

# *REPORT OF PROC & TH STAT*

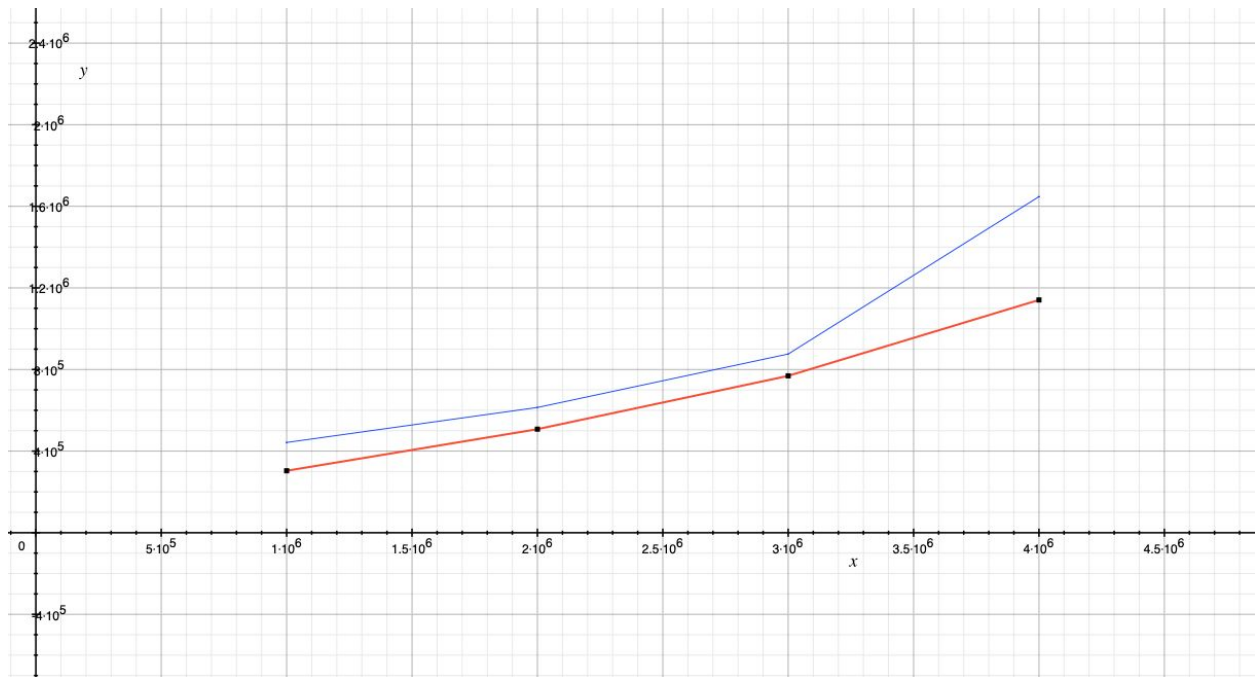
## *Procstat:*

In the first program, proc-stat.c we use the concept of multi-process programming and write a program that calculates various statistical values for a list of numbers. This program will be passed on to a sequence of numbers in a file which will be passed through an argument. We create three worker processes to compute the following statistics mean median and standard deviation. After computing these values the corresponding processes write to a shared memory globally. The mmap function establishes a memory-mapped file containing the shared-memory object. It also returns a pointer to the memory-mapped file that is used for accessing the shared memory object. Finally the parent process prints the various statistical data into a separate file once the workers have exited.

## *Thstat:*

In the second program, th-stat.c we use the concept of multi-threaded programming and write a program that calculates various statistical values for a list of numbers. This program will be passed on to a sequence of numbers in a file which will be passed through an argument. The second program demonstrates the basic pthreads API for constructing a multithread program that calculates various statistical values through various threads. In this program separate threads begin execution in a specified function. When the program begins a single thread of control begins in main(). After some initialization main() creates a second thread

that begins control in the runner function and so on. All the threads share the same global data.



The blue graph is the one of processes and red one is of threads. As we can clearly see that multi processing takes more than multi threading. Since multiprocesses take time for resource allocations.