

FileCount ReadMe:

【和谐呀~~~~~0(∩_∩)0~~】

【额，本程序是在 **Linux** 下写的，也是在 **Linux** 下测试的，理论上应该可以在 **windows** 上运行（试了一下好像可以），不过为了以防万一，如果真的不能在 **windows** 下正常运行（个人觉得其实不会发生这个“万一”的），麻烦在 **Linux** 平台下运行该程序，(∩_∩)】

输入格式：

1. 输入一个文件夹的**绝对路径**；
2. 输入你希望的排序方式，共有三种：*name*, *size*, *type*

三种排序方式说明：

(1) *name*

按照名称进行排序，排序的依据是默认的 **Unicode** 码，所以需要有两点注意，以免产生误解，

第一、字母序的**大写字母的顺序是排在小写字母前面的**（其实在 **ASCII** 码中也是这样的顺序），所以 **Wallpaper.jpg** 会排在 **a.jpg** 前面，因为 **W** 在 **a** 前面；

第二、由于 **Unicode 对汉字的排序不是按照拼音序进行的**（有一部分按照“常用度”进行排序，还有一部分按照“部首序”进行排序，具体挺繁琐的），所以文件排序输出的时候，中文文件的顺序并不是拼音序；

(2) *size*

对于一个文件夹里面的各个文件是按照 **File** 类的 **length()** 方法返回的大小从小到大排序的，但是对于其中的文件夹，由于 **length()** 对文件夹返回不定值，所以我在最终排序的时候，**先排文件夹，再按照文件的大小排文件**；

(3) *type*

针对类型进行排序，按照文件的后缀名的字典序排列文件。

输出说明：

对于一个文件夹，先进入其子文件夹进行统计（因为知道了子文件夹的内容统计才能计算出父文件夹的内容统计啊），输出其结果，然后回溯，最后才输出最外层的文件夹的统计信息。

感觉还是比较难说明，不如举个栗子（务必看两眼栗子啊，看两眼就知道我是怎么输出的了）：

【情形为：

1. 当前文件夹为 **00**，里面有两个文件夹 **Tex** 和 **Wallpaper**，还有两个 **ppt** 文件；
2. **Tex** 文件夹里面有四个文件，一个 **.sh**，三个 **.tex**；

3. Wallpaper 文件夹里面有 3 个.jpg 文件】

输入及解释如下：

Input a valid path please: /home/lpp/Documents/00/

//输入想测试的文件夹的**绝对路径**

Input a mode that you wish to sort the files(within "name", "size",
or "type"): name //输入排序类型

输出结果如下：

//分割线之间是针对单个文件夹的统计

==> ==> ==> /home/lpp/Documents/00/Text //当前统计的目录

Total Number: 4 //当前文件夹文件总数

Directory Number: 0 //当前文件夹子文件夹数

Execution Number: 0 //当前文件夹可执行文件数

M Number: 0 //当前文件夹多媒体文件数

Text Number: 0 //当前文件夹文本文件数

Directory MaxSize: 0 //最大的，以下同理

Execution MaxSize: 0

M MaxSize: 0

Text MaxSize: 0

Directory MinSize: 0 //最小的，以下同理

Execution MinSize: 0

M MinSize: 0

Text MinSize: 0

Directory Average Size: 0 //平均的，以下同理

Execution Average Size: 0

M Average Size: 0

Text Average Size: 0

Sort by name ==> ==> //以下是按照所输入的标准排序的列表

clear.sh

homework2.1.tex

homework3.tex

test.tex

==> ==> ==> /home/lpp/Documents/00/Wallpaper

Total Number: 3

Directory Number: 0
Execution Number: 0
M Number: 3
Text Number: 0

Directory MaxSize: 0
Execution MaxSize: 0
M MaxSize: 252058
Text MaxSize: 0

Directory MinSize: 0
Execution MinSize: 0
M MinSize: 67040
Text MinSize: 0

Directory Average Size: 0
Execution Average Size: 0
M Average Size: 137871
Text Average Size: 0

Sort by name ==> ==>
10117.jpg
10173.jpg
5900.jpg

==> ==> ==> /home/lpp/Documents/00 //最后才是最外层的父目录
Total Number: 10 //文件数 = 本目录的文件 + 子目录的文件，其实最后
这个的文件数就是总文件数
Directory Number: 2 //同上，这是总文件夹数
Execution Number: 0
M Number: 3
Text Number: 2

Directory MaxSize: 413613
Execution MaxSize: 0
M MaxSize: 252058
Text MaxSize: 533504

Directory MinSize: 11250
Execution MinSize: 0
M MinSize: 100067039
Text MinSize: 510969

Directory Average Size: 734701

Execution Average Size: 0

M Average Size: 137871

Text Average Size: 522236

Sort by name ==> ==>

. directory

Tex

Wallpaper

第九讲-过程抽象与异常处理.ppt

第九讲-过程抽象与异常处理.pptx

统计的类型——

在分类型统计的内容中，只统计以下内容：

多媒体文件：

音频文件(>=5 类 mp3、wma、wav、ogg、mpeg)、

视频文件(>=5 类 mp4 avi rmvb wmv mkv) 、

图片文件(>=5 类.jpg .jpeg .gif .png .bmp)

文本文件：

txt 文件、rtf 文件、MS office 文件、pdf 文件、lrc 文件

可执行文件：

exe 文件、dll 文件、bin 文件、jar 文件（针对 windows 平台）、bat 文件

备注：当统计 C 盘等系统盘的时候，不仅特别慢，还有可能会涉及到一些有权限的文件等等问题，所以统计数据会和微软的数据差一点儿。因此，建议不要统计过于庞大的盘符，测几个文件夹有效果了估计也差不多了。

毕竟同门师兄弟，点到为止嘛～