Word Conut 任务上机实习报告

薛飞跃 1700017831

一．算法步骤：

1. 阅读示例代码，map 阶段，将文本使用 StringTokenizer 生成迭代器进行分割，每次执行 map，对分割出的 string 中的每个 word 进行迭代，以（word，1）的形式写入 context。reduce 阶段，对每一个不同的 key 值，计算对应 value 的和，写入输出文件。
2. 对示例代码进行调整如下图，使用正则表达式，将非字母数字和’号的符合代替为空格。

```
public void map(Object key, Text value, Context context
                ) throws IOException, InterruptedException {
    String pattern = "[^a-zA-Z0-9-']";
    String line = value.toString();
    line = line.replaceAll(pattern, " ");

  StringTokenizer itr = new StringTokenizer(line);

  while (itr.hasMoreTokens()) {

    word.set(itr.nextToken());

    context.write(word, one);

  }

}
```

3. 分别执行两次代码，进行结果对比。

二．实验结果：

初始代码的运行结果：

```
year,   5
year.   2
year.?  1
years   18
years'  1
years.  3
yes,    1
yesterday       2
yesterday,'     1
yesterday.      2
yesterday.'     1
yet     2
yet!    1
you     65
you!'   1
you,    1
you,'   3
you.    6
you.'   1
young   15
your    6
yours   1
‘       1
‘It’s   1
‘I’m    1
        4
冷遇    1
?       2
alice@Master:~$
```

使用正则表达式替换后代码运行结果:

```
words   2
work    12
work-shop       2
worked  1
workers 4
working 7
works   1
world   14
worried 4
worry   1
worse   2
worst   2
worthless       1
would   36
wound   3
wrapped 2
write   2
wrong   4
wrote   1
yards   1
year    18
year's  1
years   21
years'  1
yes     1
yesterday       6
yet     3
you     77
young   15
your    6
yours   1
xuefeiyue@xuefeiyue-Vi
```

三．结果分析

原始代码中, 很多词和后面的符合也被统计成一个单独的词, 如“you”和“you, ”

实际上应该是一个词，而正则表达式替换后，“you”的各种形式被正确统计为同一个单词。