

Homework #1

cpe 512

Kyle Ray

August 31, 2017

Contents

[Example Program Output 2](#_Toc491878476)

[MPI 2](#_Toc491878477)

[MPI2 2](#_Toc491878478)

[Pthread 2](#_Toc491878479)

[OpenMP 2](#_Toc491878480)

[Questions 3](#_Toc491878481)

[Hybrid 3](#_Toc491878482)

[Hybrid Source Code 3](#_Toc491878483)

[Output 4](#_Toc491878484)

[Appendix 4](#_Toc491878485)

# Example Program Output

Below are the results of running each example program in the DMC environment.

## MPI

uahcls01@dmcvlogin1:Hw1> mpirun hello\_world\_MPI

Hello World from MPI Process #0

Number of MPI Processes = 8

Hello World from MPI Process #1

Hello World from MPI Process #6

Hello World from MPI Process #7

Hello World from MPI Process #2

Hello World from MPI Process #3

Hello World from MPI Process #5

Hello World from MPI Process #4

## MPI2

uahcls01@dmcvlogin1:Hw1> mpirun hello\_world\_MPI2

Hello World from MPI Process #1

Hello World from MPI Process #3

Hello World from MPI Process #5

Hello World from MPI Process #6

Hello World from MPI Process #0

Number of MPI Processes = 8

Hello World from MPI Process #7

Hello World from MPI Process #4

Hello World from MPI Process #2

## Pthread

uahcls01@dmcvlogin1:Hw1> ./hello\_world\_PTH

Hello World, from PThread 0

Number of threads = 8

Hello World, from PThread 2

Hello World, from PThread 3

Hello World, from PThread 4

Hello World, from PThread 5

Hello World, from PThread 6

Hello World, from PThread 1

Hello World, from PThread 7

## OpenMP

uahcls01@dmcvlogin1:Hw1> ./hello\_world\_OMP

Hello World from thread = 0

Hello World from thread = 5

Hello World from thread = 3

Number of Threads = 8

Hello World from thread = 7

Hello World from thread = 4

Hello World from thread = 1

Hello World from thread = 2

Hello World from thread = 6

# Questions

Add program output here.

# Hybrid

## Hybrid Source Code

/\*

MPI/OpemMP - Hello World - C++ Version (utilizing C function calling Conventions)

FILE: hybrid.cpp

Compilation on dmc.asc.edu

first set up environment by typing from the command line

module load openmpi

to compile the program type

mpic++ -o hybrid -fopenmp hybrid.cpp

to run on eight processors type

mpirun -np 4 ./hybrid

\*/

// Kyle Ray

// CPE\_512\_Intro\_to\_Parallel\_Programming

// August 31, 2017

// Homework #1

// Hybrid version utilizing MPI and OpenMP

// Create 4 main MPI processes

// Each process will have two executing threads

using namespace std;

#include <iostream>

#include <mpi.h>

#include <omp.h>

int main (int argc, char \*argv[])

{

MPI\_Status status;

int nmtsks, rank;

MPI\_Init(&argc,&argv); // Initalize MPI environment

MPI\_Comm\_size(MPI\_COMM\_WORLD,&nmtsks); //get total number of processes

MPI\_Comm\_rank(MPI\_COMM\_WORLD,&rank); // get process identity number

// Create the parallel team of two threads for this MPI process

#pragma omp parallel num\_threads(2)

{

// Allow the output statements to finish

#pragma omp critical

{

cout << "Hello World from MPI Process #" << rank << " Thread # " << omp\_get\_thread\_num() << endl << flush;

}

}

// Only root MPI process does this

if (rank == 0)

{

cout << "Number of MPI Processes = " << nmtsks << endl;

}

/\* Terminate MPI Program -- clear out all buffers \*/

MPI\_Finalize();

}

## Output



# Appendix

Add anything else that might be pertinent to the assignment.