第四章编程作业

硬链接

代码逻辑

- 硬链接的部分实现主要考虑的是如何**访问特定目录下以及其所有子目录下的文件项。**

  结合在网络上查找的一些内容以及自己的理解，逐级访问目录文件是一个递归的过程，所以我也按照递归结构来设计访问目录结构的代码，对应于在`scanner.cpp`中实现的`ScannerFile(char* dirPath)`函数。

  具体逻辑并不复杂，在当前目录下不断访问目录项，如果访问到了目录文件那么递归调用这个过程访问下一级目录的内容。

  在实现上我依赖了linux提供的修改工作路径的函数`chdir`、访问目录项的函数`opendir`，以及结束访问`closedir`。

  特别要注意的是在递归调用修改工作目录到**子目录访问结束后要退回到上一级工作目录**继续遍历目录项以及在访问时注意**过滤`.`和`..`文件。**

- 其次还需要处理硬链接的判断，我采用了STL库中的模板map来记录下inode和filename的映射关系：

  （所以其实我这里为了方便使用模板写了一个c++程序）

  那么如果映射到同一个inode值上的filename的个数大于一，认为发生了硬链接，输出inode值和filename的内容。

  ```
  1 map<int, vector<string>> record;
  ```

编译运行

因为使用了c++实现的STL库，所以使用了g++编译。

```
1 $ g++ -o scanner scanner.cpp
```

第一次运行的时候我是在过滤`.`和`..`文件之前加入record中的，输出了2869个inode上发生了硬链接，而且全部为目录文件。也就是说它们也是通过硬链接的方式来访问到对应目录文件内容。

```
inode: 606591 output_1_20220225T060246  .
inode: 606592 output_logging_20220225T060246  .
inode: 606593 Gruntfuggly.todo-tree  .
inode: 606594 gnome-control-center  ..  .
inode: 606595 wacom  .
inode: 606629 nano  .
inode: 606666 google4  .
inode: 616379 expunged  .
Count:2869
```

之后我过滤了`.`和`..`文件使之不加入record。并且在不同目录下创建了`scanner.cpp`的两个硬链接文件：

```
$ ln scanner.cpp scanner_hardlink
$ ln scanner.cpp ../scanner_hardlink2.0
```

再次运行，和预期一致

```
oslab@oslab-VirtualBox:~/桌面$ ./scanner
Failed to open the path:lock
inode: 526778 scanner_hardlink2.0  scanner_hardlink  scanner.cpp
Count:1
```