|  |
| --- |
|  |

**短学期实践实训（总结）报告**

**学 院 名 称： 媒体工程学院**

**课程（项目）名称： 程序能力训练**

**学 号：**

**姓 名：**

**班 级：**

**同 组 同 学：**

**任 课 教 师：**

**学 期： 2025-2026 学年短学期**

**《明日方舟》六星干员管理信息系统及图形化显示**

**1 选题与分工**

A) 意义：选题为《明日方舟》六星干员查询管理系统意义在于供以游戏用户查询了解六星干员的信息；

B) 系统任务：针对现有干员信息构成的CSV文件数据进行增删改查并对该数据以图标和柱状图等统计形式展示在可视化界面；

C) 分工：本人负责C语言部分，实现数据增删改查；\*\*负责Java部分，实现数据可视化。

**2 系统的设计**

C语言部分：

A) 系统刚进入会进入登录注册界面，登录身份分为管理员(admin)和普通用户(user)

|  |
| --- |
|  |
|  |
|  |
|  |

B) 登陆成功即进入菜单界面并等待输入相应编号，同样按admin和user有所区分

|  |
| --- |
|  |
|  |

C) admin添加干员界面

|  |
| --- |
|  |
|  |

D) admin删除干员界面

|  |
| --- |
|  |
|  |
|  |

E) admin修改干员界面

|  |
| --- |
|  |
|  |

F) admin和user身份显示所有干员信息

|  |
| --- |
|  |
|  |

G) admin和user身份查询特定干员信息，多个符合的可选择进一步查看详细信息

|  |  |  |
| --- | --- | --- |
|  | |  |
|  | | |
|  | | |
|  |  | |
|  |  | |

H) user修改密码界面

|  |
| --- |
|  |

I) user注销账户

|  |
| --- |
|  |
|  |

J) admin在每次修改操作后会进一步询问是否再次执行相同操作

|  |
| --- |
|  |

K) 退出操作后界面

|  |
| --- |
|  |

**3 系统的实现**

C语言部分：

A) 数据结构：本部分主要使用到的数据结构是单向链表

B) 核心代码：比较核心的函数是CSV文件数据的读取函数和保存函数

i) 读取函数解读

1. 打开文件“Officer.csv”（已声明为全局常量）以供读取。如果文件无法打开，则打印提示“打开文件失败”，然后返回；

2. 使用**fgets**遍历文件中的每一行，直到到达文件末尾；

3. 对于文件每一行，使用**strtok**按“,”把内容分隔开存储为指针数组，再利用**atoi**和**strncpy**将干员各项数据解析到Staff结构中。这个结构包含干员的编号、姓名、职业、分支、出身、组织、标签、和天赋；

4. 调用add\_node函数后将先调用create\_node函数为新数据分配创建新的node，新node的value设置为传入的干员数据data，并将其next设置为NULL；然后将新节点添加到链表末尾；

5. 完成数据读取后，关闭文件。

|  |  |
| --- | --- |
| 01  02  03  04  05  06  07  08  09  10  11  12 | /\*干员数据存储结构体\*/  **typedef** **struct** Staff {  **int** id;  **char** name[MAX\_NAME];  **char** career[MAX\_CAREER];  **char** branch[MAX\_BRANCH];  **char** background[MAX\_BACKGROUND];  **char** organization[MAX\_ORGANIZATION];  **int** cost;  **char** tag[MAX\_TAG];  **char** talent[MAX\_TALENT];  }Staff; |
| 01  02  03  04  05  06  07  08  09  10  11  12  13  14  15  16  17  18  19  20  21  22  23  24  25  26  27  28  29  30  31  32  33  34  35  36  37  38  39  40  41  42  43  44  45  46  47  48  49  50  51  52  53  54 | /\*读取源数据\*/  **void** read\_csv(Node\*\* head, **const** **char**\* filename) {  **FILE**\* file = **fopen**(filename, "r");  **if** (!file) {  **perror**("!#\*> 打开文件失败...");  **return**;      }    **char** line[1024];  **fgets**(line, **sizeof**(line), file);    //跳过标题行    **while** (**fgets**(line, **sizeof**(line), file)) {          Staff s = { 0 };  **char**\* tokens[9];  **int** token\_count = 0;            // 分割行  **char**\* token = **strtok**(line, ",");  **while** (token != NULL && token\_count < 9) {              tokens[token\_count++] = token;              token = **strtok**(NULL, ",\n");          }    **if** (token\_count < 8) **continue**;  // 跳过无效行            // 解析字段          s.id = **atoi**(tokens[0]);  **strncpy**(s.name, tokens[1], MAX\_NAME - 1);          s.name[MAX\_NAME - 1] = '\0';  **strncpy**(s.career, tokens[2], MAX\_CAREER - 1);          s.career[MAX\_CAREER - 1] = '\0';  **strncpy**(s.branch, tokens[3], MAX\_BRANCH - 1);          s.branch[MAX\_BRANCH - 1] = '\0';  **strncpy**(s.background, tokens[4], MAX\_BACKGROUND - 1);          s.background[MAX\_BACKGROUND - 1] = '\0';  **strncpy**(s.organization, tokens[5], MAX\_ORGANIZATION - 1);          s.organization[MAX\_ORGANIZATION - 1] = '\0';          s.cost = **atoi**(tokens[6]);  **strncpy**(s.tag, tokens[7], MAX\_TAG - 1);          s.tag[MAX\_TAG - 1] = '\0';            // 处理天赋字段（可能包含逗号）  **if** (token\_count > 8) {  **strncpy**(s.talent, tokens[8], MAX\_TALENT - 1);              s.talent[MAX\_TALENT - 1] = '\0';          }  **else** {              s.talent[0] = '\0';          }            add\_node(head, s);      }  **fclose**(file);  } |
| 1  2  3  4  5  6  7  8  9 | /\*创建新节点\*/  Node\* create\_node(Staff data) {      Node\* new\_node = (Node\*)**malloc**(**sizeof**(Node));  **if** (new\_node) {          new\_node->value = data;          new\_node->next = NULL;      }  **return** new\_node;  } |
| 01  02  03  04  05  06  07  08  09  10  11  12  13  14 | /\*新增节点\*/  **void** add\_node(Node\*\* head, Staff data) {      Node\* new\_node = create\_node(data);  **if** (\*head == NULL) {          \*head = new\_node;  **return**;      }        Node\* current = \*head;  **while** (current->next != NULL) {          current = current->next;      }      current->next = new\_node;  } |

ii) 保存函数解读

1. 以写入模式（“w”）打开“Officer.csv”文件。如果文件无法打开，则打印提示“打开文件失败”，然后返回；

2. 先写入标题行内容（已声明为全局常量），然后初始化一个指针current指向头节点。然后进入一个循环，遍历链表直到current指向NULL；

4. 在循环体内，使用**fprintf**函数将当前节点中的干员信息格式化输出到文件中。每条记录包括干员的编号、姓名、职业、分支、出身、组织、标签、和天赋，每条记录后跟一个换行符；

5. 更新current指针，使其指向下一个节点；

8. 循环结束后，使用**fclose**函数关闭文件，最后提示“数据已保存”。

|  |  |
| --- | --- |
| 01  02  03  04  05  06  07  08  09  10  11  12  13  14  15  16  17  18  19 | /\*保存数据\*/  **void** save\_csv(Node\* head, **const** **char**\* filename) {  **FILE**\* file = **fopen**(filename, "w");  **if** (!file) {  **perror**("!#\*> 保存文件失败...\n");  **return**;      }  **fprintf**(file, "%s\n", title\_line);      Node\* current = head;  **while** (current != NULL) {          Staff s = current->value;  **fprintf**(file, "%d,%s,%s,%s,%s,%s,%d,%s,%s\n",              s.id, s.name, s.career, s.branch, s.background,              s.organization, s.cost, s.tag, s.talent);          current = current->next;      }  **fclose**(file);  **printf**("$\*> 数据已保存...\n");  } |

iii) 其余部分函数实现都基于对数据链表的增删改查操作；

C) 补充：用户账户的数据操作类似；为防止误操作导致数据丢失，在每次修改操作执行成功后，将自动执行一次保存操作

D) 本部分遇到的困难较少，较有困难的部分是登陆注册时的密码隐式显示，经查阅资料后得到有效解决，并成功扩展为用户名长度和密码长度限制函数。

|  |  |
| --- | --- |
| 01  02  03  04  05  06  07  08  09  10  11  12  13  14  15  16  17  18  19  20  21  22  23  24  25  26  27  28  29  30  31  32  33  34  35  36  37  38  39  40  41  42  43  44  45  46  47  48  49  50  51  52  53  54  55  56  57  58  59  60  61  62  63  64 | /\*读取键入的用户名或密码\*/  **char**\* get\_name\_password(**int** flag) {      //@flag 1代表用“\*”显示密码 2代表明文显示密码 3代表显示用户名    **if** (flag == 1 || flag == 2) {  **char** password[MAX\_PASSWORD];  **int** i = 0;  **char** ch;    **while** (1) {              ch = \_getch();    **if** (ch == 13) {  **break**;              }  **else** **if** (ch == 8) {  **if** (i > 0) {  **printf**("\b \b");                      i--;                  }              }  **else** **if** (ch != ' ' && i < MAX\_PASSWORD - 1) {  **if** (flag == 1) {  **printf**("\*");                      password[i] = ch;                      i++;                  }  **else** **if** (flag == 2) {  **printf**("%c", ch);                      password[i] = ch;                      i++;                  }              }          }          password[i] = '\0';  **return** password;      }  **else** **if** (flag == 3) {  **char** name[MAX\_USER\_NAME];  **int** i = 0;  **char** ch;    **while** (1) {              ch = \_getch();    **if** (ch == 13) {  **break**;              }  **else** **if** (ch == 8) {  **if** (i > 0) {  **printf**("\b \b");                      i--;                  }              }  **else** **if** (ch != ' ' && i < MAX\_USER\_NAME - 1) {  **printf**("%c", ch);                  name[i] = ch;                  i++;              }          }          name[i] = '\0';  **return** name;      }  } |

**4 项目总结**

项目虽然相对简陋但是基本要求功能已然全部实现，通过这次项目实践对面向对象的编程思想认知更加深刻，对C程序与文件等信息交互了解更深，尤其是在编码方面理解更加深入，做了一个看似简单的项目但是各种方法面收获颇丰，坚定了我们对学完知识就要做项目实践的学习方法，实践才能快速将知识融会贯通。