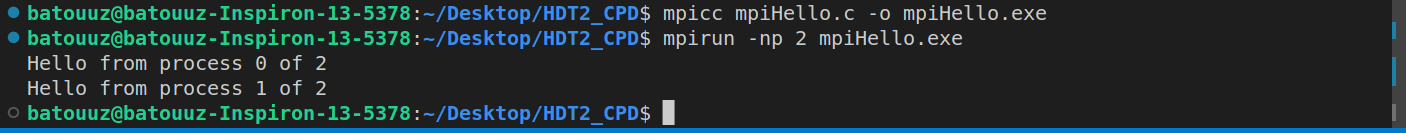
**Hoja de Trabajo 2: OpenMPI y Computación Distribuida**

**Parte 3:**

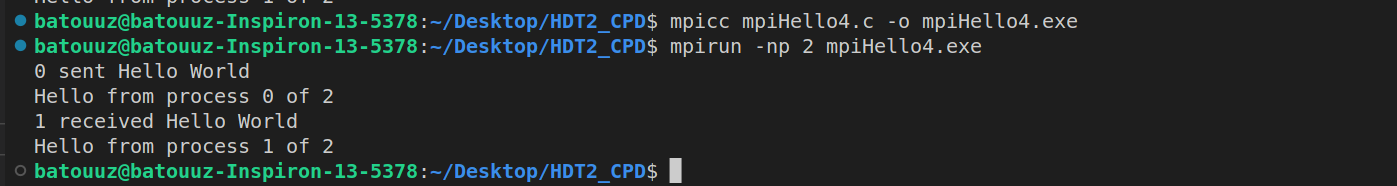
En la siguiente imagen se observa el output del programa sin modificar:



Se observa como cada uno de los procesos imprime en pantalla el mensaje, se aclara que aquí se usaron dos debido a que la máquina solamente cuenta con dos núcleos.

**Parte 4:**

Luego se observa al correrlo con el código proporcionado. Aquí se observa cómo es que se comunican dos procesos, en el que el proceso con rank 0 le envía un mensaje al rank 1 y luego este mismo lo imprime.

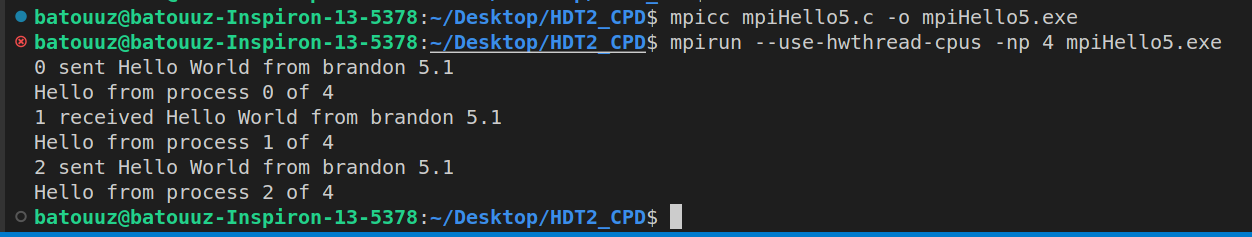




Es interesante como es que este tipo de comunicación se parece a la comunicación de procesos por medio de un socket, puede que por detrás lo este usando y con las funciones de MPI\_Send y MPI\_Recv se realice la lógica de estos.

**Parte 5:**

Aquí se observa que solamente tres procesos lograron llegar al final de programa, este se quedo corriendo y se tuvo que forzar el cierre de la corrida.

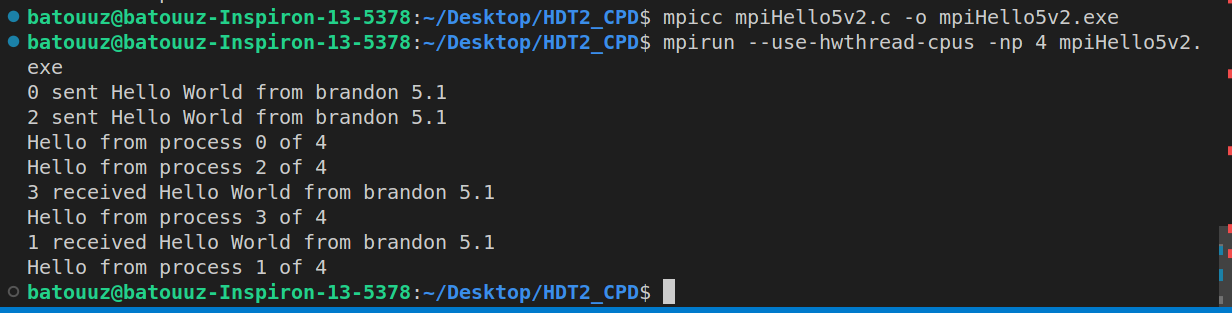




Lo que se observa es que el ultimo proceso que esta escuchando se queda esperando el mensaje, aunque se le indico al proceso que se lo enviara al proceso con un id mayor por 1.

**Parte 5.2:**

Aquí se observa como los cuatro procesos lograron terminar y no se tuvo que salir de manera prematura del programa.





Lo que estaba ocurriendo es que se estaba mandando el mensaje, pero nunca se le indico al rank impar que tiene que escuchar al proceso que tiene un id menor al de él por uno. Que es lo que se hizo en este caso.

Esta es una manera bastante interesante de realizar la comunicación entre procesos. Porque la manera a la que estoy acostumbrado es a abrir dos terminales y programar un montón de lógica para poder hacer esta comunicación, en lo personal ha sido por medio de sockets.