CS214 Project 2

A Multi-threaded Sorter

Brian Huang & Matthew Stone

It only runs 100% when also running it with valgrind

```
valgrind ./sorter -c movie_title
```

Difficulties

We found that threads are very confusing to manage and debug. We also found that mutex is a difficult to use. We have something wrong with our code which makes it work half of the time. In GDB it says CRC mismatch. **It only runs 100% when also running it with valgrind. Because the program occasionally would stop prematurely without an error, we couldn't use GDB to debug what was wrong with our code.**

Design

We designed our sorter to create a new thread for every file and directory that was encountered. The thread for the directory would recursively call itself if there were nested directories. In our header file we added prototypes for the new thread functions and parameters for this function, we also added global arrays, variables, and mutexes. We used a global linked list to store our csv data. After all the threads were finish sorting individually, the threads would add its contents to the linked list. We merged the sorted linked list, after each individual file sorted itself, using multithreading. We would merge every two nodes in the linked list, wait for them to merge, and then repeat until there was only one node. We also split up our sorting and directory navigating code into two functions to be able to use call pthread using the functions.

```
gcc -lpthread -o sorter sorter_thread.c

./sorter -c movie_title -d inputdir -o outputdir
```

Assumptions

We assumed that this project wouldn't be that difficult but it turned out to be the most difficult. We ended up spending too much time on the previous projects trying to refine it for the current, which ended up creating more problems. We assume our multithreaded sorter will be faster than our multi-process sorter due to threads being lighter and the fact that the multiprocess sorter did not have to merge all of the sorted files into one large file.

Testing Procedure

We tested various times using different numbers of files and different numbers of directories. We created the different amount of files using a python script.