

FINAL / REPORT

Can Duyar - 171044075

Solving The Problem And Design Decisions:

SERVER:

parse_args:

This function parses arguments and check if those are valid or not. If not valid then it print messages and exit. After that this function opens log file.

getColumn:

This function reads columns from first line of given file and then it returns column number.

init_table:

This function initialize table struct object. It allocated memories where it needed.

fill_table:

This function copy all data from given char* array to table object.

free_table:

This function free up table object what allocated at init_table function.

split:

This function splits a given line into column cells.

getArray:

This function copies all data from link list to an array of char* and free up link list using recToStr function and recFree function.

recToStr:

This function copies all data from linked list to an array of char*.

freeString:

This function frees allocated char* array.

recFree:

This function frees allocated link list.

write_buff:

This function sends a string to client and get conformation about is it reached to client or not.

client_handler:

Pool threads runs this function. They do their job in this function.

handle:

This function will called by pool threads from client_handler function. This function reads SQL query from client and returns result from table object.

getIndex:

This function returns index of a column by name.

terminate:

This function is called when user send a termination signal. At this function all allocated memories will be freed and program will exit after current working threads are finished.

skeleton_daemon:

This function creates a daemon process that runs in background.

getIndex:

This function returns index of a column by name.

MONITORS:

In server, while writing a log to logfile it usage monitor to synchronize. And also while update command is received. So a single thread can write to log file at a time.

CLIENT:**parse_args:**

This function parses arguments and check if those are valid or not. If not valid then it print messages and exit.

read_buff:

This function reads from server and sends a conformation.

func:

This function opens query file and reads line by line. For every line it is checking id, if its then it sends that to server and reads results and prints that. After finishing the file its exited.

I couldnt implement UPDATE command. It works with every select commands. Every pool threads can read table at same time so, it doesn't require any file locks. If it gets an UPDATE command then it just sends an error message to client. Server's main function at first calls skeleton_daemon function and creates this as a daemon process. After that it parses arguments and registers signals. And then it reads the dataset file and stores them in an self reference array of LinkList struct object. Then main function calls others functions as needed. Client's main function open and connect a socket to server socket. And then this function calls others functions as needed.

Which requirements I achieved and which I have failed:

I checked it with valgrind and there was no memory leak, and I also checked the number of zombie processes after the program running and number of zombie processes were 0. I couldnt implement UPDATE command. SELECT and SELECT DISTINCT commands work properly.