

CSE 344
SYSTEMS PROGRAMMING

HOMEWORK 03

REPORT

ÖMER ÇEVİK
161044004

1. Part 1

In that part, in program.c starts to read inputPathA.txt and inputPathB.txt as parent process **P1**. At the same time fork() syscall in main creates 4 children process which runs the **readPipe()**.

P1 process reads input files ($2^n \times 2^n$) characters which n is given by user. Then **P1** checks the read size if it is not equal to ($2^n \times 2^n$) then print to screen error message and exits. If everything is alright then prints to screen read input matrixes of A and B. Then parses each matrix to quarter parts of them. After parsing operation, **P1** continues to write pipes of children quarters of A and B. Then with **sigprocmask()** masking the **SIGCHLD** signal. And call the **setMatrixC()** function to set evaluated matrix to C matrix. In that function, parent waits the children to finish their jobs. Then continues to evaluate singular values of C matrix firstly creates the characteristic equation of matrix C using Faddeev–LeVerrier algorithm and finding lambdas using **GSL** library which is installed to gcc. GSL library functions evaluate the lambdas and **P1** finishes its job printing all singular values to screen as complex numbers which has real and imaginary parts and frees all allocated resources. Exits successfully.

Children processes reads pipes for each pipe read that the quarter parts of parent process. All pipes are unique for each child process. Children processes reads pipe which includes matrix A and matrix B quarters. Then evaluates matrix multiplications and write back to same pipe to make the pipe **bi-directional**. After they finish their jobs, frees all allocated resources and send the **SIGCHLD** signal to parent process to continue and exits successfully.

All processes handle the **SIGINT** signal to exit using **CTRL-C** user attempt with **sigaction()** syscall and sigaction struct.

inputPathA.txt:

program.c	inputPathA.txt	inputPathB.txt
1	2435507989243657	

inputPathB.txt:

program.c	inputPathA.txt	inputPathB.txt
1	4494872934496716	

Makefile:

program.c	inputPathA.txt	inputPathB.txt	Makefile
<pre>1 CC = gcc 2 CFLAGS = -c 3 LIBC = -Wall -I/usr/local/include 4 LINKC = -L/usr/local/lib 5 MATH = -lgsl -lgslcblas -lm 6 GFLAG = -g 7 VALG = valgrind -v 8 9 all:main run 10 11 main: program.o 12 \$(CC) \$(GFLAG) \$(LINKC) program.o \$(MATH) -o program 13 14 program.o: program.c 15 \$(CC) \$(GFLAG) \$(LIBC) \$(CFLAGS) program.c 16 17 run: 18 ./program -i inputPathA.txt -j inputPathB.txt -n 2 19 20 v:main runv 21 22 runv: 23 \$(VALG) ./program -i inputPathA.txt -j inputPathB.txt -n 2 24 25 clean: 26 rm -rf *.o program 27</pre>			

Console:

Make:

```
omer@omer:~/Desktop/HW03/HW03$ make
gcc -g -Wall -I/usr/local/include -c program.c
gcc -g -L/usr/local/lib program.o -lgsl -lgslcblas -lm -o program
./program -i inputPathA.txt -j inputPathB.txt -n 2
A : 2435507989243657
B : 4494872934496716

Matrix A :
50 52 51 53
53 48 55 57
56 57 50 52
51 54 53 55

Matrix B :
52 52 57 52
56 55 50 57
51 52 52 57
54 55 49 54

Child PID : 2541, Parent PID : 2540
Child PID : 2543, Parent PID : 2540
Child PID : 2542, Parent PID : 2540
Child PID : 2544, Parent PID : 2540
Matrix C :
10975 11027 10699 11333
11327 11391 11074 11705
11462 11507 11190 11819
11349 11403 11058 11721

Singular value [0] = 0.805 + 0.000i
Singular value [1] = 1.975 + 0.000i
Singular value [2] = 3.783 + 0.000i
Singular value [3] = 212.740 + 0.000i
omer@omer:~/Desktop/HW03/HW03$ _
```

Make with Valgrind:

```
==2558== /usr/lib/valgrind/../../bin/vgdb --pid=2558 ...command...
==2558==
==2558== TO DEBUG THIS PROCESS USING GDB: start GDB like this
==2558== /path/to/gdb ./program
==2558== and then give GDB the following command
==2558== target remote | /usr/lib/valgrind/../../bin/vgdb --pid=2558
==2558== --pid is optional if only one valgrind process is running
==2558==
==2558==
==2558== HEAP SUMMARY:
==2558==   in use at exit: 0 bytes in 0 blocks
==2558== total heap usage: 15 allocs, 15 frees, 160 bytes allocated
==2558==
==2558== All heap blocks were freed -- no leaks are possible
==2558==
==2558== ERROR SUMMARY: 0 errors from 0 contexts (suppressed: 0 from 0)
==2558== ERROR SUMMARY: 0 errors from 0 contexts (suppressed: 0 from 0)
--2556-- REDIR: 0x5a09ab0 (libc.so.6:__memcpy_avx_unaligned_erms) redirected to
0x4c37130 (memcpy)
Singular value [0] = 0.805 + 0.000i
Singular value [1] = 1.975 + 0.000i
Singular value [2] = 3.783 + 0.000i
--2556-- REDIR: 0x5a09ad0 (libc.so.6:__memcpy_avx_unaligned_erms) redirected to
0x4c366e0 (memmove)
Singular value [3] = 212.740 + 0.000i
==2556==
==2556== HEAP SUMMARY:
==2556==   in use at exit: 0 bytes in 0 blocks
==2556== total heap usage: 51 allocs, 51 frees, 880 bytes allocated
==2556==
==2556== All heap blocks were freed -- no leaks are possible
==2556==
==2556== ERROR SUMMARY: 0 errors from 0 contexts (suppressed: 0 from 0)
==2556== ERROR SUMMARY: 0 errors from 0 contexts (suppressed: 0 from 0)
omer@omer:~/Desktop/HW03/HW03$ _
```

CTRL-C Handler:

```
omer@omer:~/Desktop/HW03/HW03$ make
gcc -g -Wall -I/usr/local/include -c program.c
gcc -g -L/usr/local/lib program.o -lgsl -lgslcblas -lm -o program
./program -i inputPathA.txt -j inputPathB.txt -n 2
A : 2435507989243657
B : 4494872934496716

Matrix A :
50 52 51 53
53 48 55 57
56 57 50 52
51 54 53 55

Matrix B :
52 52 57 52
56 55 50 57
51 52 52 57
54 55 49 54

Child PID : 2603, Parent PID : 2602
Child PID : 2604, Parent PID : 2602
Child PID : 2606, Parent PID : 2602
Child PID : 2605, Parent PID : 2602
Matrix C :
10975 11027 10699 11333
11327 11391 11074 11705
11462 11507 11190 11819
11349 11403 11058 11721

Singular value [0] = 0.805 + 0.000i
Singular value [1] = 1.975 + 0.000i
Singular value [2] = 3.783 + 0.000i
Singular value [3] = 212.740 + 0.000i
^C
CTRL-C Signal is caught!

omer@omer:~/Desktop/HW03/HW03$ _
```

Notes:

- To run the program, I write a Makefile which exactly compiles and runs **program.c** using “**make**” command. Input A file name is “**inputPathA.txt**” and input B file name is “**inputPathB.txt**”.
- I send my input files and Makefile in .zip file.
- No errors, no warnings I caught showed in test results.
- No zombie processes.
- **CTRL – C (SIGINT)** signal is handled successfully.
- **SIGCHLD** signal is used to communicate between parent and its children processes.
- To run with “**valgrind**” I create the Makefile to run in valgrind mode using “**make v**” command. It compiles and runs.
- I used “**GSL**” mathematic library to evaluate singular values of matrix C. If you don’t have these library follow the install command above this line:

sudo apt-get install libgsl-dev

- You have to add flags to compile and run program. In my Makefile, they have already added to show up the compile and link rule is below.

gcc -Wall -I/usr/local/include -c program.c
gcc -L/usr/local/lib program.o -lgsl -lgslcblas -lm

- I used “**Faddeev–LeVerrier algorithm**” to create characteristic equation of matrix C to evaluate singular values.