

Name : Usman Manzoor

Roll no: p190068

Section : 6B

Subject: PDC

Assignment : Open MP and MPI

MPI FILE:

```
#include <mpi.h>
#include <stdio.h>
int main(int argc, char** argv)
{
    // Initialize the MPI environment
    MPI_Init( NULL, NULL ) ;

    // Get the number of processes ;
    int world_size;
    MPI_Comm_size( MPI_COMM_WORLD, &world_size );

    // Get the rank of the process
    int world_rank;
    MPI_Comm_rank( MPI_COMM_WORLD, &world_rank );

    // Get the name of the processor

    char processor_name[MPI_MAX_PROCESSOR_NAME];
```

```
int name_len;
```

```
MPI_Get_processor_name( processor_name, &name_len );
```

```
//1 Print off a hello world message
```

```
printf("hello world from processer %s, rank %d out of %d processors\n", processor_name,  
world_rank, world_size);
```

```
MPI_Finalize();
```

```
}
```

```
usman@usman-VirtualBox:~/Downloads$ mpicc tah.c -o tah  
usman@usman-VirtualBox:~/Downloads$ mpirun -np 4 ./tah  
hello world from processer usman-VirtualBox, rank 1 out of 4 processors  
hello world from processer usman-VirtualBox, rank 3 out of 4 processors  
hello world from processer usman-VirtualBox, rank 0 out of 4 processors  
hello world from processer usman-VirtualBox, rank 2 out of 4 processors
```

```
usman@usman-VirtualBox:~/Downloads$ mpirun -np 8 --oversubscribe ./tah  
hello world from processer usman-VirtualBox, rank 7 out of 8 processors  
hello world from processer usman-VirtualBox, rank 3 out of 8 processors  
hello world from processer usman-VirtualBox, rank 0 out of 8 processors  
hello world from processer usman-VirtualBox, rank 2 out of 8 processors  
hello world from processer usman-VirtualBox, rank 6 out of 8 processors  
hello world from processer usman-VirtualBox, rank 1 out of 8 processors  
hello world from processer usman-VirtualBox, rank 5 out of 8 processors  
hello world from processer usman-VirtualBox, rank 4 out of 8 processors
```

OPEN MP:

HELOO.C

```
#include<stdio.h>
```

```
#include<omp.h>
```

```
int main()
```

```
{
```

```
    int ID;
```

```

#pragma omp parallel
{
    ID = omp_get_thread_num();
    printf("hello%d ",ID);
    printf("world %d\n",ID);
}

return 0;
}

```

```

(base) mohsin@mohsin-HP-ProBook:~/semester6/PDC/openMP$ gcc -fopenmp hello.c
(base) mohsin@mohsin-HP-ProBook:~/semester6/PDC/openMP$ ./a.out
hello0 world 3
hello2 world 3
hello3 world 3
hello1 world 3

```

Critical.c

```

#include<stdio.h>
#include<omp.h>

void main()
{
    int x=0;

#pragma omp parallel num_threads(300)
    {
        x=x+1;
    }

    printf("x=%d\n",x);
}

```

```

(base) mohsin@mohsin-HP-ProBook:~/semester6/PDC/openMP$ gcc -fopenmp critical.c
(base) mohsin@mohsin-HP-ProBook:~/semester6/PDC/openMP$ ./a.out
x=299
(base) mohsin@mohsin-HP-ProBook:~/semester6/PDC/openMP$ ./a.out
x=299
(base) mohsin@mohsin-HP-ProBook:~/semester6/PDC/openMP$ ./a.out
x=299
(base) mohsin@mohsin-HP-ProBook:~/semester6/PDC/openMP$ ./a.out
x=299
(base) mohsin@mohsin-HP-ProBook:~/semester6/PDC/openMP$ ./a.out
x=297
(base) mohsin@mohsin-HP-ProBook:~/semester6/PDC/openMP$ ./a.out
x=300
(base) mohsin@mohsin-HP-ProBook:~/semester6/PDC/openMP$ ./a.out
x=300
(base) mohsin@mohsin-HP-ProBook:~/semester6/PDC/openMP$ ./a.out
x=298

```

Sync.c

```

#include<stdio.h>
#include<omp.h>

void main()
{
    int x=0;

    omp_lock_t writelock; // omp_lock_t is data type used to declare lock variable.

    omp_init_lock(&writelock); // Function omp_init_lock() is used to initialize
    lock.

                                // By default, lock is unset.

    #pragma omp parallel num_threads(300)
    {
        omp_set_lock(&writelock); //omp_set_lock() is used to set lock
        x=x+1;
        omp_unset_lock(&writelock);
    }

    printf("x=%d\n",x);

    omp_destroy_lock(&writelock);
}

```

[illegible]