

# HPC Sample Job: MPI Program

[Expand all](#) [Collapse all](#)

## Overview

This document shows a very simple "Hello, World!"-type program using OpenMPI libraries, adapted from [MPI Tutorial: MPI Hello World](#).

Refer to the [Slurm Quick Start User Guide](#) for more information on Slurm scripts.

## C Source Code

```
mpi_hw.c

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

int main(int argc, char** argv) {
    MPI_Init(NULL, NULL);
    int world_size;
    MPI_Comm_size(MPI_COMM_WORLD, &world_size);
    int world_rank;
    MPI_Comm_rank(MPI_COMM_WORLD, &world_rank);
    char processor_name[MPI_MAX_PROCESSOR_NAME];
    int name_len;
    MPI_Get_processor_name(processor_name, &name_len);
    printf("Hello world from processor %s, rank %d"
          " out of %d processors\n",
          processor_name, world_rank, world_size);
    MPI_Finalize();
}
```

## Compiling

On the login node or a compute node, load the newest OpenMPI libraries and compile the source code to an executable with the `mpicc` script (which calls `gcc` with the correct flags to support MPI):

```
[renfro@login hw]$ module load openmpi/gcc/64/2.1.2
[renfro@login hw]$ mpicc -o mpi_hw mpi_hw.c
[renfro@login hw]$ ls -l
total 16
-rwx----- 1 renfro domain users 8792 Oct 10 09:52 mpi_hw
-rw----- 1 renfro domain users 481 Oct 4 09:55 mpi_hw.c
[renfro@login hw]$
```

## Running MPI Interactively

Request some cores on the interactive partition:

```
[renfro@login hw]$ hpcshell --ntasks=30
```

Run the MPI program with `mpirun` (abridged output shown):

```
[renfro@node002 hw]$ mpirun ./mpi_hw
Hello world from processor node004, rank 29 out of 30 processors
Hello world from processor node002, rank 18 out of 30 processors
...
Hello world from processor node002, rank 14 out of 30 processors
Hello world from processor node003, rank 28 out of 30 processors
[renfro@node002 hw]$
```

Exit from the interactive partition back to the login node:

```
[renfro@node002 hw]$ exit
[renfro@login hw]$
```

## Running MPI in Batch

Make a SLURM job script named `mpi_hw.sh`, reserving 56 cores for 1 minute, with the following contents:

### `mpi_hw.sh`

```
#!/bin/bash
#SBATCH --ntasks=56
#SBATCH --time=00:01:00
module load openmpi/gcc/64/2.1.2
mpirun ./mpi_hw
```

Submit the script with the `sbatch` command:

```
[renfro@login hw]$ sbatch mpi_hw.sh
Submitted batch job 4211
```

Once the job completes, examine the file `slurm-JOBID.out` (`slurm-4211.out` for this particular run):

```
[renfro@login hw]$ wc -l slurm-4211.out # line count, should have 1 line per task (56)
56 slurm-4211.out
[renfro@login hw]$ head -n 5 slurm-4211.out ; tail -5 slurm-4211.out # should show multiple hostnames
Hello world from processor node015, rank 16 out of 56 processors
Hello world from processor node015, rank 7 out of 56 processors
Hello world from processor node015, rank 14 out of 56 processors
Hello world from processor node015, rank 18 out of 56 processors
Hello world from processor node015, rank 19 out of 56 processors
Hello world from processor node022, rank 45 out of 56 processors
Hello world from processor node022, rank 52 out of 56 processors
Hello world from processor node022, rank 41 out of 56 processors
Hello world from processor node022, rank 55 out of 56 processors
Hello world from processor node022, rank 49 out of 56 processors
[renfro@login hw]$ awk '{print $5}' slurm-4211.out | sort | uniq -c # shows how many cores used on each host
    25 node015,
    28 node022,
     3 node025,
[renfro@login hw]$
```

**How helpful was this information?**



✦

Your Rating:  Results:  79 rates