**Q3. [20 points] Compare the multi-threaded, concurrent server in Chapter 12 to the implement in Chapter 11 that uses multiple, singly-threaded processes. Which executes faster? How does the running time vary with the number of concurrent connections? You need to draw a graph or table to show different execution time for each server.**

**Answer:-**

* First of all, comparison between both the programs of Multithreaded concurrent server from Chapter 12 and Multiple Single processes with single threads could be done by compiling them with multiple clients connecting to the server programs individually.
* The major difference between the threads and the processes is that threads share the address space of the memory where the processes don’t.
* Moreover, the creation of threads is 10 to 100 times faster than the processes since threads are very lighter than the processes itself. The thread-switching time (switching between threads) is also quite less than the context switching (switching of processes).
* We compile the program of chapter 11 which has the creation of multiple process with single thread in each.
* The program in chapter 12 uses multiple threads and it is a multithreaded program.
* It is seen that the threads are faster than processes but we perform a comparison between the two after making a table and the graph as shown below:

|  |  |  |
| --- | --- | --- |
|  | **Ch 11 Multiple Singly Threaded process(sec)** | **Ch 12 Multi-threaded concurrent server (sec)** |
| 1 | 0.000296 | 0.000208 |
| 2 | 0.000230 | 0.000194 |
| 3 | 0.000495 | 0.000231 |
| 4 | 0.000354 | 0.000230 |
| 5 | 0.000243 | 0.000299 |
| 6 | 0.000199 | 0.000280 |
| 7 | 0.000212 | 0.000226 |
| 8 | 0.000292 | 0.000249 |
| 9 | 0.000255 | 0.000223 |
| 10 | 0.000342 | 0.000240 |
| **Avg.** | 0.0002918 | 0.000238 |