## 学生成绩管理系统开发文档

**背景及目的：**本系统只为练习多文档编程、团队分工合作，还有复习巩固近期所学c语言知识所用。

### 1、系统功能：

面向三个对象，校长、老师、学生

校长：账号唯一（且只有一个，固定）

功能： 1、显示在校教师和离校教师

2、重置自己和教师的密码（重置为123456）

3、添加、删除老师

4、解锁老师账号

5、退出

老师：账号唯一（编号即为账号，学号唯一）

功能： 1、添加学生（学生的学号自动生成）(单个添加 批量导入)

2、删除学生 （输入两次学号确定 删除的学生要记录保存在已退学的文件中

3、查找学生（按姓名查找 按学号查找）

4、修改学生信息（修改学生基础信息 修改学生成绩）

5、录入学生成绩（单个录入 批量导入）

6、重置学生密码

7、解锁学生账号

8、显示所有在校学生信息和显示所有退出学生信息

9、修改个人密码

10、退出

学生：账号唯一（编号即为账号，学号唯一）

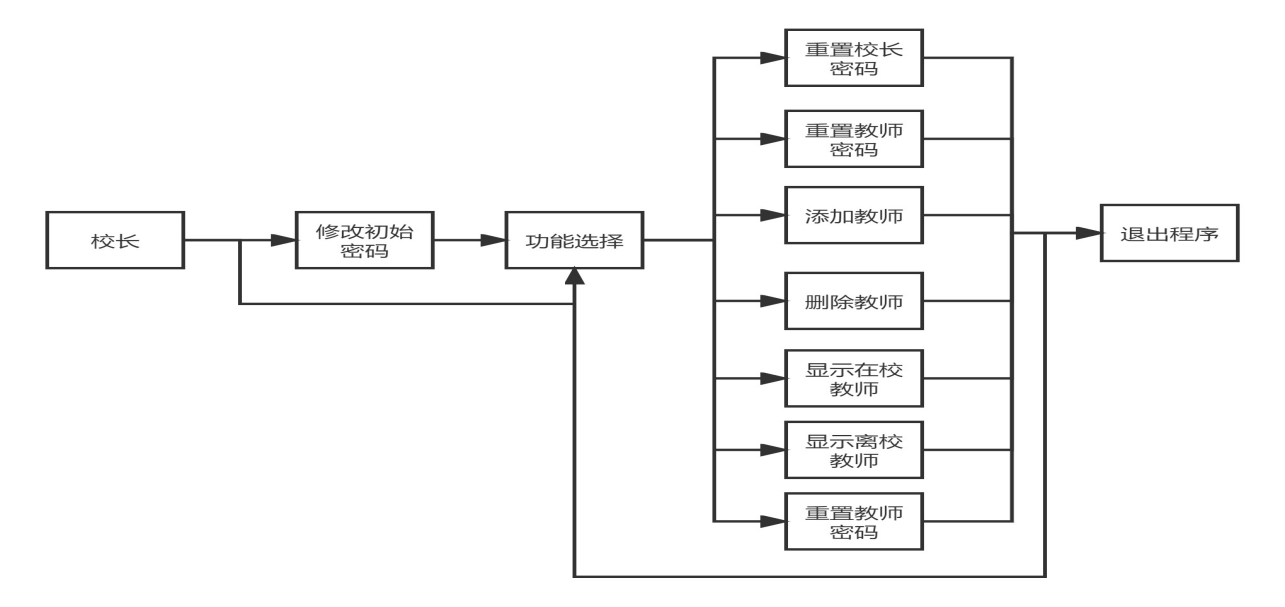
1. 查询成绩
2. 修改个人密码
3. 查看个人信息
4. 退出

共同要求：第一次登录都需要修改密码且不能为初始化密码。

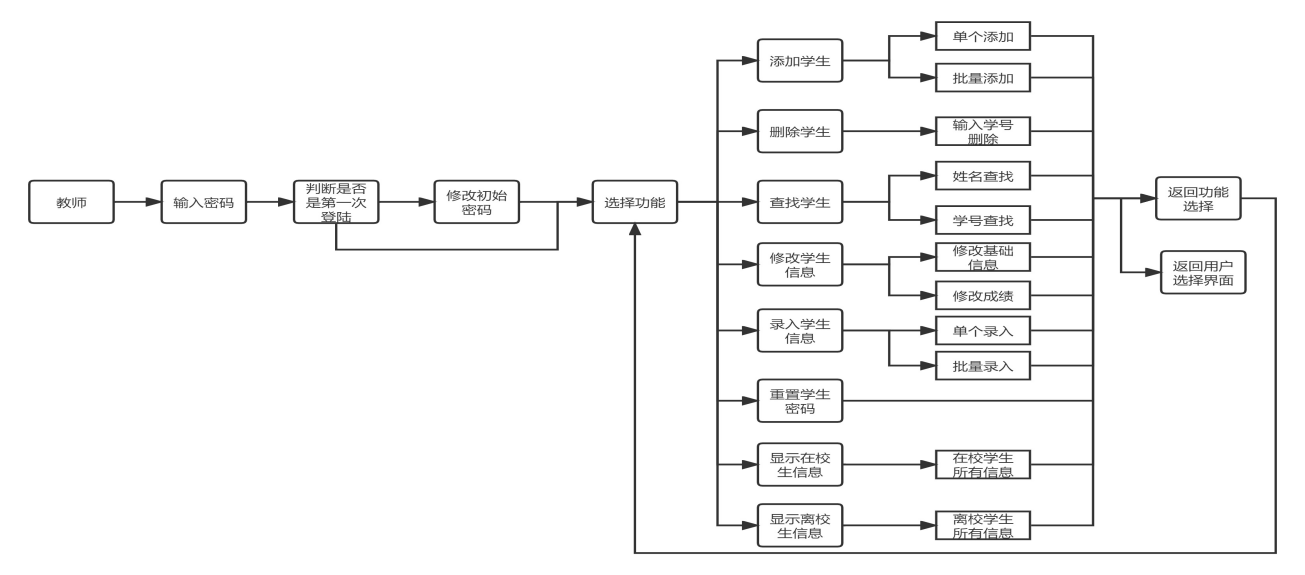
### 功能流程：

目前为1.0版本，只能单机，不能联网，界面未优化。

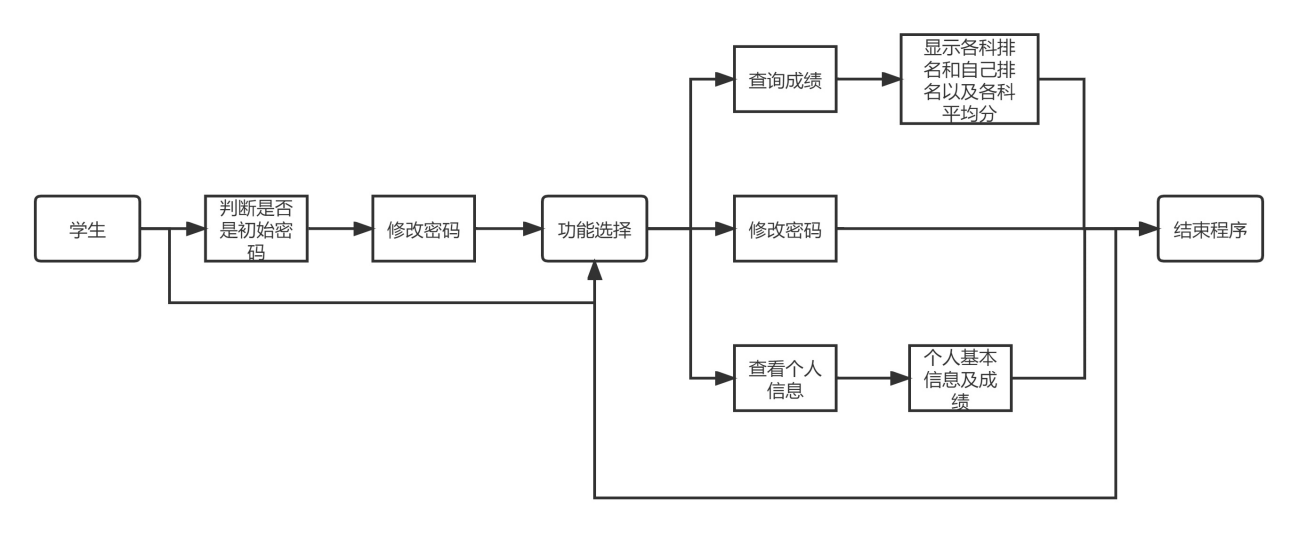
校长功能流程图：



教师功能流程图：



学生功能流程图：



### 各个功能--分工

校长：王淳隽

教师：李超泽、夏泽隆

学生：徐志海

框架：徐志海

ppt：夏泽隆，李超泽、王淳隽

文档：夏泽隆

测试与修改：夏泽隆、徐志海、李超泽

校长功能由王淳隽实现：

case '1':reset\_me(); break; //重置自己密码

case '2':reset\_t(); break; //重置教师密码

case '3':add\_tch(); break; //添加教师

case '4':del\_tch(); break; //删除教师

case '5':show\_tch(); break; //显示所有在职教师

case '6':show\_lev(); break; //显示所有离职教师

case '7':unlock(); break; //解锁教师账号

case '0':return; //退出程序

教师功能由夏泽隆和李超泽共同实现：

case '1': add\_stu(); break;

case '2': delete\_stu(); break;

case '3': find\_stu(); break;

case '4': shout\_stu(); break;

case '5': input\_stu(); break;

case '6': reset\_stu(); break;

case '7': show\_all\_in(); break;

case '8': show\_all\_out(); break;

case '9': unlock\_tea(); break;

case 'a': change\_tea\_password(); break;

case '0': return ;

学生功能由徐志海实现：

case '1':

//显示成绩

show\_grade();

break;

case '2':

//修改密码

change\_password();

break;

case '3':

//查看个人信息

look\_information();

break;

case '0':

//退出学生端

return;

}

框架--徐志海：

Teacher\* Tea\_in=NULL;

Teacher\* Tea\_out=NULL;

Student\* Stu\_in=NULL;

Student\* Stu\_out=NULL;

int Ret\_stu\_in; //在校生文件总人数

int Ret\_stu\_out; //离校生文件总人数

int Ret\_tea\_in; //在校老师文件总人数

int Ret\_tea\_out; //离校老师文件总人数

int\* Now\_teacher\_no=NULL; //存储老师工号的变量

int\* Now\_student\_no=NULL; //存储学生学号的变量

char\* Admin\_password=NULL; //存储校长密码的变量

//显示界面

static int show\_table(void)

{

system("clear");

printf("\*\*\*\*\*\*欢迎使用教务系统\*\*\*\*\*\*\n");

printf("请选择你的登录身份\n");

printf("\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\n");

printf("1、学生\n");

printf("2、老师\n");

printf("3、校长\n");

printf("0、退出程序\n");

printf("\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\n");

printf("请输入指令:");

char cmd=getch();

printf("%c\n",cmd); //回显

return cmd;

}

//将文件中的数据导入到变量中，初始化数据

bool init\_program(void)

{

//以只读的方式打开文件

FILE\* fp\_admin=fopen("admin.txt","r");

FILE\* fp\_stu\_in=fopen("stu\_in.txt","r");

FILE\* fp\_stu\_out=fopen("stu\_out.txt","r");

FILE\* fp\_tea\_in=fopen("tea\_in.txt","r");

FILE\* fp\_tea\_out=fopen("tea\_out.txt","r");

//判断是否成功打开文件

if(fp\_admin==NULL||fp\_stu\_in==NULL||fp\_stu\_out==NULL||fp\_tea\_in==NULL||fp\_tea\_out==NULL)

{

perror("\_\_func\_\_,fopen");

return false;

}

//申请内存

Now\_teacher\_no=malloc(4);

Now\_student\_no=malloc(4);

Admin\_password=malloc(13);

Stu\_in=malloc(sizeof(Student)\*MAX\_STUDENT);

Stu\_out=malloc(sizeof(Student)\*MAX\_STUDENT);

Tea\_in=malloc(sizeof(Teacher)\*MAX\_TEACHER);

Tea\_out=malloc(sizeof(Teacher)\*MAX\_TEACHER);

//从文件中读取数据并写入

fscanf(fp\_admin,"%d%d%s",Now\_teacher\_no,Now\_student\_no,Admin\_password);

int ret=1;

//读取在校学生信息到内存中

while(ret>0)

{

ret=fscanf(fp\_stu\_in,"%s %c %d %s %hhd %lf %lf %lf\n",

Stu\_in[Ret\_stu\_in].name,&Stu\_in[Ret\_stu\_in].sex,

&Stu\_in[Ret\_stu\_in].no,Stu\_in[Ret\_stu\_in].password,

&Stu\_in[Ret\_stu\_in].state,&Stu\_in[Ret\_stu\_in].chinese,

&Stu\_in[Ret\_stu\_in].math,&Stu\_in[Ret\_stu\_in].english);

Ret\_stu\_in++;

}

//读取离校学生信息到内存中

ret=1;

while(ret>0)

{

ret=fscanf(fp\_stu\_out,"%s %c %d %s %hhd %lf %lf %lf\n",

Stu\_out[Ret\_stu\_out].name,&Stu\_out[Ret\_stu\_out].sex,

&Stu\_out[Ret\_stu\_out].no,Stu\_out[Ret\_stu\_out].password,

&Stu\_out[Ret\_stu\_out].state,&Stu\_out[Ret\_stu\_out].chinese,

&Stu\_out[Ret\_stu\_out].math,&Stu\_out[Ret\_stu\_out].english);

Ret\_stu\_out++;

}

//读取在职老师信息到内存中

ret=1;

while(ret>0)

{

ret=fscanf(fp\_tea\_in,"%s %c %d %s %hhd\n",

Tea\_in[Ret\_tea\_in].name,&Tea\_in[Ret\_tea\_in].sex,

&Tea\_in[Ret\_tea\_in].no,Tea\_in[Ret\_tea\_in].password,

&Tea\_in[Ret\_tea\_in].state);

Ret\_tea\_in++;

}

//读取离职老师信息到内存中

ret=1;

while(ret>0)

{

ret=fscanf(fp\_tea\_out,"%s %c %d %s %hhd\n",

Tea\_out[Ret\_tea\_out].name,&Tea\_out[Ret\_tea\_out].sex,

&Tea\_out[Ret\_tea\_out].no,Tea\_out[Ret\_tea\_out].password,

&Tea\_out[Ret\_tea\_out].state);

Ret\_tea\_out++;

}

//关闭文件，并将文件指针置空

fclose(fp\_tea\_out);

fclose(fp\_tea\_in);

fclose(fp\_stu\_in);

fclose(fp\_stu\_out);

fp\_tea\_out=NULL;

fp\_tea\_in=NULL;

fp\_stu\_in=NULL;

fp\_stu\_out=NULL;

return true;

}

//开始程序

void start\_program(void)

{

for(;;) //死循环进入界面

{

switch(show\_table()) //界面显示并接收键值

{

case '1':

student(); //学生端

break;

case '2':

teacher(); //教师端

break;

case '3':

admin(); //校长端

break;

case '0': //退出循环

return;

}

}

}

//结束程序

void end\_program(void)

{

//以只写的方式打开文件

FILE\* fp\_stu\_in=fopen("stu\_in.txt","w");

FILE\* fp\_stu\_out=fopen("stu\_out.txt","w");

FILE\* fp\_tea\_in=fopen("tea\_in.txt","w");

FILE\* fp\_tea\_out=fopen("tea\_out.txt","w");

FILE\* fp\_admin=fopen("admin.txt","w");

//判断是否成功打开文件

if(fp\_admin==NULL||fp\_stu\_in==NULL||fp\_stu\_out==NULL||fp\_tea\_in==NULL||fp\_tea\_out==NULL)

{

perror("\_\_func\_\_,fopen");

return;

}

//将全局变量中的数据写入文件中

fprintf(fp\_admin,"%d %d %s",\*Now\_teacher\_no,\*Now\_student\_no,Admin\_password);

//将在校学生信息写入文件中

for(int i=0;i<Ret\_stu\_in-1;i++)

{

fprintf(fp\_stu\_in,"%s %c %d %s %hhd %lf %lf %lf\n",

Stu\_in[i].name,Stu\_in[i].sex,

Stu\_in[i].no,Stu\_in[i].password,

Stu\_in[i].state,Stu\_in[i].chinese,

Stu\_in[i].math,Stu\_in[i].english);

}

//将离校学生信息学入文件中

for(int i=0;i<Ret\_stu\_out-1;i++)

{

fprintf(fp\_stu\_out,"%s %c %d %s %hhd %lf %lf %lf\n",

Stu\_out[i].name,Stu\_out[i].sex,

Stu\_out[i].no,Stu\_out[i].password,

Stu\_out[i].state,Stu\_out[i].chinese,

Stu\_out[i].math,Stu\_out[i].english);

}

//将在校老师信息写入文件中

for(int i=0;i<Ret\_tea\_in-1;i++)

{

fprintf(fp\_tea\_in,"%s %c %d %s %hhd\n",

Tea\_in[i].name,Tea\_in[i].sex,

Tea\_in[i].no,Tea\_in[i].password,

Tea\_in[i].state);

}

//将离校老师信息写入文件中

for(int i=0;i<Ret\_tea\_out-1;i++)

{

fprintf(fp\_tea\_out,"%s %c %d %s %hhd\n",

Tea\_out[i].name,Tea\_out[i].sex,

Tea\_out[i].no,Tea\_out[i].password,

Tea\_out[i].state);

}

//关闭文件，并将文件指针置空

fclose(fp\_tea\_out);

fclose(fp\_tea\_in);

fclose(fp\_stu\_in);

fclose(fp\_stu\_out);

fp\_tea\_out=NULL;

fp\_tea\_in=NULL;

fp\_stu\_in=NULL;

fp\_stu\_out=NULL;

测试：

数据有：字母、字符串、限制条件外的字符或字符串

超长的字符串、数字+字符

输入定义类型外的类型等

### 版本更新与缺陷

此系统调试测试期有多个内测版本，现在版本为内测最终版，需求功能基本实现，暂未发现重大bug。后续学习其他qt、界面等技术后，可能会继续优化。

此系统缺陷：文件的读出写入，系统没有正常退出数据不会写入，文件中如果有错误数据写入会出错

代码：封装性不够，还有大量重复代码

文件：源文件太多

功能：对输入输出的限制还存在问题

界面：界面简陋

加密：密码加密简单