GROUP MEMBERS

GitHub -https://github.com/Akshat22052/OS-ASS5

1. AKSHAT KARNWAL – 2022052
2. MOHD MASOOD – 2022299

IMPLEMENTATION OF THE CODE

Parallel Programming with Threads in C++

This C++ program demonstrates parallel programming using threads for both matrix and vector operations. It incorporates the use of lambda functions and multi-threading to perform tasks concurrently. The code is structured to showcase parallelization concepts and record execution times.

Global Variables:

- **`long long t_time`**: Accumulates the total execution time across different intervals.
- **`int thread_cnt`**: Counts the number of threads used during execution.

Functions:

1. **Matrix Problem:**
   - **`thread_args1` Structure:**
     - Holds parameters for matrix-related thread operations.
   - **`thread_func1` Function:**
     - Thread function for matrix operations.
   - **`parallel_for` Function:**
     - Accepts a lambda function and executes it in parallel using threads.
     - Divides the matrix into chunks, distributes them among threads, and measures execution time.
     - Accumulates the total execution time (`t_time`).

2. **Vector Problem:**
   - **`thread_args` Structure:**
     - Holds parameters for vector-related thread operations.
   - **`thread_func` Function:**
     - Thread function for vector operations.
   - **`parallel_for` Function:**
     - Accepts a lambda function and executes it in parallel using threads.
     - Divides the vector into chunks, distributes them among threads, and measures execution time.
     - Accumulates the total execution time (`t_time`).

3. **`main` Function:**
   - Calls the `user_main` function (which can be replaced with the actual program logic).
   - Uses lambda expressions to demonstrate capturing variables by value and reference.
   - Executes both matrix and vector parallel operations.
   - Prints the total execution time.

Lambda Expressions:

- Demonstrates the use of lambda expressions to create functions on the fly and pass them as parameters.
- Example lambda functions capture variables `x` and `y` with different capture modes.

Output:

- Test case status
- The total execution time for the entire program is printed at the end.
-

CONTRIBUTION

1. Masood did the matrix.
2. Akshat did the vector.