

OS ASSIGNMENT BY GROUP-71

GROUP MEMBERS CONTRIBUTION:

- 1) Ayush Singhal: Implemented vector.cpp file.
- 2) Sanjeet Kumar Patel: Implemented matrix.cpp file.

This assignment was a combined effort from both of us where we both have contributed to our best.

SUMMARY OF ASSIGNMENT 5 IMPLEMENTATION

Program 1 (Vector Addition):

- Objective:
 - Parallel addition of two vectors ($C = A + B$).
- Input:
 - Command-line arguments for the number of threads (`numThread`) and vector size (`size`).
- Implementation:
 - Dynamic allocation of vectors A, B, and C.
 - Initialization of vectors A and B with constant values.
 - Parallel computation of vector addition using the `parallel_for` construct.
 - Verification of the result vector (C) to ensure correctness.
- Output:
 - Success message if the verification passes.

Program 2 (Matrix Multiplication):

- Objective:
 - Parallel matrix multiplication ($C = A * B$), where A, B, and C are matrices.
- Input:
 - Command-line arguments for the number of threads (`numThread`) and matrix size (`size`).
- Implementation:
 - Dynamic allocation of matrices A, B, and C.
 - Parallel initialization of matrices A, B, and C.
 - Parallel computation of matrix multiplication using the `parallel_for` construct.
 - Verification of the result matrix (C) to ensure correctness.
 - Cleanup of dynamically allocated memory after computation.
- Output:
 - Success message if the verification passes.

Common Elements:

- Both programs utilize a simple multithreading library (`simple-multithreader.h`) for parallelization.
- Command-line arguments allow flexibility in specifying the number of threads and the problem size.
- Verification steps ensure the correctness of the parallelized computations.
- Dynamic memory allocation is used for vectors and matrices, and proper cleanup is performed at the end.

GITHUB LINK:

[LINK](#)