CSE 2431 LAB 4 AUTUMN 2015

Electronic Submission: 11:59 pm Wednesday, Dec 9th

1. Goal

1) Learn how to operate files
2) Complete your mapreduce simulator

2. Introduction

Following Lab 3, in this project, you will need to implement functions to allow your mappers to read from files and your reducers to output to files.

This is your last project, so this time I am not providing any sample code. You will need to do everything by yourself. This includes but not limited to understanding the requirements, understanding related system calls, organizing your code, changing Makefile when necessary, and testing your code.

3. Requirement

3.1 Input

Your program should take three arguments as input:
1) m: number of mappers
2) r: number of reducers
3) dir: a directory that contains m input files

An example is

```
./main 3 2 /users/yangwang/testlab4
```

The directory will contain m input files input0, input1, ... You should assign each to a mapper.

Each file is a text file that contains multiple lines. Each line is a string of text that only contains a-z and space.
An example is:

```
aaa bb cc dddd
abc dd ee fffff
```

A text file could be very large. It may not fit into your machine's memory.
You can assume each line is small enough to fit into your machine's memory.
You can assume the total number of distinct words is not too large so that they fit into your machine's memory.

3.2 Output

Your program should output r text files in the same directory: output0, output1, ... Each reducer should output to one file.

Each file is a text file that contains multiple lines. Each line contains a word and its count. They should be separated by space.

An example is

```
aa 1
bb 2
dddd 4
```

Same as in Lab3, there is no requirement of how words are ordered. A word can be outputted in any file at any line. Of course, a word should not appear more than once.

4. **Hints**

You should get familiar with the following system calls: fopen, fclose, fprintf, getline.

I am not providing sample code this time. Please use google or use the linux "man" command to find how to use them properly. **Read documents carefully before writing your own programs. If you are not sure about some of their arguments, write a small program to test them.**

You are allowed to use other system calls if you are more familiar with them. Once again, if your code needs any special compiling argument, make sure you change the Makefile accordingly and submit your Makefile.