



学生成绩管理系统

学生成绩管理系统包含以下功能

一、用户注册和登录：

- 1、添加数据验证（学号唯一性、成绩0-100范围等）
- 2、从文件导入用户信息： userinfo.txt

二、显示菜单

- 1、Creat：创建学生链表
- 2、Insert：插入学生信息
- 3、Modify：修改学生信息
- 4、Delete：删除学生信息
- 5、Print：输出学生信息
- 6、Load：从文件导入学生信息
- 7、Save：保存学生信息
- 0、Exit：退出

项目算法说明

- 注册登陆(login函数)
- 菜单显示(displaymenu函数)
- 创建链表(Create_Stu_Doc()函数)
- 插入结点(InsertDoc函数)
- 修改节点(Modify函数)
- 删除结点(DeleteDoc函数)
- 输出信息 (Print_Stu_Doc函数)
- 从文件中读取数据 (loadstudents函数)
- 存入文件 (savestudents函数)

命令描述:

登录和注册(login函数)

注册: adduser函数和encrypt函数

登录checkUserValid函数: 验证账户信息

```
16 void adduser(){
17     char usr[20],pwd[8];
18     printf("Please input user name: ");
19     scanf("%s",usr);
20     printf("Please input password: ");
21     scanf("%s",pwd);
22     encrypt(pwd);
23
24     FILE *fp=fopen("userinfo.txt","a");
25     if(fp==NULL){
26         printf("Open userinfo.txt failed.");
27         return;
28     }
29
30     fprintf(fp,"%s %s\n",usr,pwd);
31     if(fclose(fp)){
32         printf("Can not close the file!\n");
33         exit(0);
34     }
35 }
```

F:\C语言\C语言\程序设计技能实训1(2025年)\students.exe

```
1. Regisgtration
2.Login in
```

/*加密算法*/

void encrypt(char *pwd)

{

int i;

/*与15异或, 实现低四位取反, 高四位保持不变*/

{

for(i = 0; i < strlen(pwd); i++){

pwd[i] = pwd[i] ^ 15;

}

}

命令描述:

登录和注册(login函数)

登录checkUserValid函数: 验证账户信息

F:\C语言\C语言\程序设计技能实训1(2025年)\students.exe

1. Regisgtration
2. Login in

```
students.cpp
/*校验用户信息的合法性, 成功返回1, 否则返回0*/
int checkUserValid(sysuser psu)
{
    FILE *fp;
    char usr[30], usr1[30], pwd[10];
    int check = 0; /*检查结果变量, 初始化为0*/
    /*连接生成待校验字符串*/
    strcpy(usr, psu.username); /*复制psu->username到usr1 */
    strcpy(pwd, psu.password); /*复制psu->password到pwd */
    encrypt(pwd); /*调用例12-2的encrypt对密码进行加密*/
    /*连接usr、空格、pwd和\n构成新字符串usr, 用于在文件中检查匹配*/
    strcat(usr, " ");
    strcat(usr, pwd);
    strcat(usr, "\n");
    /*打开文件"f12-2.txt"读入*/
    if((fp=fopen("userinfo.txt", "r")) == NULL){
        printf("File open error!\n");
        exit(0);
    }
    /*从文件读入用户信息数据, 遍历判断是否存在*/
    while(!feof(fp)){
        fgets(usr1, 30, fp); /*读入一行用户信息字符串到usr1*/
        if(strcmp(usr, usr1) == 0){ /*比较判断usr与usr1是否相同*/
            check = 1;
            printf("Login successful\n");
            break;
        }
    }
    if(!check)
        printf("Login failed\n");
    /*关闭文件*/
    if(fclose(fp))
        printf("Can not close the file!\n");
    exit(0);
}
return check;
```

命令说明:

显示菜单(displaymenu)

```
[void displaymenu(){  
    printf("1:Create \n2:Insert \n3>Delete \n4:Print \n5:Load \n6:Save \n0:Exit\n");  
}
```

```
2.Login in  
1  
Please input user name: zhou  
Please input password: 123  
1. Regisgtration  
2.Login in  
2  
Please input user name: zhou  
Please input password: 123  
Login successful  
1:Create  
2:Insert  
3>Delete  
4:Print  
5:Load  
6:Save  
0:Exit  
-
```

创建链表(Create_Stu_Doc()函数)

```
/*新建链表*/
stud_node * Create_Stu_Doc()
{
    struct stud_node * head,*p;
    int num,score;
    char name[20];
    int size = sizeof(stud_node);

    head = NULL;
    printf("Input num,name and score:\n");
    scanf("%d%s%d", &num,name, &score);
    while(num != 0){
        p = (struct stud_node *) malloc(size);
        p->num = num;
        strcpy(p->name, name);
        p->score = score;
        head = InsertDoc(head, p);    /* 调用插入函数 */
        scanf("%d%s%d", &num, name, &score);
    }

    return head;
}
```

我们可以通过在main函数中调用这个函数来创建学生信息链表。

命令描述:

插入结点(InsertDoc函数)

此函数用于增加学生信息



```
127  /* 插入操作 */
128  stud_node * InsertDoc(stud_node * head, stud_node *stud)
129  {
130      struct stud_node *ptr ,*ptr1, *ptr2;
131      ptr2 = head;
132      ptr = stud; /* ptr指向待插入的新的学生记录结点 */
133      /* 原链表为空时的插入 */
134      if(head == NULL){
135          head = ptr; /* 新插入结点成为头结点 */
136          head->next = NULL;
137      }
138      else{ /* 原链表不为空时的插入 */
139          while((ptr->num > ptr2->num) && (ptr2->next != NULL)){
140              ptr1 = ptr2; /* ptr1, ptr2各后移一个结点 */
141              ptr2 = ptr2->next;
142          }
143          if(ptr->num <= ptr2->num){ /* 在ptr1与ptr2之间插入新结点 */
144              if(head == ptr2){
145                  head = ptr;
146              }else{
147                  ptr1->next = ptr;
148              }
149              ptr->next = ptr2;
150          }else{ /* 新插入结点成为尾结点 */
151              ptr2->next = ptr;
152              ptr->next = NULL;
153          }
154      }
155      return head;
156  }
```

命令描述:

修改节点(Modify函数)

此函数用于修改学生信息



*/*修改操作*/*

```
stud_node * Modify(stud_node * head, struct stud_node stu){  
    stud_node *ptr=head;  
    while(ptr!=NULL){  
        if(ptr->num==stu.num){  
            strcpy(ptr->name,stu.name);  
            ptr->score=stu.score;  
        }  
        ptr=ptr->next;  
    }  
    FILE *fp = fopen("students.txt", "w");  
    if (fp == NULL) {  
        perror("修改失败");  
        return head;  
    }  
    ptr=head;  
    while(ptr!=NULL){  
        fprintf(fp,"%5d%20s%5d",ptr->num,ptr->name,ptr->score);  
        ptr=ptr->next;  
        if(ptr!=NULL)  
            fprintf(fp,"\n");  
    }  
    cout<<"Modify successful"<<endl;  
    fclose(fp);  
    return head;  
}
```

删除结点(DeleteDoc函数)

该函数用于删除指定的学生信息



```
/* 删除操作 */
stud_node * DeleteDoc(stud_node * head, int num)
{
    struct stud_node *ptr1, *ptr2;
    /* 要被删除结点为表头结点 */
    while(head != NULL && head->num == num){
        ptr2 = head;
        head = head->next;
        free(ptr2);
    }
    if(head == NULL){ /*链表空 */
        return NULL;
    }
    /* 要被删除结点为非表头结点 */
    ptr1 = head;
    ptr2 = head->next; /*从表头的下一个结点搜索所有符合删除要求的结点 */
    while(ptr2 != NULL){
        if(ptr2->num == num){ /* ptr2所指结点符合删除要求 */
            ptr1->next = ptr2->next;
            free(ptr2);
        }else{
            ptr1 = ptr2; /* ptr1后移一个结点 */
        }
        ptr2 = ptr1->next; /* ptr2指向ptr1的后一个结点 */
    }
    return head;
}
```

命令描述:

输出信息 (Print_Stu_Doc函数)

此函数用于遍历链表
输出学生信息



```
/*遍历操作*/  
void Print_Stu_Doc(stud_node * head)  
{  
    struct stud_node * ptr;  
  
    if(head == NULL){  
        printf("\nNo Records\n");  
        return;  
    }  
    //printf("\nThe Students' Records Are: \n");  
    printf("Num\t Name\t Score\n");  
    for(ptr = head; ptr != NULL; ptr = ptr->next){  
        printf("%d\t%s\t%d \n", ptr->num, ptr->name, ptr->score);  
    }  
}
```

命令描述:

从文件中读取数据 (loadstudents函数)

此函数用于将文件中的学生信息读取到链表中，并计数



```
[stud_node * loadstudents(const char *filename,int *count){
    FILE *fp = fopen(filename, "rb");
    [ if (fp == NULL) {
        perror("读取失败");
        return NULL;
    }
    stud_node *head=NULL,*students=NULL; //指针变量需要赋初值
    int size=sizeof(stud_node);
    //stud_node *students = (stud_node*)malloc(size);
    //head=students;
    int num=0;
    [ while(!feof(fp)){
        [ stud_node *stu = (stud_node*)malloc(size);
            if(head==NULL){
                head=stu;
                students=stu;
            }else
                students->next=stu;

            fscanf(fp,"%d%s%d",&stu->num,&stu->name,&stu->score);
            //printf("%5d%20s%5d\n",stu->num,stu->name,stu->score);
            num++;
            stu->next=NULL;
            students=stu;
        }
        printf("load %d students successful\n",num);
        // 获取文件大小
        fclose(fp);
        //displaystudents(head);
        return head;
    }
}
```

命令描述:

存入文件 (savestudents函数)

此函数用于将链表中的数据存进文件，此处可以是新建的文件

```
void savestudents(stud_node *head, const char *filename){
    FILE *fp = fopen(filename, "w");
    if (fp == NULL) {
        perror("保存失败");
        return;
    }
    while(head != NULL){
        fprintf(fp, "%5d%20s%5d", head->num, head->name, head->score);
        head = head->next;
        if(head != NULL)
            fprintf(fp, "\n");
    }
    cout << "save successful" << endl;
    fclose(fp);
}
```

本课的任务

- 理解代码的逻辑
 - 1.确定规范的目标
 - 2.解构代码架构
 - 3.逐行检查
 - 4.总结与结论
- 在提供的代码库上进行创新
 - 1、功能增强
 - 2、性能优化
 - 3、功能创新
 - 4.我们的基准分数是60分

基本分(60分) 说明:

示例1: 实现注册登录10分

step1:

注册: 将注册信息写入文件并加密, 加密算法 可用凯撒密码

step2:

登录: 验证文件的帐号密码信息

step3:

登录成功反馈Login successful 反之Login failed

优化 改进反馈是什么错误 得10分

样例



```
1. Regisgtration
2.Login in
1
Please input user name: wang
Please input password: 123
1. Regisgtration
2.Login in
2
Please input user name: wang\
Please input password: 123
Login failed
1. Regisgtration
2.Login in
```

基本分(60分) 说明:

示例2: 创建链表10分

step1:

定义结构体类型和对应结构指针

样例



step2:

定义插入函数, 实现链表创建

step3:

输入数据, 直到输入的学号为0结束输入

优化改进: 只输入学号为0时就结束输入 得10分

```
1:Create
2:Insert
3>Delete
4:Print
5:Load
6:Save
0:Exit
1
Input num,name and score:
101 wang 100
102 he 90
103 zhang 80
0 he 11
1:Create
2:Insert
3>Delete
4:Print
5:Load
6:Save
0:Exit
```

基本分(60分) 说明:

示例3: 插入节点和修改节点10分

删除:
输入学号, 删除对应信息

打印:
输入所有学生信息, 显示在屏幕

优化改进: 添加删除的数据验证 得10分

样例

```
104 hh 100
1:Create
2:Insert
3:Modify
4:Delete
5:Print
6:Load
7:Save
0:Exit
2
Input num,name and score:
105 zhang 95
1:Create
2:Insert
3:Modify
4:Delete
5:Print
6:Load
7:Save
0:Exit
5
Num Name Score
101 wang 100
102 he 120
103 li 99
104 hh 100
105 zhang 95
1:Create
```

```
1:Create
2:Insert
3:Modify
4:Delete
5:Print
6:Load
7:Save
0:Exit
5
Num Name Score
101 wang 100
102 he 120
103 li 99
104 hh 100
105 zhang 95
1:Create
2:Insert
3:Modify
4:Delete
5:Print
6:Load
7:Save
0:Exit
3
Please enter the student INFO you wish to modify:
102 he 92
Modify successful
1:Create
2:Insert
3:Modify
4:Delete
5:Print
6:Load
7:Save
0:Exit
5
Num Name Score
101 wang 100
102 he 92
103 li 99
104 hh 100
105 zhang 95
```

基本分(60分) 说明:

示例4: 删除和打印链表10分

删除:
输入学号, 删除对应信息

样例



打印:
输入所有学生信息, 显示在屏幕

优化改进: 添加删除的数据验证 得10分

```
3
Input num:
101
1:Create
2:Insert
3>Delete
4:Print
5:Load
6:Save
0:Exit
4
Num      Name      Score
102      he        90
103      zhang      80
1:Create
2:Insert
3>Delete
4:Print
5:Load
6:Save
0:Exit
```

基本分(60分) 说明:

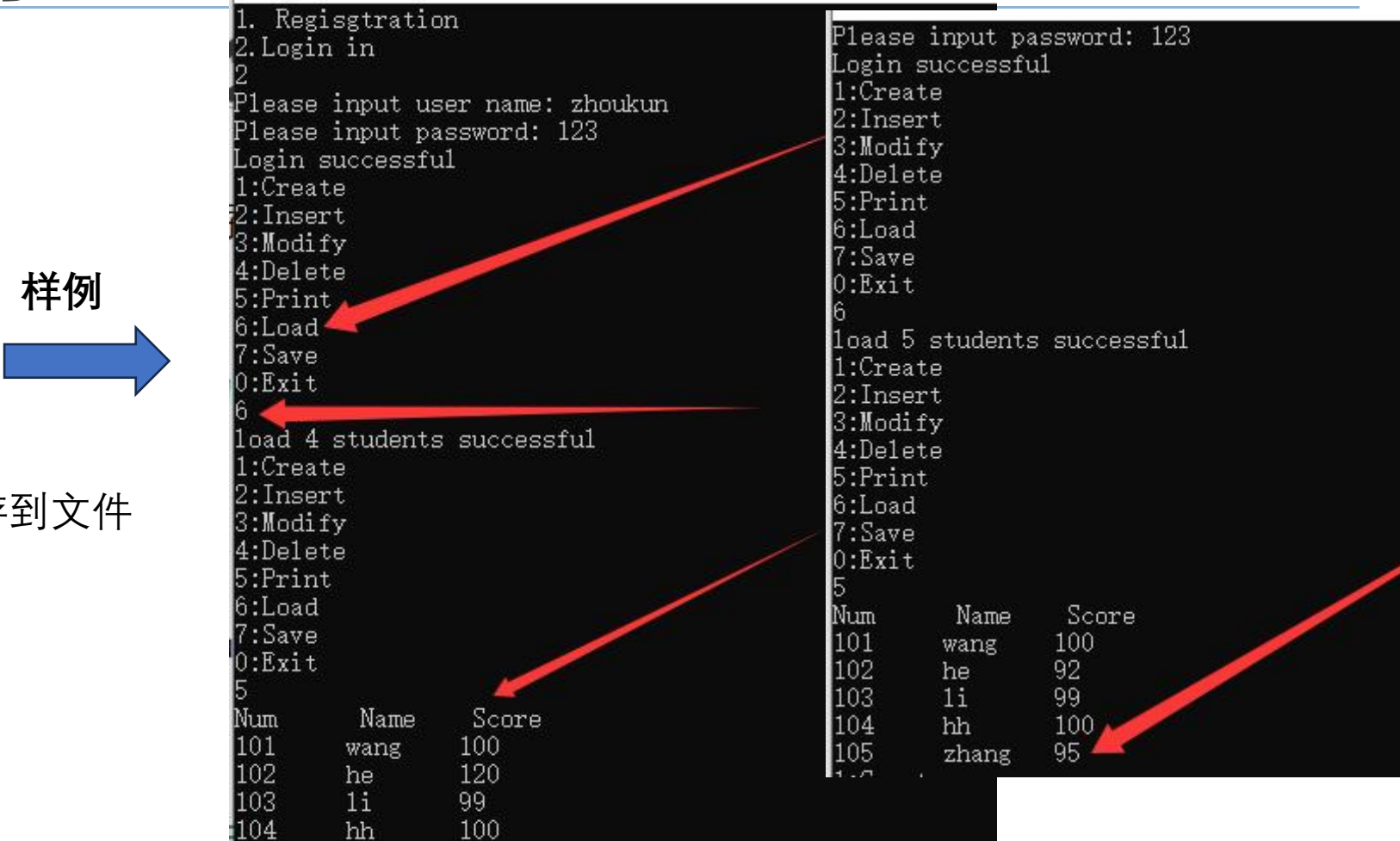
示例5: 加载和保存文件各10分

加载:
将已有的学生文件, 加载到内存中

样例

保存文件:
将操作后的数据链表中的学生信息保存到文件

优化改进: 二进制操作之类 得10分



```
1. Regisgtration
2. Login in
2
Please input user name: zhoukun
Please input password: 123
Login successful
1:Create
2:Insert
3:Modify
4:Delete
5:Print
6:Load
7:Save
0:Exit
6
load 4 students successful
1:Create
2:Insert
3:Modify
4:Delete
5:Print
6:Load
7:Save
0:Exit
5
Num      Name      Score
101      wang      100
102      he        120
103      li         99
104      hh         100
```

```
Please input password: 123
Login successful
1:Create
2:Insert
3:Modify
4:Delete
5:Print
6:Load
7:Save
0:Exit
6
load 5 students successful
1:Create
2:Insert
3:Modify
4:Delete
5:Print
6:Load
7:Save
0:Exit
5
Num      Name      Score
101      wang      100
102      he        92
103      li         99
104      hh         100
105      zhang     95
```

扩展功能

1、扩展功能举例

- (1)实现多文件系统
- (2)添加数据验证（学号唯一性、成绩0-100范围等）
- (3)支持模糊查询（姓名部分匹配）
- (4)奖学金评定
- (5)实现用户注册和登录//这个讲解演示 不算
- (6)使用二进制文件提高存储效率 //这个要对应保存和导入两个模块
- (7)其它自定义功能在提供的代码库上进行创新

2、扩展功能分数是20分

3、所有功能代码质量**10分**。