CSE 344 – SYSTEM PROGRAMMING

MIDTERM PROJECT

151044072 – ALPER YAŞAR

Purpose of this Project?

Creating a simple client – server application. I am used FIFOs, pipes and and shared memory for this Project. All these supplying ınterprocess communication. There are 3 programs to be implemented: client X, server Y and server Z. There will be a single instance of Y and of Z, and there can be an arbitrary number of client processes X, all running concurrently. Each client will submit a matrix to the server Y, and receive a response (from Y or Z) about whether it is invertible or not.

0)Data

The data file will be in CSV (ASCII) format. It will be comma separated with no quotation marks. It will contain n integer values in each of its first n rows (n >= 2). The file contents will denote a square real matrix of size n. Ex datas:

12, 24, 17, 16, 5

4, 3, 75, 128, 36

9, 78, 42, 645, 48

71, 48, 155, 486, 41

47, 78, 96, 41, 52

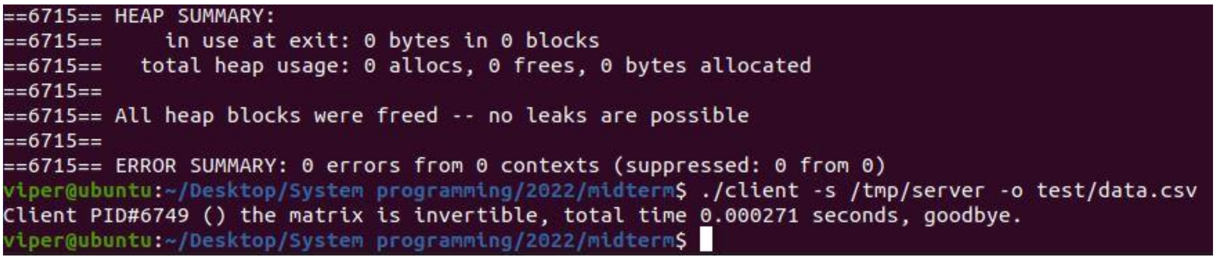
I read datas in client server :

readDatas(); first open ‘inputpath’ and check the path is okey. After check I’m taking first row for giving info as mentioned above each of its first n rows (n >= 2). If ıt’s not then printing error and exiting.

Passing file characters until to integer then taking all values to integer array and checking again is it a square matrix. If not printing error and exiting.

Client server;

After reading datas from input file, I’m creating a client path for reading the responds. Each client server called creating a client path for reading. Then create this respond path then sending datas to server and starting to wait respond from server. Reading fifo is doing this wait problem. When the respond turn calculating the spent time and checking is array is invertible or not. And printing these as standard output.



```
==6715== HEAP SUMMARY:
==6715==     in use at exit: 0 bytes in 0 blocks
==6715==   total heap usage: 0 allocs, 0 frees, 0 bytes allocated
==6715==
==6715== All heap blocks were freed -- no leaks are possible
==6715==
==6715== ERROR SUMMARY: 0 errors from 0 contexts (suppressed: 0 from 0)
viper@ubuntu:~/Desktop/System programming/2022/midterm$ ./client -s /tmp/server -o test/data.csv
Client PID#6749 () the matrix is invertible, total time 0.000271 seconds, goodbye.
viper@ubuntu:~/Desktop/System programming/2022/midterm$
```

Checking the result with valgrind.

ServerY;

ServerY is a deamon process. So when created the server y process exiting. And server y continue to run in background. Firstly assigning signals with sigaction and opening file locks. Taking some variables to global variables and using them in shared memory.

Why did I used these as shared memory? I'm changing the results in difference process. For example, holding invertible as shared memory. Because increasing it in sub-worker.

Shared memory in serverY:

> glob_var -> numbers of running sub-process. Also using check how many percent of process using. If the equal to total number. Then sending the results to the serverZ.
>
> total_req -> total request to coming to serverY.
>
> Invertible -> number of invertible matrix.
>
> forw -> number of forwarded to serverZ.

After that opening server log file. I used lock when a process trying write to log. Done process job with log file then unlocking the file. I'am creating a fifo for reading the structure coming from client.

In this structure holding datas, matrix size and process id(pid).

Creating the pipe for communication with sub-process.

Creating new fork and executing the serverZ in this process.

Creating the children of process as giving in the argument. If less than 2 printing error and exiting.

Reading struct from fifo and writing it to pipe. If sub-workers of serverY is full then sending it to serverZ.

Sub-workers reading waiting in endless loop, and waiting for reading the pipe. When pipe came process beginning the calculating. Firstly, start with the calculation of determinant and finding all cofactors. If determinant not equal to zero then it's invertible. And returning true. Increasing the invertible number and respond to client server. After all finishing sleeping child for seconds giving in argument.

There is example outputs of filelogs in the below.

```
Sat Apr 16 09:41:33 2022
Server Y (path.log, p=4, t=5) started
Sat Apr 16 09:41:36 2022
Y:Worker PID#6475 is handling client PID#6502, matrix size 5x5, pool busy 1/4
Sat Apr 16 09:41:36 2022
Y:Worker PID#6475 responding to client PID#6502, the matrix is invertible
Sat Apr 16 09:41:36 2022
Y:Worker PID#6476 is handling client PID#6502, matrix size 5x5, pool busy 2/4
Sat Apr 16 09:41:36 2022
Y:Worker PID#6476 responding to client PID#6502, the matrix is invertible
Sat Apr 16 09:41:38 2022
Y:Worker PID#6477 is handling client PID#6503, matrix size 5x5, pool busy 3/4
Sat Apr 16 09:41:38 2022
Y:Worker PID#6477 responding to client PID#6503, the matrix is invertible
Sat Apr 16 09:41:38 2022
Y:Worker PID#6474 is handling client PID#6503, matrix size 5x5, pool busy 4/4
Sat Apr 16 09:41:41 2022
Y:Forwarding request of client PID#6504, to serverZ matrix size 5x5, pool busy 4/4
Sat Apr 16 09:41:41 2022
Y:Forwarding request of client PID#6504, to serverZ matrix size 5x5, pool busy 4/4
Sat Apr 16 09:41:41 2022
Z:Worker PID#6478 is handling client PID#6504, matrix size 5x5, pool busy 1/4
```

ServerZ:

ServerZ is a deamon process also. And serverZ continue to run in background with serverY. Firstly assigning signals with sigaction and opening file locks. Taking some variables to global variables and using them in shared memory.

Why did I used these as shared memory? I'm changing the results in difference process. For example, holding invertible as shared memory. Because increasing it in sub-worker.

Shared memory in serverY:

> glob_var -> numbers of running sub-process. Also using check how many percent of process using. If the equal to total number. Then sending the results to the serverZ.

total_req -> total request to coming to serverY.

Invertible -> number of invertible matrix.

forw -> number of forwarded to serverZ.

I try to used struct for memory shared. Add and subtract but did not going well.

When I add struct I can not pass value to new process. All time get init pid and give error. So I used shared memory for how many occurs.

After that opening server log file. I used lock when a process trying write to log. Done process job with log file then unlocking the file. I'am creating a fifo for reading the structure coming from client.

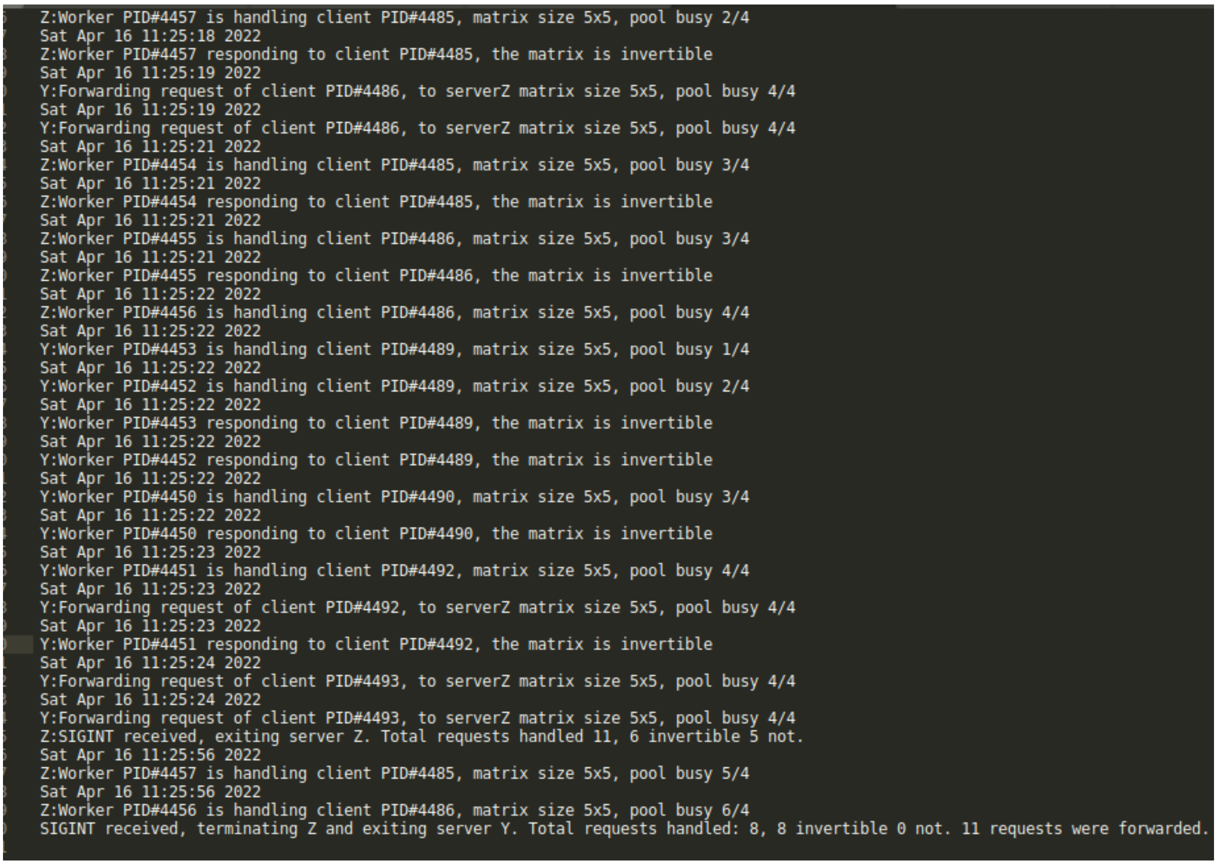
In this structure holding datas, matrix size and process id(pid).

Creating the pipe for communication with sub-process.

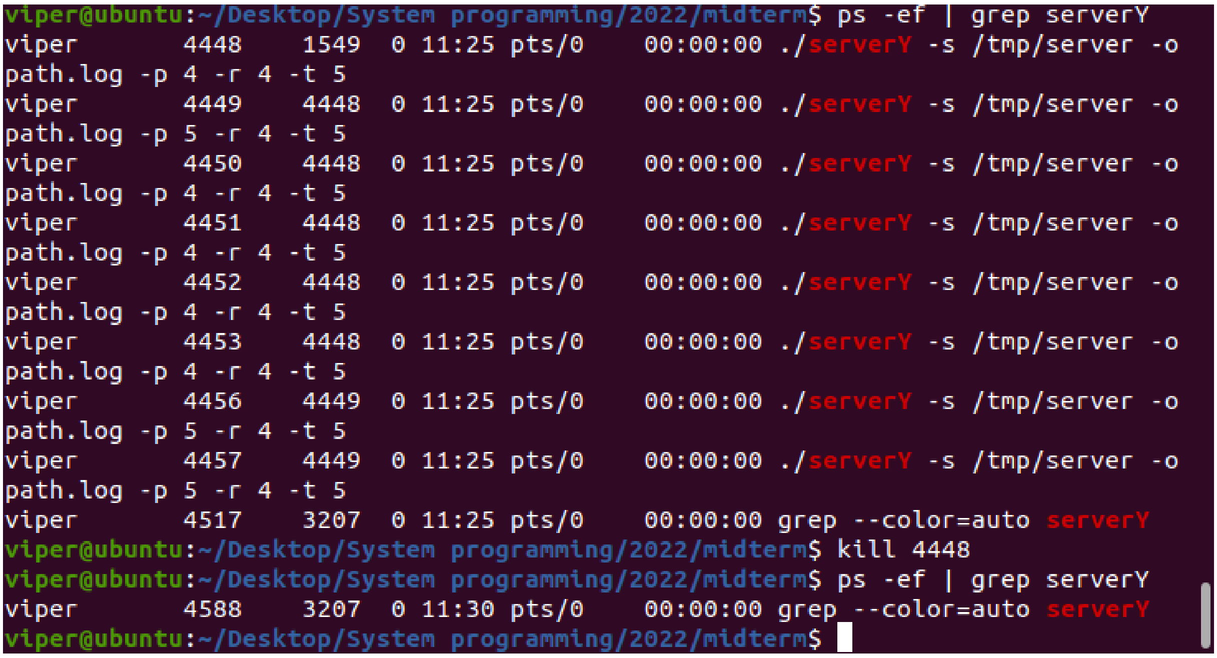
Creating the children of process as giving in the argument. If less than 2 printing error and exiting.

Sub-workers reading waiting in endless loop, and waiting for reading the pipe. When pipe came process beginning the calculating. Firstly, start with the calculation of determinant and finding all cofactors. If determinant not equal to zero then it's invertible. And returning true. Increasing the invertible number and respond to client server. After all finishing sleeping child for seconds giving in argument.

Last result after finish:



```
Z:Worker PID#4457 is handling client PID#4485, matrix size 5x5, pool busy 2/4
Sat Apr 16 11:25:18 2022
Z:Worker PID#4457 responding to client PID#4485, the matrix is invertible
Sat Apr 16 11:25:19 2022
Y:Forwarding request of client PID#4486, to serverZ matrix size 5x5, pool busy 4/4
Sat Apr 16 11:25:19 2022
Y:Forwarding request of client PID#4486, to serverZ matrix size 5x5, pool busy 4/4
Sat Apr 16 11:25:21 2022
Z:Worker PID#4454 is handling client PID#4485, matrix size 5x5, pool busy 3/4
Sat Apr 16 11:25:21 2022
Z:Worker PID#4454 responding to client PID#4485, the matrix is invertible
Sat Apr 16 11:25:21 2022
Z:Worker PID#4455 is handling client PID#4486, matrix size 5x5, pool busy 3/4
Sat Apr 16 11:25:21 2022
Z:Worker PID#4455 responding to client PID#4486, the matrix is invertible
Sat Apr 16 11:25:22 2022
Z:Worker PID#4456 is handling client PID#4486, matrix size 5x5, pool busy 4/4
Sat Apr 16 11:25:22 2022
Y:Worker PID#4453 is handling client PID#4489, matrix size 5x5, pool busy 1/4
Sat Apr 16 11:25:22 2022
Y:Worker PID#4452 is handling client PID#4489, matrix size 5x5, pool busy 2/4
Sat Apr 16 11:25:22 2022
Y:Worker PID#4453 responding to client PID#4489, the matrix is invertible
Sat Apr 16 11:25:22 2022
Y:Worker PID#4452 responding to client PID#4489, the matrix is invertible
Sat Apr 16 11:25:22 2022
Y:Worker PID#4450 is handling client PID#4490, matrix size 5x5, pool busy 3/4
Sat Apr 16 11:25:22 2022
Y:Worker PID#4450 responding to client PID#4490, the matrix is invertible
Sat Apr 16 11:25:23 2022
Y:Worker PID#4451 is handling client PID#4492, matrix size 5x5, pool busy 4/4
Sat Apr 16 11:25:23 2022
Y:Forwarding request of client PID#4492, to serverZ matrix size 5x5, pool busy 4/4
Sat Apr 16 11:25:23 2022
Y:Worker PID#4451 responding to client PID#4492, the matrix is invertible
Sat Apr 16 11:25:24 2022
Y:Forwarding request of client PID#4493, to serverZ matrix size 5x5, pool busy 4/4
Sat Apr 16 11:25:24 2022
Y:Forwarding request of client PID#4493, to serverZ matrix size 5x5, pool busy 4/4
Z:SIGINT received, exiting server Z. Total requests handled 11, 6 invertible 5 not.
Sat Apr 16 11:25:56 2022
Z:Worker PID#4457 is handling client PID#4485, matrix size 5x5, pool busy 5/4
Sat Apr 16 11:25:56 2022
Z:Worker PID#4456 is handling client PID#4486, matrix size 5x5, pool busy 6/4
SIGINT received, terminating Z and exiting server Y. Total requests handled: 8, 8 invertible 0 not. 11 requests were forwarded.
```



```
viper@ubuntu:~/Desktop/System programming/2022/midterm$ ps -ef | grep serverY
viper       4448    1549  0 11:25 pts/0    00:00:00 ./serverY -s /tmp/server -o
path.log -p 4 -r 4 -t 5
viper       4449    4448  0 11:25 pts/0    00:00:00 ./serverY -s /tmp/server -o
path.log -p 5 -r 4 -t 5
viper       4450    4448  0 11:25 pts/0    00:00:00 ./serverY -s /tmp/server -o
path.log -p 4 -r 4 -t 5
viper       4451    4448  0 11:25 pts/0    00:00:00 ./serverY -s /tmp/server -o
path.log -p 4 -r 4 -t 5
viper       4452    4448  0 11:25 pts/0    00:00:00 ./serverY -s /tmp/server -o
path.log -p 4 -r 4 -t 5
viper       4453    4448  0 11:25 pts/0    00:00:00 ./serverY -s /tmp/server -o
path.log -p 4 -r 4 -t 5
viper       4456    4449  0 11:25 pts/0    00:00:00 ./serverY -s /tmp/server -o
path.log -p 5 -r 4 -t 5
viper       4457    4449  0 11:25 pts/0    00:00:00 ./serverY -s /tmp/server -o
path.log -p 5 -r 4 -t 5
viper       4517    3207  0 11:25 pts/0    00:00:00 grep --color=auto serverY
viper@ubuntu:~/Desktop/System programming/2022/midterm$ kill 4448
viper@ubuntu:~/Desktop/System programming/2022/midterm$ ps -ef | grep serverY
viper       4588    3207  0 11:30 pts/0    00:00:00 grep --color=auto serverY
viper@ubuntu:~/Desktop/System programming/2022/midterm$
```

It's a deamon process. So I can not access process from terminal. So Checking the mother process and sending kill signal. And signal handler terminating the process. Clearing all zombies.

Checking program with valgrind and no memory leak.

What I did not?

I can not use shared memory as I thinking. In the serverZ must be a struct queue. Adding new item to queue and taking from queue. But I can not solve this problem.