文件系统设计文档

**项目设计方案：**

1. 设计一个主函数，用于演示，名为main()，主要工作是，初始化全局变量，调用函数创建和初始化虚拟磁盘，然后展示菜单等待用户输入操作。
2. 设计一个名为startsys()函数，检测是否存在已经创建的虚拟磁盘，如果没有则创建一个，调用函数格式化它，然后设置当前目录为根目录。
3. 设计一个my\_format()函数，格式化我们创建的虚拟磁盘，将第一块作为引导块，然后建两个FAT表，其余设为空闲
4. 然后执行用户的选择操作，如果用户调用“更改当前目录函数”，设计一个my\_cd()函数，则先打开该文件夹的父目录，然后读取目录信息，在查找指定目录是否存在，如果存在则打开，并把当前目录设置为这个目录。
5. 如果用户选择“创建子目录函数”，则设计一个函数my\_mkdir()，先打开当前目录，检查新建的文件夹是否已经存在，若不存在则为它分配空闲文件表项和空余盘块，创建这个目录，然后，初始化它
6. 如果用户选择“删除子目录函数”则设计一个rmdir()函数，打开当前目录，查找是否存在制定的目录，检查目录是否为空和是否被打开，如果符合要求，回收磁盘块，修改FAT，清除该目录的信息。
7. 如果用户选择“显示目录”，则设计my\_ls()函数，然后读取当前目录信息，然后以一定的格式显示出来。
8. 如果用户选择“创建文件”，则设计my\_create()函数,先分配一个空闲打开文件表项，然后打开目标文件的父目录，读出该父目录文件内容到内存，为新文件分配一个盘块和目录项，准备好新文件的FCB的内容，关闭父目录文件
9. 如果用户选择“删除文件”，则设计my\_rm()函数，先打开父目录文件然后读取目录内容，如果文件已经打开，就把它关闭，然后回收磁盘块并修改FAT，并且删除这个目录项。
10. 如果用户选择“打开文件函数”,则设计my\_open()函数，首先读出父目录文件的内容，并且检查要打开的文件是不是存在，给它分配一个空表项，把读写指针置为0，
11. 如果用户选择“关闭文件函数”,则设计my\_close()函数，检测，如果他在内存中被修改，则关闭的时候写回硬盘，回收打开文件表表项
12. 如果用户选择 “写文件”，则设计my\_write()函数，提示并等待用户输入写方式，然后提示用户输入内容，然后调用do\_write（）将键入的内容写入到文件中
13. 如果用户选择 “读文件“，则设计my\_read()函数，这个比较方便，主要就是调用do\_read()来读取文件。
14. 如果用户选择“退出文件系统”，则设计my\_exitsys()函数，首先打开磁盘上的myfsys文件，再将虚拟磁盘空间中的所有内容保存到磁盘上的myfsys文件中，关闭myfsys文件，最后撤销用户打开文件表，释放其内存空间和释放虚拟磁盘空间。

**任务分配：**

1. my\_format：对文件存储器进行格式化，即按照文件系统的结构对虚拟磁盘空间进行布局，并在其上创建根目录以及用于管理文件存储空间等的数据结构。
2. my\_mkdir：用于创建子目录。
3. my\_rmdir：用于删除子目录。
4. my\_ls：用于显示目录中的内容。
5. my\_cd：用于更改当前目录。
6. my\_create：用于创建文件。
7. my\_open：用于打开文件。
8. my\_close：用于关闭文件。
9. my\_write：用于写文件。
10. my\_read：用于读文件。
11. my\_rm：用于删除文件。
12. my\_exitsys：用于退出文件系统。

汪冠鸿13055519: 1，2，3  
林剑峰13055619: 5，7，8  
纪文韬 13055617: 4，6，11  
王立敏 13055632: 9，10，12

**王立敏**

**int my\_write(int fd)**

**功能：**将用户通过键盘输入的内容写到fd所指定的文件中。

**输入：**fd： open()函数的返回值，文件的描述符；

**输出：**实际写入的字节数。

**功能描述：**

1. 检查fd的有效性

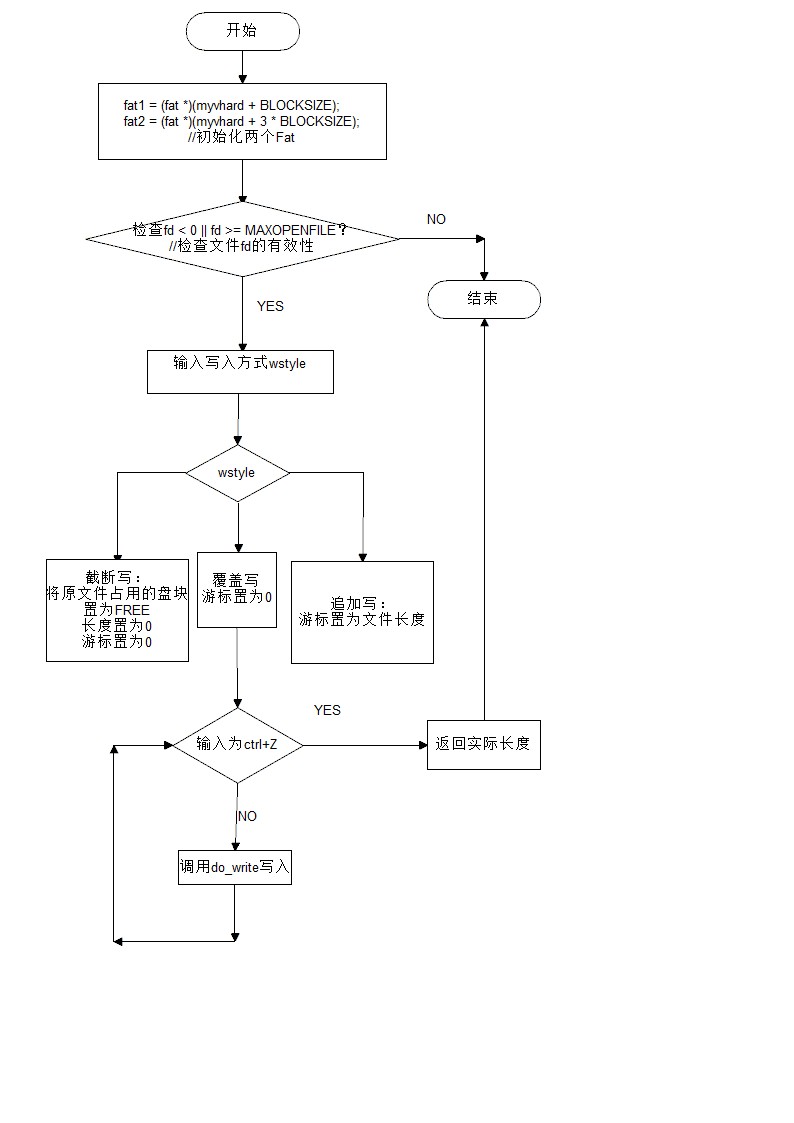
2. 提示并等待用户输入写方式

3. 提示用户输入内容

4. 调用do\_write（）将键入的内容写入到文件中

5. 返回写入的字节数

**流程图：**

****

**int do\_write(int fd，char \*text，int len，char wstyle)**

**功能：**将指定的内容写到相应的文件中去

**输**入：

fd： open()函数的返回值，文件的描述符；

text：指向要写入的内容的指针；

len：本次要求写入字节数

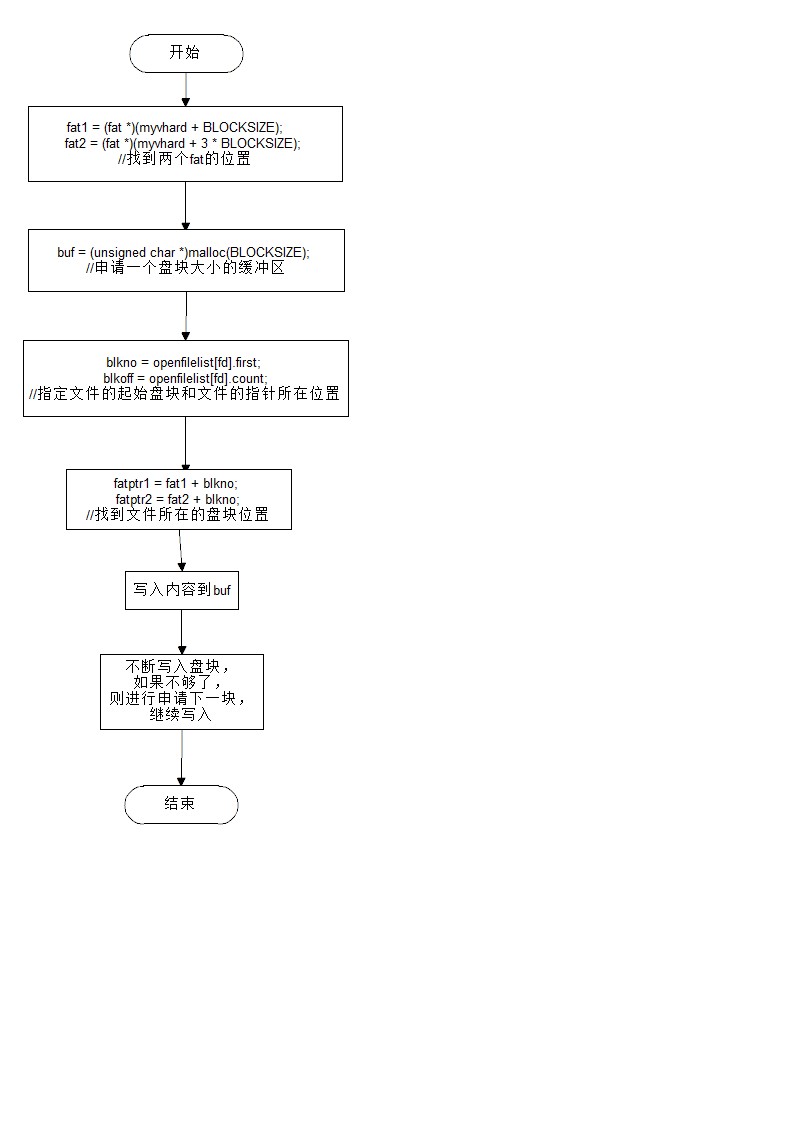
wstyle：写方式

**输出：**实际写入的字节数。

**功能描述：**

1. 申请缓冲区buf
2. 将读写指针转化为逻辑块块号和块内偏移off
3. 将逻辑块块号转换成对应的磁盘块块号blkno
4. 将内容暂存到缓冲区buff
5. 将buff中的内容转移到盘块中

**流程图：**

****

**int myread (int fd, int len)**

**功能：**读出指定文件中从读写指针开始的长度为len的内容到用户空间中**。**

**输入：**

fd： open()函数的返回值，文件的描述符；

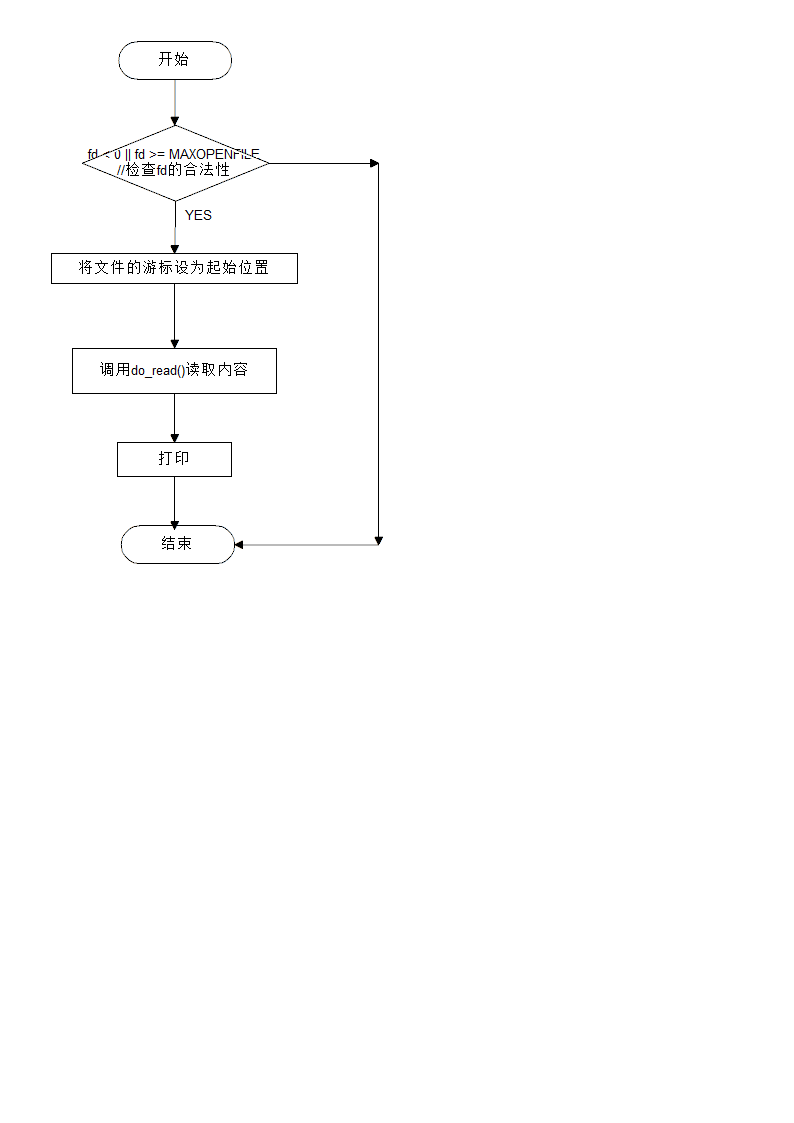
len: 要从文件中读出的字节数。

**输出：**实际读出的字节数。

**功能描述：**

1. 检查fd的有效性
2. 调用do\_read()将指定文件中的len字节内容

**流程图**



**int do\_read (int fd, int len,char \*text)**

**功能：**被my\_read()调用，读出指定文件中从读写指针开始的长度为len的内容到用户空间的text中。

**输入：**

fd： open()函数的返回值，文件的描述符；

len: 要求从文件中读出的字节数。

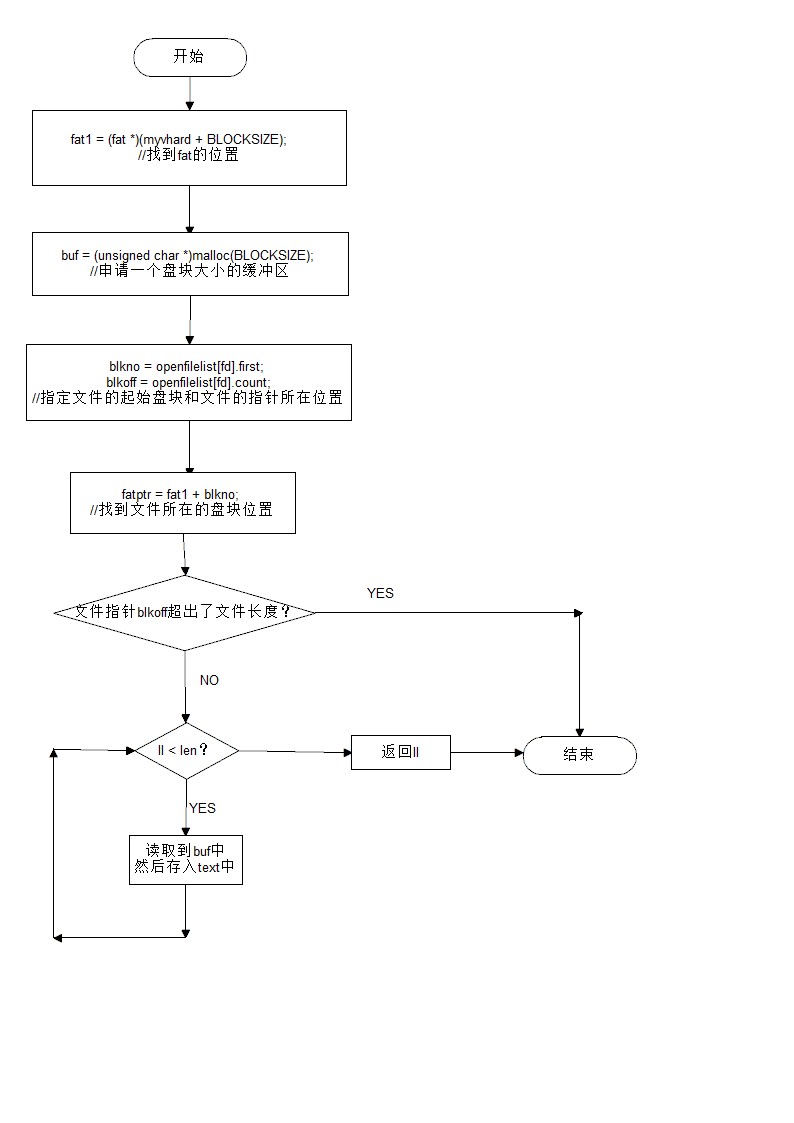
text：指向存放读出数据的用户区地址

**输出：**实际读出的字节数。

**功能描述：**

1. 申请缓冲区buf
2. 将读写指针转化为逻辑块块号及块内偏移量off
3. 查找FAT表，找到该逻辑块所在的磁盘块块号
4. 不断把内容读入带buf中，然后转入到text中，直到读取完毕
5. 释放buf

**流程图：**

****

**void my\_exitsys()**

**功能：**退出文件系统。

**输入：**无

**输出：**无。

**功能描述：**

1打开磁盘上的myfsys文件；

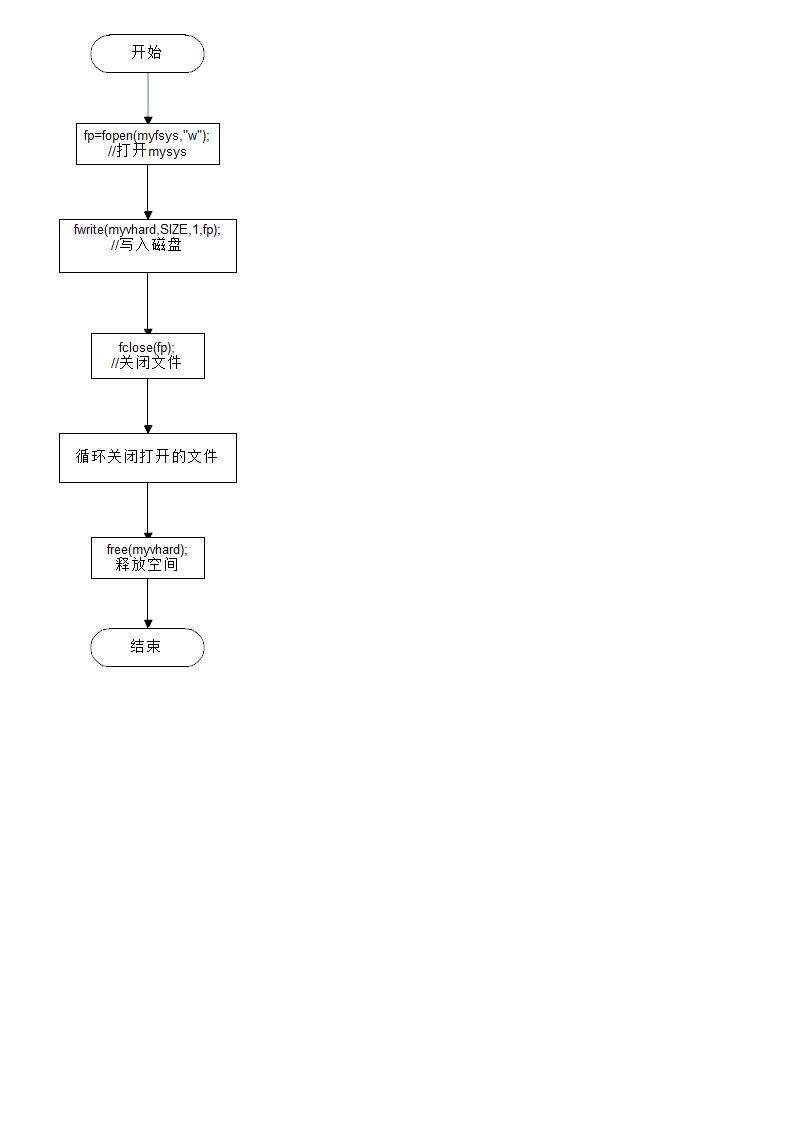
2将虚拟磁盘空间中的所有内容保存到磁盘上的myfsys文件中；

3关闭myfsys文件；

4撤销用户打开文件表，释放其内存空间

5 释放虚拟磁盘空间。

**流程图：**

****

**汪冠鸿**

**1.磁盘格式化函数 my\_format()**

(1）对应命令：my\_format

(2）命令调用格式：my\_format

(3）函数设计格式：void my\_format()

(4）功能：对虚拟磁盘进行格式化，布局虚拟磁盘，建立根目录文件（或根目录区）。

(5）输入：无

(6）输出：无。

(7）函数需完成的工作：

① 将虚拟磁盘第一个块作为引导块，开始的8个字节是文件系统的魔数，记为“10101010”；在之后写入文件系统的描述信息，如FAT表大小及位置、根目录大小及位置、盘块大小、盘块数量、数据区开始位置等信息；

② 在引导块后建立两张完全一样的FAT表，用于记录文件所占据的磁盘块及管理虚拟磁盘块的分配，每个FAT占据两个磁盘块；对于每个FAT中，前面5个块设置为已分配，后面995个块设置为空闲；

③ 在第二张FAT后创建根目录文件root，将数据区的第1块（即虚拟磁盘的第6块）分配给根目录文件，在该磁盘上创建两个特殊的目录项：“.”和“..”，其内容除了文件名不同之外，其他字段完全相同。

程序流程图：



**2.创建子目录函数 my\_mkdir()**

(1）对应命令：my\_mkdir

(2）命令调用格式：my\_ mkdir dirname

(3)函数设计格式：void my\_mkdir(char \*dirname)

(4)功能：在当前目录下创建名为dirname的子目录。

(5)输入：

dirname：新建目录的目录名。

(6）输出：无。

(7）函数需完成的工作：

① 调用do\_read()读入当前目录文件内容到内存，检查当前目录下新建目录文件是否重名，若重名则返回，并显示错误信息；

② 为新建子目录文件分配一个空闲打开文件表项，如果没有空闲表项则返回-1，并显示错误信息；

③ 检查FAT是否有空闲的盘块，如有则为新建目录文件分配一个盘块，否则释放①中分配的打开文件表项，返回，并显示错误信息；

④ 在当前目录中为新建目录文件寻找一个空闲的目录项或为其追加一个新的目录项;需修改当前目录文件的长度信息，并将当前目录文件的用户打开文件表项中的fcbstate置为1；

⑤ 准备好新建目录文件的FCB的内容，文件的属性为目录文件，以覆盖写方式调用do\_write()将其填写到对应的空目录项中；

⑥ 在新建目录文件所分配到的磁盘块中建立两个特殊的目录项“.”和“..”目录项，方法是：首先在用户空间中准备好内容，然后以截断写或者覆盖写方式调用do\_write()将其写到③中分配到的磁盘块中；

⑦ 返回。

程序流程图：



3. 删除子目录函数 rmdir()

(1）对应命令：my\_ rmdir

(2）命令调用格式：my\_ rmdir dirname

(3）函数设计格式：void my\_rmdir(char \*dirname)

(4）功能：在当前目录下删除名为dirname的子目录。

(5）输入：

dirname：欲删除目录的目录名。

(6)输出：无。

(7）函数需完成的工作：

① 调用do\_read()读入当前目录文件内容到内存，检查当前目录下欲删除目录文件是否存在，若不存在则返回，并显示错误信息；

② 检查欲删除目录文件是否为空（除了“.”和“..”外没有其他子目录和文件），可根据其目录项中记录的文件长度来判断，若不为空则返回，并显示错误信息；

③ 检查该目录文件是否已经打开，若已打开则调用my\_close()关闭掉；

④ 回收该目录文件所占据的磁盘块，修改FAT；

⑤ 从当前目录文件中清空该目录文件的目录项，且free字段置为0：以覆盖写方式调用do\_write()来实现；

⑥ 修改当前目录文件的用户打开表项中的长度信息，并将表项中的fcbstate置为1；

⑦ 返回。

程序流程图：



**林剑峰**

一.打开文件

函数名：int my\_open(char \*filename)

功能：打开文件。

输入：filename 欲打开文件的文件名 。

输出： 若打开成功，返回打开的文件的id，否则返回-1。

函数需完成的工作：

① 检查该文件是否已经打开，若已打开则返回-1，并显示错误信息；

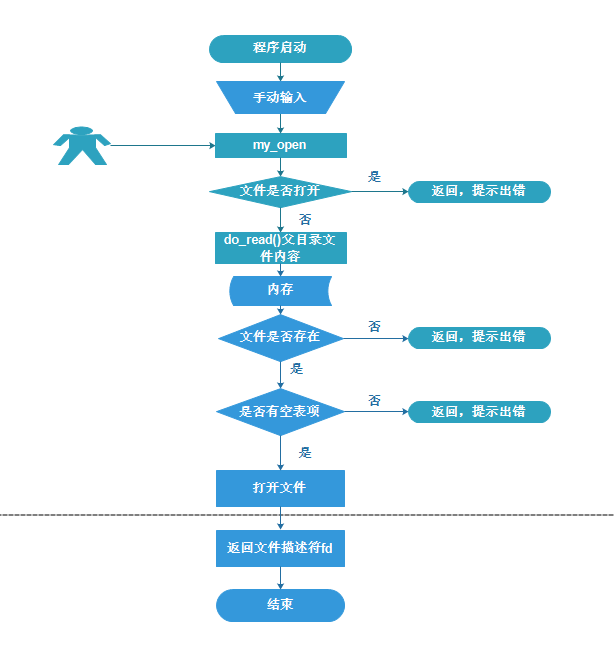
② 调用do\_read()读出父目录文件的内容到内存，检查该目录下欲打开文件是否存在，若不存在则返回-1，并显示错误信息；

③ 检查用户打开文件表中是否有空表项，若有则为欲打开文件分配一个空表项，若没有则返回-1，并显示错误信息；

④ 为该文件填写空白用户打开文件表表项内容，读写指针置为0；

⑤ 将该文件所分配到的空白用户打开文件表表项序号（数组下标）作为文件描述符fd返回。

流程图：



二.关闭文件

函数名：int my\_close(int fd)

功能：关闭前面由my\_open()打开的文件 描述符为fd的文件。

输入：fd 打开文件的id.

输出： 无。

函数需完成的工作：

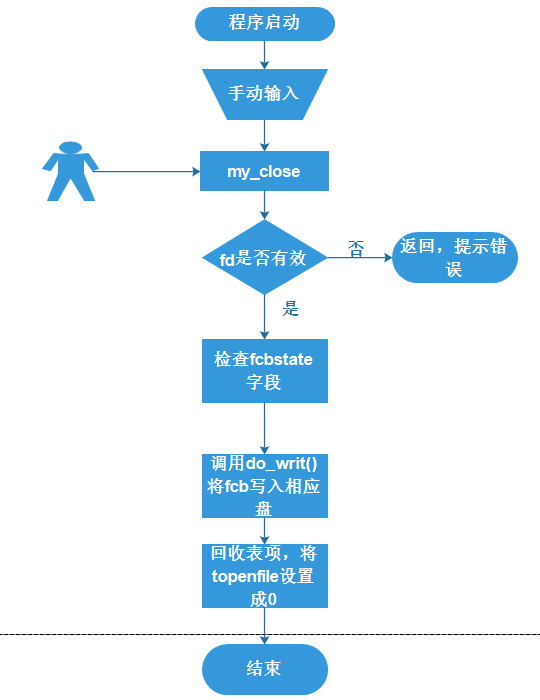
① 检查fd的有效性（fd不能超出用户打开文件表所在数组的最大下标），如果无效则返回-1；

② 检查用户打开文件表表项中的fcbstate字段的值，如果为1则需要将该文件的FCB的内容保存到虚拟磁盘上该文件的目录项中，方法是：打开该文件的父目录文件，以覆盖写方式调用do\_write()将欲关闭文件的FCB写入父目录文件的相应盘块中；

③ 回收该文件占据的用户打开文件表表项（进行清空操作），并将topenfile字段置为0；

④ 返回。

流程图：



三.目录更改

函数名：void my\_cd（char \*dirname）

功能：改变当前目录到指定的名为dirname的目录。

输入：dirname：新的当前目录的目录名；。

输出：无。

函数需完成的工作：

① 调用my\_open()打开指定目录名的父目录文件，并调用do\_read()读入该父目录文件内容到内存中；

② 在父目录文件中检查新的当前目录名是否存在，如果存在则转③，否则返回，并显示出错信息；

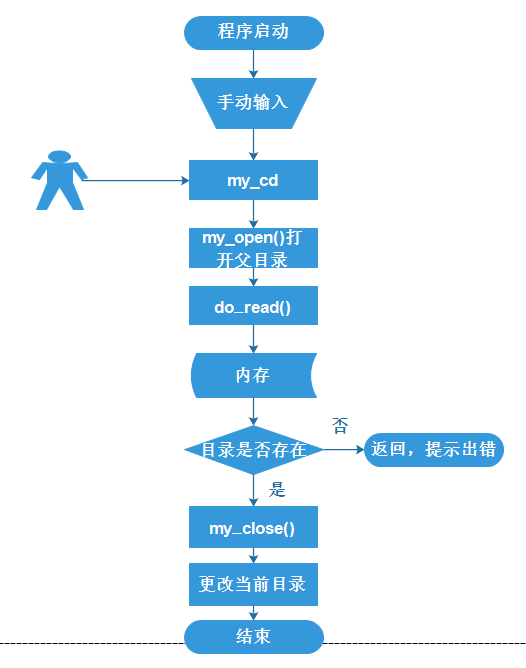
③ 调用my\_close()关闭①中打开的父目录文件；

④ 调用my\_close()关闭原当前目录文件；

⑤ 如果新的当前目录文件没有打开，则打开该目录文件；并将ptrcurdir指向该打开文件表项；

⑥ 设置当前目录为该目录。

流程图：



**纪文韬**

4．更改当前目录函数my\_cd()

（1）对应命令：my\_cd

（2）命令调用格式：my\_cd dirname

（3）函数设计格式：void my\_cd(char \*dirname)

（4）功能：改变当前目录到指定的名为dirname的目录。

（5）输入：

dirname：新的当前目录的目录名；

（6）输出：无

（7）函数需完成的工作：

① 调用my\_open()打开指定目录名的父目录文件，并调用do\_read()读入该父目录文件内容到内存中；

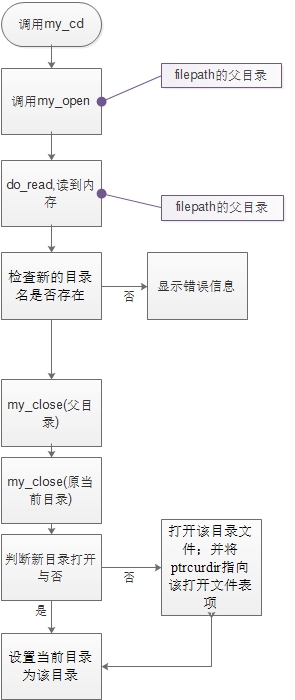
② 在父目录文件中检查新的当前目录名是否存在，如果存在则转③，否则返回，并显示出错信息；

③ 调用my\_close()关闭①中打开的父目录文件；

④ 调用my\_close()关闭原当前目录文件；

⑤ 如果新的当前目录文件没有打开，则打开该目录文件；并将ptrcurdir指向该打开文件表项；

⑥ 设置当前目录为该目录。



6．删除子目录函数rmdir()

（1）对应命令：my\_ rmdir

（2）命令调用格式：my\_ rmdir dirname

（1）函数设计格式：void my\_rmdir(char \*dirname)

（2）功能：在当前目录下删除名为dirname的子目录。

（3）输入：

dirname：欲删除目录的目录名。

（4）输出：无。

（5）函数需完成的工作：

① 调用do\_read()读入当前目录文件内容到内存，检查当前目录下欲删除目录文件是否存在，若不存在则返回，并显示错误信息；

② 检查欲删除目录文件是否为空（除了“.”和“..”外没有其他子目录和文件），可根据其目录项中记录的文件长度来判断，若不为空则返回，并显示错误信息；

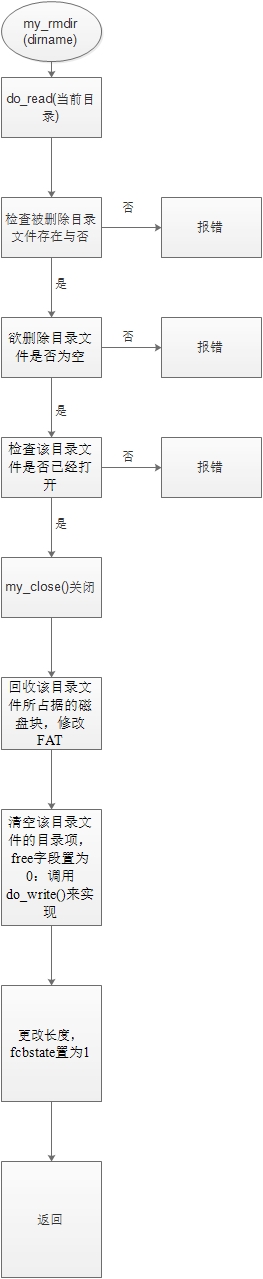
③ 检查该目录文件是否已经打开，若已打开则调用my\_close()关闭掉；

④ 回收该目录文件所占据的磁盘块，修改FAT；

⑤ 从当前目录文件中清空该目录文件的目录项，且free字段置为0：以覆盖写方式调用do\_write()来实现；

⑥ 修改当前目录文件的用户打开表项中的长度信息，并将表项中的fcbstate置为1；

⑦ 返回。



11．关闭文件函数my\_close()

（1）对应命令：my\_close

（2）命令调用格式：my\_close fd

（3）函数设计格式：void my\_close(int fd)

（4）功能：关闭前面由my\_open()打开的文件描述符为fd的文件。

（5）输入：

fd：文件描述符。

（6）输出：无。

（7）函数需完成的工作：

① 检查fd的有效性（fd不能超出用户打开文件表所在数组的最大下标），如果无效则返回-1；

② 检查用户打开文件表表项中的fcbstate字段的值，如果为1则需要将该文件的FCB的内容保存到虚拟磁盘上该文件的目录项中，方法是：打开该文件的父目录文件，以覆盖写方式调用do\_write()将欲关闭文件的FCB写入父目录文件的相应盘块中；

③ 回收该文件占据的用户打开文件表表项（进行清空操作），并将；

④ 返回。

