System Programming HW#5 Report

Name:Mustafa Tokgöz *Student Id:171044077*

1 HOW TO SOLVE THIS PROBLEM

Firstly, I initialize the variables for using system V semaphores, thread variables and SIGINT handler. I take the arguments given from user then I check if it is wrong or not. I round down the thread number if it is necessary. I create the global variables to use in threads and calculations. I use calloc function for global 2D arrays. Then I open the files and read the file into arrays as matrices. If a file doesn't content enough character then I return the fatal error and resources back. After that I create the threads. Then I wait the threads and collect the outputs of the threads with join function. Then I write them to output file as csv format. In signal handler function, I give all resources back. In thread function, I use mutex and conditional variable. I use synchronization barrier in the lecture slides. In first part of all threads, I calculate the matrix product by separating given matrices into number of thread. So I calculate 2^n/m column of result matrix. After that I wait the other threads to do it by barrier. Then In second part, I calculate the 2D DFT for our result matrix by separating result matrix into thread number. There is a formula of 2D DFT and in this formula there is addition number but in our task is separating this task into threads. In this condition, result can be change in order to addition number actually thread number. In synchronization barrier, I use pthread_cond_wait and pthread_cond_broadcat function to synchronize. Threads return 2D array that has 2^n/m column with pthread_exit function.

2 DESIGN DECISIONS

- I print to stdout or stderr using write function.
- I use mutex and condition variable and initilize them statically.
- I print the timestamp at the beginning of each line.
- I use sigaction for handling SIGINT.
- I use getopt function to take arguments from user.
- I use clock function to calculate the time that past.

- I use math library for calculating sin and cos and using pi number.
- I use pthread_exit function instead of return.

3 TRIAL VARIOUS VARIABLES

I try n and m variables differently. I saw that program finish time decreases by increasing m. When increasing n then time will increase. For example When I run the program with n 4 and m 2 then total time is 0.008 for my input. But when I increase the thread then total time will be smaller. I run the program with n 4 and m 4 then total time is 0.006. So there is a time recovery in our program by increasing thread number and thread number effects directly to it. I tried the program with different inputs.

4 WHICH REQUIREMENTS I ACHIEVED

I achieved all requirements of the homework. I creating thread, waiting threads with join function, synchronization barrier part, printing timestamp, calculating time past. calculating matrix product and dft parts, synchronization, reading file and writing file, SIGINT handler, checking arguments. I achieved all requirements.