System Programming Homework 2

Muhammet Fikret ATAR

1801042693

Problem Solving and Design Decisions:

->Received arguments(inputFilePath, outputFilePath) using get opt and suppressed to terminal if there is an error.
->File operations are done with system call.

->Consecutively read 3 bytes are interpreted as integer coordinates in a 3-dimensional space.

->Child file is created for exeve operations. In this file, child processes will perform their operations.

->Args and environment variable(consecutively read 3 bytes are interpreted as integer coordinates in a 3-dimensional) are forwarded with child process exec. Apart from these, the output fille path is also sent to the Child file.

->Fork+exec paradigms applied while reading as 30 bits during file read operation. In this way, memory usage is avoided. As a result, the buufer is sent to the child file with execve every 30 bits read.

->Whenever a new buffer is created during file reading, fork creates new child process and delivers the read the buffer to the created childs.

->Run child.c file with exeve.

->Child processes that make the covariance calculation collect their results in double type 10*3 array in child.c file.

->In this point , the children processes must not access the output file simultaneously for writing, the fcnt method comes into play and prevents write operations in the same time to file.

->The process P must not start collecting the outputs before all the children processes have

finished their calculations. To solve this problem, wait call is used and parent waits for children's operations.

->After these operations, parent process reaches the matrices from the file with the name we take as an argument .

->The norm of the coordinates is calculated in The Frobenius norm(The Frobenius norm is an extension of the Euclidean norm to. and comes from the Frobenius inner product on the space of all matrices.by wiki)

->The matrices whose norms are closest to each other are detected and printed to the terminal with stdout.

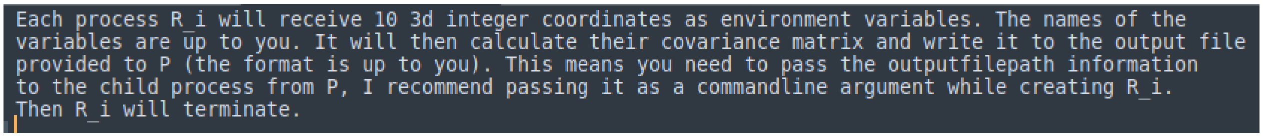
->Finally, If P received SIGINT, all conditions(free all of its resources, close open files, and remove the output file, clean-up after its children and terminate) are done successfully.

Which requirements i achieved and which i have failed:

- This program works all the cases. I have not failed.

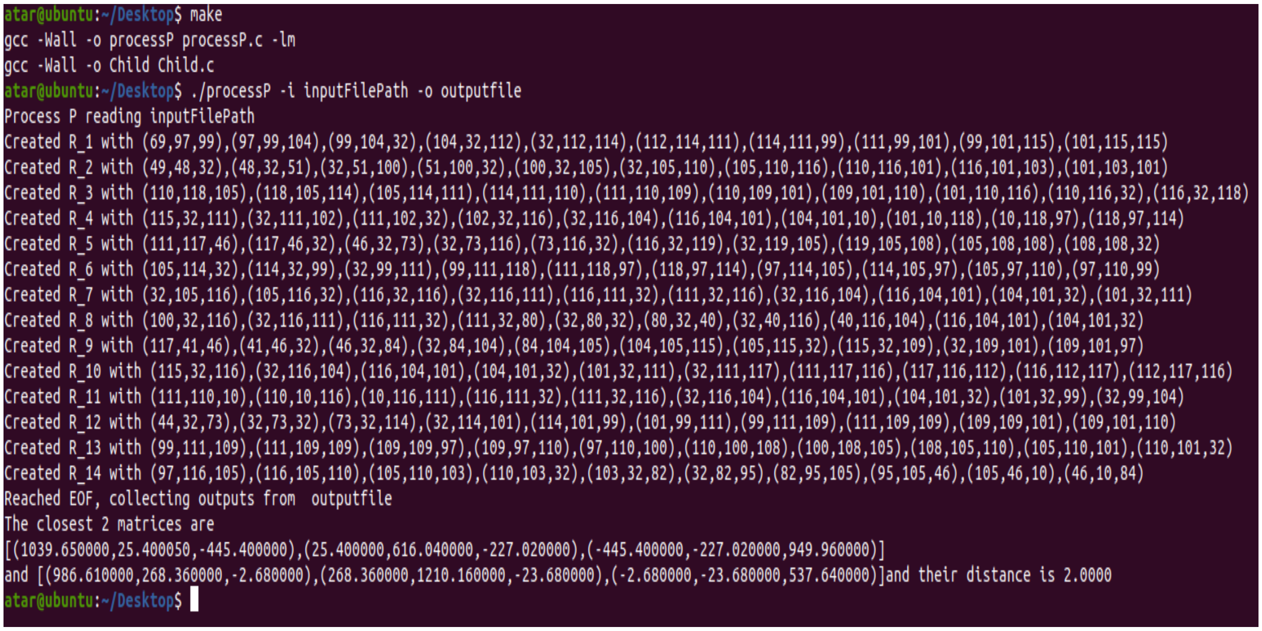
(File read, write, lock using syscalls, signal handling, parent child signal relations,multiple child process, fork, exec family ,make files, waiting for the child process to finish their task)

InputFilePath



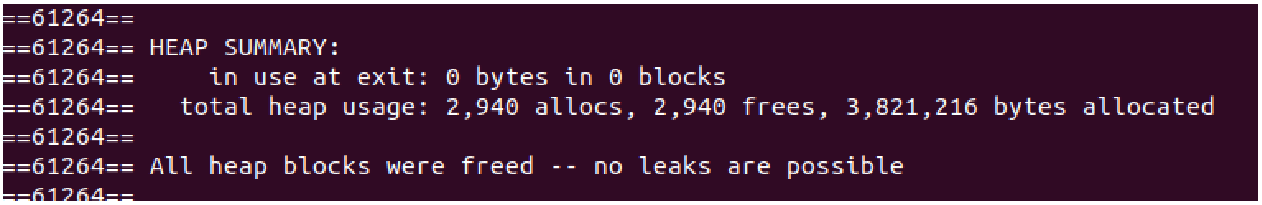
```
Each process R_i will receive 10 3d integer coordinates as environment variables. The names of the
variables are up to you. It will then calculate their covariance matrix and write it to the output file
provided to P (the format is up to you). This means you need to pass the outputfilepath information
to the child process from P, I recommend passing it as a commandline argument while creating R_i.
Then R_i will terminate.
```

Running And Result



```
atar@ubuntu:~/Desktop$ make
gcc -Wall -o processP processP.c -lm
gcc -Wall -o Child Child.c
atar@ubuntu:~/Desktop$ ./processP -i inputFilePath -o outputfile
Process P reading inputFilePath
Created R_1 with (69,97,99),(97,99,104),(99,104,32),(104,32,112),(32,112,114),(112,114,111),(114,111,99),(111,99,101),(99,101,115),(101,115,115)
Created R_2 with (49,48,32),(48,32,51),(32,51,100),(51,100,32),(100,32,105),(32,105,110),(105,110,116),(110,116,101),(116,101,103),(101,103,101)
Created R_3 with (110,118,105),(118,105,114),(105,114,111),(114,111,110),(111,110,109),(110,109,101),(109,101,110),(101,110,116),(110,116,32),(116,32,118)
Created R_4 with (115,32,111),(32,111,102),(111,102,32),(102,32,116),(32,116,104),(116,104,101),(104,101,10),(101,10,118),(10,118,97),(118,97,114)
Created R_5 with (111,117,46),(117,46,32),(46,32,73),(32,73,116),(73,116,32),(116,32,119),(32,119,105),(119,105,108),(105,108,108),(108,108,32)
Created R_6 with (105,114,32),(114,32,99),(32,99,111),(99,111,118),(111,118,97),(118,97,114),(97,114,105),(114,105,97),(105,97,110),(97,110,99)
Created R_7 with (32,105,116),(105,116,32),(116,32,116),(32,116,111),(116,111,32),(111,32,116),(32,116,104),(116,104,101),(104,101,32),(101,32,111)
Created R_8 with (100,32,116),(32,116,111),(116,111,32),(111,32,80),(32,80,32),(80,32,40),(32,40,116),(40,116,104),(116,104,101),(104,101,32)
Created R_9 with (117,41,46),(41,46,32),(46,32,84),(32,84,104),(84,104,105),(104,105,115),(105,115,32),(115,32,109),(32,109,101),(109,101,97)
Created R_10 with (115,32,116),(32,116,104),(116,104,101),(104,101,32),(101,32,111),(32,111,117),(111,117,116),(117,116,112),(116,112,117),(112,117,116)
Created R_11 with (111,110,10),(110,10,116),(10,116,111),(116,111,32),(111,32,116),(32,116,104),(116,104,101),(104,101,32),(101,32,99),(32,99,104)
Created R_12 with (44,32,73),(32,73,32),(73,32,114),(32,114,101),(114,101,99),(101,99,111),(99,111,109),(111,109,109),(109,109,101),(109,101,110)
Created R_13 with (99,111,109),(111,109,109),(109,109,97),(109,97,110),(97,110,100),(110,100,108),(100,108,105),(108,105,110),(105,110,101),(110,101,32)
Created R_14 with (97,116,105),(116,105,110),(105,110,103),(110,103,32),(103,32,82),(32,82,95),(82,95,105),(95,105,46),(105,46,10),(46,10,84)
Reached EOF, collecting outputs from  outputfile
The closest 2 matrices are
[(1039.650000,25.400050,-445.400000),(25.400000,616.040000,-227.020000),(-445.400000,-227.020000,949.960000)]
and [(986.610000,268.360000,-2.680000),(268.360000,1210.160000,-23.680000),(-2.680000,-23.680000,537.640000)]and their distance is 2.0000
atar@ubuntu:~/Desktop$ 
```

Valgrind – No memory leak



```
==61264==
==61264== HEAP SUMMARY:
==61264==     in use at exit: 0 bytes in 0 blocks
==61264==   total heap usage: 2,940 allocs, 2,940 frees, 3,821,216 bytes allocated
==61264==
==61264== All heap blocks were freed -- no leaks are possible
==61264==
```

Compile and Run

make

./processP   –i inputFilePath   –o outputFilePath