CSE344 – System Programming - Homework #2
REPORT

I created 2 processes running simultaneously with fork. If pid is not 0,them p1 is running .Else p2 is running.I locked for each writing and reading.I opened the input file in P1 and found each line to the coordinates and the equation, and wrote it in the temp file I created with mkstemp.I created a mask in the critical section and caught the SIGINT signal but the SIGSTOP signal could not be blocked.I kept which signals were send with sigpending.P1 calls the SIGUSR2 signal when every line writes the temp file. In the handler I wrote, I checked if the signal was called with the global variable.After writing to the temp file, I closed the input and temp file.I sent the SIGUSR1 signal that P1 finished the input file. A global variable changes in handler and it is understood that the file writing process of P1 is finished.Then the number of bytes read by P1, the number of equations it calculates and the signals sent in the critical section are printed on the screen.After P1 is finished, P2 is a zombie process, and since this prevents the program from terminating, so I used wait at the end of P1. At the end, all the work of P1 is finished by calling the SIGTERM signal and writing a message on the screen with my own handler.

In P2, the temp file opens in both reading and writing mode. Then a line is read from the temp file.If the file is empty and P1 is not terminated, it is expected that P1 will send a signal with sigsuspend.If the file is empty and P1 finished writing to the file, P2 stops reading the file and continues its work.If the file is not empty, a line is read and with the lseek, the beginning of the file.When each line is read, the sizes of these lines are collected and when each line is read, the array that it has null character as much as the value collected is written temp file.This ensures that the line read from the temp file is deleted.While this is happening, the line read has mae, mse and rmse and these values are kept in 3 different arrays (to use when finding the mean and standard deviation later).It is written to the new line output file containing extra the new mae, mse and rmse.After reading the entire contents of the temp file, P2 finds the mean and standard deviation of the errors it reads and prints them on the screen.Then, temp and output files are closed, pointers are free and input file and temp file are removed with unlink.At the end, all the work of P2 is finished by calling the SIGTERM signal and writing a message on the screen with my own handler.

OUTPUTS

Input includes 280 bytes
DFSDsd2f3l4hk7
y05/*s/nv-*
*lsdv6v3
rwfg7)63342!^%^
dg64
8
/
//*9/8908i
tygjhlf+56t+gb
^+
DFSDsd2f3l4hk7
y05/*s/nv-*
*lsdv6v3
rwfg7)63342!^%^
dg64
8
/
//*9/8908i
tygjhlf+56t+gb
^+
DFSDsd2f3l4hk7
y05/*s/nv-*
*lsdv6v3
rwfg7)63342!^%^
dg64
8
/
//*9/8908i
tygjhlf+56t+gb
^+
DFSDsd2f3l4hk7
y05/*s/nv-*
*lsdv6v3
rwfg7)63342!^%^
dg64
8

Terminal Output is

```
zeynep@zeynep-VirtualBox:~/Desktop$ make clean
rm -rf *.o program
zeynep@zeynep-VirtualBox:~/Desktop$ make all
gcc -c program.c
gcc program.o -lm -o program
zeynep@zeynep-VirtualBox:~/Desktop$ ./program -i  inputPath -o outputPath
file : /tmp/tempzBe0r0
P1 read 280 bytes
P1 estimated 14 line equations
No signal sent in the critical section in P1
Mean for MAE : 24.162
Mean for MSE : 713.512
Mean for RMSE : 26.234
Standard derivation for MAE : 5.971
Standard derivation for MSE : 273.814
Standard derivation for RMSE : 5.218
Sinyal SIGTERM is caught 15.
Sinyal SIGTERM is caught 15.
zeynep@zeynep-VirtualBox:~/Desktop$
```

Output File is

```
(68, 70), (83, 68), (115, 100), (50, 102), (51, 108), (52, 104), (107, 55), (121, 48), (53, 47), (42, 115), -0.452x+115.273, 18.586,
475.370, 21.803
(47, 110), (118, 45), (42, 42), (108, 115), (100, 118), (54, 118), (51, 114), (119, 102), (103, 55), (41, 54), 0.030x+84.917, 30.592,
1008.064, 31.750
(51, 51), (52, 50), (33, 94), (37, 94), (100, 103), (54, 52), (56, 47), (47, 47), (42, 57), (47, 56), 0.317x+48.668, 18.241, 420.208,
20.499
(57, 48), (56, 105), (116, 121), (103, 106), (104, 108), (102, 43), (53, 54), (116, 43), (103, 98), (94, 43), 0.293x+50.437, 28.942,
931.602, 30.522
(68, 70), (83, 68), (115, 100), (50, 102), (51, 108), (52, 104), (107, 55), (121, 48), (53, 47), (42, 115), -0.452x+115.273, 18.586,
475.370, 21.803
(47, 110), (118, 45), (42, 42), (108, 115), (100, 118), (54, 118), (51, 114), (119, 102), (103, 55), (41, 54), 0.030x+84.917, 30.592,
1008.064, 31.750
(51, 51), (52, 50), (33, 94), (37, 94), (100, 103), (54, 52), (56, 47), (47, 47), (42, 57), (47, 56), 0.317x+48.668, 18.241, 420.208,
20.499
(57, 48), (56, 105), (116, 121), (103, 106), (104, 108), (102, 43), (53, 54), (116, 43), (103, 98), (94, 43), 0.293x+50.437, 28.942,
931.602, 30.522
(68, 70), (83, 68), (115, 100), (50, 102), (51, 108), (52, 104), (107, 55), (121, 48), (53, 47), (42, 115), -0.452x+115.273, 18.586,
475.370, 21.803
(47, 110), (118, 45), (42, 42), (108, 115), (100, 118), (54, 118), (51, 114), (119, 102), (103, 55), (41, 54), 0.030x+84.917, 30.592,
1008.064, 31.750
(51, 51), (52, 50), (33, 94), (37, 94), (100, 103), (54, 52), (56, 47), (47, 47), (42, 57), (47, 56), 0.317x+48.668, 18.241, 420.208,
20.499
(57, 48), (56, 105), (116, 121), (103, 106), (104, 108), (102, 43), (53, 54), (116, 43), (103, 98), (94, 43), 0.293x+50.437, 28.942,
931.602, 30.522
(68, 70), (83, 68), (115, 100), (50, 102), (51, 108), (52, 104), (107, 55), (121, 48), (53, 47), (42, 115), -0.452x+115.273, 18.586,
475.370, 21.803
(47, 110), (118, 45), (42, 42), (108, 115), (100, 118), (54, 118), (51, 114), (119, 102), (103, 55), (41, 54), 0.030x+84.917, 30.592,
1008.064, 31.750
```