GTU Department of Computer Engineering
CSE 344 - SPRING 2022
MIDTERM REPORT

Burak ÇİÇEK
1901042260

1.Design Decisions and Concept

I created the respective ServerY and ServerZ as daemons and prevented them from being created more than once by creating a file to prevent it from being run twice the first time it was run. My design is as follows:

-Server creates as many workers as the Y pool size.

-The request reaches ServerY first.

-Appropriate workers immediately grab the client's request.

-I can easily check the availability of workers with the help of shared semaphore.

-As I send the necessary information to the workers with Pipe, it also gets rid of the busy waiting while reading.

-If there is no suitable worker, I transfer the incoming request to serverZ.

-ServerZ shares the incoming request with its children with the help of shared memory.

-In ServerZ, if there is no suitable worker for the request, the request is queued and is grabbed by the appropriate worker.

-I used the design in the book to send a special response for each client.

-I used semaphores to track busy or idle workers in ServerY. But since I put these semaphores in shared memory, I was able to share them between processes.

- At the same time, I get support from shared semaphores in order to keep the number of invertible, non-invertible and forwarded requests on Server Y. This way, I can easily track them.

-I use the same logic that I used in the above article in ServerZ, only because I can't forward the requests in Server Z to a place, my semaphore count there is less.

-Relevant response is sent to each client and I keep the information of whether the matrix is invertible or not with an integer value.

-My request contains the client's PID, the corresponding one-dimensional matrix, and a dimension of it as the matrix is square.

-My request contains the client's PID, the corresponding one-dimensional matrix, and a dimension of it as the matrix is square.

-There is a value in my response that indicates whether it is invertible or not.

-I check that the -p and r values are also greater than or equal to 2

-When the -SIGINT signal comes, I delete everything, close the necessary files, close the application and print all the statistics to the log file.

-I can get the relevant values of the flags using getopt

-I use the file locks we learned in the lessons so that the synchronization is not disturbed while writing to the file.

-I am using FIFO, Pipe and Shared Memory parts correctly in this assignment.

-I am checking whether the matrix is square or not in the client.