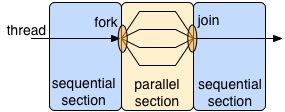
**Ejercicio 2**

OpenMP y MPI de cual elemento taxonómico anterior es.

**OpenMP** es un framework de programación paralela para C/C++ y Fortran. Ha ganado bastante popularidad en los últimos años, principalmente por su simplicidad y buen performance.

Hay muchos tipos de paralelismo y patrones para aprovechar ese paralelismo. OpenMP utiliza un modelo de fork y join anidado. Por defecto, un programa de OpenMP es un programa secuencial normal, exceptuando las regiones que el programador explícitamente declara para que se ejecuten en paralelo. En la región paralela, el framework crea (hace un fork) un número de threads. Típicamente estos threads ejecutan las mismas instrucciones, solo que en diferentes porciones de los datos. Al final de la región paralela, el framework espera por todos los threads a que terminen (hace join) antes de dejar la región y continuar secuencialmente.



OpenMP utiliza memoria compartida, esto significa que todos los threads pueden acceder al mismo espacio de direcciones. La alternativa a esto sería memoria distribuida, que es lo principal en clusters donde los datos deberían de ser movidos explícitamente entre espacios de direcciones. Muchos programadores encuentran la memoria compartida más fácil de programar, dado que no tienen que preocuparse de mover los datos, pero es usualmente más difícil de implementar en hardware de una manera escalable.

Rta: **openMp es elemento de MISD**

**MPI** (Message Passing Interface) como es un interfaz estandarizada para la realización de aplicaciones paralelas basadas en pasaje de mensajes. El modelo de programación que subyace tras MPI es MIMD (Multiple Instruction streams, Multiple Data streams) aunque se dan especiales facilidades para la utilización del modelo SPMD (Single Program Multiple Data), un caso particular de MIMD en el que todos los procesos ejecutan el mismo programa, aunque no necesariamente la misma instrucción al mismo tiempo. El pasaje de mensajes es un paradigma ampliamente usado en determinadas clases de máquinas paralelas, especialmente aquellas con memoria distribuida. Aun habiendo bastantes variaciones, los conceptos básicos de las comunicaciones entre procesos por medio de mensajes están bien estudiados, en los últimos 15 años se han hecho progresos sustanciales en lo que hace a las aplicaciones de este paradigma y cada distribuidor ha implementado su propia variante, así es como recientemente muchos sistemas han demostrado ser eficientes y portables. De esta manera y con el tiempo es como se ha definido la sintaxis y la semántica de un núcleo de librerías de rutinas que hace del pasaje de mensajes una herramienta absolutamente útil para un amplio rango de usuarios e implementable sobre variedad de computadoras.

Rta. MPI es MIMD (Multiple Instruction, Multiple Data) aunque se dan especiales facilidades para la utilización del modelo SPMD (Single Program Multiple Data)