实验五：简单文件系统的实现

1. 实验目的

通过具体的文件存储空间的管理、文件的物理结构、目录结构和文件操作的

编程实现，加深对文件系统内部数据 、功能以及实现过程的理解。

1. 实验要求

(1) 在内存中开辟一个虚拟磁盘空间作为文件存储分区，在其上实现一个简单的基于多级目录的单用户单任务系统中的文件系统。在退出该文件系统时，应将该虚拟文件系统以一个文件的方式保存到磁盘上，以便下次可以再将它恢复到内存的虚拟磁盘空间中。

(2) 文件存储空间的分配可采用显式链接分配或其他的办法。

(3) 空闲磁盘空间的管理可选择位示图的办法。如果采用位示图来管理文件存储空间，并采用显式链接分配方式，那么可以将位示图合并到 FAT 中。

(4) 文件目录结构采用多级目录结构。为了简单起见，可不使用索引结点，其中每个目录项应包含文件名、物理地址、长度等信息，还可以通过目录项实现对文件的读和写的保护。

(5) 要求提供以下操作命令：

my\_format：对文件存储器进行格式化。

my\_mkdir：用于创建子目录。

my\_rmdir：用于删除子目录。

my\_ls：用于显示目录中的内容。

my\_cd：用于更改当前目录。

my\_create：用于创建文件。

my\_open：用于打开文件。

my\_close：用于关闭文件。

my\_write：用于写文件。

my\_read：用于读文件。

my\_rm：用于删除文件。

exitsys：用于退出文件系统。

1. 实验设计方案
2. 实验总体流程



（2）主要函数设计思路及程序流程图

1. 主函数int main()

功能：调用其他功能函数，提示用户输入命令并根据用户输入执行对应的功能函数。

函数设计思路：



2. 进入文件系统函数void startsys ()

功能：在虚拟磁盘上建立文件系统或读入已有的文件系统文件在虚拟磁盘上恢复文件系统。

函数设计思路：



3. 退出文件系统函数void exitsys ()

功能： 关闭当前打开的所有文件，将虚拟磁盘上的全部内容写到磁盘上的文件系统文件，释放用户打开表及虚拟磁盘的内存空间。

函数设计思路：



4. 磁盘格式化函数void my\_format ()

功能： 对虚拟磁盘进行格式化

函数设计思路：



5. 更改当前目录函数void my\_cd(char\* dirname)

输入参数：dirname 新的当前目录的目录名

功能：将指定的目录dirname设为新的当前目录

函数设计思路：



6. 创建子目录函数void my\_mkdir(char\* dirname)

输入参数： 新建目录的目录名dirname

功能：在当前目录下创建dirname子目录

函数设计思路：



7. 删除子目录函数void my\_rmdir(char\* dirname)

输入参数： 欲删除目录的目录名dirname

功能：删除当前目录下的dirname子目录

函数设计思路：



8. 显示目录函数void my\_ls()



功能：显示当前目录内容

函数设计思路：

9. 创建文件函数int my\_create(char\* filename)

输入参数：预创建文件的文件名filename

输出参数：若文件创建失败，则返回-1

功能：在当前目录下创建文件filename

函数设计思路：



10. 删除文件函数void my\_rm(char\* filename)

输入参数：欲删除文件的文件名filename

功能：在当前目录下删除文件filename

函数设计思路：



11. 打开文件函数void my\_open(char\* filename)

输入参数：欲打开文件的文件名filename

输出参数：打开文件的文件描述符

功能：打开指定文件

函数设计思路：



12. 关闭文件函数void my\_close (int fd)

输入参数：欲关闭文件的文件描述符fd

输出参数：欲关闭文件的父目录的文件描述符fd，没有父目录或者出错则返回-1

功能：关闭指定文件

函数设计思路：



12. 写文件函数int my\_write (int fd)

输入参数：待写文件的描述符fd

输出参数：实际写入的字节数

功能：将用户通过键盘输入的内容写到fd所指定的文件中

函数设计思路：



13. 实际写文件函数int do\_write (int fd, char \*text, int len, char wstyle)

输入参数：待写文件的描述符fd，存放待写入内容的指针text，本次要写入的字节数len，写方式wstyle

输出参数：实际写入的字节数

功能：将text中内容以wstyle的方式写入到磁盘文件中

函数设计思路：



14. 读文件函数int my\_read (int fd)

输入参数：待读文件的描述符fd

输出参数：实际读出的字节数

功能：读出指定文件的内容，并显示到终端上

函数设计思路：



15. 写文件函数int my\_write (int fd, int len, char \*text)

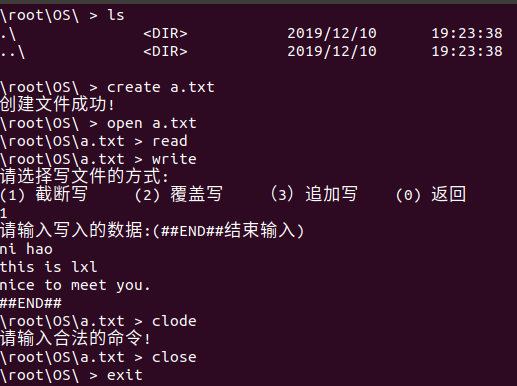
输入参数：待读文件的描述符fd，要从文件中读的字节数len，接收读出数据的字符数组text

输出参数：实际读出的字节数

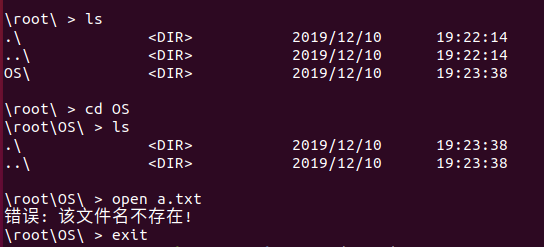
功能：读出指定文件中从读写指针开始的长度为len的内容到text中

函数设计思路：

1. 实验中遇到的问题
2. 退出文件系统后，再次启动上一次最后写入的文件不存在。



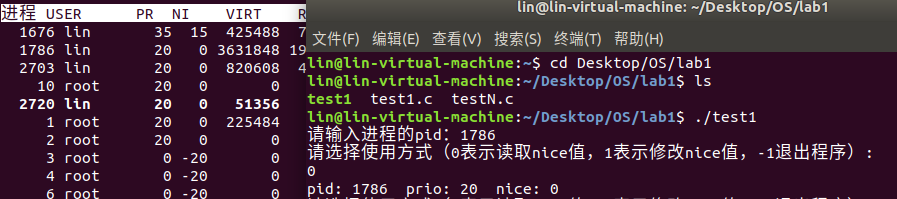
在OS目录下创建文件a.txt后，退出文件系统。再次运行文件系统，系统从myfsys文件中将文件系统在虚拟磁盘中恢复，得到的结果如下：



出错原因及解决方法：

1. 实验结果讨论

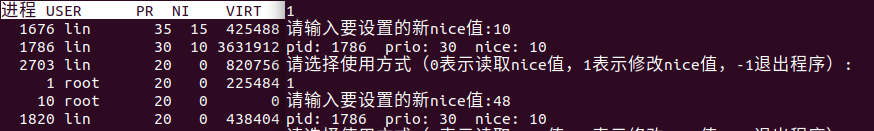
运行测试程序，实验结果的截图如下：



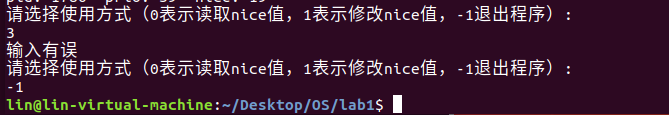
开始时，输入进程的pid为1786，输入0，获取进程的prio值及nice值，得到结果20 0，利用top命令查看进程1786的prio与nice值，看出结果正确；



接着，输入1，修改进程的nice值为10，由于10在nice值的范围（-20~19）内，进程的nice值得到修改，返回值为34 14，利用top命令看到进程1786的prio与nice值确实得到了修改。



接着，输入1，修改进程的nice值为48，由于nice值的取值范围是-20~19，超出范围根据set\_user\_nice函数的定义，进程的nice值不被修改。同样利用top命令验证结果，符合预期。



输入3，报错提示。

输入-1，退出程序。