# 函数管理系统文档

## 概述

本系统是一个基于 C 语言的函数管理工具，旨在帮助用户管理函数的相关信息。系统通过文件存储函数数据，支持添加、删除、修改、查询和显示函数信息等功能，适合需要对函数进行规范化管理的场景。

## 功能说明

### 1. 添加函数信息

\*\*功能描述\*\*：录入新函数的信息，包括编号、函数名、函数原型、功能描述、返回值和说明，并保存到文件。

\*\*实现函数\*\*：`add()`

\*\*文件操作\*\*：

以追加模式打开 `function.txt`。

将用户输入的信息追加写入文件。

### 2. 删除函数信息

\*\*功能描述\*\*：删除全部函数信息，根据编号删除特定函数。

\*\*实现函数\*\*：

del\_all()：清空文件内容。

del()：根据用户输入的编号删除对应记录，并更新其他记录的编号。

\*\*文件操作\*\*：

使用临时文件 `function1.txt` 保存未删除的记录。

删除原始文件 `function.txt`，并重命名临时文件。

### 3. 修改函数信息

\*\*功能描述\*\*：

按编号修改函数信息。

按名称（支持部分匹配）修改函数信息。

\*\*实现函数\*\*：

id\_revise()：通过用户输入的编号精确定位函数，并修改其信息。

name\_revise()：通过部分匹配查找函数，用户选择需要修改的记录并更新。

\*\*文件操作\*\*：同样使用临时文件function1.txt进行操作，替换原文件。

### 4. 查询函数信息

\*\*功能描述\*\*：

按编号精确查询函数信息。

按名称（支持部分匹配）查询函数信息。

\*\*实现函数\*\*：

find\_id()：精确匹配编号并输出信息。

find\_prototype()：通过 strstr()进行部分匹配，输出所有符合条件的记录。

\*\*文件操作\*\*：以只读模式打开 function.txt，逐行匹配。

### 5. 显示全部函数信息

\*\*功能描述\*\*：读取并显示function.txt中的所有记录。

\*\*实现函数\*\*：display()

\*\*文件操作\*\*：逐行读取文件内容并输出。

### 6. 退出系统

\*\*功能描述\*\*：退出系统。

\*\*实现函数\*\*：通过 `exit(0)` 实现。

## 文件结构

### 头文件：`test.h`

定义核心数据结构 function：

typedef struct {

int id; // 函数编号

char easy\_name[50]; // 函数名

char name[50]; // 函数原型

char function[50]; // 函数功能

char my\_return[50]; // 返回值

char illustrate[50]; // 说明

} function;

声明所有功能函数。

### 主程序：`main.c`

\*\*主要功能\*\*：

调用 menu() 显示用户菜单。

通过switch-case 结构处理用户输入并调用相应的功能函数。

### 功能实现：`test.c`

\*\*包含函数\*\*：实现了所有功能模块的具体逻辑，包括文件操作和用户交互。

## 数据存储

### 文件格式：`function.txt`

每条记录存储为一行，字段之间用空格分隔，示例：

1 printf int\_printf(const\_char\*\_format,...) Outputs\_formatted\_text\_to\_standard\_output integer Indicates\_number\_of\_characters\_written

2 scanf int\_scanf(const\_char\*\_format,...) Reads\_formatted\_input\_from\_standard\_input integer Indicates\_number\_of\_items\_read

字段顺序：编号 函数名 函数原型 函数功能 返回值 说明。

注意：各个字符串之间以一个空格隔开，且不支持中文，英文语句说明中的空格以下划线‘\_’代替。程序运行需要管理员权限以完成对文件的读写，且系统环境及数据文件均使用utf-8编码以避免乱码。

### 临时文件

在修改和删除操作中，使用临时文件 `function1.txt` 存储中间结果，避免直接操作原文件。

## 用户界面

### 菜单结构

通过 `menu()` 函数打印操作选项：

欢迎使用函数管理系统

1. 添加函数信息

2. 删除函数信息

3. 修改函数信息

4. 查询函数信息

5. 显示函数信息

6. 退出系统

请选择你要进行的操作：

\*\*用户通过输入对应数字选择功能。

## 异常处理

1. \*\*输入验证\*\*：

使用 num()函数检查用户输入是否为数字。

对于非法输入，提示用户重新输入。

2. \*\*文件操作错误\*\*：

检查文件是否成功打开，失败时输出错误信息并退出对应操作。

3. \*\*记录不存在\*\*：

在查询或修改时，提示用户未找到匹配记录。

##编译说明

使用cmake进行项目管理，vscode进行代码编写，gcc编译，utf-8编码。项目已上传github，地址：github.com/3052309087/project。