

Sistemas Operativos

Multiprocesadores - I

Facultad de Informática  
UNIVERSIDAD NACIONAL DE LA PLATA

---

---

---

---

---

---

---

---

Sistemas Operativos

☒ Versión: Abril 2023

☒ Palabras Claves: Multiprocesadores, SMP, Redes, Distribuidos, UMA, NUMA

Algunas diapositivas han sido extraídas de las ofrecidas para docentes desde el libro de Stallings (Sistemas Operativos) , el de Silberschatz (Operating Systems Concepts)

Facultad de Informática  
UNIVERSIDAD NACIONAL DE LA PLATA

---

---

---

---

---

---

---

---

Origen

☒ Desde su inicio, la industria de las computadoras se ha orientado fundamentalmente a buscar un poder de cómputo cada vez mayor.

☒ Las necesidades actuales demandan cada vez mayor poder de cómputo (física, astronomía, biología, calculo de modelos)

☒ En el pasado, la solución era siempre hacer que el reloj operara a mayor velocidad

Facultad de Informática  
UNIVERSIDAD NACIONAL DE LA PLATA

---

---

---

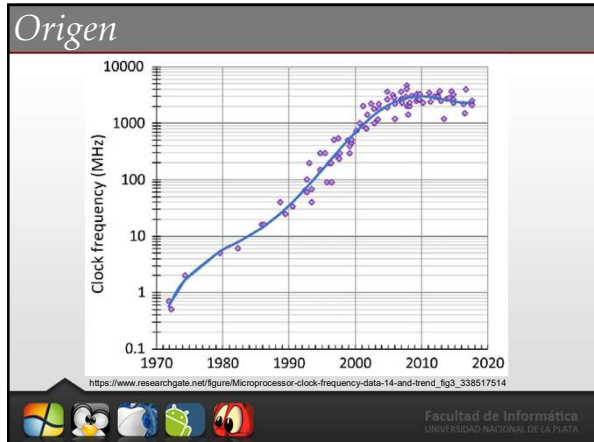
---

---

---

---

---



---

---

---

---

---

---

---

---

- ### Origen
- ☒ En la actualidad, lograr mayor velocidad es físicamente complejo:
    - ☒ Ninguna señal eléctrica se puede propagar más rápido que la velocidad de la luz
    - ☒ Problemas de disipación de calor (muchos transistores juntos en poco espacio)
    - ☒ Problemas de consumo eléctrico
  - ☒ La solución al problema es el cómputo en paralelo y/o distribuido
  - ☒ Contar con varias CPU que operen a velocidad "normal" y que en conjunto provean la potencia de cómputo necesaria
- Facultad de Informática  
UNIVERSIDAD NACIONAL DE LA PLATA

---

---

---

---

---

---

---

---

- ### Origen – Cuestiones Físicas
- ☒ Teoría de la relatividad de Einstein → ninguna señal eléctrica se puede propagar más rápido que la velocidad de la luz: 20 cm/nseg en cobre o FO
  - ☒ En una CPU con un reloj de 10 GHz, las señales deberían viajar más de 2 cm en total.
  - ☒ En una CPU de 100 GHz la longitud total de la ruta sería cuando mucho de 2 mm.
  - ☒ Una CPU de 1 THz (1000 GHz) tendría que ser más pequeña que 100 micrones
  - ☒ Problema fundamental: la disipación del calor.
- Facultad de Informática  
UNIVERSIDAD NACIONAL DE LA PLATA

---

---

---

---

---

---

---

---

## Esquemas

- ✓ Si tenemos que resolver un problema en una única CPU, el esquema de trabajo es sencillo
- ✓ Si tenemos varios problemas y varias CPU, a priori podríamos asignar estáticamente una tarea a cada una de ellas: ← no es lo mas eficiente
  - ✓ Deberá existir un coordinador que se encargue de repartir las tareas
- ✓ Al existir múltiples CPU, la complejidad aumenta en lo que refiere a distribución de tareas, pasaje de mensajes y acceso a memoria:



Facultad de Informática  
UNIVERSIDAD NACIONAL DE LA PLATA

---

---

---

---

---

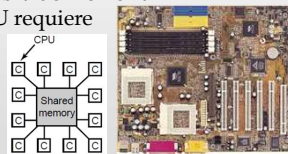
---

---

---

## Esquemas

- ✓ **1. Multiprocesadores con Memoria Compartida**
  - ✓ La comunicación entre las CPU es a través de la memoria compartida
  - ✓ Cada CPU tiene el mismo acceso que otras a la memoria física a través de un único BUS físico
  - ✓ Para acceder a una palabra de memoria por lo general cada CPU requiere de 2 a 10 nseg.
  - ✓ Existe un único espacio lógico de direcciones para todos los procesos



Facultad de Informática  
UNIVERSIDAD NACIONAL DE LA PLATA

---

---

---

---

---

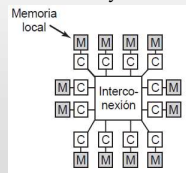
---

---

---

## Esquemas

- ✓ **2. Multicomputadora con memoria independiente / pasaje de mensajes:**
  - ✓ Varios pares de CPU-memoria se conectan a una interconexión de alta velocidad pasando mensajes
  - ✓ Cada memoria es local para una sola CPU y puede ser utilizada sólo por esa CPU
  - ✓ El retardo del paso de mensajes es de entre 10 a 50 µseg



Facultad de Informática  
UNIVERSIDAD NACIONAL DE LA PLATA

---

---

---

---

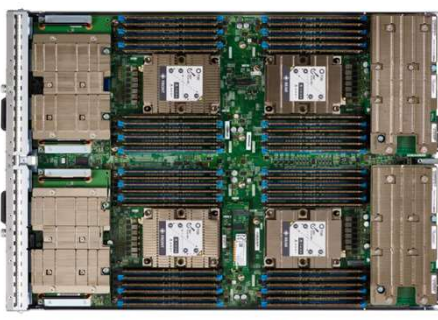
---

---

---

---

## Esquemas



le



Facultad de Informática  
UNIVERSIDAD NACIONAL DE LA PLATA

---

---

---

---

---

---

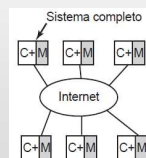
---

---

## Esquemas

### 3. Sistemas Distribuidos:

- ✓ Conecta sistemas de cómputo completos a través de una red
- ✓ Cada sistema tiene su propia memoria, y se comunican mediante el paso de mensajes
- ✓ El retardo del paso de mensajes es de entre 10 a 100 mseg
- ✓ Cada nodo de cómputo es una computadora completa
- ✓ Heterogeneidad de sistemas y hardware



Facultad de Informática  
UNIVERSIDAD NACIONAL DE LA PLATA

---

---

---

---

---

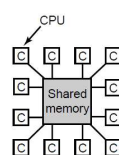
---

---

---

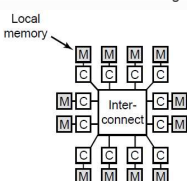
## Esquemas (cont.)

2 a 10 nseg



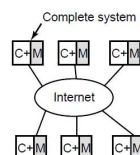
(a)

10000 a 50000 nseg



(b)

10 a 100 mseg



(c)

Diferencias en las velocidades de comunicación



Facultad de Informática  
UNIVERSIDAD NACIONAL DE LA PLATA

---

---

---

---

---

---

---

---

## 1. Multiprocesadores con M. Compartida

- ☒ Dos o mas CPU comparten el acceso a la RAM
- ☒ Los programas se ejecutan en cualquier CPU
- ☒ Cada proceso ve un espacio normal de direcciones virtuales
- ☒ La CPU puede escribir cierto valor en una palabra de memoria y después puede volver a leer esa palabra y obtener un valor distinto (tal vez porque otra CPU lo cambió). Esto puede ser deseable o un problema
- ☒ Si se sincronizan correctamente, una CPU escribe y la otra Lee.



Facultad de Informática  
UNIVERSIDAD NACIONAL DE LA PLATA

---

---

---

---

---

---

---

---

## Multiprocesadores con M. Compartida (cont.)

- ☒ Por lo general, los sistemas operativos realizan tareas regulares como lo son:
  - ☒ Manejo de System Calls
  - ☒ Administración de Memoria
  - ☒ Administración de E/S (sist. de archivos y dispositivos)
- ☒ Algunas características Particulares en sistemas Multiprocesadores:
  - ☒ Sincronización de procesos
  - ☒ Administración de Recursos
  - ☒ Planificación de CPU
- ☒ Todas estas características se ven afectadas en sistemas Multiprocesadores. Hay que atacar las problemáticas que pueden ocurrir



Facultad de Informática  
UNIVERSIDAD NACIONAL DE LA PLATA

---

---

---

---

---

---

---

---

## Hardware de Multiprocesador

- ✓ A nivel Hardware, cada procesador puede direccionar toda la memoria
- ✓ Dependiendo del procesador y la velocidad de acceso a la memoria, los podemos clasificar en:
  - ♦ UMA (Uniform Memory Access)
  - ♦ NUMA (Non-uniform memory Access)



Facultad de Informática  
UNIVERSIDAD NACIONAL DE LA PLATA

---

---

---

---

---

---

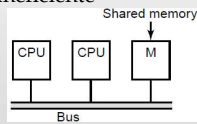
---

---

### Multiprocesadores UMA con arquitecturas basadas en bus

#### ☑ Hardware - UMA

- ✓ Los multiprocesadores más simples se basan en un solo bus para comunicarse con la memoria
- ✓ Antes de acceder al BUS, se debe comprobar que el mismo no se encuentre ocupado
- ✓ A medida que aumenta la cantidad de CPUs, el acceso a memoria se torna mas ineficiente
- ✓ La principal limitación es el ancho de banda del BUS
- ✓ Se tiene mucho tiempo de CPU ocioso



---

---

---

---

---

---

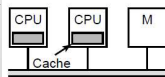
---

---

### Multiprocesadores UMA con arquitecturas basadas en bus

#### ☑ Hardware - UMA (CPU con cache)

- ✓ Una evolución, es agregar una **cache** a cada CPU
- ✓ Al permitir almacenar datos en la cache, el acceso al BUS compartido se reduce.
- ✓ Si el bloque se almacena en la cache en modo RO (read only), puede estar en varias cache
- ✓ Si el bloque se almacena en modo RW, solo puede residir en una cache
- ✓ Aparecen mecanismos de protección para evitar datos "sucios"
- ✓ Copias "limpias" vs. Copias "sucias"



---

---

---

---

---

---

---

---

### Multiprocesadores UMA con arquitecturas basadas en bus

#### ☑ Hardware - UMA (cont. CPU con cache)

- ✓ Aparecen mecanismos de protección para evitar datos "sucios":
  - Cuando una CPU intenta escribir una palabra que esta en una o más caches se manda un mensaje al bus para informar:
    - Si las otras CPU tienen copia limpia (la misma), la descartan para que solo quede la que se va a modificar
    - Si otra CPU tiene copia sucia, la escribe a memoria e informa a la CPU para mantener coherencia
  - A este conjunto de reglas se le conoce como protocolo de coherencia de cachés y es uno de muchos



---

---

---

---

---

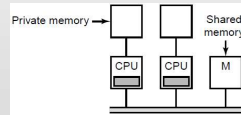
---

---

---

### Multiprocesadores UMA con arquitecturas basadas en bus

- ✓ Hardware – UMA (CPU con cache y memoria local)
  - ✓ Otra alternativa es asignar a cada CPU un área de memoria local que es accedida por un BUS dedicado. → Caché + Memoria Privada
  - ✓ La memoria compartida solo se utiliza para escribir variables compartidas
  - ✓ Reduce el uso del BUS, pero requiere de una asistencia activa por parte del compilador



Facultad de Informática  
UNIVERSIDAD NACIONAL DE LA PLATA

---

---

---

---

---

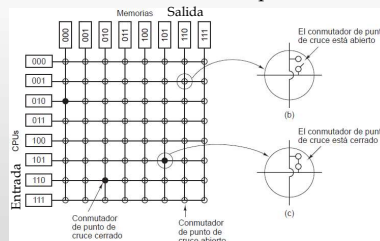
---

---

---

### Multiprocesadores UMA con interruptores de barras cruzadas

- ✓ Inclusive con el agregado de cache, un único BUS limita a 16 o 32 CPU aproximadamente
- ✓ Para lograr mejor performance se necesita otro esquema de interconexión:
- ✓ Interruptores de barras cruzadas:
  - n CPU y n bancos de memoria
  - Permite hasta n conexiones simultáneas
  - Si 2 CPU acceden al mismo módulo hay contención



Facultad de Informática  
UNIVERSIDAD NACIONAL DE LA PLATA

---

---

---

---

---

---

---

---

### Multiprocesadores UMA con redes de conmutación multietapa

- ✓ Inclusive con el agregado de cache, un único BUS limita a 16 o 32 CPU aproximadamente
- ✓ Para lograr mejor performance se necesita otro esquema de interconexión:
- ✓ Redes de conmutación multi-etapa:
  - ✓ Se basa en el concepto de conmutación de n entradas x n salidas
  - ✓ Los mensajes que llegan en cualquiera de las líneas de entrada se pueden conmutar a cualquiera de las líneas de salida
  - ✓ Se forma un mensaje que ingresa al sistema e indica número de módulo, dirección de memoria operación y valor. El mismo es ruteado por el sistema



Facultad de Informática  
UNIVERSIDAD NACIONAL DE LA PLATA

---

---

---

---

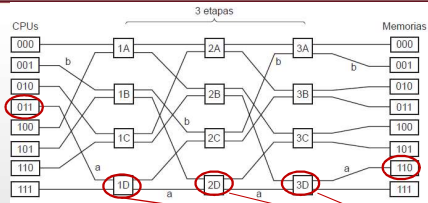
---

---

---

---

### Multiprocesadores UMA con redes de conmutación multietapa



- ❑ La CPU 011 solicita leer un ítem del módulo de memoria 110
- ❑ La CPU envía un mensaje READ al switch 1D con Módulo 110.
- ❑ EL switch analiza el bit más significativo a izquierda y lo utiliza para rutear: 0 rutear hacia arriba, 1 hacia abajo. En este caso se rutea a 2D, posteriormente a 3D y este ultimo al banco de memoria 110



Facultad de Informática  
UNIVERSIDAD NACIONAL DE LA PLATA

---

---

---

---

---

---

---

---

### Multiprocesadores UMA con redes de conmutación multietapa

- ✓ Redes de conmutación multi-etapa
  - ✓ Los mensajes entre CPU y memoria están compuestos por 4 partes:
 

Módulo	Dirección	CódigoOp	Valor

    - ✓ Módulo: Indica el módulo de memoria que se quiere acceder
    - ✓ Dirección: Dirección de memoria dentro del módulo
    - ✓ CódigoOp: Operación a realizar (READ o WRITE)
    - ✓ Valor (opcional): Valor a escribir (unicamente para WRITE)
- ❑ La ventaja frente al esquema de barras cruzadas es que solo requiere 12 switches para realizar la misma tarea
- ❑ Presenta mayor nivel de bloqueos en los circuitos que el esquema de barras cruzadas
- ❑ Los procesadores UMA y las técnicas anteriores tienen la característica de ser poco escalables y muy costosos.



Facultad de Informática  
UNIVERSIDAD NACIONAL DE LA PLATA

---

---

---

---

---

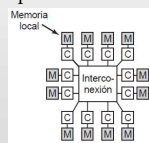
---

---

---

### Hardware – NUMA (Non-uniform memory Access)

- ❑ Cuando es necesario escalar a mayor número de CPUs, hay que pensar en tecnologías que no garanticen el mismo tiempo de acceso a la memoria.
- ❑ Las tecnologías NUMA permiten escalar en número de CPUs
- ❑ Al igual que en UMA, se posee un único espacio de memoria visible por todas las CPUs
  - ❑ El acceso a la memoria remota es más lento que el acceso a la memoria local, ya que se requiere acceso a un bus compartido
- ❑ El rendimiento es menor que en máquinas UMA a mismo Clockrate de CPU, pero es menos costoso



Facultad de Informática  
UNIVERSIDAD NACIONAL DE LA PLATA

---

---

---

---

---

---

---

---



### Hardware – NUMA (Non-uniform memory Access)

- ☑ Características de las máquinas NUMA:
  - ☑ Hay un solo espacio de direcciones visible para todas las CPUs.
  - ☑ El acceso a la memoria remota es mediante instrucciones LOAD y STORE.
  - ☑ El acceso a la memoria remota es más lento que el acceso a la memoria local.



Facultad de Informática  
UNIVERSIDAD NACIONAL DE LA PLATA

---

---

---

---

---

---

---

---

### Hardware – NUMA (Non-uniform memory Access)

- ☑ Se pueden implementar mecanismos de cache para acelerar los tiempos de acceso a la memoria
- ☑ Cuando no hay cache, al sistema se lo llama NC-NUMA (No Cache NUMA), cuando hay cache se lo llama CC-NUMA (Cache Coherent NUMA)
- ☑ En el uso de CC-NUMA, es importante mantener una coherencia en las caches
- ☑ El método mas común para construir grandes procesadores basados en CC-NUMA es el multiprocesador basado en directorios:
  - ✓ Mantiene una BD que indica donde está cada línea y su estado (limpia o sucia (modificada))
  - ✓ La BD debe almacenarse en hardware de propósito específico por cuestiones de performance



Facultad de Informática  
UNIVERSIDAD NACIONAL DE LA PLATA

---

---

---

---

---

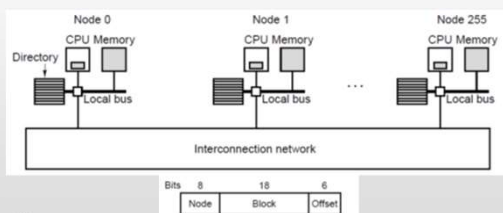
---

---

---

### Hardware – NUMA (Non-uniform memory Access)

- ☑ Hardware – NUMA (cont.)
  - ☑ Una dirección de memoria debe traducirse en:  
nodo+línea+desplazamiento



Facultad de Informática  
UNIVERSIDAD NACIONAL DE LA PLATA

---

---

---

---

---

---

---

---

### Chips Multinúcleo

- ✓ A medida que la tecnología avanza, los transistores se hacen mas pequeños y aparece la posibilidad de agregar más de uno a un chip.
- ✓ Al tener mas transistores se puede:
  - ✓ Agregar más memoria cache → Está demostrado que la tasa de aciertos no se incrementa demasiado
  - ✓ Agregar mas velocidad de clock a una CPU → Sigue existiendo un único hilo de ejecución
  - ✓ Agregar más CPU al mismo chip (núcleos), los que podrían compartir la cache y la memoria principal → Se logra paralelismo
- ✓ Es importante que el software se diseñe teniendo en cuenta los aspectos del hardware para aprovecharlo al máximo



Facultad de Informática  
UNIVERSIDAD NACIONAL DE LA PLATA

---

---

---

---

---

---

---

---

### Tipos de SO Multiprocesador

- ☐ Hasta el momento hicimos un análisis del Hardware Multiprocesador
- ☐ Analicemos como el hardware es manejado por los distintos Sistemas Operativos (**software**)
- ☐ Existen diversas metodologías posibles para la administración en esquemas multiprocesadores desde el lado del **software**:



Facultad de Informática  
UNIVERSIDAD NACIONAL DE LA PLATA

---

---

---

---

---

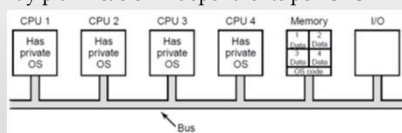
---

---

---

### Tipos de SO Multiprocesador

1. **Cada CPU con su SO (modelo poco utilizado):**
  - ☐ Se divide estáticamente la memoria para cada CPU con su copia privada (las CPUs operan independientes)
  - ☐ Se comparte el código de SO
  - ☐ Cada CPU cuenta con su propio conjunto de procesos
    - Desbalance en la carga de trabajo
  - ☐ Hay planificación independiente por CPU



Facultad de Informática  
UNIVERSIDAD NACIONAL DE LA PLATA

---

---

---

---

---

---

---

---

### Tipos de SO Multiprocesador

- ❑ Existen diversas metodologías posibles para la administración en esquemas multiprocesadores desde el lado del software:

#### 1. Cada CPU con su SO:

- ❑ Cada CPU atrapa y maneja las SysCalls de sus procesos
- ❑ Los procesos quedan "atados" a una única CPU
- ❑ No se pueden compartir paginas
  - ❑ Pasaje de mensajes
  - ❑ Memoria desperdiciada
- ❑ Cache de disco, cada CPU tiene su propia copia
  - ♦ Inconsistencia de la información



Facultad de Informática  
UNIVERSIDAD NACIONAL DE LA PLATA

---

---

---

---

---

---

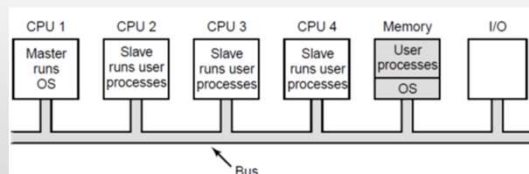
---

---

### Tipos de SO Multiprocesador

#### 2. Maestro - Esclavo

- ❑ Única copia del SO y de su información
- ❑ Todas las SysCalls se redirigen a una CPU
- ❑ Está CPU puede ejecutar procesos si "le sobra tiempo"



Facultad de Informática  
UNIVERSIDAD NACIONAL DE LA PLATA

---

---

---

---

---

---

---

---

### Tipos de SO Multiprocesador

#### 2. Maestro - Esclavo

- ❑ Resuelve problemas del modelo anterior
  - ❑ Se mantiene una única cola de planificación
  - ❑ Cuando una CPU está libre, pide un proceso al master
  - ❑ Se pueden asignar las páginas entre todos los procesos de manera dinámica.
- ❑ Problema, con muchas CPUs hay un cuello de botella en el Maestro
  - ♦ Ej: Si el 10% del tiempo se atienden SysCalls, con 10 CPU el master se saturaría, con 11 estaría sobrecargado



Facultad de Informática  
UNIVERSIDAD NACIONAL DE LA PLATA

---

---

---

---

---

---

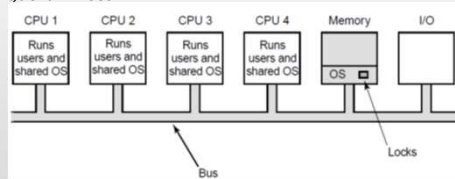
---

---

### Tipos de SO Multiprocesador

#### 3. SMP - Multiprocesadores Simétricos

- ☐ Soluciona el inconveniente de saturación de una única CPU
- ☐ Única copia del SO en memoria y cualquier CPU puede ejecutarlo
- ☐ Cuando se invoca una System Call, es ejecutada por la CPU que la invocó



Facultad de Informática  
UNIVERSIDAD NACIONAL DE LA PLATA

---

---

---

---

---

---

---

---

### Tipos de SO Multiprocesador

#### 3. SMP - Multiprocesadores Simétricos (cont.)

- ☐ Provee equilibrio entre procesos y memoria, ya que solo hay un único conjunto de tablas del SO
- ☐ No hay cuello de botella, ya que no hay una CPU master
- ☐ Problemas
  - ♦ Dos o mas CPUs ejecutando código del SO en un mismo instante de tiempo
  - ♦ Dos CPUs seleccionando el mismo proceso para ejecutar, o seleccionan la misma página de la memoria libre!



Facultad de Informática  
UNIVERSIDAD NACIONAL DE LA PLATA

---

---

---

---

---

---

---

---

### Tipos de SO Multiprocesador

#### 3. SMP - Multiprocesadores Simétricos (cont.)

- ♦ Posibles soluciones a los problemas planteados:
  1. Utilizar "locks" para las estructuras del SO:
    - ♦ Considerar a todo el SO como una gran sección crítica. Cualquier CPU puede ejecutar el SO, pero solo una a la vez.
    - ♦ Se comportaría como el modelo maestro-esclavo
    - ♦ Es un modelo poco utilizado, debido a la mala performance que provee



Facultad de Informática  
UNIVERSIDAD NACIONAL DE LA PLATA

---

---

---

---

---

---

---

---

## Tipos de SO Multiprocesador

### 3. SMP - Multiprocesadores Simétricos (cont.)

- ♦ Posibles soluciones a los problemas planteados:

#### 2. Lock por estructura(s):

- ♦ Existen varias secciones críticas independientes, cada una protegida por su propio mutex
- ♦ Mejora el rendimiento
- ♦ Dificultad para determinar cada sección crítica
- ♦ **Ciertas estructuras pueden pertenecer a más de una sección crítica**, lo cual ante bloqueos podría generar **Deadlocks**
- ♦ Es el esquema que generalmente se utiliza



Facultad de Informática  
UNIVERSIDAD NACIONAL DE LA PLATA

---

---

---

---

---

---

---

---

## Sincronización de Multiprocesadores

- ☑ Es necesario que las CPU de un multiprocesador se encuentren sincronizadas (acceso a regiones críticas, estructuras, etc.).
- ☑ En entornos uniprocador si un proceso realiza una llamada al sistema que requiera acceder a cierta tabla crítica del kernel, el código del kernel sólo tiene que deshabilitar las interrupciones antes de tocar la tabla.
- ☑ En sistemas multiprocesadores, se deshabilitan las interrupciones de una CPU, pero otra podría generarlas...
- ☑ Surge la necesidad de contar con un protocolo de mutex apropiado para garantizar la exclusión mutua.



Facultad de Informática  
UNIVERSIDAD NACIONAL DE LA PLATA

---

---

---

---

---

---

---

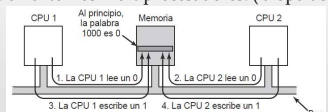
---

## Sincronización de Multiprocesadores

- ✓ Una posibilidad para garantizar la exclusión mutua es el uso de TSL (Probar y establecer bloqueo):

- ✓ Lee la palabra de memoria y la almacena en un registro. Al mismo tiempo escribe un 1 en la memoria para hacer el lock (2 accesos al BUS). Cuando termina libera (escribe 0). En uniprocadores esta implementación es correcta.

- ✓ El problema surge en entornos multiprocesadores: (la operación no es indivisible)



- ✓ Ambas CPU obtuvieron un 0 de la instrucción TSL, por lo que ambas tienen acceso a la sección crítica



Facultad de Informática  
UNIVERSIDAD NACIONAL DE LA PLATA

---

---

---

---

---

---

---

---

### Sincronización de Multiprocesadores

- ✓ La solución al problema anterior, es que en multiprocesadores la instrucción TSL bloquee el acceso al BUS
- ✓ Se necesita soporte de hardware para poder implementarlo
- ✓ Genera Carga en la memoria y el BUS, ya que la CPU que solicita debe mantener el bloqueo y las otras CPU deben esperar liberación del bloqueo
- ✓ No es lo más eficiente...



Facultad de Informática  
UNIVERSIDAD NACIONAL DE LA PLATA

---

---

---

---

---

---

---

---

### Sincronización de Multiprocesadores

- ✓ Surge una nueva solución que es el uso de cache (para evitar el bus), pero también genera problemas:
  - ♦ Al leer la palabra en la cache, generalmente se realizan modificaciones
  - ♦ Como se modifica, se deben invalidar todas las copias de las otras caches (trashing), lo que causa que se deba escribir el valor a la memoria y las otras CPUs lo releen.
  - ♦ Esto genera mayor uso del BUS
  - ♦ Además al tener un lockeo establecido por una CPU, las otras están continuamente consultando por la liberación



Facultad de Informática  
UNIVERSIDAD NACIONAL DE LA PLATA

---

---

---

---

---

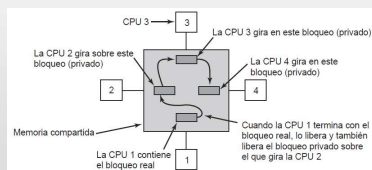
---

---

---

### Sincronización de Multiprocesadores

- ✓ Otras soluciones
  - ♦ Cada CPU tiene su propia variable de lockeo en cache
  - ♦ La CPU que no puede obtener el bloqueo se agrega a una lista y espera en su propio lock
  - ♦ Agregar "delays" entre cada intento de TSL



Facultad de Informática  
UNIVERSIDAD NACIONAL DE LA PLATA

---

---

---

---

---

---

---

---