



西安电子科技大学  
XIDIAN UNIVERSITY

软件学院  
School of Software

# 并行计算

## 课程实验报告

实验名称: Python 基础编程

任课教师: 徐悦牲

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

# 课程实验报告

## 一、实验名称

第 3 次实验：Python 基础编程

## 二、实验日期

2018 年 6 月 6 日 软件学院实验室 G346

## 三、实验学生

15130130273 石明皓

## 四、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。

## 五、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 中输入一系列没有顺序的数，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

## 六、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)函数接收一系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。

## 七、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!')
```

## 八、实验结果

### 题目一:

```
file_1.dat中book出现的次数: 2
file_2.dat中book出现的次数: 3
book出现的总次数: 5

Process finished with exit code 0
```

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：

```
[3, 7, 12, 5, 3, 10, 11, 9, 4, 2, 4]
[2, 3, 3, 4, 4, 5, 7, 9, 10, 11, 12]

Process finished with exit code 0
```

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。

## 九、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 并程序编程打下基础。