

UNIVERSIDAD MAYOR DE SAN ANDRÉS

FACULTAD CIENCIAS PURAS Y NATURALES

CARRERA INFORMÁTICA



EXAMEN INF 317 - PRIMER PARCIAL

ESTUDIANTE: SURCO NINA WILLIAMS RODRIGO

CI: 7022363 LP.

MATERIA: INF-317 SISTEMAS EN TIEMPO REAL Y DISTRIBUIDO

PARALELO: A

DOCENTE: PhD. MOISES SILVA CHOQUE

GESTIÓN: II/2023

LA PAZ – BOLIVIA

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

La taxonomía de Flynn es un sistema de clasificación utilizado en la arquitectura de computadoras para categorizar las diferentes arquitecturas de computadoras según la cantidad y la forma en que procesan las instrucciones y los datos.

- **SISD (Single Instruction, Single Data): "Instrucción única, dato único".** En este tipo de arquitectura, una única instrucción se ejecuta en un único dato a la vez. Es el tipo de arquitectura más común y se asemeja a una computadora convencional de un solo procesador.
- **SIMD (Single Instruction, Multiple Data): "Instrucción única, datos múltiples".** En este caso, una única instrucción se ejecuta en múltiples conjuntos de datos de manera simultánea. Es típico de arquitecturas de procesadores vectoriales y matriciales, donde se realizan operaciones paralelas en conjuntos de datos.
- **MISD (Multiple Instruction, Single Data): "Instrucciones múltiples, dato único".** Este tipo de arquitectura implica que múltiples instrucciones operan sobre un único conjunto de datos. Aunque es una clasificación teórica, rara vez se encuentra en sistemas reales debido a su complejidad y falta de aplicaciones prácticas.
- **MIMD (Multiple Instruction, Multiple Data): "Instrucciones múltiples, datos múltiples".** En este caso, múltiples instrucciones se ejecutan en múltiples conjuntos de datos de forma simultánea. Es el tipo de arquitectura utilizada en sistemas multiprocesador, clústeres y supercomputadoras, donde múltiples núcleos o procesadores ejecutan programas independientes.

De las librerías usadas tenemos:

OpenMP: utiliza una arquitectura MIMD (Multiple Instruction Multiple Data) para el procesamiento paralelo en sistemas de memoria compartida (SMP). OpenMP proporciona un modelo de programación de hilos que permite a los desarrolladores escribir código paralelo de forma más sencilla y eficiente.

MPI (Message Passing Interface): utiliza una arquitectura MIMD (Multiple Instruction Multiple Data) para el procesamiento paralelo en sistemas de memoria distribuida. MPI permite la comunicación entre procesos y la coordinación de la ejecución en sistemas distribuidos y se utiliza comúnmente en aplicaciones de simulación y modelado.

Multiprocessing: utiliza una arquitectura MIMD (Multiple Instruction Multiple Data) para el procesamiento paralelo en sistemas de memoria compartida (SMP) y distribuida. La biblioteca multiprocessing de Python permite la creación de procesos que pueden ejecutarse en paralelo en diferentes núcleos o incluso en diferentes máquinas, lo que permite una mayor velocidad de procesamiento y escalabilidad.

Threads(NET): utiliza una arquitectura SISD (Single Instruction Single Data) o MIMD (Multiple Instruction Multiple Data) dependiendo de la implementación. Los hilos son subprocesos de ejecución que pueden ser programados para ejecutarse en paralelo en diferentes núcleos de procesamiento. Los hilos también pueden ser implementados en sistemas de memoria compartida (SMP) o distribuida. Los hilos se utilizan comúnmente en aplicaciones que requieren una alta concurrencia, como servidores web o aplicaciones de bases de datos.

2. Mediante OpenMP realice la división de una frase en dos, por ejemplo “tres tristes tigres tragaban trigo por culpa del bolívar”; la división sería “tres tigres trigo culpa bolívar” y “tristes tragaban por del”.

https://github.com/WillCona/PrimerParcial_INF317/tree/main/Ejercicio2

3. Realice lo mismo con MPI utilizando al menos 4 hilos.

https://github.com/WillCona/PrimerParcial_INF317/tree/main/Ejercicio3

4. Con multiprocessing genera la siguiente serie 2, 2, 5, 4, 10, 6, 17,... (10000 posiciones sin restricción de procesador).

https://github.com/WillCona/PrimerParcial_INF317/tree/main/Ejercicio4

5. La numeral 4 con NET manejando hilos.

https://github.com/WillCona/PrimerParcial_INF317/tree/main/Ejercicio5

6. La numeral 4 con MPI.

https://github.com/WillCona/PrimerParcial_INF317/tree/main/Ejercicio6

7. Realice el cálculo de PI con NET.

https://github.com/WillCona/PrimerParcial_INF317/tree/main/Ejercicio7