Parallel
Homework 6

Christian Winwood
A02074021

March 01 2021

1 Implementation

My implementation is fairly simple. I have a root process that sends and receives all the tasks and then the other processes do all the work. The root first prints the header and then process loops through and grabs all the coordinates. It then systematically loops through the processes and sends one coordinate to each process. Once each process has a task, the root process receives the correct color from each of them and prints it to the file. When everything is done, the root sends out a kill command and every process exits. The individual processes get a coordinate from the root, calculate the mandlebrot, and using the number of iterations calculates a color. It then sends the color to the root process and waits for another coordinate.

2 Compile and Run

The compile and run commands are the same they have always been. First you use

mpic++ mandlebrot.cpp

in order to compile the program. This creates an a.out files that you need to run. To run it you use the command

mpirun –oversubscribe -np ¡number-of-processes¿ a.out

and the program will run.

3 Desired Output

There should be no out put to the console and instead a .ppm file will be generate and placed in the root directory.