EEL 5733/4732 Advanced Systems Programming
Assignment 1

In this assignment, you are going to implement three programs: Mapper, Reducer, and Combiner.

Mapper program gets the input from the standard input that contains tuples in the form (userID, action, topic). userID is a 4-digit identification for the users of some social media site. Action can be one of the following letters: P for posting, L for liking, D for disliking, C for commenting, and S for sharing. Topic is a string of maximum 15 characters. The mapper program processes these tuples to generate a weighted profile in the form of (userID, topic, score) based on a set of scoring rules, which are defined as P=50, L=20, D=-10, C=30, S=40. You can assume that the tuples are sorted according to the userID field, but tuples that belong to the same userID may not be sorted according to the topic. As an example, the input may look like

(1111,P,history)
(1111,S,entertainment)
(1111,L,history)
(1111,L,cosmetics)
(2222,L,sports)
(2222,S,sports)
(3333,S,photography)
(3333,L,art)
(3333,P,art)

and the correct output (on the standard output) would be

(1111,history,50)
(1111,entertainment,40)
(1111,history,20)
(1111,cosmetics,20)
(2222,sports,20)
(2222,sports,40)
(3333,photography,40)
(3333,art,20)
(3333,art,50)

Note that the Mapper program should output each tuple (userID, topic, score) as soon as it processes the corresponding (userID, action, topic) tuple (important for better performance!). Also, each tuple should be output on a separate line as shown above.

Reducer program gets the tuples of the form (userID, topic, score) from the standard input and generates tuples of the form (userID, topic, total score) on the standard output. Assuming that it gets the output of Mapper as shown above as the input, it generates the output by adding up the scores for each topic as follows:

(1111,history,70)
(1111,entertainment,40)
(1111,cosmetics,20)
(2222,sports,60)
(3333,photography,40)
(3333,art,70)

Like Mapper's input, the tuples are sorted according to the userID field but tuples that belong to the same userID may not be sorted according to the topic. **Note that the Reducer should output the tuple as soon as it realizes that there won't be any more tuples that belong to the same user**. You can assume that the Reducer's input has a single tuple per line.

Combiner program gets its input from the standard input that contains tuples in the form (userID, action, topic) on each line and generates tuples of the form (userID, topic, total score) on each line. Each tuple is output on a separate line. **You should use fork, exec, pipe, and dup2 system calls in the Combiner and reuse the Mapper and Reducer programs such that the Combiner, Mapper, and the Reducer get executed in separate processes**. Please test your code on a Linux system (Ubuntu 20 and newer versions are recommended). **Important Note:** Implementations of Combiner programs that do not make effective use of these system calls will not get any credit.

Make sure that all programs detect the end of file character while reading from the standard input.

Note: Even though you will implement each program as getting their input from the standard input, you can still provide the input in a file using the input redirection operator <. Similarly, you can use the output redirection operator > to write the output to a file. So, you can test each of the programs you will implement as follows:

$ mapper < input.txt > mapper_output.txt

$ reducer < mapper_output.txt > reducer_output.txt

$ combiner < input.txt > combiner_output.txt

Make sure combiner_output.txt is the same as reducer_output.txt content-wise, which you can check using the diff command. Keep in mind that the Combiner program needs to be implemented as a multi-process program.

The TA will test your code assuming that it accepts input from the standard input and writes to the standard output. So, please make sure to test your code as suggested above. Otherwise, grading of your submission will be delayed.

Submission: Submissions will be through the ELearning portal by the due date. You should submit your source code (in C/C++) for the three programs along with a Readme file and preferably a Makefile to show the details of compiling your code.

Late submission policy: For every 6 hours late, 10% of the total score will be deducted.