**实验三：目录树的遍历**

* **实验目的**

掌握与文件和目录树有关的系统调用和库函数。

* **实验要求**

1、编写程序myfind

命令语法：

myfind <pathname> [-comp <filename> | -name <str>…]

命令语义：

（1）myfind <pathname> 的功能：

除了具有与程序4-7相同的功能外，还要输出在<pathname>目录子树之下，文件长度不大于4096字节的常规文件，在所有允许访问的普通文件中所占的百分比。程序不允许打印出任何路径名。

（2）myfind <pathname> -comp <filename>的功能：

<filename>是常规文件的路径名（非目录名，但是其路径可以包含目录）。命令仅仅输出在<pathname>目录子树之下，所有与<filename>文件内容一致的文件的绝对路径名。不允许输出任何其它的路径名，包括不可访问的路径名。

（3）myfind <pathname> -name <str>…的功能：

<str>…是一个以空格分隔的文件名序列(不带路径)。命令输出在<pathname>目录子树之下，所有与<str>…序列中文件名相同的文件的绝对路径名。不允许输出不可访问的或无关的路径名。

<pathname>和<filename>均既可以是绝对路径名，也可以是相对路径名。<pathname>既可以是目录，也可以是文件，此时，目录为当前工作目录。

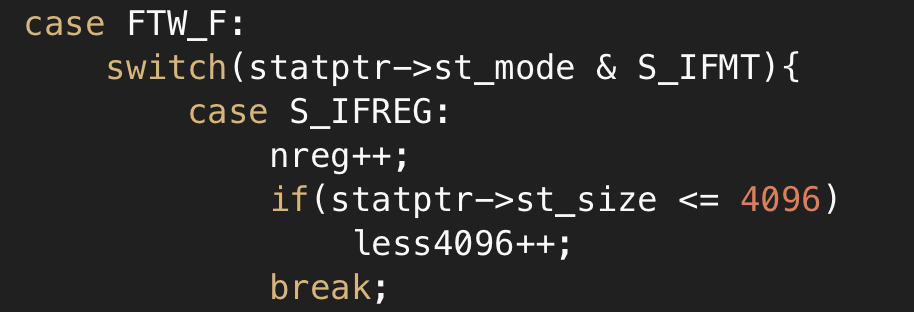
2、注意尽可能地提高程序的效率。注意避免因打开太多文件而产生的错误。

3、遍历目录树时，访问结点（目录项）的具体操作应当由遍历函数dopath携带的函数指针参数决定。这样程序的结构清晰，可扩充性好。

* **实验过程**

**功能1**

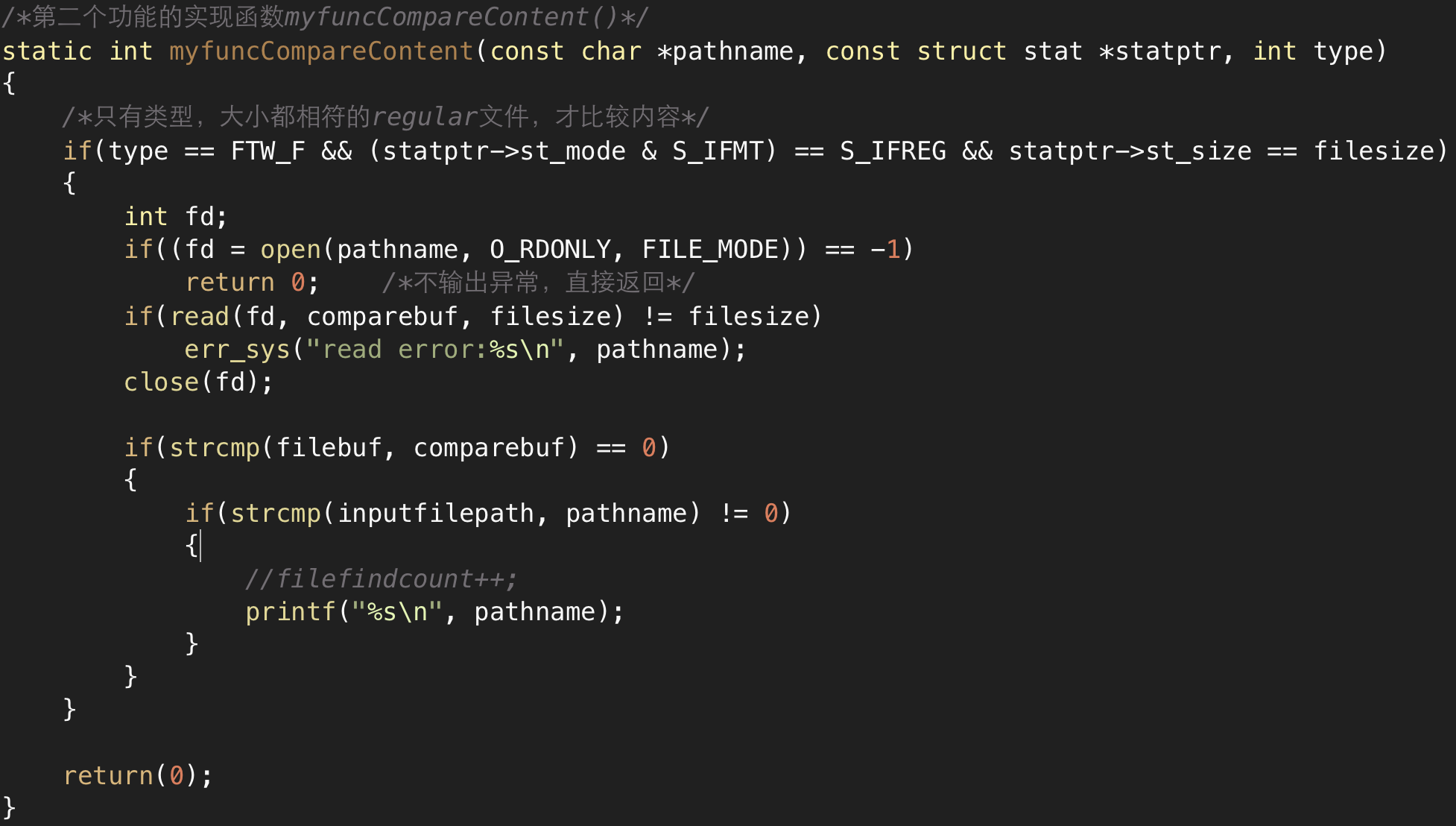
只要在原来的基础上加个对 文件长度不大于4096字节的普通文件 的判断就行了。

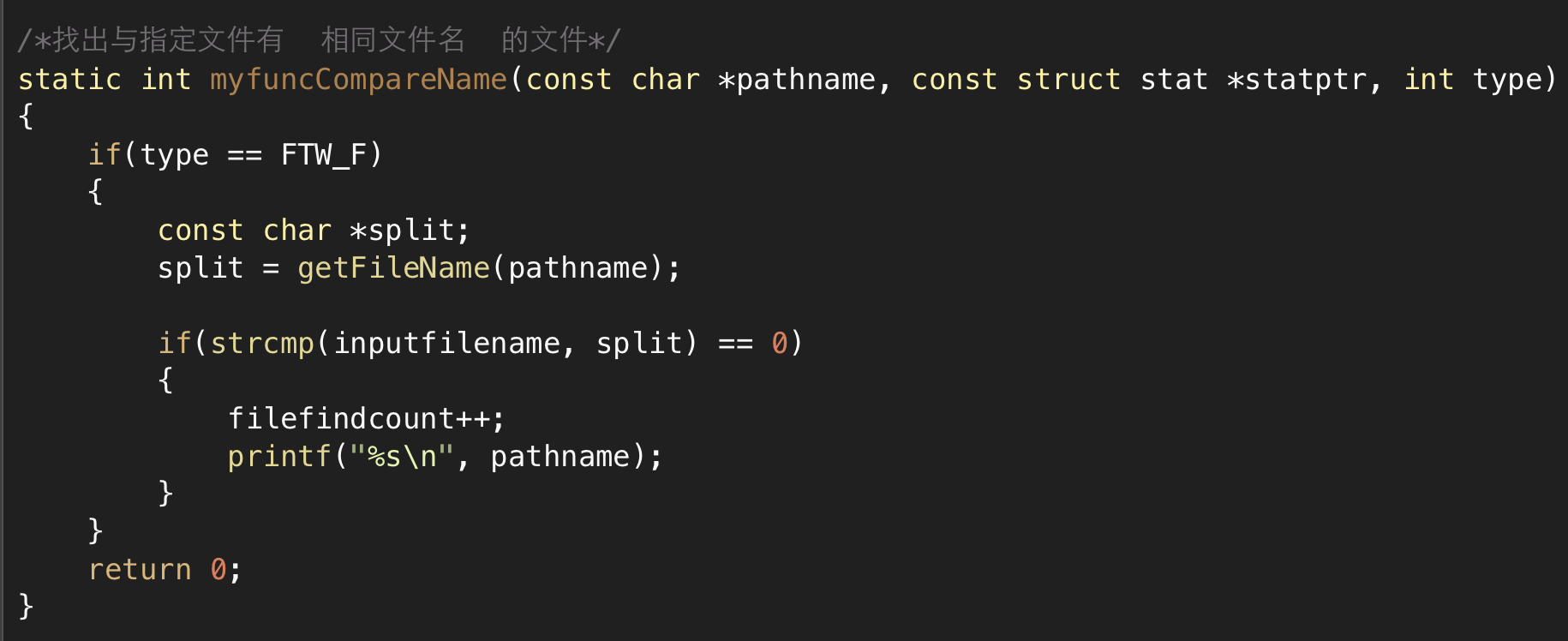
****

**功能2**

思路：对于普通文件，首先比较文件大小，文件大小相同再进行内容的比较。这样可以有效避免无用的打开、读入文件的操作。

要比较内容，首先需要读入，比较的话用 strcmp() 即可。



**功能3**

这个功能要解决的一个核心问题是：如何从路径中提取出文件名，比如从 fullpath = ‘./unix/ftw4.c’ 提取出 ftw4.c。一个简单的做法就是从末尾向前查找，直到找到第一个 ‘/’，则这个位置之后的内容就是文件名了，再进行比较即可。

利用 chdir() 和 getcwd() 可以获取到绝对路径.

* **实验结果**

本程序由：error2e.c, apue.h, ourhdr.h, myfind.c, pathalloc.c这5个文件构成。

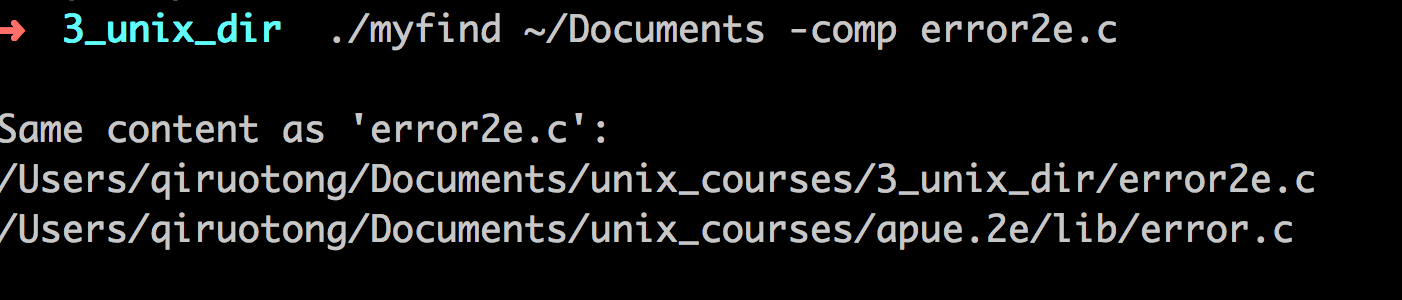
编译过程：

1 .执行语句：gcc -o myfind \*.c.进行编译;

2. 编译通过后会生成一个名字为myfind 的可执行程序；

3. 如执行myfind ~/Documents/unix, 则对应功能1

4. 如执行myfind ~/Documents/unix -comp error2e.c， 则执行功能2

5. 如执行myfind ~/Documents/unix -name error2e.c pathalloc， 则执行功能3

* **个人体会**

遍历目录树这个实验做起来确实不轻松，其实也是参考着别人的代码做出来的。期间还补习了函数指针，以及静态全局变量等知识漏洞。