Sharing Data

Sharing Data

- int MPI_Pack(const void* inbuf, int incount, MPI_Datatype datatype, void* outbuf, int outsize, int* position, MPI_Comm comm);

- int MPI_Unpack(const void* inbuf, int insize, int* position, void* outbuf, int outcount, MPI_Datatype datatype, MPI_Comm comm);

MPI_Pack

- It is used for packing data into a contiguous buffer. The purpose of packing is to prepare data for transmission, typically when we want to send non-contiguous data in a single message. This function is often used in conjunction with MPI_Unpack on the receiving side.

MPI_Unpack

- This function is used for unpacking data that was previously packed using MPI_Pack.

Contiguous Buffer

- A contiguous buffer refers to a block of memory where the elements are stored consecutively without any gaps.

```
int array[5] = {1, 2, 3, 4, 5};
```

Problem Statement

Create a program that reads an integer and a double-precision value from standard input (from process 0).

Then, it communicates this to all of the other processes with an MPI_Bcast call. Use MPI_Pack to pack the data into a buffer.

Problem Statement

The program will continuously ask for the two input values.
At each iteration, the evolution needs to be shown.
Finally, if the user introduces a negative value input, all the processes will terminate.

MPI_Unpack

- This function is used for unpacking data that was previously packed using MPI_Pack.

Broadcast

- **MPI_Bcast** distributes data from one process (the root) to all others in a communicator.

```
int array[100];
int root = 0;
…
MPI_Bcast(array,100,MPI_INT,root,MPI_COMM_WORLD);
```