Cesar Cantu-Perez CSE 3320 – 001

ID#1001463763

**Assignment #2 Lab Report**

**Purpose:**

The purpose of the experiment is to demonstrate the effectiveness of using multiple threads to complete a process vs. using only a single thread.

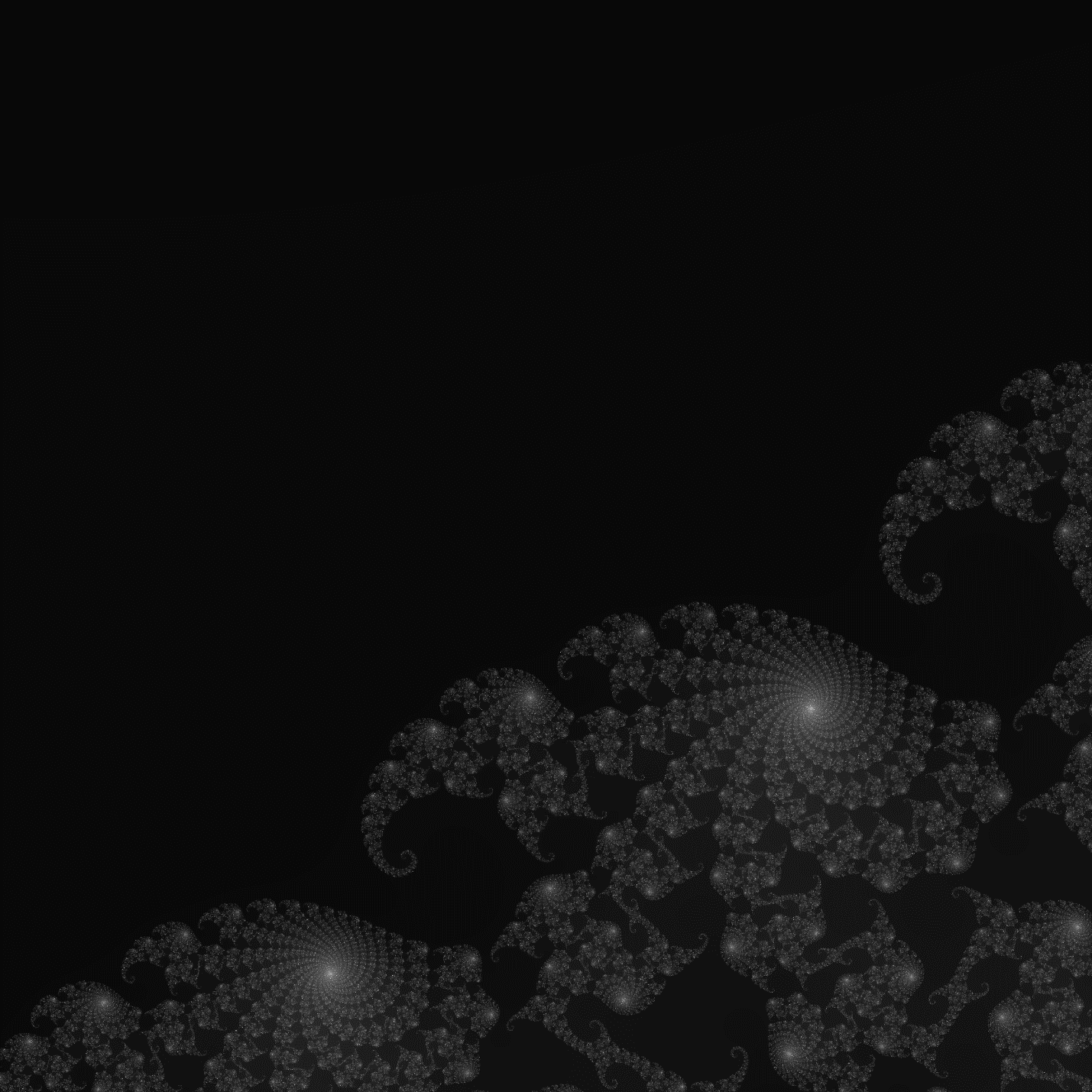
**Modification of mandel.c:**

To fulfill the purpose mentioned above, I modified the given mandel.c to allow for multiple threads to be executed rather than just one. I also added a way to show the time it took to create the image. What mandel.c does is create a Mandelbrot image that can vary in shape and size depending on the variables you give it. What I added was a new parameter “n” that when called alongside the program gives the number of threads being used to create the image. To test if my program worked I used the following commands:

After using the “make” command in the command line, I typed:

./mandel -x 0.286925 -y 0.014295 -s .000015 -m 1500 -W 2000 -H 2000 -n 3 -o mandelmulti.bmp

In this example that I used, n = 3 which means that the number of threads used to create the image saved as “mandelmulti.bmp” is 3.

The image that is produced is this: 

**Lab Process:**

In this lab I will be using my modified mandel.c to test two different images, A and B, by having the program create them a number of times to measure the fastest time it took in microseconds(μs). The images for A and B are as follows:

A: ./mandel -x -.5 -y .5 -s 1 -m 2000

B: ./mandel -x 0.2869325 -y 0.0142905 -s .000001 -W 1024 -H 1024 -m 1000

The images look like this:

Image A -A picture containing text, silhouette

Description automatically generated

Circles with other circles coming out of them moved a little off center.

Image B - A picture containing light

Description automatically generated

A lightly shaded spiral

I took images A and B and created them 10 times for each configuration of threads to find the fastest time in microseconds(μs) it took to create them.

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
|  | **1**  **thread** | **2 threads** | **3 threads** | **4 threads** | **5 threads** | **10 threads** | **50 threads** |
| **Image A** | 1460408 μs | 1321738 μs | 937660 μs | 670382 μs | 636938 μs | 352684 μs | 166176 μs |
| **Image B** | 3599063 μs | 1902709 μs | 1404917 μs | 1034403 μs | 868530 μs | 451336 μs | 357718 μs |

**Results/Conclusion:**

The shape of the two images are very different from one another. One looks like one single concrete shape and the other looks like a spiral that seems to go on forever. They look so different because image B is a very zoomed in version of image A in a very specific spot. So in reality both images go on forever. The graphs show that if you have more threads, the program runs faster. So then the optimal thread in this experiment for both of the images is 50 threads.

The results in this experiment have shown that the more threads you have running the program, the better the performance. This lab has shown me the performance difference between single thread and multi thread programs.