

# Programming project 1

---

## Q1

1. First, login to Quanah cluster and start an interactive session by using “qlogin” command and request for following resources
  - a. Queue: omni
  - b. Project: quanah
  - c. Parallel Environment: Shared Memory with 2 CPU cores

Login with ssh command: **ssh [vinhtngu@quanah.hpcc.ttu.edu](mailto:vinhtngu@quanah.hpcc.ttu.edu)**

Start an interactive session: **quanah:\$ qlogin -P quanah -q omni -pe sm 2**

Where:

- P: cluster (or project) quanah
- Q: queue of omni
- Pe: environment of sm with 2 CPU cores

Output: **compute-8-14:\$**

2. Copy the directory /lustre/work/examples/quanah/hello-world/ to home folder

compute-8-14:\$ **cp -r /lustre/work/examples/quanah/hello-world/ ./**

3. Inside the “mpi” directory there is a “mpi\_hello\_world.c” file which contains a sample code of MPI programming in C language. Compile the code by using *mpicc* command once with GNU and once with Intel version of OpenMPI and IntelMPI (IMPI) and name the executable outputs as following:

**\$ mpicc -o output\_name program\_name.c**

---

---

#####

compute-8-14:/hello-world/mpi\$ **module load gnu7 openmpi**

compute-8-14:/hello-world/mpi\$ **mpicc -o mpi\_hello\_world-gnu-ompi  
mpi\_hello\_world.c**

compute-8-14:/hello-world/mpi\$ **module purge**

#####

#####

compute-8-14:/hello-world/mpi\$ **module load gnu7 impi**

compute-8-14:/hello-world/mpi\$ **mpicc -o mpi\_hello\_world-gnu-impi  
mpi\_hello\_world.c**

compute-8-14:/hello-world/mpi\$ **module purge**

#####

#####

compute-8-14:/hello-world/mpi\$ **module load intel openmpi**

compute-8-14:/hello-world/mpi\$ **mpicc -o mpi\_hello\_world-intel-ompi  
mpi\_hello\_world.c**

compute-8-14:/hello-world/mpi\$ **module purge**

#####

---

```
#####
```

```
compute-8-14:/hello-world/mpi$ module load intel impi
```

```
compute-8-14:/hello-world/mpi$ mpicc -o mpi_hello_world-intel-impi  
mpi_hello_world.c
```

```
compute-8-14:/hello-world/mpi$ module purge
```

```
#####
```

Then we have the following files in the system

```
compute-8-14:/hello-world/mpi$ ls  
mpi.sh                mpi_hello_world-intel-impi  
mpi_hello_world-gnu-impi  mpi_hello_world-intel-ompi  
mpi_hello_world-gnu-ompi  mpi_hello_world.c
```

4. Now try to execute any of the compiled outputs by calling `mpirun` command and using 2 CPU cores. (Please keep in mind that for each executable file, the right modules should be loaded into your environment.):

```
compute-8-14:/hello-world/mpi$ module purge
```

```
compute-8-14:/hello-world/mpi$ module load intel openmpi
```

```
compute-8-14:/hello-world/mpi$ mpirun -np 2 mpi_hello_world-intel-ompi
```

We have the following outputs:

```
Hello world from processor compute-8-14, rank 1 out of 2 processors
```

```
Hello world from processor compute-8-14, rank 0 out of 2 processors
```

5. Inside the `hello-world` directory you will see `openmp` and `pthread` directories as well. Try to compile the C codes inside these directories by using Intel compiler only:

Go back to the parent folder: **`cd ..`**

Go to `openmp` folder: **`cd openmp`**

---

Make sure to unload all module: ***module purge***

Load intel module only: ***module load intel/18.0.3.222***

Compile current c file: ***icc -o icc\_hello\_openmp-intel -qopenmp hello-openmp.c***

Run the compiled file: ***mpirun -np 2 ./icc\_hello\_openmp-intel***

Output:

*Hello World from thread = 1*

*Hello World from thread = 0*

*Number of threads = 4*

*Hello World from thread = 3*

*Hello World from thread = 2*

*Hello World from thread = 0*

*Number of threads = 4*

*Hello World from thread = 2*

*Hello World from thread = 1*

*Hello World from thread = 3*