

DISTRIBUTED SYSTEM: HADOOP MAPREDUCE

ALI ABDULMADZHIDOV

9 December 2016

Mapper.py

```

#!/usr/bin/python
# -*- coding: utf-8 -*-

import sys
import re
import os

special = ("Media:", "Special:", "Talk:", "User:", "User_talk:",
"Project:", "Project_talk:", "File:", "File_talk:", "MediaWiki:",
"MediaWiki_talk:", "Template:", "Template_talk:", "Help:",
"Help_talk:", "Category:", "Category_talk:", "Portal:", "Wikipedia:",
"Wikipedia_talk:")
extens = (".jpg," ".gif," ".png," ".JPG," ".GIF," ".PNG," ".txt,"
".ico")

def readInput(file):
    for line in file:
        yield line.split()

def checkSpecial(word):
    splitted = word.split(":")
    return len(splitted)>1 and splitted[0]+":" not in special

reg = "-([0-9]+)-"

def main(separator='\t'):
    data = readInput(sys.stdin)
    filename = os.environ["mapreduce_map_input_file"]
    date = re.search(reg, filename).group(0)[1:-1]
    for words in data:
        if words[0] == "en" and checkSpecial(words[1]) and words[1]
[0].istitle() and words[1][len(words[1])-4:] not in extens:
            print '%s%s%s%s' % (words[1], separator, words[2],
separator, date)

if __name__ == "__main__":
    main()

```

Reducer.py

```
#!/usr/bin/python
# -*- coding: utf-8 -*-

from itertools import groupby
from operator import itemgetter
import sys

def read_mapper_output(file, separator='\t'):
    for line in file:
        yield line.rstrip().split(separator)

def main(separator='\t'):
    data = read_mapper_output(sys.stdin, separator=separator)
    for item, group in groupby(data, itemgetter(0)):
        try:
            a = {}
            total_count = 0
            for item, count, date in list(group):
                a[date] = count
                total_count+=int(count)
            # total_count = sum(int(count) for item, count, date in
group)

            if total_count<=100000:
                print "%s%s%d" % (total_count, separator, item)
            else:
                resp = "%s%s%d" % (total_count, separator, item)
                for date, count in a.iteritems():
                    resp+="{0}{1}:{2}".format(separator, date, count)
                print resp
        except ValueError:
            pass
if __name__ == "__main__":
    main()
```

hadoop_start.sh

```
hadoop jar /usr/lib/hadoop-mapreduce/hadoop-streaming-2.6.0-
cdh5.9.0.jar \
    -D mapred.output.compress=true \
    -D
mapred.output.compression.codec=org.apache.hadoop.io.compress.GzipCodec
\
    -input /inputdata/* \
    -output /tmp/student9/output \
    -file mapper.py \
    -file reducer.py \
    -mapper mapper.py \
    -reducer reducer.py
```

Output

10 top lines

```
214334 Independence_Day:_Resurgence      20160701:1210    20160702:1129
20160703:1399    20160704:1634    20160705:1270    20160706:908
20160707:710
163809 The_Purge:_Election_Year          20160701:1488    20160702:874
20160703:1185    20160704:1059    20160705:766     20160706:542
20160707:708
159028 Batman_v_Superman:_Dawn_of_Justice 20160701:940
20160702:995     20160703:983     20160704:755     20160705:931
20160706:1059    20160707:918
103455 UFC_Fight_Night:_dos_Anjos_vs._Alvarez 20160701:254
20160702:315     20160703:319     20160704:274     20160705:230
20160706:329     20160707:647
81787  Captain_America:_Civil_War
76673  X-Men:_Apocalypse
75300  The_Ultimate_Fighter:_Team_Joanna_vs._Team_Cl%C3%A1udia
71796  List_of_Naruto:_Shippuden_episodes
61939  The_Purge:_Anarchy
54160  The_Divergent_Series:_Allegiant
```