Benjamin Berger

Project 4

Goal:

Performing matrix multiplication where each individual multiply is handled by a new fork.

Outline:

Input a file containing 2 matrices that will be multiplied. There will be N number of multiplications so for each one of them, call a different fork to do it.

Solution:

First thing I did was set up the initial matrix. I used a global matrix because I would not be altering them. I had the infrastructure from Project 3 so I used the same code for that. I then figured out how big I was going to need my result matrix and allocated space for that.

Then I allocate a shared array that will be able to hold all of the calculations.

I then calculated the number of multiplications I would need and ran a for loop for this many times.

Inside the for loop I called fork();

In the child it gets the iteration of the for loop and makes the calculation at that position and throws it into the shared array.

For the parent process it waits for the child to be done and then access the shared array and adds those values to the resulting matrix.

Once all the forks have been completed, the program prints out and exits.

Flow:

It is hard to draw a diagram for the flow of this project because it is impossible to tell what is being executed first, and that’s fine.

In essence though, the child had to finish before the parent could run the join.