**OpenMp Code Running Commands**

**Windows System: Run in CMD**

1. **Install mingw:** <https://www.geeksforgeeks.org/installing-mingw-tools-for-c-c-and-changing-environment-variable/>
2. **Check the version:** g++ -v
3. **Update:** mingw-get upgrade --recursive
4. **Compile Run:**  g++ -fopenmp DEMO.cpp -o DEMOout
5. **Output:** ./DEMOout.exe or DEMOout.exe

**Demo Code:**

#include <stdio.h>

#include <omp.h>

int main() {

#pragma omp parallel num\_threads(3)

{

int id = omp\_get\_thread\_num();

int data = id;

int total = omp\_get\_num\_threads();

printf("Greetings from process %d out of %d with Data %d\n", id, total, data);

}

printf("parallel for ends.\n");

return 0;

}

**Output :**

C:\Users\ HPC\_Practical>demo.exe

Greetings from process 2 out of 3 with Data 2

Greetings from process 0 out of 3 with Data 0

Greetings from process 1 out of 3 with Data 1

parallel for ends.

**Linux System: Run in Terminal**

1. **Check the GCC version:** gcc –version
2. **Installation :** sudo apt install gcc
3. **Configuring OpenMP:** echo |cpp -fopenmp -dM |grep -i open **If not :** sudo apt install libomp-dev
4. **Setting the number of threads:** export OMP\_NUM\_THREADS=8
5. **Compile:** gcc -o demo -fopenmp demo.c
6. **Execute:** ./demo

**Demo Code:**

// OpenMP header

#include <omp.h>

#include <stdio.h>

#include <stdlib.h>

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

{

int nthreads, tid;

// Begin of parallel region

#pragma omp parallel private(nthreads, tid)

{

// Getting thread number

tid = omp\_get\_thread\_num();

printf("Welcome to GFG from thread = %d\n",tid);

if (tid == 0) {

// Only master thread does this

nthreads = omp\_get\_num\_threads();

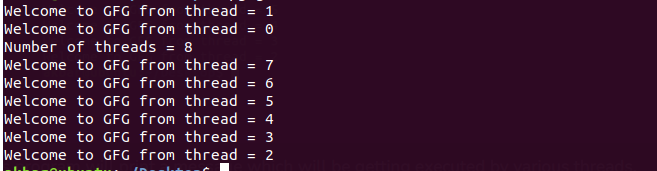
printf("Number of threads = %d\n",nthreads);

}

}

}

**Output :**

****