Parallel Programming Basics

A screenshot of a computer

Description automatically generated

In this tutorial program, I learned that the iterations are in out of order every time the code is executed. The iterations also correspond to the thread number. There are normally 4 threads in the Raspberry PI, each thread iterates 4 times each, as shown above. This is referred to as the data decomposition pattern because the PI is spreading/decomposing the amount of work that needs to be done across multiple threads.

A screenshot of a cell phone

Description automatically generated

As this program runs every time it loops, the sequential and parallel sum never change. I tried to change the number of threads, even then nothing changed. This is because each of the accumulator’s needs to be private to the thread of its own, so that it can produce the correct answer. This type of variable sum is relying on what all other threads are doing.

A screenshot of a computer

Description automatically generated

Here we can see that the parallel sum has changed because we have uncommented the two statements in line 39 of the program. Which are “// #pragma omp parallel for // reduction(+:sum)”. This time the parallel sum changed because OpenMP is being implemented.