

Final Report

BERKAY BAKIŞOĞLU 141044061

June 2021

1 Server

1.1 Main

Simply check parameters and calls daemonizeServer function.

1.2 isServerExist

When server trying to open,it checks a file which is hardcoded,if its exist,it means a server is running.Otherwise it creates that file and writes its pid to inside.

1.3 DaemonizeServer/Main Thread

Before the server creation,first isServerExist function checks the server file.Then its legal to create server forks and close the parent and do setsid to create a new session,closes terminal ,reads the file, bind its port to host, listen max 100 and creates worker threads.Then until signal received,firstly it waits for ready threads using mutex and condition variable,if there are any It will accept client requestp,pass file descriptors to socketQueue by duplicating,close itself and post cond signal to wake any thread which is waiting.

1.4 WorkerThreads

After a worker created,it increases readyWorker variable by one and send condition signal to main thread to let it know any thread ready.Then it will wait with condition variable and mutex until queue has any client sockets in it.It gets the socket and decrements variable of readyThreads by one.Then in a loop,it waits client for;Sending the query size,if it is -999 which i choose for end,it will breaks the loop and return waits for new connection from main thread.Else it gets the query from client with another recv.After getting query;

1.5 WriterReader

Worker checks the query to detect if it reader or writer by select or update. If query is reader, then it increases the value of waiting readers by one, if its writer increases waitingWriters. If both of them is not true, it does nothing cause it will probably be a syntax error when parsing. Then Readers waits on their condition variable and mutex until there is no thread is writing or waits for write. Writers pass this and waits on their condition variable and mutex until there is no thread is writing. Then it parse the query and get results, then waits for 0.5 sec and sends client the respond size and result of query. It decrements reader or writer variable depends on its type, signals condition variables which is needed, Then it returns back and receive another queries until client sends the quit size.

1.6 Recv,Send

This functions might not get full size of variable to buffer when reading or writing, thats why a function using sizes of variables constantly recv or send until the desired value is processed.

1.7 Variable Protection

Log file protected by fileMutex, socket queue protected by socket mutex inside their functions.

1.8 Database Struct

I decided to keep database variables as 3d char array. It may be more effective with B-tree or something but arrays are easy to implement and $O(n)$ to access which is not bad.

2 Clients

Client checks parameter and do connect server, then it reads the query file line by line, if its id equals to selected line, it will send to server query size and query to server, then it starts the timer and waits for respond size and result from server. After all lines processed, it sends -999 message to server to let it know that client job has done and exits.