# OPERATING SYSTEMS 1

INSTRUCTOR: DR. SATHYA PERI
Hitesh Sehrawat - ma17btech11004@iith.ac.in

## 1.Design of the Program

**I have made 2 programs to determine mean, median and standard deviation of a data set.**

**The 1st program uses the process creation concept i.e creating multiprocess to accomplish provided task by dividing it into different tasks.So, the program creates 3 child processes to calculate the above mentioned stats,and the parent process is used to print them.**

**While the 2nd program uses the concept of creating ”pthreads” i.e. to create threads and assigning simpler tasks to each thread to achieve multithreading. So the program here creates 3 threads to calculate the statistics followed by their output.**

**Both the programs takes the input as ‘N’ (no. of data values) followed by ‘N’ data values to calculate statistics. Where input can be given on command line or by input text file.**

## 2. Working

### (1) Using multiprocessing

**This mechanism starts with creating a IPC using the POSIX shared memory. Creating a shared memory for parent and child processes using function ”shm_open()”, and allocating size to it using the function “ftruncate()”. This is followed by memory mapping using the function “mmap”.**

**Then the parent is forked using “fork()” function,creating a child process.Now in the child process where pid is 0, fork the child process, that will result in creating a child of child process that will be exactly 2 processes.Assign these 2 processes to calculate 2 of the 3 given stats which are to be calculated.**

**Now wait till the processes are done , once done,in the parent process “fork()” again to create a child process, thus and use this process to calculate the third stat.Wait for this process And thus use the parent function to Output the statistics.And at the end unlink the shared memory using shm_unlink().**

## (2) Using Multithreading

**This is done using “pthread” library , the function “pthread_t” assigns the thread ids.In given program i have made 3 threads.Each to calculate 3 stats.The pthread_create prepares the threads,and assigns the work to the thread id passed as its argument.**

```
pthread_create(pthread_t *thread, const pthread_attr_t *attr, void *(*start_routine) (void *), void *arg)
```

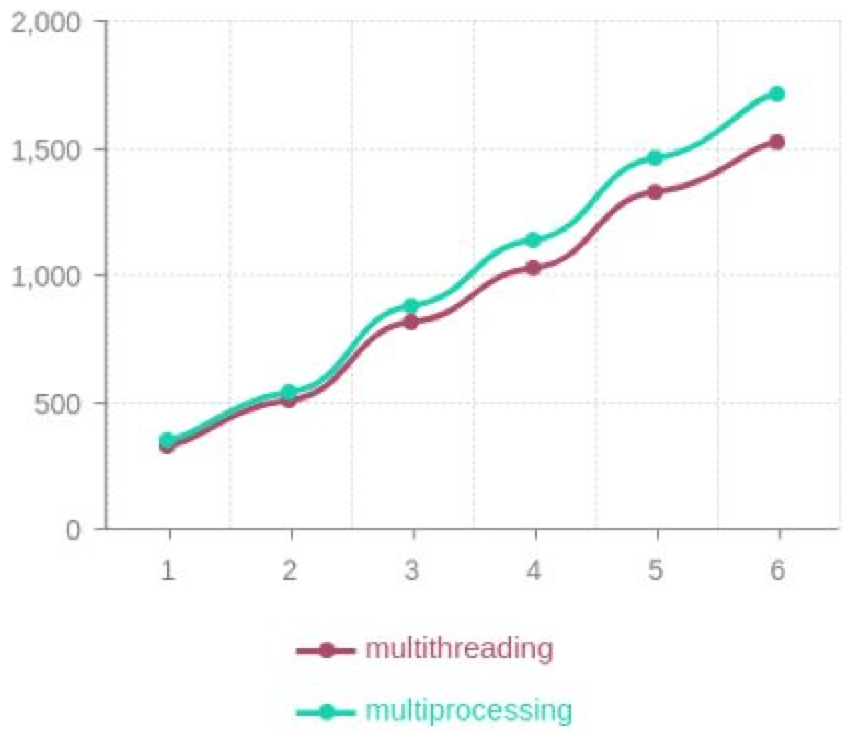
**The program is followed by pthread_join().The pthread_join() function waits for the thread specified by thread to terminate.  If that thread has already terminated, then  pthread_join() returns immediately.  The thread specified by thread must be joinable.**

**After the threads have calculated the mean,median and SD, the program output the respective stats.**

## 3. Analysis from various Inputs

**The table given below shows the time required by each of the two mechanism to output the statistics of data set.**

| Total values in data set( in 10^6) | Time taken by multithreading (in ms) | Time taken by multiprocessing (in ms) |
|---|---|---|
| **1** | **321.45** | **345.35** |
| **2** | **502.32** | **534.46** |
| **3** | **812.33** | **877.49** |
| **4** | **1023.22** | **1135.78** |
| **5** | **1322.64** | **1455.88** |
| **6** | **1520.12** | **1710.20** |



## 4. Conclusion

**Multiprocessing and multithreading both increases the performance of the system. Multiprocessing is adding more number of or CPUs/processors to the system which increases the computing speed of the system. Multithreading is allowing a process to create more threads which increase the responsiveness of the system. Multithreading increases the responsiveness as if one thread of a process is blocked or performing the lengthy operation, the process still continues. The second benefit of multithreading is resource sharing as several threads of a process share same code and data within the same address space.**

**Usually multithreading is faster as its creation is faster and economical too but that depends on the task we are computing.**

**If the various task are large separate with only occasional communication between them, then multiple processes offers the advantage of being able to split the processes across different compute servers.**

**Hence in the program, as multithreading takes the advantage of overhead of process creation and terminations,thus multithreading in above is better.**