CS-214 Systems Programming
Professor Francisco

Group Members:

Richard Travers	NetID: rdt58
Syed Abbas Haider	NetID: sah300

Background:

For this project we are looking for a number, a target in an array. Array will be divided into chunks according to the size of the thread/Process.

We change the size of the array and the threads/Processes for every test case. For the size of the threads/Processes the max limit is 250 so we did not set it more than 250. By dividing the size of array with the upper bound we can always figure how many processes and threads we need.

$$\frac{Size\ of\ Array}{Upper\ bound} = \text{number of Thread/Process needed.}$$

So, In total we ran 4 test cases. We change size of array and the size of the thread/Processes every time. We run 150 iterations of each test case in order to get the most accurate data, and then ran the program twice in order to solidify the data.

TestA:

TestA takes an array size of 20,000 and sets the thread/Process size to 250.

$$\frac{20{,}000}{250} = 80 \text{ threads/processes (workers)}$$

We create 80 threads/Processes and have each one search for our target number and return a status of nonzero for threads and zero for Processes if the target is found.

TestB:

TestB takes array size of 5000 and uses the previous thread/Process size.

$$\frac{5000}{250} = 20 \text{ threads/processes (workers)}$$

We create 20 threads/Processes and search for the target.

TestC:

TestC takes array size of 20,000 and sets the thread/Process size to 100. This is to compare the runtime to Test A and Test E.

$$\frac{20000}{100} = 200 \text{ threads/processes (workers)}$$

200 workers are created to search the array.

TestD:

TestD takes array size of 5000 and sets the thread/Process size to 100. This is to compare the runtime to Test B and Test F.

$$\frac{5000}{100} = 50 \text{ threads/processes (workers)}$$

50 workers are created to search the array.

TestE:

TestE takes array size of 20,000 and sets the thread/Process size to 50. This is not only a stress test on the system, but also provides useful data to compare to the previous 20000 sized array tests.

$$\frac{20000}{50} = 400 \text{ threads/processes (workers)}$$

The largest amount of workers in the program are created here.

TestF:

TestF takes array size of 5000 and sets the thread/Process size to 50.

$$\frac{5000}{50} = 100 \text{ threads/processes (workers)}$$

50 workers are created to search the array.