

CS 222

Project 2 - Makefile and Recursion

This project is to practice multiple components compilation, recursion, and structure. It is an extension to HW6.

The Mandelbrot Set

The Mandelbrot set is a set of all **complex numbers** that satisfy the following condition: for any c in the Mandelbrot set, the absolute value of $\text{mandelbrot}(n)$ remains bounded when n tends to infinity, where

$$\begin{aligned}\text{mandelbrot}(n) &= \text{mandelbrot}(n-1) * \text{mandelbrot}(n-1) + c; \\ \text{mandelbrot}(0) &= c.\end{aligned}$$

Here the addition and multiplication are complex number addition and multiplication.

Specifically, for our homework, if the absolute value of $\text{mandelbrot}(15)$, $|\text{mandelbrot}(15)|$, is smaller than 10000 for a given c , then c is in the Mandelbrot set. If we consider a c as a 2D coordinates on a plane, the points in the area from $(-2.0, -1.12)$ to $(0.47, 1.12)$ will contain the Mandelbrot set. Our goal is to calculate an array of points that covers the Mandelbrot set in this area and display it. You may save the area in a file if you want the area to contain many points beyond your display.

Implementation

First, if you haven't completed HW6, you will need to finish it without any problem before starting this project. We need to use the complex functions in this project.

Second, let's extend HW6 to include `mandelbrot.c`, which implement the recursive function `mandelbrot(n)`. In general, you should avoid using global variables, so the recursive function has to carry another input value, c , and `mandelbrot(n)` checks whether c belongs to the Mandelbrot set. In other words, your function should be `mandelbrot(n, c)` instead. Optionally, for the purpose of learning, you may define a global variable c in `mandelbrot.c`, and use "extern" in `mandelbrot.h` to make it visible in `main.c`, so you can just use `mandelbrot(n)`.

Also, `mandelbrot(n)` uses complex number addition and multiplication, so you should include `complex.h` in `mandelbrot.c` to allow `mandelbrot(n)` to access complex functions. In the recursive function, $|\text{mandelbrot}(n)|$ can be quite large beyond the computer's representation. Therefore, in the recursion, if $|\text{mandelbrot}(n-1)|$ is bigger than 10000, your recursion should consider that it is not bounded, so you return `mandelbrot(n) = (10000, 10000)` instead of the calculated number to indicate that current c is not in the Mandelbrot set. Also, in the recursion, `mandelbrot(n-1)` is calculated twice, which is really wasting effort, so you should just calculate once for efficiency.

Third, in order for `main.c` to see the recursive function and c , you need to include `mandelbrot.h`. Because you use complex number in `main.c`, you need to include `complex.h` as well.

In `main.c`, for all point c in the area from $(-2.0, -1.12)$ to $(0.47, 1.12)$, we can check the corresponding `mandelbrot(15)` to see if it is in the Mandelbrot set. If it is in the set, which means $|\text{mandelbrot}(15)|$ is smaller than 10000, we print a "#"; otherwise, we print a "-" or an empty space. Therefore, we can actually print an image of the Mandelbrot set. For example, you can check 40×30 points with a double for loop. That is, we start in x direction from -2.0 with step size $(0.47 - (-2.0)) / 40 = 0.06175$ to 0.47, and in y direction from -1.12 with step size $(1.12 - (-1.12)) / 30 = 0.077$ to 1.12. You may notice that this arrangement will be an up-side-down image, but the image is symmetric, so we can ignore this problem.

If you want, which is not required for this project, you may save the image in a two dimensional array of

characters first and then print out the image. This way, you can actually flip the image. Better yet, you can save this image in a file, so you can check out a much larger image in the file with many more points.

The final executable is called mandelbrot. Make sure your Makefile will compile and generate the three object files, and link them together with correct dependencies.

Testing and Submitting

Test your program on Mason to make sure it compiles with “make” and runs properly. While on Mason, create a script file listing your programs to the screen in the following orders: Makefile, mandelbrot.h, mandelbrot.c, and main.c. Demonstrate a sample run with the Mandelbrot image. Submit all files (except object and executable files) to Blackboard.