



北京林业大学

操作系统实习报告

学 院	信息学院
专业名称	计算机科学与技术创新实验班
班 级	计创 18
学 号	181002222
姓 名	连月菡
辅导老师	李巨虎

2020 年 06 月 06 日

北 京 林 业 大 学

2019 学年—2020 学年第 2 学期计算机算法设计与实践实验报告书

专业：计算机科学与技术(创新实验班) 班级：计创 18

姓 名：连月菡 学 号：181002222

实验地点：家 任课教师：李巨虎

实验题目：文件系统模拟

实验环境：Visual Studio 2019 Community

一. 文件系统格式

逻辑结构:定长记录文件。文件中存储的是一个固定长度的序列，每条记录 中包含了一个 FCB（即下文的 struct directory）。

物理结构:串联文件。文件信息存放在不连续的物理空间中，使用文件分配表 FAT 记录存放顺序，每个物理块中的内容存放下一个需要访问的物理块的块号。

存取方式:顺序存储。

磁盘管理:位示图 MAP。当该块被分配时置 1，否则置 0。

模拟系统盘二进制存储形式——dat files(.dat)

二. 命令功能

1. cd [PATH/DIRECTORY]

通过 directory 进入当前目录底下的子目录

通过 path 进入绝对路径下的目录

如：可以通过 cd ../.. 返回两层父目录

2. Dir

显示当前目录底下的子目录名称、创建时间、<DIR>，子文件名称、大小、创建时间

3. help

显示所有可以调用的命令功能

4. time

显示当前时间

5. ver

显示当前版本（用的是 Windows 10 的版本号）

6. md [DIRECTORY]

在当前目录底下新建一个目录名为 DIRECTORY 的目录

7. mkfile [FILENAME]

在当前目录底下新建一个文件名为 FILENAME 的文件

然后输入占用块数，可以实现非连续块存储同一文件

最后输入文件内容，目前不能实现回车的放入

8. more [FILENAME/DIRECTORY]

显示文件和目录创建时间

如果是文件，就显示文件名以及文件内部内容

如果是目录，就显示目录名以及目录包含的目录项个数

9. rd [DIRECTORY]

删除当前目录底下目录名为 DIRECTORY 的目录（当目录中有子目录项时，也

会循环删除子目录项），并且回收所占用空间（重置 map）。

10. del [FILENAME]

删除当前目录下名为 FILENAME 的文件并回收占用空间（重置 map）

11. rename [FILENAMEA][FILENAMEB]

将当前目录底下名为 FILENAMEA 的文件更名为 FILENAMEB

12. copy [FILENAME][PATHDIRECTORY]

将当前目录（相对路径）或任意路径（绝对路径）下的文件 FILENAME 复制到 PATHDIRECTORY 目录底下

13. move [FILENAME][PATH]

将当前目录（相对路径）或任意路径（绝对路径）下的文件 FILENAME 移动到 PATHDIRECTORY 目录底下（原文件删除）

14. xcopy [DIRECTORY][PATH]

将当前目录下的子目录项（相对路径）或任意目录（绝对路径）下的文件夹
DIRECTORY 复制到 PATH 底下（原文件夹保留，且原文件夹中包含的目录
项也会
一并复制到目标路径下）

15. import [FILENAME][PATH]

将系统盘中的指定文件复制到虚拟盘的当前目录（相对路径）或任意目录（绝对路径）下，并可以同时复制文件存储内容，并在虚拟盘中通过 more 进行输出

展示

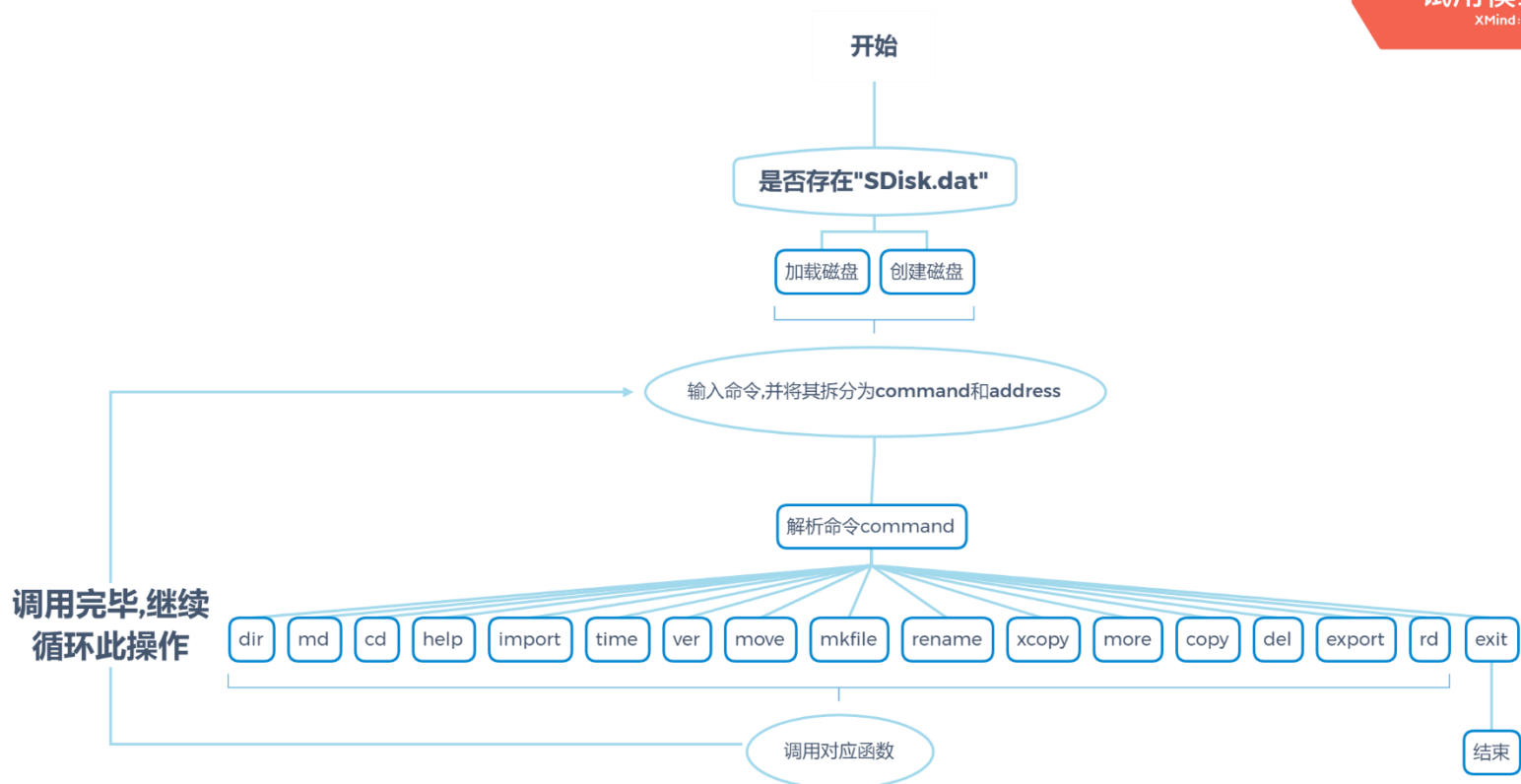
16. export [FILENAME][PATH]

将虚拟盘中指定文件（相对/绝对路径）复制到系统盘中，并可在系统盘中
查看文件内容

17. **多线程设计思路：**用一个线程来和用户进行交互， 接受请求并将请求转换为对应的消息，通知后台维护虚拟磁盘驱动器的线程，程序可同时运行多次，

每个程序都可接受用户的请求，但后台只有一个线程在维护虚拟磁盘驱动器上的内容。

三． 模拟文件系统整体流程概述



四. 有关数据结构及函数流程图说明

1. 磁盘信息结构体

```

1 struct diskinfo    //磁盘信息数据结构
2 {
3     unsigned long size;        //磁盘大小, 单位字节
4     unsigned long blocksize;   //磁盘的盘块大小, 单位字节
5     unsigned long blocknum;    //磁盘的盘块数
6     unsigned long clustersize; //一簇的大小, 单位盘块
7     unsigned long usedspace;   //已用空间, 单位字节
8     unsigned long remainspace; //剩余空间, 单位字节
9     unsigned long usedblock;   //已用盘块
10    unsigned long remainblock;  //剩余盘块
11 };
    
```

2. 目录项结构体

```
1 struct directory    //目录项, 即FCB
2 {
3     string name;      //文件前缀或目录名
4     string type;      //扩展名
5     bool isdir;       //目录标示符
6     time_t buildtime; //创建时间
7     time_t accesstime; //访问时间
8     time_t modtime;   //修改时间
9     unsigned short int beginblock; //文件起始盘块或目录盘块
10    long filesize;    //文件大小或目录所含目录项数目
11    string inside;    //文件内部存储内容
12 };
```

3. 全局变量

```
全局变量

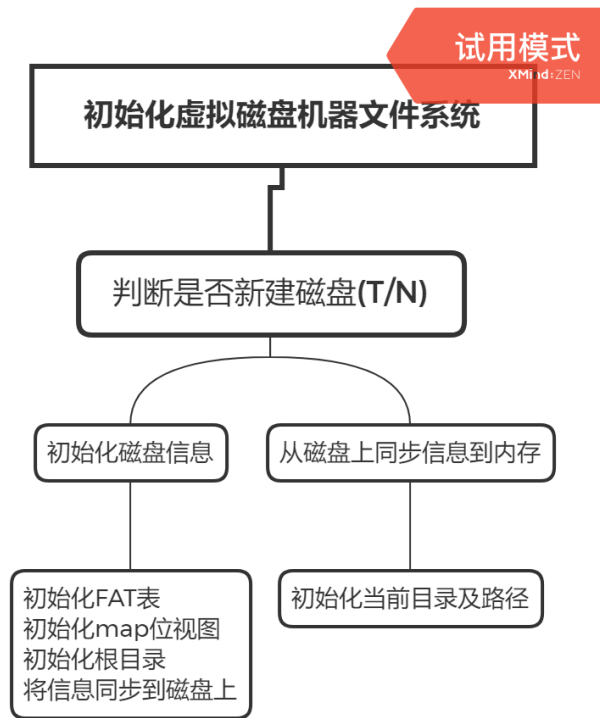
1 static string globalpath;    //当前路径
2 static string diskpath = "test_import.dat"; //虚拟磁盘路径
3 static fstream disk;        //虚拟磁盘
4 static diskinfo dinfo;      //磁盘信息
5 static unsigned short int fat[DATASIZE];    //FAT表1
6 static unsigned short int map[DATASIZE / 16]; //位示图
7 static int curblock;        //当前目录盘块 (相对位置)
8 static directory curdir[7], temp[7]; //当前目录列表和临时目录列表
9 static char right[] = ".", parent[] = "..";
```

4. 命令结构体

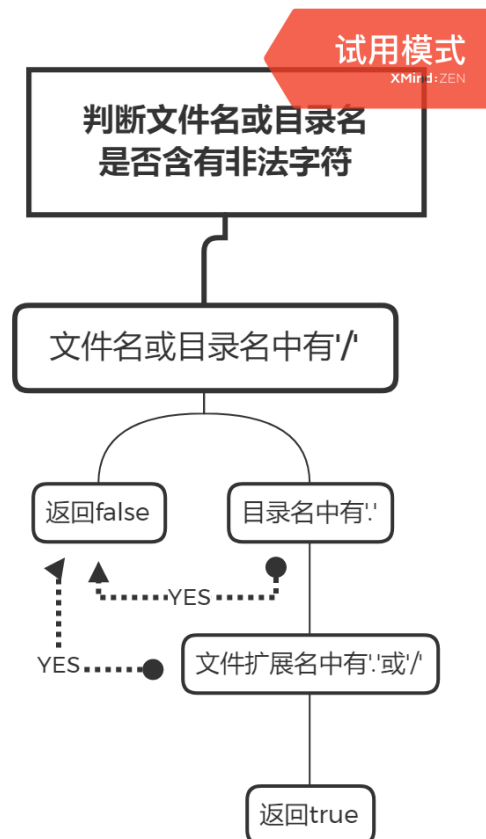
```
命令结构体

1 struct CMD
2 {
3     string cmd;
4     string command; //命令
5     string address; //地址 (如果有两个地址也会同时存在这个string内)
6 }c; //命令结构体, 包含命令和地址, 为全局变量
```

5.初始化虚拟磁盘文件系统



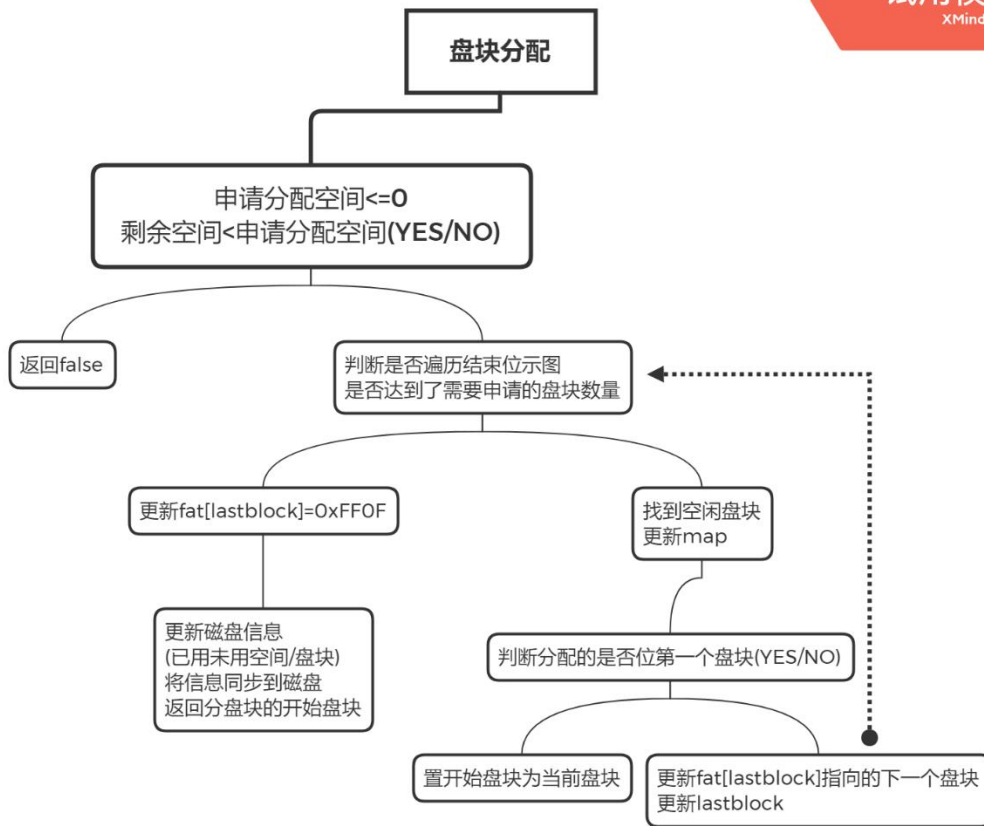
6.判断文件名或目录名是否含有非法字符



7. 盘块分配

试用模式

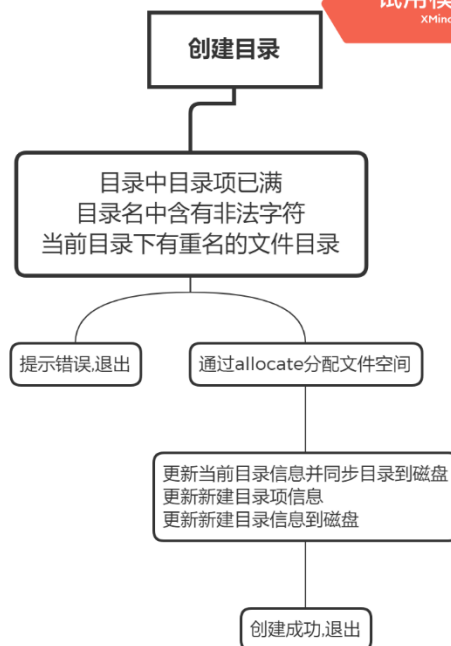
XMind:ZEN



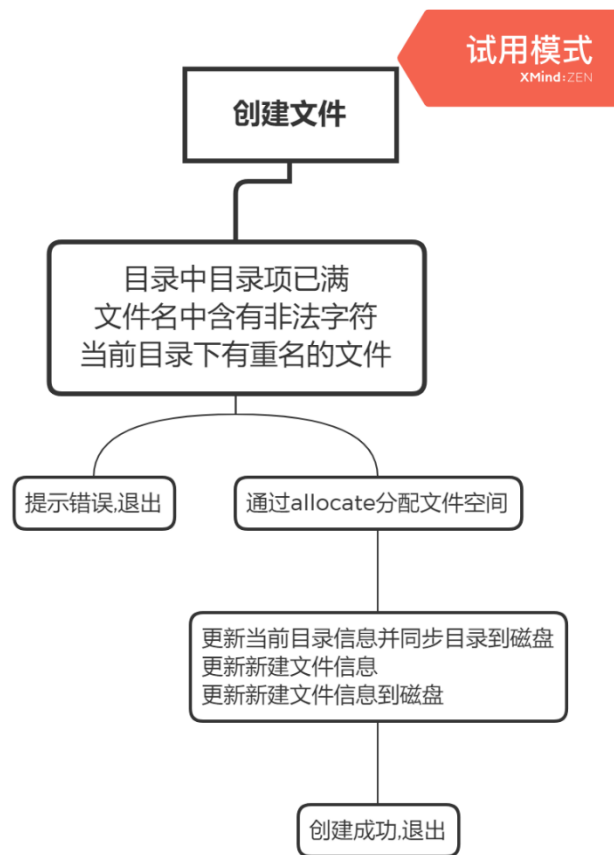
8. 创建目录

试用模式

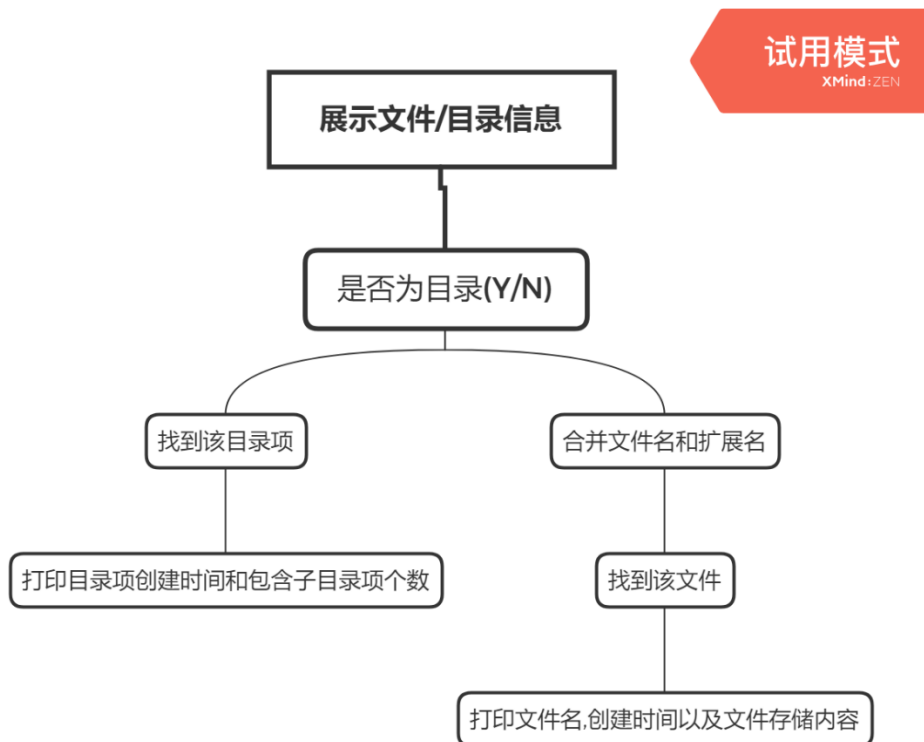
XMind:ZEN



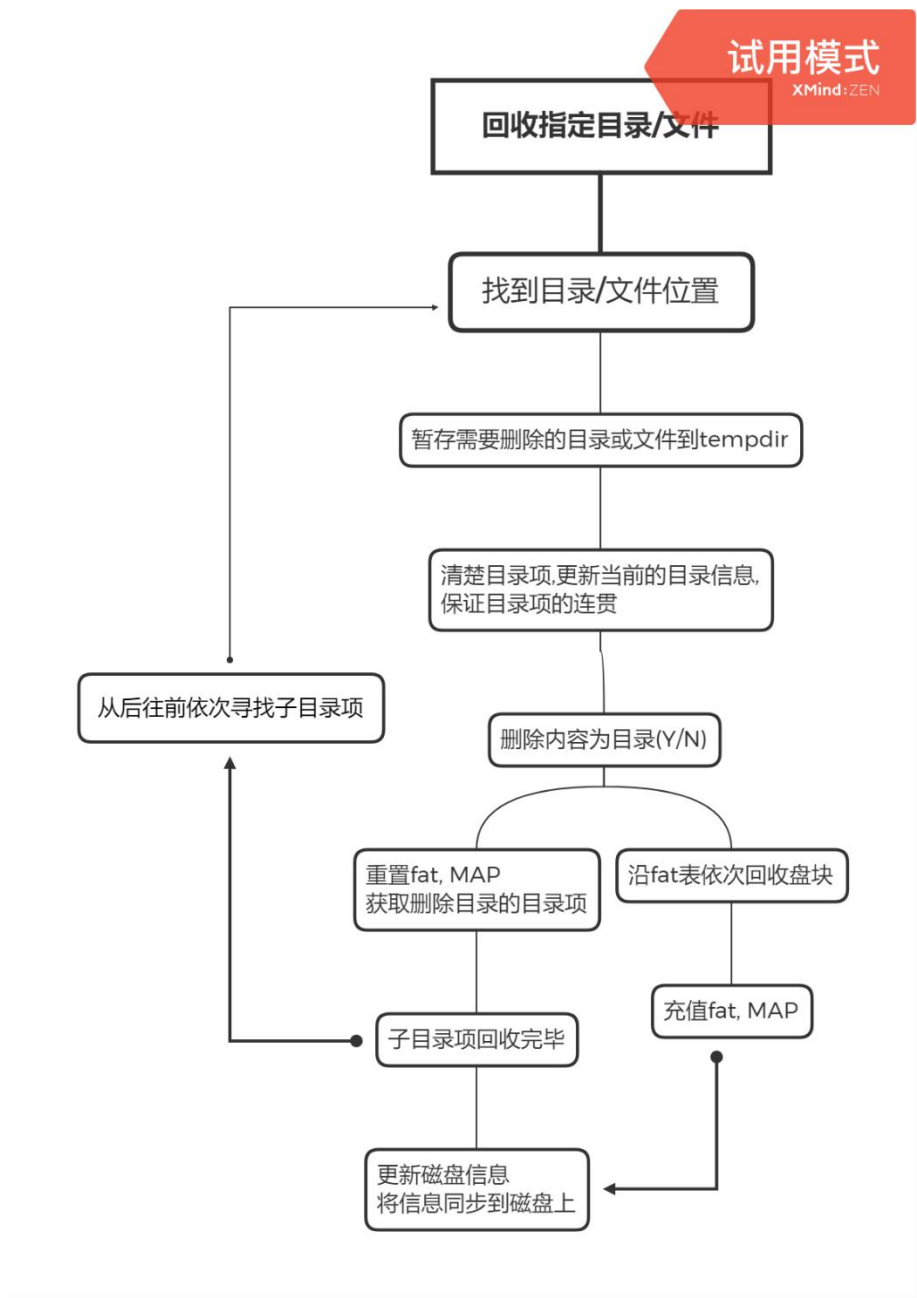
9.创建文件



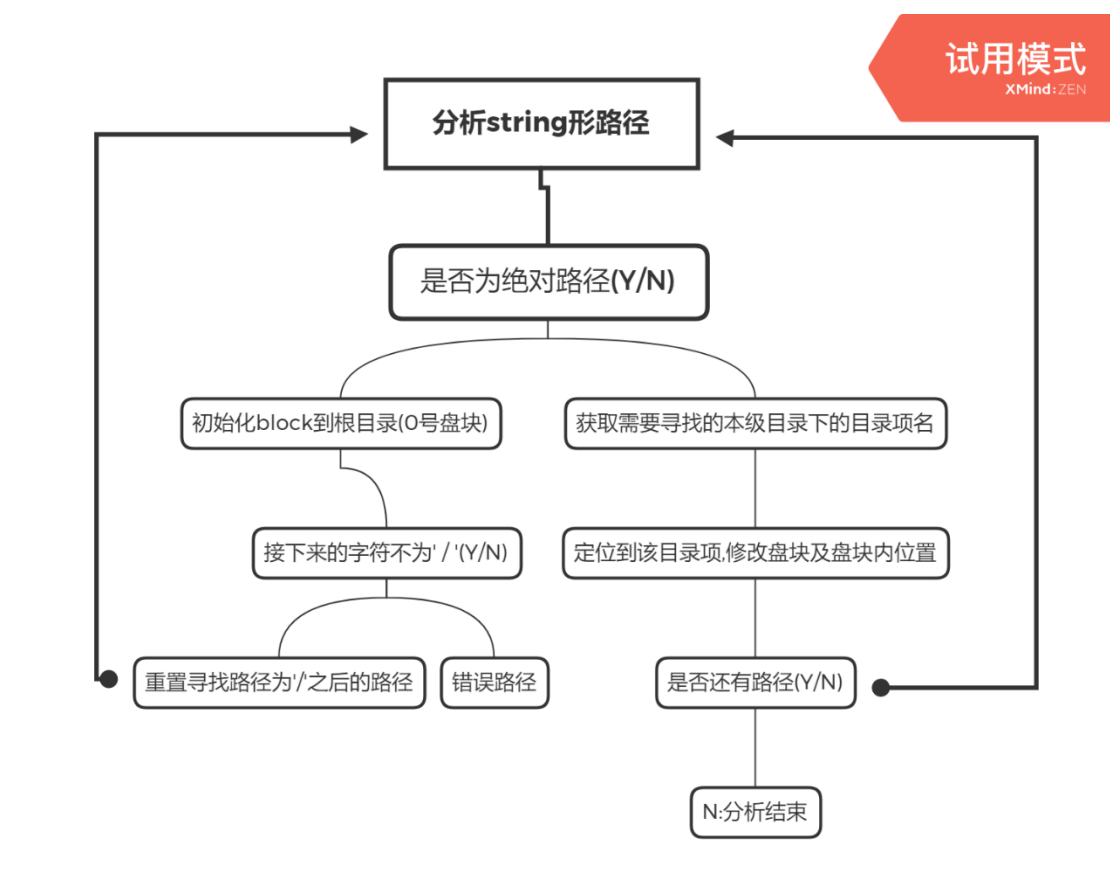
10.目录信息



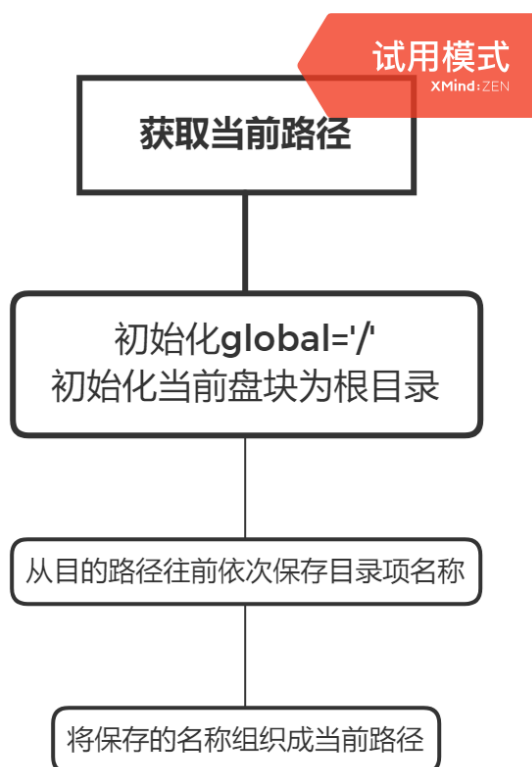
11.回收指定目录/文件



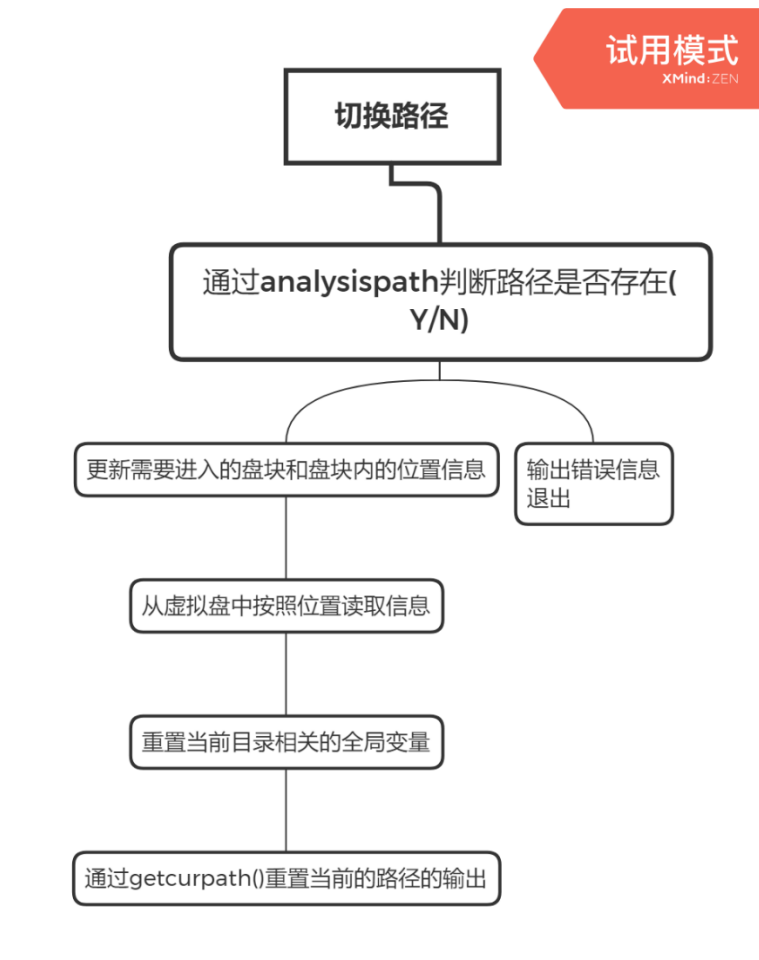
12.分析 string 型路径



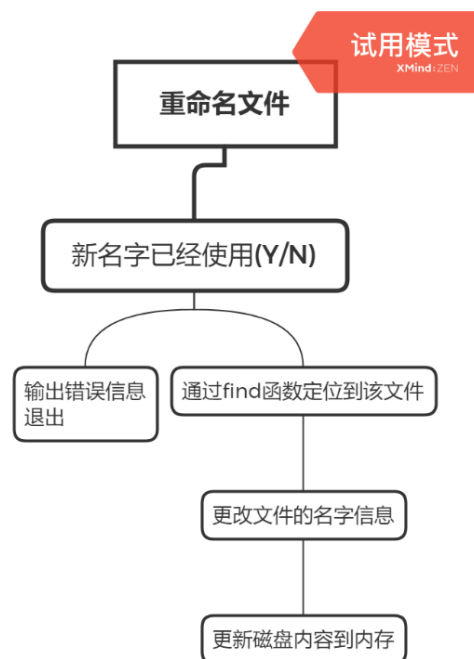
13. 获取当前路径



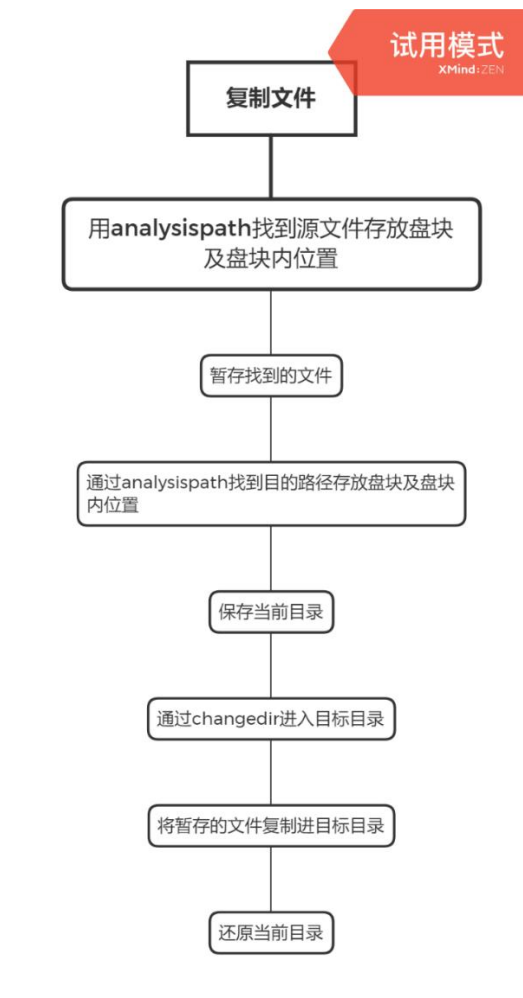
14. 切换路径



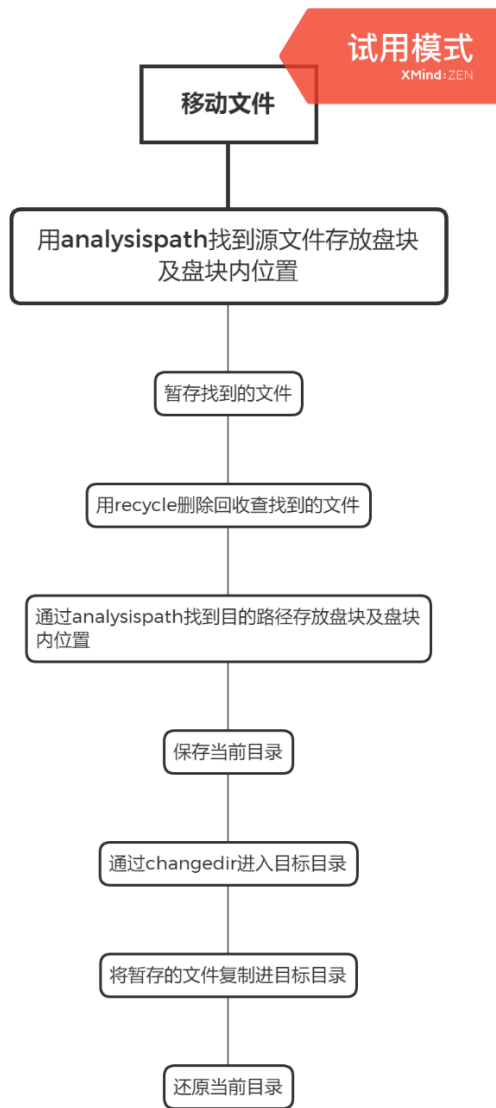
15. 重命名文件



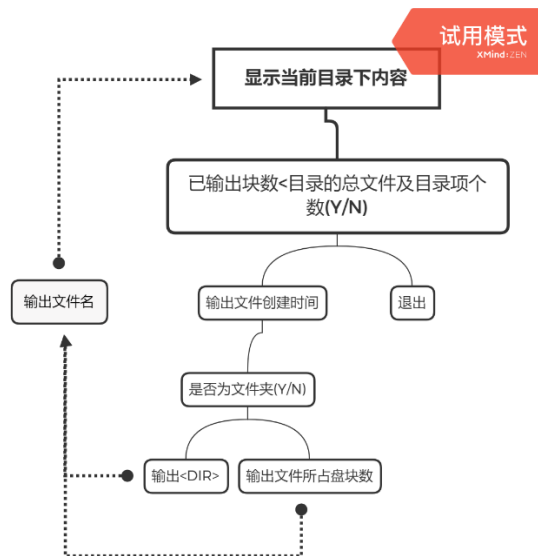
16.复制文件



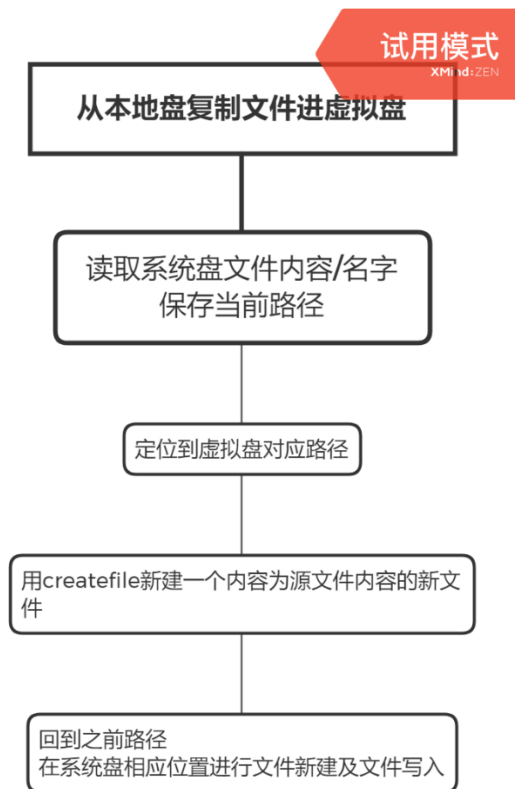
17.移动文件



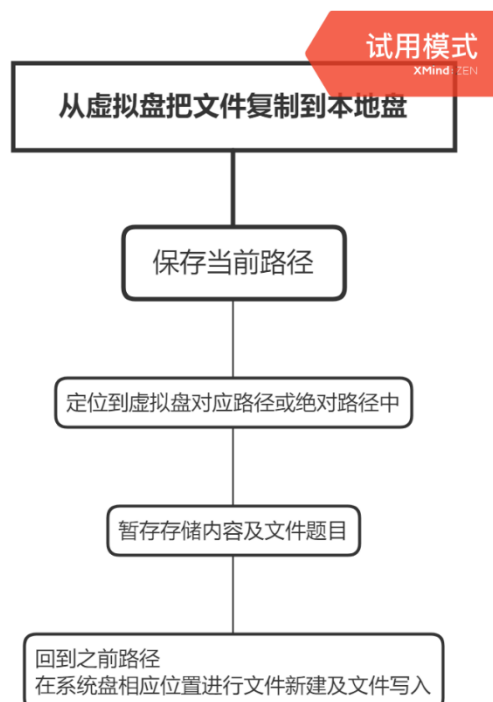
18.显示当前目录下的内容



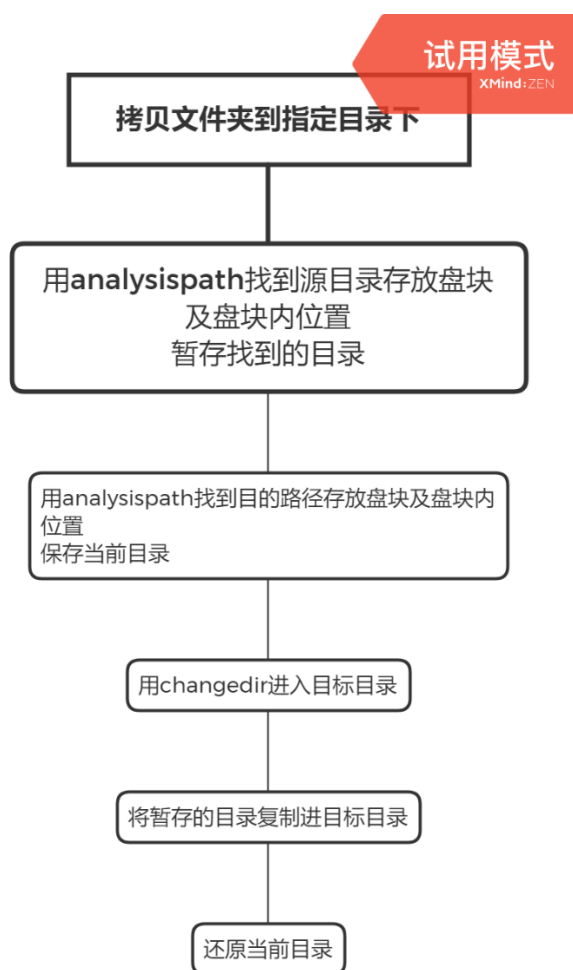
19.从本地盘复制文件进虚拟盘



20. 从虚拟盘复制文件进本地盘



21. 拷贝文件夹到指定目录下



五. 主要函数文字说明

1. Void init_diskinfo()

初始化磁盘信息

2. Void init_fat()

Fat 表初始化，第一个盘块初始化为已用 `fat[0] = 0xFFOF`，其余未用盘块置 0

3. Void init_map()

map 位示图初始化，第一个盘块置 1，其余盘块置 0

4. Bool init_disk()

建立虚拟磁盘的文件流，如果没有磁盘则新建，有磁盘则可以选择载入或格式化磁盘，载入磁盘返回 `true`，新建或格式化磁盘返回 `false`

5. Void init_dirtime(directory *dir)

初始化目录项的时间信息，如新建时间等

6. Void init_dir(directory *dir)

初始化新建目录的共同目录项字段（如 type, name 等）

7. Void init_root()

初始化根目录并将其同步到磁盘上

8. Void disktoemem()

同步磁盘内容到内存中，包括磁盘信息、fat 表、位示图和当前目录

9. Void memtodisk()

同步内存中的磁盘信息、fat 表、位示图到磁盘上

10. Void init_fs()

初始化 Sdisk.dat 及其文件系统

11. Void analysisname(string filename, sting &name, string &type)

将文件名分解成文件名和扩展名

12. Bool isName(bool isdir, string filename)

判断文件名或目录名中是否有非法字符 “/”“.”等

13. Unsigned short int allocate(unsigned int size)

为文件或目录分配 size 个盘块，修改 map 位示图、FAT 表相关信息，返回分配的最开始的盘块号

14. Int find(string name, int block)

根据文件名在 block 所指盘块下找到文件或目录项，返回其在该盘块中的位置

15. Void createdir(string name)

在当前目录下新建一个名为 name 的目录

16. Void create_file(string filename, int size, string inside)

在当前目录下新建一个名为 filename、大小为 size 个盘块，内容为 inside 的文件

17. Void show_more(string filename, int block)

在 block 盘块下找到名为 filename 的文件或目录项并显示相关信息（创建时间、文件/目录项名、文件内部信息、目录项含子目录项个数）

18. Void recycle(int dirblock, int pos)

回收 dirblock 盘块下 pos 位置的目录项/文件及其内部所有信息，释放盘块空间

19. Void delfile(int pos)

删除当前目录下 pos 位置的文件并释放空间

20. Void rmdir(int pos)

删除当前目录下 pos 位置的目录项及其内部所有内容并释放空间

21. Bool analysispath(string path, int *block, int *pos)

分析 path，得到路径所指向的盘块和在盘块内的位置，如果 path 存在返回 true，否则返回 false

22. Void getcurpath()

通过当前的 block 和 pos 获取当前路径，重置 curdir 和 globalpath

23. Void rename(string oldname, string newname)

将当前目录下名为 oldname 的文件/目录项重命名为 newname

24. Void changedir(string path)

通过 path 重置当前的 block 和 pos

25. Void copyfile(string src, string den)

将 src 文件复制到 den 目录下

26. Void dir()

显示指定目录下的内容

27. Void help()

显示帮助信息——命令输入方式

28. Void time()

显示当前时间

29. Void ver()

显示当前版本信息

30. Void import()

将系统盘中的文件复制到虚拟盘的指定目录下

31. Void export()

将虚拟盘中任意目录下的文件复制到系统盘中

32. Void xcopy()

将任意目录复制到任意目录下

33. Void exit()

退出系统

六. 模拟磁盘详细说明

1. 二进制磁盘文件中存放的信息：

0	磁盘基本信息(磁盘大小,盘块大小,盘块数,已用/剩余空间/盘块数)
1	FAT 表
64	MAP 位示图
128	数据块

2. 目录项包括：

名字
创建时间
起始盘块
属性（文件 / 目录）
数据内容

3. 每个块包括：7 个目录项（除去当前目录和父目录之外，一个目录下可以保存 5 个子目录或文件）

4. MAP 和 FAT:

MAP 是位示图，由于是 short int，每一个 MAP 有 16 位，所以 MAP 的大小为 DATASIZE / 16

FAT 表，由于一个块对应一个表，所以 FAT 大小为 DATASIZE

5. 对目录的存放与读取

- 每个盘块可以存 7 个目录项
- 全局变量 `curdir[7]` 来存放当前盘块
- 可以通过 $(\text{DATABEG} + \text{curdir}[1].\text{beginblock}) * \text{blocksize}$ 来定位子目录项或当前目录或父目录在磁盘中的存放位置
- 如果需要读取目录项 FCB, 则读取 `sizeof(directory)` · 如果需要读取盘块所有信息, 则读取 `sizeof(curdir)`

6. 对文件的存放与读取

- 文件会有需要存放在多个块中的情况
- 用 FAT 表进行对连续块的读写, FAT 初始化: `FAT[0] = 0xFF0F` (结束位), `FAT[i] = 0 (i ∈ [1, DATASIZE])` · 比如说文件 1 需要 3 个块进行存储, 经过查询 `map` 位示图, 得到空闲块为

1、3、4, 则 `FAT[1] = 3; FAT[3] = 4; FAT[4] = 0xFF0F`

七. 测试说明

1. dir

```
SDisk:/>dir
2020-5-12  14:11      <DIR>      .
2020-5-12  14:11      <DIR>      ..
```

2. md

```
SDisk:/>md os
SDisk:/>dir
2020-5-12  14:11      <DIR>      .
2020-5-12  14:11      <DIR>      ..
2020-5-12  14:14      <DIR>      os
```

3. time

```
SDisk:/>time
current time: 14:15:39
```

4. ver

```
SDisk:/>ver
Microsoft Windows [版本 10.0.17134.81]
```

5. cd

```
SDisk:/>cd os
SDisk:/os/>dir
2020-5-12  14:14      <DIR>      .
2020-5-12  14:11      <DIR>      ..
```

6. mkfile

```
SDisk:/os/>mkfile os.txt
Input size: 1
Input content:osososos
SDisk:/os/>dir
2020-5-12  14:14          <DIR>          .
2020-5-12  14:11          <DIR>          ..
2020-5-12  14:16                  1      os.txt
```

7. more

```
SDisk:/os/>more os.txt
Build time: Fri Jun 12 14:16:57 2020
os.txt
osososos
```

8. rd

```
SDisk:/>rd os
SDisk:/>dir
2020-5-12  14:11          <DIR>          .
2020-5-12  14:11          <DIR>          ..
```

9. rename

```
SDisk:/>dir
2020-5-12  14:22          <DIR>          .
2020-5-12  14:22          <DIR>          ..
2020-5-12  14:22          <DIR>          os
SDisk:/>rename os haha
SDisk:/>dir
2020-5-12  14:22          <DIR>          .
2020-5-12  14:22          <DIR>          ..
2020-5-12  14:22          <DIR>          haha
SDisk:/>_
```

10. copy

```
SDisk:/>md haha
SDisk:/>mkfile 1.txt
Input size: 1
Input content:os
SDisk:/>copy 1.txt haha
SDisk:/>cd haha
SDisk:/haha/>dir
1970-0-1   8:0                  1      .
2020-5-12  15:5          <DIR>          ..
2020-5-12  15:5                  1      1.txt
```

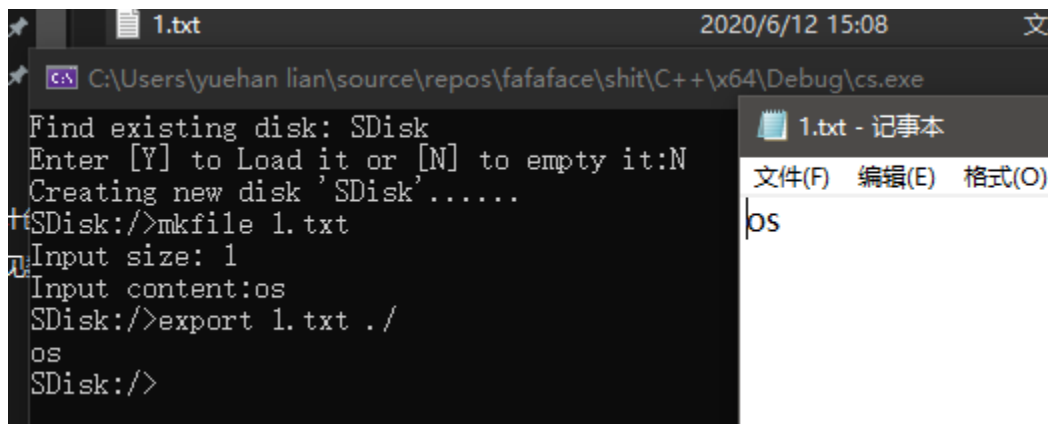
11. import

```

SDisk:/>md os
SDisk:/>import 123.txt os
SDisk:/>cd os
SDisk:/os/>dir
2020-5-12  15:11          <DIR>          .
2020-5-12  15:11          <DIR>          ..
2020-5-12  15:11                  1      123.txt
SDisk:/os/>

```

12.export



The screenshot shows a terminal window with the following commands and output:

```

Find existing disk: SDisk
Enter [Y] to Load it or [N] to empty it:N
Creating new disk 'SDisk'.....
SDisk:/>mkfile 1.txt
Input size: 1
Input content:os
SDisk:/>export 1.txt ./
os
SDisk:/>

```

Overlaid on the terminal is a Notepad window titled "1.txt - 记事本" showing the content "os".

13.xcopy

```

SDisk:/>xcopy c os
SDisk:/>cd os
SDisk:/os/>dir
2020-5-12  15:11          <DIR>          .
2020-5-12  15:11          <DIR>          ..
2020-5-12  15:11                  1      123.txt
2020-5-12  15:12          <DIR>          a
1970-0-1   8:0              0      c.

```

八. 实验分析

Q1: 不清楚数据结构的设计包含内容

A1: 经过搜索及调查, 决定新建一个 `diskinfo` 数据结构来存储磁盘相关信息, 其中包含磁盘大小、盘块数量、空间使用情况等; 再新建一个目 `directory` 结构体来存储该文件夹所包含的信息, 包括文件名、创建时间、在虚拟盘中存储盘块起始地址、文件大小、目录所含目录项等。

Q2: 在目录底下新建子目录, 再使用 `more` 查看'当前目录'时, 可以看到 `filesize` 有更新, 但是如果 `cd` 到父目录再用 `more` 查看'directory'时候, `filesize` 就没有更新。

A2: 发现是因为没有同时更新父目录里的 `FCB`, 所以通过 `temp[1]("../")` 找到父目录, 然后更新父目录 `block[pos]` 里面的 `filesize`, 就可以实现在父目录中查看 `directory`, 得到正确的子目录数。

最后, 对文件系统有了更好的理解, 可以合理和准确的使用 `map` 位示图和 `FAT` 表, 对文件的逻辑存储和物理存储有各方面的了解。在进行回收和复制文件夹

的过程中，更加巩固了对递归函数的书写技能。在测试的过程中，逐渐发现各种可以优化的地方，如可以同时通过绝对 路径和相对路径进入子目录，在超出目录项范围的时候应适时阻止然后输出错误信息，对程序的健壮性有了更深入的想法