

Grado en Ingeniería Informática

Arquitectura de Computadores

Práctica 1

Contenido

Práctica 1: Fundamentos básicos de paralelización: Paso de mensajes.....	3
Introducción.....	3
Instalación de MPICH.....	3
Entregables	3
Ejercicio 1	3
Ejercicio 2	4
Ejercicio 3	4
Entrega.....	4

Práctica 1: 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 `recv`.

Instalación de MPICH

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>

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 -version
```

Entregables

- Memoria
- Código fuente de los siguientes ejercicios:

Ejercicio 1

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

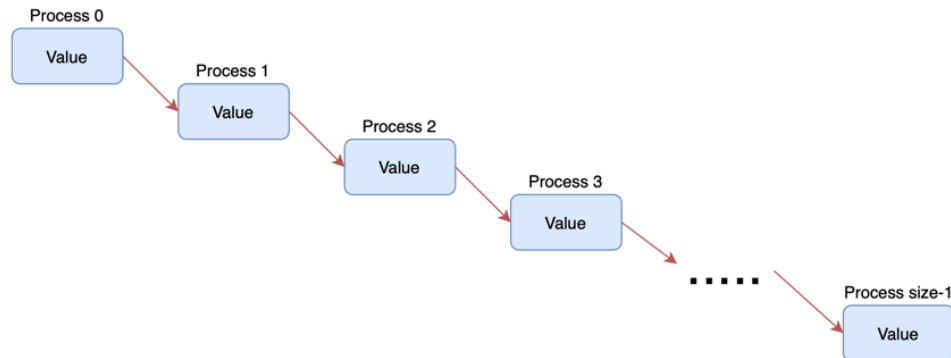
“Hola mundo, soy el proceso X de un total de Y.”

cuando el número total de tareas es $Y=50$ y X un rango de 0 a 49.

Calcular el tiempo de ejecución de cada proceso y realizar una gráfica explicando los resultados a medida que aumenta el número de procesos.

Ejercicio 2

Implementar un programa usando MPI, donde el proceso 0 toma un dato del usuario y lo envía al resto de nodos en anillo. 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

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

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

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

Entrega

Es imprescindible subir los entregables a la actividad del **campus** y enviármelos por **correo** (lrodriguezso@nebrija.es) antes del **11 de noviembre a las 7:59 am**.