Gebze Technical University
Computer Engineering

System Programming
CSE344 – 2021

FINAL-REPORT

Yusuf Abdullah ARSLANALP
151044046

0.1 How I Solved This Problem

0.1.1 Data Structure

CSV file loaded into an 3D character array. The array created dynamically with malloc. And freed at the end of the program.

The most significant drawback of array is deleting record. But we have no delete command in the homework. And array has constant accessing complexity. And array easy to implement in C. So I used array.

0.1.2 Sending Data

The data sent from server to client might be so big. So it can't be sent as a whole. So I sent data from server to client as rows. But client should know if data is completed or not. For this reason server first send the length of the row and then sends the row. Zero row length means that all rows are sent. So no more rows will sent after zero line length.
The client side of data communicatian as the following figures:

```
90      while( 1 )
91      {
92          read( fd , &size, sizeof(int));
93          //printf("size: %d\n", size );
94          if( size == 0 )
95          {
96              break;
97          }
98          read( fd , response, size );
99          printf( "%s\n", response );
100     }
101 }
```

0.1.3 Preventing Multiple Instantiation

I used temporary file for preventing multiple instantiation of server. At the beginning of the program server check if there is a file with name of "15104404_temp". If the file exists it prints an error message and terminate. If no file with the specified name it creates the file. And at the end of the program it removes the file temporarily created file.

0.2 Which requirements I achieved

- The report prepeared via latex. (latex folder is in the homework)
- All synchronization problems solved with condition variables and mutexes.
- No input file modified.
- I have a make file. It only compiles files

- The server run as a single instance.
- Server is a deamon. (No possible double instantiation)
- I loaded csv in array form (dynamically allocated).
- Thread pool works as expected.
- The threads sleeps additionally 0.5 seconds to simulate intensive database execution.
- All queries works properly. (select star, select distinct, select column, update)
- On SIGINT all processes terminated. But there are some memory leak.
- No warning with -W flag
- If the required command line arguments are missing/invalid, The program prints usage information and exit.
- No busy waiting.
- The make file only compiles the program.

0.3 Which Requirements I Fail:

- Memory leak on server.
- SQL commands are not perfectly parsed. Some valid commands might not be parsed.

0.4 Notes:

- For creating demon and creating sockets I used some code from internet. But for deamons there are already samole code in course slides.

Screen soot for the log file of server as follows:

```
1  Executing with parameters:
2  -p 20000
3  -o /home/yusuf/Desktop/System/Final/send2_test/send/pathToLogFile
4  -l 5
5  -d /home/yusuf/Desktop/System/Final/send2_test/send/test4.csv
6  Loading dataset...
7  Dataset loaded in 0.25 seconds with 17 records.
8  A pool of 5 threads has been created
9  Thread #4: waiting for connection
10 Thread #3: waiting for connection
11 Thread #2: waiting for connection
12 Thread #1: waiting for connection
13 Thread #0: waiting for connection
14 A connection has been delegated to thread id #4
15 Thread #4: received query '5 SELECT * FROM TABLE;'
16 Thread #4: query completed, 16 records have been returned.
17 Termination signal received, waiting for ongoing threads to complete.
18
```