

# 北京交通大学 计算机实验报告

## 实验报告（lab5）

年    级：大学一年级  
学    号：20281274  
姓    名：杜海玮  
专    业：计算机科学  
任课老师：邹琪

二零二一年五月

## 目 录

1) 程序实现.....	1
A) 外部调用实验 4 程序 (system 函数) .....	1
B) 设计文件目录关系.....	1
C) 实验五配置文件.....	3
D) 自检函数设计与相对位置阐述.....	3
2) 综述与学习：程序设计文档.....	4
A) 程序设计文档的要求.....	4
B) 程序设计文档的目的.....	5
3) 实验总结.....	5
A) 实验总结.....	5
B) 对小组其他成员的文档审查报告.....	6

## 1) 程序实现

### A) 外部调用实验 4 程序 (system 函数)

#### 1.1 函数声明与作用

```
int AssCommand(int type, int mode) { //type=0为文本文件, 1为
    if (mode == 0) {
        if (type == 0) {
            system("lab3.exe r");
        }
    }
}
```

图 1-1 system 函数 (读取字符串)

System 函数默认在当前目录中充当命令行角色, 如图, 我使用该调用当前目录中的 lab4 文件, 使用 random 条数据记录。它也接受字符串变量命令, 如下图:

```
strcat_s(path, 1000, name);
strcpy_s(Lab4DefaultPathDat, 1000, path);
strcat_s(Lab4DefaultPathDat, 1000, ".dat");
char command[2200] = "lab3.exe r ";
strcat_s(command, 1000, path);
char dat[10] = " d";
strcat_s(command, 1000, dat);
system(command);
```

图 1-2 system 函数 (读取字符串变量)

### B) 设计文件目录关系:

DataSet	2021/5/25 16:34	文件夹	
Debug	2021/5/25 17:04	文件夹	
OutputData	2021/5/12 22:01	文件夹	
conf.ini	2021/5/25 15:11	配置设置	1 KB
lab3.exe	2021/5/25 15:26	应用程序	79 KB
Lab5.cpp	2021/5/8 17:12	C++ Source	2 KB
Lab5.vcxproj	2021/5/21 15:53	VC++ Project	8 KB
Lab5.vcxproj.filters	2021/5/25 16:29	VC++ Project Fil...	3 KB
Lab5.vcxproj.user	2021/5/8 15:19	Per-User Project...	1 KB
Lab5_call.cpp	2021/5/25 16:37	C++ Source	3 KB
Lab5_call.h	2021/5/9 17:16	C/C++ Header	1 KB
Lab5_conf.cpp	2021/5/23 9:42	C++ Source	5 KB
Lab5_conf.h	2021/5/23 22:36	C/C++ Header	2 KB
Lab5_conf.ini	2021/5/25 16:40	配置设置	1 KB
Lab5_data.h	2021/5/25 14:55	C/C++ Header	1 KB
Lab5_end.cpp	2021/5/23 22:36	C++ Source	2 KB
Lab5_end.h	2021/5/23 23:00	C/C++ Header	1 KB
Lab5_loadfile.cpp	2021/5/25 17:04	C++ Source	9 KB
Lab5_loadfile.h	2021/5/24 23:02	C/C++ Header	1 KB
Lab5_main.cpp	2021/5/25 14:55	C++ Source	8 KB
Lab5_main.h	2021/5/8 16:49	C/C++ Header	1 KB
Lab5_support.cpp	2021/5/8 16:49	C++ Source	1 KB
Lab5_support.h	2021/5/8 17:12	C/C++ Header	1 KB
Lab5_supportt.h	2021/5/8 16:49	C/C++ Header	1 KB
Lab5_view.cpp	2021/5/24 22:04	C++ Source	3 KB
Lab5_view.h	2021/5/23 22:36	C/C++ Header	1 KB

图 1-2 文件目录关系

我将四个重要子文件/目录置于 lab5 工作目录下，这样，各个程序可以方面的直接调用彼此而不必使用 chdir 命令。

```

char txt[10] = ".txt";
char dat[10] = ".dat";
char FullPath[1000];
int acc = _access(Stat.filesavepath, 0);
if (acc == 0) { // 目录存在
    _chdir(Stat.filesavepath);
    strcpy_s(FullPath, Stat.filesavepath);
    strcat_s(FullPath, Stat.filename);
    if (Stat.FileType == 't') {
        strcat_s(Stat.filename, txt);
    }
    if (Stat.FileType == 'd') {
        strcat_s(Stat.filename, dat);
    }
}

```

图 1-3 后缀名按用户指定加入

#### C) 实验五配置文件:

如图 1-2, 实验五配置文件是 Lab5\_conf.ini, 实验 4 配置文件是 conf.ini, 他们共处 lab5 程序根目录下, 即相对根目录为空, 可以直接用 system 命令调用。方便调用。Lab5\_conf.ini 中或 1 或 0, 0 代表自动模式, 1 代表手动模式。

#### D) 自检函数设计与相对位置阐述

```

/*****
函数名称:
    系统自检函数
功能描述:
    检查系统必备的配置文件与lab4执行文件是否就位
函数参数:
    无
返回值:
    0为要素未就绪, 1为准备完毕
模块历史:
    杜海玮于2021年5月9日创建本模块, email:20281274@bjtu.edu.
*****/

int sysCheck() {
    if(_access("lab3.exe", 0))return 0;
    if (_access("Lab5_conf.ini", 0))return 0;
    if (_access("conf.ini", 0))return 0;
    chdir("Lab5");
    return 1;
}

```

图 1-4 系统自检函数

如图所示,系统自检函数将检查 lab4 程序是否存在于指定位置,lab4 和 lab5 配置文件是否存在于指定位置。

众所周知文件路径有相对位置与绝对位置两种,而采用相对位置能从当前工作目录出发导向目标目录,在实验 5 中,我使用了相对目录定位实验 4 生成的数据记录文件,即 DataSet\DataFile.txt (或 DataFile.txt),这样,在自动模式下我的读取函数就能准确抓取目标数据文件,配置文件同理也可准确读取。

## 2) 综述与学习：程序设计文档

### A) 程序设计文档的要求

程序设计文档包括了程序的设计要求、程序实现目标、程序各子文件细节、程序各函数细节等关于程序各个功能实现的抽象要求与标准。该文档起到了程序开发的指导作用。

## B) 程序设计文档的目的

我认为其中程序的设计要求、程序实现目标、程序各子文件要求应该在程序主体尚未动工时明确，这样才能更好的从全局角度编写代码。而其中的一些细节性记录则应随着程序的编写逐步完善，通过总结性的表格与对比统计，可以从程序本身浩瀚的代码中抽象出逻辑本质，进一步精炼设计思路。这样可以加快程序编写速度，加强程序编写质量。

综合来说，程序设计文档在程序开发中起到了提纲挈领的作用，它的编写与程序编写同步，它们互为表里，相辅相成。

## 3) 实验总结

### A) 实验总结

#### 1. 实验收获：

通过本次实验我实践了链表的创建、初始化、加载数据与释放内存操作，熟悉了这一数据结构的运用。

通过本实验我深刻认识了多源文件在工程中的应用，通过良好的分类与代码管理，我的程序保持了一个清晰的结构，增强了可读性。

通过本实验我熟悉了 `system` 函数的使用，掌握了其两种输入方式的用法，并成功多次调用 `lab4` 程序以完成 `lab5` 要求。

通过本实验我深刻认识了二进制文件的读写机制，完善了对 `fopen` 函数的 `w`、`r`、`r+`、`wb` 和 `rb` 模式的认识。

通过本实验我认识了文件的相对路径在多文件管理、写入、读取中的作用，它方便地简化了程序操作。

通过本实验我认识了二维数组、结构体数组、指针数组与链表这四种数据结构的具体存在形式，加深了对它们的理解。

#### 2. 实验困难：

对二维数组的指针形式认识模糊：通过查阅资料、咨询同学等方式完善了对二维数组指针形式的理解，认识了该形式的灵活性。

在读取数据时报告内存不足：通过仔细阅读代码，发现是使用了函数局部变量形式的分配内存方式，这样导致了一旦函数关闭则分配的内存消失。解决方法是再增加一层指针，例如二维数组的输入参数形式为\*\*\*int，这样就可以像以往那样直接在数据的真实地址上修改数据，分配内存。其中的嵌套关系较难理解，需要多实践掌握。

在读取二进制数据文件时出现乱码：通过查阅资料、询问同学、请教老师等方式发现是写入与读写不规范，fopen 的二进制写入模式应该是 wb，读取模式应该是 rb，使用其他模式会造成乱码，指针停滞不前，干脆利落的报错等情况。

在打开文件时报文件指针为空：通过仔细审查代码，发现是前面调用同样文件时没有进行关闭操作，这种低级错误仍然犯，需要警醒。

## B) 对小组其他成员的文档审查报告

陈明强同学：他的函数整合了相对于我的子函数，在读写上没有我的中转站结构直接读取，更加快速，但缺少鲁棒性。

高原同学：free 函数的循环操作更合理，函数注释上更详细。