Tonya Shulkey (018145619)

Evan Marquez (018131930)

CECS 326 Sec 04

March 12, 2021

Design: For the design of the code I used the skeleton code given in the Sample. I created the two global arrays. I used the two functions sorter and merger. In the main function I created the first thread and assigned the empty array with the same size as the array with the unsorted list to it. The thread is sent to the merger and there I created two more arrays to split the list in half and sort them using two more threads. To do this in a way that would keep the size of the array in mind and sort it correctly I added an if statement for whether the list is even or odd. If it was odd it would subtract 3 from the index of the list. If it was even it would subtract 2 as it inputted the values. (It may make more sense to see in the actual code). You can see in Figure 1 that both even and odd arrays will be sorted correctly. With that it would decide how to split the right side up. It puts the numbers in the two arrays and assigns it to the two threads when they are created. It is then sent to the sorter which I used bubble sort to sort the array. I chose bubble sort because the directions says to use any sorting algorithm and that is the one I’m most comfortable with. When it returns the left and right sides should be sorted and all that’s left is for the last thread to finish its job. It merges the two sides, using half of the merge sort algorithm, together and prints it. I printed the unsorted array along with the sorted array just so you can see if it worked correctly and that it wasn’t already sorted. This project used mostly what we knew about creating and joining threads to finish. You can see the design in the source code called project2.c.

In this project three threads worked together to finish a task. The threads were also able to use passing parameters to send the arrays into the sorter to complete its side of sorting. It shows that the first thread created would wait until the last two threads created finished their job before it would finish and terminate. The hardest part was making sure the list would sort whether it was even or odd. That’s why I had to make an if statement to take care of the cases. I also had to make sure the sorter method would be able to tell the size of the array so I had to input that at the beginning of the list as well so that when the thread passed it through the parameter there would be no problems.

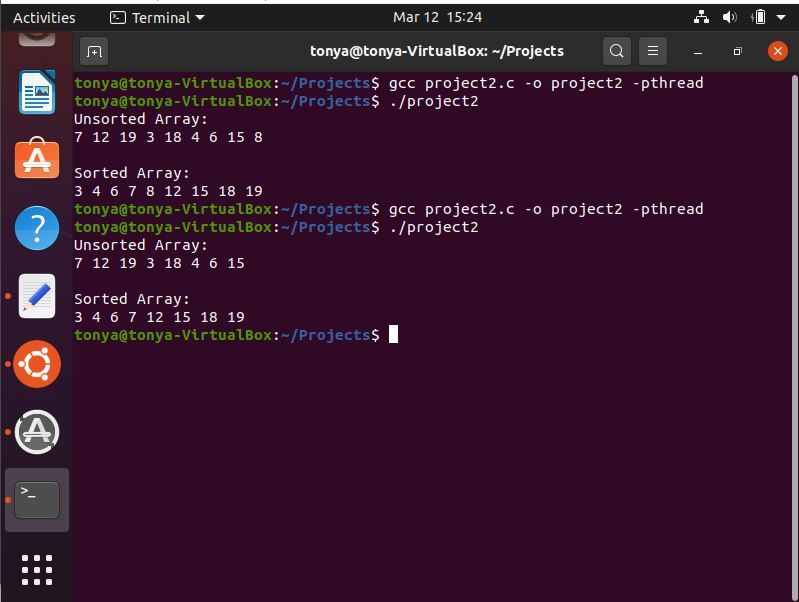


Figure 1

Individual contributions:

Tonya Shulkey: Worked on the code came up with the ideas and implemented it. Created the README file and the Makefile. Made a video of the code running. Wrote the Report.

Evan Marquez: Worked on the code came up with the ideas and implemented it.

We both worked on the code together while on a call.