**Class:** Final Year (Computer Science and Engineering)

**Year:** 2024-25 **Semester:** 1

**Course:** High Performance Computing Lab

**Practical No. 10**

**PRN : 21510074**

**Name : Yash Nawale**

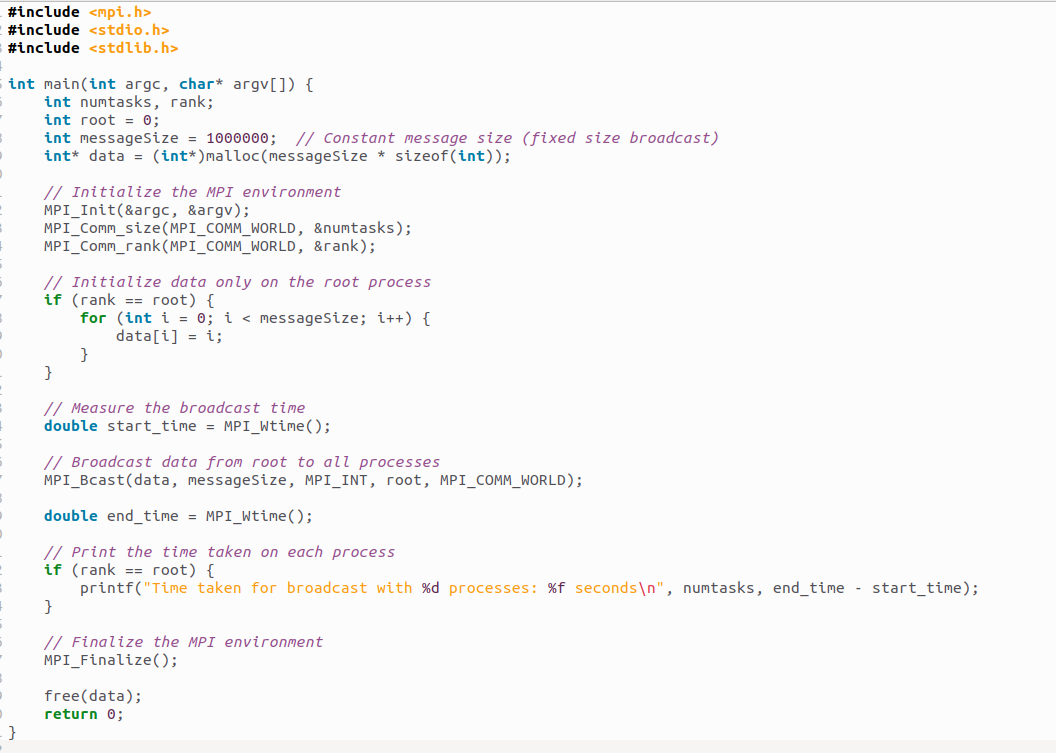
**Batch : B4**

**Title of practical :** Analysis of MPI Programs

**Problem Statement 1:**

Execute the MPI program (Program A) with a fixed size broadcast. Plot the performance of the broadcast with varying numbers of processes (with constant messagesize). Explain the performance observed.

**Screenshot:**



A screenshot of a computer program

Description automatically generated

**Analysis:**

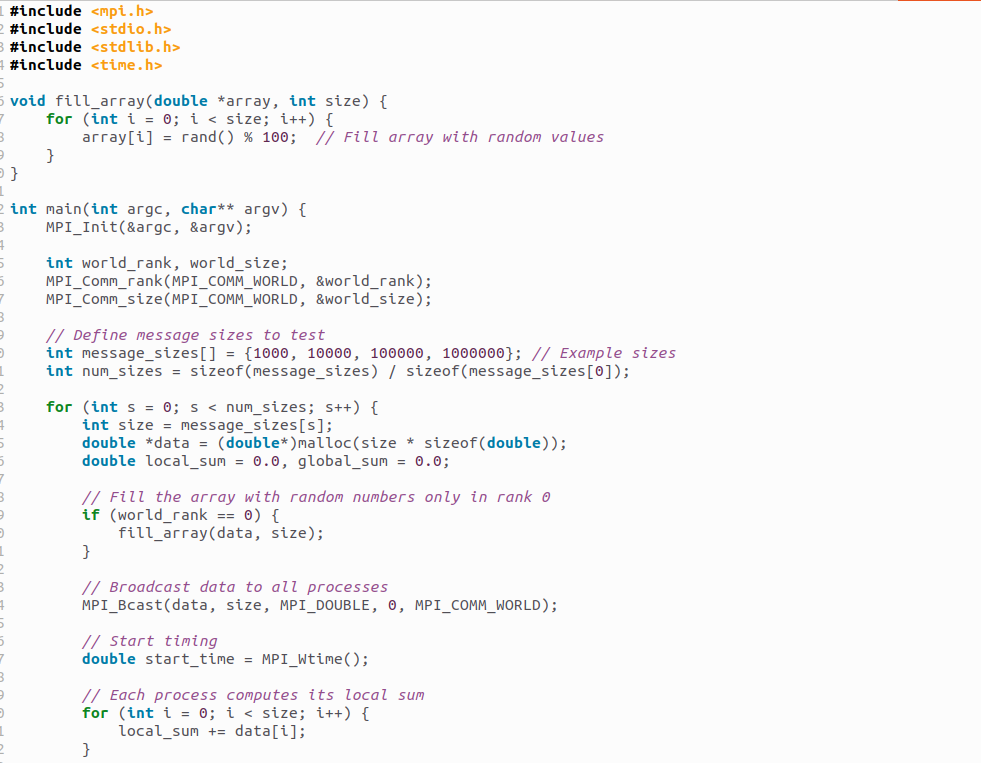
* 5 Processes: The broadcast time is low because the number of participants is small, and communication overhead is minimal.
* 10 Processes: Doubling the number of processes causes a noticeable increase in time due to the increased communication between processes.
* 20 Processes: As the number of processes continues to increase, the broadcast time rises. Communication and synchronization overhead between processes become more significant.
* 30 Processes: With 30 processes, the broadcast time increases further, likely due to network congestion and MPI's internal tree-based broadcast algorithm handling more nodes.

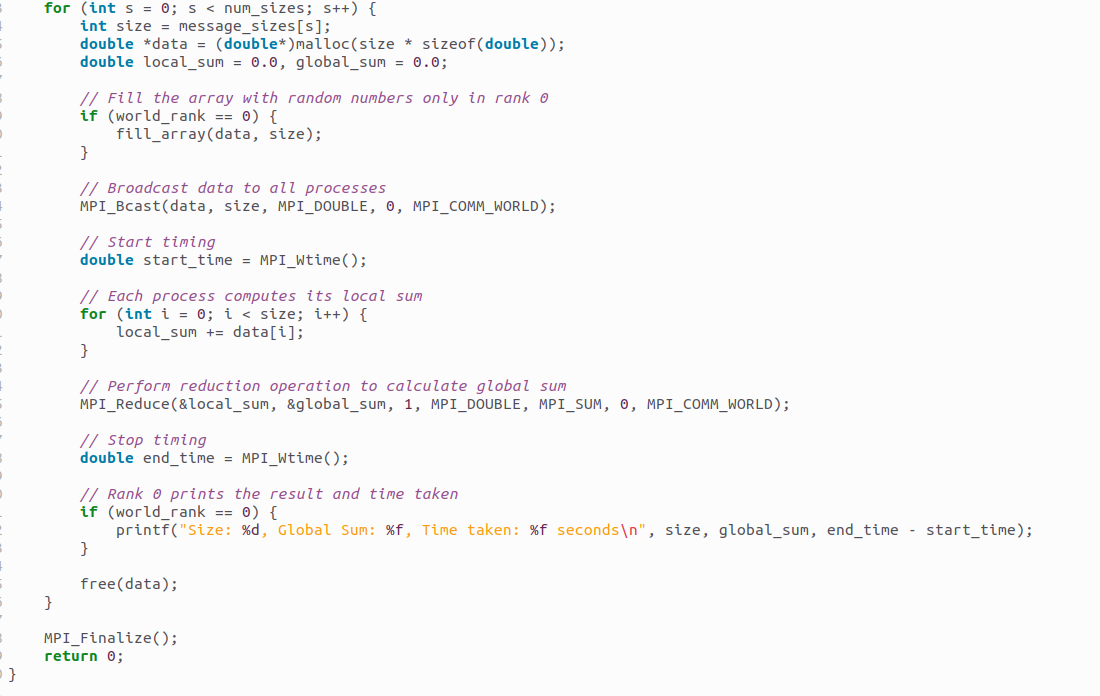
As the number of processes increases, the time for a fixed-size broadcast operation increases due to communication overhead and the scalability of the broadcast algorithm. In real systems, optimizations can be applied, but the trend will generally show increasing time with more processes.

**Problem Statement 2:**

Repeat problem 2 above with varying message sizes for reduction (Program B). Explain the observed performance of the reduction operation.

**Screenshot:**





A screenshot of a computer program

Description automatically generated

**Analysis:**

* 1,000 Elements: With a small message size, the reduction time is very low. The overhead of the reduction is minimal because the amount of data being transmitted between processes is small.
* 10,000 Elements: As the message size increases by a factor of 10, the time also increases slightly, but it’s still within a very small range due to the efficient reduction algorithms used by MPI.
* 100,000 Elements: With larger messages, the time grows more significantly. This increase in time reflects the growing communication and synchronization overhead involved in reducing larger amounts of data across processes.
* 1,000,000 Elements: At this point, the reduction time becomes more substantial, as each process is handling a much larger chunk of data, and the communication between processes starts to dominate the performance.

[GitHub Link :-](https://github.com/Shreyak810/HPC-LAB/tree/main/Assignment-10)

<https://github.com/YashNawale26/High-Performance-Computing>