Shilp Shah (RUID: 170009205)
Mathew Varghese (RUID 170009651)

Assignment 2 Documentation

Approach

- Our approach for this assignment changed significantly from when we started compared to when the assignment was actually due. Our initial approach was to just replaced the code from the previous assignment but instead of creating new processes, we create new threads. This approach led us into many problems as threads act very different from processes. So we had to change our approach to make the code thread-safe and handle the new requirements of the project. In the end, we decided to restructure our code to implement arrays instead of linked lists, and changed our struct object as well. We started by reading in the CSV files and making struct objects for the data. Once this was done, we were able to use some of our code from the previous assignment to determine CSV files and sub-directories. New threads were created and we made sure to use mutexes to lock and unlock thread sensitive data. Then we just merged the files into one large file with the proper output name.

Starting Out

- For this assignment, we recognized that it looked very similar to the previous assignment but instead of creating a new process, we create new threads for files and sub-directories. So we started by changing our old code to and replaced the code for creating new processes with creating new threads. But this posed a lot of problems as we realized later that threads act very differently from processes. We realized our current approach would not work with multithreading so we had to change our approach.

Adjusted Code Structure

- Our new approach involved us having to change some of our basic code from the previous assignment. Instead of using linked lists, we started using arrays because they are a lot easier to handle in sorting functions. We also changed our struct to have a field variable for each of the columns (see code below):

```
typedef struct record
{
	char *color;
	char *director;
	char *numCritic;
	char *duration;
//continue for each column name
```

- The code to determining a CSV file and subdirectories was similar from the previous assignments. Once all the data from the CSV files was put into structs, we put that into a global array. This involved using mutex lock and unlock because multiple threads would be writing to the same array (see code below):

```
void addgarr(struct record curr[],int hi){
	pthread_mutex_lock(&mutex);
	//Manipulate variables and add to the global array
	pthread_mutex_unlock(&mutex);
}
```

- Then we merge the files by taking the first 2 elements of the array and calling the mergesort function again then storing the result back into the array. Initially we had this implemented as a stack but decided to stick with arrays because they are easier to understand in code. But our implementation essentially acts as a stack by popping the first 2 elements, sorting them and then pushing one sorted element back on.

Issues

- We ran into a lot of issues with threads in our first approach. We realized that a lot of our functions from the previous assignment were not thread safe and we could not use them. This is why we restructure our code to better avoid these problems. Even with the new approach, we ran into some issues along the way. One of the issues we faced was having

inconsistent results each time we ran our code. This was really weird because we could not predict the result of our program, sometimes it would give Segmentation Fault while other times it would work. This issue was resolved by realizing we were not properly using mutex lock and unlock on data that multiple threads are writing to. Also, we had trouble merging the files into one sorted file. We got it working so that each file is sorted but if there are multiple files, we had trouble sorting the total data in the output file. Another issue we ran into was dealing with a subset data from the masterset of data given. We were not able to handle having less than the full 28 columns because of the way we were storing the data into the struct in our new approach. We resolved this issue last because we felt it was more important to have proper working code before dealing with other edge cases, such as having a subset of data.

Test Cases

- Our first test case was simply the full metadata of movies given. We wanted to make sure the thread was created and the data was sorted properly on a single file first before working with multiple files.
- Next, we tested having multiple CSV files in a single directory to see if they were merged properly into one file. The result was a file that is twice the size with duplicated to each row next to each other (see sample below)

The first few rows of the input files:

color	director_name	num_critic_	duration	director	actor_3_fa	actor_2_n	actor_1_fa	gross	genres	actor_1_n	movie_tit	num_vote	cast_total	actor_3_n	facenumb	plot_keyv	movie_im	num_user	langua
Color	James Cameron	723	178	0	855	Joel Davic	1000	7.61E+08	Action|Ac	CCH Poun	AvatarÂ	886204	4834	Wes Studi	0	avatar|fut	http://ww	3054	English
Color	Gore Verbinski	302	169	563	1000	Orlando B	40000	3.09E+08	Action|Ac	Johnny De	Pirates of	471220	48350	Jack Daver	0	goddess|r	http://ww	1238	English
Color	Sam Mendes	602	148	0	161	Rory Kinn	11000	2E+08	Action|Ac	Christoph	SpectreÂ	275868	11700	Stephanie	1	bomb|esp	http://ww	994	English
Color	Christopher Nolan	813	164	22000	23000	Christian I	27000	4.48E+08	Action|Th	Tom Hardy	The Dark K	1144337	106759	Joseph Go	0	deception	http://ww	2701	English
	Doug Walker			131		Rob Walke	131		Documen	Doug Wall	Star Wars:	8	143		0		http://www.imdb.com/titl		

First few rows of the output file:

Color	A. Raven (	3	97	0	94	Vanilla Ice	639		Action|Ac	Scott Levy	The Helix.	534	1188	Jennifer S	2	cnn report	http://ww	9	English	USA
Color	A. Raven (	3	97	0	94	Vanilla Ice	639		Action|Ac	Scott Levy	The Helix.	534	1188	Jennifer S	2	cnn report	http://ww	9	English	USA
Color	Aaron Har	29	87	0	94	Michael M	160		Drama|Hc	Jordi Vilas	CircleÂ	13279	776	Kaiwi Lym	0	alien abdu	http://ww	59	English	USA
Color	Aaron Har	29	87	0	94	Michael M	160		Drama|Hc	Jordi Vilas	CircleÂ	13279	776	Kaiwi Lym	0	alien abdu	http://ww	59	English	USA
Color	Aaron Sch	160	100	11	970	Robert Du	13000	9176553	Drama|Mu	Bill Murra	Get LowÂ	19147	19330	Bill Cobbs	1	funeral|fu	http://ww	97	English	USA
Color	Aaron Sch	160	100	11	970	Robert Du	13000	9176553	Drama|Mu	Bill Murra	Get LowÂ	19147	19330	Bill Cobbs	1	funeral|fu	http://ww	97	English	USA
Color	Aaron Selt	99	85	64	729	Carmen El	3000	48546578	Comedy|F	Alyson Ha	Date Movi	50415	6539	Fred Willa	0	female pr	http://ww	613	English	USA
Color	Aaron Selt	99	85	64	729	Carmen El	3000	48546578	Comedy|F	Alyson Ha	Date Movi	50415	6539	Fred Willa	0	female pr	http://ww	613	English	USA

This was sorted on director_name and I skipped the null values. But in the output file, it is clear to see that there are 2 rows of each entry. This is the expected result of sorting 2 identical files.

- The next case involved not having the same number of headers between CSV files. One CSV file had only 4 columns while another file had all 28 columns. The expected result is that the files are sorted based on what the user entered and the columns that are not in the CSV file are filled with NULL values. When testing our code, we got the correct output.