《电子证据分析》课程实验报告

|  |  |  |  |
| --- | --- | --- | --- |
| 实验名称 |  | 姓名 |  |
| 班级 |  | 学号 |  |
| 实验日期 |  | 指导老师 |  |
| 实验环境  (实验软硬件要求) | Win10、python、vscode编辑器、bandzip压缩工具 | | |
| 实验目的 | （给出本次实验所涉及并要求掌握的知识点）  掌握并且了解文件数据的读取方式  了解在windows操作系统中文件名的后缀的作用  能够使用变成进行对文件进行判断 | | |

|  |  |
| --- | --- |
| 实验内容 | 1. 将一个jpg文件和一个zip文件合并为一个jpg文件。查看在windows操作系统中，对这个合并的文件改动文件后缀发生的变化，并且简述其背后的原理。   首先随意使用一个JPG文件即可，将其命名为1.jpg，这里将实验使用的JPG文件保存路径如下：    然后，随意创建一个压缩包，这里使用bandzip软件将同目录下的check.py压缩成为zip文件。    随后在这里打开cmd窗口。（我使用了win10商店中的windows terminal，一样可以使用cmd，并且支持直接在对应目录打开cmd）  并且输入对应的指令，copy /b 1.jpg+check.zip 2.jpg  如下：    随后就可以发现一个名为2.jpg的文件出现在了这个目录下。    双击打开2.jpg可以看到就是一张和1.jpg一样的照片而已。    随后修改2.jpg的后缀，将jpg改为zip后重新打开查看。    在双击文件之后发现了这个文件又是一个压缩包，并且里面的内容就是之前的文件。  而有趣的是，如果在输入指令的时候交换了jpg和zip的顺序将指令输入成了：copy /b check.zip+1.jpg 3.jpg (之所以是3是因为之前用了2了就不好再用2了)  这样的话，这么弄出来的文件是没办法看到原本jpg文件的图片的，但是压缩文件的内容却还在。      这是因为操作系统在读取jpg文件的时候是从文件头开始读取的，这样也就是从”FFD8”开始读取，然后读到”FFD9”就结束了，而在读取zip文件的时候，操作系统不一样，它是从文件末尾开始找哈夫曼树的，然后再开始读取下去。所以在前面的时候直接使用jpg+zip操作的话就是可以两个文件都正常操作。而当指令的顺序反过来之后，因为文件头不是FFD8而后缀又是jpg，操作系统会没有办法识别出来，所以没办法看到图片。   1. 编写一个小程序，可以做到批量处理在之前实验一中的文件是否为隐藏了数据的JPG文件。   这边因为比较懒……就直接用了python，C语言的版本后续我会研究写出来的……  这个小程序主要是为了满足两个需求，一个是能够检测到jpg文件是否存在隐藏的数据。另外就是可以支持输入地址后递归查询所有的子目录。  代码如下：   1. *# -\*- coding: gbk -\*-* 2. import os 3. import sys 4. def RecursiveFileSearch(path): 5. if (not os.path.exists(path)): *# 判断文件(文件夹)是否存在* 6. print(path + "不存在！！") 7. return 0 8. if (os.path.isfile(path)): 9. answer = check(path) 10. if (answer == 1): 11. print(path + "只是个JPG文件") 12. elif (answer == 0): 13. print(path + "是个JPG文件，且里面存在隐藏的数据！") 14. elif (os.path.isdir(path)): 15. dirlist = os.listdir(path) 16. for dir in dirlist: 17. dir = path + '\\' + dir 18. RecursiveFileSearch(dir) 19. def check(file): 20. *# 首先需要检查是不是文件* 21. fsize = os.path.getsize(file) 22. if (fsize <= 4): 23. return 2 24. *# 优先排除掉过小的文件，避免过多的文件读取* 25. with open(file,'rb+') as f1: 26. temp = f1.read(2) 27. f1.close() 28. with open(file,'rb') as f2: 29. f2.seek(-2,2) 30. temp2 = f2.read(2) 31. f2.close() 32. if (temp == b'\xff\xd8'): 33. *# 这个ffd8是jpg文件开头* 34. if (temp2 == b'\xff\xd9'): 35. *# 这个ffd9是jpg文件结尾* 36. return 1 37. *# 文件是jpg文件，尾部没有其他的东西嗷* 38. else: 39. return 0 40. *# 貌似不是正常的JPG，尾部有东西！* 41. else: 42. return 2 43. *# 文件头有问题，至少从头查看不是JPG* 44. def main(argv): 45. RecursiveFileSearch(argv[1]) 46. if \_\_name\_\_ == "\_\_main\_\_": 47. *# 下面提供两种启动的方式，一种使用了sys库，使用的是命令行，另外一种是简单地input函数* 48. *# main(sys.argv)* 49. path = input("请输入路径：") 50. RecursiveFileSearch(path)   阿这……这个代码是在上课的时候码的，可能还会存在一些小问题，但是至少我目前在使用的时候没有遇到……     1. jpg就是我之前截的图片，2.jpg是我之前用指令生成的合并文件     的确可以识别出来两者的区别，并且我输入目录的时候是到python一层的，但是它识别到了后面子目录里的文件。 |
| 实验总结 | （对本次实验涉及到的知识点的重难点归纳、实验心得、思考与建议）  文件隐藏的知识其实还是对之前的文件数据的进一步的学习。个人认为不难理解。  编程如果使用python的话确实十分简单，但是如果使用C语言的话就会遇到一些奇怪的问题，这里想请问一下老师如果用C语言写的话，要将读取的二进制流写入到哪一种的数据类型里面，这里不是太清楚…… |
| 评分 |  |