Systems Programming Final Project Report

Student name: UGURKAN ATES

Student number: 151044012

BIG NOTE = this program somehow only works perfectly on CSE 344 debian with

```
valgrind --leak-check=full -v ./your_program
```

PLEASE TEST LIKE THIS . CLIENTS doesn’t need valgrind check

THANK YOU !

In this project, we were supposed to implement client-server based application with threadpools .

The application is supposed to be of the following form. Server program is supposed to simultaneously handle connections from multiple clients,.

Server = serverHomework is acting like central thread maker. When it reads providers from file it creates a thread for each in threadpool. Every thread has a struct associated to it. It has queue size of 2, tasksCompleted so far,logTime starting with thread created,price,quality and name of that thread which means provider name.

We are supposed to get a request from client (which opens on different terminals and you can open as many you want). Each different terminal requests comes socket connection. I immediately create a new thread to this connection .This thread uses thread_selection to choose best with their own interest. According to their interest thread_selection assigns already created provider threads to make cheating (taylorFunction to calculate) and result cosinus value while holding time etc..

Now I’m gonna tell things I was challenged pretty hard while making final project.

1- Threadpool creation – sync
This was very hard because I had to use multiple mutex and 1 condition variable to fix it. Synronization plays big part on making this project reality. Because everything needs to be fixed in order to log.dat work correctly. Also it was a proper must to thread array work correctly.

2- Socket Creation- Getting data for first time
I never used sockets before since I didn’t do last year final.This was pretty hard considering there was so limited info and its tapping another type of programming entirely. I got help from internet references about topic and how handle a basic communication , found book useless on this part. Because book code is not working exactly as it says.

3- FILE synchronization
Log.dat went through a lot of stages to work correctly.There are lot of thread due providers and connections it makes it hard to get which one is first came etc. Found solution with using another mutex.

4- MEMORY leaks – valid free()
This was literally hardest part because Debian system is acting like a little girl in this topic. I made entire homework inside of native linux 18 sarah system with normal gcc and tried both on IDE and terminal countless time. But on linux Debian system provided by school program acted weirdly. I had to search 10 -12 hours after I completed final. My eyes were bleeding , I solved most of memory problems but correct output still not came to me. It only works correctly with valgrind option. There are still 1-2 leaks but those are due pthread and couıldnt be solved unless used thread_cancel . Thread cancel brokes my algorithm

PLEASE RUN program with valgrind enabled.

Thanks.