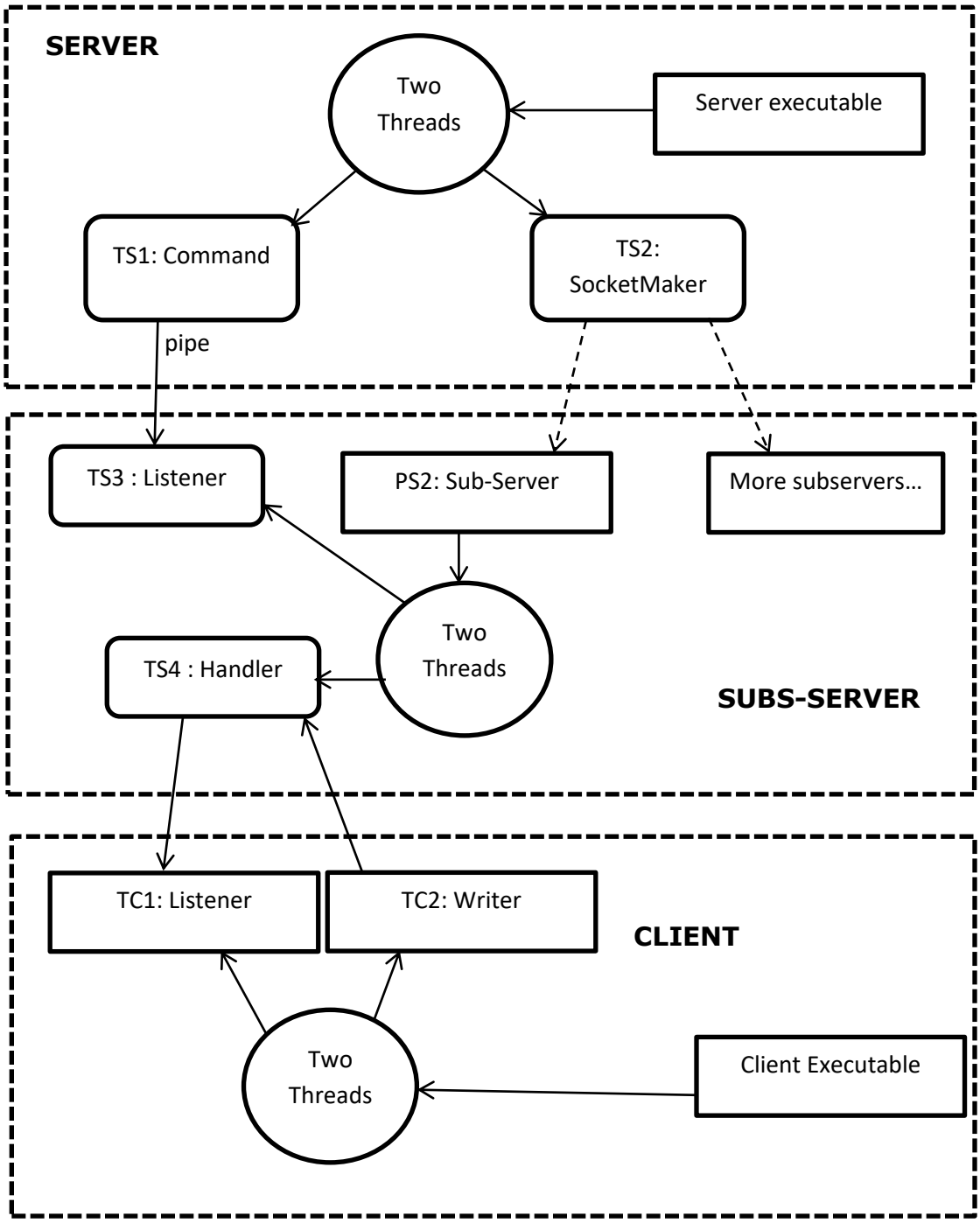
System Programing

Syed Ahsan Tanweer
09381

Architecture

The general flow of the program is as follows:



Above diagram denotes the whole architecture.
It is pertinent to mention that the client can have a “Connected” or “Disconnected” State. The threads exist only in connected state.
Similarly, the “PS2: Subserver”, would also have more child process from the execs that it would be doing on behalf of the client.

Flow:

The server starts running, and immediately bifurcates into two different threads. One thread is for listening and accepting incoming connections, while the other is for receiving input and allowing commands to be executed from the server.
The thread responsible for incoming connection actively makes new process for each new client that gets connected.
Relevant information is saved in a global struct for the use by the “Command Thread”.
The client commands are entertained by the subserver process, relayed through sockets.
The server commands are relayed to the subserver process, through pipes.
On disconnection, there is two way message between the client and subserver, which causes termination of the socket and termination of subserver, and disconnection of the client.

The Client:

The Client has two threads as seen in the diagram. One thread listens to the output from the subserver, and immediately prints them out. While the second thread continuously keep asking for input, and pushes this towards the subserver.

Server:

The server creates two thread at the start of its execution. One of the thread is used for continuously taking and parsing input from the user. While the second is used for looping through the socket, and creating process for each client.

Sub-Server:

The sub server basically have two threads. One is used to listen to the server, and respond accordingly. The other is used to cater to the clients request.

Server / sub-server Communication:

A protocol has been established between the server and the sub server.
The server (TS1) maintains a struct, which contains all the information about the sub server. The struct contains the following info:

1. The IP of the client
2. The Port Number
3. The socket
4. FD of the Subserver

This means that that server can simply write on the pipe to relay any message that it wants the subserver to perform. A protocol has been established, which allow them to seamlessly communicate. The server first sends a 4 character input, representing the type of action that is required, followed by details of the action. The protocol include:

1. List = Show list of all processes running
2. Kiln = Kill Process by Name
3. Kilp = Kill Process by pid
4. Msge = Message
5. Dcnt = Disconnect

Limitations:

1. There exists a general lack of input validation, arising due to lack of time, more than a lack of concept. The commands work as they are supposed to, but may behave abnormally if invalid or insufficient input is provided.
2. A successful prompt is given even if Exec fails
3. Sometime table may get badly formatted due to misplaced output from the client.