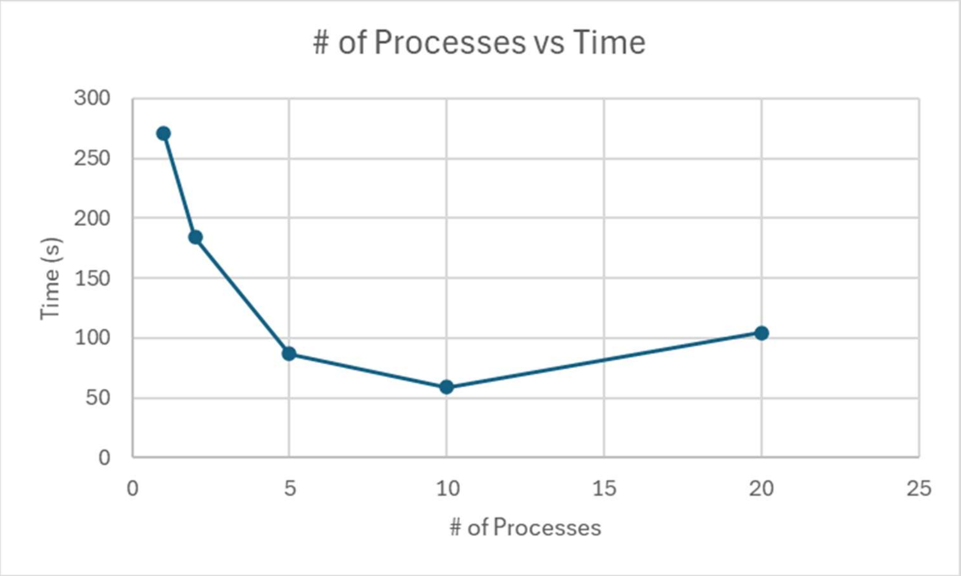
Jacob Tyree
Lab11

Brief Overview of Implementation:

The modifications made to mandel.c allowed the program to use multiple processes when processing and creating the images. For most increases in processes, this allowed the program to complete in less time. Some modifications include adding an integer variable that tracks the number of processes the user wants to use. This is inputted through the command line. By default, it's set to 1, but the user can change it to however many processes they would like. Additionally, they can request a specific number of images that they want to generate. Moving down to the forking, the total number of forks is dependent on the number of processes the user requests. The program then goes through the typical creation of images. However, the difference with each image is that the scale of the image and the color is changed depending on the index of the for loop.

Below are the results of the runtime with 1,2,5,10, and 20 children processes.



Discussion:

Looking at the graph, it is shown that the time it takes to run the program decreases as the number of processes increase. However, once it is tested with 20 processes, it increases. This results in the statement that increasing the number of processes does not always speed up the operation. I assume that this is because of the computer that I am running the program on, as it can only handle so many processes before it maxes out and eventually gets slower trying to handle all the processes that are running through the code.