

INF – 317

Sistemas en Tiempo Real y Distribuidos

Nombres y Apellidos: Boris Orlando Roque Huanca

C.I.: 9868420 LP

Docente: Ph. D. Moisés Martin Silva Choque

Fecha de Entrega: 30/10/2023

1er Examen Parcial - Ejercicio 1

Explique cuál es la relación de la taxonomía de Flynn y cada una de las librerías utilizadas hasta el momento.

R.- La Taxonomía de Flynn es una clasificación de arquitecturas de computadoras propuesta por Michael J. Flynn en 1972.

Se la puede describir mediante el siguiente recuadro:

	Una instrucción	Múltiples instrucciones
Un dato	SISD	MISD
Múltiples datos	SIMD	MIMD

Se desglosan de la siguiente manera:

- Una instrucción, un dato (SISD: Single Instruction stream, Single Data stream)

Computador secuencial que no explota el paralelismo en las instrucciones ni en flujos de datos. Es la Arquitectura Von-Neumann. Un único procesador ejecuta un solo flujo de instrucciones para operar datos en una única memoria. Se ejecuta una única instrucción y un dato en cada ciclo de reloj. Puede utilizar técnicas de segmentación o de pipelining. Ejemplos de arquitecturas SISD son las máquinas con uni-procesador o monoprocesador tradicionales como el PC o los antiguos mainframe.

- Una instrucción, múltiples datos (SIMD: Single Instruction stream, Multiple Data streams)

Todas las unidades ejecutan la misma instrucción sincronizadamente, pero con datos distintos. Es un computador que explota varios flujos de datos dentro de un único flujo de instrucciones para realizar operaciones que pueden ser paralelizadas de manera natural. Puede ser de Arquitectura vectorial o Arquitectura matricial.

- Múltiples instrucciones, un dato (MISD: Multiple Instruction streams, Single Data stream)

Poco común debido al hecho de que la efectividad de los múltiples flujos de instrucciones suele precisar de múltiples flujos de datos. Sin embargo, este tipo se usa en situaciones de

paralelismo redundante, como por ejemplo en navegación aérea, donde se necesitan varios sistemas de respaldo en caso de que uno falle. También se han propuesto algunas arquitecturas teóricas que hacen uso de MISD, pero ninguna llegó a producirse en masa. Algunos autores consideran que las arquitecturas vectoriales supersegmentadas o vectorial escalar forman parte de este modelo ya que en un momento dado se pueden estar manipulando un dato (el vector) por varias instrucciones, no obstante, no existe consenso al respecto.

- Múltiples instrucciones, múltiples datos (MIMD: Multiple Instruction streams, Multiple Data streams)

Varios procesadores autónomos que ejecutan simultáneamente instrucciones diferentes sobre datos diferentes. Los sistemas distribuidos suelen clasificarse como arquitecturas MIMD; bien sea explotando un único espacio compartido de memoria, o uno distribuido.

Relación con las Librerías Usadas Hasta el Momento

Durante el avance en clases exploramos 4 librerías fundamentales: Multiprocessing de Python, Threading de C#, OpenMP de C y MPI de C. En base a los conceptos previamente descritos, se realiza la siguiente observación:

- Multiprocessing de Python:

Se lo puede acomodar en la categoría MIMD (Multiple Instruction streams, Multiple Data streams), debido a que esta permite la creación de procesos independientes que pueden ejecutar diferentes instrucciones en diferentes conjuntos de datos al mismo tiempo. Cada proceso tiene su propio flujo de instrucciones y datos.

- Threading de C#:

Se lo puede acomodar en la categoría SISD (Single Instruction stream, Single Data stream) ya que aunque los hilos (threads) en C# pueden ejecutarse de manera concurrente, generalmente comparten el mismo espacio de memoria, por lo tanto, operan sobre un solo flujo de datos. Aunque múltiples hilos pueden ejecutar diferentes instrucciones, siguen operando sobre el mismo conjunto de datos.

- OpenMP de C:

Se lo podría acomodar principalmente en la categoría SIMD (Single Instruction stream, Multiple Data streams), pero también puede ser MIMD dependiendo del uso o implementación que se le dé. OpenMP se utiliza para paralelizar bucles y operaciones en C. En su forma más común, permite la ejecución simultánea de la misma instrucción en diferentes conjuntos de datos (Lo que está relacionado con SIMD). Sin embargo, OpenMP también puede utilizarse para la ejecución de múltiples instrucciones en diferentes conjuntos de datos en sistemas multiprocesador, lo que se alinea con la categoría MIMD.

- MPI de C:

Se lo puede ubicar en la categoría MIMD (Multiple Instruction streams, Multiple Data streams).

MPI (Message Passing Interface) es una interfaz estándar utilizada para la comunicación entre procesos en sistemas distribuidos y paralelos. Permite la ejecución de diferentes instrucciones en diferentes conjuntos de datos en diferentes nodos de un clúster de computadoras, cayendo tranquilamente dentro de la categoría MIMD.