

任务 1：隐私信息管理系统

一、实验题目

采用结构体数组或链表存储数据，设计一个小型的隐私信息管理系统。

二、实验目的

通过小型管理系统的开发，引导学生熟悉模块化程序设计方法，培养学生综合运用知识解决实际问题的能力。重点训练知识点包括函数、数组、指针、结构体、文件等。

三、需求分析

1.问题提出：设计一个小型隐私管理系统，用来保存用户的一些隐私数据。

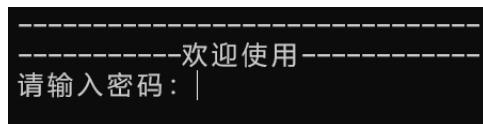
2.设计知识点：循环、分支、函数、结构体数组、指针、文件。

3.功能要求：

- (1) 建立隐私信息模块，每条隐私信息包含账号位置、账号描述、账号名和密码；
- (2) 根据不同需求（如描述、账号名）可查询到指定信息；
- (3) 可以实现增加、删除、修改信息的功能；
- (4) 显示所有的录入信息；
- (5) 将信息写入文件，并可以从文件中读取信息；
- (6) 为保证系统安全性，需要设置系统使用密码。

四、总体设计

1.进入系统密码验证



2.利用 switch 语句设计“menu”

```
*****menu*****
1.add
2.modify
3.delete
4.description-SEARCH
5.name-SEARCH
6.search all
7.File_Reading
8.File_Writting
0.exit
*****
select(0-7):|
```

每个功能选项由一个函数实现，选取选项即调用相应函数

1-3 分别可以实现增、改、删的功能；

4-5 可以实现按照指定方式查找；

6 可以显示录入的全部信息；

7-8 分别可以写入和读入文件。

3.输入增加信息 add () 函数：使用循环输入方式实现信息输入，利用结构体数组保存信息；

```

select(0-7):1
请输入账号位置: 111
请输入账号描述: 111
请输入账号名称: 111
请输入账号密码: 111
是否继续添加信息? 是请输入1, 否则输入0: 1
请输入账号位置: |

```

4.修改信息 `modify()` 函数: 使用循环比较, 定位需要修改信息在结构体数组中的位置, 再进行信息覆盖实现修改;

5.删除信息 `delete_()` 函数: 使用循环比较, 定位需要删除信息在结构体数组中的位置, 将下一组数据覆盖到本组, 实现删除功能;

6.查找信息 `search()` 函数: 使用循环比较, 定位需要修改信息的位置, 直接输出相应信息;

7.展示信息 `search_all()` 函数: 采用循环遍历数组, 并输出所有信息;

```

select(0-7):6
-----
账号位置:111
账号描述:111
账号名:111
密 码:111
-----
账号位置:222
账号描述:222
账号名:222
密 码:222
-----
账号位置:333
账号描述:333
账号名:333
密 码:333
-----
```

8.写入文件 `File_Writing()` 函数: 采用指针及文件, 实现创建文件并写入信息的功能;

9.读取文件 `File_Reading()` 函数: 循环遍历文件数据, 实现读取文件的功能。

五、详细设计

1.各模块函数名称及设计数据储存方式

(1) 主函数:

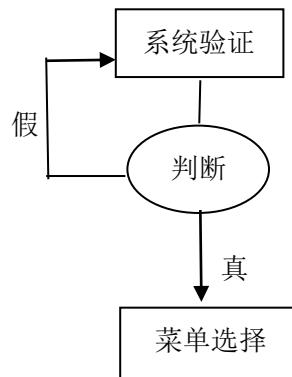
首先进行系统验证, 调用 `key()` 函数, 采用分支语句, 若 `key()` 为真则进入下一模块, 否则退出系统。

验证成功后进入菜单选项, 采用 `switch` 语句, 调用 `menu_select()` 函数, `case` 分别对应 `menu` 中的选项, 实现选择选项进入对应模块的功能。

采用无限循环包含 `switch` 语句, 除非选择 `case 0: exit` 退出系统, 否则每完成一项功能后再次跳出主菜单供用户选择。

(2) `key()` 函数

定义两个数组, 分别存放系统密码和用户输入密码, 采用分支语句, 用 `strcmp` 作为条件比较两个数组, 采用 `for` 循环嵌套两个 `if` 语句, 若条件为真即密码正确, 进入系统, 否则密码错误, 错误时重新输入密码, 若三次连续输错则退出系统;



(3) add () 函数

采用 while 循环，条件设置为判断 flag 真假，若为真则继续添加下一条信息，否则返回主菜单。输入的信息则被保存在结构体数组中；

(4) modify () 函数

定义一个数组临时储存用户输入的信息，通过 while 循环和 strcmp 将此数组与保存账户名的数组比较，查找到信息具体位置，实现在结构体数组中修改信息的功能；

(5) delete_ () 函数

与 modify () 函数相似，都是通过遍历查找比较找到指定信息，不同的是 delete_ 是通过后一组数据直接覆盖前一组数据进而实现删除功能的；

(6) search_name () 、 search_des () 、 search_all () 函数

Search 功能的实现是 modify 和 delete 功能前一部分的截取，都是通过 while 和 strcmp 循环遍历比较找到指定信息。Modify 和 delete 在找到后需要进行下一步操作，Search 功能则是找到后直接输出信息；

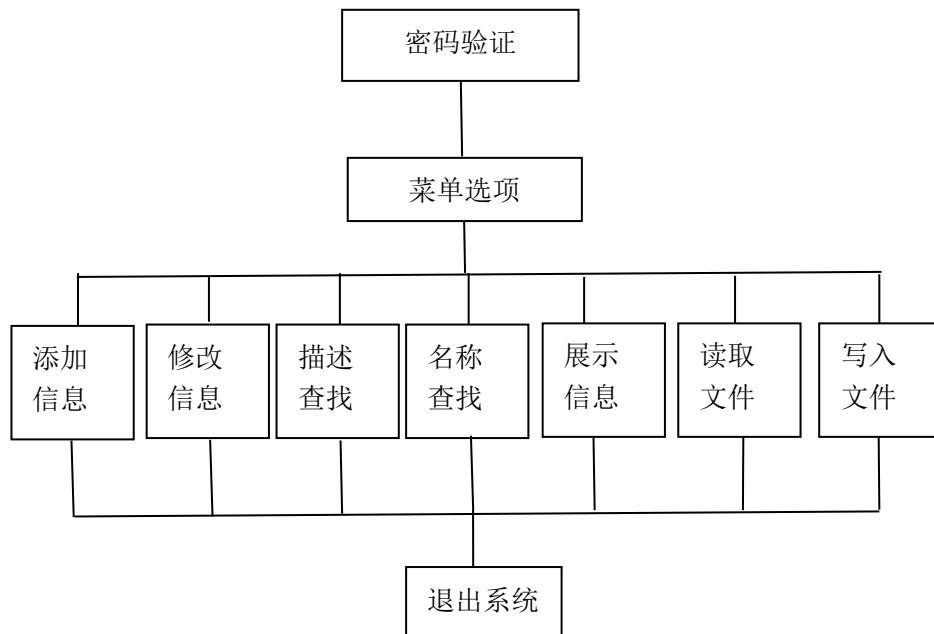
(8) File_Writing () 函数

定义指针指向文件，数组保存文件名，用 w+ 和 while 循环创建并写入文件；

(9) File_Reading () 函数

定义指针指向文件，数组保存文件名，用 r 读取文件，while 循环读取数据直到文件结束，若没有读取到数据则跳出循环。

2.模块图



六、实验结果与分析

1.第一次调试发现密码验证部分有问题，无论密码是否正确都进入到了系统，显示出主菜单，问题出在主函数中调用 key () 函数的部分。

修改了 key () 函数的循环部分和主函数中控制 key () 的部分，当密码输入正确时可以进入系统，显示主菜单，密码输入错误时，提示重新输入密码，当三次输错密码时则被判定为非法用户，阻止进入系统。

2.本系统中许多函数功能的实现都很相似，例如 modify、delete 和 search 都是先通过遍历查找定位，再进行后续操作。其中 while 循环和结构体数组起到了主要作用，因此掌握好这两部分的知识对本系统编写十分重要；而 File_Writing 和 File_Reading 都是关于文件读写的内容，学好指针与文件也很重要。

3.除了功能实现的问题，在操作界面美观上我也进行了多次修改，search _all 的部分要展示所有账号的全部信息，一开始我设置的布局和指导书中表格相似，但经过输入不同数据后发现在本文对齐上比较困难，影响整体查看效果，因此我在循环中添加了分割线，并将每组数据的信息分行单独展示，可以避免长度不一导致的展示效果。

```
select(0-7):6
账号位置:111
账号描述:111
账号名:111
密 码:111
-----
账号位置:222
账号描述:222
账号名:222
密 码:222
-----
账号位置:333
账号描述:333
账号名:333
密 码:333
```

七、小结与心得体会

尽管本系统已经可以作为一个隐私管理系统来使用，但其中还存在很多问题和可以改善的地方。比如在菜单选择上可以设置多个菜单，将同类功能放在一个选项下再进行细分。关于 modify 模块也可以改进，目前的修改功能只能修改全部信息，无法选择只修改其中一种信息，在未来的时间里我也将继续改进，希望可以将此系统的功能更加完善。

整个系统编写的过程对我来说是阻碍不断的，但也是收获颇丰的，每天课设时间在机房的编写都会遇到不同的错误和问题，但好在通过网上查找资料以及午饭和晚饭时间问同学问老师都解决了。因此，遇到问题再解决问题无疑对我是一种成长。在此特别感谢信安四班的糜炎灿同学、我的室友董雨彤还有我的 C 语言向老师以及课设实验李老师对我的指导和帮助！

任务 2：根据指定汉字生成电子印章

一、实验题目

根据指定汉字生成电子印章：从键盘输入 4 个汉字，程序能根据汉字的编码在隶书 56 点阵汉字字库文本文件 LiShu56.txt 中查找相应汉字的字型点阵信息，然后根据查得的字型点阵信息制作电子印章。

二、实验目的

通过小型应用程序的开发，培养综合运用知识解决实际问题的能力，训练分析问题、算法设计、编码及调试程序的能力。重点训练知识点包括数组、指针、结构体、文件等。

三、需求分析

建立印章 bmp 文件，按照 bmp 文件的格式写入文件头、信息头，从键盘输入 4 个汉字，利用汉字点阵文本对应输入的四个汉字，在 bmp 中写入相应的印章图案。

四、总体设计

1. 功能设计：

- (1) 创建一个空的 bmp 文件，写入文件头和信息头；
- (2) 在命令行输入四个汉字，并导入汉字字库文本文件，根据输入的汉字在库中准找所对应的编码；
- (3) 将编码转化为字符并显示，将数据写入空的 bmp 文件，转化为印章图片。

```
C:\Users\陈\Desktop\ccc>2.exe 2.bmp LiShu56.txt  
请输入你想制作印章的汉字： |
```

```
请输入你想制作印章的汉字：印章试验  
BM[0] = d3  
BM[1] = a1  
BM[2] = d5  
BM[3] = c2  
BM[4] = ca  
BM[5] = d4  
BM[6] = d1  
BM[7] = e9  
将编码转化为字符并显示 : d3a1d5c2cad4d1e9  
恭喜您！BMP文件已经成功生成！  
请在当前目录下查看生成的BMP文件2.bmp
```



2.bmp

2. 模块划分

- (1) 结构体 BITMAPFILEHEADER 存储文件头；
- (2) 结构体 BITMAPINFOHEADER 存储信息头；
- (3) 结构体 RGBDATA 存储图片 rgb 数据。
- (4) 主函数分为：
 - a) 输出提示语句，写明本系统操作方法；
 - b) 用命令行中给出的文件名新建一个空的 BMP 文件，将用户输入的汉字转化为编码在库中查询；

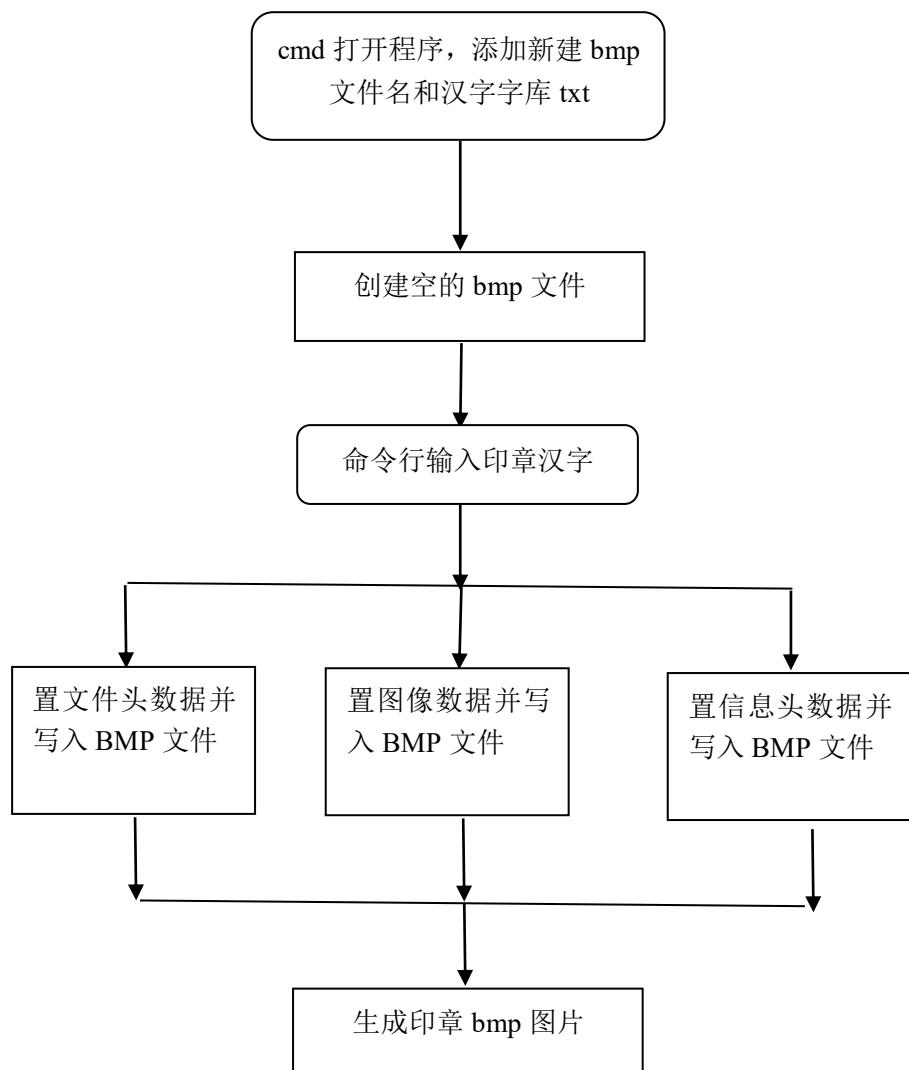
- c) 置文件头和信息头数据写入 BMP 文件;
- d) 置图像数据并写入 bmp 文件。

五、详细设计

1. 数据结构

- (1) 定义结构体 BITMAPFILEHEADER 存储文件头, BITMAPINFOHEADER 存储信息头, RGBDATA 存储图片 rgb 数据;
- (2) 定义文件指针, *bmpData 指向图像数据, *fp1 指向 BMP 文件, *fp2 指向存储汉字字库的 txt 文件。
- (3) 定义 width 为图像宽度, height 为图像高度, datasize 为数据尺寸即图片大小。
- (4) 定义字符数组 str_key[4]作为每个字的标记, str_col[120]为行列标记。

2. 模块功能



(1) 第一步：创建 BMP 文件并转换汉字编码

用命令行中给出的文件名新建一 BMP 文件，此时还是一个空文件，cmd 打开两个的两个命令参数 argv[1],argv[2]，分别为空的 BMP 文件以及汉字字库 txt 文件。使用 “wb+” 命令写入 fp1，“r” 命令读入 fp2。

按照编码读取命令行中输入的四个汉字，并将编码转化为字符并显示，将汉字的编码转到 str 数组里，用于后续查找。

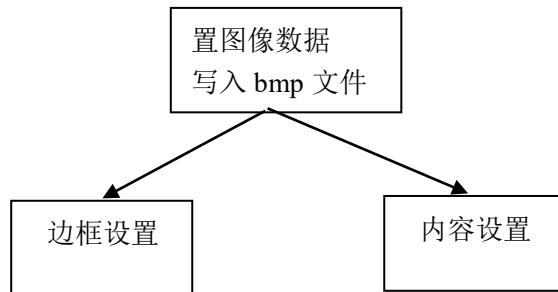
(2) 第二步：置文件头数据并写入 BMP 文件

bmdheader 存储文件头数据，置文件头数据并利用 “fwrite” 命令将数据写入 fp1 的 BMP 文件中。

(3) 第三步：置信息头数据并写入 BMP 文件

Bmiheader 存储信息头数据，置信息头数据并利用 “fwrite” 命令将数据写入 fp1 的 BMP 文件中。

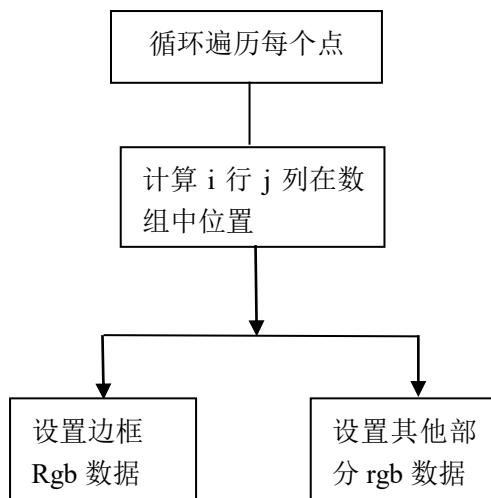
(4) 第四步：置图像数据并写入 BMP 文件



分配足够内存，让 bmpData 指向这块内存，用于存放图像各像素点的 RGB 分量值，在 bmpData 所指内存中置图像 RGB 数据，然后将所有数据写入 BMP 文件。

边框制作：

定义 i, j 分别储存每行每列的像素点数据，使用嵌套循环遍历图片中每个像素数据，计算 bmp 图片中第 i 行第 j 列图像数据在 bmpData[] 数组中的位置，在循环中使用分支语句为边框写入 rgb 数据。



内容制作：

安排四个字的位置，采用每个字做不同的偏移，第一二三四个字分别是(4,56)(56,56)(4,4)(56,4)。

使用循环匹配查找汉字，每次从 str 数组中截取 4 个字符串，进行匹配，在表中找到汉字，将指针置于文件头，从头开始寻找，每次从汉字表中读取一行，放入 stri 中一定要把-1、0 去掉。使用分支语句判断，当匹配到汉字编码时开始标记数据，将汉字字库中的匹配汉字分类标记，标记“，”为不合法字符，遇到时直接跳过，“x”为组成汉字的部分，设置 rgb 值，其他字符设置 rgb 为背景色数据。每找到一个匹配汉字后循环重新从第一行找起节约时间。所有循环结束后将 bmpData 所指的 RGB 数据一次性写入 BMP 文件，释放 bmpData 所指的内存空间，置 bmpData 为空指针并关闭 fp1 所指文件。

六、实验结果与分析

1.第一次制作出的印章图片背景面积过大，四个汉字均集中在图片左上角，分析后发现在是主函数中定义的图像宽度与高度过大。计算后将图片宽高均修改为 120。此时印章大小合适。

```
long width=120; //图像宽度  
long height=120; //图像高度
```

2.边框的设计采用数组，修改边框粗细需要计算好图像大小和字体大小安排，由于本程序采用一维数组，因此需要考虑 i, j 两个分量，当边框粗细设置为 4 时，应作设置如下：

```
if((i >= 0 && i <= 3)|| (i >= 116 && i <= 119) || (j >= 0 && j <= 3) || (j >= 116 && j <= 119))
```

(为美观设置将边框调整为 119，防止边框与字体内容重合)

3.rgb 分量的设置控制印章颜色生成，可以对照 rgb 分量表修改自己所需要的颜色

```
if(str_col[j] == 'X')  
{  
    bmpData[k].rgbBlue = 196;  
    bmpData[k].rgbGreen = 203;  
    bmpData[k].rgbRed = 255;  
}  
else  
{  
    bmpData[k].rgbBlue = 230;  
    bmpData[k].rgbGreen = 240;  
    bmpData[k].rgbRed = 255;
```

七、小结与心得体会

在打点描绘上可以采用一维数组，也可以采用二维数组，本次编写采用一维数组，在以后也可以试着改用二维数组尝试。

本次课设任务的编写相对任务一更加困难，刚拿到手时完全不知从何做起，在仔细阅读了指导书以及网上相关资料后才有了一定的认识。首先实验运用了老师发的完整程序生成印章，然后阅读研究了 demo 的编写，了解了每个部分所对应控制的模块，然后按照任务要求做对应修改。在本次学习中有很多我不懂的地方，程序编码几乎是一句一句啃下来的，非常感谢我的同学郭瑞园，在我每天定时定点的打扰下他仍然不厌其烦的一句一句为我解答。而我也很开心在学会了之后能去为其他同学做解答。课设对到这里也圆满完成了，此次我学到了很多。