This C code demonstrates parallel computation of the sum of elements using MPI (Message Passing Interface). Here's a simplified explanation:

Include Header Files:

#include <stdio.h>: Includes standard input-output functions.

#include <stdlib.h>: Includes memory allocation functions.

#include <mpi.h>: Includes MPI library functions for parallel computing.

Define Constants:

#define UNITSIZE 5: Defines the size of each unit of data to be processed.

Main Function:

Declares variables rank, size, send\_buffer, receive\_buffer, new\_receive\_buffer, root, i, total\_elements, and total\_sum.

Initializes MPI with MPI\_Init.

Retrieves rank and size of the MPI communicator with MPI\_Comm\_rank and MPI\_Comm\_size, respectively.

Data Distribution:

In the root process (rank 0), allocates memory for send\_buffer and prompts the user to enter elements.

Uses MPI\_Scatter to distribute portions of send\_buffer to all processes.

Partial Computation:

Each process computes the sum of its portion of data (received in receive\_buffer) and stores the intermediate result in receive\_buffer[0].

Gathering Intermediate Results:

In the root process, allocates memory for new\_receive\_buffer.

Uses MPI\_Gather to gather intermediate results from all processes into new\_receive\_buffer.

Final Sum Calculation:

In the root process, computes the final sum by summing up the elements in new\_receive\_buffer.

Finalization and Memory Deallocation:

Finalizes MPI with MPI\_Finalize.

Frees dynamically allocated memory for send\_buffer, receive\_buffer, and new\_receive\_buffer.

Return:

Returns 0 to indicate successful execution.