CSE 231: Operating System **Assignment-5 (Section-A)**

# SimpleMultithreader: Using Multithreading with Ease

## Group Details

**Group Member 1:**

Name: Himanshu
Roll no: 2023241
Gmail ID : himanshu23241@iiitd.ac.in

**Group Member 2 :**

Name: Harsh Gupta
Roll no : 2023229
Gmail ID : harsh23229@iiitd.ac.in

## Contribution:

- All the work was collectively and equally done by both the participants, Himanshu and Harsh Gupta
- The code programming and compilation for Multithreader was collectively discussed and executed.
- Code compilation was carried out on an Intel-based machine.

**Himanshu :**

1) VECTOR
    a. Structure for single-index
    b. (Thread_func_for) Function executed by each thread in parallel_for
    c. parallel_for function for vector

**Harsh Gupta :**

1) MATRIX
    a. Structure for double-index
    b. (thread_func_double_for) Function executed by each thread in parallel_for
    c. parallel_for function for matrix

## **Code Documentation:**

SIMPLE MULTITHREADER

Multithreading is a technique used to perform multiple tasks concurrently, improving the overall performance of a program. This library simplifies the process of parallelizing loops, allowing developers to harness the power of multicore processors.

### Function explanation:

Data Structures: These structures (thread_function_vector and thread_function_matrix) are used to encapsulate the parameters required for executing 1D and 2D loops in parallel. They include the loop boundaries (start, end, start1, end1, start2, end2) and a lambda function (lambda) that represents the operation to be performed in each iteration.

Thread Functions: These functions represent the entry points for threads executing 1D and 2D loops, Respectively.

Thread Function for : This function takes a ThreadData1D structure as an argument, executes the lambda function for each iteration of the loop and measures the elapsed time.

Thread Function double for : This function takes a ThreadData2D structure as an argument, executes the lambda function for each iteration of the 2D loop and measures the elapsed time.

Parallel For Functions: These functions implement the parallelization logic for single-dimensional and two-dimensional loops. They create and manage threads, dividing the work among them based on the specified number of threads (numThreads).

- **parallel_for Method for the vector file**
    - Accepts a lambda function representing the loop body.
    - Runs the loop body in parallel using a specified number of Pthreads (numThreads).
    - Dynamically creates Pthreads without utilizing task/thread pools.
    - Ensures the exact number of threads specified by the programmer, including the main thread.

- **parallel_for Method for the matrix file**
    - Designed for parallelizing two-dimensional for-loops.
    - Utilizes C++11 lambda expressions for the loop body.
    - Creates a new set of Pthreads for each call, terminating upon the scope's completion.

## **Github repository link:**

- GitHub repo: https://github.com/himanshu23241/Operating_System-CSE231