Design Requirements

In this lab, you will write a Mandelbrot fractal generator. Your system must initially display the fractal between -2.5 – 1i and 1 + 1i. From this point, your system must identify an interesting point at which to zoom, using a 4:3 aspect zoom window. To identify this point, use the pixel value which was determined to not be in the set after greater than 90% of the total iterations.

Your system will incrementally zoom into this point at an exponential rate, at:

1.5^{zoom\_level}.

At each zoom level, re-render the frame. Use floating point values when generating the fractal.

Begin with one processor, then scale up to four processors, one at a time. Parallelize the algorithm using the data parallel method (i.e. assign an equal number of pixels to each processor). Don’t forget to use barriers to synchronize the processors on each frame.

Measure the time required to generate each frame. After the program finishes, calculate the average time per frame. Do this for each system configuration.

Project Submission

Each group must submit all files containing their originally-written (non-generated) C source code to Dropbox.

In addition, each group must submit a report that details their performance results for each multiprocessor configuration, measured as:

1. average time to generate each frame over at least 10 frames,
2. average number of cycles to compute and paint each pixel over at least one frame,
3. number of cycles to compute one iteration of the polynomial evaluation loop, and
4. number of cycles per floating point operation performed in the polynomial evaluation loop

The report should contain all implementation-specific details, such as:

- the maximum iteration count,
- the method for finding the zoom point,
- the color calculation function (make sure there is no overflow)