**MPI PROGRAM**

#include <stdio.h>

#include <mpi.h>

int main(int argc, char\*\* argv) {

int rank, size;

MPI\_Init(&argc, &argv);

MPI\_Comm\_rank(MPI\_COMM\_WORLD, &rank);

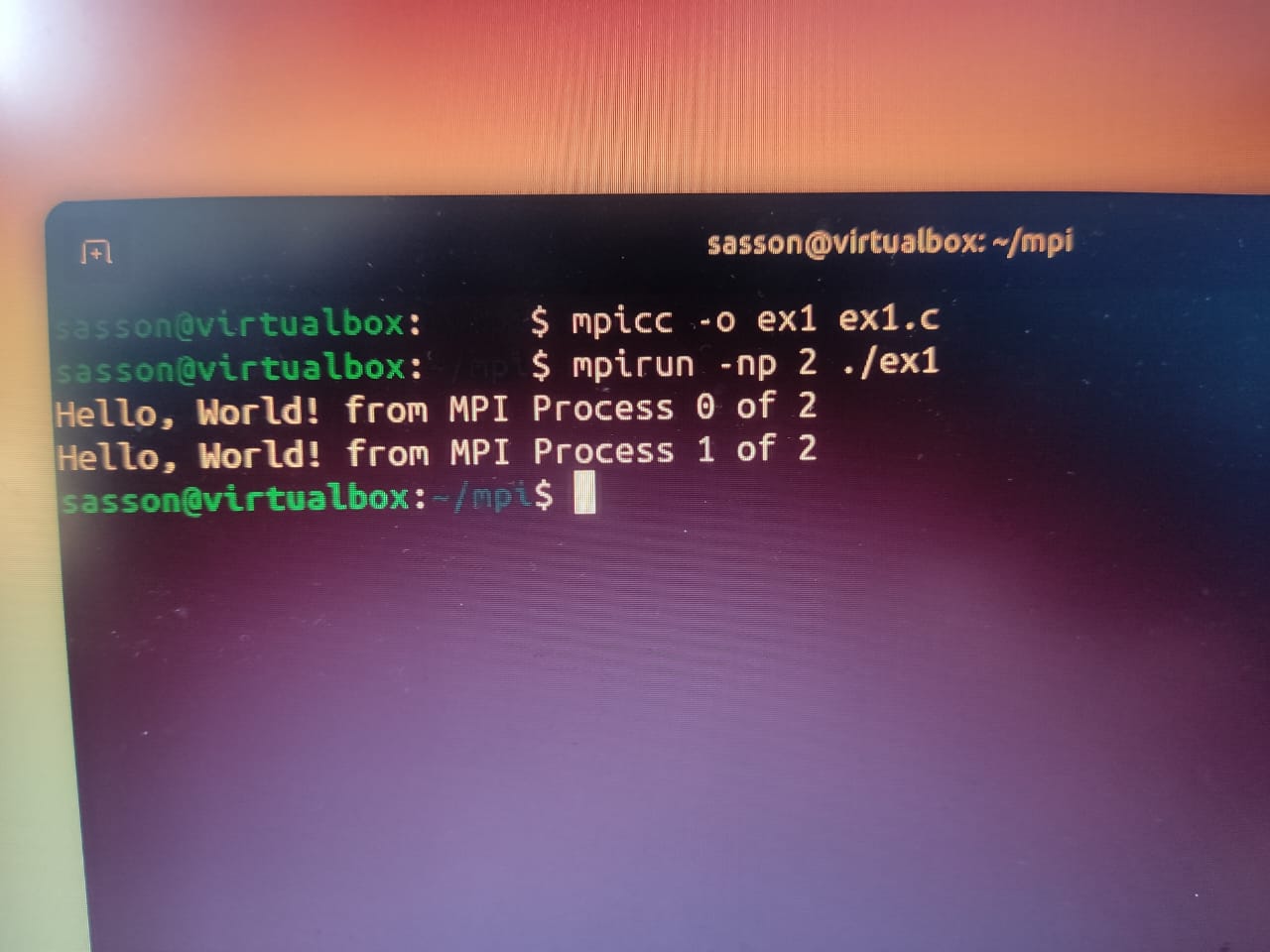
MPI\_Comm\_size(MPI\_COMM\_WORLD, &size);

printf("Hello, World! from MPI Process %d of %d\n", rank, size);

MPI\_Finalize();

return 0;

}

****

**MPI Program to find smallest between 2 nos and implement bcast constraint in the program**

Program :

#include <stdio.h>

#include <mpi.h>

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

MPI\_Init(&argc, &argv);

int rank, size;

MPI\_Comm\_rank(MPI\_COMM\_WORLD, &rank);

MPI\_Comm\_size(MPI\_COMM\_WORLD, &size);

double start\_time, end\_time;

if (rank == 0) {

int num1 = 15;

int num2 = 8;

start\_time = MPI\_Wtime();

MPI\_Bcast(&num1, 1, MPI\_INT, 0, MPI\_COMM\_WORLD);

MPI\_Bcast(&num2, 1, MPI\_INT, 0, MPI\_COMM\_WORLD);

printf("Process %d broadcasting numbers: %d and %d\n", rank, num1, num2);

} else {

int my\_num1, my\_num2;

MPI\_Bcast(&my\_num1, 1, MPI\_INT, 0, MPI\_COMM\_WORLD);

MPI\_Bcast(&my\_num2, 1, MPI\_INT, 0, MPI\_COMM\_WORLD);

printf("Process %d received numbers: %d and %d\n", rank, my\_num1, my\_num2);

int smallest = (my\_num1 < my\_num2) ? my\_num1 : my\_num2;

MPI\_Reduce(&smallest, NULL, 1, MPI\_INT, MPI\_MIN, 0, MPI\_COMM\_WORLD);

}

if (rank == 0) {

end\_time = MPI\_Wtime();

printf("Elapsed time: %f seconds\n", end\_time - start\_time);

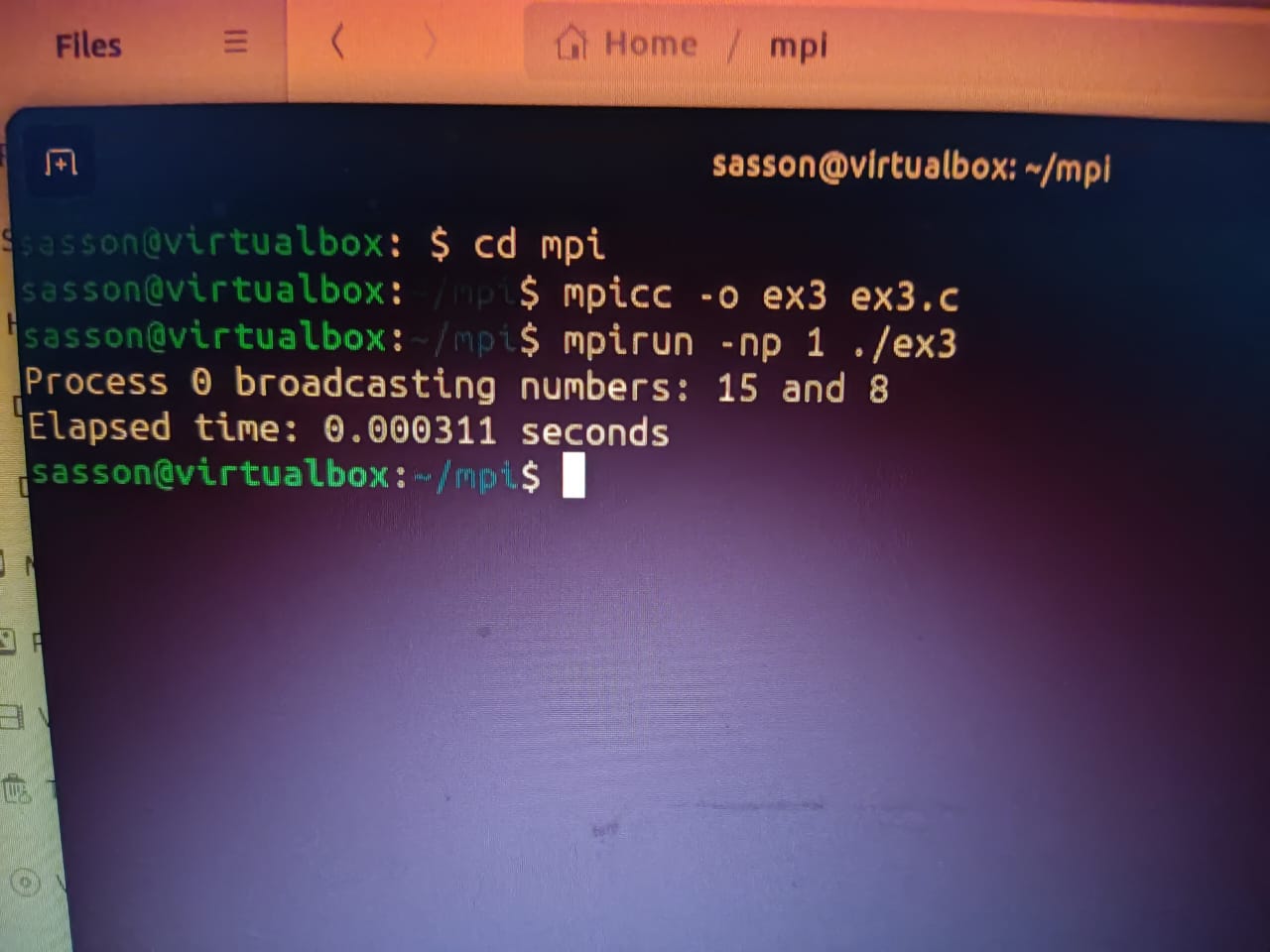
}

MPI\_Finalize();

return 0;

}

**Output**

****

**OPENMP PROGRAM**

**PROGRAM :**

#include <stdio.h>

#include <omp.h>

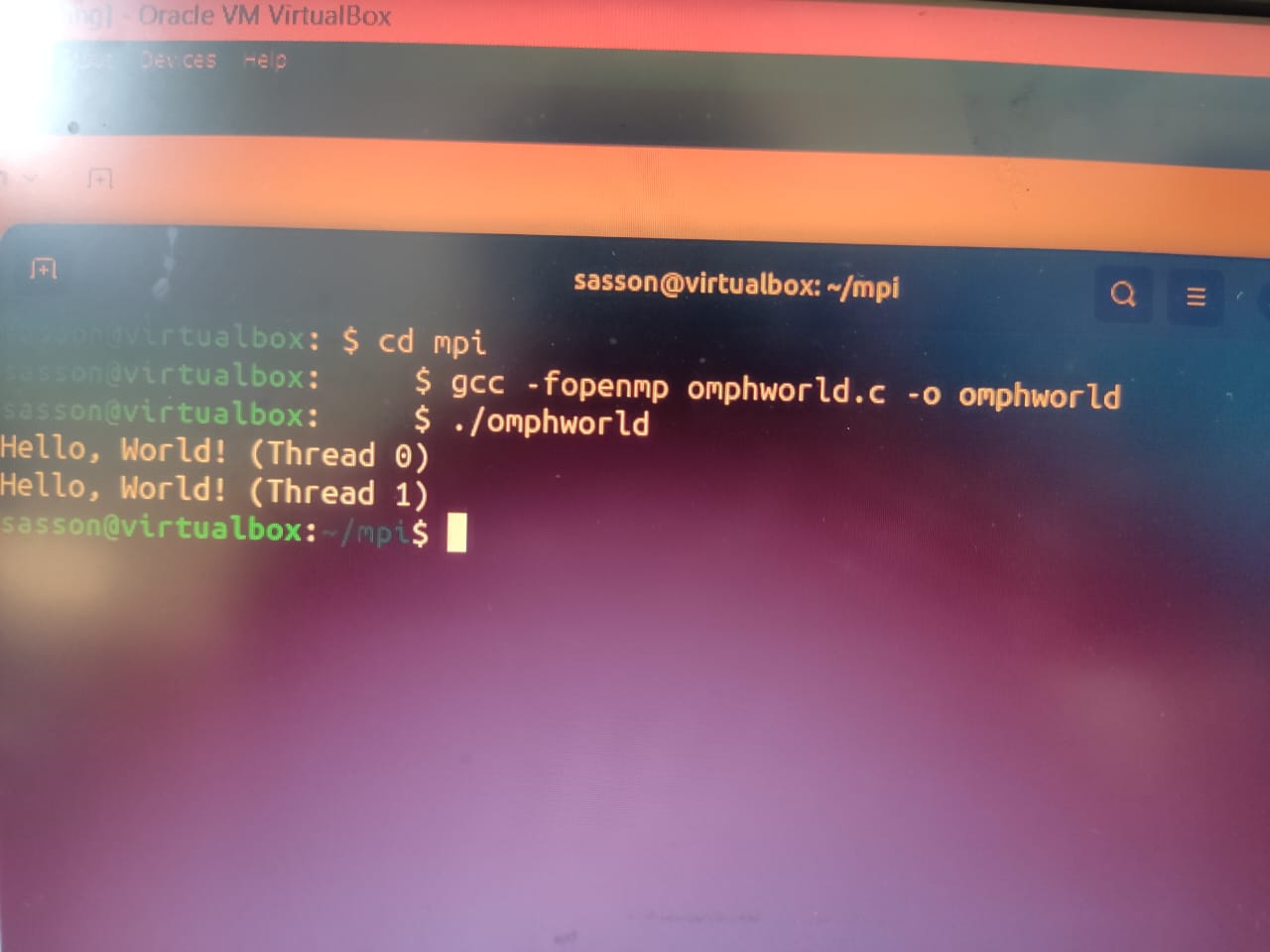
int main(){

{

int thread\_id = omp\_get\_thread\_num();

printf("Hello, World! (Thread %d)\n", thread\_id);}

return 0;

}