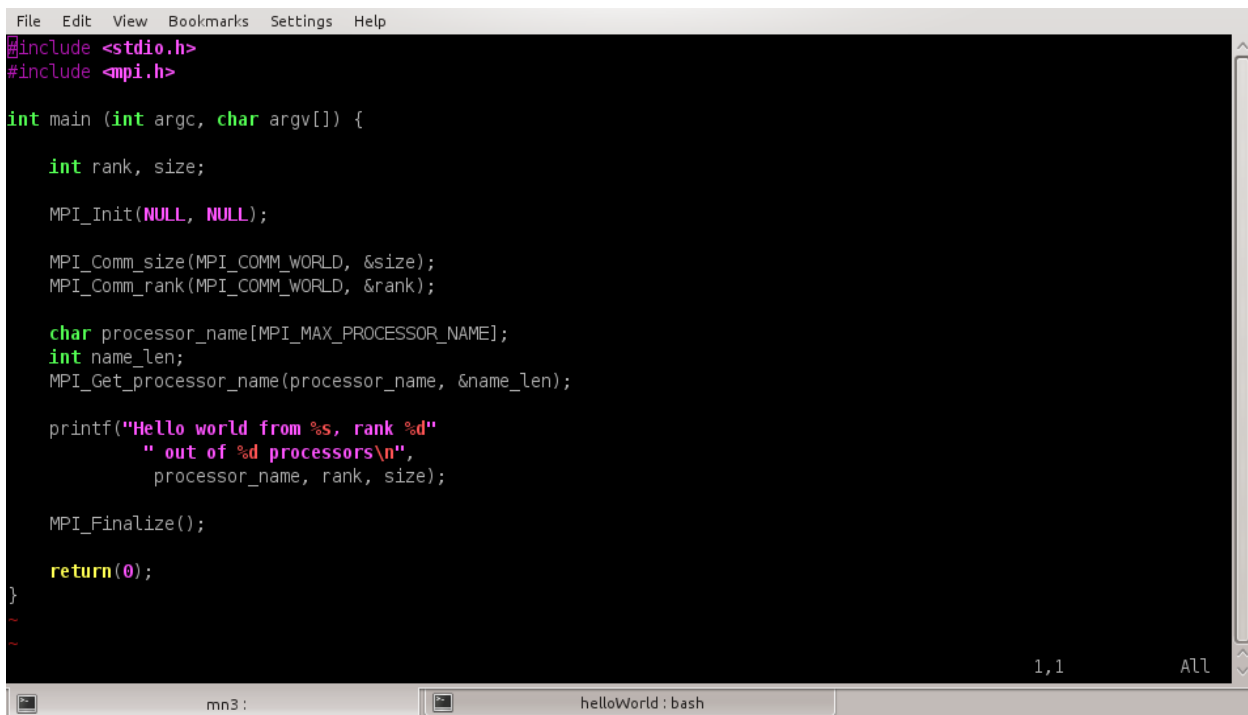


Practical MPI

HelloWorld

Login to MareNostrum with your student account and find the source files in the professor account:

/home/nct00/nct00002/helloWorld/

A screenshot of a code editor window with a dark background and light-colored text. The editor has a menu bar at the top with 'File', 'Edit', 'View', 'Bookmarks', 'Settings', and 'Help'. The code is written in C and includes MPI headers. It defines a main function that initializes MPI, gets the processor name and rank, and prints a message. The code is as follows:

```
File Edit View Bookmarks Settings Help
#include <stdio.h>
#include <mpi.h>

int main (int argc, char argv[]) {

    int rank, size;

    MPI_Init(NULL, NULL);

    MPI_Comm_size(MPI_COMM_WORLD, &size);
    MPI_Comm_rank(MPI_COMM_WORLD, &rank);

    char processor_name[MPI_MAX_PROCESSOR_NAME];
    int name_len;
    MPI_Get_processor_name(processor_name, &name_len);

    printf("Hello world from %s, rank %d"
           " out of %d processors\n",
           processor_name, rank, size);

    MPI_Finalize();

    return(0);
}
```

The status bar at the bottom shows 'mn3:' on the left and 'helloWorld : bash' on the right. The bottom right corner of the editor shows '1,1' and 'All'.

HelloWorld MPI Source code

Copy the files to your student accounts, compile the program and submit it to the execution queue.

- cp /home/nct00...
- make
- bsub < submit-mpi.sh

View the output files for messages from the batch system and the actual output from your program.

How many processes can you execute on one node and how can you influence this?

Run the application on a different number of processors and observe the name of the processor in the output file.

The MareNostrum Userguide can be downloaded at

<http://www.bsc.es/support/MareNostrum3-ug.pdf>