

TCE3411/TSN3151 – Parallel Processing

Laboratory 2: “Hello World”, MPI Programming I

Lab Sheet

Guideline:

As for all the TCE3411/TSN3151 lab sessions, it is expected that you at least try out all lab exercises individually. However, you are encouraged to DISCUSS on your answers or assist each other in small groups. All the guides that you required are given in the slides uploaded for this lab. I will not be demonstrating or lecturing in class but I will be around for you to ask questions.

Introduction

This laboratory aims to provide an example of a simple MPI “Hello World” program where messages can be passed to several processes.

Sample Program

Below is a sample file (hello.c):

```
#include <stdio.h>
#include <mpi.h>

int main(int argc, char *argv[])
{ int numprocs, rank, namelen;
  char processor_name[MPI_MAX_PROCESSOR_NAME];

  MPI_Init(&argc, &argv);
  MPI_Comm_size(MPI_COMM_WORLD, &numprocs);
  MPI_Comm_rank(MPI_COMM_WORLD, &rank);
  MPI_Get_processor_name(processor_name, &namelen);

  printf("Process %d on %s out of %d\n", rank,
        processor_name, numprocs);

  MPI_Finalize();
}
```

Compilation

Once you have your MPI example program, you can compile and link it with the MPI version that supports the network type you need.

```
mpicc hello.c -o hello
```

Running MPI Program

To run your program with 4 processes:

```
mpirun -np 4 ./hello
```

How many processors should you run? Type `nproc` to find out how many processors your PC has, and use that number.

Figure out the answers to these questions. They may help you answer the quiz during the next lecture.

- a. Why do the numbers in the sample `hello.c` come out of order?
- b. Name the two commands to initialize and finalize a MPI program.
- c. Explain the function of `MPI_Comm_World`.
- d. Describe the function of `MPI_Comm_rank`.