System Programming HW-3 Report

Command line arguments

As It requested, command line arguments are taken by getopt() function and in case of inappropriate command line arguments or insufficient input files, an error message is written on terminal and exiting through **clean_resources()** function which is registered by atexit() function. In **clean_resources()** function opened files are closed.

Input files must contain (2^n) x (2^n) ascii characters. For example if n is 2, input files are supposed to contain 4x4 ascii character matrices.

At the beginning…

Operations are started with **do_homework(...)** function, in this function files are read and corresponding matrices are initialized.
Then **start_multi_process(...)** function is called, in here pipes are prepared and moved on **organize_children_processes(…)**

Organize_children_processes(…) function

- Firstly, a signal handler is registered using sigaction() function, in the signal handler function SIGINT and SIGCHLD signals are handled.
- Sub matrices are produced from the given two matrices to calculate matrix multiplication. Then sent two by two to the corresponding processes to calculate each quarter of result matrix.
- To not to cause blocking the program because of unused pipe fd's. Those fd's are closed.
- To do synchronization barrier, SIGCHLD signal is waited in the signal handler function, and each time SIGCHLD signal is received number of child processes are decrement by one. So the parent process waits for the num_of_children_process counter to be zero to continue its execution. This barrier is located just after sending the sub matrices to children processes.

Managing Pipes

In Children processes:

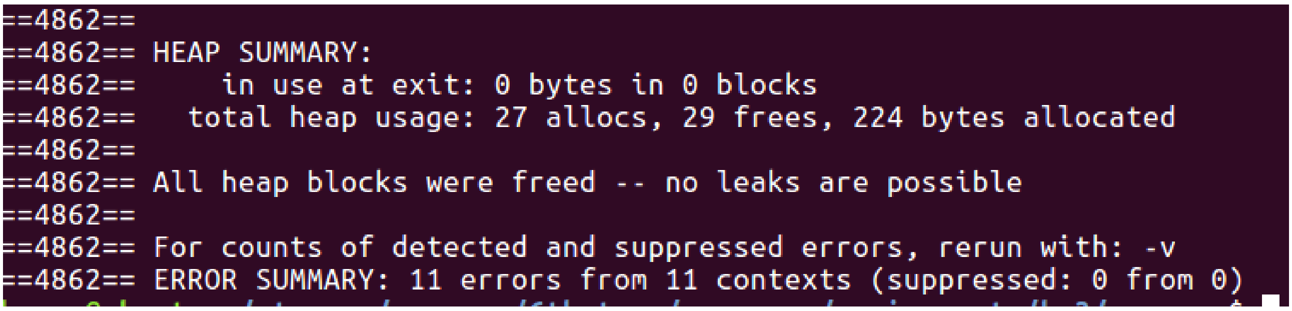
For each child process, as soon as it starts, unused pipes fd's are closed and after its execution is done, the fd's of its pipe are closed.

In Parent process:

Parent sends needed data for calculation then closes the pipes write fd's, and waits for children to finish their tasks then read with read fd's of pipes then closes those fd's too.

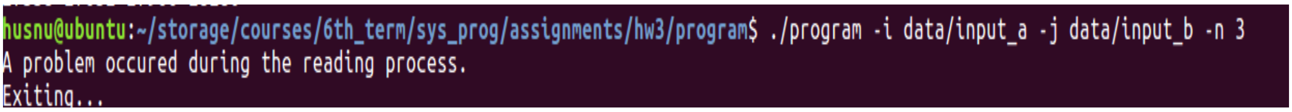
Screen Shots

***Memory leak control**



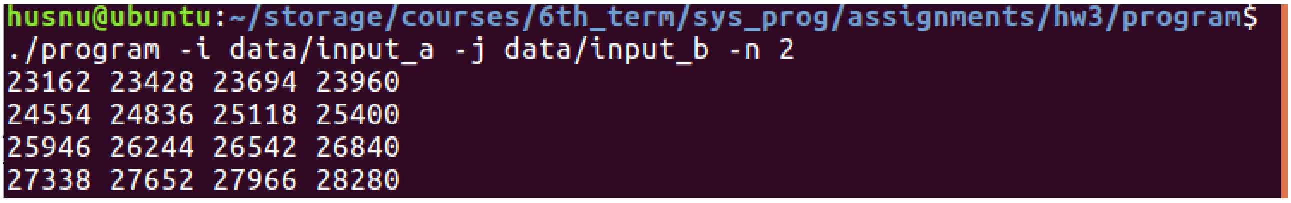
```
==4862==
==4862== HEAP SUMMARY:
==4862==     in use at exit: 0 bytes in 0 blocks
==4862==   total heap usage: 27 allocs, 29 frees, 224 bytes allocated
==4862==
==4862== All heap blocks were freed -- no leaks are possible
==4862==
==4862== For counts of detected and suppressed errors, rerun with: -v
==4862== ERROR SUMMARY: 11 errors from 11 contexts (suppressed: 0 from 0)
```

*** In sufficient file content error and termination...**



```
husnu@ubuntu:~/storage/courses/6th_term/sys_prog/assignments/hw3/program$ ./program -i data/input_a -j data/input_b -n 3
A problem occured during the reading process.
Exiting...
```

*** Normal termination, program exits after printing result matrix.**



```
husnu@ubuntu:~/storage/courses/6th_term/sys_prog/assignments/hw3/program$
./program -i data/input_a -j data/input_b -n 2
23162 23428 23694 23960
24554 24836 25118 25400
25946 26244 26542 26840
27338 27652 27966 28280
```