

Describe de manera teórica los siguientes conceptos: SISD, SIMD, MISD y MIMD. Indique además que lenguajes aplican a estos.

SISD (Una Instrucción, Un Dato)

En esta arquitectura, un solo procesador ejecuta una instrucción a la vez sobre un único conjunto de datos. Este enfoque secuencial es fácil de entender y programar, pero puede ser ineficiente para tareas que podrían beneficiarse de la paralelización.

Características:

- **Un solo procesador:** Solo un núcleo realiza operaciones.
- **Un solo conjunto de instrucciones:** Las instrucciones se ejecutan en el orden en que se presentan.
- **Un solo conjunto de datos:** Los datos se procesan uno por uno.

Ejemplo práctico:

- **Procesamiento de textos,** donde se realiza una edición a la vez.

Lenguajes comunes: C, C++, Java, Python.

SIMD (Una Instrucción, Múltiples Datos)

En esta arquitectura, un solo procesador aplica la misma instrucción a varios datos simultáneamente. Esto es particularmente útil en tareas donde la misma operación se necesita aplicar a un conjunto grande de datos.

Características:

- **Una instrucción, múltiples datos:** Se procesan múltiples elementos en paralelo con la misma operación.
- **Ideal para operaciones repetitivas:** Como en gráficos o cálculos científicos.

Ejemplo práctico:

- **Procesamiento de imágenes,** donde se aplica un filtro a cada píxel de una imagen.

Lenguajes comunes: CUDA, OpenCL, extensiones de C/C++.

MISD (Múltiples Instrucciones, Un Dato)

Este modelo es más teórico y menos común en la práctica. Implica varios procesadores que ejecutan diferentes instrucciones sobre el mismo conjunto

de datos. Es útil en sistemas donde se requiere alta confiabilidad, como en aplicaciones críticas.

Características:

- **Múltiples instrucciones:** Diferentes procesadores realizan diversas operaciones sobre un único conjunto de datos.

Ejemplo práctico:

- **Sistemas de control de aeronaves,** donde se realizan diversas verificaciones y ajustes simultáneamente sobre datos de vuelo.

Lenguajes comunes: No hay lenguajes específicos, pero se podrían utilizar Ada o lenguajes de sistemas embebidos.

MIMD (Múltiples Instrucciones, Múltiples Datos)

Esta arquitectura permite que varios procesadores ejecuten diferentes instrucciones en diferentes conjuntos de datos al mismo tiempo, proporcionando una gran flexibilidad y eficiencia. Es la arquitectura más común en computadoras modernas.

Características:

- **Múltiples instrucciones y múltiples datos:** Cada procesador puede realizar tareas distintas en paralelo.
- **Escalable y eficiente:** Puede aumentar el rendimiento al añadir más procesadores.

Ejemplo práctico:

- **Servidores web,** donde diferentes procesos manejan solicitudes simultáneamente, cada uno ejecutando instrucciones diferentes.

Lenguajes comunes: MPI, OpenMP, C++ con hilos, Go, Python (con bibliotecas de multiprocessing).