

Gebze Technical University

CSE 344 – 2022

System Programming

Homework – 2 Report

Muhammed Bedir ULUCAY

1901042697

Algorithm Design:

Parent process reading 3 bytes for each time making a system call this is a coordinate for a 3D space and if we collect this coordinate 10 times we are creating a child process for calculating the covariance matrix of these points. And after all child calculations are done parent process calculates the Frobenius Norm of these matrices and try to find out which two matrices are closest to each other. For the parent process, it must be suspended until all the child processes are have done their calculations so it creates a sectorization barrier and we are waiting for a for loop with wait(NULL) function, and with these all resources of child processes giving the OS. Then parent processes continue the execution from where they left off. And reading the output file according to the determined output. The determined type for the output is comma-separated each number separated with a comma. Reading the calculated all matrices and starting the Frobenius calculation for all matrices. The finding the closest two matrices and printing the results to the console.

Interrupt Handling:

When the parent process gets a SIGINT signal with Ctrl+C command from the terminal it is executing immediately the handling function and a sig_atomic_t flag variable is converting 0 to 1. And a function from time to time checks this variable the process is to get the SIGINT signal. If the signal arrived the parent process. It is giving the OS all allocated memory, opened files, and wait(NULL) call for each child (including rm command). For avoiding the some error (like double free(void*)) giving the variable initial values.

Ex:

```
sig_atomic_t is_get = 0;           int *child_processIds = NULL;
int output_fd = -1;                char* inputFile = NULL;
char* num = NULL;
```

When SIGINT arrives the child process child is also giving its resources to the OS.

For the critical region generally, I use the SIGINT like **a when an interrupt occurs after a[3] and not allocate for the a[4] this cause an error in the handler function with free(*) so like these situations the mask for SIGINT and delay the signal arrive for the current process.