

## HDT #2

Nombre: Gabriel Alejandro Vicente Lorenzo Carné: 20498

**Fecha de Entrega:** 26 de septiembre, 2023.

**Descripción:** En esta hoja de trabajo resolverá un conjunto de ejercicios creando soluciones paralelas de un programa secuencial mediante el uso de MPI.

**Entregables:** Deberá entregar un documento con las respuestas a las preguntas planteadas en cada ejercicio (incluyendo diagramas o screenshots si es necesario), junto con todos los archivos de código que programe debidamente comentados e identificados. La entrega de la hoja es individual.

**Materiales:** necesitará una máquina virtual con Linux.

**Contenido:**

**Instalación de MPI:**

**Linux/WSL:**

Alguna versión de MPI viene instalada por defecto en muchas de las distribuciones de Linux. Verifique si ya tiene instalado MPI en su sistema mediante:

```
$ mpicc -v
```

Si no tiene instalado MPI en Linux o WSL, instálelo mediante:

```
$ sudo apt-get install openmpi*  
$ sudo apt install openmpi-bin  
$ sudo apt install mpich
```

Si les da error en la instalación, es por la ausencia de los archivos para compilar Fortran. Pueden resolver esto actualizando los paquetes luego de la falla:

```
$ sudo apt-get update  
$ sudo apt-get upgrade
```

**Ejercicio 1 (20 puntos)**

Cree un archivo llamado `mpi_hello.c`. En este archivo cree un programa que, utilizando la librería y API de MPI, imprima un mensaje “Hello World, I’m process # of #!” desde distintos procesos. La cantidad de procesos deberá ser ingresada desde línea de comando al momento de ejecutar su programa. Se le recomienda utilizar 4 procesos para sus pruebas. Se deberá producir un resultado como el siguiente:

```
Hello world!, I'm process 0 of 4
Hello world!, I'm process 1 of 4
Hello world!, I'm process 2 of 4
Hello world!, I'm process 3 of 4
```

**\*\* Recuerde que es posible que los mensajes que usted imprime en consola no se vean en el orden “correcto” \*\***

*Imagen #1 Captura de pantalla del código creado y resultado de ejecución*

The screenshot shows the Visual Studio Code editor with a C file named `mpi_hello.c`. The code is as follows:

```
1 // Paralela
2 // HDT #2
3 // Gabriel Vicente 20498
4 #include <stdio.h>
5 #include <mpi/mpi.h>
6
7 int main(int argc, char** argv) {
8     int rank, size;
9
10    // Inicialización de MPI
11    MPI_Init(&argc, &argv);
12    MPI_Comm_rank(MPI_COMM_WORLD, &rank);
13    MPI_Comm_size(MPI_COMM_WORLD, &size);
14
15    // Verifica el número de argumentos de línea de comandos
16    if (argc != 2) {
17        if (rank == 0) {
18            printf("Uso: mpxec -n <num_procesos> %s <num_procesos>\n", argv[0]);
19        }
20        MPI_Finalize();
21        return 1;
22    }
23
24    int num_procesos;
25    sscanf(argv[1], "%d", &num_procesos);
26
27    // Verifica que el número de procesos coincida con el especificado
28    if (num_procesos != size) {
29        if (rank == 0) {
30            printf("El número de procesos especificado debe coincidir con -n en mpxec.\n");
31        }
32        MPI_Finalize();
33        return 1;
34    }
35
36    // Imprime un mensaje con el número de proceso y el tamaño del grupo
37    printf("Hello World, I'm process %d of %d!\n", rank, size);
38
39    // Finalización de MPI
40    MPI_Finalize();
41    return 0;
42 }
```

The terminal output shows the execution of the program with 3 processes:

```
gabo@Ubuntu:~/Desktop/Paralela/HDT2$ mpicc -o mpi_hello mpi_hello.c
gabo@Ubuntu:~/Desktop/Paralela/HDT2$ mpxec -n 3 ./mpi_hello 3
Hello World, I'm process 0 of 3!
Hello World, I'm process 1 of 3!
Hello World, I'm process 2 of 3!
```

*Imagen #2 Resultado de ejecución*

```
gabo@Ubuntu:~/Desktop/Paralela/HDT2$ mpicc -o mpi_hello mpi_hello.c
gabo@Ubuntu:~/Desktop/Paralela/HDT2$ mpxec -n 3 ./mpi_hello 3
Hello World, I'm process 0 of 3!
Hello World, I'm process 1 of 3!
Hello World, I'm process 2 of 3!
```

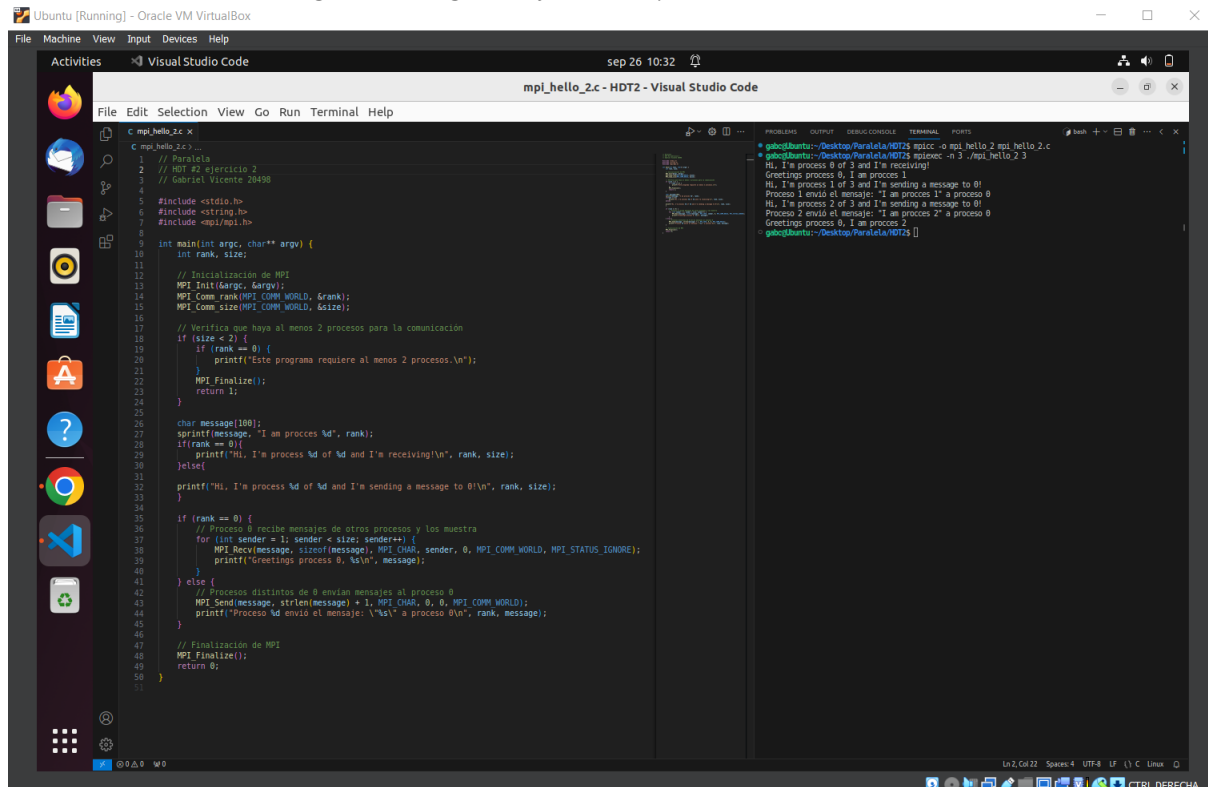
*Se utilizaron 3 debido a que esa es la cantidad de cores que posee la máquina virtual*

## Ejercicio 2 (40 puntos)

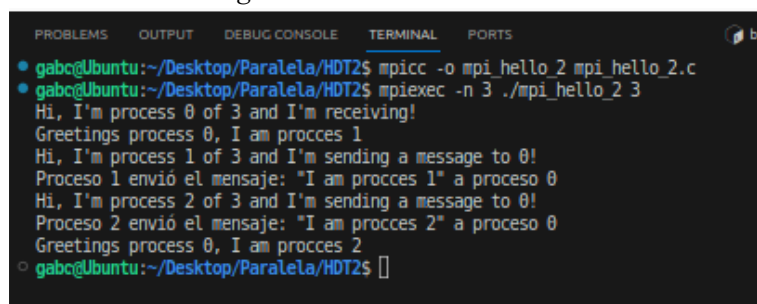
Cree un archivo llamado `mpi_hello_2.c` que sea copia del archivo del ejercicio 1. Modifique su código para que todos los procesos que envíen al proceso 0 un mensaje y que su proceso 0 sea el que reciba todos los mensajes y los despliega en pantalla. Cada proceso debe desplegar en consola quién es y qué acción está realizando. Los procesos que envían un mensaje deben indicar a quién envían y desde qué proceso envían el mensaje. Se deberá producir un resultado como el siguiente:

```
Hi, I'm process 1 of 4 and i'm sending a message to 0!
Hi, I'm process 2 of 4 and i'm sending a message to 0!
Hi, I'm process 3 of 4 and i'm sending a message to 0!
Hi, I'm process 0 of 4 and i'm receiving!
Greetings process 0, I'm process 1 of 4!
Greetings process 0, I'm process 2 of 4!
Greetings process 0, I'm process 3 of 4!
```

Imagen#3 Código del ejercicio 2 y resultados en terminal



Imagen#4 Resultados en terminal

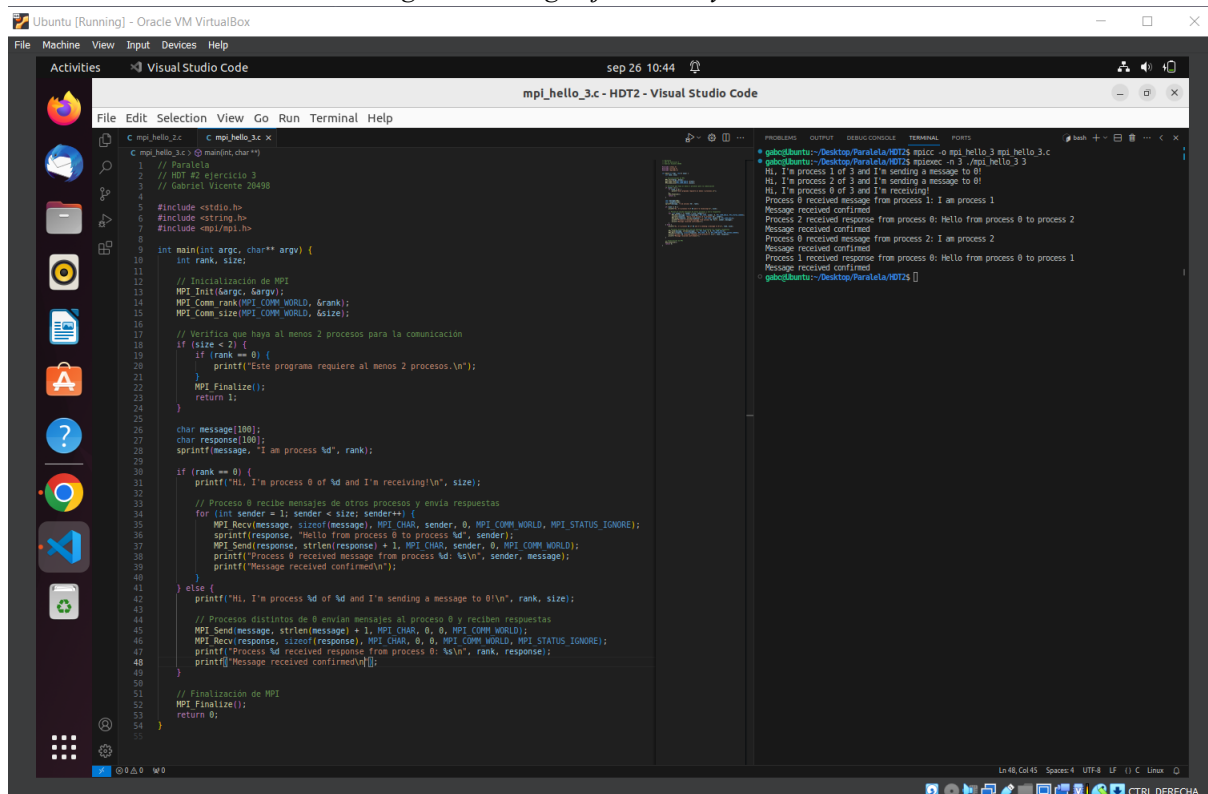


### Ejercicio 3 (40 puntos)

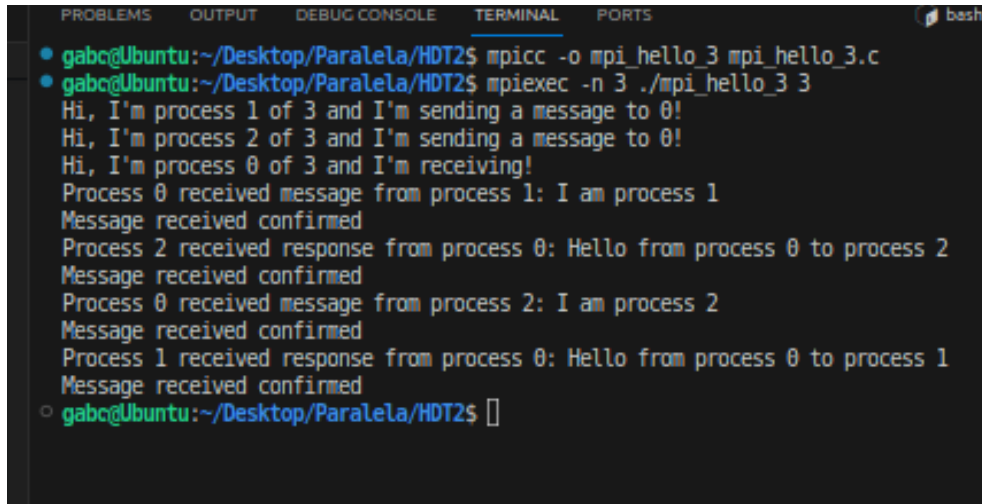
Cree un archivo llamado `mpi_hello_3.c` que sea copia del archivo del ejercicio 2. Una buena comunicación no estaría completa si no existe una confirmación de recepción. Modifique su código para que todos los procesos que envían un mensaje al proceso 0 reciban un mensaje de respuesta de parte del proceso 0. Cada proceso debe desplegar en consola quién es y qué acción está realizando. Los procesos que envían un mensaje deben de indicar a quien envían y desde qué proceso envían el mensaje. Se deberá producir un resultado como el siguiente:

```
Hi, I'm process 1 of 4 and I'm sending!
Hi, I'm process 2 of 4 and I'm sending!
Hi, I'm process 3 of 4 and I'm sending!
Hi, I'm process 0 of 4 and i'm receiving!
Sending message from process 1 of 4 to process 0!
Hi, I'm process 0 of 4 and i'm sending a response to 1!
Sending message from process 2 of 4 to process 0!
Sending response from process 0 to process 1!
Hi, I'm process 0 of 4 and i'm sending a response to 2!
Sending response from process 0 to process 2!
Sending message from process 3 of 4 to process 0!
Hi, I'm process 0 of 4 and i'm sending a response to 3!
Sending response from process 0 to process 3!
```

*Imagen #5 Código ejercicio 3 y resultados*



*#Imagen #6 Resultados en terminal*



```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS  bash
• gabc@Ubuntu:~/Desktop/Paralela/HDT2$ mpicc -o mpi_hello_3 mpi_hello_3.c
• gabc@Ubuntu:~/Desktop/Paralela/HDT2$ mpiexec -n 3 ./mpi_hello_3 3
Hi, I'm process 1 of 3 and I'm sending a message to 0!
Hi, I'm process 2 of 3 and I'm sending a message to 0!
Hi, I'm process 0 of 3 and I'm receiving!
Process 0 received message from process 1: I am process 1
Message received confirmed
Process 2 received response from process 0: Hello from process 0 to process 2
Message received confirmed
Process 0 received message from process 2: I am process 2
Message received confirmed
Process 1 received response from process 0: Hello from process 0 to process 1
Message received confirmed
○ gabc@Ubuntu:~/Desktop/Paralela/HDT2$
```