**2 OpenMP y MPI de cual elemento taxonómico anterior es.**

Existen fundamentalmente dos modelos de programación ampliamente extendidos: el modelo de paso de mensajes a través de librerías basadas en el estándar MPI, y el de memoria compartida con la utilización de OpenMP.

**OpenMP (Open Multi-Processing).- Es una interfaz de programación de aplicaciones (API) para la programación multiproceso de memoria compartida en múltiples plataformas.**

**OpenMP trabaja con el taxonomico.**

* ***MISD:*** Los sistemas con flujo MISD tienen un número de unidades de procesamiento que realizan diferentes operaciones mediante la ejecución de diferentes instrucciones en el mismo conjunto de datos.

**Caracteristicas de OpenMP**

* **Paralelismo de memoria compartida basado en threads**
* **Modelo fork-join**
* **Basado en directivas al compilador**
* **Soporta paralelismo dentro del paralelismo**
* **6Threads dinamicos**

***Los tipos de variables en OpenMP son tres:***

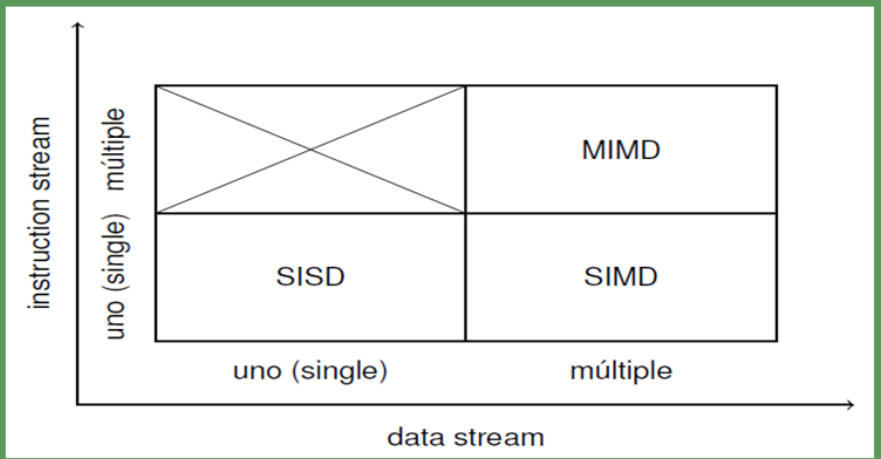
* **Shared:** Todos los hilos acceden a la misma variable.
* **Private:** Cada hilo tiene su propia variable que, particularmente, se llaman igual en todos los hilos.
* **Reduction:** La variable, durante toda la ejecución de la parte paralela, se comporta como privada (cada hilo tiene su copia), pero cuando terminan, todos devuelven el valor final sobre la misma variable. Ejemplo de uso: buscar el máximo (todos los hilos buscan el máximo y después el hilo principal recolecta todos estos máximos y elige el máximo total). A partir de muchos datos se quiere obtener un resultado.  
  Las operaciones que se pueden realizar son: +, \*, -, ^, |, ||, & o &&.

**MPI (Message Passing Interface).-** Es un estándar que define la sintaxis y la semántica de las funciones contenidas en una biblioteca de paso de mensajes diseñada para ser usada en programas que exploten la existencia de múltiples procesadores.

MPI fue creado en 1993 como un estándar abierto por fabricantes y usuarios de sistemas paralelos. MPI ha sido el primer esfuerzo realizado para producir un interface estándar para el paso de mensajes. Sesenta personas expertas en el uso de sistemas basados en el paso de mensajes (incluyendo PVM, PARMACS, y EPCC's CHIMP) y representantes de diferentes organizaciones, usuarios y vendedores de sistemas paralelos tanto de U.S. como de Europa, formaron el FORUM MPI que fue encargado de definir el estándar. MPI-2 apunta a ampliarse a otras áreas de programación distribuida

***MPI trabaja con estos taxonomico.***

* SISD (Single Instruction, Single Data) : Computadora de [von Neumann](https://www.monografias.com/trabajos28/arquitectura-von-neumann/arquitectura-von-neumann.shtml).
* SIMD (Single Instruction, Multiple Data) : En estas [computadoras](https://www.monografias.com/trabajos15/computadoras/computadoras.shtml) se aplica una instrucción a un [grupo](https://www.monografias.com/trabajos14/dinamica-grupos/dinamica-grupos.shtml) de datos simultáneamente.
* MIMD (Multiple Instruction, Multiple Data) : Los [procesadores](https://www.monografias.com/trabajos5/sisope/sisope.shtml) obedecen automáticamente sus propias secuencias de instrucciones y las aplican a sus propios datos.



*Imagen 1. Se muestra las taxonomico con que trabaja MPI.*

***Características de MPI.***

* Interfaz genérica que permite una implementación optimizada en cualquier sistema
* paralelo.
* Es una biblioteca que incluye interfaces para FORTRAN, C y C++.
* Define varias formas de comunicación lo que permite programar de manera natural
* cualquier algoritmo en paralelo.
* Está pensado para crear bibliotecas paralelas.

**Los tipos de datos son:**

|  |  |
| --- | --- |
| **MPI datatype** | **C datatype** |
| MPI\_CHAR | signed char |
| MPI\_SHORT | signed short int |
| MPI\_INT | signed int |
| MPI\_LONG | signed long int |
| MPI\_UNSIGNED\_CHAR | unsigned char |
| MPI\_UNSIGNED\_SHORT | unsigned short int |
| MPI\_UNSIGNED | unsigned int |
| MPI\_UNSIGNED\_LONG | unsigned long int |
| MPI\_FLOAT | float |
| MPI\_DOUBLE | double |
| MPI\_LONG\_DOUBLE | long double |
| MPI\_BYTE |  |
| MPI\_PACKED |  |