

Machine Exercise No.4 – Multithreaded Matrix Multiplication

In this exercise, students will implement a program that performs matrix multiplication using multiple threads.

Write a program that multiplies two large matrices, for example 100×100 in size. Divide the computation across a specified number of threads, where each thread is responsible for calculating either a row block or a submatrix of the result. Once the program is working, measure the execution time with different numbers of threads (1, 2, 4, and 8) and analyze the results to determine the speedup achieved through multithreading.

Machine Exercise No.5 – Thread Cancellation

The objective of this exercise is to explore the concepts of **asynchronous** and **deferred cancellation** in multithreaded programming.

Write a program that creates a worker thread running an infinite loop that prints numbers. From the main thread, issue a cancellation request using the **pthread_cancel()** function. Experiment with two scenarios: in the first, use asynchronous cancellation so that the worker thread is terminated immediately regardless of its current state; in the second, use deferred cancellation by having the thread periodically call **pthread_testcancel()** inside the loop to check for cancellation requests before stopping. Compare the behavior in both cases and observe how the program handles resource cleanup when threads are cancelled.