Project about distributed calculator

Evaldas Dmitrišin IT 1st year 3rd group student

How to use the code

1. Clone this repository!
2. In your local machine go into the task3 direcotory
3. Type: 'make' in the terminal to create executable files
4. In the first terminal run ./server portnumber
5. In the other terminals run ./client host portnumber. If you are running this server and client on the same machine run localhost for the host option
6. In order to remove the compiled files run in the terminal 'make clean'

How the code works

This code uses simple socket communication. It communicates via strings and numbers. First the client is given a choice menu. The client has to input the corresponding number. Which is then validated by the forked child1 server process. If the input was correct aka 1-3 then child1 server sends the client back a Success you chose "option" string. By this option we determine what will be done. The code distinguishes what needs to be done by using strstr which checks if the response contains the "option" in the response string. Then we proceed with the option and ask for client input. The input is once again validated by the child1 server. And if the input is correct we fork the parent process again so that we have a child2 responsible for piping between the child1 which is responsible for communicating with the client and it child2. While the main parent is responsible for listening for new connections.

The basic concept of how the piping works is that the child1 responsible for communicating sends the client inputted numbers which need to be calculated after verifying that they are correct. Child2 recieves the number from a pipe and calculates the result which then is piped through a second pipe back to the child1.

The code uses signals as well although primative the client side of things ctrl+c immediately shuts down the client and the child it was communicating with. The server signal is slightly more complicated as it takes 3 ctrl+C to shut down and after the third ctrl+c it kills all the parents children before exiting thus hopefully avoiding leaving deamons roaming in the system. 1 major oversight is that shutting down the main server **DOES NOT** kill the clients so they are left not responding.

Below I will document my successes/failures and general updates about the code.

2024/05/03

I played around a bit with sockets and server/client functionality. Have not included a loop so the converstation goes on forever nor have I made any actual functionality but played around with sending data back and forth between the server and the client! Time spent 1h 30 min

Made a bit of functionality this was more to get used to fully using sending and recieving data from client and server. Overall I think I got how it works even if I do quite a bit of errors before succeding. Time spent another 1 hour.

2024/05/04

Made it so factorial is calculated but it needs some more revision in terms of error handaling.

2024/05/06

Made the error handaling for factorial so it should run without any problems I still plan on making more descriptive error for if you have the number too big. Time spent: 1 hour!

I found out you can build the files using Makefile and then just running make! This is really good for considering I have already deleted the server once running gcc -o client.c client instead of gcc client.c -o client. So this will make sure I never make this mistake again!!! Time spent aroung 40 minutes because of tutorials and implementation.

2024/05/07

Made it so the make file uses -lm flag for Makefile so now it can utilize the sqrt function otherwise it would not work. Also made it so the code works for the second option so now it calculates the triangle area. The error handaling is not working yet and I plan to also make it so it displays what kind of triangle it is.

2024/05/08

Played around with the markdown language. Also made error checkings for sending and recieving data. However ultimetely I will change it because there is a lot of redundant code.

2024/05/13

Made new functions for recieving and sending data which checks whether or not they were succefull making the code more error proof. I'm currently thinking what the third operation should be whether it's a simple calculator or something else.

Refactored the code as well as made it so the triangle area calculation also prints what kind of triangle it is based on the lenghts of the sides. Will start trying to implement the forking Time spent 45 minutes.

Alright I managed to make it so that the server is forked for each user. There was a lot of bugs from the previous version such as not having cleared variable and so on. Even now it has a lot of bugs that need to be fixed. Time spent ~1 hour.

2024/05/14

Added simple signaling and bug fixes. It turns out the bugs from the simple hello server message which I forgot to remove and spent 1 hour on this :/ Anyways aside from that made sure the buffer is cleared inside the triangleArea calculation

2024/05/16

I added my first pipe for factorial. Will do the same with the Triangle area later today. Also remade the factorial calculation function. Also later plan for today is to add the option for specifying in the command line what the ip and port is so we can have not just localhost communication but also from vm to vm

I managed to make the piping work for AreaCalculation as well. As well as making that the script takes in the portnumber and the host for client. It works on the 1 machine if you write localhost have yet to try running in 2 separate machine. I want to wait until I make sure error handaling is made correctly.

2024/06/17

I started trying to make it so the client works in a loop as well as the server but there seems to be a problem with the pipes or more correctly with pipes that calculate factorial result since from the second loop it says no matter what the input is that the answer is 1? Will try to fix today if I can

The loop works now but there is a problem with singnaling now and the loop stops at 6 factorial or 3 triangle. I assume it's because of pipes somehow being overflown because the timing is consisten when the loop stops working

2024/06/18

Made it so the code handles infinite loop with the multiple clients. Took over 3 hours. Added a little bit more secure exiting as now all the server children are killed after the third ctrl+C. Added so that the user actually exits if the client inputs number 3 in the choice menu. **Know bug** if you choose option 2 in the very first cycle of the loop TH appears for no reason? It is gone afterward just in the very first option.