

OS lab11: 模拟简易文件系统

简易文件系统模拟:

实验要求: 模拟一个简易的多级文件系统, 并且模拟用户在一个终端来进行目录切换和文件操作。在此次实验中, 所有文件名及路径名不包含空格, 换行等特殊符号。即可以通过一次 `cin >> str` 输入路径名或文件名。

程序运行

```
root$/%:
```

\$符号前是当前的用户, \$和%之间是当前的工作目录。

需要实现的功能, 指令格式与 demo 相同

➤ `man`: 显示功能目录

```
root$/%:man
Manual:
0) pwd (current work directory)
1) mkdir(p) (path) directory_name
2) dedir(p) (path) directory_name
3) search file_name
4) create(p) (path) file_name omode
5) cat(p) (path) file_name
6) delete(p) (path) file_name
7) write(p) (path) file_name buff wmode
8) reposition(p) (path) file_name pos
9) truncate(p) (path) file_name cnt
10) ls(p) (path)
11) open(p) (path) file_name
12) close(p) (path) file_name
13) regusr user_name password
14) delusr user_name [root only]
15) disusr
16) login user_name
17) lsfile
18) chmod(p) (path) file_name mode
19) exit
root$/%:
```

`man` 目录显示参数 命令后加 `p` 为提供路径选项。

➤ `pwd`: 显示当前工作目录的路径

```
root$/%:pwd
/
root$/%:
```

➤ `ls`: 显示当前工作目录下的所有子目录与文件

➤ `lsp path`: 显示路径为 `path` 目录下的所有子目录与文件

➤ `mkdir dir_name`: 在当前工作目录创建一个名称为 `dir_name` 的子目录。

➤ `mkdirp path dir_name`: 在路径为 `path` 目录下创建一个名称为 `dir_name` 的子目录

```

root$/%:mkdir cs302
root$/%:ls
cs302
root$/%:mkdirp /cs302/ cs203
root$/%:lsp /cs302/
cs203
root$/%:

```

- `cd dir_name`: 进入当前工作目录名称为 `dir_name` 的子目录
- `cdp path`: 进入路径为 `path` 的子目录

```

root$/%:cd cs302
root$/cs302/%:pwd
/cs302/
root$/cs302/%:cdp /
root$/%:pwd
/
root$/%:

```

- `dedir dir_name`: 在当前工作目录删除一个名称为 `dir_name` 的子目录。
- `dedirp path dir_name`: 在路径为 `path` 目录下删除一个名称为 `dir_name` 的子目录

```

root$/%:dedirp /cs302/ cs203
root$/%:lsp /cs302/
root$/%:dedir cs302
root$/%:ls
root$/%:

```

- `create file_name omode`: 在当前工作目录创建一个名称为 `file_name` 的文件，并指定文件的权限 `omode`。(omode 以 0~7 的数字表示，1 执行权限，2 写权限，4 读权限)
- `createp path file_name`: 在路径为 `path` 目录下创建一个名称为 `file_name` 的文件，并指定文件的权限 `omode`。

```

root$/%:create hello 7
root$/%:mkdir cs
root$/%:createp /cs/ hello 7
root$/%:ls
cs
hello
root$/%:lsp /cs/
hello
root$/%:

```

- `search file_name`: 查找所有文件名为 `file_name` 的文件，并输出文件所在路径

```

root$/%:search hello
/hello
/cs/hello
root$/%:

```

- open file_name: 在当前工作目录打开一个名称为 file_name 的文件。
- openp path file_name: 在路径为 path 目录下打开一个名称为 file_name 的文件
- lsfile: 显示所有打开的文件。
并逐行输出以打开文件的
" file_path file_name file_creator file_opener file_omode"

```
root$/%:open hello
root$/%:openp /cs/ hello
root$/%:lsfile
/ hello root root 7
/cs/ hello root root 7
root$/%:
```

只有打开的文件，且权限对应才可以进行相应的读写操作

- write file_name buff wmode: 在当前工作目录名称为 file_name 的文件按照 wmode 写入 buff
- write path file_name buff wmode: 在路径为 path 目录下名称为 file_name 的文件按照 wmode 写入 buff
wmode: 文件写入的方式，0 从末尾写入。1 从文件指针处开始替换。2 从文件指针处开始插入。写完后文件指针指向文件末尾。
- reposition file_name pos: 设置当前工作目录 file_name 文件的文件指针为 pos
- repositionp path file_name pos: 设置路径为 path 目录下 file_name 文件的文件指针为 pos
- cat file_name: 显示当前工作目录名称为 file_name 文件的内容
- catp path file_name: 显示路径为 path 目录下的 file_name 文件
- truncate file_name cnt: 保留当前工作目录名称为 file_name 文件的前 cnt 个字符
- truncatep path file_name cnt: 保留路径为 path 目录的名称为 file_name 文件的前 cnt 个字符

```
root$/%:write hello world 0
write ok
```

```
root$/%:cat hello
world
root$/%:writep /cs/ hello world 0
write ok
root$/%:catp /cs/ hello
world
root$/%:
```

```
root$/%:reposition hello 0
reposition ok
root$/%:write hello ww 1
write ok
root$/%:cat hello
wwrld
root$/%:truncate hello 2
root$/%:cat hello
ww
root$/%:._
```

- `chmod file_name`: 改变当前目录下 `file_name` 文件的权限。
- `chmod path file_name`: 改变路径为 `path` 目录下的 `file_name` 的文件权限
- `close file_name`: 关闭当前目录下 `file_name` 文件。
- `closep path file_name`: 关闭路径为 `path` 目录下的 `file_name` 的文件

```
root$/%:lsfile
/ hello root root 7
/cs/ hello root root 7
root$/%:close hello
root$/%:lsfile
/cs/ hello root root 7
root$/%:
```

- `delete file_name`: 删除当前工作目录下的 `file_name` 文件
- `deletep path file_name`: 删除路径为 `path` 目录下的 `file_name` 文件

```
root$/%:ls
cs
hello
root$/%:delete hello
file does not exist
```

```
root$/%:ls
cs
root$/%:
```

- `regusr username password`: 创建用户
- `delusr username`: 删除用户
- `disusr`: 显示所有用户 总有一个 `root` 用户，密码为 111
- `login username`: 登陆为 `username` 用户

```
root$/%:regusr DBG 123
register ok
root$/%:regusr GBD 321
register ok
root$/%:disusr
root
DBG
GBD
root$/%:login DBG
password:123
login ok
DBG$/%:
```

测试方式:

我们会预置一系列指令。对于合法指令，应正确执行得到相应的结果。

对于无法正确执行的指令，应给出对应提示:

目录已经存在: `directory already exists`

目录不存在: `directory does not exist`

文件已经存在: `file already exists`

文件不存在: `file does not exist`

文件写入成功: `write ok`

文件重定位成功: reposition ok
文件重定位失败(pos 位置不合理): reposition failed
文件已经打开: file already open
文件未打开: file does not open
文件操作失败(权限不对应): permission denied
用户注册成功: register ok
用户已经存在: user already exists
用户不存在: user does not exist
用户登陆成功: login ok
密码错误: password wrong
查看/删除用户权限不够: permission denied
没有找到文件: file does not found

具体的输出请参考 demo 截图。部分错误存在逻辑关系，如未打开文件进行文件操作，此时应仅返回文件未打开错误。

评分方式：将预置的指令与答案进行比对。若**完全一致**则通过该测试点。否则失败。

样例测试数据运行过程截图：

```
root$/%:mkdir cs302
root$/%:mkdir cs302
directory already exists
root$/%:ls
cs302
root$/%:create txt 1
root$/%:write txt hello 0
file does not open
root$/%:open txt
root$/%:lsfile
/ txt root root 1
root$/%:write txt hello 0
permission_denied
root$/%:chmod txt 7
root$/%:write txt hello 0
write ok
root$/%:cat txt
hello
root$/%:createp /cs302/ txt 3
root$/%:openp /cs302/ txt
root$/%:writep /cs302/ txt world 0
write ok
root$/%:catp /cs302/ txt
permission denied
root$/%:chmodp /cs302/ txt 6
root$/%:catp /cs302/ txt
world
root$/%:search txt
/txt
/cs302/txt
```

```
root$/%:search txt
/txt
/cs302/txt
root$/%:closep /cs302/ txt
root$/%:lsfile
/ txt root root 7
root$/%:close txt
root$/%:dedir cs302
root$/%:ls
txt
root$/%:mkdir 233
root$/%:cd 233
root$/233/%:regusr ww 222
register ok
root$/233/%:login ww
password:222
login ok
ww$/233/%:cdp /
ww$/%:disusr
permission denied
ww$/%:login root
password:111
login ok
root$/%:disusr
root
ww
root$/%:
```

评分标准:

实现了全部功能, 通过测试组数给分(最高 80)

代码质量(可读性, 注释, 长度, 逻辑是否清晰 最高 20)

对于未能实现所有要求功能的同学, 请在可以写一份报告, 写出实现的功能, 根据实现功能给分, 所给分数不超过 70 分。

作业以 学号.cpp 的形式上传