Andrew Hernandez & Michael Belmont

Project Part 2

# Design, Difficulties, and Assumptions

Coming into this project, we both were clueless about how multi-threading worked as well as what a mutex or a semaphore were. Multi-threading is more complicated than multi-processing as well so we were aware that it would prove to be a challenge.

This program requires two parameters: -c (to specify sorting by columns) and a column name to sort on.

There are two optional flags: -d (followed by a specified directory to search through) and -o (followed by a specified directory to output to).

Assuming proper parameters, the program searches through the directory given (or the current directory if -d is not specified) and sorts each valid CSV within the directory and subdirectories.

Upon finding each subdirectory and each CSV, the program creates a thread that either navigates through the subdirectory or sorts a CSV file encountered in one of said directories. After each CSV is taken into the program and sorted, the program compiles all of the records into one large CSV and makes sure that it is also sorted.

We attempted to use a globally declared linked list to store TIDs of created threads, that way we could join them all later, but ultimately could not get this successfully implemented. We used a mutex lock to make sure the structure was not being accessed by multiple threads at once, but we continuously encountered segmentation faults that stumped us. Even after solving one error, more would arise in its place.

If our program did work as intended, we would have expected the threads to be slightly slower than the processes version of the sorter. Since threads share resources we expected this to result in slower speeds but less space consumption. On the other hand, processes require more space but can offer speed as an advantage.

We did run the time utility to test how each sorter handles 1 CSV and got the following results:

Multiprocess sorter:

real 0m0.025s

user 0m0.000s

sys 0m0.001s

Multithread sorter:

real 0m1.330s

user 0m0.411s

sys 0m0.092s

# How to use the program

Compile using: *gcc -g -pthread mergesort.c sorter.c -o sorter*

Then, run with: *./sorter -c columnName -d /folder/folder/folder -o /folder/folder/folder*