Instruction Manual

Kickstarting the project:
- Run server and client through terminal
- Server is bind on port 55000 originally, but can be changed (port forwarding required for across network access). Running command: ./server
- Running command for client: ./client ServerIP ServerPortNumber

Commands

Server side:

Server is interactive for the following commands:
1. list : prints the list of all processes requested by all clients e.g. gedit, bitmap. Processes are empty line separated to distinguish processes spanned by different clientHandlers.
2. list Index: prints list of all processes spawned by a specific clientHandler identified by the index in the "clients" table
3. print sentence: prints entire sentence followed by print on all client's screens
4. print index sentence: prints sentence on the client screen identified by aforementioned index
5. conn list: displays active clientHandlers information and through them, the information of clients including client port number, ip.
6. exit: terminates the connection process

Client side:
1. add a b c … (adds provided lists of numbers and prints result)
2. sub a b c … (subtracts 1st number from the rest of the numbers and prints result)
3. mul a b c … (multiplies all numbers and prints result)
4. div (divides 1st number from the rest of the numbers and prints result)
5. list (prints all running processes that were requested by client to run using the following command)
6. list all (lists all processes, either active or inactive, processes are killed using the following command)
7. run programName (runs the program e.g. bitmap, gedit)
8. kill name (kills process of the given name)
9. kill processID (kills process of given pid)
10. exit (terminates client)

ClientHandlers are spawned for communicating and processing requests of each client. It, and the client, both are gracefully terminated once the connection breaks between the two eg. either terminates.

Limitations:

- If SIGINT is sent from controlling terminal, all connections would end
- Div, sub partition between the first number and the remaining
- Negative numbers count as an invalid input (tho they can be in output)
- First argument of print and list can be either completely alphanumeric or completely numerical
- Buffer size

Architecture Diagram

There is 1 to one mapping between each client process and client Handler processes. Clients can be many.
Connection process is 1.

Terminal

Client

Thread 1 (read from user write to clientHandler)

Client Process

Thread 2 (read from server, print on terminal)

SocketFD

SocketFD

Server

ClientHandler

ServerInteractionThreadReader

Reads frim clientHandlers and prints to terminal

ServerInteractionThread

Reads from terminal, writes to clientHandler

ClientHandlerThread

Connection Process

These two threads communicate with all ClientHandlers through Multiplexed I/O

Communicates with connection process

Spawns Handlers. Maintains record. Communicates with Client Handler for commands like list, print

Terminal input