CSE344 System Programming HW3 Report

Program

In program, it reads the "input1" and "input2" file in P1(parent process) and converts it 2 matrixes. It divides the converted matrix into 4 separate equal parts. It sends these divided matrices to the children through the pipe for the children to read with 4 write ends of 8 pipes in total. After that, the children created by the parent process with 4 forks read the data from the read ends and multiply the 2 separate matrix pieces in the data they read, obtain a quarter of the new matrix and then write them to the writing end of the pipe. And finally, the parent process merges the quarters of the main matrix written by children, and calculates singular values and prints on the screen.

Program's functions

read_lock(...): It runs like a normal read system call but this function also provides read lock and unlock by using flock structure. So I use this function everywhere instead of read.

dmatrix(…): Allocate a double matrix with subscript range m[nrl..nrh][ncl..nch].

dvector(…): Allocate a double vector with subscript range v[nl..nh].

free_dvector(…): Free a double vector allocated with dvector().

pythag(…): Compute $(a2 + b2)^{1/2}$ without destructive underflow or overflow.

svdcmp(…): Given a matrix a[1..m][1..n], this routine computes its singular value decomposition, A = U.W.VT. The matrix U replaces a on output. The diagonal matrix of singular values W is output as a vector w[1..n]. The matrix V (not the transpose VT) is output as v[1..n][1..n].

matrix_multiplication(…): Multiplicates the [row1]x[col1] matrix1 and [row2]x[col2] matrix2 then writes the given pipe.

char_to_int_array(…): Converts the string(char array) to 2d int array by separating tokens using strtok.

merge_quarters(…): Merges 4 quarter matrix into one big matrix(Matrix C).

read_input(…): Read [2^n]*[2^n] input in fd and converts 2d [2^n]X[2^n] integer array then returns it.

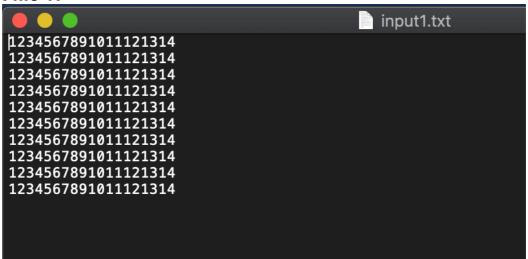**find_quarter(…):** Finds and returns the desired quarter in the given matrix.

sigint_handler(…): SIGINT handler function.

sigchld_handler(…): SIGCHLD handler function.

main(…): In the beginning, evaluates command line parameters and checks for their suitability. Then goes to the main processes: Firstly, creates the 4 children by using for loop in fork(). In for loop all children take the quarters of the 2 different matrices sent by the parent and multiply the result and send the parent again. After the for loop the parent reads the two different (or same) file and writes the pipe to send children. After sent waits children synchronously. After children finished. Collect the wrote data and merge them and calculates singular values and prints on the screen.

Sample Execution Screenshots

Input file 1:

Input file 2:

Outputs:


```
→  SYS-HW3-SON ./program -i input1.txt -j input2.txt -n 1
input1.txt input2.txt 1
Parent/Process 1 (PID=35391) started

**Matrix A**
49 50
51 52

**Matrix B**
10 53
54 55

Child 3/Process 4 (PID=35394) closing pipe
Child 1/Process 2 (PID=35392) closing pipe
Child 4/Process 5 (PID=35395) closing pipe
Child 2/Process 3 (PID=35393) closing pipe
SIGCHLD caught
Child (PID=35394) waited.
Child (PID=35392) waited.
SIGCHLD caught
Child (PID=35395) waited.
SIGCHLD caught
Child (PID=35393) waited.

**Matrix C**
3190 5347
3318 5563

**Singular Values**
0.515 8984.576
→  SYS-HW3-SON
```

Figure 1-(n=1)


```
→  SYS-HW3-SON ./program -i input1.txt -j input2.txt -n 2
input1.txt input2.txt 2
Parent/Process 1 (PID=36230) started

**Matrix A**
49 50 51 52
53 54 55 56
57 49 48 49
49 49 50 49

**Matrix B**
10 53 54 55
56 57 49 48
49 49 49 50
49 51 49 52

Child 1/Process 2 (PID=36231) closing pipe
Child 2/Process 3 (PID=36232) closing pipe
Child 4/Process 5 (PID=36234) closing pipe
Child 3/Process 4 (PID=36233) closing pipe
SIGCHLD caught
Child (PID=36232) waited.
Child (PID=36231) waited.
SIGCHLD caught
SIGCHLD caught
Child (PID=36234) waited.
Child (PID=36233) waited.

**Matrix C**
8337 10598 10143 10349
8993 11438 10947 11169
8067 10665 10232 10435
8085 10339 9898 10095

**Singular Values**
40156.813 253.758 0.105 2.543
→  SYS-HW3-SON
```

Figure 2- (n=2)

```
[→  SYS-HW3-SON ./program -i input1.txt -j input2.txt -n 3
input1.txt input2.txt 3
Parent/Process 1 (PID=36823) started

##Matrix A##
49 50 51 52 53 54 55 56
57 49 48 49 49 49 50 49
51 49 52 10 49 50 51 52
53 54 55 56 57 49 48 49
49 49 50 49 51 49 52 10
49 50 51 52 53 54 55 56
57 49 48 49 49 49 50 49
51 49 52 10 49 50 51 52

##Matrix B##
10 53 54 55 56 57 49 48
49 49 49 50 49 51 49 52
49 53 49 54 49 55 10 53
54 55 56 57 49 48 49 49
49 50 49 51 49 52 49 53
49 54 49 55 10 53 54 55
56 57 49 48 49 49 49 50
49 51 49 52 49 53 49 54

Child 4/Process 5 (PID=36827) closing pipe
Child 1/Process 2 (PID=36824) closing pipe
Child 2/Process 3 (PID=36825) closing pipe
SIGCHLD caught
Child (PID=36827) waited.
SIGCHLD caught
Child (PID=36825) waited.
Child 3/Process 4 (PID=36826) closing pipe
SIGCHLD caught
Child (PID=36824) waited.
SIGCHLD caught
Child (PID=36826) waited.

##Matrix C##
19314 22167 21189 22138 18817 21925 18861 21756
17972 21106 20228 21112 18088 20932 17973 20667
16254 19119 18161 19034 16243 19199 16058 18947
19178 22181 21286 22231 19089 21998 18729 21773
16289 19013 18179 18950 16023 18721 15886 18489
19314 22167 21189 22138 18817 21925 18861 21756
17972 21106 20228 21112 18088 20932 17973 20667
16254 19119 18161 19034 16243 19199 16058 18947

##Singular Values##
156051.032 634.321 419.943 0.000 195.174 253.867 0.000 0.000
→   SYS-HW3-SON
```

Figure 3- (n=3)