

## Operating System Homework 1

This homework is to analyze and compare the performance of a computing task using multiple processes and threads.

For the computing task, prepare a large array of integers/strings/structures with contents randomly generated with repetitions. Do not sort this array. The task is to find out the number of times an inquiry element occurs in the array. Write a C program (without any threads or child process) to perform the task and measure the execution time (using `gettimeofday()`). A typical output of the program may look like "Integer 13 occurs 5 times in the array." Keep the same input and inquiry for all program tests to ensure the correctness of any program modification.

For multi-process version, the parent processes divides the array into multiple segments, one for each child process. Each child process returns (using `exit()`) the number of times the inquiry element occurs in that segment. The parent process takes the sum of all results reported by child processes. Refer to the following page for getting the exit status of child process.

<https://support.sas.com/documentation/onlinedoc/ccompiler/doc700/html/lr2/zid-9832.htm>

For multi-threaded version, the main thread divides the array into multiple segments, one for each working thread. That are two ways for the main thread to take the sum of all results reported by working threads.

- 1) Each working thread uses `pthread_exit()` to return the number of times the inquiry element occurs in that segment and the main thread uses `pthread_join()` to get the result.
- 2) Define a global counter to keep the final result. Each working thread adds to the global counter the number of times the inquiry element occurs in that segment. You may need mutex to avoid data inconsistency.

**Goal:** You have to figure out how some factors affect performance of multi-process and multi-thread programming. These factors under consideration include the number of segments (i.e., the number of child processes/working threads) and the size of the input data (i.e., the size of the array).

**Report:** Hand in your report in PDF format that includes the description of your platform (CPU type, memory size, kernel version, physical or virtual machine, etc.) and the measurement results (using figures or tables). Discuss possible justification for the results. Also hand in your source codes.