

# Operating Systems

## Report: Assignment 2

Harsh Raj  
MA17BTECH11003

---

### Statistics Formulae Used :

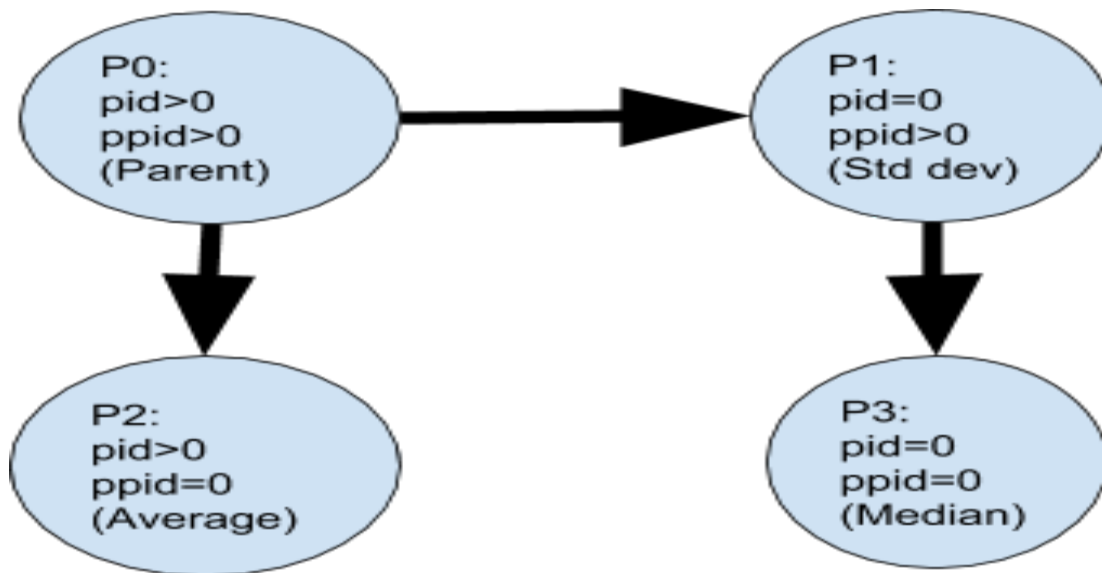
Mean :	Standard Deviation :
$\bar{X} = \frac{\sum X}{n}$	$SD = \sqrt{\frac{\sum  x - \bar{x} ^2}{n}}$
<b>Median :</b>	
<p>If the total number of numbers(n) is an odd number, then the formula is given below:</p> $Median = \left( \frac{n+1}{n} \right)^{th} term$ <p>If the total number of the numbers(n) is an even number, then the formula is given below:</p> $Median = \frac{\left( \frac{n}{2} \right)^{th} term + \left( \frac{n}{2} + 1 \right)^{th} term}{2}$	

---

## Design and Working : proc-stat.cpp

This is a multiprocess program that takes in a sequence of numbers and calculates the Average, Standard Deviation and Median values.

- *struct statistics* contains the variables *avg*, *std\_dev* and *median* which will have the final results. It is mapped to shared memory for all the processes so that all can access.
- 3 Worker Processes are forked from parent process, one each for finding Average, Std Deviation and Median.

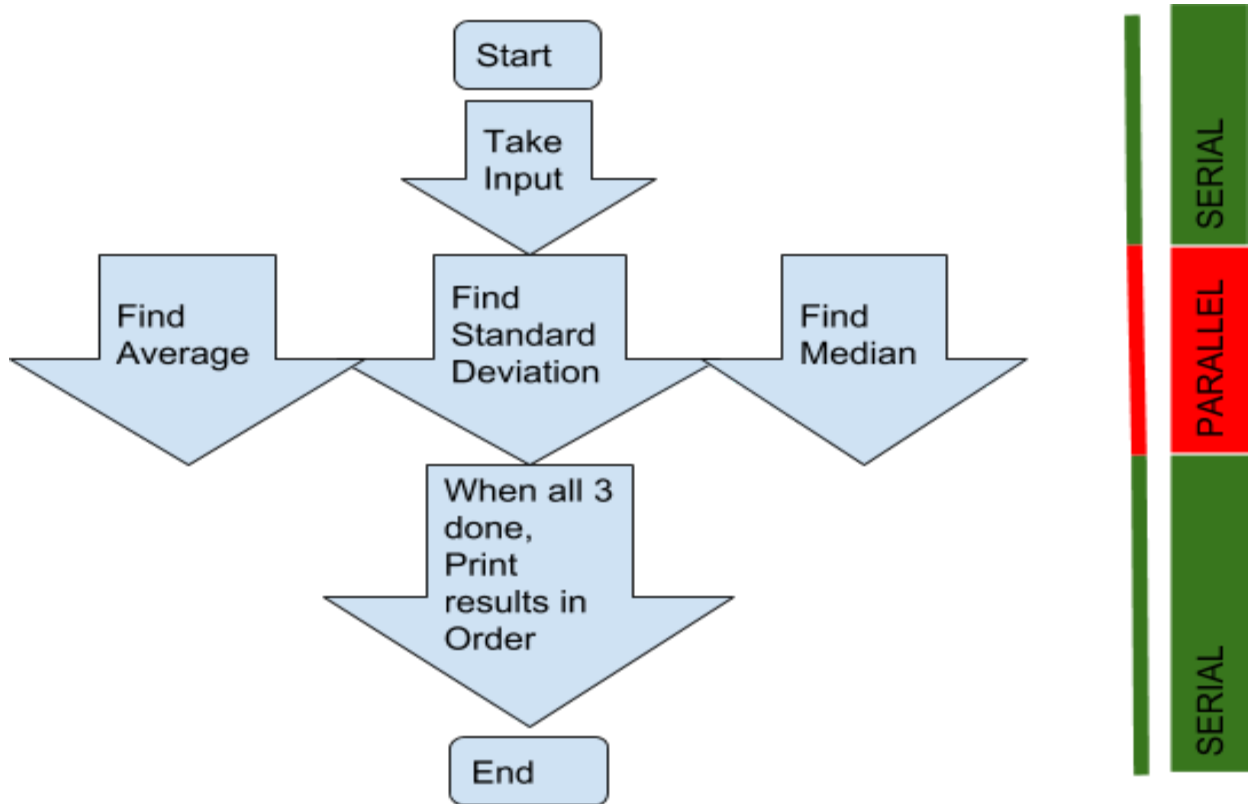


- Process P1 waits for P3 to terminate before exiting, and Process P0 waits for P1 and P2 to terminate before exiting. Thus, all processes are given the time to complete their respective work.
- Each process calculates the respective values, and writes it to the shared memory struct. Finally, when all processes are done executing, then main parent process writes the results.

## Design and Working : th-stat.cpp

This is a multithreaded program that calculates the statistical values of average, standard deviation and median for a given sequence of numbers.

- 
- The variables *avg*, *std\_dev*, and *median* are global variables for this program. They can be accessed by all threads.
  - After taking input, three independent threads with thread IDs tid1, tid2, tid3 are created. Each of these threads works on finding the average, standard deviation, and median of the given numbers, by independently and parallelly executing the *calculate\_avg()*, *calculate\_std\_dev()*, *calculate\_median()* functions respectively.
  - We wait for all three threads to complete their execution before printing result, so that final (correct) result is printed.

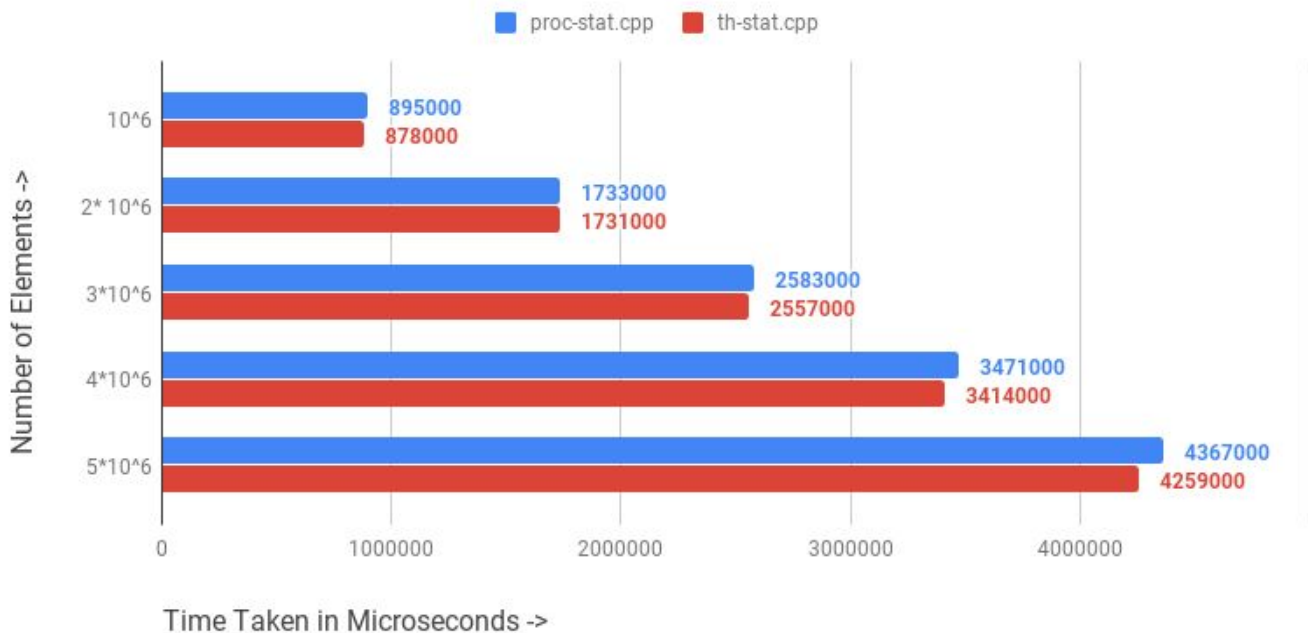


---

## Performance Comparison: Multi-Threads vs Multi-Processes

On comparing the performance of proc-stat.cpp with th-stat.cpp , we found the following results:

Running Time Comparison: proc-stat.cpp and th-stat.cpp



This is justified because of the following differences between processes and threads:

- Processes can be independent, while threads exist as subsets of a process
- Processes carry considerably more state information than threads (in the Process Control Block), whereas multiple threads within a process share process state as well as memory and resources
- Context switching between threads in the same process is typically faster than context switching between processes.