May 7, 2017

INTERACTIVE CLIENT-SERVER APPLICTION

SIR SHABBIR MUKHI | SYSTEMs PROGRAMMING

Report in printed form

* Architecture
* All limitations
* Flow of the Code/Logic
* Anything additional you want to highlight or you are proud of

The server and client both needs to be interactive and as per class discussion and architecture should be implemented. Exact mix of technique – threads, process and/or multiplexed I/O is your choice but you should be able to justify whatever you choose

| Task | Done |
| --- | --- |
| Mathematical Operations ( add , multiply, divide , subtract) |  |
| Run ( fork and exec – creation of a new process, return success or failure) |  |
| Kill ( by PID , by name , all -pertaining to client’s request) |  |
| List ( all processes, active processes – PID , name, status, start time, end time, elapsed time, should not contains processes which are terminated after failure in exec-pertaining to client’s request, updating lists based on signals sent from within or outside the process) |  |
| Implementation of communication mediums between client and server processes, and between main/central server and sub/child servers ( sockets / pipes-take care of record boundaries) |  |
| Server must be able to handle multiple clients(pre-fork or fork upon accept) |  |
| Interactivity in server using threads for separate functions such as addition, taking input from clients, etc (re-use threads by blocking threads assisted with multiplexed I/O if necessary) |  |
| Interactive client by using multi-threading ( thread for taking input from user, another for taking response from the server) |  |
| Connect and Disconnect on client |  |
| Close sent by client should close all processes associated with that client on the associated server process, and should return to client and close message sock |  |
| Exit and help on client |  |
| SERVER COMMANDS |  |
| List (all processes, display on central server, collect individual clients’ lists-use pipes or sockets with identification of client whose list is being printed, use of dynamic arrays/linked list/vector) |  |
| Kill (from central server to it’s children to kill a process based on PID, name or kill all processes) |  |
| Message from central server to child processes (sub servers) to print desired message |  |
| Disconnect <Client Name/IP> Disconnect a specific Client after termination all its processes and terminate its server peer and then client should disconnect too |  |