

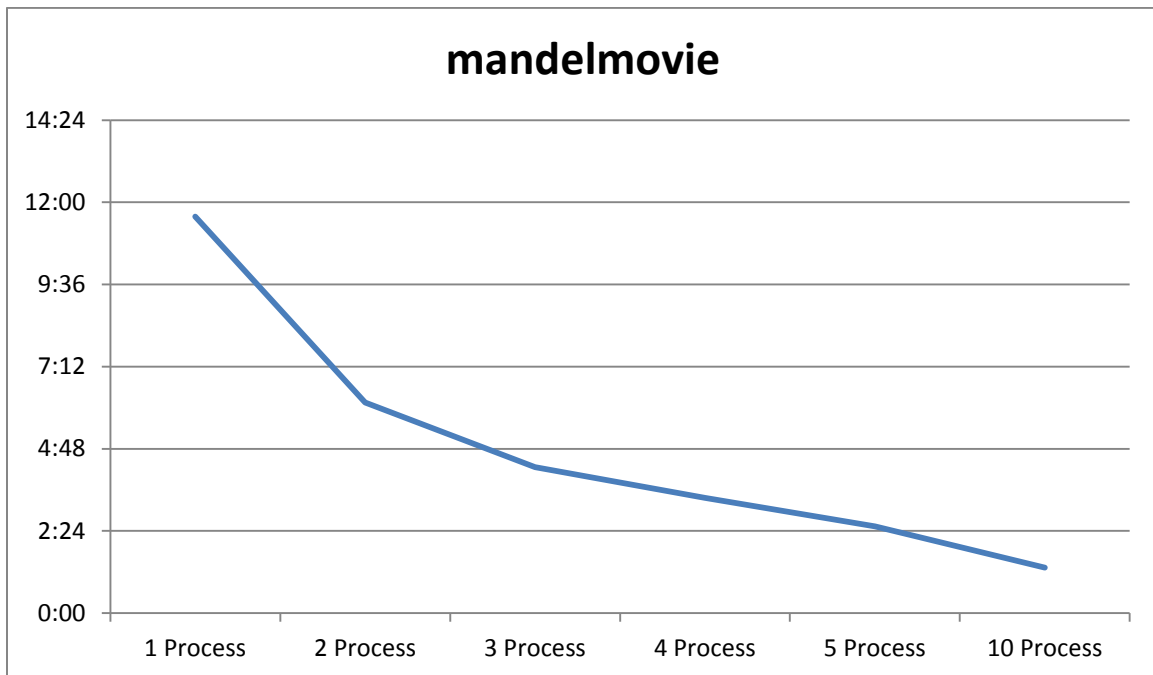
## What was the purpose of this experiment and the experimental setup?

The purpose of this project was to teach yourself how to use operating system calls to increase the efficiency of a task, specifically by running its computations in parallel via multiple processes and then again into multiple threads. Mandelbrot images were used because they are fairly interesting and require more intensive computations than students are typically expected to be performing at their level. The project was performed on University of Notre Dame machines (specifically [student00.cse.nd.edu](mailto:student00.cse.nd.edu)) and command line arguments were used as follows:

(Final frame rendered for mandelmovie)

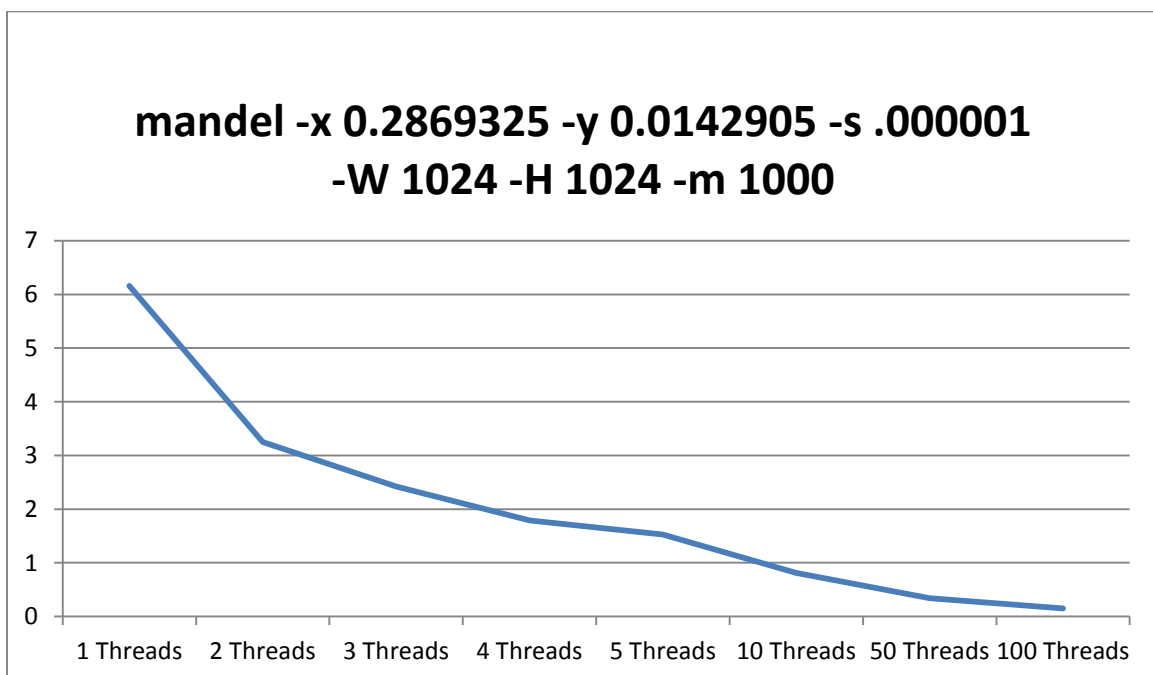
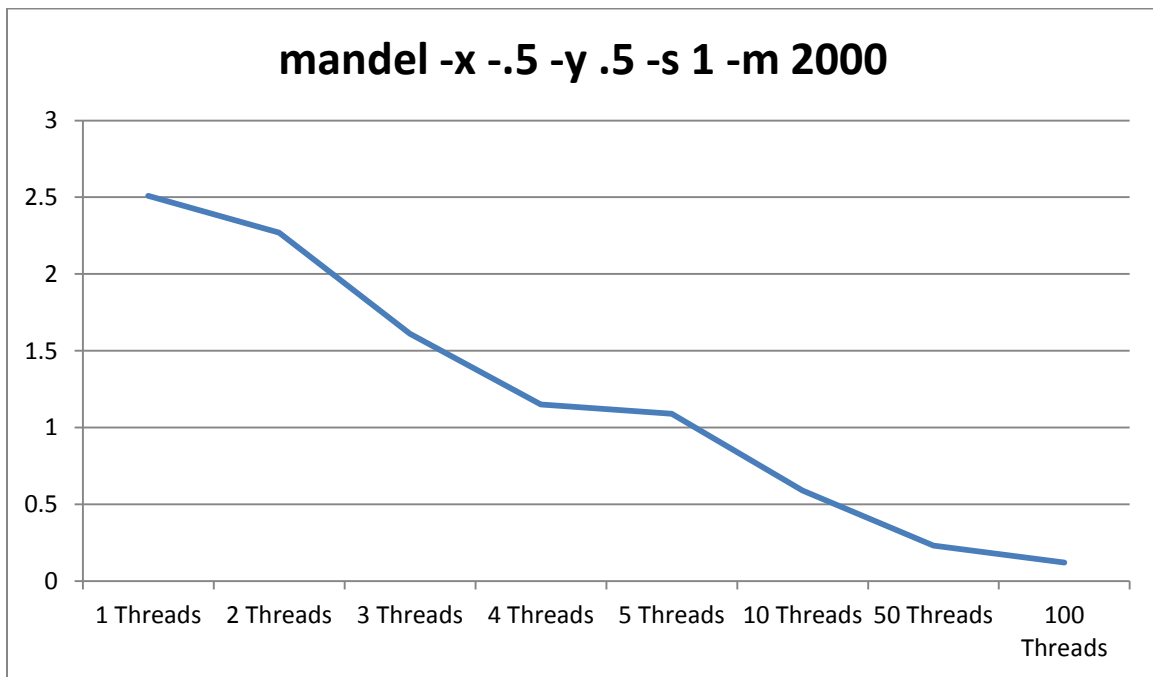
```
./mandelmovie -x 0.38941 -y -0.265 -s 0.000001 -m 2000 -W 1680 -H 1050
```

Graph of **mandelmovie** execution time for each of 1, 2, 3, 4, 5, and 10 processes running simultaneously.



As the number of parallel processes increases, the program's execution time decreases because there is higher throughput. The optimal number of parallel processes is around 2-3; any more than that and any additional speedup begins to taper off. It is not possible to have too many parallel processes, but adding more processes can only be achieved by using more and more hardware, which can get expensive.

Graph of **mandelmovie** execution time for two different configurations at 1, 2, 3, 4, 5, 10, 50, and 100 threads respectively.



Configuration A has an optimal of around **4 threads** while Configuration B has an optimal around **5 threads**. This is likely because Configuration B has much more complex parameters in its computation - such as image dimensions (1024x1024) – that it can benefit more from parallelization than Configuration A before tapering off.