

Distributed Computing Lab Programs

1. Demonstration of MPI_Send and MPI_Recv
2. Demonstration of MPI_Scatter and MPI_Gather
3. Demonstration of MPI Broadcast operation
4. Demonstration of MPI_Reduce and MPI_Allreduce (MPI_MAX, MPI_MIN, MPI_SUM, MPI_PROD).
5. Write an OpenMP program that computes the sum of the first N integers using a parallel for loop. Use the `#pragma omp` for directive along with the private and reduction clauses.
6. Write an OpenMP program to compute the Fibonacci sequence using task parallelism. The program should use a recursive function where each Fibonacci computation for `fib(n-1)` and `fib(n-2)` is created as an independent task.
7. Estimate the value of pi using: Parallelize the code by removing loop carried dependency.
8. Write a parallel C program using OpenMP in which each thread obtains its thread number and adds it to a global shared variable. Use the `#pragma omp critical` directive to avoid race conditions. Print the intermediate updates and the final sum.
9. Write a program to sort an array of n elements using both sequential and parallel merge sort (using Section). Record the difference in execution time.