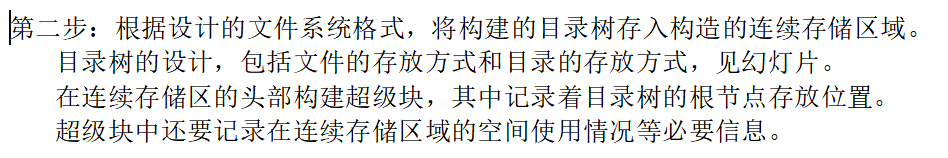
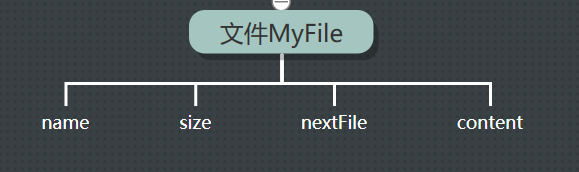
## 课程设计优缺点记录

**缺点：**

1.没有使用超级块存储管理系统文件：



在程序实现中，设计有目录树结构，主要在多级连接的文件结构体中没有考虑这一属性，没有包含超级存储块，在结构体MyFile中，包含有以下属性：



name:文件名

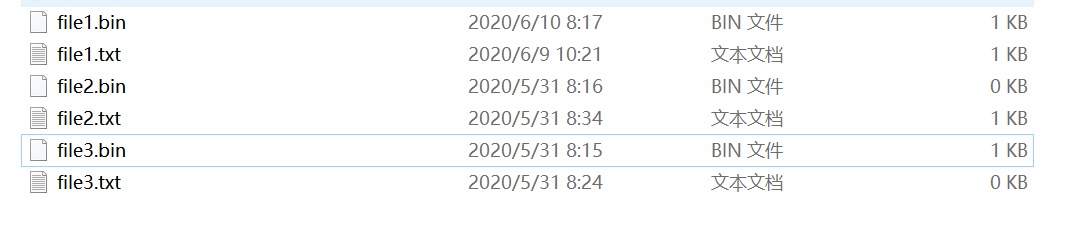
size:文件大小(以B为单位)

nextFile:指向后继文件的指针

content:文件存储内容（二进制文件，后缀为.dat）

1. 尝试创建非二进制的.txt .doc等文本文件，但是效果不好：export import函数指令实现了对文本文件的操作，但由于系统不同，虚拟系统能力不够，对于文件的创建无法识别是.txt .doc .xls .cpp等具体类型，暂时只可以对文本内容进行导入导出操作，无法实现统一的.docx .xls .txt类型文件直接从系统导入本地磁盘驱动；

本地磁盘G:\\OS\_CD文件夹与虚拟文件系统交互状态截图：



3.没有合理地使用系统调用函数，原因：对于系统调用函数不清楚。尝试使用SetCurrentDirectory()改变盘符，但是系统报错，调试未果；

**优点：**

**综述：**

**仿照学习windows的链接文件结构设计此虚拟文件系统，提高磁盘的空间利用率，不存在碎片问题：**

1. **有利于文件的插入和删除**
2. **有利于文件动态扩充**

1.Convert()函数的编写：

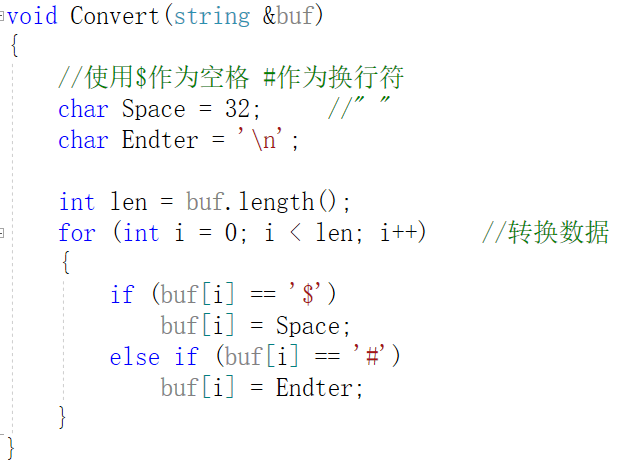
由于string类型定义的content，每一次写入文件，一旦遇到空格或者换行符，就会自动结束输入，默认丢弃后续的输入内容，作为下一次命令的输入，这样的操作不利于文件的输入和读写；

例如：

输入：1010101 1010100

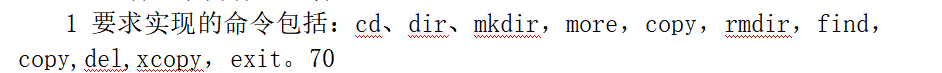
输出：1010101

所以这里自定义一种输入方法Convert()函数：以$代表空格符，以#代表换行符：



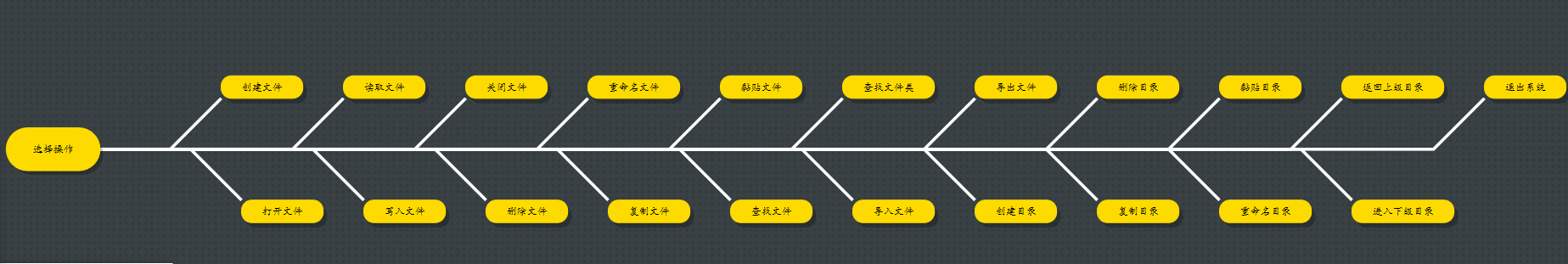
2.仿照实现的命令更加多元化和多样化，自适应性强

实验基本要求：



除了基于对文件的操作，还包括对目录的创建，删除，重命名，复制，黏贴，查询某一后缀文件(dir \*.txt)等等操作

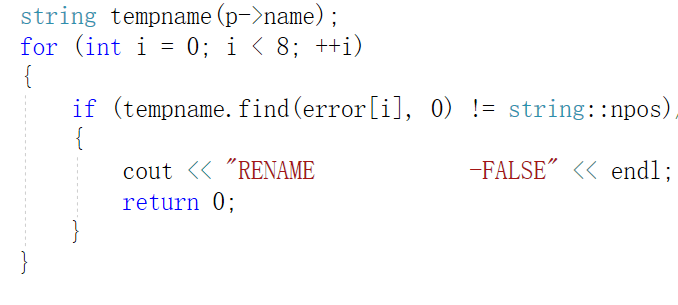
主要有：



3.检测命名的合法性：

不管是文件名，还是目录名，都会出现命名不规范的问题，例如名称中出现”:””\*””\\”等等，故而需要对创建的文件和目录进行命名检测：

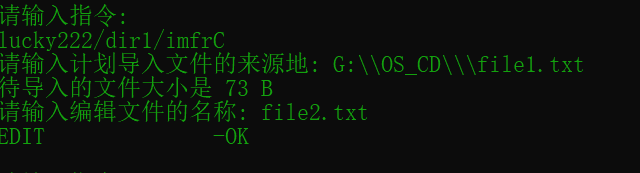
string error[] = { "/","\\", ":","<",">","|","\*","&" }; //命名中的非法字符



4.实现两组export import ：

系统外部（与本地磁盘文件进行交互）：

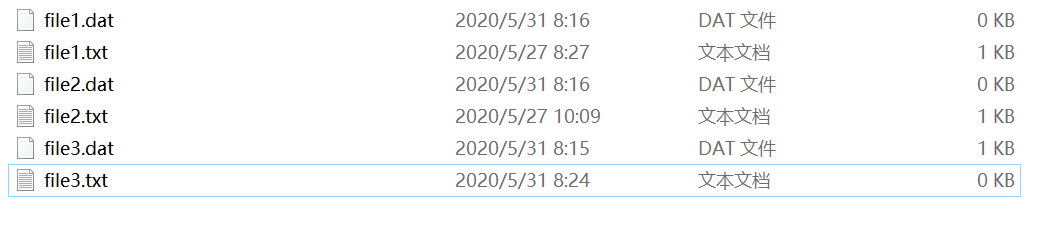
从本地导入文件到虚拟系统：



将虚拟系统文件导出到本地：

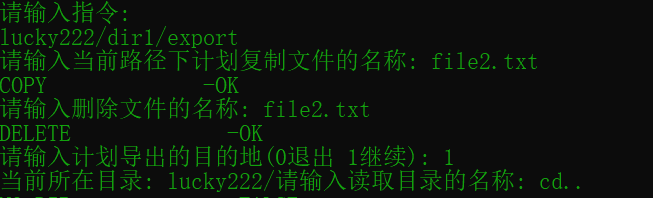


结果截图：

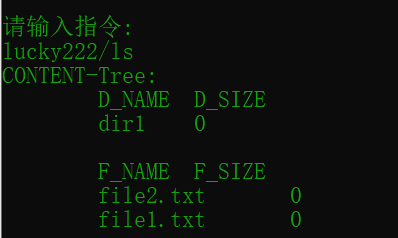


系统内部（虚拟系统内部例如磁盘X与磁盘Y进行交互）：

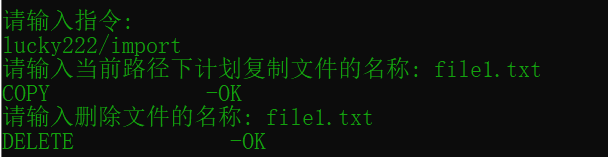
文件系统内部互相导出：



导出结果截图显示：

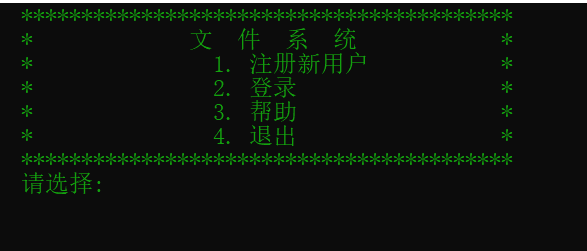


文件系统内部导入：

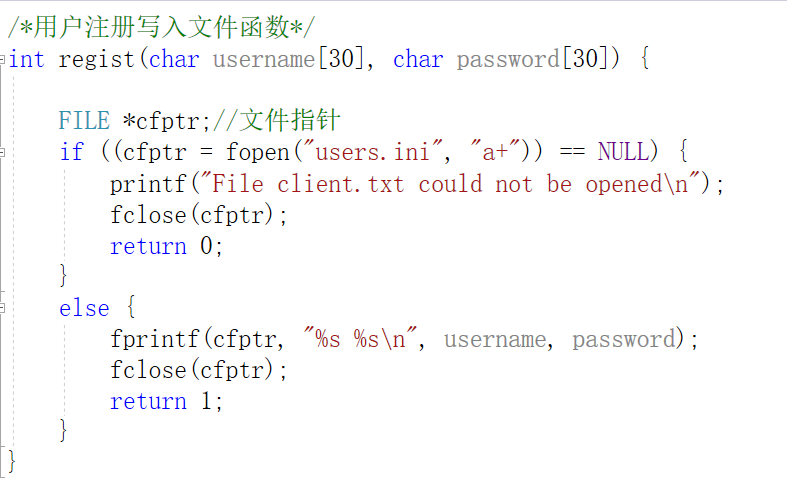


5.用户注册和用户登录：

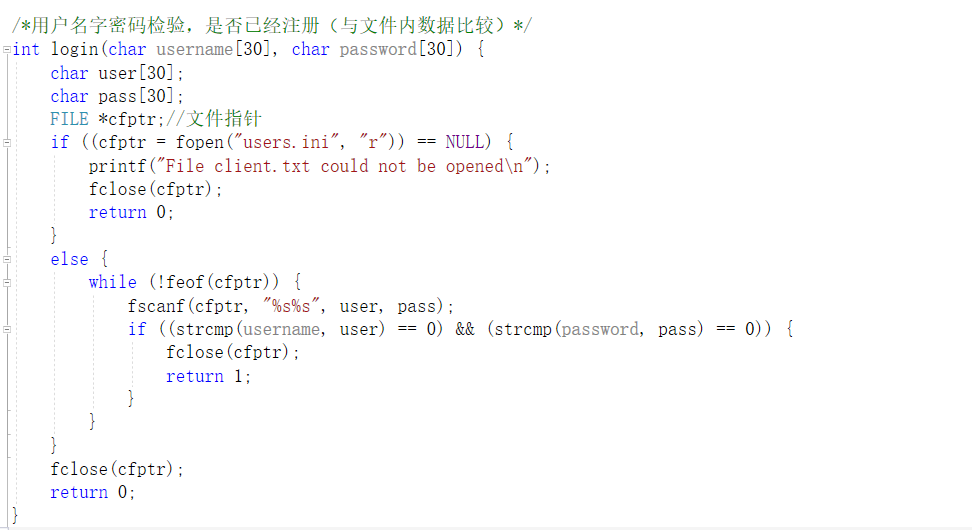
允许一个新用户使用文件系统，首先是注册账户，并将其相关信息保存在user.ini配置文件中，在下一次登录是扫描此文件，对用户名和账户密码进行意义检测匹配：



注册函数设计：

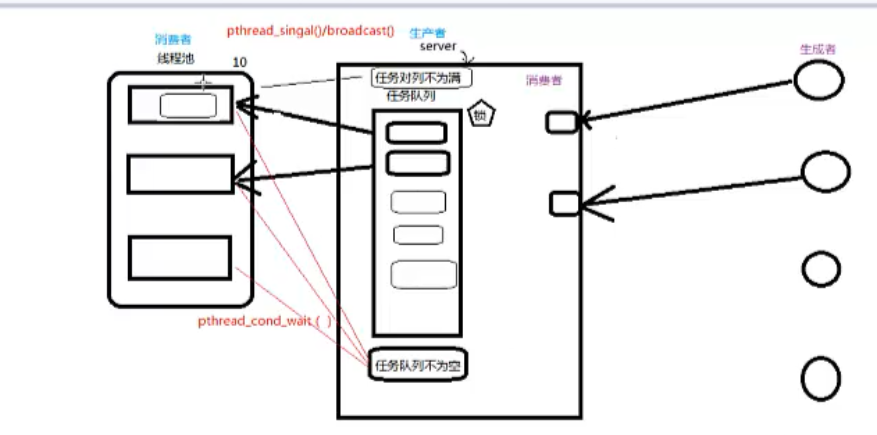


登录验证函数设计：

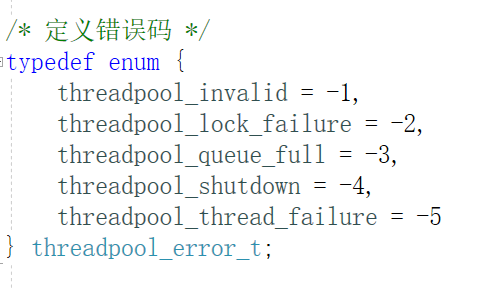


6.线程池：

1. 循环算法维护线程池，相当于生产者消费者模型中的苹果框，实现自我管理，不依靠外界另外添加条件变量
2. 多线程设计思想仿照服务器-客户端C/S模型:

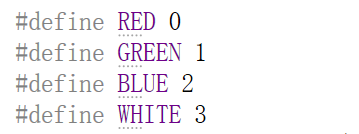


1. 线程池设计有特定的错误编码，一旦初五，返回对应出错类型，便于查找分析出错原因，有利于快速解决问题：

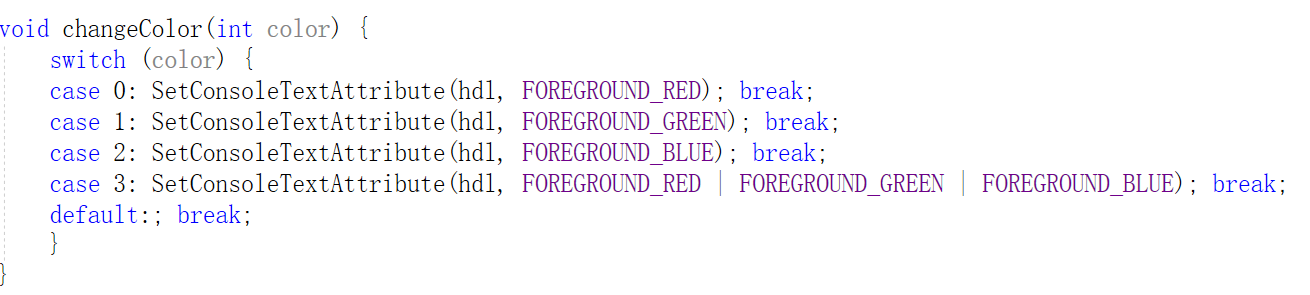


1. 界面的颜色调换操作：

实现宏定义：



函数设计：



实现结果：

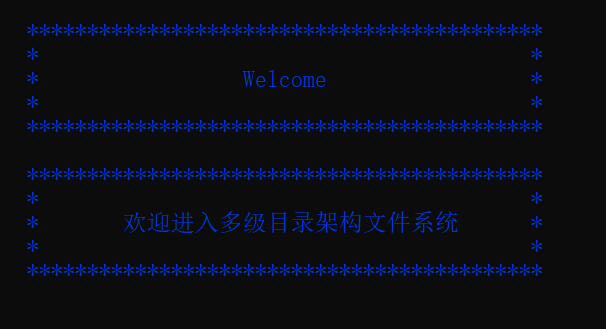
绿色：



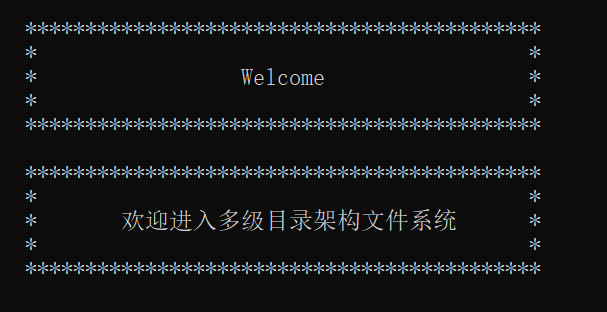
红色：



蓝色：



白色：



**调试笔记记录：**

1.如何解决LINK2005 LINK1169报错(知其所以然)

从 C/C++ 源程序到可执行文件要经历两个阶段 :

(1) 编译器将源文件编译成汇编代码，然后由汇编器 (assembler) 翻译成机器指令 ( 再加上其它相关信息 ) 后输出到一个个目标文件 (object file, VC 的编译器编译出的目标文件默认的后缀名是 .obj) 中；

(2) 链接器 (linker) 将一个个的目标文件 ( 或许还会有若干程序库 ) 链接在一起生成一个完整的可执行文件。

具体查看文档链接：

<https://blog.csdn.net/liufyhui/article/details/43567245>

2.阅读DOS系统感悟：

1. 系统内部没有发现具体的fs文件夹；
2. 部分汇编程序无法读明白；
3. 有待进一步研读，自己设计的太低端，可以读一读“简单系统”的源码，领悟设计思路等等；

3.无法检测到#include<pthread.h>如何解决：

1. 下载pthreads-w32-2-9-1-release.zip
2. 配置环境

在pthread.h的35行的位置添加下面一条代码：

**#define HAVE\_STRUCT\_TINMESPEC**

1. **配置文件：**

　　dll——动态链接库

　　include——头文件

　　lib——静态链接库

具体参见文档链接：

<https://www.cnblogs.com/miaoqianling/p/10544401.html>