# Arquitectura de Computadores

## Grado en Ingeniería Informática

## Práctica 1

UNIVERSIDAD NEBRIJA

# Fundamentos básicos de paralelización: Paso de mensajes

## Introducción

En esta práctica se aprenderán los aspectos generales de la programación en MPI así como el entorno sobre el que se desarrollarán las sesiones de prácticas.

Los objetivos fijados son los siguientes:

- Preparación del entorno de trabajo: instalación MPI.
- Compilación de programas MPI.
- Ejecución de programas en varios procesos de forma paralela.
- Estructura de un programa MPI.
- Iniciar y finalizar el entorno MPI con MPI_Init y MPI_Finalize.
- Identificador (rango) del proceso con MPI_Comm_rank.
- Consultar el número de procesos lanzados con MPI_Comm_size.
- Primitivas send y recieve.

## Instalación de MPICH

Previamente a la instalación, es necesario actualizar los paquetes de nuestra distribución:

```
sudo apt update
```

A continuación es necesario tener instalado un compilador de C:

```
sudo apt install gcc
gcc –version
```

Por último se instala MPICH:

```
sudo apt install mpich
mpiexec –versión
```

Toda la documentación de instalación y uso se puede encontrar en la página oficial del proyecto: https://www.mpich.org/

- Guía Instalación: https://www.mpich.org/static/downloads/3.3.1/mpich-3.3.1-installguide.pdf

- Guía Usuario: https://www.mpich.org/static/downloads/3.3.1/mpich-3.3.1-userguide.pdf

## Compilación y ejecución de programas MPI

- Compilación: **`mpicc codigo_fuente.c -o ejecutable`**
- Ejecución: **`mpirun –np <number> ejecutable`**

## Entregables

- Memoria
- Código fuente de los siguientes ejercicios:

## Ejercicio 1 (2 Puntos)

Implementar un programa usando MPI, que imprima por salida estándar:

**"Hola mundo, soy el proceso X de un total de Y."**

cuando el número total de tareas es Y y X un rango de 0 a Y-1.

Calcular el tiempo de ejecución **de cada proceso** (desde que se inicia hasta que termina de imprimir "Hola mundo…") cuando hay 1, 20 y 50 procesos.

Explicar los resultados según varía el número de procesos (se puede realizar una gráfica para ayudar con la explicación).

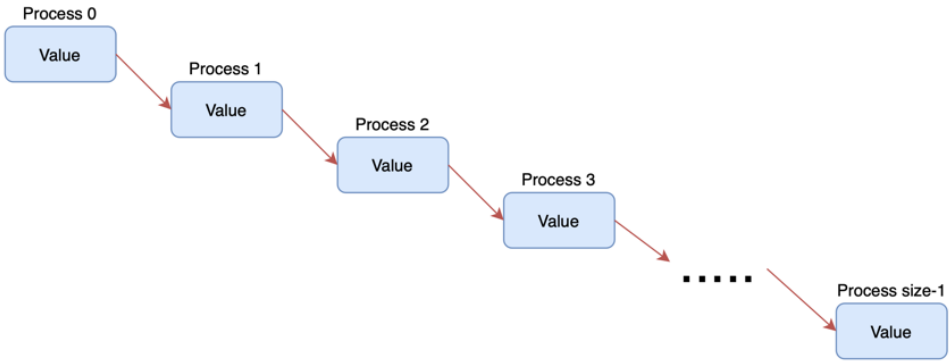
Contestar a las siguientes preguntas (ilustrándolas con los resultados obtenidos en cada ejecución):

1. ¿Los procesos imprimen el hola mundo y el tiempo en orden? ¿A qué se debe esto?

2. ¿Qué pasa con el tiempo de ejecución si eliminamos las barreras (MPI_Barrier())? ¿Y con el orden de ejecución? ¿Porqué?

## Ejercicio 2 (4 puntos)

Implementar un programa usando MPI, donde el proceso 0 toma un dato del usuario y lo envía al siguiente nodo, que lo envía al siguiente y así hasta llegar al último nodo. Esto es, el proceso i recibe de i-1 y transmite el dato a i+1, hasta que el dato alcanza el último nodo:




Asumir que el dato que se transmite es un entero y que el proceso cero lee el dato del usuario.

## Ejercicio 3 (4 puntos)

Modificar la implementación del ejercicio 2 para que el dato introducido por el usuario dé tantas vueltas como este indique en anillo.

1. ¿Qué desventaja se aprecia en este tipo de comunicaciones punto a punto a medida que aumentan el número de procesos requeridos? Razonar la respuesta.

2. ¿Cómo podría mejorar el sistema y su implementación? Razonar la respuesta.