

Havan Patel -hp397
Charmian Goh -csg102

HOW TO COMPILE: You can compile 3 ways:

- 1) `./your_exefile -c fieldname -d directorypath -o outputpath :`
ex)./your_exefile -c movie_title -d /ilab/users/hp397 -o /ilab/users/hp397/outputdir
- 2) `./your_exefile -c fieldname -d directorypath :`
ex)./your_exefile -c gross -d /ilab/users/hp397
- 3) `./your_exefile -c fieldname`
ex)./your_exefile -c director_name

FIRST:

So, the program first checks for valid input is entered or not. If userinput is not valid the program will exit, give error message and stop running the program. You have to try again. We check this fields by using `getOpt()` function.

1. The program check if the `-c` column name entered is valid for example:
“-c random_name” will abort the entire project since there is no such field. There has to be same name as the 27 fields like: “color, or director_name,or gross, or movie_title
.....etc”
2. If `-d` is not provided it will search the current directory path you are in. And if `-d` is specified it will check the given directory path (for example: `-d /ilab/users/hp397/proj1`)
3. If `-o` is not provided it will output the sorted csv files in the same directory as the original csv file. And if the `-o` is specified it will output all sorted csv file in that path provided (for example:(`-o /ilab/users/hp397/proj1/output_directory`)

Second:

After, the **FIRST** step above is successful it will search directories and .csv files.

HOW CSV IS FORKED=>In order to fork .csv file, we check, 1) if the file is “.csv” and 2) then it will check if the file is not “sorted.csv”. meaning that, if will ignore all .csv file that has name “sorted” in it. If both or one of the condition is false it will not sort and fork the processes on those files. 3) if both condition (1,2) are successful then it will fork and sort that csv file.

However, if the header of that “.csv” file is invalid format and it doesn't contain all 28 fields, it will not sort the file, but it will fork. 4) After sorting the csv files the output of that .csv file will be in the same directory as original .csv file was, if `-o` is not given. If `-o` is given it will output all those sorted .csv file in the path that the user has given. The sorting and reading algorithm is same as the proj0.

NOTE ABOUT SORTING: I have made all int(fields) to doubles and if the data in double column are empty it will write “0” to it and in char if the values are empty it will stay empty “ ”. Also, if the movie_title contains comma in quotation ex)“The, Dark Knight” it reads it with comma in it. I

don't remove the commas from the quotation marks. It sort with commas. Another, there is no special handling with special characters.

HOW DIRECTORIES ARE FORKED=>In order to fork directories, we check 1) if the path given by -d is valid then, it will go through all directories and fork them and also look for ".csv" files, or if -d is not given by user then it will go through the current directories you are in. The library we used to deal with going through directories we used "dirent.h" 2) if the program is unable to open directory it will throw an error and exit the program, however it will fork and print the main's process only or the initial process.

BOTH .CSV AND DIRECTORIES ARE SEARCHING RECURSIVELY.

THIRD:

In the terminal i then print all the pids and number of processes which also includes(the processes of invalid files as well). I print them onto the terminal(stdout). However there is just small gap between printing the "child processes" and "total number of processes"