

Brian Ma 171001550

Jonathan Hong 170002315

Systems Programming Assignment 1

Documentation

While designing our project, we thought about completing the independent functionalities first. So we began with making sure the given arguments were given and stored. We ran into almost no problems, except for figuring out which flags could be left optional and where any edge cases could break our code. After testing the flags, we could move onto directory and file traversals. The program searches through the appropriate directories using the `traverse` function. `Traverse` opens a file stream in the given or current directory and checks the `dirent` structure of each file to see if its a file or a directory. If it is a file and it is a csv file, the program forks and `sortCSV` is called, and if it is a directory, the program is still forked but `traverse` is called, and the program continues from there. Onto the fork processes, every time we found a directory or .csv file, we created a child process using `fork()` and differentiated it from the parent by using the return value of the fork system call. We made sure to make the parents `wait()` in order to avoid and zombie or orphan processes.

In terms of problems we faced, mainly we had problems with the makefile. Because we did not properly set up a makefile for the last assignment, we had no experience in how to properly prepare the two .c files and the header files to use a makefile. We had many errors that did not arise when we simply included `mergesort.c` that were very time consuming to solve. Finally once all the errors were fixed we created a thorough directory structure to test all the edge cases we could think of. The directory structure used to test was as follows. Directories are marked with `dir-`. The purposes to of each file is shown

- Source Directory
 - `dir-dir1` //to test if the program would only search in a given directory
 - `dir-dir3`
 - `Alfadh.csv` //to test if the program would ignore improperly formatted csv files
 - `Dir-dir2.csv` //to test if the program could differentiate between an actual csv file or just a folder that ends in .csv
 - `dir-Dir4`
 - `Movie_metadat.csv` //to test if the program could sort a csv
 - `You.csv` //to test if the program would ignore improperly formatted csv files
 - `You` //to test if the program would ignore non csv files
 - `Wowsogood.csv` //to test if the program could sort a csv
 - `dir-brian` //to test if the program would output to a given directory
 - `scannerCSVsorter.c` and all other code files

- Haha.csv //to test if the program would search in current directory if no input dir was given

Test cases section part

How To Use Our Code:

Step 1: run `make` and it compiles with gcc

Step 2: execute `./scannerCSVsorter -c <column> -d <inputDir> -o <outputDir>