GTU

**DEPARTMENT OF COMPUTER ENGINEERING**

**CSE 344 – Spring 2022**

**HOMEWORK 2  
REPORT**

SÜLEYMAN GÖLBOL 1801042656

# REQUIREMENTS

***NONFUNCTIONAL REQUIREMENTS***

1. Portability → The application should be portable. All computers that have Linux Distro and GCC compiler can run the program. Also, it can be run on Windows when Windows Subsystem for Linux 2 activated.

2. Maintainability → In case of an error occurrence, the system uses perror in order to give feedback on terminal.

3. Performance → The system should initially be able to process as many entries as possible. Each request must be processed with different terminals. The system’s performance should be fast enough to show user the feedback.

***FUNCTIONAL REQUIREMENTS***

In order to compile the program, user have to use “make” command that uses gcc. If make or gcc is not installed user can install it via *“sudo apt-get install build-essential”* command.   
Make command runs 2 commands.

**gcc -Wall main.c sg\_process\_p.c -lm -o processP   
gcc -Wall sg\_matrix.c sg\_process\_r.c -lm -o processR**  
The reason that we are creating 2 executables is because of child – parent relation. In order to get the results we need to run executable like below with input and output paths as command line arguments.

**./processP -i files/inputFile.dat -o files/outputFile.dat**

If input file exists and we have permission to save the output file, the executable will run successfully.

1. **PROBLEM SOLUTION APPROACH**

Firstly, the first big problem for me was the Covariance Matrix calculation.

In order the create covariance matrix I found the size. Because of each child has 10x3 matrix for the coordinates (10 rows for different coordinates and 3 columns for dimensions) the size of the covariance matrix must be 3x3.

For the formula of covariance matrix, first we need to get a matrix of deviation scores: a\_1x1, a\_1x2, . . . , a\_10x3 .

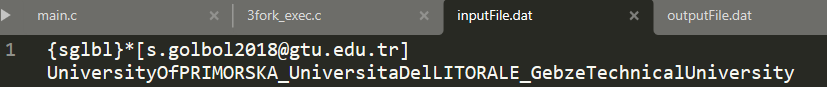
Formula of deviation score matrix:

Deviation score matrix: a = A – O \* A \* ( 1 / n )

In the formula above, O shows a matrix full of One’s. And the size of it is 10x10 (because of coordinates) to multiply.  
After we get the deviation score; if we multiply transpose of the matrix with this matrix, we get a new matrix that is n (size, in our case 10) times bigger than covariance matrix.

And if the divide this value to n (10), we get the covariance matrix.  
  
 Covariance matrix: a‘ \* a / n

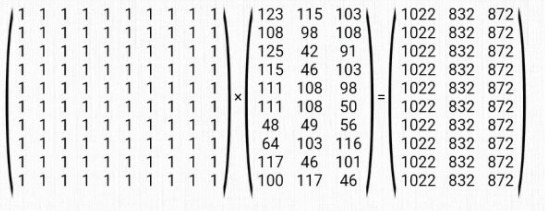
**Example** from my input file:



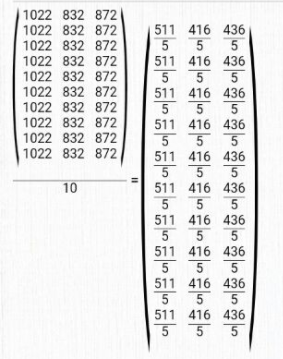
My input file starts with this line and when my program reads first 10 coordinates (30 bytes);  
It gives this output to stdout:

To calculate covariance matrix, first I needed to get the deviation score matrix.

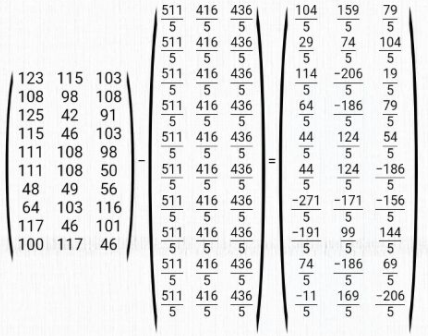
O \* A



O \* A \* ( 1 / n )

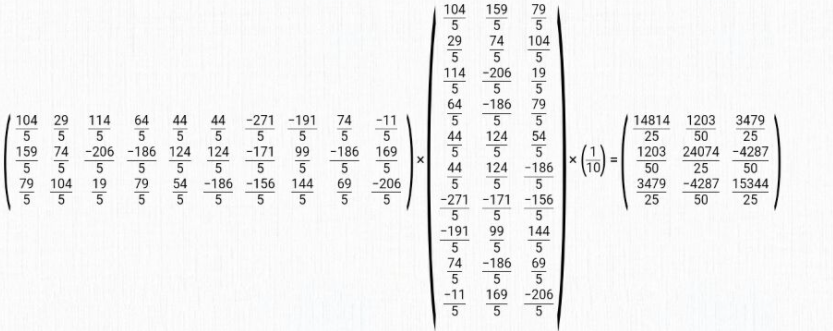


a = A – O \* A \* ( 1 / n )



Now if we multiply its transpose matrix with itself and then if we divide to the size we will get covariance matrix.

Covariance matrix: a‘ \* a / n



This value is my covariance matrix of first child.

After this, for the frobenius norm, it wasn’t difficult. I just summed the squares of output values and then get the square root of it.

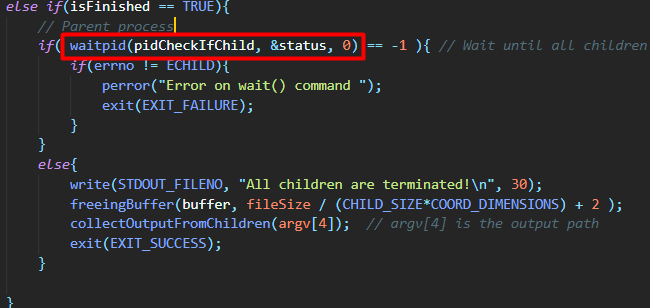
**INTEGER/DOUBLE - STRING CONVERSION**

Another big problem that I was into was printing integer or double values using write(STDOUT\_FILENO, …) system call.

write() doesn’t have formatter like printf does. So it needs char\* as argument and itoa is not C standard function for every environment so I tried to create a int variable and send it’s address to function but it didn’t work while printing so I tried to cast to void\* and dereferenced it , also it didn’t work. So I wrote my own itoa function that works for nonnegative ascii integers.

Also I wrote my own double to string function for double value printing. I thought 2 parts of double(before dot, after dot) as different values then I converted them to string with putting “.” between these two.

**WAITING**  
While I was writing my code, parent needed to wait for all the children to finish so I used waitpid in order to success.



Because of that’s a system call I also checked errno to make sure that it succeeded. After waiting I freed everything and collected outputs from children.

**3 ) TEST CASES AND RESULTS**

# 3.a) ./processP -i files/inputFile.dat -o files/outputFilePath.dat

# It reads file from files/inputFile.dat path and writes output to the outputFilePath.dat path. Then calculates distance and prints to stdout.

# 3.b) ./processP Invalid Command Line Argument to Check System Calls

# In the case of invalid command line argument program exits with perror and prints instructions.

# 

# 3.c) Interrupting with Signals

çç

FOTO

çç

# *VALGRIND MEMORY RESULTS*

# The output from valgrind about heap and leaks is like below:

# 

# *CHECKING FOR ZOMBIE PROCESSES*

