

Hoja de Trabajo # 2

Fecha de Entrega: 29 de septiembre, 2024.

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
```

MacOS:

- Tener instalado Xcode y Xcode cmd tools

```
$ xcode-select --install
```

- Tener instalado el compilador GNU o el compilador Intel

- Descargar el paquete tar.gz desde el sitio de OpenMPI . **No olvide revisar que obtenga la versión más reciente.**

- <https://download.open-mpi.org/release/open-mpi/v4.1/openmpi-4.1.5.tar.gz>

- Copiar a una ubicación temporal y desempacar el archivo

```
$ tar zxvf openmpi-4.1.5.tar.gz
```

- **(Este proceso es tardado)** Navegar al directorio creado y/o buscar el script de instalación. Especificar el compilador instalado y si se desea cambiar el directorio de instalación, modificar la bandera --prefix:

```
$ ./configure CC=gcc-11 CXX=g++-11 F77=gfortran FC=gfortran --  
prefix=/usr/local
```

- **(Este proceso es tardado x2)** Si la creación de configuración es exitosa, compilar el código:

```
$ make all
```

- Crear y ejecutar la instalación:

```
$ sudo make install
```

- Verificar que el directorio de instalación esté en el PATH de la máquina:

```
$ echo $PATH
```

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" ****

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 despliegue en pantalla.

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 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!
```

Ejercicio 2 (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!
```