CS242

A4

Andy Mahoney

This way there are two classes that can impalement runnable, the other classes that hold data shouldn’t be responsible for the things related to multi-threading. Also you don’t want a single client application spinning up more clients you just want something to handle the IO.

* 1. - This way there can be threads open to constantly listen and it adds a further layer of abstraction and modularity to the project. It’s called a listener because it never creates and sends its own data it just looks for anything being sent so it can echo it, it just listens.

2 - this way the server can listen and send things to each client individually, and when one disconnects not everyone looses their connection too, or if someone new joins, the whole program doesn’t need to restart and re-initialize. The listener doesn’t create data or send it anywhere, it simply receives and prints. ServerSideClientIO sends data, receives data, has its own methods and data structures and actually represents something more than just a hollow object.

2.2 - you want to make sure that broadcasting from multiple users doesn’t overlap, for example if one person says “hello” and another says “what’s up” that you don’t get “hewhllat’os up”. remove also needs to be synchronized so that all actions complete before removing them from the list this is so code doesn’t get a null pointer because the user was in the middle of an action while quitting.

3 - I just made a LISTUSERS function that iterates through the list of users and sends their data.