Nevra Gürses
161044071

Gebze Technical University

Computer Engineering

CSE344 – System Programming

HOMEWORK #2

REPORT

NEVRA GÜRSES

161044071

Nevra Gürses
161044071

1 INTRODUCTION

1.1 Problem Definition

There are 2 processes that are parent process and child process.This 2 processes must run concurrently.In below I will explain tasks of this 2 processes.

Parent Process (P1): Parent process takes file name in command line argument that is inputpath.It reads every couple unsigned bytes in given file and this couple is converted 2D coordinate (x,y).For every 10 coordinate, least squares method is applied for this 2D coordinates and then line equation (ax +b) is found.This calculation is considered critical section and SIGINT and SIGSTOP signals are not interrupt calculation in this critical section.After every 10 coordinate and line equation calculation ,this coordinates and equation are writed in temporary file that file is created via mkstemp.After finish contents, result of how many bytes was read and how many line equation was estimated and which signals were sent for P1 that is in critical region are printed on screen and open files are closed.

Child Process (P2): Child process read temporary file that is created by parent process.For every line, it calculates mean absolute error, mean squared error and root mean squared error between the coordinates and the estimated line.This calculation is also considered as critical section and SIGINT and SIGSTOP signals are not interrupt calculation in this area.After it reads a line,then removes that line and writes this line in outputpath that is specified in comment line argument as 10 coordinates, the line equation (ax+b) and the three error estimates in a comma separated form.After no more input,p2 terminate and print on screen for each error metric, its mean and standard deviation.After it close all files.

If SIGTERM signal comes in either P1 or P2, processes handle with it and close open files and remove input file and temporary file from disk.

- Important part is that Parent Process and Child process must run concurrently.

2 METHOD

2.1 Problem Solution

Parent Process(P1): For make tasks of parent process, I create a function.In this function,I read input file for every 20 byte .After reading for each 20 byte ,I call another function to make 2D coordinates and apply least sqauares method for finding line equation.In leastSquaresMethod function I create 2D coordinate by couple of unsigned bytes.I recorded this 10 coordinate in string.And with this 2D coordinates,I make least sqaures method and I find line equation.After I find line eqaution,I record this equation in string that also keeps 10 coordinate.This leastsquareMethod calculation must be critical section.So I block SIGINT and SIGSTOP signals by sigprocmask system call and I unblocked this signals at the end of function.And I notice that SIGSTOP signal can not blocked.After,I return string that includes 10 coordinate and line

eqaution.And I write this string in temporary file that is created via mkstemp.I make this operations until end of input file.

Functions that I use for Parent Process(P1) is:

void processP1(int pid,int fdInput,int fdTemp): This function reads input file,calls leastSquaresMethod function and writes result in temporary file.And at the end of this function,it prints how many line was read,how many line equation was estimated and which signals was sent while P1 was in critical section.For provide concurrently working with child process,when first 20 byte is writed in temporary file,SIGUSR2 signal is sent for child process that has pid in function argument.

char* leastSquaresMethod(char* arr,int bytes): In this function, I create 10 coordinate by 20 unsigned byte.I apply leastSquaresMethod and I find line equation.I block SIGINT and SIGSTOP signals while in critical region.At the end of this function I return a string that includes 10 coordinate and line equation.In this function I also handle for all signals.If a signal is sending in critical region,I keep this signal in array by signal handler.

void findingSignalHandler(int no):In this function, I handle signals in critical region of P1.I keep signals that are sent in critical region of P1.

ssize_t readFile(int fd, void *buffer, size_t byteCount): This function reads given file for given byte.It locks file while reading and unlock file at the end of function.

ssize_t writeFile(int fd,char * str): This function writes given string in given file.While it is writing,it locks file and unlock at the end of file.

Child Process(P2): For make tasks of child process,I create function.This function reads every line in given temporary file.After reading every line,It calls calculateErrors functions.In calculateErrors function,according to given line mean absolute error, mean squared error and root mean squared error are calculated.This calculations is considered critical section so,I block SIGINT and SIGSTOP signals and at the end of this function,I unblock this signals.And I return a string from this function that includes 10 coordinate,1 line equation and 3 error calculation.After I write this string in output file and I remove this reading line from temporary file by removeLine function.I make this operations until end of temporary file.

Functions that I use for Child Process(P2) is:

void processP2(int fdTemp2,int fdOutput,char* name):This function reads every line in temporary file.After, it calls calculateErrors function.Calculate errors function returns a string.This function writes this string in output file.And after it calls removeLine function for remove operation for read line.

char* calculateErrors(char *line): This function copies string that is in function argument firstly.That string is a line in temporary file.After it records every coordinate and line equation in an array.After, it calculates errors.Prints this errors,means,and standard deviations on screen.Then,It records this errors in string that is copy of line.And at the end,this function returns string that includes 10 coordinate,1 line equation and 3 error calculations.Error

calculations are considered as critical region so SIGINT and SIGBLOCK signals are blocked and at the and of this function,this signals unblocked.

int removeLine(char* path,int counter,int keepByte,int sizeFile): This function removes given line from file.If I explain function arguments: Keepbyte is how many byte will remove, Counter is starting point of remove operation.Path is file name and sizeFile is size of file.

int findFileSize(int fdInput,char*buf):This function find line size of a file.

ssize_t readFile(int fd, void *buffer, size_t byteCount): This function reads given file for given byte.It locks file while reading and unlock file at the end of function.

ssize_t writeFile(int fd,char * str): This function writes given string in given file.While it is writing,it locks file and unlock at the end of file.

Above functions are parent process and child process functions.For provide parent and child relationship,I make fork in main function.If result of fork is 0 that is child process and if result is not zero that is parent process.In parent process I call processP1 function and For child process I call processP2 function.

For providing concurrently working:

In processP1 function,when first result is wrotten in temporary file,I send SIGUSR2 signal for child process.And also after calling this processP1 function I send SIGUSR1 signal for child process so it can understand parent process is finished and so child process can remove/delete input file.

Handle for SIGUSR1 and SIGUSR2 signals:

void handler(int signum) : This function handle with SIGUSR1 and SIGUSR2 signal.

Handle for SIGTERM signal:

void sigTermHandler(int sigNo): In this function I close all open files and I remove input file and temporary file from disk.

Nevra Gürses
161044071

3 RUNNING RESULTS

Running Result 1:

Input file:

home > nevra > Desktop > hw2 > input.txt

```
d29321983bhsbhdbsfs2d29321983bhsbhdbsfs2
a68dsbf7d98sf23ıc432sahfwğfsya8dhzdw9w7r
sfoıhdsıfh98sadğyfğ8
32324rıuwer3ry23yrr8
893802hewh93278ıewoh
9238euwjds9032eıod93
4u3ejd023opewldşs30e
876e7uı2wehdı93y4rrh0y34reho0823y5rrf008
```

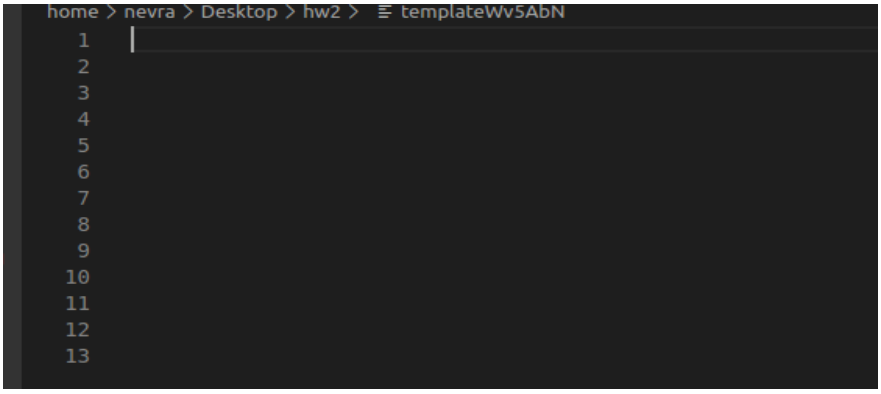
After running program:

Output File:

home > nevra > Desktop > hw2 > output.txt

```
(100,50), (57,51), (50,49), (57,56), (51,98), (104,115), (98,104), (100,98), (115,102), (115,50), 0.372x+45.75, 21.359, 536.268, 23.157
(100,50), (57,51), (50,49), (57,56), (51,98), (104,115), (98,104), (100,98), (115,102), (115,50), 0.372x+45.75, 21.359, 536.268, 23.157
(10,97), (54,56), (100,115), (98,102), (55,100), (57,56), (115,102), (50,51), (196,177), (99,52), 0.540x+45.75, 20.118, 550.432, 23.461
(51,50), (115,97), (104,102), (119,196), (159,102), (115,121), (97,56), (100,104), (122,100), (119,57), 0.632x+28.89, 22.062, 953.061, 30.872
(119,55), (114,10), (115,102), (111,196), (177,104), (100,115), (196,177), (102,104), (57,56), (115,97), 0.590x+30.39, 29.905, 1677.260, 40.954
(100,196), (159,121), (102,196), (159,56), (10,51), (50,51), (50,52), (114,196), (177,117), (119,101), 0.387x+73.43, 45.755, 2747.274, 52.414
(114,51), (114,121), (50,51), (121,114), (114,56), (10,56), (57,51), (56,48), (50,104), (101,119), 0.376x+47.49, 23.880, 670.765, 25.899
(104,57), (51,50), (55,56), (196,177), (101,119), (111,104), (10,57), (50,51), (56,101), (117,119), 0.684x+30.86, 14.894, 313.006, 17.692
(106,100), (115,57), (48,51), (50,101), (196,177), (111,100), (57,51), (10,52), (117,51), (101,106), 0.556x+33.97, 19.776, 570.077, 23.876
(100,48), (50,51), (111,112), (101,119), (108,100), (197,159), (115,51), (48,101), (10,56), (55,54), 0.499x+40.42, 21.019, 587.190, 24.232
(101,55), (117,196), (177,50), (119,101), (104,100), (196,177), (57,51), (121,52), (114,114), (104,48), 0.533x+29.90, 33.890, 1792.107, 42.333
(121,51), (52,114), (101,104), (111,48), (56,50), (51,121), (53,114), (114,102), (48,48), (56,10), -0.082x+82.42, 34.512, 1344.966, 36.674
```

Temporary File After Running Program:



(Note : All lines is deleted while program is running.)

Nevra Gürses
161044071

Terminal:

```
nevra@ubuntu:~/Desktop/hw2$ ./hw2First -i input.txt -o output.txt
~~~~~~~~~~ PROCESS P1 ~~~~~~~~~~
Number of Bytes that is read: 240
Number of estimated line equation: 12
There were not sent signals to P1 while it was critical section.
Handler caught SIGUSR1 signal.

Handler caught SIGUSR2 signal.

~~~~~~~~~~ PROCESS P2 ~~~~~~~~~~
MAE:21.359, MSE:536.268, RMSE:23.157, Mean:193.595, Standard Deviation: 242.307
MAE:21.359, MSE:536.268, RMSE:23.157, Mean:193.595, Standard Deviation: 242.307
MAE:20.118, MSE:550.432, RMSE:23.461, Mean:198.004, Standard Deviation: 249.208
MAE:22.062, MSE:953.061, RMSE:30.872, Mean:335.331, Standard Deviation: 436.815
MAE:29.905, MSE:1677.260, RMSE:40.954, Mean:582.706, Standard Deviation: 773.979
MAE:45.755, MSE:2747.274, RMSE:52.414, Mean:948.481, Standard Deviation: 1271.941
MAE:23.880, MSE:670.765, RMSE:25.899, Mean:240.181, Standard Deviation: 304.470
MAE:14.894, MSE:313.006, RMSE:17.692, Mean:115.197, Standard Deviation: 139.877
MAE:19.776, MSE:570.077, RMSE:23.876, Mean:204.576, Standard Deviation: 258.453
MAE:21.019, MSE:587.190, RMSE:24.232, Mean:210.814, Standard Deviation: 266.142
MAE:33.890, MSE:1792.107, RMSE:42.333, Mean:622.777, Standard Deviation: 826.849
MAE:34.512, MSE:1344.966, RMSE:36.674, Mean:472.051, Standard Deviation: 617.245
nevra@ubuntu:~/Desktop/hw2$
```

Running Result 2:

Input File:

```
1  87654323456776544567
2  qt87eqwteglbad$1wqge
3  2192uw89smo8923uejd1
4  2873y4e72d78e2382132
5  132511243621534651 23
6  456782131241246214 23
7  254326546324526344 32
8  abagsvagdvtasdasydva
```

Nevra Gürses
161044071

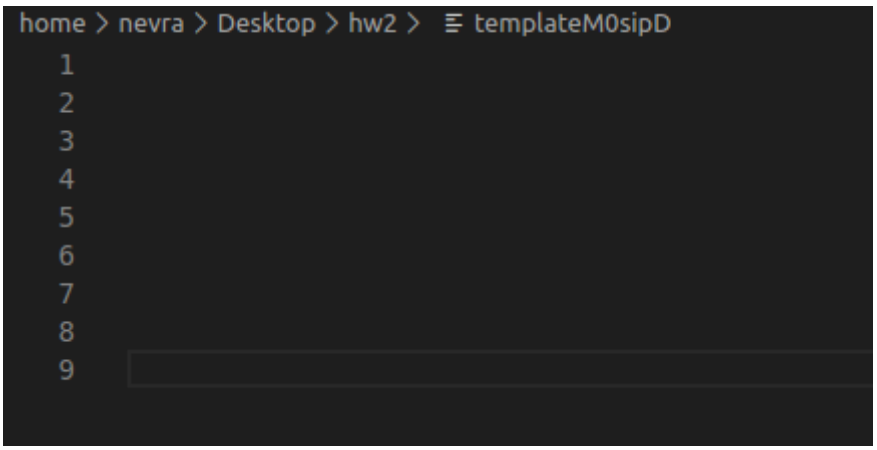
After running program:

Output File:

home > nevra > Desktop > hw2 > output.txt

```
1  (56,55), (54,53), (52,51), (50,51), (52,53), (54,55), (55,54), (53,52), (52,53), (54,55), 0.710x+15.42, 0.626, 0.511, 0.715
2  (10,113), (116,56), (55,101), (113,119), (116,101), (103,108), (98,97), (100,197), (159,196), (177,119), 0.226x+97.00, 30.048, 1542.928, 39.280
3  (113,103), (101,10), (50,49), (57,50), (117,119), (56,57), (115,109), (111,56), (57,50), (51,117), 0.360x+42.20, 25.721, 1000.419, 31.629
4  (101,106), (100,196), (177,10), (50,56), (55,51), (121,52), (101,55), (50,100), (55,56), (101,50), -0.275x+98.27, 34.001, 2060.486, 45.393
5  (51,56), (50,49), (51,50), (10,49), (51,50), (53,49), (49,50), (52,51), (54,50), (49,53), 0.041x+48.78, 1.492, 4.146, 2.036
6  (51,52), (54,53), (49,50), (51,10), (52,53), (54,55), (56,50), (49,51), (49,50), (52,49), 0.823x+4.72, 5.811, 91.859, 9.584
7  (50,52), (54,50), (49,52), (50,51), (10,50), (53,52), (51,50), (54,53), (52,54), (51,50), 0.038x+49.58, 1.046, 1.607, 1.268
8  (52,53), (50,54), (51,52), (52,51), (50,10), (97,98), (97,103), (115,118), (97,103), (100,118), 1.265x+-20.23, 5.529, 60.236, 7.761
9
```

Temporary File After Running Program:



Terminal:

```
nevra@ubuntu:~/Desktop/hw2$ ./hw2First -i input.txt -o output.txt
~~~~~~~~~~ PROCESS P1 ~~~~~~~~~~
Number of Bytes that is read: 170
Number of estimated line equation: 8
There were not sent signals to P1 while it was critical section.
Handler caught SIGUSR1 signal.

Handler caught SIGUSR2 signal.

~~~~~~~~~~ PROCESS P2 ~~~~~~~~~~
MAE:0.626, MSE:0.511, RMSE:0.715, Mean:0.617, Standard Deviation: 0.083
MAE:30.048, MSE:1542.928, RMSE:39.280, Mean:537.418, Standard Deviation: 711.012
MAE:25.721, MSE:1000.419, RMSE:31.629, Mean:352.590, Standard Deviation: 458.091
MAE:34.001, MSE:2060.486, RMSE:45.393, Mean:713.293, Standard Deviation: 952.621
MAE:1.492, MSE:4.146, RMSE:2.036, Mean:2.558, Standard Deviation: 1.145
MAE:5.811, MSE:91.859, RMSE:9.584, Mean:35.751, Standard Deviation: 39.704
MAE:1.046, MSE:1.607, RMSE:1.268, Mean:1.307, Standard Deviation: 0.231
MAE:5.529, MSE:60.236, RMSE:7.761, Mean:24.508, Standard Deviation: 25.279
nevra@ubuntu:~/Desktop/hw2$
```

Nevra Gürses
161044071

Running Result 3:

Input File:

```
1  01111111111111100000
2  00101001010100101010100101001010100101010101
3  01010101001111010101010
4  01010100101010101001100101001010100101010101
5  01010101001111010101001010100101010101010011
6  01010101001000010101010
7  01010100101010101010011
8  01010101001111010101010
9
```

After Running Program:

Output File:

```
home > nevra > Desktop > hw2 > output.txt
1  (48,49), (49,49), (49,49), (49,49), (49,49), (49,49), (49,49), (49,49), (48,48), (48,48), 0.667x+16.33, 0.122, 0.046, 0.215
2  (48,10), (48,48), (49,48), (49,48), (48,49), (48,49), (48,49), (48,48), (49,48), (49,48), 5.833x+-237.83, 1.089, 3.549, 1.884
3  (49,48), (49,48), (48,49), (48,49), (48,48), (49,48), (49,48), (49,48), (48,49), (48,49), -0.800x+87.20, 0.125, 0.049, 0.221
4  (48,49), (48,49), (10,48), (49,48), (49,48), (49,48), (49,48), (48,49), (49,49), (49,48), 0.010x+47.94, 0.450, 0.226, 0.475
5  (49,48), (49,48), (49,48), (10,48), (49,48), (49,48), (49,48), (48,49), (48,49), (48,49), 0.007x+47.98, 0.404, 0.203, 0.451
6  (48,49), (48,49), (48,48), (49,49), (48,48), (49,48), (49,48), (48,49), (48,49), (48,49), -0.381x+67.00, 0.392, 0.183, 0.428
7  (48,48), (49,48), (49,48), (49,48), (49,10), (48,49), (48,49), (48,49), (48,49), (48,48), -10.167x+536.66, 0.585, 1.039, 1.019
8  (49,49), (49,49), (48,49), (48,49), (48,49), (48,48), (49,48), (49,48), (49,48), (48,49), -0.400x+68.00, 0.371, 0.172, 0.415
9  (48,49), (48,49), (48,49), (48,49), (48,48), (49,49), (10,48), (49,48), (49,48), (49,48), 0.014x+47.89, 0.450, 0.226, 0.476
10 (49,48), (48,49), (48,48), (48,48), (49,48), (49,48), (49,48), (10,48), (49,48), (49,48), 0.002x+47.99, 0.166, 0.090, 0.299
11 (49,48), (48,49), (48,49), (48,49), (48,49), (48,49), (48,48), (49,49), (10,48), (49,48), 0.017x+47.86, 0.402, 0.204, 0.451
12 (49,48), (49,48), (49,48), (48,49), (49,49), (49,48), (49,48), (49,48), (49,48), (10,115), -1.717x+132.14, 0.093, 0.038, 0.195
13
```

Temporary File After Running Program:

```
home > nevra > Desktop > hw2 > template6mBlNW
1
2
3
4
5
6
7
8
9
10
11
12
13
```

Nevra Gürses
161044071

Terminal:

```
nevra@ubuntu:~/Desktop/hw2$ ./hw2First -i input.txt -o output.txt
~~~~~~~~~~ PROCESS P1 ~~~~~~~~~~
Number of Bytes that is read: 240
Number of estimated line equation: 12
There were not sent signals to P1 while it was critical section.
Handler caught SIGUSR1 signal.

Handler caught SIGUSR2 signal.

~~~~~~~~~~ PROCESS P2 ~~~~~~~~~~
MAE:0.122, MSE:0.046, RMSE:0.215, Mean:0.128, Standard Deviation: 0.069
MAE:1.089, MSE:3.549, RMSE:1.884, Mean:2.174, Standard Deviation: 1.025
MAE:0.125, MSE:0.049, RMSE:0.221, Mean:0.132, Standard Deviation: 0.070
MAE:0.450, MSE:0.226, RMSE:0.475, Mean:0.384, Standard Deviation: 0.112
MAE:0.404, MSE:0.203, RMSE:0.451, Mean:0.353, Standard Deviation: 0.107
MAE:0.392, MSE:0.183, RMSE:0.428, Mean:0.334, Standard Deviation: 0.108
MAE:0.585, MSE:1.039, RMSE:1.019, Mean:0.881, Standard Deviation: 0.210
MAE:0.371, MSE:0.172, RMSE:0.415, Mean:0.320, Standard Deviation: 0.106
MAE:0.450, MSE:0.226, RMSE:0.476, Mean:0.384, Standard Deviation: 0.112
MAE:0.166, MSE:0.090, RMSE:0.299, Mean:0.185, Standard Deviation: 0.087
MAE:0.402, MSE:0.204, RMSE:0.451, Mean:0.352, Standard Deviation: 0.107
MAE:0.093, MSE:0.038, RMSE:0.195, Mean:0.109, Standard Deviation: 0.065
```