

TP3 : MPI

Exercise 1 : Hello World Program

Display for each process launched by `mpirun` one line containing :

- Hello World ;
- the rank of the process called ;
- the total number of processes.

Exercise 2 : Ping Pong with Vectors

Realize a ping-pong program with two processes. The goal is to change the size of the table (thanks to random numbers) for each loop (a send and a receive operation for each process). The program must take an integer as input in order to set a limit for the number of loop.

Exercise 3 : Wall Time

Add to ping-pong program the wall time at the end of the program using the function `MPI_WTIME()`.

```

startTime=MPI_WTIME();
//Some work
endTime=MPI_WTIME();
std::cout << "Wall Time: " << endTime-startTime << std::endl,

```

Exercise 4 : Ping Pong with Non-Blocking Communications

Realize another ping-pong version with non-blocking communications. During the communication of a table, calculate the sum of this table.

Exercise 5 : Array Assignment

In order to do a data decomposition, the master first initializes an array and then distributes an equal portion. After the other tasks receive their portion of the array, they perform an addition operation to each array element. They also maintain a sum for their portion of the array. The master task does likewise with its portion of the array. As each of the non-master tasks finish, they send their updated portion of the array to the master. Communications have to be used to collect the sums maintained by each task. Finally, the master task displays selected parts of the final array and the global sum of all array elements.