();能够实现按职工号, 姓名, 电话查找, 并显示;
```

```
void modify();从文件中读取数据, 并根据要求输入职工号, 查找并修改, 最后重新写入文件。
```

```
void insert();从文件中读取数据, 新添员工信息, 最后重新写入文件;
```

```
void del();从文件中读取数据, 输入职工号, 查找到职工信息, 并删除, 最后重新写入文件
```

```
void paixu();根据菜单选择按某一条件进行排序。
```

```
void saveFile(int enNum);将数据写到文件中
```

```
void saveFile2(int num, employee *em)删除原文件, 将数据写到另一个文件, 并且重命名为原文件
```

```
int ReadFile();将数据从文件中读出到em当中;
```

```
void choose2(int sum);选择要执行的排序方式。
```

```
void searchNum(employee em[], int sum);按照员工的职工号进行查找
```

```
int BinSrch1(employee r[], int length, int k);采用二分法对职工号查找
```

```
void BinSort(employee *em, int length);采用二分法对职工一趟查找
```

```
void paixuTeleph();按照员工的电话号码进行排序
```

```
void QKSort(employee r[], int low, int high);采用快速排序根据电话号码对职工排序
```

```
int QKPass(employee r[], int left, int right);快速排序中的一趟排序
```

```
void paixuName();
```

```
void paixuNUM();按照员工的职工号进行折半法排序
```

● **功能实现过程**

录入员工信息

```
void input()
```

```
{
```

```
    int n;
```

```
    printf("\t请输入你要加入的员工数量\n");
```

```
    scanf("%d", &n);
```

```
    for (int i = 0; i <= n; i++)
```

```
    {
```

```
        printf("\t输入第%d个员工的编号\n", i);
```

```
    }  
    scanf(“将em数组填写完成”);  
    将数组内容写入文件中  
    printf(“员工信息已经写入文件employee-list.txt中\n”);  
  
}
```

采用二分法查找员工信息

```
int BinSrch1(employee r[], int length, int k)  
/*在有序表l中折半查找其关键字等于k的元素，若找到，则函数值为该元素在表中的  
位置*/  
{  
    int 低端, 高端, 中间;  
    低端=0;  
    高端=数组的长度; /*置区间初值*/  
    while( 低端 <= 高端)  
    {  
        中间点=(低端+高端) / 2;  
        if (要查找的元素==r[mid]. num)  
            return (mid); /*找到待查元素*/  
        else  
            if (要查找的元素<r[mid]. num)  
                高端=低端-1; /*未找到，则继续在前半区间进行查找*/  
            else  
                low=低端+1; /*继续在后半区间进行查找*/  
    }  
    return (-1);  
}
```

修改员工信息

```
void modify()  
{  
    int n;  
    int mid;  
    int 数组的大小= 从文件中读取数据();  
    printf(“数据已经从employee-list.txt中读到em数组中\n”);  
    printf(“一共有%d个员工\n”, sum);  
}
```

```
int 员工号;
char 姓名;
char 性别;
int 年龄;
char 学历;
char 职务;
int 电话;
char 地址;
printf("请输入你要修改的员工的工号: ");
scanf("%d",&职工号);
返回职工的编号=采用二分查找法;
printf("已经找到, 原始信息为: ");
printf("职工号: 姓名: 年龄: 性别: 学历: 职务: 电话: 住址");
printf("可以修改的属性");
printf("1:职工号 2:姓名 3: 年龄 4:性别 5: 学历 6: 职务7: 电话 8: 住址\n");
printf("输入你要修改的属性: (1——8) \n");
scanf(" "选择要执行的选择");
switch (n)
{
case 1:printf("输入职工号: \n");
scanf(职工号);

    修改数组 的职工号属性
    break;
case 2:printf("输入姓名: \n");
scanf(姓名);

    修改此职工的姓名;
    break;
case 3:printf("输入性别: \n");
scanf(性别);

    修改员工的性别;
    break;
case 4:printf("输入年龄: \n");
scanf(年龄);
```

```
    修改此员工的年龄;
    break;
case 5:printf("输入学历: \n");
    scanf("学历");

    修改此员工学历;
    break;
case 6:printf("输入职务: \n");
    scanf("职务");

    修改此员工的职务
    break;
case 7:printf("输入电话: \n");
    scanf("电话");

    修改此员工的电话号码
    break;
case 8:printf("输入住址: \n");
    scanf("住址");

    修改此员工的住址;
    break;

default:printf("输入错误, 程序终止\n");
    退出程序;
    break;
}
将修改后的数组保存到文件中;
```

插入员工信息

```
void insert()
{
    int 数组的大小= 从文件中读取数据();
    int 员工号;
    char 姓名;
```

```
char 性别;
int 年龄;
char 学历;
char 职务;
int 电话;
char 地址;
printf("请输入:      职工号: \t姓名: \t年龄:\t性别: \t学历: \t职务: \t电话: \t住址: \t\n");
scanf(职工号, 姓名, 年龄, 性别, 学历, 职务, 电话, 住址);
将新添加的职工号, 姓名, 年龄, 性别, 学历, 职务, 电话, 住址增加到数组;
将数组重新写到文件中;

}
删除员工信息
void del()
{
    int 员工个数=1;
    int num, mid;
    int 数组的大小= 从文件中读取数据();
    printf("输入你要删除几个员工: ");
    scanf(员工数量);
    for (int m = 1; m <= 员工数量; m++)//数数, 几个, 1作为起点; 序号, 第几个, 0作为
    起点
    {
        printf("请输入你要删除的职工号: ");
        scanf(工号);
        返回此员工的序号 = 二分法查找到员工;
        for (int i = 此员工工号; i <= 数组末尾; i++)
        {
            把第1个员工的职工号, 姓名, 年龄, 性别, 学历, 职务, 电话, 住址全部向前移
        }
        printf("删除完毕! \n");
    }
    重新写入文件;
    对员工的工号进行折半排序
void BinSort( employee *em, int length)
```

```
/*对记录数组r进行折半插入排序, length为数组的长度*/
{
    int i, j;
    employee 员工的一个结构体;
    int 低端, 高端, 中间;
    for (i = 2; i <= 长度; ++i)
    {
        员工的一个结构体= em[i];
        低端 = 1; 高端 = i - 1;
        while 低端 <= 高端)                /* 确定插入位置*/
        {
            mid = (低端 + 高端) / 2;
            if (x.num < em[中间].num)
                高端 = 中间 - 1;
            else
                低端 = 中间 + 1;
        }
        for (j = i - 1; j >= 低端; --j)    em[j + 1] = em[j];        /* 记录依次向
后移动 */
        em[低端] = x;                    /*
插入记录 */
    }
}/*BinSort*/

}
对员工的电话进行快速排序
void QKSort(employee r[], int low, int high)
/*对记录数组r[low..high]用快速排序算法进行排序*/
{
    int 中枢元素;
    if (低端 < 高端)
    {
        中枢元素= QKPass(待排序数组, 低端, 高端); /*调用一趟快速排序, 将枢轴元素为界
划分两个子表*/
        QKSort(待排序数组, 低端, 中枢元素-1); /*对左部子表快速排序*/
    }
}
```

```
    QKSort(待排序数组, , 中枢元素+1, 高端); /*对右部子表快速排序*/

}
}
int  QKPass(employee  r[], int left, int right)
/*对记录数组r中的r[left]至r[right]部分进行一趟排序, 并得到基准的位置, 使得排序后的
结果满足其之后(前)的记录的关键字均不小于(大于)于基准记录*/
{
    employee  一个员工信息;
    int  低端, 高端;
    x = r[低端];          /* 选择基准记录*/
    low =低端;
    high =高端;
    while (低端<high)
    {
        while (低端<高端 && r[高端 ].teleph>= x.teleph)
            /* high从右到左找小于x.key的记录 */
            高端 --;
        if (低端 <高端)
        {
            r[低端] = r[高端 ];
            低端++;
        }
        /* 找到小于x.key的记录, 则进行交换*/
        while (低端<high && r[低端].teleph<x.teleph)    /* low从左到右找大于x.key的
记录 */
            低端++;
        if (低端<high)
        {
            r[高端] = r[low];
            高端--;
        } /* 找到大于x.key的记录, 则交换*/
    }
    r[低端] = x;          /*将基准记录保存到low=high的位置*/
    return 低端;         /*返回基准记录的位置*/
} /* QKPass */
```

● 系统程序功能图

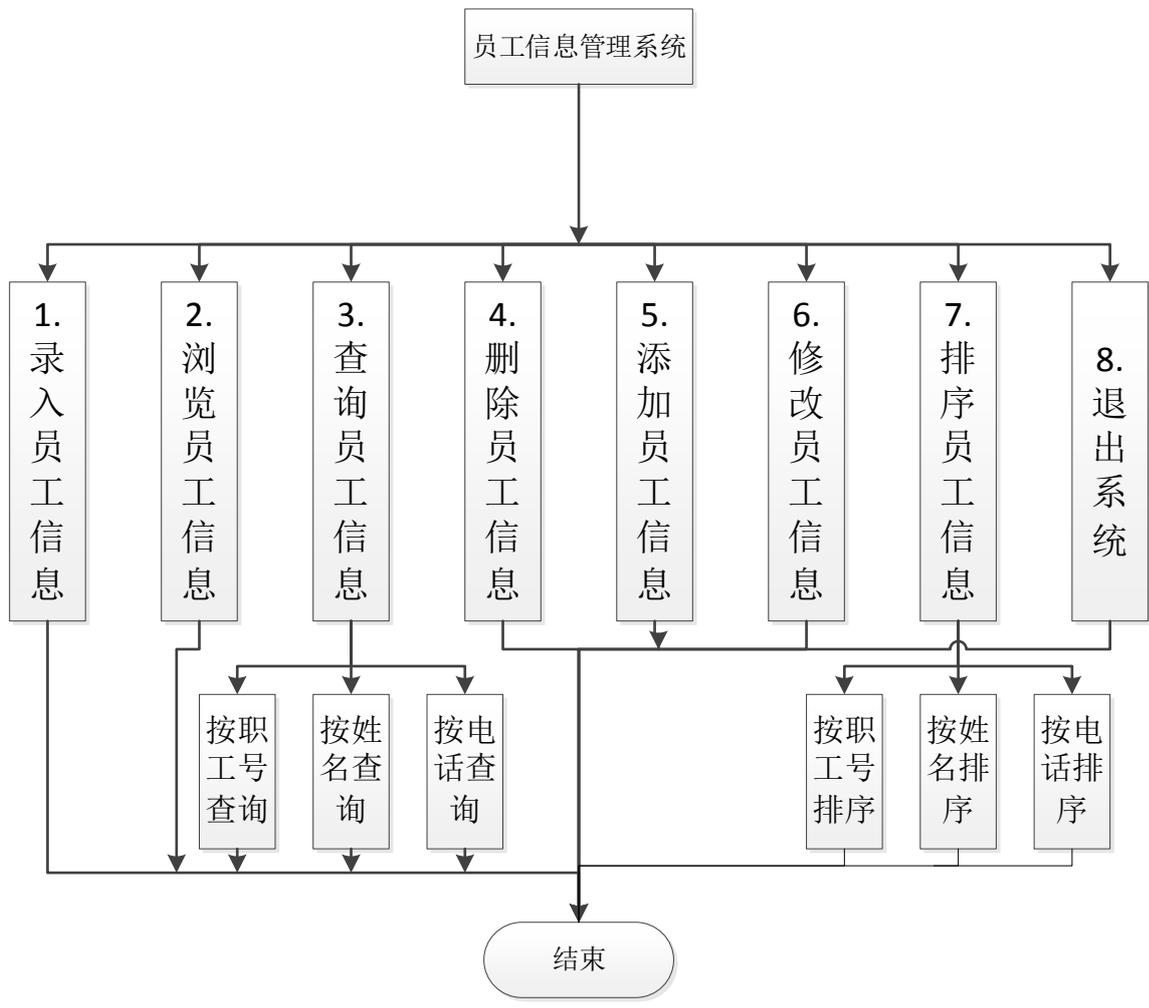


图 1：系统程序功能图

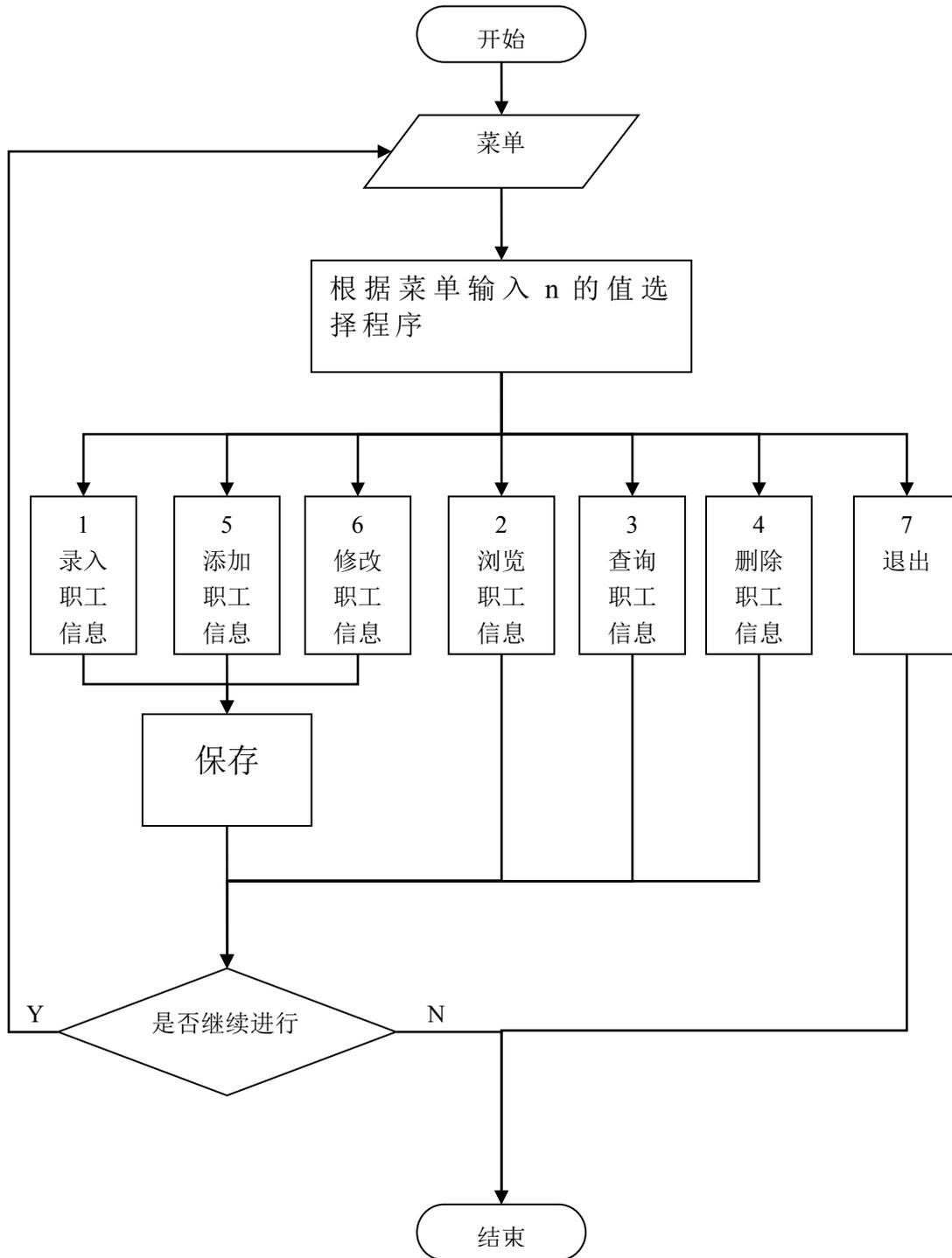


图 2：总设计流程图

1. 录入员工信息：将员工的信息写入数组中，最后写入文件中

```

void input()/
{
    int n;
    printf("\t请输入你要加入的员工数量\n");
    scanf("%d", &n);
    for (int i = 0; i <= n; i++)
    {

```

```
printf("\t输入第%d个员工的编号\n", i);
scanf("%d", &em[i].num);
getchar();
printf("\t输入第%d个员工的姓名\n", i);
scanf("%s", &em[i].name);
getchar();
printf("\t输入第%d个员工的性别\n", i);
scanf("%c", &em[i].sex);
getchar();
printf("\t输入第%d个员工的学历\n", i);
scanf("%s", &em[i].xueli);
getchar();
printf("\t输入第%d个员工的职务\n", i);
scanf("%s", &em[i].zhiwu);
getchar();
printf("\t输入第%d个员工的电话\n", i);
scanf("%d", &em[i].teleph);
getchar();
printf("\t输入第%d个员工的住址\n", i);
scanf("%s", &em[i].addr);
}
saveFile(n);
printf("员工信息已经写入文件employee-list.txt中\n");
}
```

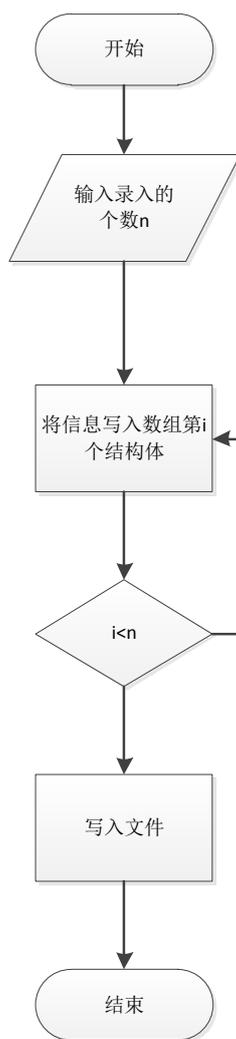


图 3：录入员工信息

2. 查找员工信息

```
void search()
{
    int sum;
    sum = ReadFile();
    printf("数据已经从 employee-list.txt 中读到 em 数组中\n");
    printf("一共有%d 个员工\n", sum);
    menu2();
    choose2(sum);
}
```

```
void menu2()
{
    printf("1. 按职工号查询\n");
    printf("2. 按姓名查询\n");
    printf("3. 按电话号码查询\n");
}
void choose2(int sum)
{
    int num;

    printf("请输入你的选择: ");
    scanf("%d", &num);
    switch (num)
    {
        case 1: searchNum(em, sum); break;
        case 2: searchName(sum); break;
        case 3: searchTeleph(em, sum); break;
        default: printf("输入错误, 请重新输入");
            choose2(sum);
            break;
    }
}
```

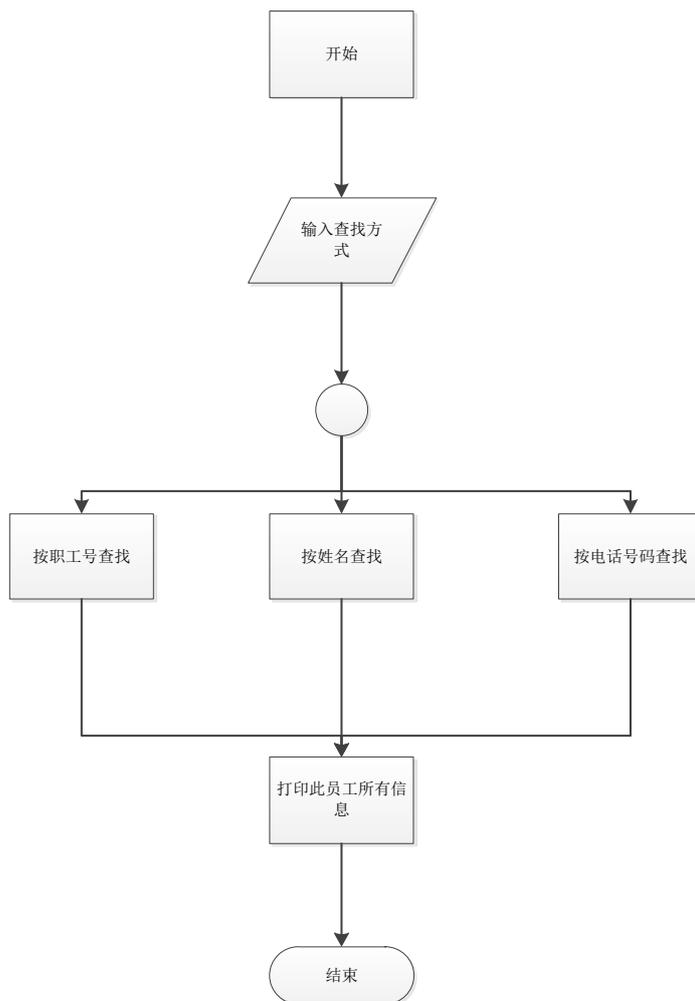


图 4 : 查找员工信息

(1) 按职工号查找员工信息: 将文件读出, 输入用户输入的职工号, 然后用二分法进行查找

```
void searchNum(employee em[],int sum)
{
    int n,mid;
    printf("输入你要查找的职工号:");
    scanf("%d",&n);
    mid=BinSrchl(em, sum,n);
    if (mid == -1)
    {
        printf("找不到!!!请重新输入\n");
        searchNum(em, sum);
    }
    else
    {
```

```
printf("职工号: %d\t姓名: %s\t年龄: %d\t性别: %c\t学历: %s\t职务: %s\t电话: %d\t住址: %s\t\n", em[mid].num, em[mid].name, em[mid].age, em[mid].sex, em[mid].xueli, em[mid].zhiwu, em[mid].teleph, em[mid].addr);

    }

}
```

```
int BinSrch1(employee r[], int length, int k)
/*在有序表1中折半查找其关键字等于k的元素, 若找到, 则函数值为该元素在表中的位置*/
{
    int low, high, mid;
    low=0;
    high=length; /*置区间初值*/
    while( low <= high)
    {
        mid=(low+high) / 2;
        if (k==r[mid]. num)
            return (mid); /*找到待查元素*/
        else
            if (k<r[mid].num)
                high=mid-1; /*未找到, 则继续在前半区间进行查找*/
            else
                low=mid+1; /*继续在后半区间进行查找*/
    }
    return (-1);
}
```

(1) 按姓名查找

```
void searchName(int sum)
{
    char str[20];
    int mid;
    int i = 0;
    printf("输入你要查找的姓名");
```

```
scanf("%s", str);
while (true)
{
    if (*str == *(em[i].name))
    {
        mid = i;
        printf("职工号: %d\t 姓名: %s\t 年龄: %d\t 性别: %c\t 学历: %s\t 职务: %s\t
电话: %d\t 住址: %s\t\n", em[mid].num, em[mid].name, em[mid].age, em[mid].sex,
em[mid].xueli, em[mid].zhiwu, em[mid].teleph, em[mid].addr);

        break;
    }
    i++;
    if (i == sum + 1)
    {
        printf("!!!查无此人, 请从新输入\n");
        searchName(sum);
    }
}
}
```

(1) 按电话号码查找

```
void searchTeleph(employee em[], int sum)
{
    int n, mid;
    printf("输入你要查找的电话号码: ");
    scanf("%d", &n);
    mid = BinSrch2(em, sum, n);
    if (mid == -1)
    {
        printf("找不到!!!请重新输入\n");
        searchNum(em, sum);
    }
    else
```

```
{
    printf("职工号: %d\t 姓名: %s\t 年龄: %d\t 性别: %c\t 学历: %s\t 职务: %s\t 电
话: %d\t 住址: %s\t\n", em[mid].num, em[mid].name, em[mid].age, em[mid].sex,
em[mid].xueli, em[mid].zhiwu, em[mid].teleph, em[mid].addr);

}

}

int BinSrch2(employee r[], int length, int k)
/*在有序表 l 中折半查找其关键字等于 k 的元素, 若找到, 则函数值为该元素在表中的
位置*/
{
    int low, high, mid;
    low = 0;
    high = length; /*置区间初值*/
    while (low <= high)
    {
        mid = (low + high) / 2;
        if (k == r[mid].teleph)
            return (mid); /*找到待查元素*/
        else
            if (k < r[mid].teleph)
                high = mid - 1; /*未找到, 则继续在前半区间进行查找*/
            else
                low = mid + 1; /*继续在后半区间进行查找*/
    }
    return (-1);
}
```

3. 修改员工信息: 将文件中的数组读出, 采用二分折半法查找到, 进行修改, 最后写入文件。

```
void modify()
{
    int n;
    int mid;
    int sum= ReadFile();
    printf("数据已经从employee-list.txt中读到em数组中\n");
```

```
printf("一共有%d个员工\n", sum);
int num;
char name[20];
char sex;
int age;
char xueli[20];
char zhiwu[20];
int teleph;
char addr[30];
printf("请输入你要修改的员工的工号: ");
scanf("%d",&num);
mid=BinSrch1(em, sum, num);
printf("已经找到, 原始信息为: ");
printf("职工号: %d\t姓名: %s\t年龄: %d\t性别: %c\t学历: %s\t职务: %s\t电话: %d\t住址: %s\t\n", em[mid].num, em[mid].name, em[mid].age, em[mid].sex, em[mid].xueli, em[mid].zhiwu, em[mid].teleph, em[mid].addr);
printf("可以修改的属性: \n");
printf("1:职工号 2:姓名 3: 年龄 4:性别 5: 学历 6: 职务7: 电话 8: 住址\n");
printf("输入你要修改的属性: (1——8) \n");
scanf("%d",&n);
switch (n)
{
case 1:printf("输入职工号: \n");
scanf("%d",&num);

em[mid].num=num;
break;
case 2:printf("输入姓名: \n");
scanf("%s", name);

strcpy( em[mid].name ,name);
break;
case 3:printf("输入性别: \n");
scanf("%c", &sex);

em[mid].sex = sex;
break;
```

```
case 4:printf("输入年龄: \n");
    scanf("%d", &age);

    em[mid].age = age;
    break;
case 5:printf("输入学历: \n");
    scanf("%s", xueli);

    strcpy(em[mid].zhiwu, zhiwu);
    break;
case 6:printf("输入职务: \n");
    scanf("%s", zhiwu);

    strcpy(em[mid].zhiwu, zhiwu);
    break;
case 7:printf("输入电话: \n");
    scanf("%d", &teleph);

    em[mid].teleph = teleph;
    break;
case 8:printf("输入住址: \n");
    scanf("%s", addr);

    strcpy(em[mid].addr, addr);
    break;

default:printf("输入错误, 程序终止\n");
    exit(0);
    break;
}
saveFile2(sum, em);
```

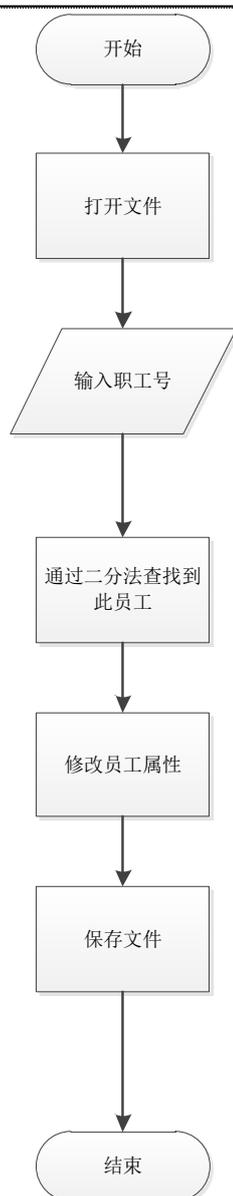


图5：修改员工信息

4. 插入员工信息：从文件中读取数组，在末尾增加员工信息，最后写入文件保存

```
void insert()  
{  
    int sum = ReadFile();  
    printf("%d\n ", sum);  
    int num;  
    char name[20];  
    char sex;  
    int age;  
    char xueli[20];  
    char zhiwu[20];
```

以上内容仅为本文档的试下载部分，为可阅读页数的一半内容。如要下载或阅读全文，请访问：<https://d.book118.com/236030102102010213>