README

- There are 5 files in the folder server1.c , server2.c , server3. , client.c and readme.
- Download the folder and unzip it.
- **Copy all the files of the folder to the home directory and run the files from there only.**
- All the server will implement basic calculations (+,-,*,/) on positive integers (input can be 0 as well).
- The division will be an integral division.
- For subtraction first number >= 2nd number will give right output
- For division 2nd number cannot be 0.
- **Write expressions without spaces in between** . For eg : “8+90”,”7-2”,”900*67”,”80/4”

SERVER1

For running the first server type in terminal “gcc server1.c -o server” “./server 9898”. 9898 is the port no. any other port no. can also be taken.

Open another terminal and type “gcc client.c -o cli”.
“./cli 127.0.0.1 9898”.
The port no. for server and client should match.

If another terminal is opened and cmd “./cli 127.0.0.1 9898” is given then the server will not connect as it is a single processor server.
It will give error “Error in connection”.

SERVER2

For running the second server type in terminal “gcc server2.c -o server”
“./server 9898”. 9898 is the port no. any other port no. can also be taken.

Open another terminal and type “gcc client.c -o cli”.
“./cli 127.0.0.1 9898”.
The port no. for server and client should match.

This is a multi processor server .
We can open many terminals and run the cmd “./cli 127.0.0.1 9898”. This terminal will cater to all the clients.

SERVER3

For running the second server type in terminal “gcc server3.c -o server”
“./server 9898”. 9898 is the port no. any other port no. can also be taken.

Open another terminal and type “gcc client.c -o cli”.
“./cli 127.0.0.1 9898”.
The port no. for server and client should match.

This is a single processor(multi threading) server which uses select system calls .
We can open many terminals and run the cmd “./cli 127.0.0.1 9898”. This terminal will cater to all the clients.

PERFORMANCE COMPARISON

- The 2nd server is a multi processor server it uses fork system calls to create child processes whenever a new client requests the servers.In fork system calls the code data and instructions as well as other things like registers and stacks get separated

for 2 processes.So naturally the CPU switch and other things get slower comparative to other servers over here.

- In the third server there is only one process running .In threads only stuff like stacks and registers get separated . However the code data and instructions remain the same for all the threads .Thats why OS treats different threads (User level thread) as the same process.Thats the reason CPU switch is comparatively faster here.
- So in the fork system call we are using more computer resources as compared to select.Select command allows to monitor multiple file descriptors, waiting until one of the file descriptors become active.