

# Multitarea y Control de Concurrency

Universidad Autónoma de Nuevo León

Facultad de Ingeniería Mecánica y Eléctrica

EQUIPO 2

## **SISTEMAS OPERATIVOS**

Docente: Norma Edith Marín Martínez

Grupo: 004    Salón: 3103



# Integrantes

MATRICULA	NOMBRE	CARRERA
1950074	José Ahmed Vela Canales	IAS
1949469	Saúl Edrei Silva Rodríguez	IAS
1948705	Gabriel Monroy García	IAS

# Índice



Introducción----- 4



Windows ----- 5



Tipos de procesos concurrentes ----- 19



Tipos de interacciones dentro de la concurrencia----- 22



Procesos concurrentes ----- 25



Referencias bibliográficas ----- 27

# Introducción

En tecnología de la información y Ciencias de la computación, especialmente en los campos de programación el **control de concurrencia** asegura corregir resultados para las operaciones que se generan, al obtener esos resultados tan pronto como sea posible.

Sistemas informáticos, tanto software y hardware, consisten en módulos o componentes. Cada componente está diseñado para funcionar correctamente, es decir, las reglas para obedecer o para cumplir con cierta consistencia. Cuando los componentes que operan simultáneamente interactúan por mensajería o por compartir acceso a datos (en memoria o almacenamiento de información), consistencia de un determinado componente puede ser violado por otro componente. El área general del control de concurrencia proporciona reglas, métodos, metodologías de diseño, y teorías para mantener la consistencia de los componentes operando simultáneamente mientras interactúan y así la consistencia y la corrección de todo el sistema.



Windows



# Procesos e hilos

Un proceso es un contenedor para los recursos y demás elementos que serán utilizados por los hilos (o threads), encargados de ejecutar las instrucciones que componen el programa.

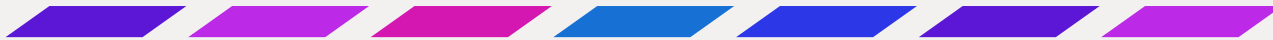


# Resumen de las fases en que se divide la creación de un proceso:

Abrir el fichero de imagen (.exe) que se ejecutará mediante el proceso.



Crear el objeto para el bloque ejecutivo de proceso (`_EPROCESS`).



Crear el hilo de ejecución primario (pila, contexto y bloque ejecutivo de hilo `_ETHREAD`).



Comenzar la ejecución del hilo primario.



En el contexto del nuevo proceso y su hilo asociado completar la inicialización del espacio de direcciones (p.e. mediante la carga de las DLL requeridas) y empezar la ejecución del programa.



# Estados de energía

Características sistema S1:

Consumo de energía

Menor consumo que en S0 y mayor que en los demás estados de suspensión. El reloj del procesador está apagado y los relojes de bus se detienen.

Reanudación de software

El control se reinicia donde lo dejó.

Latencia de hardware

Normalmente, no más de dos segundos.

Contexto de hardware del sistema

Todo el contexto retenido y mantenido por el hardware.



# Características:

Sistema S2:

El estado de energía del sistema S2 es similar a S1, salvo que el contexto de CPU y el contenido de la memoria caché del sistema se pierden porque el procesador pierde energía. El estado S2 tiene las siguientes características:

Consumo de energía

Menor consumo que en el estado S1 y mayor que en S3. El procesador está desactivado. Se detienen los relojes de bus; algunos buses pueden perder energía.

Reanudación de software

Después de la reactivación, el control se inicia desde el vector de restablecimiento del procesador.

Latencia de hardware

# Sistema S3

Consumo de energía

Menos consumo que en el estado S2. El procesador está apagado y algunos chips de la placa base también podrían estar apagados.

Reanudación de software

Después del evento de reactivación, el control se inicia desde el vector de restablecimiento del procesador.

Latencia de hardware

Casi indistinguible de S2.

Contexto de hardware del sistema

# Sistema 4

Consumo de energía

Desactivado, excepto en el caso de la corriente complicada del botón de encendido y dispositivos similares.

Reanudación de software

El sistema se reinicia desde el archivo de hibernación guardado. Si no se puede cargar el archivo de hibernación, es necesario reiniciar. La reconfiguración del hardware mientras el sistema está en el estado S4 podría dar lugar a cambios que impiden que el archivo de hibernación se cargue correctamente.

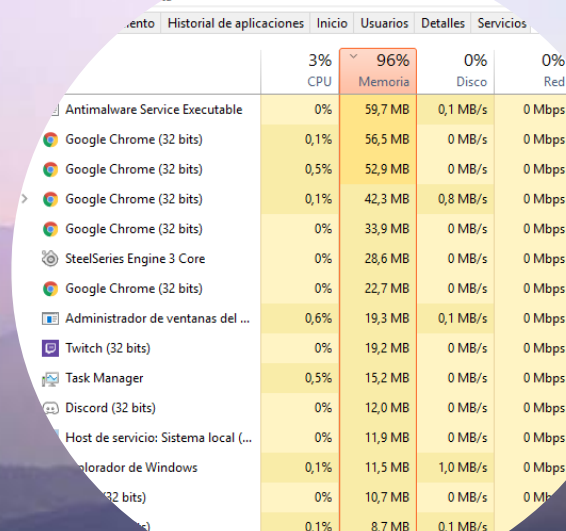


# Transiciones

- Las transiciones son animaciones que se usan para mantener a los usuarios orientados durante los cambios de estado de la interfaz de usuario (UI) y las manipulaciones de objetos.
- Las buenas transiciones se sienten naturales, a menudo dando la ilusión de que los usuarios interactúan con objetos reales.

# Control y Gestión de Windows

1. El Sistema de Windows lleva a cabo su control y gestión desde el administrador de tareas, el administrador de tareas es una aplicación integrada en los sistemas operativos de Windows con la cual podremos obtener información de los programas y procesos que se ejecutan en el equipo, además de proporcionar los indicadores de rendimientos más utilizados.



The screenshot shows the Windows Task Manager Performance tab. At the top, there are tabs for 'Inicio', 'Usuarios', 'Detalles', and 'Servicios'. Below these, a table displays system resource usage. The 'Memoria' (Memory) section is highlighted in red, showing 96% usage. The 'Disco' (Disk) and 'Red' (Network) sections show 0% usage. The 'CPU' section shows 3% usage. Below the resource usage, a list of running processes is shown with columns for CPU, Private Memory, Working Set, and I/O.

	3% CPU	96% Memoria	0% Disco	0% Red
Antimalware Service Executable	0%	59,7 MB	0,1 MB/s	0 Mbps
Google Chrome (32 bits)	0,1%	56,5 MB	0 MB/s	0 Mbps
Google Chrome (32 bits)	0,5%	52,9 MB	0 MB/s	0 Mbps
Google Chrome (32 bits)	0,1%	42,3 MB	0,8 MB/s	0 Mbps
Google Chrome (32 bits)	0%	33,9 MB	0 MB/s	0 Mbps
SteelSeries Engine 3 Core	0%	28,6 MB	0 MB/s	0 Mbps
Google Chrome (32 bits)	0%	22,7 MB	0 MB/s	0 Mbps
Administrador de ventanas del ...	0,6%	19,3 MB	0,1 MB/s	0 Mbps
Twitch (32 bits)	0%	19,2 MB	0 MB/s	0 Mbps
Task Manager	0,5%	15,2 MB	0 MB/s	0 Mbps
Discord (32 bits)	0%	12,0 MB	0 MB/s	0 Mbps
Host de servicio: Sistema local (...)	0%	11,9 MB	0 MB/s	0 Mbps
Explorador de Windows	0,1%	11,5 MB	1,0 MB/s	0 Mbps
Google Chrome (32 bits)	0%	10,7 MB	0 MB/s	0 Mbps
Google Chrome (32 bits)	0,1%	8,7 MB	0,1 MB/s	0 Mbps

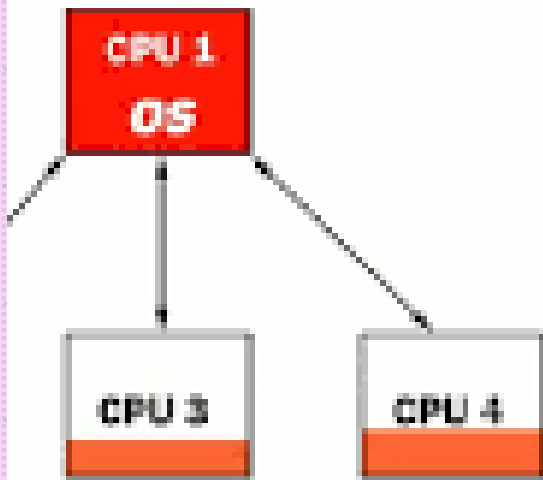


# Multiprocesamiento simétrico

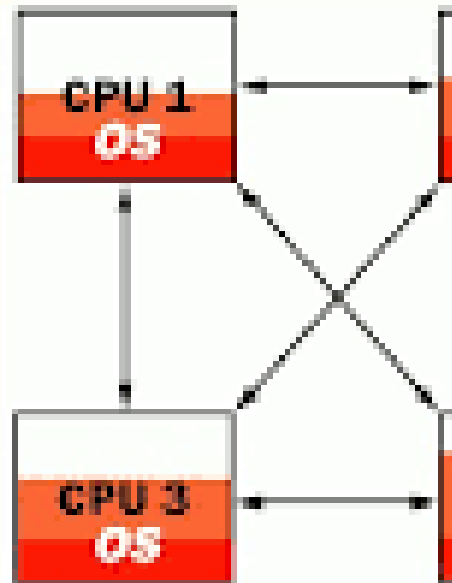
1. El multiprocesamiento simétrico es aquel en el que todo el procesador ejecuta las tareas en el sistema operativo. No tiene una relación maestro-esclavo como multiprocesamiento asimétrico. Todos los procesadores aquí, se comunican utilizando la memoria compartida.

Bases para la comparación	Multiprocesamiento simétrico
<b>BASIC</b>	Cada procesador ejecuta las tareas en el sistema operativo.
<b>Proceso</b>	El procesador toma los procesos de una cola preparada común o puede haber una cola preparada para cada procesador.
<b>Arquitectura</b>	Todo el procesador en multiprocesamiento simétrico tiene la misma arquitectura.
<b>Comunicación</b>	Todos los procesadores se comunican con otro procesador mediante una memoria compartida.
<b>Fracaso</b>	Si un procesador falla, la capacidad de cálculo del sistema se reduce.
<b>Facilitar</b>	El multiprocesador simétrico es complejo ya que todos los procesadores deben estar sincronizados para mantener el equilibrio de la carga.

## Multiproceso OS (continuación)



Asimétrico



Simétrico

# Diferencias clave entre multiprocesamiento simétrico y asimétrico

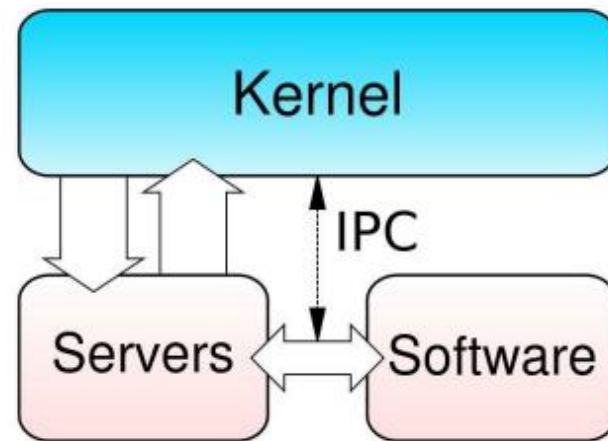
El punto más distinguible entre el multiprocesamiento simétrico y asimétrico es que las tareas en el sistema operativo son manejadas solamente por el procesador maestro en el multiprocesamiento asimétrico. Por otro lado, todos los procesadores en multiprocesamiento simétrico ejecutan las tareas en el sistema operativo.

En el multiprocesamiento simétrico, cada procesador puede tener su propia cola privada de procesos listos, o pueden tomar procesos de una cola lista común. Pero, en el multiprocesamiento asimétrico, el procesador maestro asigna procesos a los procesadores esclavos.

Todo el procesador en Multiprocesamiento Simétrico tiene la misma arquitectura. Pero la estructura de los procesadores en multiprocesador asimétrico puede diferir.

# Micronucleos

1. Un micronúcleo (en inglés, microkernel o  $\mu$ kernel) es un tipo de núcleo de un sistema operativo que provee un conjunto de primitivas o llamadas mínimas al sistema para implementar servicios básicos como espacios de direcciones, comunicación entre procesos y planificación básica. Todos los otros servicios (gestión de memoria, sistema de archivos, operaciones de E/S, etc.), que en general son provistos por el núcleo, se ejecutan como procesos servidores en espacio de usuario.



## Ejemplos de micronúcleos

- AIX
- La familia de micronúcleos L4
- El micronúcleo Mach, usado en GNU Hurd y en Mac OS X
- BeOS
- Minix
- MorphOS
- QNX
- RadiOS
- VSTa
- Hurd

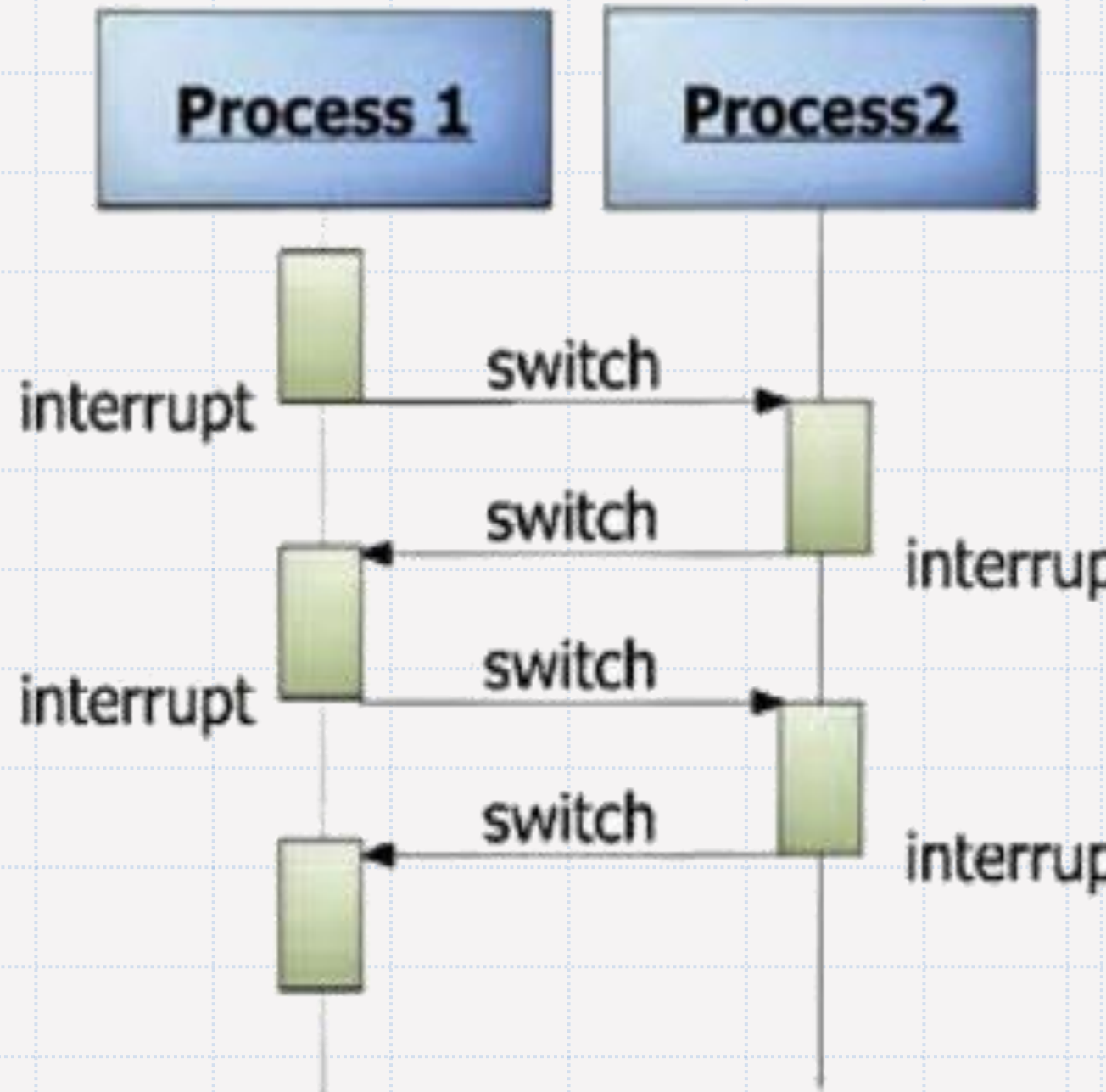


# Gestion de hilos

1. Existen dos maneras de poder gestionar los hilos en un sistema operativo:
2. **Hilos de usuario (ULT):** Mediante este método una aplicación realiza todo el trabajo de la gestión de hilos mientras el kernel no es consciente de la existencia de estos.
3. **Hilos de Kernel (KLT):** Mediante este método los hilos son gestionados directamente por el kernel y en la aplicación no existe ningún código para la gestión de estos. Este método es usado en Windows 2000 y algunas versiones de Linux

# Multiprocesos

- Microsoft Windows permite la implementación de multitareas preventivas, para esto, Windows divide el tiempo de procesador disponible entre los diferentes procesos o hilos que lo requieran, entonces cuando un hilo se suspende debido a que su parte del tiempo se agota esto permite la ejecución de otro hilo diferente. El sistema al cambiar de un hilo a otro guarda el contexto del hilo reemplazado y restaura el contexto guardado del siguiente subproceso en la cola. Debido a que cada hilo se ejecuta en un tiempo aproximado de 20 milisegundos, el sistema da la sensación de ejecutar múltiples hilos a la vez.



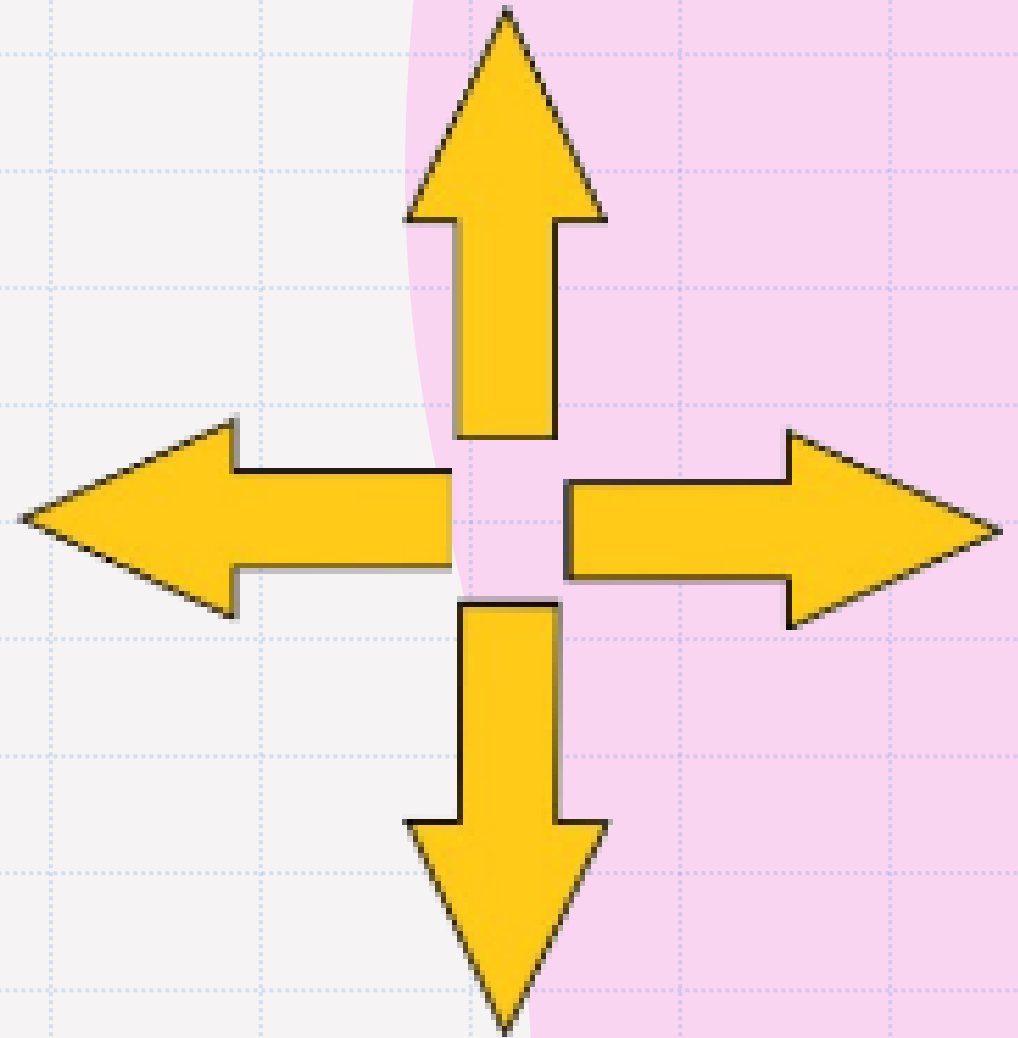


# Tipos de procesos concurrentes

Los procesos que se ejecutan de manera concurrente en un sistema operativo se pueden clasificar en dos tipos:

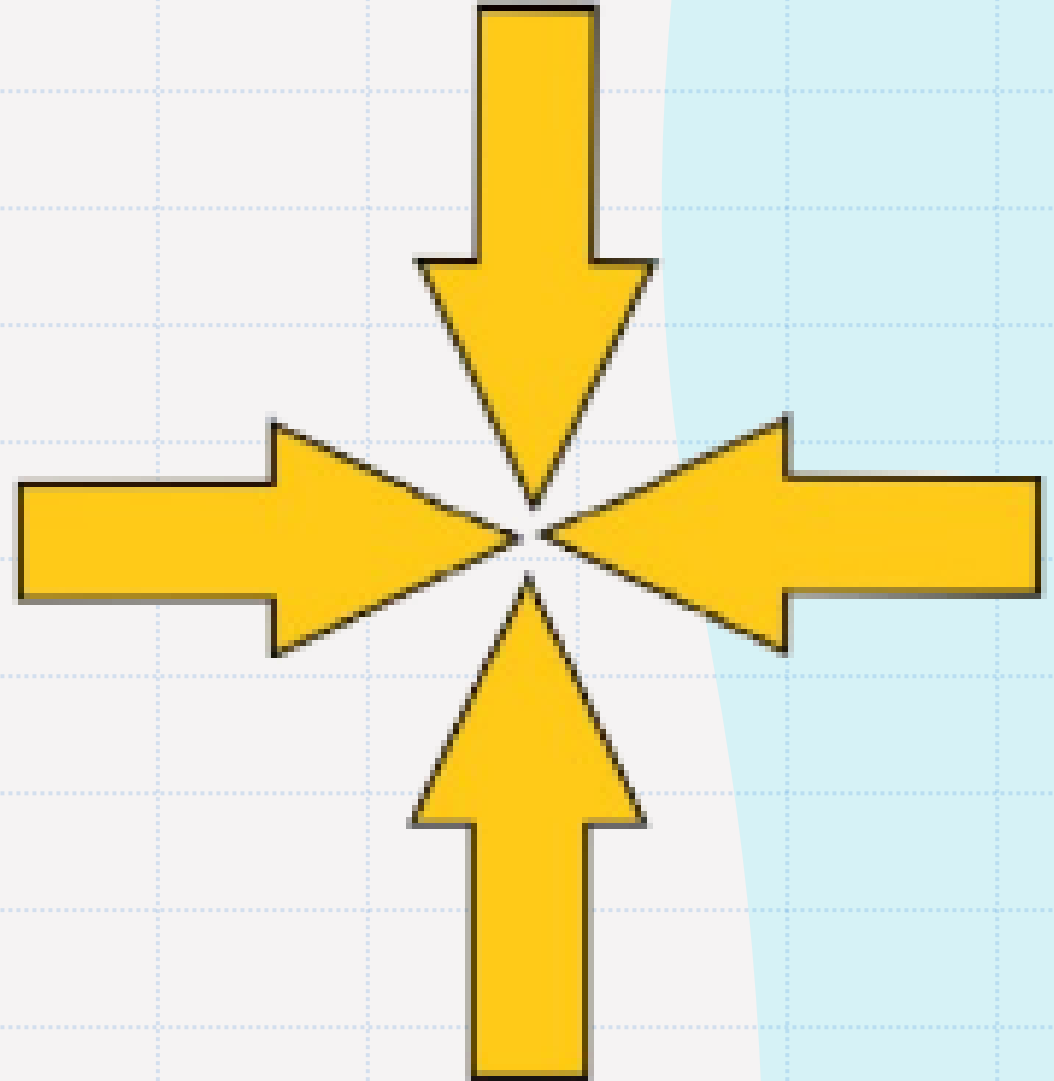
# Procesos independientes

- Este tipo de procesos se ejecutan sin necesidad de recibir ayuda de otro proceso, por lo cual no puede afectar ni ser afectado por otros procesos que se ejecuten en el sistema, a su vez estos procesos compiten por el uso de los recursos. Un ejemplo de este tipo de procesos puede ser un programa que calcula 1000 cifras decimales de  $\pi$ .



# Procesos cooperativos

Este tipo de procesos están diseñados para que trabajen de manera conjunta con otros procesos para la realización de una actividad. Estos procesos deben ser capaces de intercambiar datos entre ellos. Un ejemplo de procesos cooperantes sería un proceso que escribe en la terminal la cadena “abc” y en otro la cadena “cba”.





# Tipos de interacciones dentro de la concurrencia

# Primero comunicacion

Si tenemos varios procesos ejecutando y queremos que interactúen entre ellos para solucionar un determinado problema, y por tanto, necesitan intercambiar información, es decir necesitan comunicarse.

Mediante la comunicación se consigue que la ejecución de un proceso influya en la ejecución de otro.



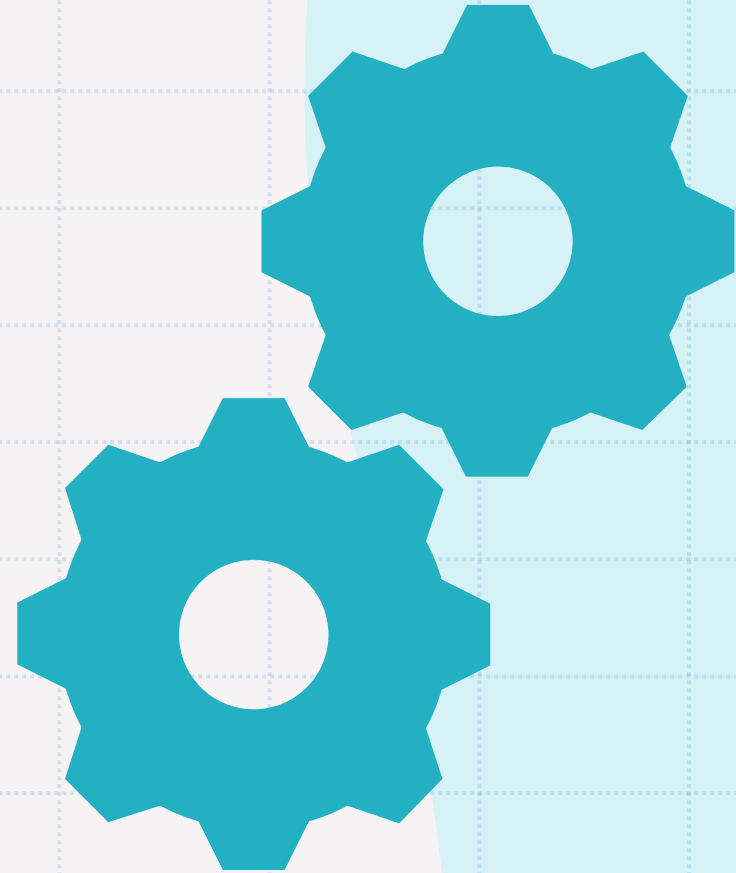




# Segunda sincronizacion

- La sincronización (sinónimo de espera) es habitualmente necesaria cuando dos procesos se comunican puesto que la velocidad a la que se ejecutan no es predecible
- La necesidad de contar con comunicación y sincronización puede verse en términos de competencia y cooperación entre procesos.
- La competencia ocurre cuando el proceso requiere el uso exclusivo de un recurso.
- La cooperación ocurre cuando dos procesos trabajan en distintas partes del mismo problema y por lo general incluyen a la comunicación y a la sincronización.

# Procesos concurrentes



## PROCESOS CONCURRENTES

Un programa concurrente es aquel en el que ciertas unidades de ejecución internamente secuenciales (procesos o threads), se ejecutan paralela o simultáneamente.

### PROCESO CONCURRENTE INDEPENDIENTE

es aquel que ejecuta sin requerir la ayuda o cooperación de otros procesos.

Un claro ejemplo de procesos independientes son los diferentes intérpretes de mandatos que se ejecutan de forma simultánea en un sistema

### PROCESO CONCURRENTE COOPERATIVO

Idea secundaria

Idea secundaria

– Interacción motivada porque los procesos se comunican y sincronizan entre sí para alcanzar un objetivo común. Por ejemplo, los **procesos compilador y ensamblador** descritos anteriormente son dos procesos que deben comunicarse y sincronizarse entre ellos con el fin de producir código en lenguaje máquina

– **Interacción motivada** porque los procesos comparten o compiten por el acceso a recursos físicos o lógicos. Esta situación aparece en los distintos tipos de procesos anteriormente comentados. Por ejemplo, dos procesos totalmente independientes pueden competir por el acceso al disco.

# Conclusión

Todos los mecanismos de control de concurrencia deben asegurar la consistencia de los objetos y cada transacción atómica será completada en un tiempo finito. Un método de control de concurrencia es correcto si es serializable, es decir existe una secuencia equivalente en que las operaciones de cada transacción aparecen antes o después de otra transacción pero no entremezcladas. Una ejecución serial de transacciones es siempre correcta.

Se debe sincronizar las transacciones concurrentes de los usuarios, extendiendo los argumentos para la serializabilidad y los algoritmos de control de concurrencia para la ejecución en ambientes distribuidos. La finalidad del control de concurrencia es asegurar la consistencia de los datos al ejecutar transacciones, y que cada acción atómica sea completada en un tiempo finito.

# Conclusiones Individuales

## **José Amhed Vela Canales:**

La multitarea es esencial en la informática moderna, permitiendo la ejecución simultánea de múltiples tareas o procesos. Sin embargo, para garantizar un funcionamiento eficiente y sin conflictos en sistemas multitarea, es fundamental implementar técnicas sólidas de control de concurrencia.

## **Saúl Edrei Silva Rodríguez:**

La relación entre la multitarea y el control de concurrencia es evidente en la informática moderna. La multitarea permite a los sistemas realizar múltiples tareas al mismo tiempo, mientras que el control de concurrencia garantiza que esta multitarea se ejecute de manera armoniosa al evitar interferencias y conflictos entre los procesos concurrentes. Ambos conceptos se complementan y son esenciales para lograr sistemas informáticos robustos.

## **Gabriel Monroy García:**

El control de concurrencia desempeña un papel crítico en la programación y gestión de sistemas informáticos, especialmente en entornos multitarea. Garantizar que múltiples procesos compartan recursos de manera segura y ordenada es esencial para prevenir conflictos y garantizar la integridad de los datos.



# Enlace al video

<https://youtu.be/5H3SZLSDvJE>

# Referencias bibliograficas

H. (2022, 12 mayo). *Animaciones y transiciones - Win32 apps*. Microsoft Docs. Recuperado 26 de agosto de 2022, de <https://docs.microsoft.com/es-es/windows/win32/uxguide/vis-animations> Mario Alejandro Moreno Coronel

WEBEDIA BRAND SERVICES. (2014). El administrador de tareas de Windows: qué es y cómo funciona. Julio 14, 2014, de xatakawindows Sitio web: <https://www.xatakawindows.com/bienvenidoawindows8/el-administrador-de-tareas-de-windows-que-es-y-como-funciona>

Manuel M., J. (2019, 30 septiembre). *Diferencias entre Multiprocesamiento Simétrico y Asimétrico*. PC Solución. Recuperado 26 de agosto de 2022, de <https://pc-solucion.es/2018/04/18/diferencias-entre-multiprocesamiento-simetrico-y-asimetrico/>

Kare G. 11/05/2021 España  
[https://www.google.com/url?sa=t&source=web&rct=j&url=https://docs.microsoft.com/es-es/windows/win32/power/system-power-states&ved=2ahUKEwiMIJ\\_2uOX5AhWekWoFHez-CPcQFnoECBUQAQ&usg=AOvVaw2USNdulk-Qp87pR5tIVAMR](https://www.google.com/url?sa=t&source=web&rct=j&url=https://docs.microsoft.com/es-es/windows/win32/power/system-power-states&ved=2ahUKEwiMIJ_2uOX5AhWekWoFHez-CPcQFnoECBUQAQ&usg=AOvVaw2USNdulk-Qp87pR5tIVAMR)