Alon Flor(af656)

Wenjie Wu (ww258)

Ztian Qin (zq42)

Seong Kon Kim (sk1299)

**IMPORTANT: NO MORE THAN 150 ROWS PER FILE. Every time the client connect to the server, it is going to print ip address.**

Client:

The client will go through the directory to find .cs files. Once the thread found a file, it will create a new thread and send the file to the server. However, the file will be convert to a char \*. In the string, the first one will be the column number of the header which is going to sort by. Then we add a “,” after the number to make sure when the server read it, it will know where the number ends, since the number could be more than 9. If the number is 10, then the first and second number will combine to the number we want. After the column number, there will be a number represent to the number of char for the first row including the “,”. This number is used to tell the server, how many char will be in the first row, and when it reconstruct the .csv file, it will know where to add a new line. Then, after the char numeber, add a comma, and then we just copy the whole first row which is already get rid of the new line. After this, we add the number of char of second row… Just repeat the same thing until we done all the rows.

When the client connect to the server, we will change the host name in command line to ip address, which is necessary for connecting to the server. And it will wait to read the server message said done the sort. Until all the threads done, and return to the parent process, the client will send dump request with the message “\*\*\*”, to let server to sort all the files together. Once it get the all-sorted file from server, it will do the same thing did in server that reconstruct the char\* to .csv file and then output the file. If there are multiple clients, it will just receive the all-sorted file.

Server:

The server will wait for the client connect, and once it connected, it will receive the char\* which is actually the .csv file in a different format. The server will reconstruct the .csv file. With the help of the numbers we added into the char\*, we did this in client part. We get the first number, which is for mergesort to know which column to sort. And read the second number, which will be the number of char in the first row ( including the comma). From this number, we know how many of the following char is going to be in the first row. Then we add the new line after the last char. And to do the same thing until finish the char\*. After reconstructed the .csv file, we do the mergesort to do the task from client. And when we receive the dump request, just make all the sorted files together ( the sorted files are stored in the master list) and do the mergesort again. After made the all-sorted file, the server will send the all-sorted file back to client in form of char\* with the “+++” at the end. But this will be ignored when client reconstruct the .csv file. When we put all the files together, we will just put the header once, which is the first row.

And also, the server will output the clients' addresses. Every time the client connect to the server, it is going to print ip address.

If there are multiple clients connected to the server, the server will use multithreads as well. Once there is the client come in, the server will create a new thread to deal with the client.