

**并 行 计 算**

课程实验报告

实验名称：Python基础编程

任课教师：徐悦甡

课程班级：15级 云计算方向

学号姓名：15130130273 石明皓

提交日期：2018年6月6日

**软件学院本科生《并行计算》**

**课程实验报告**

1. **实验名称**

第3次实验：Python基础编程

1. **实验日期**

2018年6月6日 软件学院实验室G346

1. **实验学生**

15130130273 石明皓

1. **实验目的**

本次实验通过给出2个Python基础编程的例子，要求运用Python基础语法正确编写并实现其程序逻辑。熟悉Python的编程模式与开发方法，以备日后Python并行程序设计的深入学习。

1. **实验内容**

题目一：

使用Python基础编程，统计两个文件中单词“book”出现的总次数。有两个文件，file\_1.dat与file\_2.dat，文件内容如下：

file\_1.dat内容：and, with, we, me, university, with, book, computer, country, book

file\_2.dat内容：bag, boy, book, school, teacher, student, book, book

要求正确地使用Python的文件操作，首先统计出file\_1.dat中出现2次，file\_2.dat中出现3次，然后统计出“book”出现的总次数（5次）。

题目二：

使用Python基础编程，完成快速排序函数的编写。

输入：从文件data.dat或data.txt中输入一列没有顺序的数，如：

3,7,12,5,3,10,11,9,4,2,4

输出：经过快排，正确的排序结果，如：

2,3,3,4,4,5,7,9,10,11,12

1. **程序思路、结构**

题目一：

先分别读入file\_1.dat和file\_2.dat，注意文件读取操作应加入错误处理try…except，将其中的内容按行读出，分别转换为两个字符串line1和line2，使用replace函数消除这两个字符串中的空格，并使用split函数按逗号分隔这两个字符串的各个单词，分别存入各自的的字符串数组word1和word2中；

循环读取word1和word2中的每个字符串，判断是否为“book”，若是，则各自的计数值sum1或sum2自增一；

遍历完毕后，file\_1.dat和file\_2.dat中的“book”个数就统计出来了，而sum1+sum2即为“book”出现的总次数；

将sum1、sum2和sum1+sum2写入文件中，完成。

题目二：

定义partition(list, low, high)函数用于确定快速排序中的分界点pivot的最后位置，然后定义quicksort(list, low, high)函数接收一列数字执行快排操作；

main函数中将待排序的数字定义到list中，向quicksort()函数传入list，0，len(list)-1这三个参数，函数执行完毕后，原list中存储的数字便完成了排序；

最后将list存入指定文件中即可，注意添加读写文件的异常处理。

1. **程序代码**

题目一：

def book\_sum():  
 try:  
 f1 = open('./file\_1.dat', 'r')  
 except IOError:  
 print('File open fail!')  
  
 try:  
 f2 = open('./file\_2.dat', 'r')  
 except IOError:  
 print('File open fail!')  
  
 line1 = f1.readline()  
 line2 = f2.readline()  
  
 word1 = line1.replace(' ', '').split(',')  
 word2 = line2.replace(' ', '').split(',')  
  
 sum1 = 0  
 for i in word1:  
 if i == 'book':  
 sum1 = sum1 + 1  
 print('file\_1.dat中book出现的次数：', sum1)  
  
 sum2 = 0  
 for i in word2:  
 if i == 'book':  
 sum2 = sum2 + 1  
 print('file\_2.dat中book出现的次数：', sum2)  
  
 print('book出现的总次数：', sum1 + sum2)  
  
 filename = 'python\_book\_sum\_output.txt'  
 with open(filename, 'w')as f:  
 f.write('file\_1.dat中book出现的次数：' + str(sum1) + '\n')  
 f.write('file\_2.dat中book出现的次数：' + str(sum2) + '\n')  
 f.write('book出现的总次数：' + str(sum1 + sum2) + '\n')  
 f.close()  
  
if \_\_name\_\_ == '\_\_main\_\_':  
 book\_sum()

题目二：

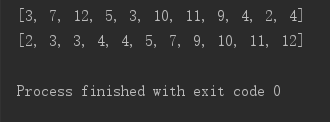
def quicksort(list, low, high):  
 if low < high:  
 middle = partition(list, low, high)  
 quicksort(list, low, middle - 1)  
 quicksort(list, middle + 1, high)  
  
def partition(list, low, high):  
 x = list[high] # 最后一个数作为pivot  
 i = low - 1  
 for j in range(low, high):  
 if list[j] <= x:  
 i = i + 1 # i表示小于pivot的最后一位数  
 list[i], list[j] = list[j], list[i]  
 list[i + 1], list[high] = list[high], list[i + 1] # j遍历完整个list后，最后一个数pivot换到小于pivot和大于pivot之间  
 return i + 1 # 返回分界点  
  
if \_\_name\_\_ == '\_\_main\_\_':  
 list = [3, 7, 12, 5, 3, 10, 11, 9, 4, 2, 4]  
  
 print(list)  
 quicksort(list, 0, len(list) - 1) # low=0,high=10  
 print(list) # list = [2, 3, 3, 4, 4, 5, 7, 9, 10, 11, 12]  
  
 filename = 'quicksort\_output.txt'  
 try:  
 with open(filename, 'w')as f:  
 f.write(str(list))  
 f.close()  
 except IOError:  
 print('File open fail!')

1. **实验结果**

题目一：



从程序运行结果可知，book在file\_1.dat中出现次数为2，在file\_2.dat中出现次数为3，出现的总次数为5，结果无误，说明Python实现“book”计数成功。

题目二：

从程序运行结果可知，原数字序列排序成功，排序结果无误，且代码实现的排序运用了快速排序的思路，说明使用Python实现快速排序成功。

1. **总结建议**

经过本次实验，我对Python程序设计有了更深的理解，进一步熟悉了Python的基础编程语法，为今后难度更高的Python并行程序编程打下基础。