SYSTEM PROGRAMMING – HW_3 – REPORT

YAĞIZ DÖNER – 141044062

Firstly, the command line parsed to the variable with getopt(). Then, I read the inputs and wrote them to allocated 2D arrays as inputA and inputB. Open files were closed and then I created quarter 2D arrays as firstA, firstB, secandA, etc... My inputA and inputB were parsed to these quarters. If there is a sufficient character in the file, program exits gracefully. For this, I checked all the inputs is between 33 and 126. Because these values are valid according to ASCII Table .At the end, input A and B were deallocated with free().

I created pipes between P1 and its children P2, P3, P4, P5 and then fork().

P1 wrote the quarters to the pipes and children waited until their pipe was ready.

Parent process waited its children as a synchronous barrier.

And then, parent read the pipes and fill the quarters and then combine them to the res 2D arrays. (All free() operations was made explicitly.)

For the singular value decomposition, I found a code from web. Because, in the homework pdf, it told us to copy and paste external resources.

I use this : http://www.mymathlib.com/matrices/linearsystems/singular_value.html

Parent process wrote the result matrix to the terminal and then calculate svd. The svd result was written to the terminal by the parent. The svd result may be uncorrect. Because, I use external code as I said before.

General topics of homework like this.

Also, I wrote signal handlers for SIGCHLD and SIGINT. Both of them works well. In terms of SIGINT program exits gracefully with the help of exit(). Because exit function close all of the open files and free all allocated memory. On the other end, in terms of SIGCHLD, parent waits its children synchronously.

For the signals, I wrote sleep() functions as a comment in the code. In the demo, I use them for how to signal works.

Sample Output like these…

its shows to getopt() -->

Normal Input :

Signal Handling :

SIGCHLD   SIGINT

No ZOMBIES .

YAĞIZ DÖNER - 141044062