

Sistemas Operativos

Martín Silva

Alejandro Mansilla

Licenciatura en Ciencias de la Computación

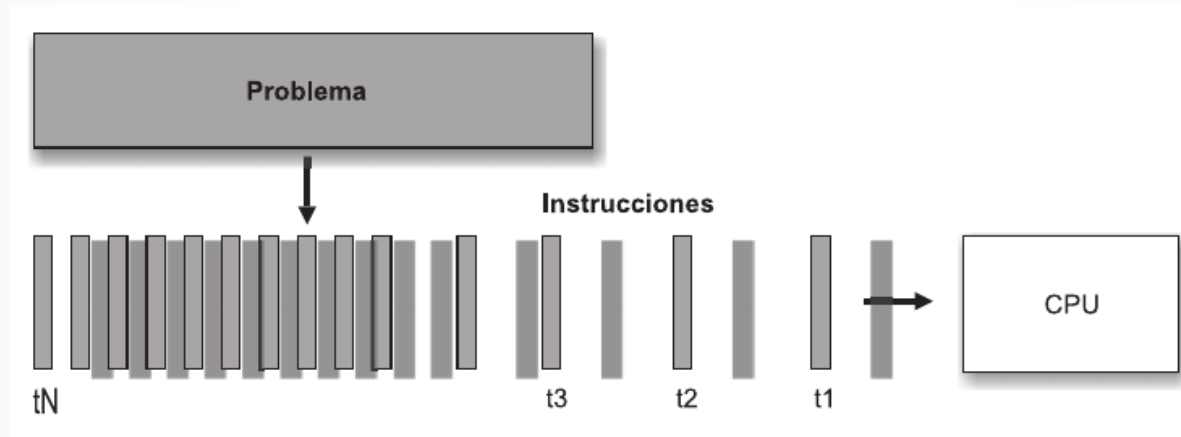
2019



Sistemas Operativos

Multiprocesamiento

- La ley de Moore (1965) expresa que aproximadamente cada dos años se duplica el número de transistores en un circuito integrado. Sin embargo, cuando se formuló no existían los microprocesadores, inventados en 1971.

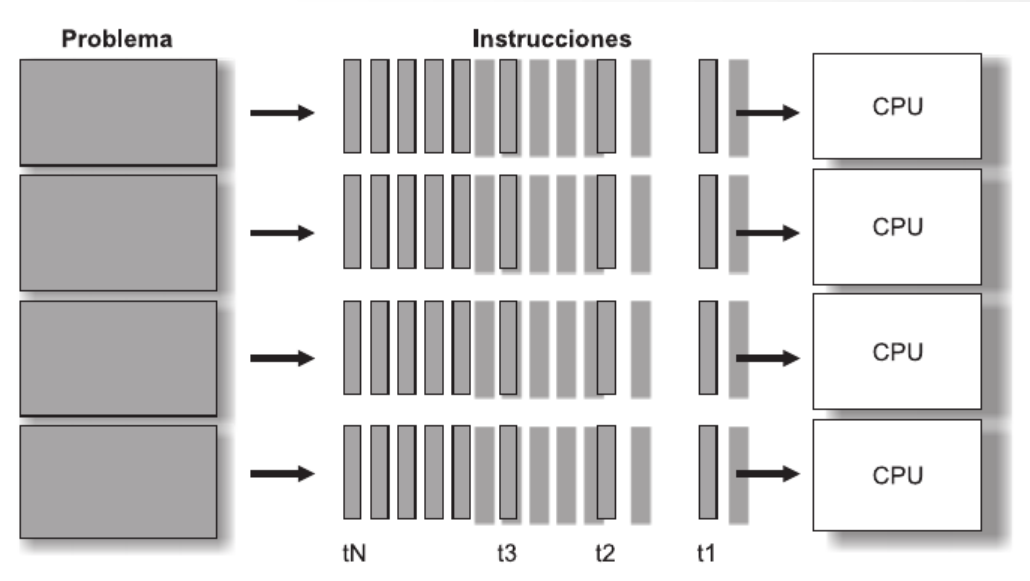


- Los programadores conciben programas partiendo un problema en secuencias de instrucciones, que el procesador ejecuta “secuencialmente”.
- Esto no ha sido nunca completamente cierto. A nivel de las microoperaciones se generan muchas señales de control al mismo tiempo.

Sistemas Operativos

Multiprocesamiento

- Para incrementar el rendimiento de un microprocesador se les aumenta la velocidad mediante un reloj más rápido.
- Sin embargo, no podemos seguir en esa línea indefinidamente porque hay razones de la Física que nos imponen restricciones.
- La otra solución es la de utilizar varios procesadores.
- Un problema grande se puede dividir en partes más pequeñas que pueden procesarse en forma paralela.



En este modelo, el problema se divide en varias partes y el programa se desarrolla de manera explícitamente paralela gracias a lenguajes y bibliotecas especializadas. Por ejemplo el producto de dos matrices.

Sistemas Operativos

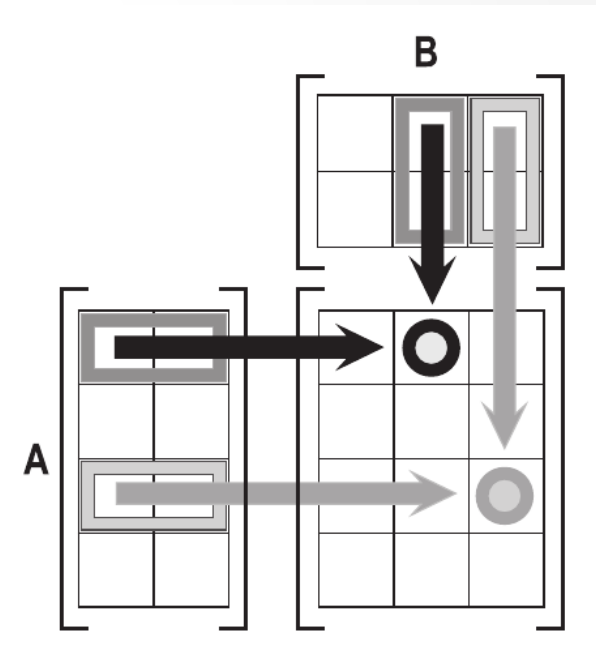
Multiprocesamiento

Dadas dos matrices A y B, tales que el número de columnas de la matriz A sea igual al número de filas de la matriz B, el producto A por B será una nueva matriz C en la que cada elemento $c_{i,j}$ estará definido por

$$c_{i,j} = \sum_{r=1}^n a_{rj} \cdot b_{ri}$$

Cada procesador puede ocuparse de obtener un elemento $c_{i,j}$ si le brindamos una fila de A y una columna de B.

- Si se parte la aplicación en múltiples procesos éstos se comunicarán utilizando formas de comunicación inter procesos (IPC) tales como tuberías, semáforos o memoria compartida.
- Los procesos deben ser capaces de bloquearse a la espera de eventos tales como mensajes de otros procesos, y deben coordinar el acceso a objetos compartidos con algo como cerrojos.
- Se puede desarrollar un programa en C (o Fortran) utilizando la biblioteca MPI (*Message Passing Interface*).



Sistemas Operativos

Multiprocesamiento - Speedup

- No todos los problemas pueden paralelizarse de esta manera.
- Las aplicaciones se diseñan e implementan mediante hilos. El núcleo ejecuta en varios procesadores (o núcleos) toma fragmentos de hilos de distintas aplicaciones y las ejecuta de manera asíncrona.
- El rendimiento tampoco escala linealmente con el agregado de procesadores.

$$S_P = \frac{T_1}{T_P}$$

- Siendo P el número de procesadores, T_1 el tiempo de ejecución con el algoritmo secuencial con un procesador y T_p el tiempo de ejecución con el algoritmo paralelo con P procesadores, la mejora o *speedup* S de P procesadores será el cociente entre ambos.
- El *speedup* lineal o ideal se da con $S_p = P$ (si duplicamos el número de procesadores duplicamos la velocidad)
- Pero la proporción de aumento de la velocidad con P procesadores no es P, sino más bien menor que P.
- Cuando varios procesadores (o núcleos) cooperan para llevar a cabo una tarea, se incurre en cierto “gasto extra” que, unido a la contención por los recursos compartidos, reduce la ganancia que cabría esperar.

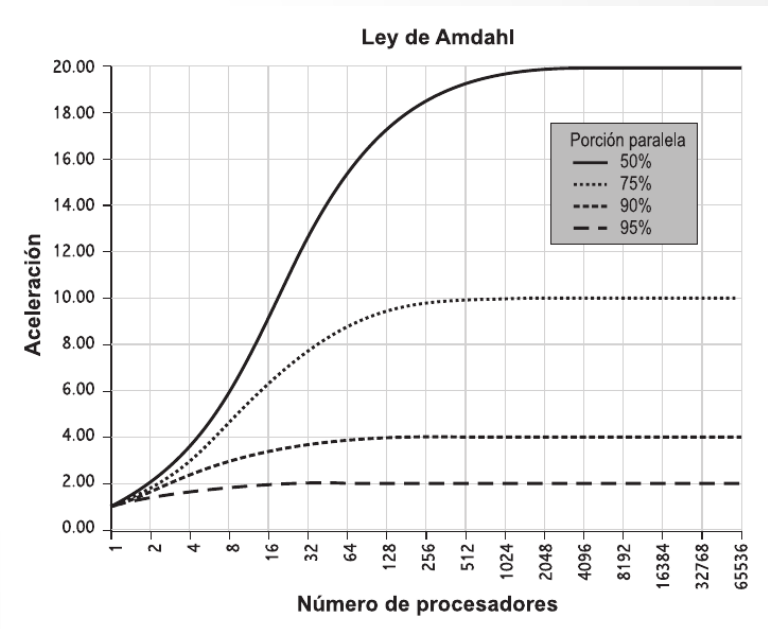
Sistemas Operativos

Multiprocesamiento

- Amdahl (1967) demostró que existe un máximo en la mejora esperada en el sistema cuando hay una mejora en una porción de éste.
- En el caso especial de la paralelización, la Ley de Amdahl declara que si F es la fracción de un cálculo secuencial y $1-F$ la fracción que puede ser paralelizada, entonces la máxima aceleración o *speedup* que se puede lograr utilizando P procesadores es:

$$\frac{1}{(F + (1 - F)/P)}$$

- En el límite, con P tendiendo a infinito, la máxima aceleración tiende a $1/F$.



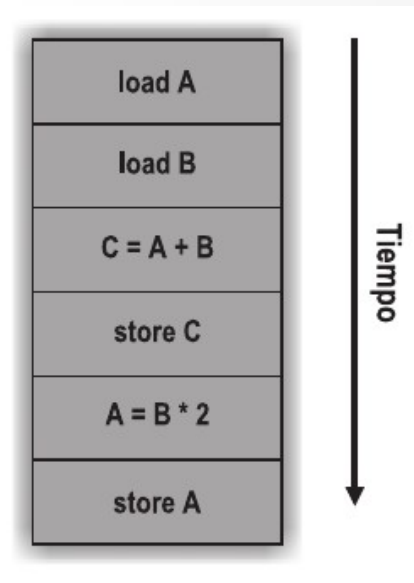
Sistemas Operativos

Multiprocesamiento – Taxonomía de Flynn

- Flynn (1972) eligió dos características:
 - El número de flujos de instrucciones
 - El número de flujos de datos
- Una computadora con un flujo de instrucción y un flujo de datos es SISD.

| | Single instruction | Multiple instructions |
|---------------|--------------------|-----------------------|
| Single data | SISD | MISD |
| Multiple data | SIMD | MIMD |

- Computadora netamente serial, explícitamente no paralela.
- La CPU actúa sobre un único flujo de instrucciones durante un ciclo de reloj y tiene un único flujo de datos utilizado como entrada durante ese ciclo de reloj. Su ejecución es determinística.
- Es el concepto de arquitectura serie de Von Neumann.
- Todas las máquinas SISD poseen un registro simple -el contador de programa- que asegura la ejecución serie.
- Ningún computador netamente SISD se fabrica en la actualidad.

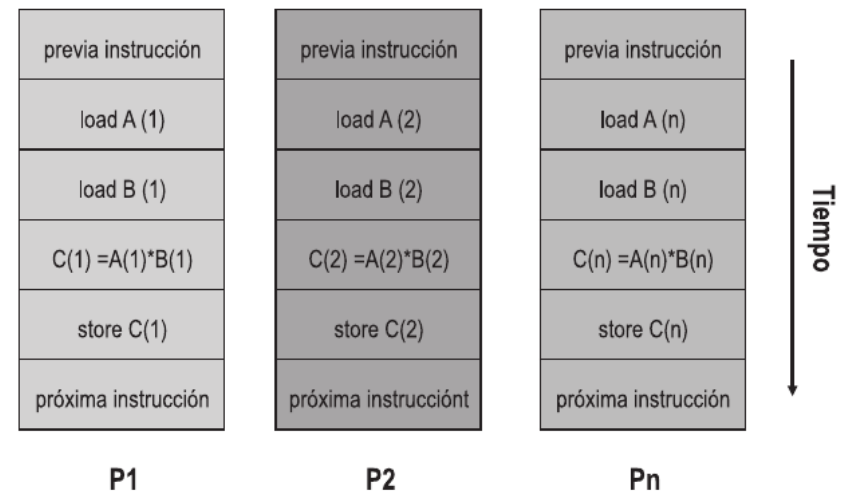


Sistemas Operativos

Multiprocesamiento – Taxonomía de Flynn

- Flynn (1972) eligió dos características:
 - El número de flujos de instrucciones
 - El número de flujos de datos
- Una computadora con un flujo de instrucción y varios flujos de datos es SIMD
- Útiles para los cálculos que repiten los mismos cálculos en varios conjuntos de datos.
- Por ejemplo, sumando todos los elementos de 64 vectores independientes. Paralelismo de datos.
- Un procesador vectorial y un GPU (*Graphical Processing Unit*) caen en esta categoría.
- La paga para cada trabajador en una empresa.

| | Single instruction | Multiple instructions |
|---------------|--------------------|-----------------------|
| Single data | SISD | MISD |
| Multiple data | SIMD | MIMD |

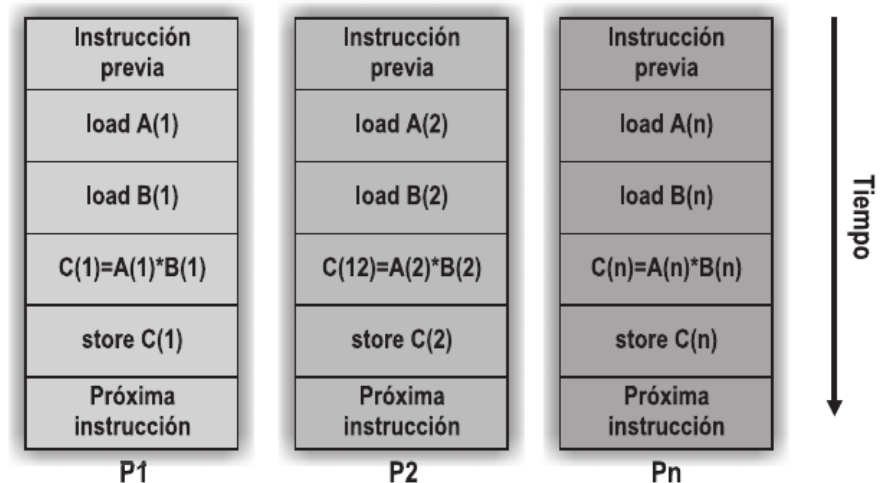


Sistemas Operativos

Multiprocesamiento – Taxonomía de Flynn

- Flynn (1972) eligió dos características:
 - El número de flujos de instrucciones
 - El número de flujos de datos
- Una computadora con un flujo de varias instrucciones y sólo un flujo de datos es MISD.
- Ninguna de las computadoras comúnmente conocidas se ajusta a este modelo ya que implica que varios procesadores operen sobre los mismos datos y acuerden en el resultado.
- Sería aplicable en caso de ser necesario un sistema tolerante a fallos.
- Por ejemplo la computadora de control de vuelo del transbordador espacial.

| | Single instruction | Multiple instructions |
|---------------|--------------------|-----------------------|
| Single data | SISD | MISD |
| Multiple data | SIMD | MIMD |

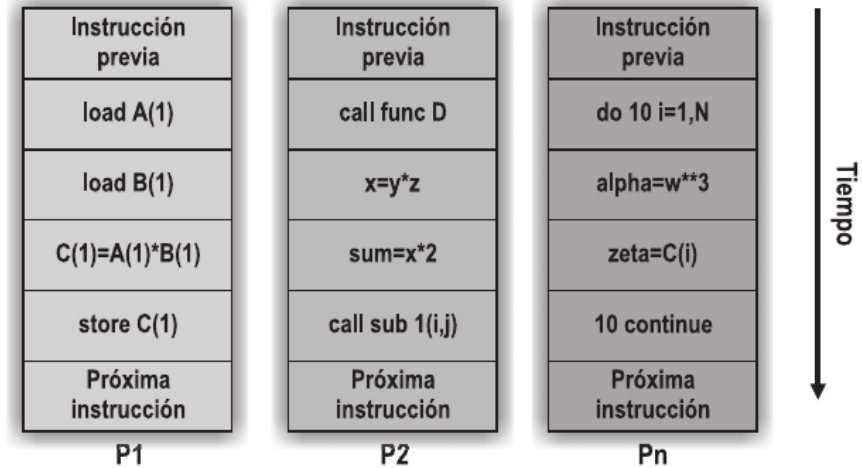


Sistemas Operativos

Multiprocesamiento – Taxonomía de Flynn

- Flynn (1972) eligió dos características:
 - El número de flujos de instrucciones
 - El número de flujos de datos
- MIMD, con múltiples flujos de instrucciones y flujo de datos múltiple.
- Se refiere a un grupo de procesadores independientes, cada uno con su propio contador de programa y datos.
- Si tenemos procesadores separados operando sobre datos independientes, tendremos un **paralelismo de control**.
- Las MIMD son las más complejas, pero son las que ofrecen una mayor eficiencia en la ejecución concurrente o paralela.
- Todos los sistemas distribuidos son MIMD.
- La primera arquitectura SMP fue desarrollada por Burroughs en 1962. El multiprocesador MIMD simétrico D825.

| | Single instruction | Multiple instructions |
|---------------|--------------------|-----------------------|
| Single data | SISD | MISD |
| Multiple data | SIMD | MIMD |

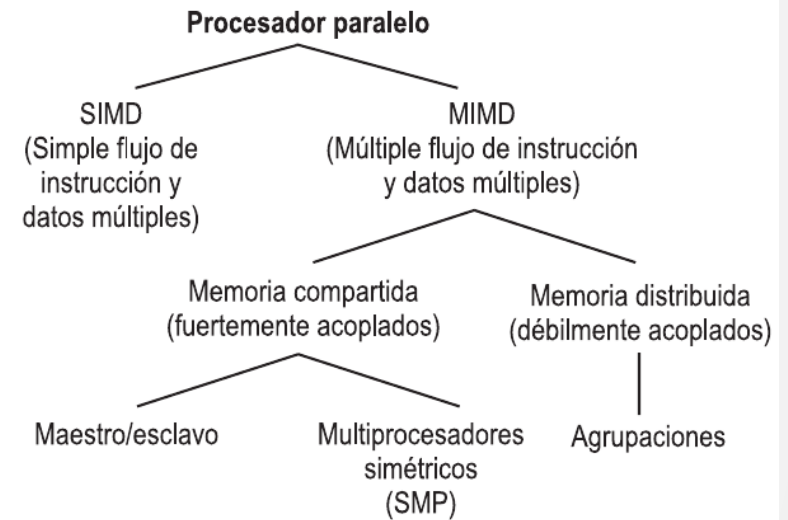


Sistemas Operativos

Multiprocesamiento – Múltiples procesadores

La clasificación MIMD puede, a su vez, subdividirse de acuerdo con el medio por el cual se comunican:

- Si cada procesador tiene una memoria dedicada, cada elemento de procesamiento es en sí una computadora.
- Si la comunicación se realiza mediante una red, los elementos están débilmente acoplados:
 - Multicomputadora, agrupación o *cluster*, Beowulf.
- Si los computadores comparten una memoria en común y se comunican entre sí a través de esta memoria, tendremos un sistema multiprocesador de memoria compartida. Están fuertemente acoplados.
- Estas dos arquitecturas no son excluyentes entre sí.

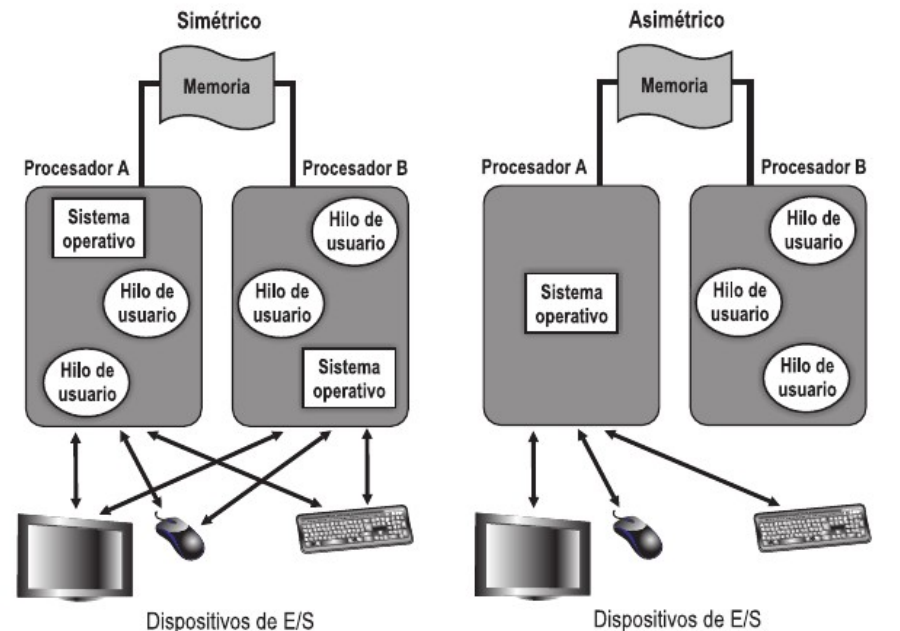


Sistemas Operativos

Multiprocesamiento – AMP y SMP

De acuerdo a cómo son asignados los procesos a los procesadores: AMP o SMP

- AMP: *Asymmetric Multiprocessing* o *master/slave*.
- SMP: *Symmetric Multiprocessing*.
- En AMP el núcleo ejecuta en un procesador y en el otro programas de usuario.
- En SMP el núcleo puede ejecutar en cualquier procesador.
 - Usualmente cada procesador hace su propia planificación.
 - El término “simétrico” en un SMP indica que cualquier unidad de software debería ser capaz de ejecutar en cualquier procesador. No hay un procesador “maestro”.

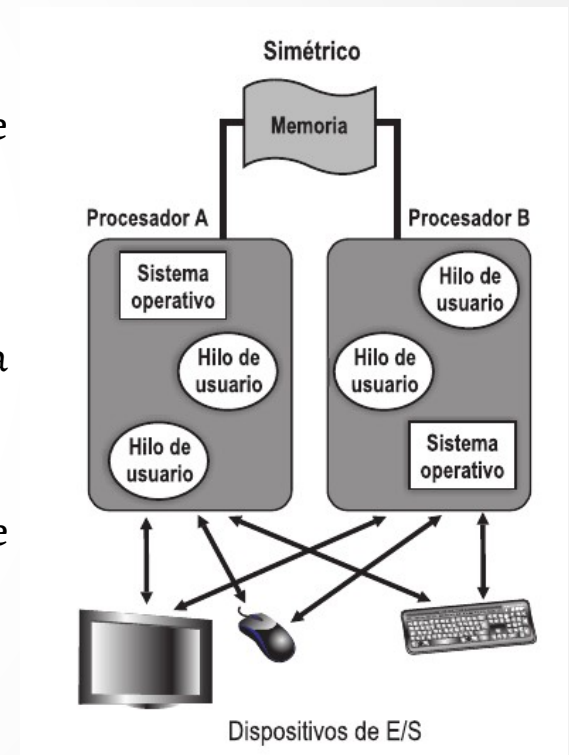


Sistemas Operativos

Multiprocesamiento – SMP

Ventajas

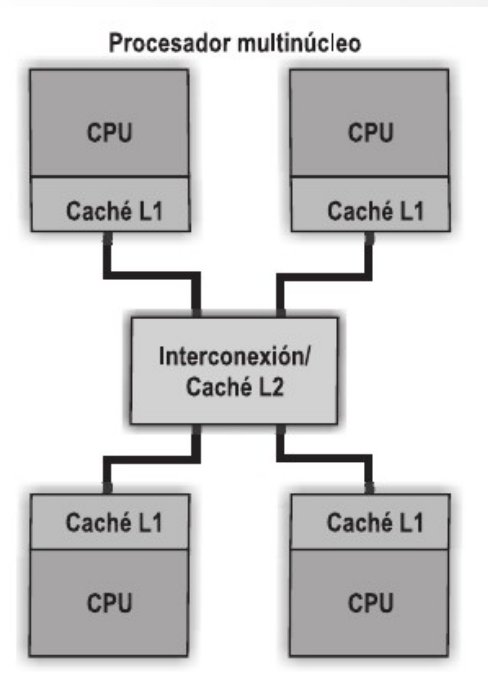
- Son una manera rentable de aumentar el rendimiento.
- Ofrecen una imagen única de sistema ya que el SO está compartido entre todos los procesadores.
- Aplican varios procesadores a un único problema.
- El SO hace el balanceo de cargas.
- El modelo de programación del uniprocador se puede usar en un sistema multiprocador de memoria compartida.
- Son escalables para datos compartidos.
- Todos los datos son direccionables por todos los procesadores y se mantienen coherentes.
- No hay necesidad de usar bibliotecas de pase de mensajes.
- Cada vez hay más y más aplicaciones y herramientas.



Sistemas Operativos

Multiprocesamiento – SMP - Organizaciones

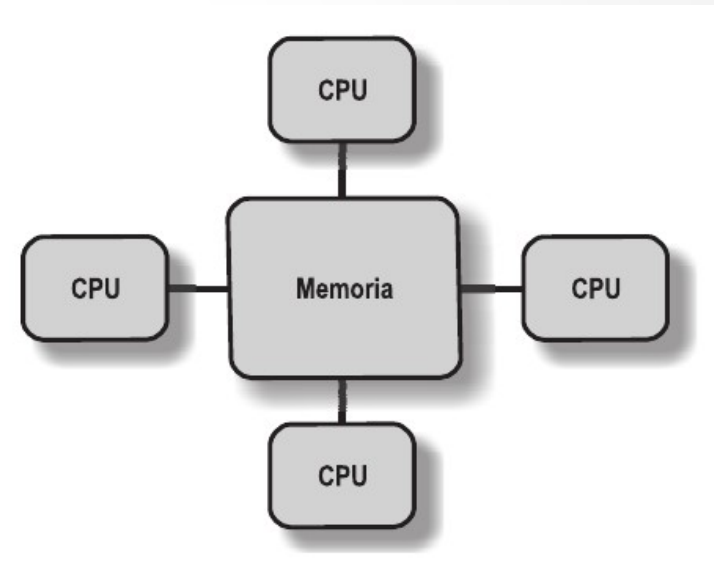
- Hay varios procesadores y cada uno tiene su propia unidad de control, unidad aritmético-lógica y registros.
- Cada procesador tiene acceso a una memoria principal compartida (misma latencia) y a los dispositivos de E/S a través de un bus.
- Los procesadores se pueden comunicar entre ellos a través de la memoria, mediante mensajes de información de estado dejada en un espacio de direcciones compartido.
- Si imaginamos que hemos reducido casi toda la arquitectura de un *cluster* hasta hacerla caber en un *chip*, obtendremos un CMP (*Chip level multiProcessing*) con el que podremos realizar computación multinúcleo (*multi-core computing*) como el que vemos en la figura con cuatro núcleos
- También existen los sistemas de múltiples hilos simétricos (SMT, de Symmetric Multi-Threading); por ejemplo, los procesadores con tecnología *Hyper-Threading* (HT) de Intel.



Sistemas Operativos

Multiprocesamiento – SMP – Memoria UMA

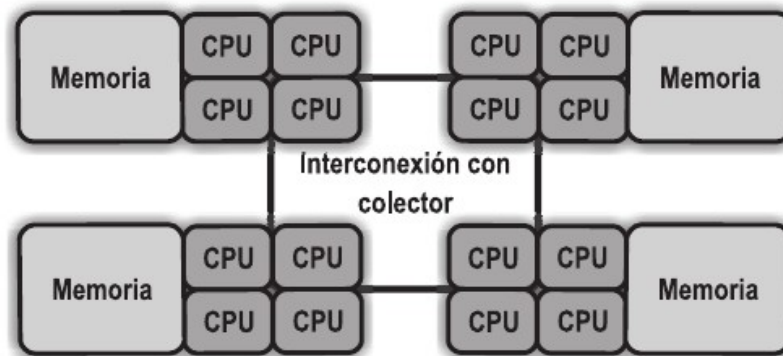
- Si todos los procesos acceden a la memoria de manera uniforme, y si el tiempo de acceso o la localización de memoria es independiente del proceso que realiza la petición o del chip de memoria donde se encuentre el dato solicitado, entonces nos encontramos con una arquitectura UMA (*Uniform Memory Access*, Acceso Uniforme a Memoria)
- A veces también se la ha llamado CC-UMA o Cache Coherent UMA porque si un procesador actualiza una ubicación en esta memoria compartida, esta actualización se hace visible a todos los otros procesadores.
- Esta coherencia de caché se lleva a cabo a nivel de hardware.



Sistemas Operativos

Multiprocesamiento – SMP – Memoria NUMA

- Tenemos otro diseño de memoria en el que ésta se accede en posiciones relativas de otro proceso o memoria compartida entre procesos.
- Los tiempos de acceso diferirán (serán no-uniformes).



- Un procesador podrá acceder a su propia memoria local de forma más rápida que a la memoria no local (memoria local de otro procesador o memoria compartida entre procesadores).
- En NUMA, cada dirección en el espacio de direcciones global está asignada a un nodo fijo.
- Cuando los procesadores acceden a algún dato, se hace una copia en sus cachés locales pero el espacio continúa reservado en el nodo original.