

UNIVERSIDAD DEL VALLE DE GUATEMALA

FACULTAD DE INGENIERÍA
CC3069 COMPUTACIÓN PARALELA Y
DISTRIBUIDA

Hoja de Trabajo 2

Introducción al uso y conceptos de MPI

Por: Samuel A. Chamalé
Email: cha21881@uvg.edu.gt

Fecha de entrega: 24/09/2024

Índice

1. Repositorio	1
2. Introducción	1
3. Ejercicio 1 [mpi_hello]	1
3.1. Captura de Pantalla	1
4. Ejercicio 2 [mpi_hello_2]	2
4.1. Modificaciones Clave	2
4.2. Captura de Pantalla	3
5. Ejercicio 3 [mpi_hello_3]	3
5.1. Modificaciones Clave	3
5.2. Captura de Pantalla	4
6. Conclusiones	4

1. Repositorio

<https://github.com/chamale-rac/mpi-seq2par>

2. Introducción

En este reporte se detallan las soluciones implementadas para los ejercicios 1, 2 y 3 relacionados con la programación paralela utilizando MPI (Message Passing Interface). A lo largo del documento, se muestran capturas de pantalla y diagramas relevantes, que ilustran el funcionamiento de los programas. Cada ejercicio tiene su código debidamente comentado e identificado, siguiendo las instrucciones de la tarea.

3. Ejercicio 1 [mpi_hello]

En este ejercicio se implementó un programa que imprime un mensaje “Hello World, I’m process # of #!” desde distintos procesos. El programa fue probado utilizando 4 procesos para verificar su correcto funcionamiento.

3.1. Captura de Pantalla

A continuación se muestra una captura de pantalla del resultado obtenido al ejecutar el programa con 4 procesos.

```
☑ run on [GNU/Linux] at [2024-09-24 18:01:53]
➡ schr mpi-seq2par X mpirun -np 4 ./mpi_hello
Hello World, I'm process 1 of 4!
Hello World, I'm process 2 of 4!
Hello World, I'm process 3 of 4!
Hello World, I'm process 0 of 4!
☑ run on [GNU/Linux] at [2024-09-24 18:01:56]
➡ schr mpi-seq2par X mpirun -np 4 ./mpi_hello
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!
☑ run on [GNU/Linux] at [2024-09-24 18:01:58]
➡ schr mpi-seq2par X mpirun -np 4 ./mpi_hello
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!
☑ run on [GNU/Linux] at [2024-09-24 18:02:00]
➡ schr mpi-seq2par X █
```

Figura 1: Resultado del Ejercicio 1.

4. Ejercicio 2 [mpi_hello_2]

En el segundo ejercicio, se modificó el código para que todos los procesos envíen un mensaje al proceso 0, y este sea el que reciba y despliegue todos los mensajes.

4.1. Modificaciones Clave

Para este ejercicio, las modificaciones clave incluyen el uso de las funciones `MPI_Send` y `MPI_Recv` para enviar y recibir mensajes entre los procesos. A continuación se destacan las partes del código donde se realizaron estos cambios:

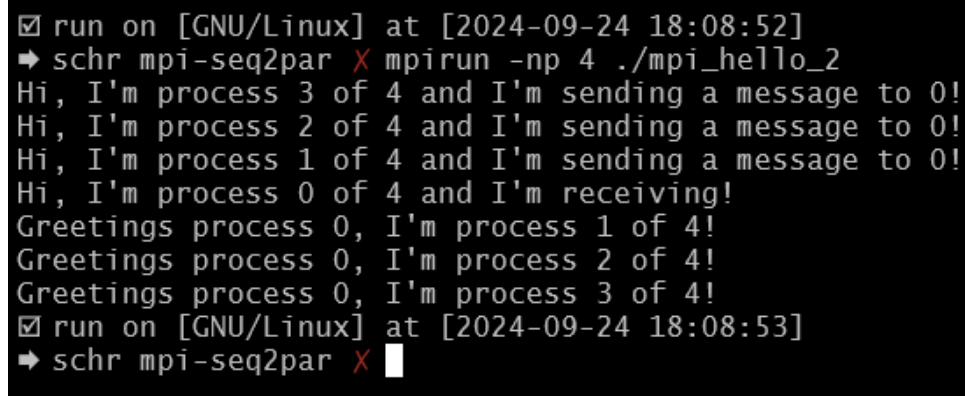
```
// Código de envío en los procesos diferentes al 0
MPI_Send(message, 100, MPI_CHAR, 0, 0, MPI_COMM_WORLD);
```

```
// Código de recepción en el proceso 0
MPI_Recv(message, 100, MPI_CHAR, i, 0, MPI_COMM_WORLD, MPI_STATUS_IGNORE);
```

El proceso 0 utiliza un bucle para recibir mensajes de todos los demás procesos, mientras que los demás procesos envían un mensaje a este.

4.2. Captura de Pantalla

La siguiente captura muestra la salida del programa modificado al enviar mensajes al proceso 0.



```
[x] run on [GNU/Linux] at [2024-09-24 18:08:52]
-> schr mpi-seq2par X mpirun -np 4 ./mpi_hello_2
Hi, I'm process 3 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 1 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!
[x] run on [GNU/Linux] at [2024-09-24 18:08:53]
-> schr mpi-seq2par X [ ]
```

Figura 2: Resultado del Ejercicio 2.

5. Ejercicio 3 [mpi_hello_3]

En el tercer ejercicio, se añadió la funcionalidad de que el proceso 0 envíe un mensaje de respuesta de confirmación a cada proceso que le haya enviado un mensaje previamente.

5.1. Modificaciones Clave

Las modificaciones clave para este ejercicio incluyen que el proceso 0, después de recibir un mensaje, envíe una respuesta de vuelta al proceso que originó la comunicación. Las funciones `MPI_Send` y `MPI_Recv` se utilizan tanto para enviar como para recibir mensajes de confirmación:

```
// Proceso 0 enviando respuesta de confirmación
```

```
sprintf(response, "Response from process 0 to process %d", i);
```

```
MPI_Send(response, 100, MPI_CHAR, i, 0, MPI_COMM_WORLD);
```

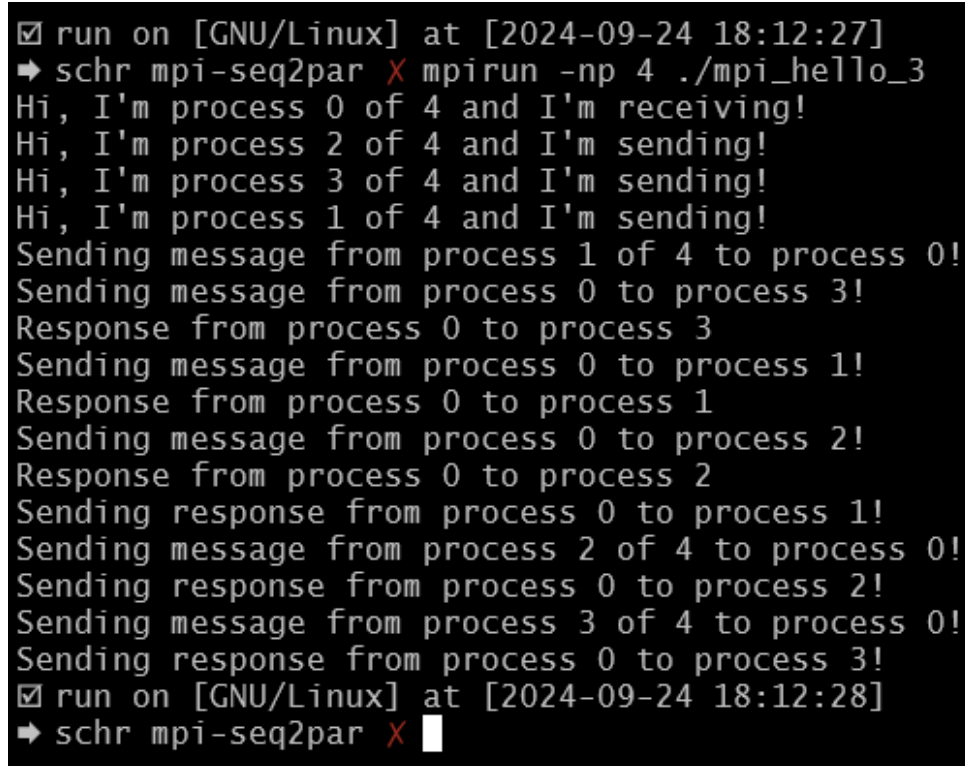
```
// Otros procesos recibiendo la respuesta
```

```
MPI_Recv(response, 100, MPI_CHAR, 0, 0, MPI_COMM_WORLD, MPI_STATUS_IGNORE);
```

Esta modificación garantiza una comunicación completa en ambas direcciones: primero el envío y luego la confirmación de la recepción.

5.2. Captura de Pantalla

En la siguiente figura se puede observar el resultado al ejecutar el programa del Ejercicio 3.



```
[x] run on [GNU/Linux] at [2024-09-24 18:12:27]
-> schr mpi-seq2par X mpirun -np 4 ./mpi_hello_3
Hi, I'm process 0 of 4 and I'm receiving!
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 1 of 4 and I'm sending!
Sending message from process 1 of 4 to process 0!
Sending message from process 0 to process 3!
Response from process 0 to process 3
Sending message from process 0 to process 1!
Response from process 0 to process 1
Sending message from process 0 to process 2!
Response from process 0 to process 2
Sending response from process 0 to process 1!
Sending message from process 2 of 4 to process 0!
Sending response from process 0 to process 2!
Sending message from process 3 of 4 to process 0!
Sending response from process 0 to process 3!
[x] run on [GNU/Linux] at [2024-09-24 18:12:28]
-> schr mpi-seq2par X
```

Figura 3: Resultado del Ejercicio 3.

6. Conclusiones

A través de los ejercicios planteados, se han desarrollado habilidades esenciales para la implementación de programas paralelos utilizando MPI. Se exploraron aspectos clave de la comunicación entre procesos, como el envío y recepción de mensajes, y se implementó una confirmación de recepción de mensajes por parte del proceso central.

Referencias

- [Mpi] *Using MPI with C — Research Computing, University of Colorado Boulder documentation*. <https://curc.readthedocs.io/en/latest/programming/MPI-C.html>. Accessed: 2024-09-24. n.d.

Índice de figuras

1.	Resultado del Ejercicio 1.	2
2.	Resultado del Ejercicio 2.	3
3.	Resultado del Ejercicio 3.	4