GIT Department of Computer Engineering
CSE 344 - Spring 2020
Homework 2 Report

Esra EMİRLİ
151044069

1 Problem Solution Approach

Arguments checks were performed with getopt. Then, sigaction was created for the child and parent and filled with default values. SIGUSR1, SIGUSR2, SIGPIPE signals were selected for the child to be used. When more than one SIGUSR1 was sent from parent to child, the child could not handler them. Therefore, SIGPIPE was used to eliminate this deficiency.SIGUSR1 and SIGUSR2 were considered sufficient for the parent. At critical point, the signals to be masked were added to sigset to mask the SIGINT and SIGSTOP signals. A tempfile was created via mkstemp. Fork is made and operations are separated according to pid.

1.1 Process P1 (Parent)

Ascii characters are kept in the input. Transactions are made only for ascii characters.

First, it looks at how many 20's are there to read in 20 bytes. The first cycle depends on this. If it is in the process sequence, it reads 20 bytes and saves each byte to the x, y arrays as int. It creates the critical region to calculate after the first 20 ends. Masks previously assigned signals that SIGINT and SIGSTOP. Calculations are made by following the steps below.

Step 1: For each (x,y) point calculate x^2 and xy

Step 2: Sum all x, y, x^2 and xy, which gives us Σx, Σy, Σx^2 and Σxy (Σ means "sum up")

Step 3: Calculate Slope **m**:

$$\mathbf{m} = \frac{N\,\Sigma(xy) - \Sigma x\,\Sigma y}{N\,\Sigma(x^2) - (\Sigma x)^2}$$

(N is the number of points.)

Step 4: Calculate Intercept **b**:

$$\mathbf{b} = \frac{\Sigma y - m\,\Sigma x}{N}$$

Step 5: Assemble the equation of a line

$$y = mx + b$$

Done!

https://www.mathsisfun.com/data/least-squares-regression.html

The masking is unblocked when the calculations are finished.Before writing to the temp file, it is checked if the file is used by the child, whether it is locked. If locked, the child is suspended until a signal that he is done. if not, it locks and goes into writing. Writes using writeToFile function and unlocks the file. This process takes place until the inputPath reading is finished. After the inputPath is finished, it sends the SIGPIPE to the child to inform child that parents job is over. Closes the file it read and it prints the number of lines and bytes that it reads on the screen. When Parent learns that the

child is over, he closes the input file and sends the SIGTERM signal to terminate program.

1.2 Process P2 (Child)

When the parent ends, it will send the child a SIGUSR1. So the main cycle of the child is connected to the SIGUSR1 .The other loop depends on the number of lines it will read.

First, since the temp file is empty, it was first suspend until any signal was received from the parent. When the parent sends the signal(SIGUSR2) to end the child's suspend, the number of lines to be read in the hander is increased. Thus, the inner loop is entered and whether the file to be read is locked or not. If is locked, child is suspended and the parent is expected to unlock and send the SIGUSR2 signal. If not, the file is locked to prevent the parent from taking any action during reading.

The file is read character by character and all these characters are converted to numerical values and kept in the arrays. In this reading, it is firstly checked whether there is a numerical value, if it is a numerical value, it is recorded and translated into the array up to non-numerical value in order to hold the 2-digit numbers. Also, similar work is done in getLSM function for decimal numbers. When the reading process is finished, the lock is removed.

Signals previously assigned to the critical region were masked before coming to the calculation section. The masking was unblocked after the calculation was completed.

Then mathematical calculations were made according to the formulas below.

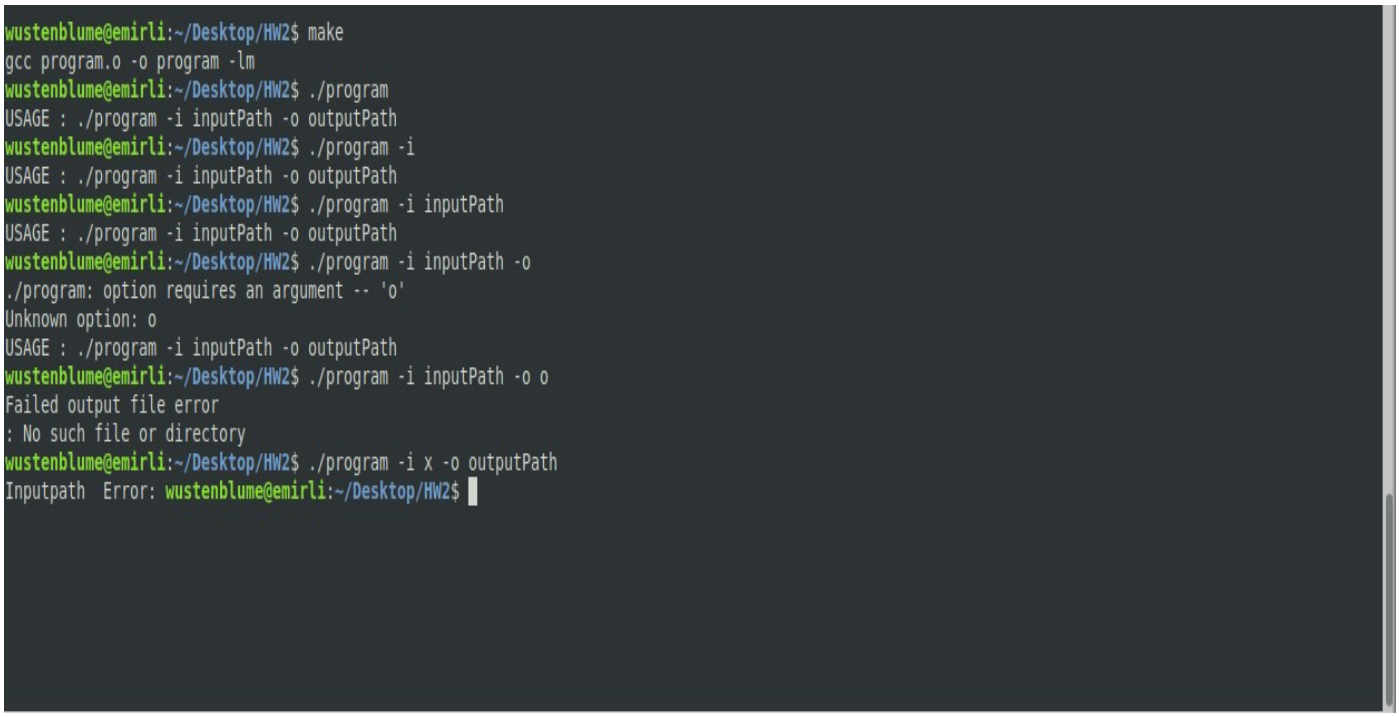
Mean squared error	$\mathrm{MSE} = \frac{1}{n}\sum_{t=1}^{n} e_t^2$
Root mean squared error	$\mathrm{RMSE} = \sqrt{\frac{1}{n}\sum_{t=1}^{n} e_t^2}$
Mean absolute error	$\mathrm{MAE} = \frac{1}{n}\sum_{t=1}^{n} \lvert e_t \rvert$
Mean absolute percentage error	$\mathrm{MAPE} = \frac{100\%}{n}\sum_{t=1}^{n} \left\lvert \frac{e_t}{y_t} \right\rvert$

And these results were given to the writeToOutput function to write to the outputPath. it was then suspended until a new signal came from the parent. If the parent end signal comes up, when the child is finished, child prints the calculations on the screen and sends USR1 to tell parent that its job is over. The child closes the temp file and deletes it.

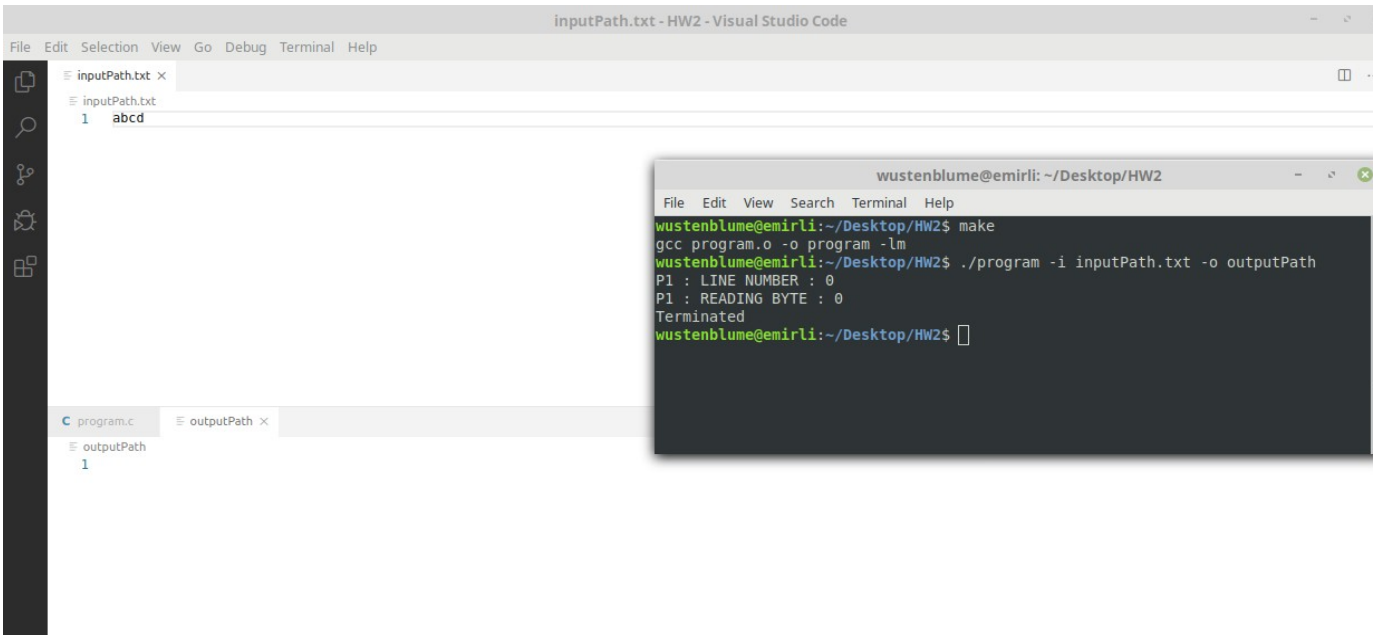
- I did not fully understand how you want the critical region to work. I gave the signals to be blocked but I did not handler.
- In P1, I could not do the part of the values in the critical region.

2 Running And Results

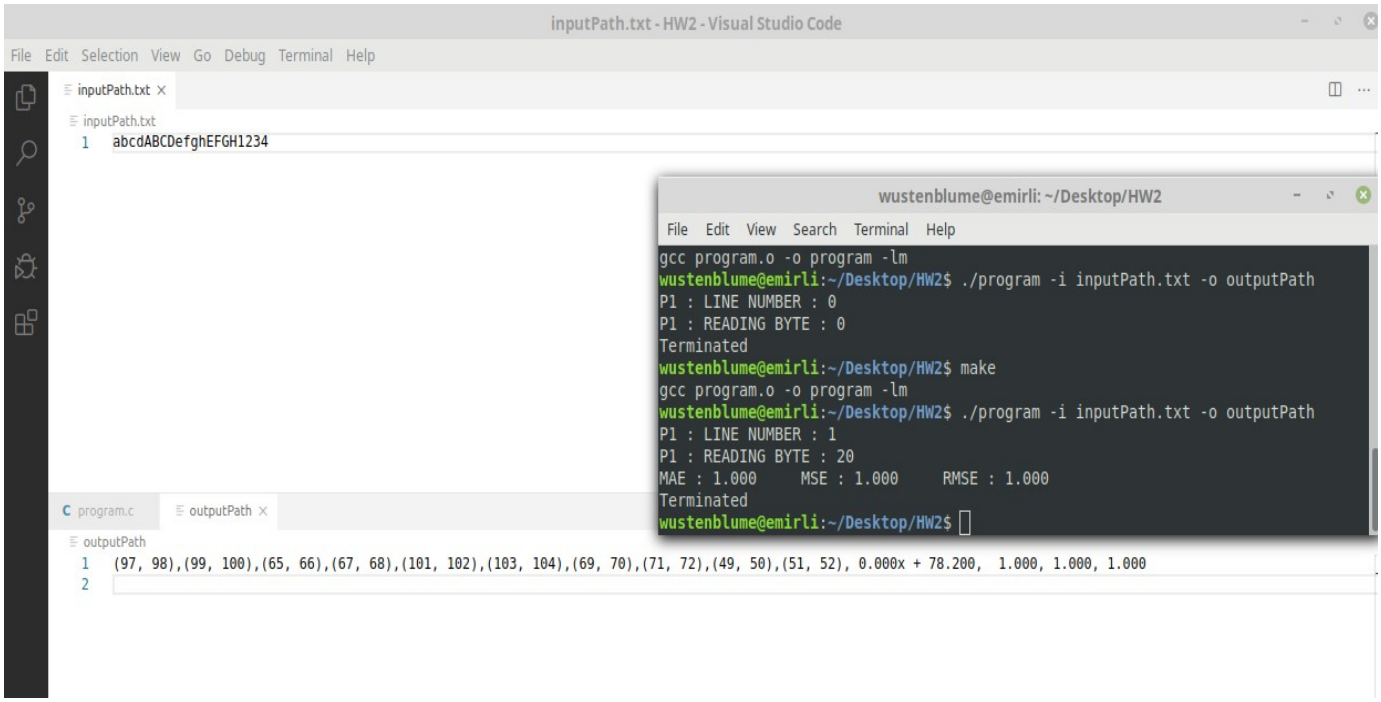
- Argument controls



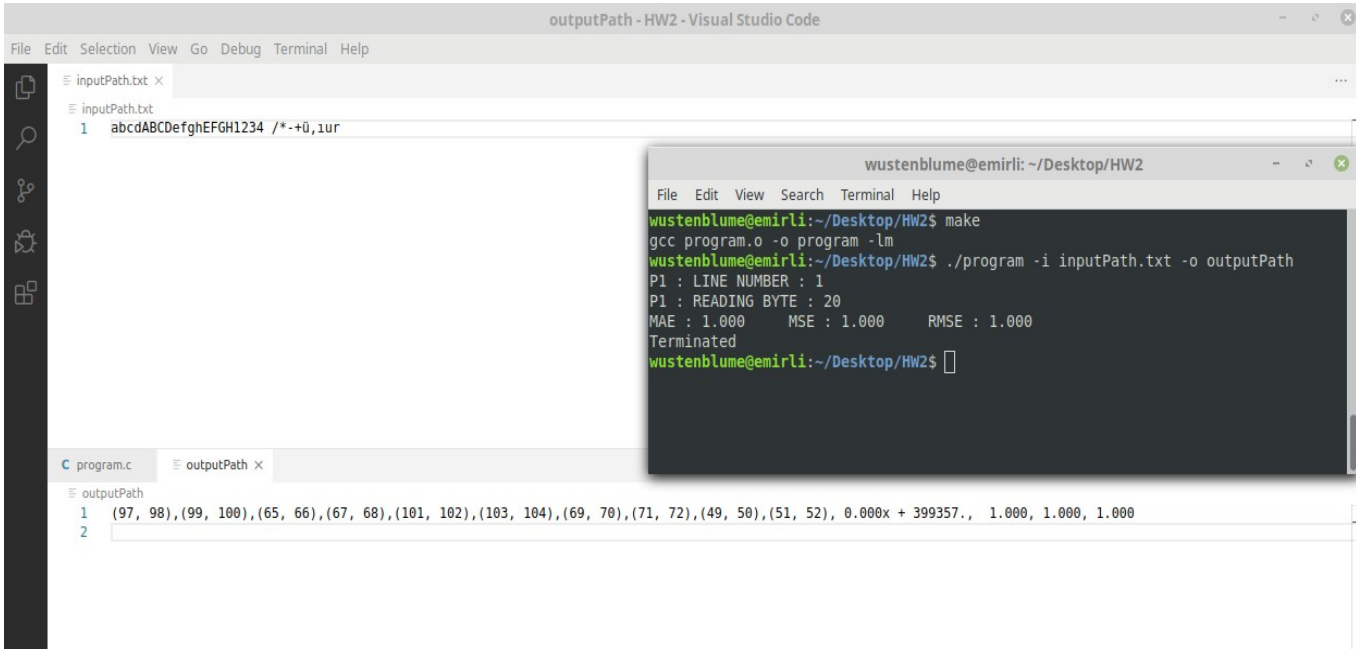
- Less than 20 bytes



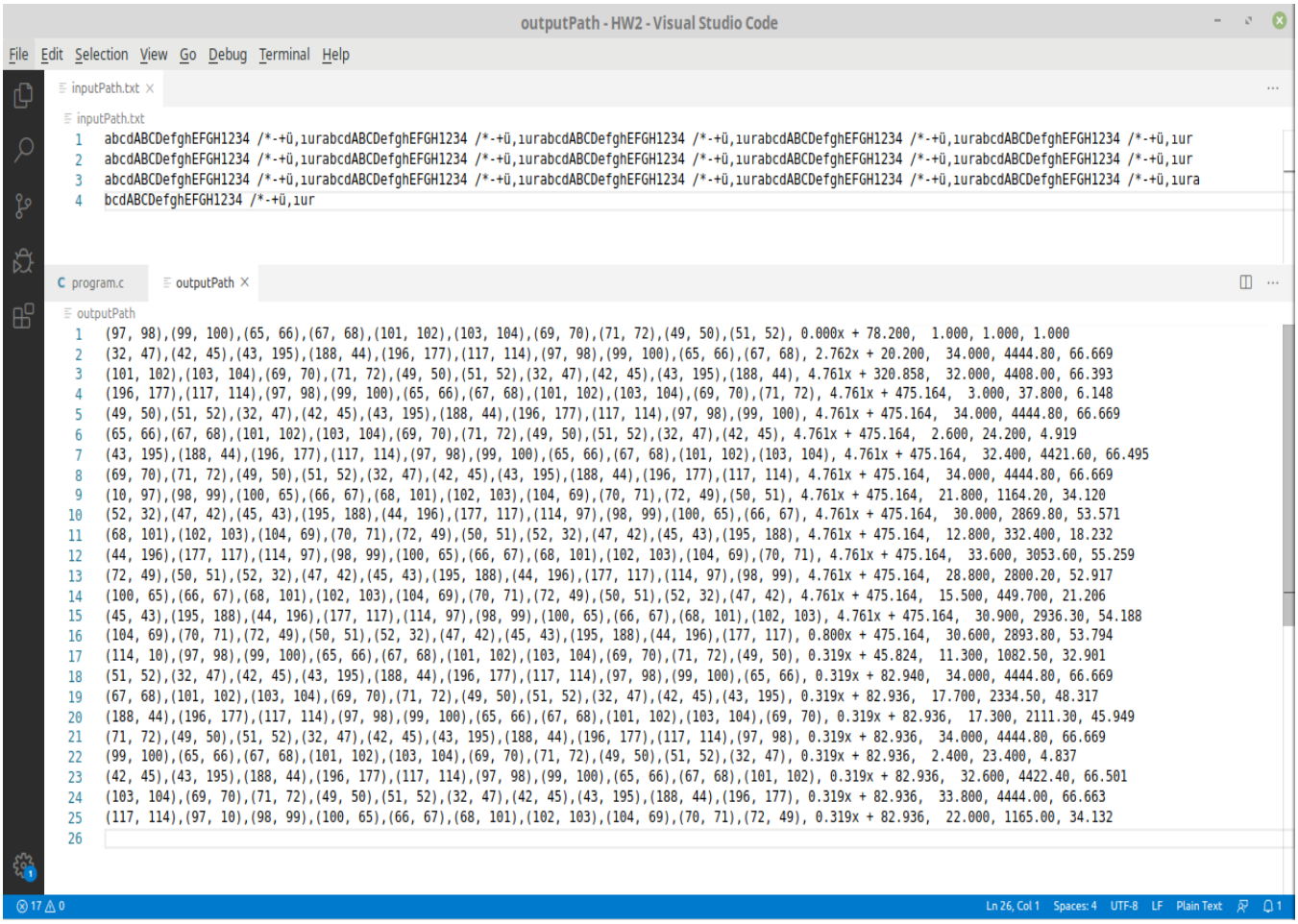
- 20 bytes



- Bigger than 20 bytes, less than 40 bytes



- Much more




```
wustenblume@emirli:~/Desktop/HW2$ make
gcc program.o -o program -lm
wustenblume@emirli:~/Desktop/HW2$ ./program -i inputPath.txt -o outputPath
P1 : LINE NUMBER : 25
P1 : READING BYTE : 500
MAE : 1.000     MSE : 1.000     RMSE : 1.000
MAE : 34.000    MSE : 4444.800  RMSE : 66.669
MAE : 32.000    MSE : 4408.000  RMSE : 66.393
MAE : 3.000     MSE : 37.800    RMSE : 6.148
MAE : 34.000    MSE : 4444.800  RMSE : 66.669
MAE : 2.600     MSE : 24.200    RMSE : 4.919
MAE : 32.400    MSE : 4421.600  RMSE : 66.495
MAE : 34.000    MSE : 4444.800  RMSE : 66.669
MAE : 21.800    MSE : 1164.200  RMSE : 34.120
MAE : 30.000    MSE : 2869.800  RMSE : 53.571
MAE : 12.800    MSE : 332.400   RMSE : 18.232
MAE : 33.600    MSE : 3053.600  RMSE : 55.259
MAE : 28.800    MSE : 2800.200  RMSE : 52.917
MAE : 15.500    MSE : 449.700   RMSE : 21.206
MAE : 30.900    MSE : 2936.300  RMSE : 54.188
MAE : 30.600    MSE : 2893.800  RMSE : 53.794
MAE : 11.300    MSE : 1082.500  RMSE : 32.901
MAE : 34.000    MSE : 4444.800  RMSE : 66.669
MAE : 17.700    MSE : 2334.500  RMSE : 48.317
MAE : 17.300    MSE : 2111.300  RMSE : 45.949
MAE : 34.000    MSE : 4444.800  RMSE : 66.669
MAE : 2.400     MSE : 23.400    RMSE : 4.837
MAE : 32.600    MSE : 4422.400  RMSE : 66.501
MAE : 33.800    MSE : 4444.000  RMSE : 66.663
MAE : 22.000    MSE : 1165.000  RMSE : 34.132
Terminated
wustenblume@emirli:~/Desktop/HW2$
```