# Assignment – 2

## Group number-29


1. Manvitha Aathmuri – R11847781
2. Raviteja Sirisanagandla - R11851152
3. Sai Tejaswini Chirumamilla - R11840694
4. Preetham Alwala - R11846544
5. Girish Gandham - R11870388
6. Ganesh Reddy Mannem - R11849144


## MPI Barrier Synchronization Implementation:

Barrier synchronization is a critical concept in parallel computing, ensuring that a group of processes waits for each other to reach a specific point in their execution before proceeding further. In this project, we have implemented barrier synchronization among a group of processes using the Message Passing Interface (MPI) in C. The goal is to design an efficient barrier synchronization mechanism that minimizes the number of steps for send-receive operations.

## Problem Statement:

For each process in a group of P processes, the requirement is to implement a barrier synchronization mechanism. The behavior of this barrier synchronization is such that no process can leave the code segment until all other processes in the group have entered their respective barrier code segments. The challenge is to accomplish this efficiently with a minimal number of send-receive operations.

## Technical Idea:

The “mybarrier” function, which was created specifically for this code, synchronizes barriers. Process will not move out of the code segment until every other process inside of an MPI communicator has reached a point of synchronization. Using bitwise XOR operations, the algorithm determines the partner processes for message exchange and determines the number of synchronization steps based on the communicator's size. After the barrier, it performs some computation before and after, and then finalizes MPI.

## Conclusion:

The implementation of barrier synchronization among a group of processes using MPI in C provides an efficient solution for parallel computing environments. By leveraging a binary tree-based approach, the algorithm minimizes the number of send-receive operations and ensures that processes synchronize effectively. This project contributes to the advancement of parallel computing techniques and can be a valuable tool for applications requiring synchronization among multiple processes.

## Steps to Execute:

- Open the Group_29_assignment_2.c file's location in terminal.
- Compile the file Group_29_assignment_2.c using the below command:
  mpicc Group_29_assignment_2.c -o Group_29_assignment_2 -lm
- Execute the file using below command for the output to be visible in console log:
  mpirun -np 6 ./ Group_29_assignment_2

## Output: