

Universidad Autónoma de Nuevo León

Facultad de Ingeniería Mecánica y Eléctrica



FIME

FACULTAD DE INGENIERÍA MECÁNICA Y ELÉCTRICA

Unidad de Aprendizaje: Sistemas Operativos

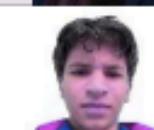
Catedrática: Dr. Norma Edith Marín Martínez

Multitarea y Control de Concurrencia

Semestre: Agosto- Diciembre 2025

Hora: M3, Grupo 004, Salón 4211

Equipo 2

Fotografía	Nombre	Matrícula	Carrera	Porcentaje
	Alvarado Cantú Lesly Elizabeth	2177856	ITS	100%
	Ávila Ignacio Jesús Emiliano	2118747	IAS	100%
	Balderas García Ana Victoria	2106047	IAS	100%
	Garza Cruz Santiago	2177955	ITS	100%
	Guevara Ochoa Eduardo Miguel	2053056	ITS	100%

Fotografía	Nombre	Matrícula	Carrera	Porcentaje
	Méndez Sánchez Marco Antonio	2177912	ITS	100%
	Pérez Rodríguez Jazmín	2103948	ITS	100%
	Ramón López Anthony Joel	2058255	ITS	100%
	Valdez Silva Ángel Karim	2041829	ITS	100%

ÍNDICE

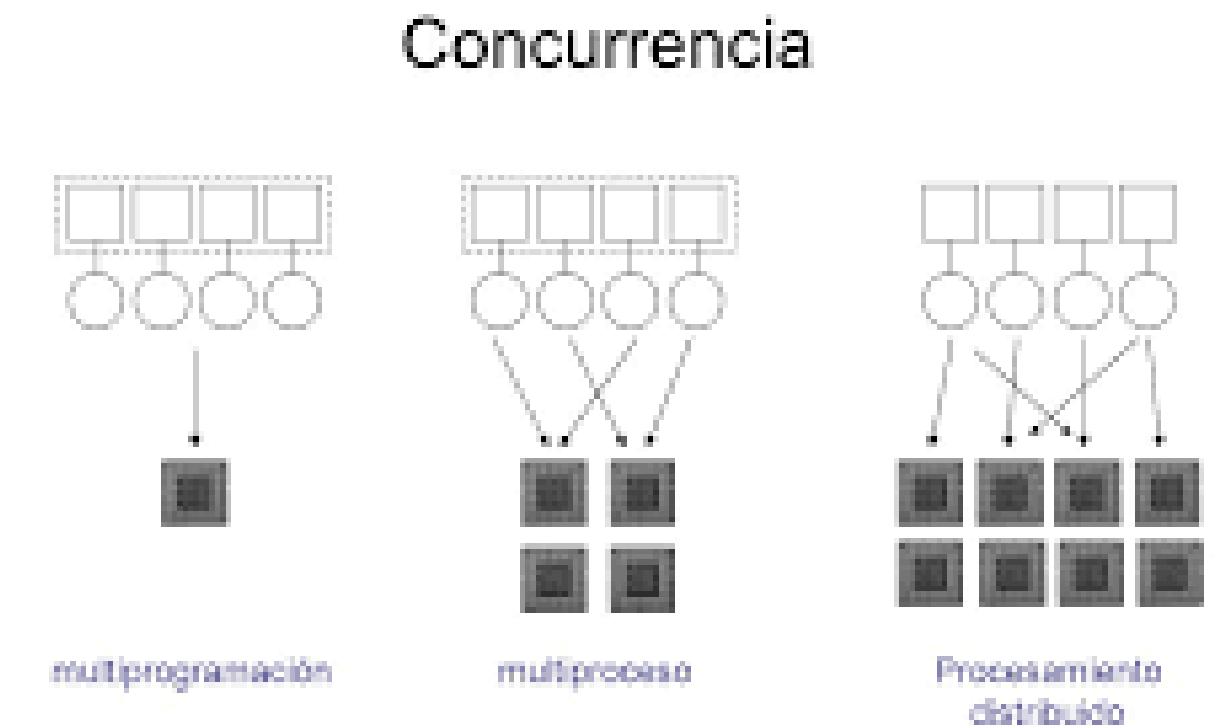
03	Introducción	13	Estados y...	34	Tipos de interacciones...	50	SMP en...
04	Concurrencia	20	Tipos de...	36	Control y gestión...	54	Cuadro sinóptico
05	Partes que...	26	Modelos de...	41	Multiprocesamiento...	59	Conclusión grupal
06	Procesos	31	Ventajas de la...	43	Micro núcleos	60	Conclusiones
10	Hilos	33	Tipos de procesos...	48	Gestión de Hilos		

INTRODUCCIÓN

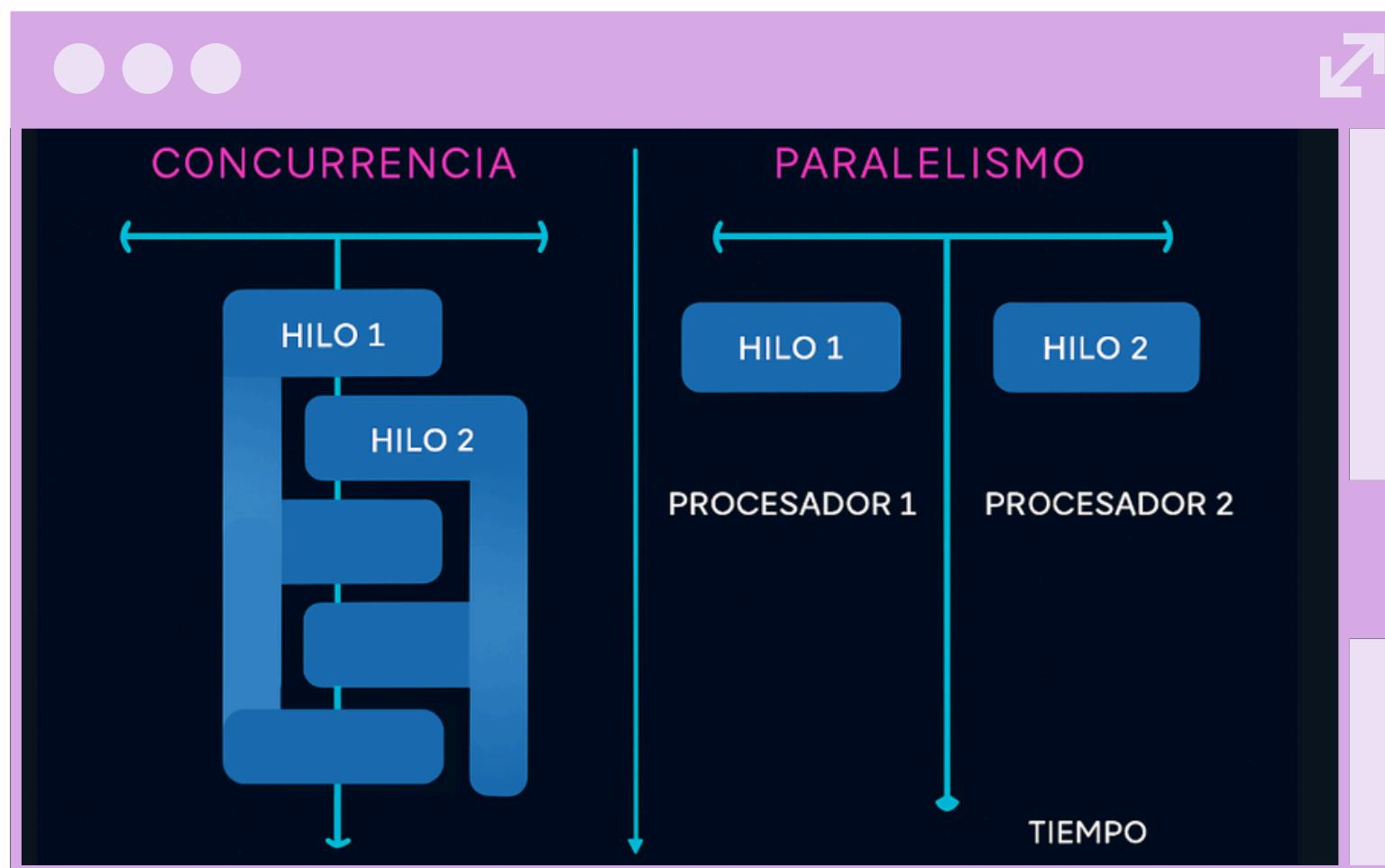
Los procesos e hilos son fundamentales para la gestión de recursos y la ejecución de tareas en sistemas operativos.

La concurrencia es la ejecución de múltiples tareas o procesos que pueden o no ejecutarse realmente en paralelo.

El multiprocesamiento simétrico es una arquitectura de sistema que permite la ejecución de múltiples procesos o hilos en múltiples procesadores.



CONCURRENCIA



La concurrencia es una **propiedad** del sistema operativo, que permite que varios procesos estén activos dentro del mismo periodo de tiempo, y el orden en el que suceden no está definido ni garantizado.

En la concurrencia no se puede saber con certeza cuál tarea se ejecutará antes o después, lo que sí sabemos es que no ocurre al mismo tiempo, pues alterna entre los procesos.

PARTES QUE COMPONEN A LA CONCURRENCIA

1. Procesos e hilos de ejecución.
2. Planificación (Scheduling).
3. Cambio de contexto (Context Switch).
4. Sincronización.
5. Comunicación entre procesos (IPC – Inter-Process Communication).
6. Gestión de recursos compartidos.

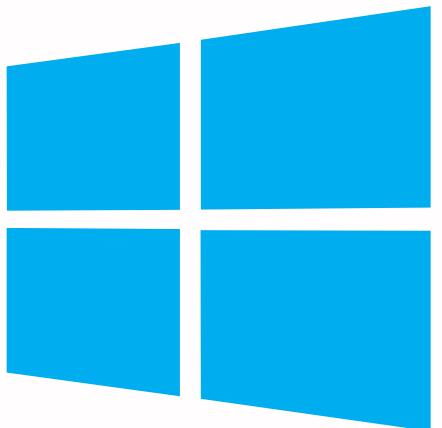


PROCESOS

En la actualidad existen una infinidad de sistemas operativos, cada uno diseñado para atender diversas necesidades. Sin embargo, a pesar de las diferencias, todos comparten un elemento fundamental: **los procesos**.

El proceso es un concepto clave en todo sistema operativo, es la unidad de trabajo de una computadora.

ubuntu®

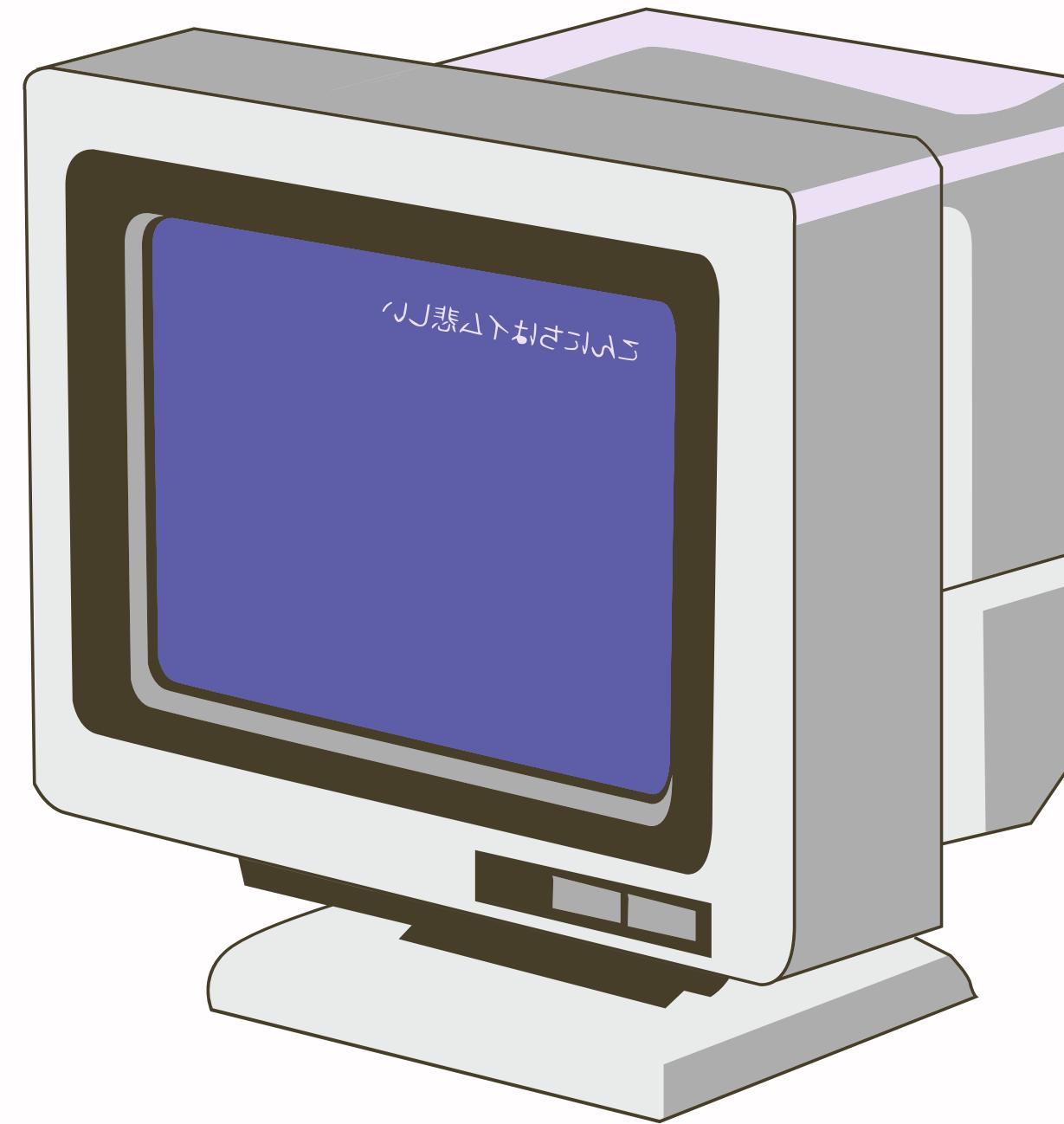




Las computadoras en sus orígenes contaban con capacidades muy limitadas: solo podían ejecutar un programa a la vez, sin posibilidad de realizar múltiples tareas simultáneamente.

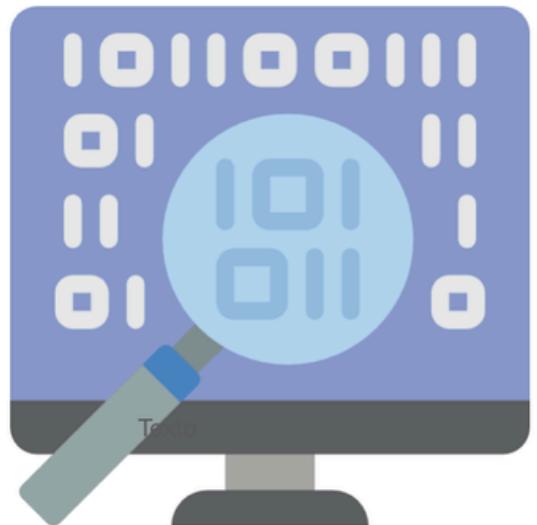
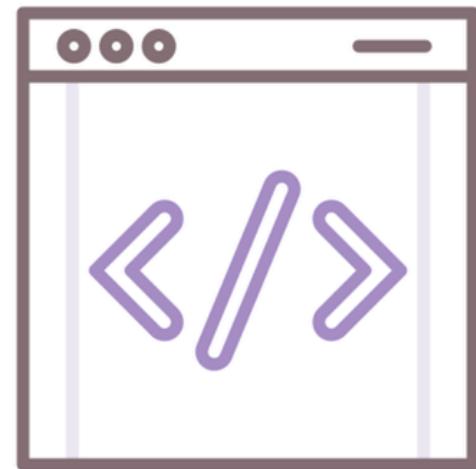
Con el paso del tiempo y el avance tecnológico, surgieron nuevas capacidades como la concurrencia, el paralelismo y el multiprocesamiento, que permitieron a los sistemas operar de manera más eficiente.

A raíz de esta evolución apareció el concepto de “proceso”



UN PROCESO ES LA EJECUCIÓN COMPLETA DE UN PROGRAMA

Los programas son simples conjuntos de instrucciones, no pueden ejecutarse por sí solos. El sistema operativo se encarga de darle vida a estas instrucciones transformándolas en procesos: asignándoles recursos, gestionando su estado y permitiendo su interacción con el sistema.



UNA BREVE ANALOGÍA...

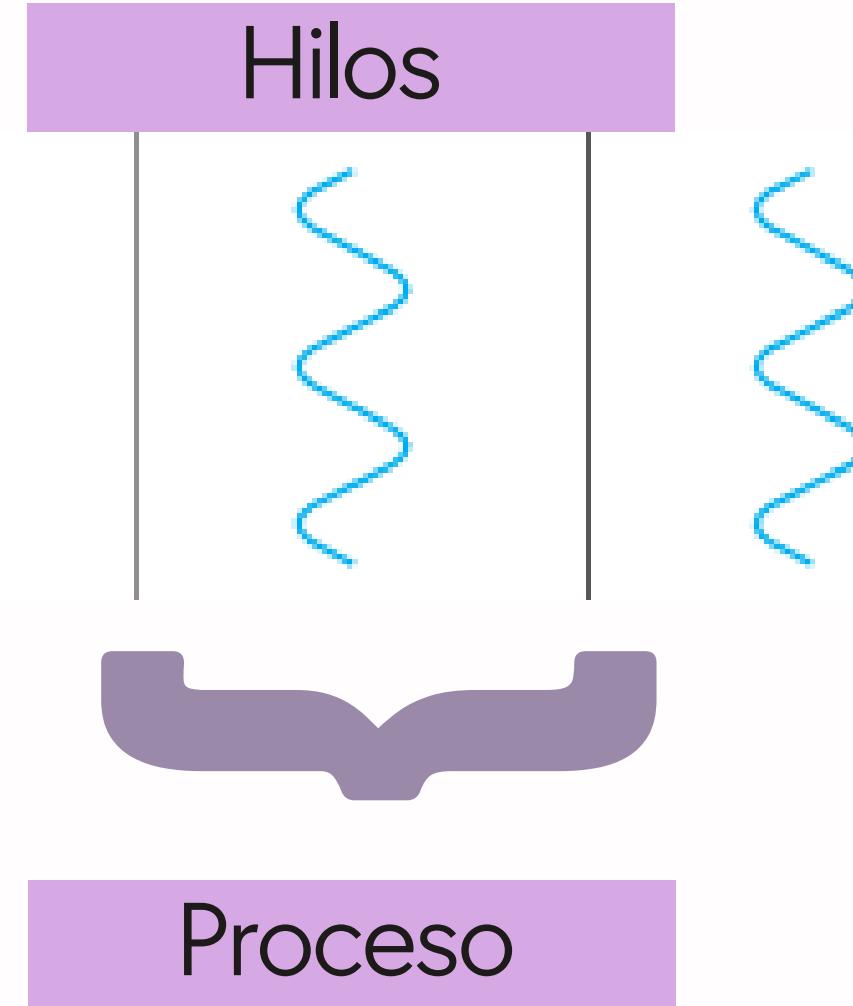
Las recetas de cocina son un conjunto de instrucciones a seguir para obtener determinado platillo. Por si solas no hacen nada, no se enciende la estufa ni es posible que la comida aparezca mágicamente.

Es el chef quien transforma las instrucciones en una acción. El acto de cocinar es equivalente a un proceso. Cocinar es la ejecución completa de la receta - El proceso es la ejecución completa de un programa.



HILOS

Los hilos son la **unidad básica de ejecución dentro de un proceso, cada hilo ejecuta una parte del proceso.** Tradicionalmente un proceso contenía solamente un único hilo, pero en sistemas modernos los procesos tienen la capacidad de contener más de un hilo, lo que permite realizar varias tareas concurrentemente.



PARA COMPRENDER MEJOR...

Los procesos de un hilo solo pueden hacer una tarea a la vez, los procesos de múltiples hilos pueden ejecutar varias tareas concurrentemente.

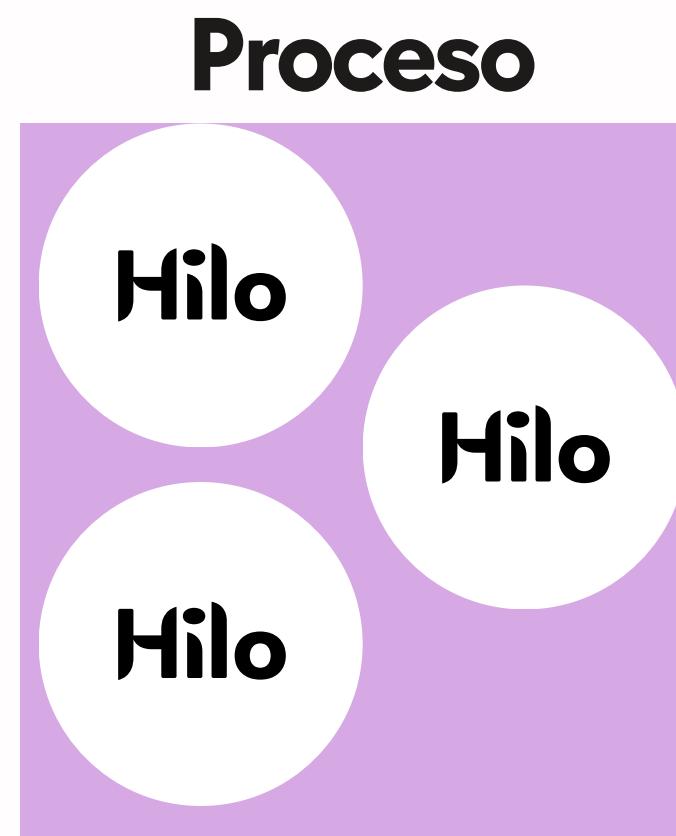
EJEMPLO: USAR YOUTUBE

Proceso: Youtube ejecutándose como aplicación

Hilos

- Reproducción de video
- Reproducción de audio
- Comentarios y likes

Cada hilo se encarga de una función específica sin bloquear a los demás.



PROCESOS E HILOS

ASPECTO	PROCESOS	HILOS
Unidad de ejecución	Programa completo en ejecución	Subunidad dentro de un proceso
Cambio de contexto	Se realiza el intercambio entre procesos	Se realiza el intercambio entre los hilos del proceso
Independencia	Tiene su propia memoria y recursos	Comparte memoria y recursos con otros hilos del mismo proceso
Ejemplo	Abrir dos navegadores distintos	Abrir una pestaña en Chrome que reproduce video mientras carga comentarios

ESTADOS Y TRANSICIONES

En informática, una «transición de estado» se refiere a una relación dirigida entre dos estados, que define un desencadenante, una condición y un comportamiento que provocan un cambio de estado dentro de una máquina de estados. Especifica cómo un sistema pasa de un estado a otro según ciertos criterios.

UNINTERRUPTIBLE SLEEP

Generalmente este proceso se encuentra esperando una operación de entrada/salida con algun dispositivo

Ejemplo: metes ropa a la lavadora y estas esperando a que termine de lavar, no puedes sacar la ropa a mitad del proceso porque esta bloqueada la puerta , el lavado (proceso) no puede interrumpirse hasta que termine la operacion de E/S

R U N N I N G

Corriendo, el proceso esta corriendo en el procesador

Ejemplo: Estás escribiendo en tu laptop. En ese momento tus manos están en acción sobre el teclado: el proceso se está ejecutando activamente. Todo tu enfoque y energía están corriendo, igual que un proceso que está en la CPU en ese instante.

INTERRUPTIBLE SLEEP

Espera interruptible, el proceso se encuentra esperando a que se cumpla algún evento

Ejemplo: Estás viendo tu celular esperando un mensaje de WhatsApp. Puedes quedarte "dormido" en la espera, pero si vibra el celular (evento externo), reaccionas de inmediato y sigues la acción. Así funciona un proceso en espera interruptible

STOPPED

Detenido, un proceso que ha sido detenido mediante el envío de alguna señal generalmente

Ejemplo: Estas jugando un videojuego en tu consola, pero tienes que ponerle pausa, así que presionas el botón de pausa, el juego (proceso) sigue estando ahí, la pantalla se queda congelada mas no avanza.

ZOMBIE

Proceso terminado, pero cuyo padre aún sigue «vivo» y no ha capturado el estado de terminación del proceso hijo, y por consiguiente, no lo ha eliminado de la tabla de procesos del sistema

Ejemplo: Cuando alguien termino de comer, pero sigue sentado en la mesa porque su amigo aun no recoge el plato, la persona ya no esta comiendo (proceso terminado), pero su rastro (el plato) sigue ocupando espacio en la mesa.

EJEMPLO GRAFICO

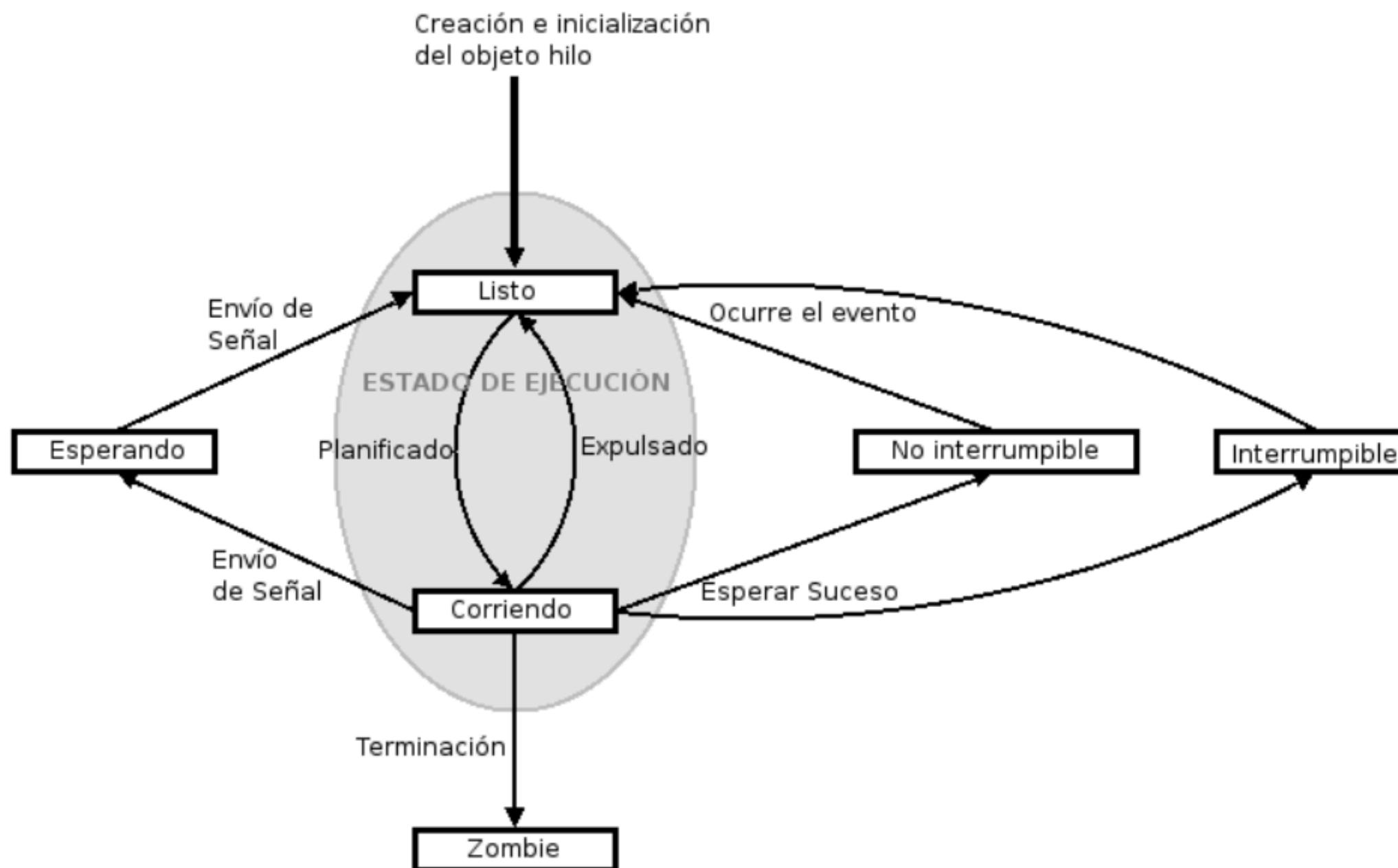


Figura 7.2: Estados de un procesos en Linux

TIPOS DE CONCURRENCIA

No existe un solo tipo de concurrencia, ya que existen distintos factores que las diferencian como: el número de procesadores disponibles, la ubicación física de los procesos, la estructura del software y la organización del mismo sistema operativo.

TIPOS DE CONCURRENCIA

Multiprogramación

- Varios procesos en un sistema monoprocesador (una sola CPU).
- No hay paralelismo real, sino pseudoparalelismo: los procesos se turnan en ráfagas de tiempo.
- Incluye el tiempo compartido, donde varios usuarios interactúan a la vez.





TIPOS DE CONCURRENCIA

Multiprocesamiento (Paralelismo real)

- Varios procesos ejecutándose en un sistema multiprocesador (varias CPUs).
- Aquí sí puede haber ejecución simultánea real.
- El planificador debe decidir qué proceso corre en qué CPU.



TIPOS DE CONCURRENCIA

Procesamiento distribuido

- Procesos ejecutándose en varias computadoras conectadas en red.
- Se comunican por paso de mensajes.
- Puede darse en clusters (red local rápida) o en grids (computadoras distribuidas geográficamente).

Aplicaciones estructuradas

- Programas que se diseñan como un conjunto de procesos o hilos concurrentes.
- Ejemplo: servidores web que atienden múltiples clientes al mismo tiempo.

TIPOS DE CONCURRENCIA





TIPOS DE CONCURRENCIA

Estructura del sistema operativo

- Algunos sistemas operativos están diseñados internamente como procesos concurrentes.
- Esto mejora la modularidad y eficiencia del propio sistema operativo.



MODELOS DE PROGRAMACIÓN CONCURRENTE



Los modelos de programación concurrente son las distintas maneras de organizar cómo varios procesos o hilos trabajan juntos al mismo tiempo dentro de un sistema operativo.

Los modelos declaran cómo los procesos comparten recursos, cómo se comunican y cómo evitan "estorbarse" unos a otros. A continuación, mostramos algunos de estos modelos junto con sus características.

MULTIPROGRAMACIÓN

Varios procesos comparten el mismo procesador, turnándose en intervalos muy pequeños.

ESTRUCTURA DEL SO

El propio Sistema Operativo está diseñado como un conjunto de procesos concurrentes.

MULTIPROCESAMIENTO

Varios procesos se ejecutan realmente al mismo tiempo, pero en diferentes procesadores.

Por
Contexto

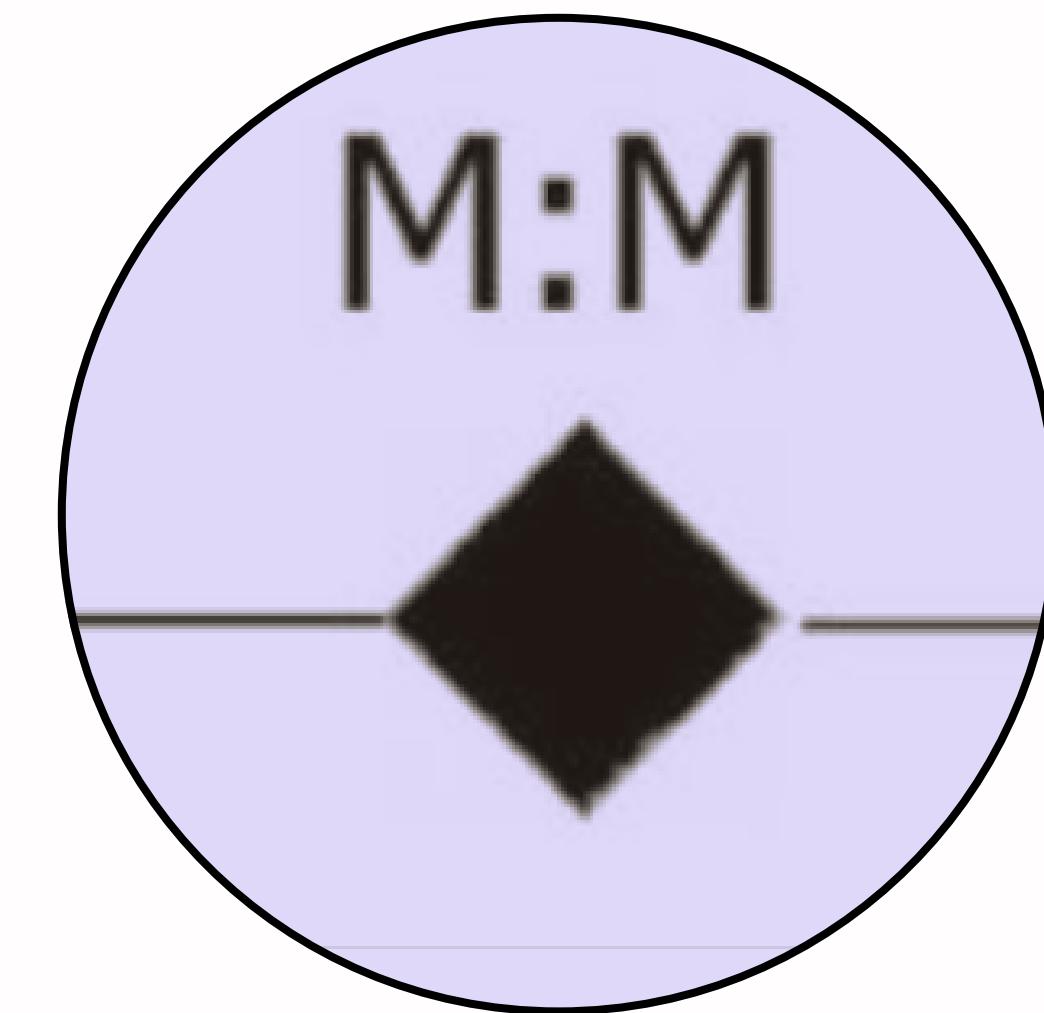
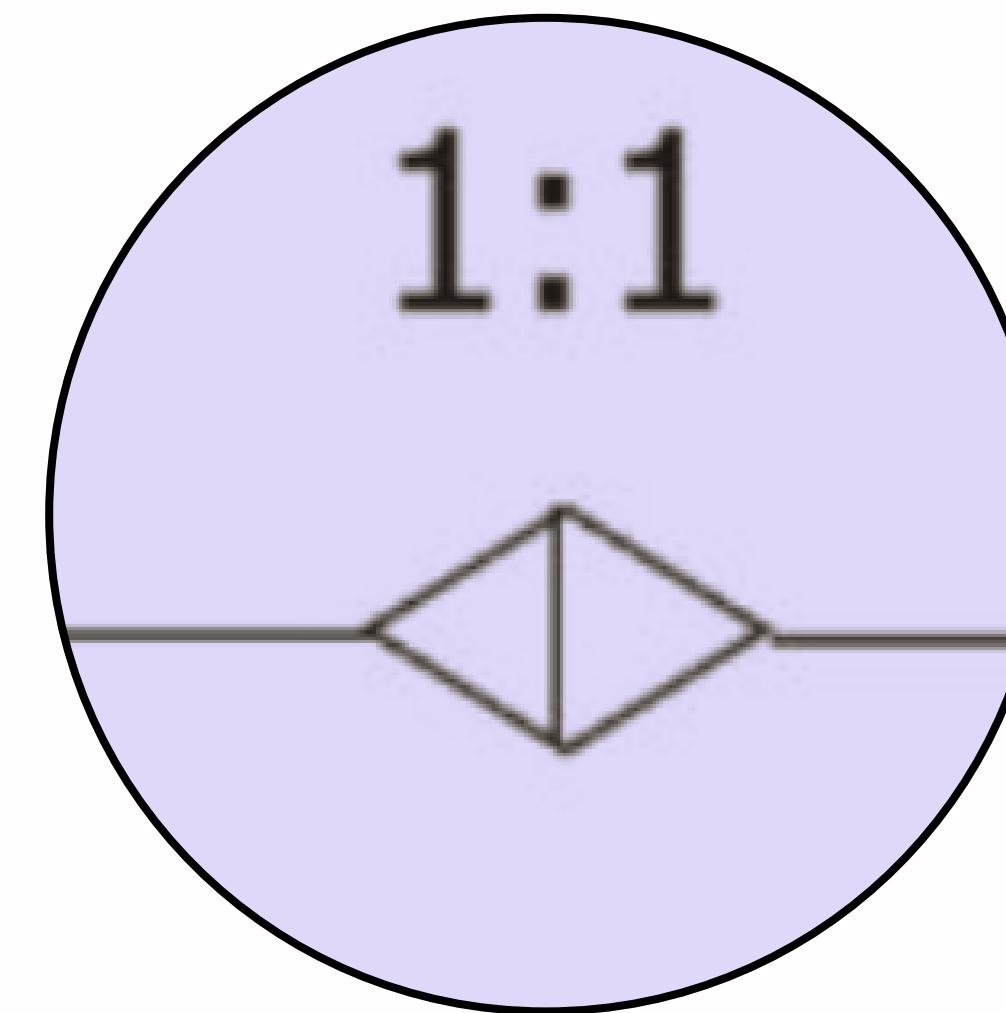
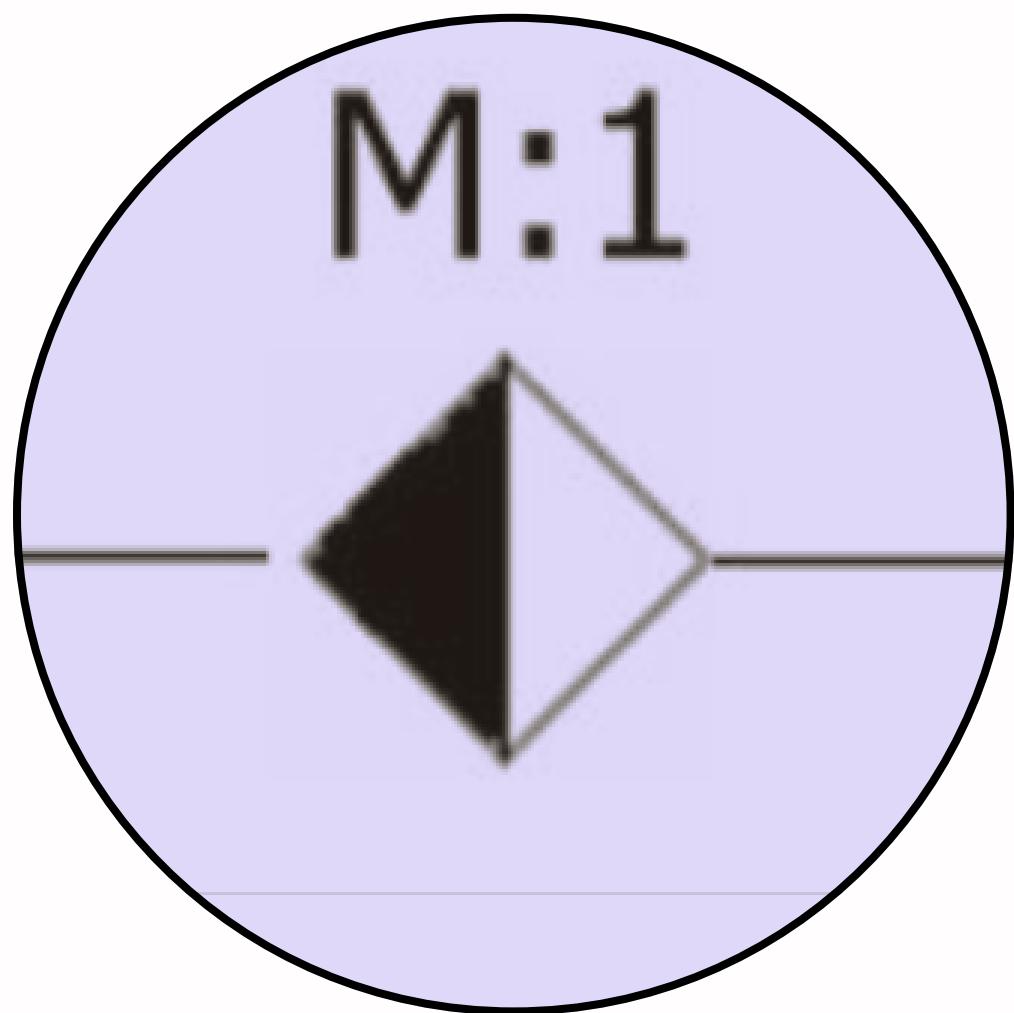
APLICACIONES ESTRUCTURADAS

Una aplicación está dividida en varios procesos o hilos que trabajan de forma concurrente.

DISTRIBUIDO

Procesos que se ejecutan en varias computadoras conectadas por una red.

Modelos basados en unidades de concurrencia

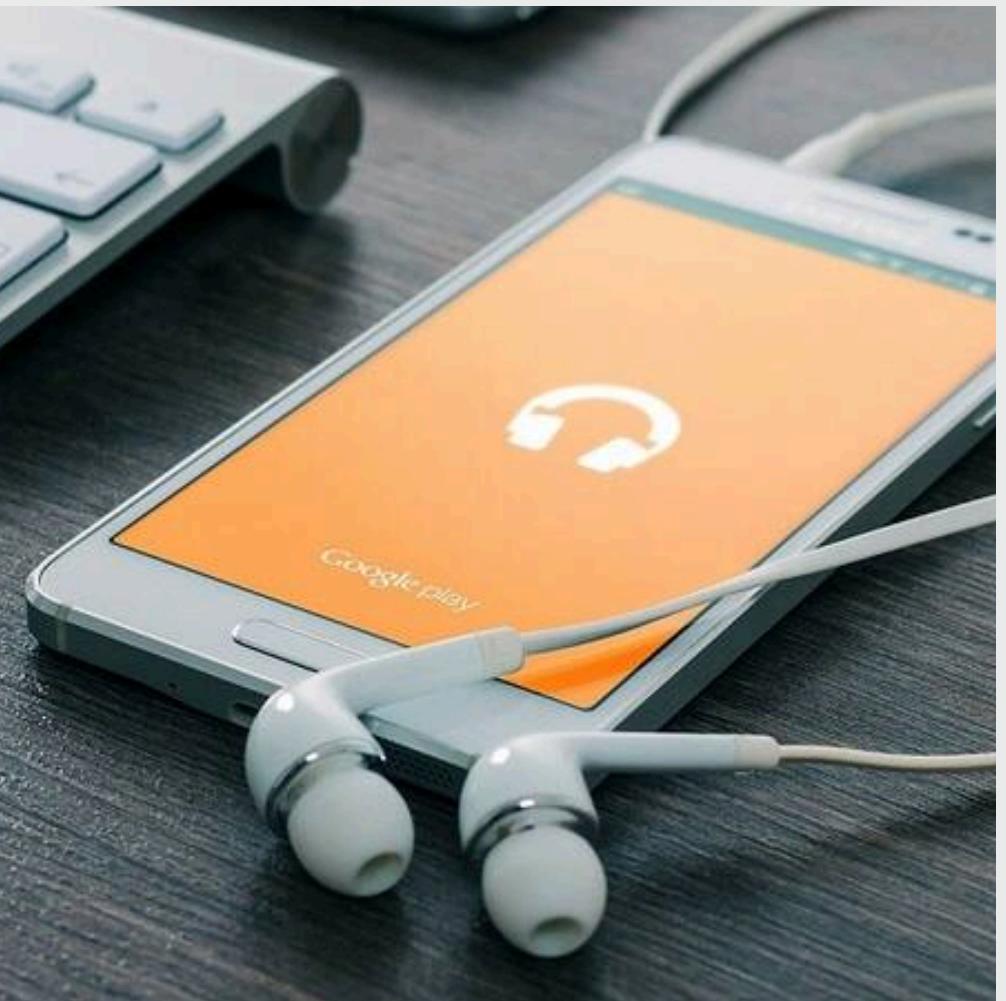


Modelos de patrones de organización de tareas con hilos

Es el cómo se organiza un grupo de hilos para realizar un proceso.

1. **Jefe/Trabajador:** Un hilo principal (el "jefe") se encarga de recibir las tareas y repartirlas entre varios hilos trabajadores.
2. **Equipo de Trabajo (Work Crew):** Todos los hilos son iguales y toman tareas de una cola compartida.
3. **Línea de Ensamblado (Assembly Line):** Cada hilo se especializa en una etapa distinta y va pasando el trabajo al siguiente.





Modelos según paradigma de ejecución

- **Algorítmicos**: El programa sigue un plan fijo, paso a paso, desde el inicio hasta el final. Hacen cálculos, procesan datos o ejecutan tareas sin esperar nada externo.
- **Controlados por eventos**: El programa no avanza solo, sino que espera a que algo ocurra (un evento) y reacciona a eso. Estos programas son los que manejan la interacción con el usuario o el entorno.

VENTAJAS DE LA EJECUCIÓN

- Eficiencia en el Uso de Recursos
- Maximización de la utilización de la CPU y dispositivos de E/S.
- Balanceo de carga automático
- Ahorro de memoria y energía con virtualización
- Reducción del tráfico de bus y mejora del rendimiento de caché
- Mejora del Rendimiento General de las Aplicaciones y del Sistema
- Agilización del trabajo
- Traslape de actividades
- Creación y destrucción de hilos más rápida
- Planificación rápida de hilos a nivel de usuario
- Mayor flexibilidad con hilos de kernel
- Baja latencia con hilos emergentes
- Aceleración de la ejecución en aplicaciones paralelas
- Simplificación del Desarrollo y la Estructuración de Programas
- Modelo de Programación simplificado
- Compartición de espacio de direcciones
- Organización y estructuración de programas
- Abstracción de hardware
- Planificación personalizada

VENTAJAS DE LA EJECUCIÓN

- Comunicación eficiente entre procesos
- Mayor portabilidad del código
- Facilita la comprensión de rutinas
- Fiabilidad y Seguridad del Sistema
- Respuesta a fallos
- Aislamiento en sistemas multiusuario/multitarea
- Prevención de congelamientos
- Detección y recuperación de interbloqueos
- Simplificación de la separación de recursos con virtualización
- Flexibilidad y Adaptabilidad
- Asignación flexible de memoria.
- Entornos personalizados con máquinas virtuales
- Facil migración de máquinas virtuales
- RPC asíncronas para mayor paralelismo
- Middleware basado en coordinación



TIPOS DE PROCESOS CONCURRENTES

Procesos Independientes

- No comparten memoria ni dispositivos.
- Su ejecución es determinista (no se ven afectados por otros procesos).

Ventajas

- Más simples de poderse programar.
- No requieren mecanismos de sincronización.

Ejemplos

- Editor de texto abierto junto con un reproductor de música.
- Juegos ejecutados en una PC mientras corre un servicio en segundo plano.

Procesos Cooperativos

- Comparten datos, recursos o comunicación.
- Su resultado depende del orden de ejecución y sincronización.
- Necesitan mecanismos de coordinación para evitar errores.

Ejemplos

- Productor-consumidor: Un proceso produce datos y otro lo procesa.
- Cliente-servidor: El servidor responde a múltiples solicitudes concurrentes de clientes.

Problemas más comunes

- Condiciones de carrera
- Bloqueos mutuos.

TIPOS DE INTERACCIONES DENTRO DE LA CONCURRENCIA

Competencia por Recursos (Exclusión mutua)

1. Cuando dos o más procesos quieren usar un recurso a la vez.
2. Recursos críticos

Problemas:

- Condiciones de carrera: El resultado depende del orden de ejecución.
- Inconsistencias en datos.

Soluciones

- Semáforos
- Mutex
- Monitores
- Spinlocks

Cooperación (Comunicación entre Procesos, IPC)

- Los procesos colaboran para resolver una tarea más compleja.

Formas de comunicación:

1. Memoria compartida:
 - Espacio común accesible por varios procesos
 - Rápida, pero requiere sincronización para evitar inconsistencias.
2. Paso de mensajes:
 - Comunicación explícita enviando y recibiendo mensajes.
 - Más segura, aunque con mayor sobrecarga.

TIPOS DE INTERACCIONES DENTRO DE LA CONCURRENCIA

Sincronización

- Asegura que los procesos sigan un orden correcto de ejecución.
- Implica que un proceso debe esperar a otro antes de poder continuar.

Técnicas de sincronización

1. Semáforos binarios
2. Variables de condición
3. Barreras

Bloqueo Mutuo (Deadlock)

- Situación en la que varios procesos esperan recursos retenidos por otros.
- Condiciones para que pueda ocurrir:
 1. Exclusión mutua
 2. Retención y espera
 3. No apropiación
 4. Espera circular

Estrategias de manejo:

- **Prevención:** Eliminar alguna condición.
- **Evasión:** Evitar estados inseguros.
- **Detección y recuperación:** Permitirlo y luego solucionarlo.

El sistema operativo (SO) gestiona esta concurrencia con tres grandes funciones:

1. ESTRUCTURAR Y RASTREAR LA EJECUCIÓN

- Procesos y hilos tienen estados (*nuevo, listo, en ejecución, bloqueado, terminado*).
- El SO mantiene un *PCB* (Process Control Block) por proceso, con registros, contador de programa, estado, prioridad, etc.
- Un cambio de contexto guarda/restaura el estado de CPU para pasar de un proceso a otro.
-

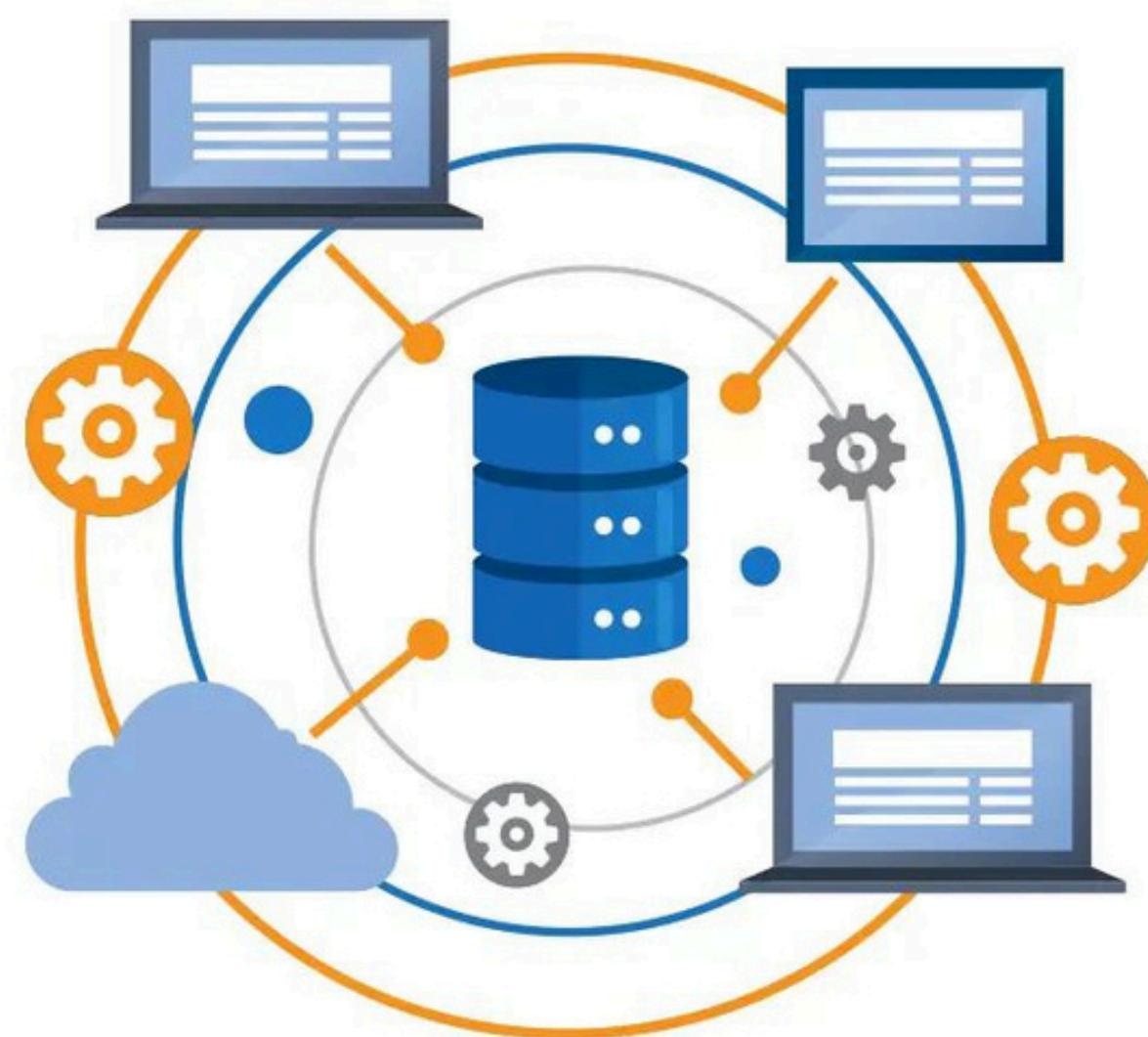
2. SINCRONIZAR Y PROTEGER EL ACCESO A RECURSOS COMPARTIDOS

- Problemas típicos: condiciones de carrera, secciones críticas, interbloqueos (*deadlocks*) y inanición (*starvation*).
- Mecanismos: bloqueos (*locks*), semáforos, monitores y variables de condición; además de regiones críticas bien definidas y protocolos de ordenación (p. ej., prioridad por herencia para mitigar inversión de prioridades)
-

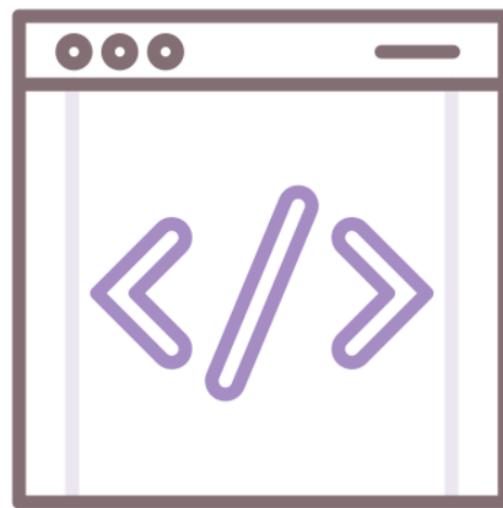
3. PLANIFICAR QUIÉN CORRE Y CUÁNDΟ

- El planificador decide la asignación del CPU a procesos/hilos, buscando un equilibrio entre rendimiento (*throughput*), tiempos (espera

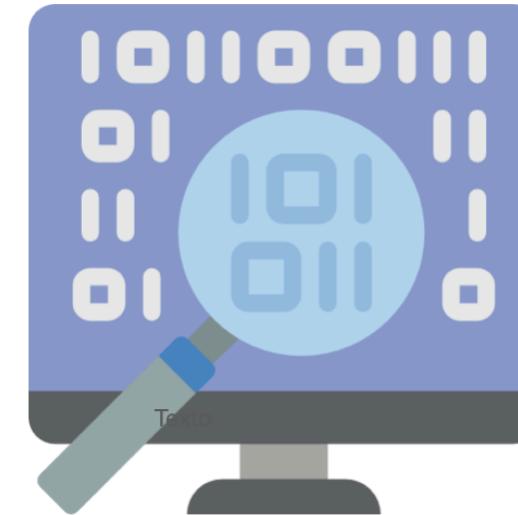
CONTROL Y GESTIÓN DE LA CONCURRENCIA



¿CÓMO ADMINISTRA PROCESOS EL SISTEMA OPERATIVO?



PROGRAMA

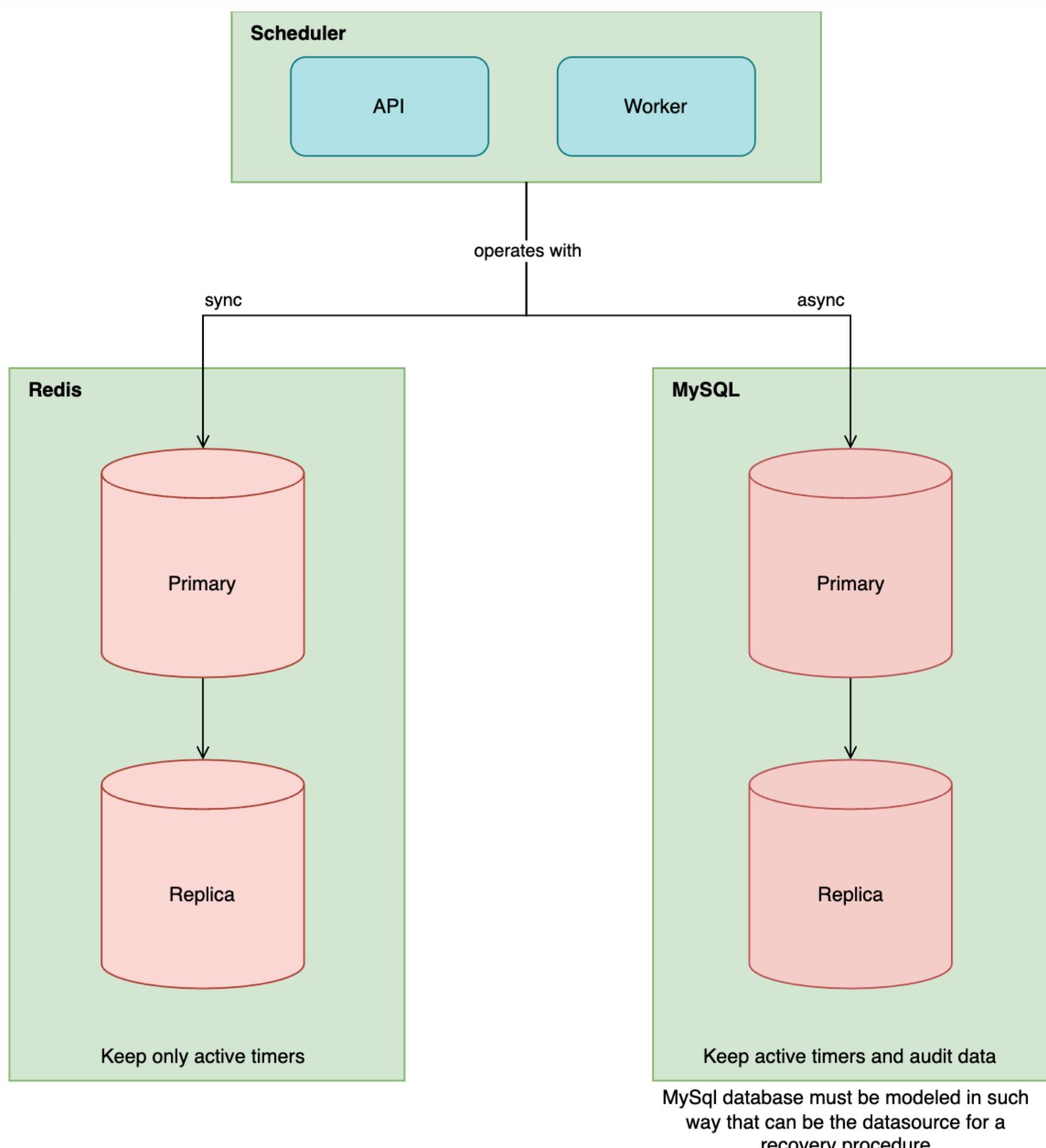


PROCESO

El sistema operativo (SO) administra los procesos mediante varias funciones clave:

- **Creación y terminación:** usa llamadas al sistema (ej. fork, exec en Unix o APIs en Windows) para iniciar o finalizar procesos.
- **Estados de proceso:** cada proceso pasa por estados (nuevo, listo, en ejecución, bloqueado, terminado).
- **Planificación (scheduling):** decide qué proceso se ejecuta en la CPU, usando algoritmos como FIFO, Round Robin, Prioridades, Multinivel.
- **Colas de procesos:** mantiene colas de listos (esperando CPU), bloqueados (esperando E/S o recurso) y ejecución.
- **Cambio de contexto:** guarda y restaura la información (registros, contador de programa, etc.) cuando cambia de un proceso a otro.
- **Sincronización y comunicación:** gestiona el acceso concurrente a recursos con semáforos, monitores, mutex, y permite comunicación entre procesos (IPC: tuberías, colas de mensajes, memoria compartida).
- **Protección y control:** asegura que los procesos no interfieran entre sí y que los recursos se asignen de manera justa y segura.

CONCEPTO DE PLANIFICADOR (SCHEDULER)



El planificador (scheduler) es el componente del sistema operativo encargado de decidir qué proceso se ejecutará y cuándo en la CPU o en otros recursos.

El planificador (scheduler) es un componente fundamental del sistema operativo, encargado de tomar decisiones sobre el orden y el momento en que los procesos acceden a los recursos del sistema, principalmente la CPU, pero también memoria, dispositivos de entrada/salida u otros recursos. En otras palabras, el scheduler organiza la ejecución de los procesos para que el sistema funcione de manera eficiente, justa y estable.

TIPOS PRINCIPALES

- Planificador a largo plazo: decide qué procesos entran al sistema (controla la carga de trabajo).
- Planificador a medio plazo: suspende o reactiva procesos para equilibrar la carga (swap).
- Planificador a corto plazo: el más importante, selecciona el siguiente proceso que usará la CPU (usa algoritmos como FIFO, Round Robin, prioridades).

OBJETIVOS DEL SCHEDULER

- Maximizar el uso del procesador.
- Reducir tiempos de espera.
- Garantizar equidad entre procesos.
- Dar prioridad según necesidades (interactividad, tiempo real, etc.).

CONCEPTO DE PLANIFICADOR (SCHEDULER)



PLANIFICACIÓN A CORTO PLAZO

La planificación a corto plazo selecciona el proceso que se ejecutará de inmediato en la CPU, buscando optimizar tiempos de respuesta y uso del procesador mediante algoritmos como FCFS, SJF/SRTF, prioridades, Round Robin y MLFQ.



PLANIFICACIÓN A CORTO, MEDIANO Y LARGO PLAZO

PLANIFICACIÓN A MEDIANO PLAZO

La planificación a mediano plazo controla la multiprogramación suspendiendo o reanudando procesos según la disponibilidad de memoria y la carga del sistema, evitando problemas como el thrashing y manteniendo estabilidad.

PLANIFICACIÓN A LARGO PLAZO

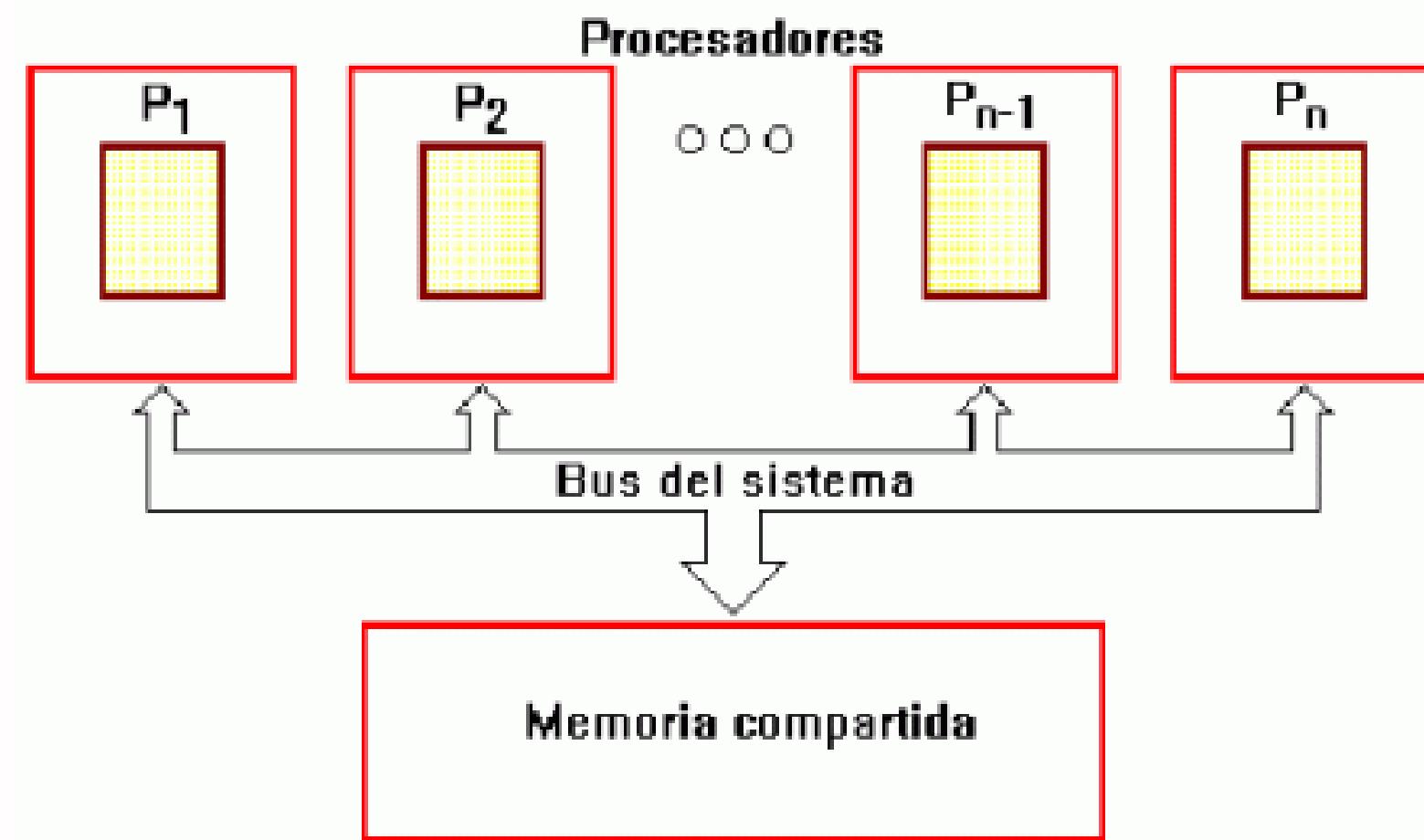
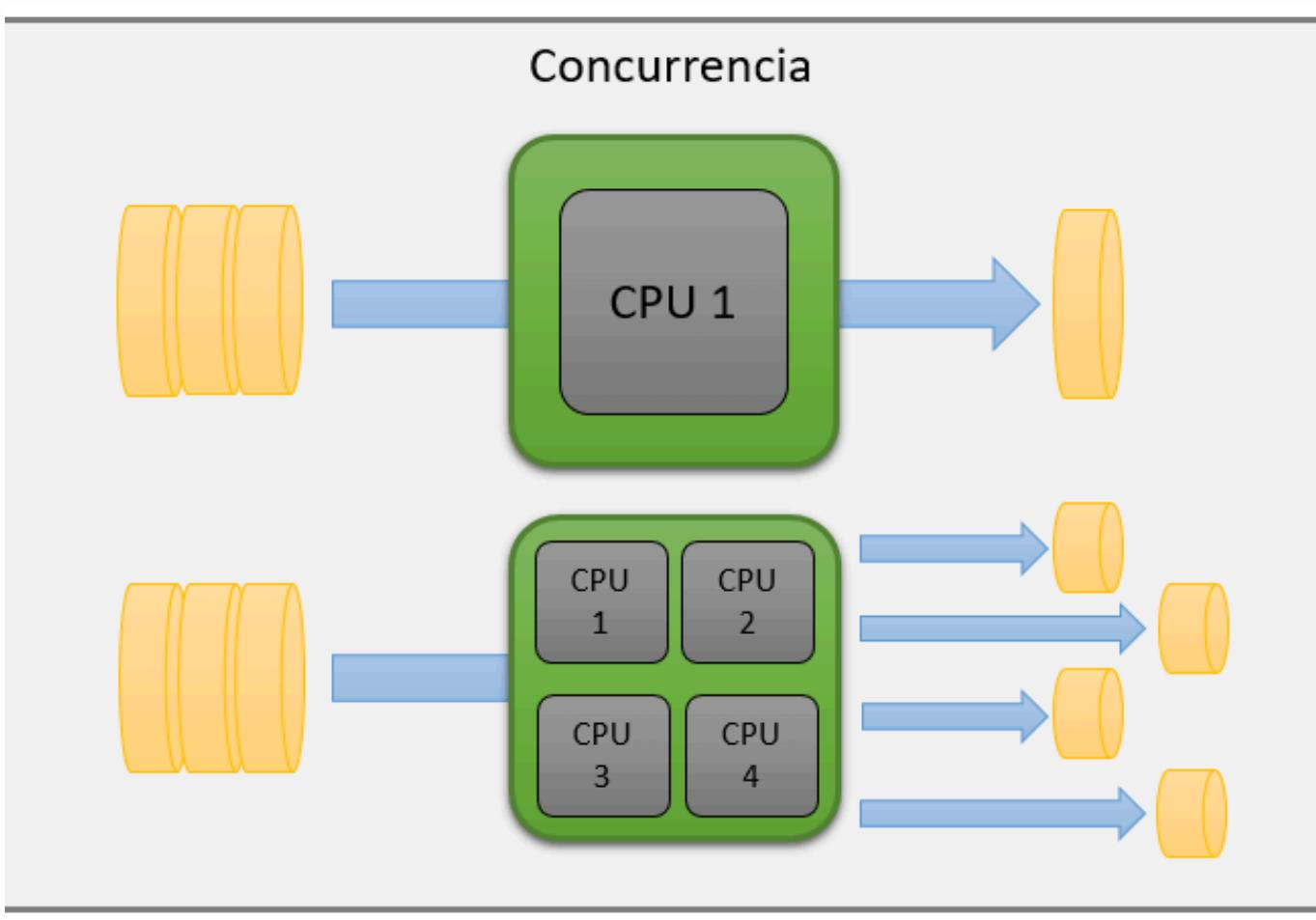
La planificación a largo plazo regula la entrada de trabajos al sistema, decidiendo qué procesos o usuarios pueden ingresar, con el fin de equilibrar el nivel global de multiprogramación y aplicar políticas de servicio como cuotas o prioridades.



MULTIPROCESAMIENTO SIMETRICO (SMP)

Si los procesadores comparten una memoria común, entonces cada procesador accede a los programas y datos almacenados en la memoria compartida, y los procesadores se comunican entre sí a través de dicha memoria

MULTIPROCESAMIENTO SIMETRICO



VENTAJAS DEL SMP CONTRA EL AMP

- Balance de carga más eficiente

En **SMP**, cualquier procesador puede ejecutar cualquier tarea, lo que permite repartir mejor el trabajo.

En **AMP**, el procesador maestro puede convertirse en cuello de botella porque debe coordinar a los demás.

- Mayor tolerancia a fallos

En **SMP**, si un procesador falla, el sistema puede seguir funcionando con los demás.

En **AMP**, si falla el procesador maestro, el sistema completo queda inutilizable.

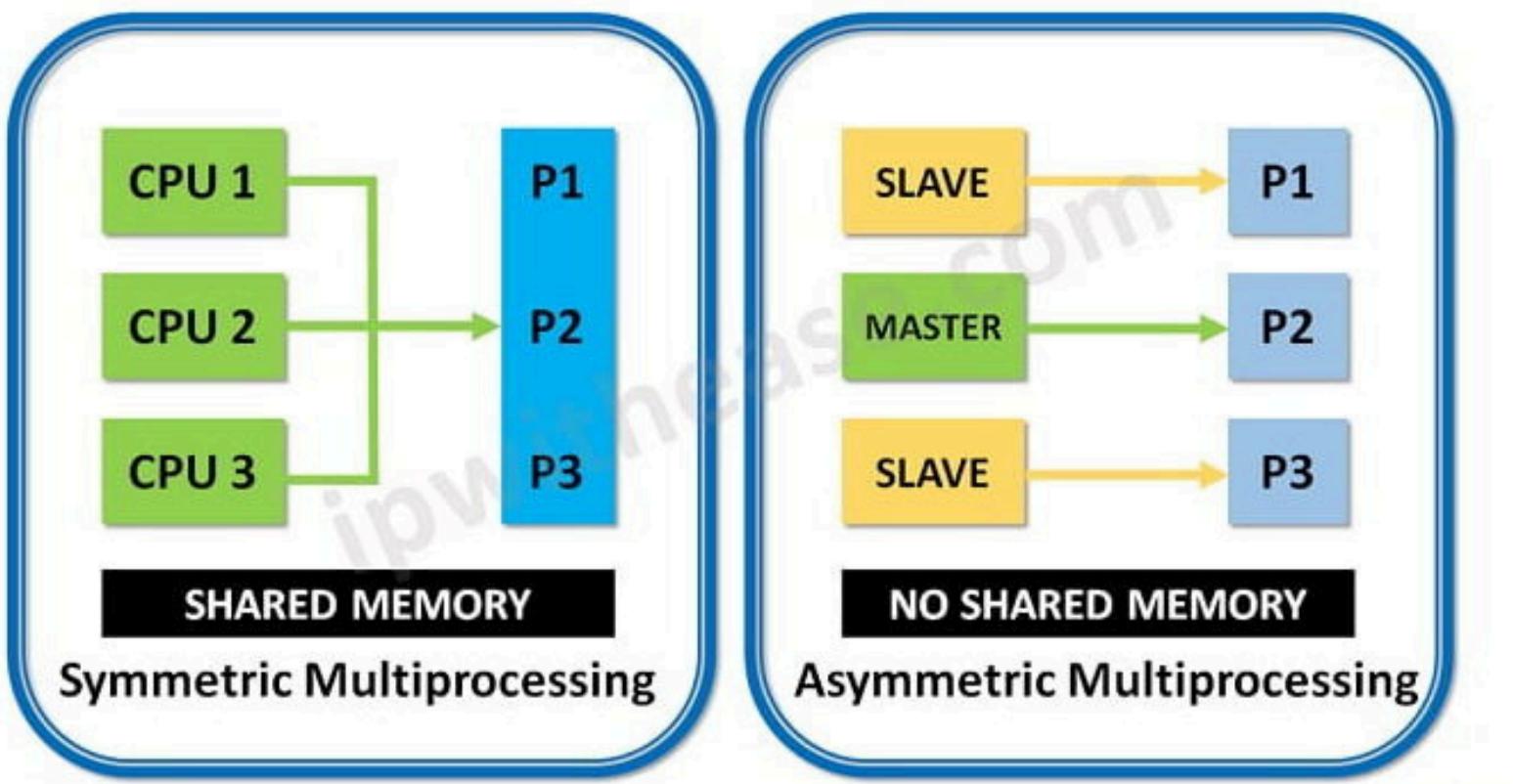
- Mejor aprovechamiento del hardware

En **SMP**, todos los procesadores se utilizan de forma flexible según la carga.

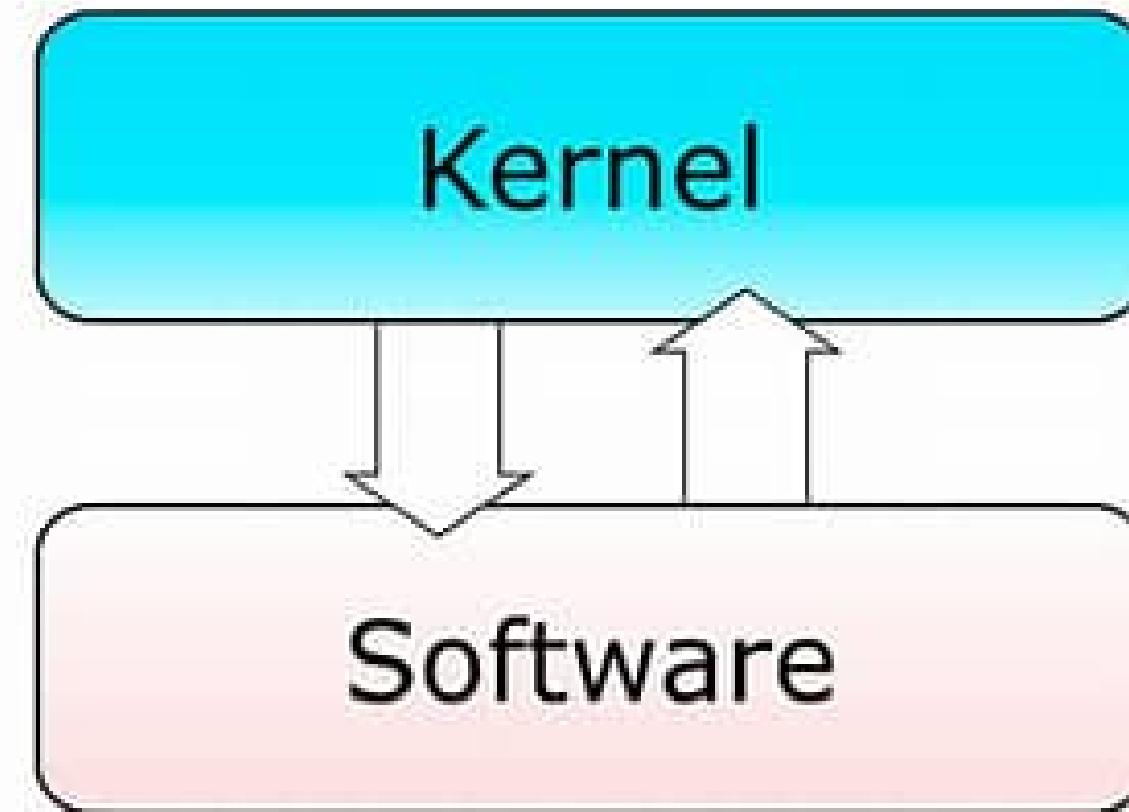
En **AMP**, algunos procesadores pueden quedar inactivos si el maestro falla.

s.

SYMMETRIC vs ASYMMETRIC MULTIPROCESSING



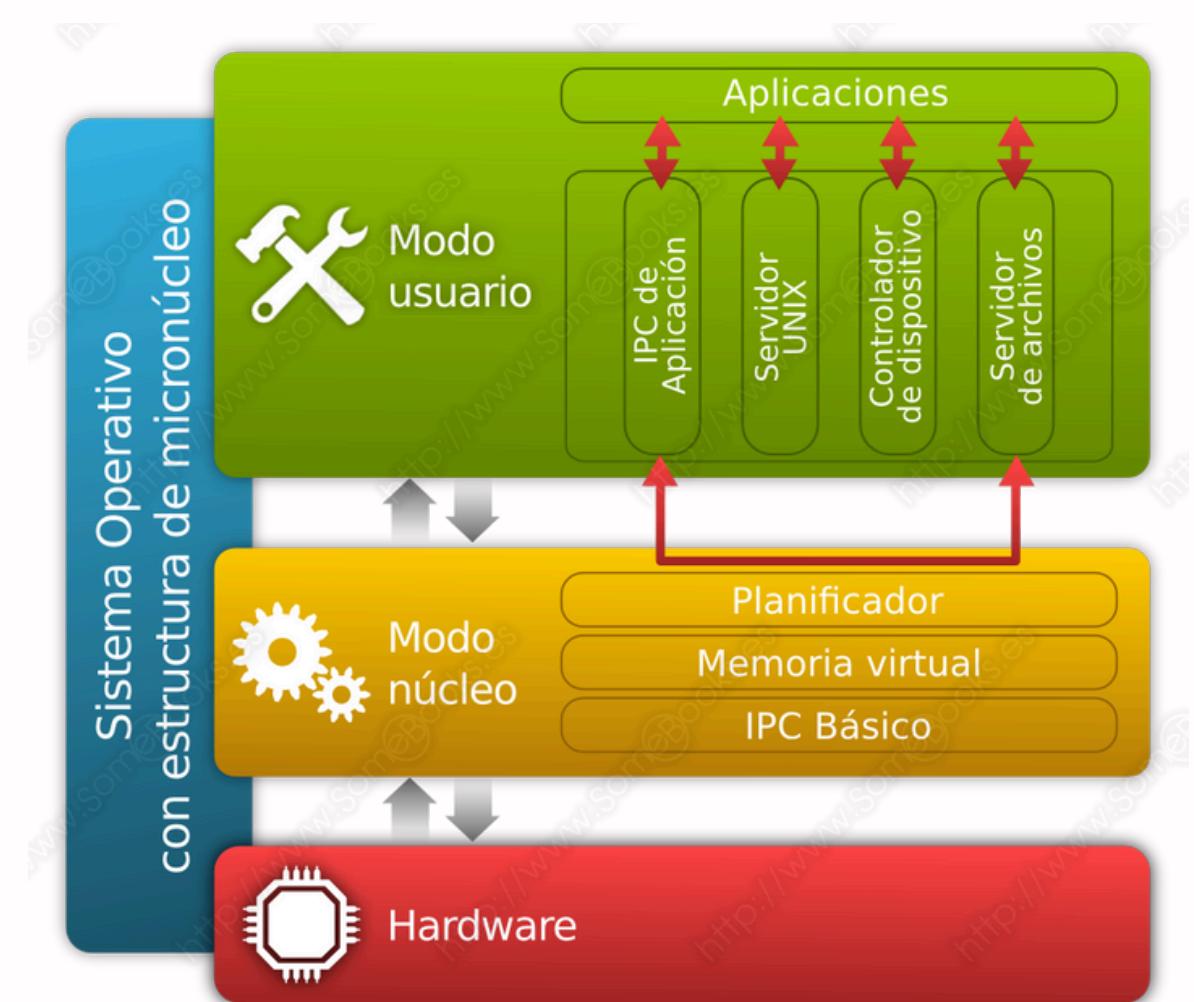
MICRO NÚCLEOS



¿QUÉ ES UN KERNEL?

El kernel o núcleo, es una parte fundamental del sistema operativo que se encarga de conceder el acceso al hardware de forma segura para todo el software que lo solicita, el kernel es una pequeña e invisible parte del sistema operativo, pero lo más importante, ya que sin esta no podría funcionar. Todos los sistemas operativos tienen un kernel, incluso Windows 10, pero quizás el más famoso es el kernel de Linux, que ahora además está integrado en Windows 10 con sus últimas actualizaciones.

MICRO NÚCLEOS



Microkernel es un software o código que contiene la cantidad mínima requerida de funciones, datos y características para implementar un sistema operativo. Proporciona una cantidad mínima de mecanismos, lo que es suficiente para ejecutar las funciones más básicas de un sistema operativo. Permite implementar otras partes del sistema operativo ya que no impone muchas políticas.

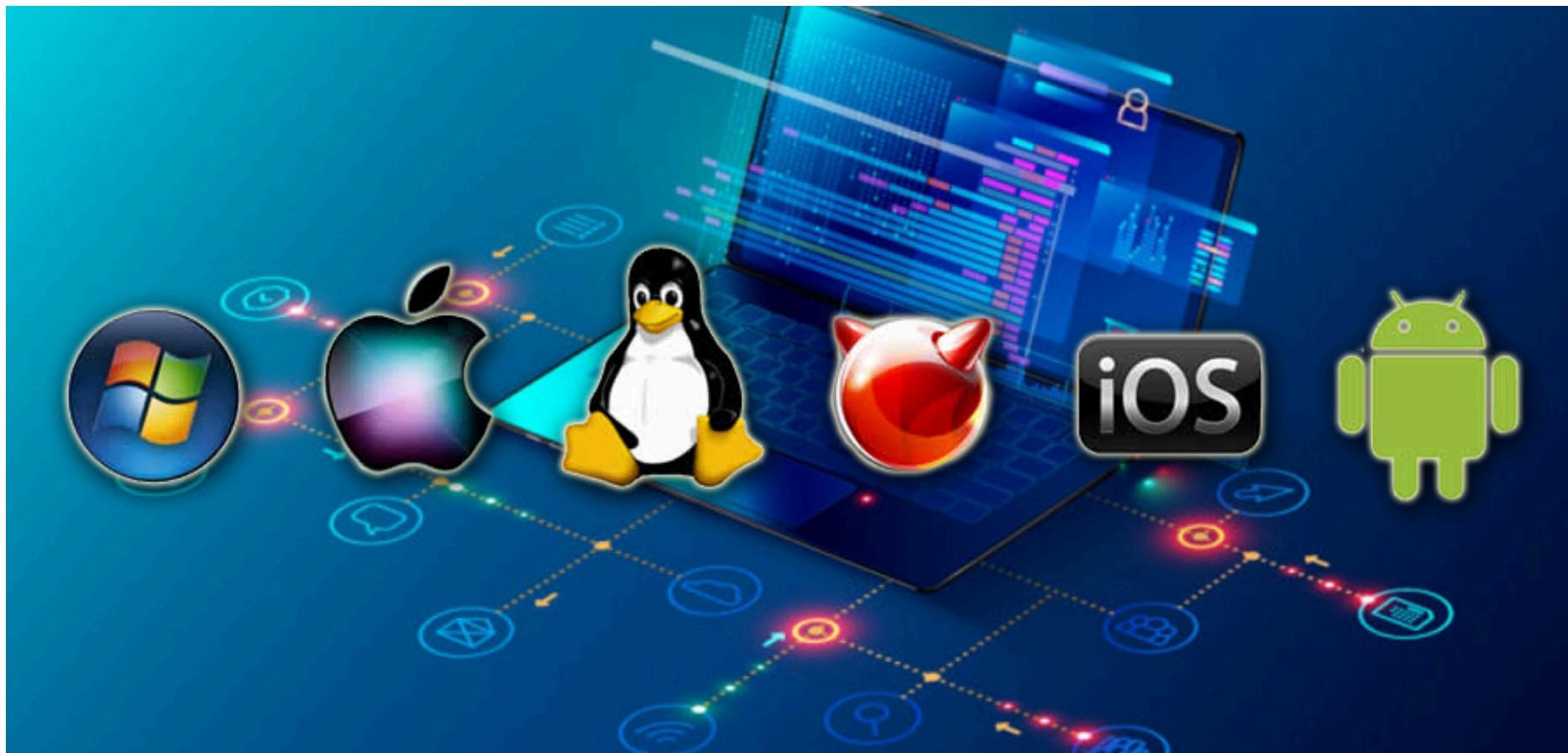
Este enfoque modular permite una gran flexibilidad, ya que los programas de espacio de usuario pueden ser fácilmente añadidos o eliminados según lo necesario.

DIFERENCIAS ENTRE KARTEL MONOLÍTICO Y MICROKERNEL



El kernel puede clasificarse en dos categorías, microkernel y kernel monolítico. Microkernel es aquel en el que los servicios de usuario y los servicios de kernel se mantienen en un espacio de direcciones separado. Sin embargo, en los servicios de usuario kernel monolíticos y en los servicios de kernel, ambos se mantienen en el mismo espacio de direcciones. El microkernel es más seguro que el kernel monolítico, ya que, si un servicio falla en el micro kernel, el sistema operativo no se ve afectado.

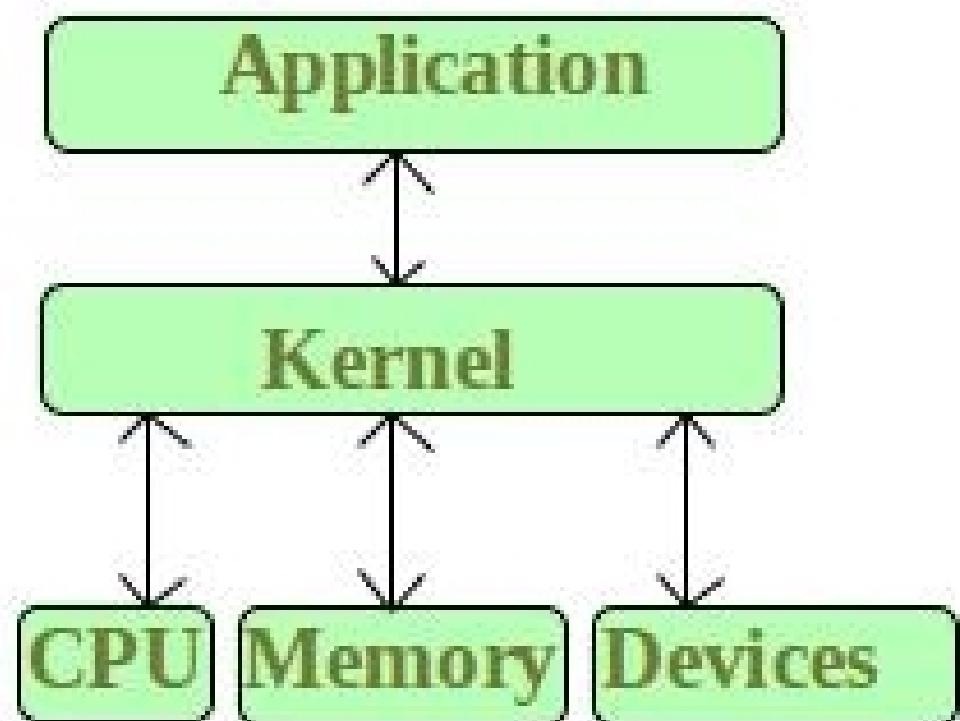
VENTAJAS Y DESVENTAJAS DEL MICROKERNEL



Ventajas

- **Estabilidad y seguridad:** Los fallos en servicios no afectan al núcleo, lo que mejora la estabilidad general del sistema.
- **Facilidad de mantenimiento:** Es más sencillo actualizar o reemplazar servicios sin recomilar todo el sistema.
- **Menor tamaño:** Al incluir solo lo esencial, el microkernel ocupa menos espacio en memoria.

VENTAJAS Y DESVENTAJAS DEL MICROKERNEL



Desventajas

- **Rendimiento inferior:** La comunicación entre el núcleo y los servicios puede ser más lenta.
- **Complejidad en la implementación:** La separación de servicios puede hacer que la implementación sea más complicada.
- **Menor compatibilidad:** Algunos microkernels pueden no ser compatibles con hardware más antiguo.

EJEMPLOS DE SISTEMAS OPERATIVOS CON MICROKERNEL



Algunos ejemplos de **sistemas operativos** son:

- **MINIX**: Un sistema operativo educativo diseñado para demostrar los principios de los sistemas operativos
- **QNX**: Un sistema operativo en tiempo real utilizado en sistemas embebidos.

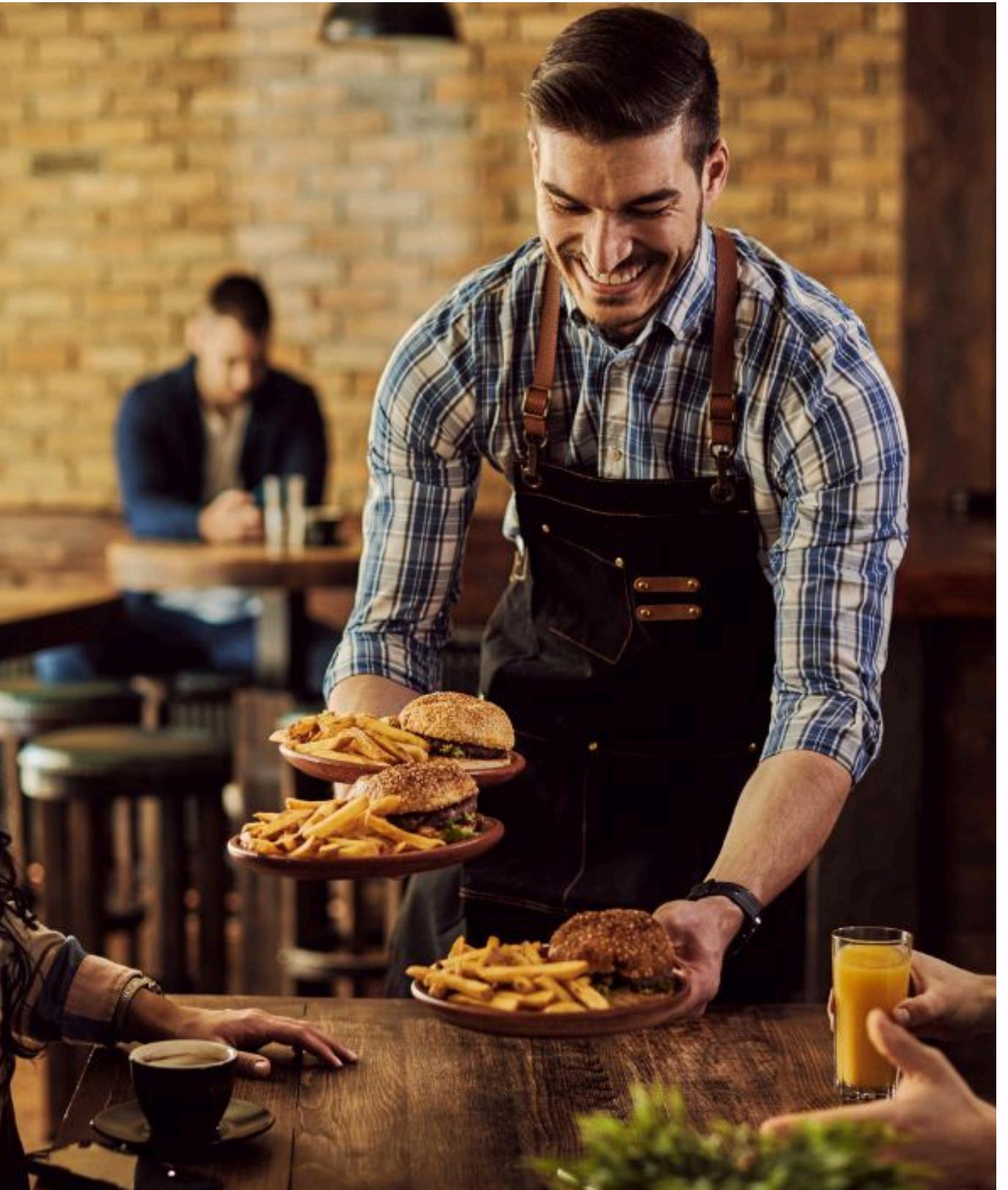
Sistemas híbridos

- Windows NT/Windows 2000/Windows 10: usan núcleo híbrido, mezcla de monolítico y microkernel.

GESTIÓN DE HILOS

La gestión de hilos se refiere al manejo y coordinación de los hilos, para que los procesos y programas realicen múltiples secuencias simultáneamente.

Imaginemos un restaurante (proceso) con una cocina, comidas y mesas (recursos compartidos). En este restaurante, los meseros (hilos) realizan tareas específicas como tomar pedidos, servir alimentos y cobrar. Sin embargo, necesitan un gerente de meseros (gestor de hilos) para coordinarse y evitar el caos. Este gerente decide cuántos meseros trabajarán según la demanda, prioriza mesas y resuelve problemas, como evitar que varios meseros tomen la misma orden o despedir a un mesero si ya no es necesario.



COMPOSICIÓN

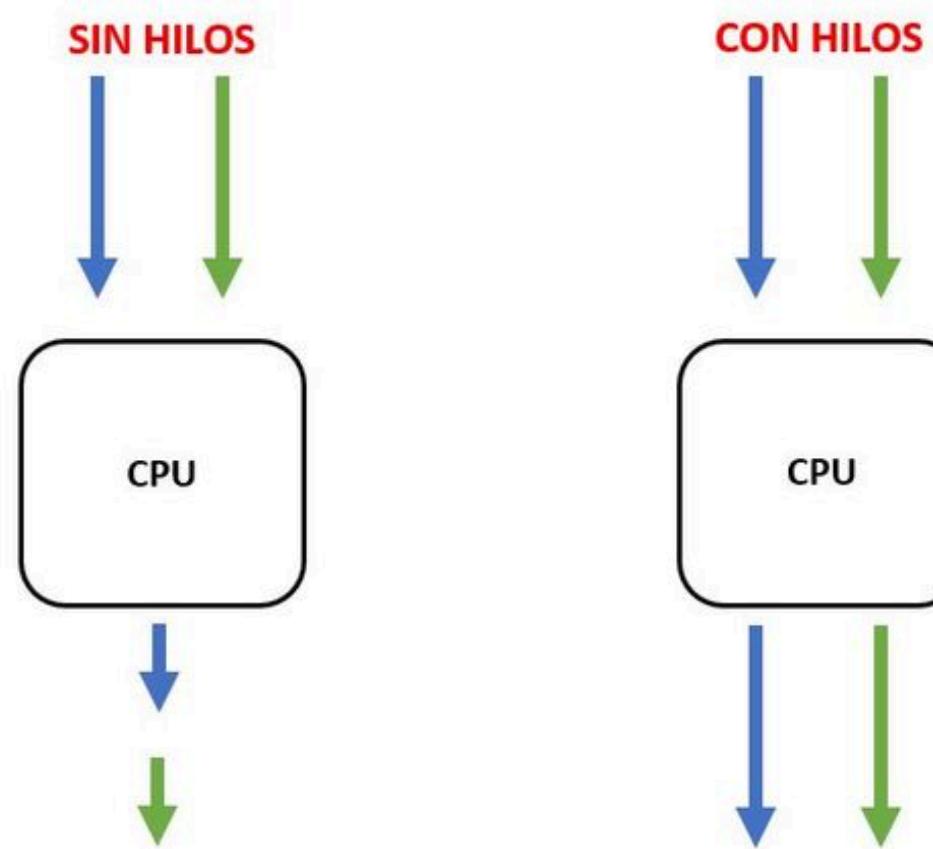


→ Los hilos se crean cuando un proceso o algun otro hilo lo instancie. Al finalizar la tarea que lleva el hilo, este se elimina, liberando recursos.

→ El sistema operativo decide que hilo se ejecutara y en que momento.

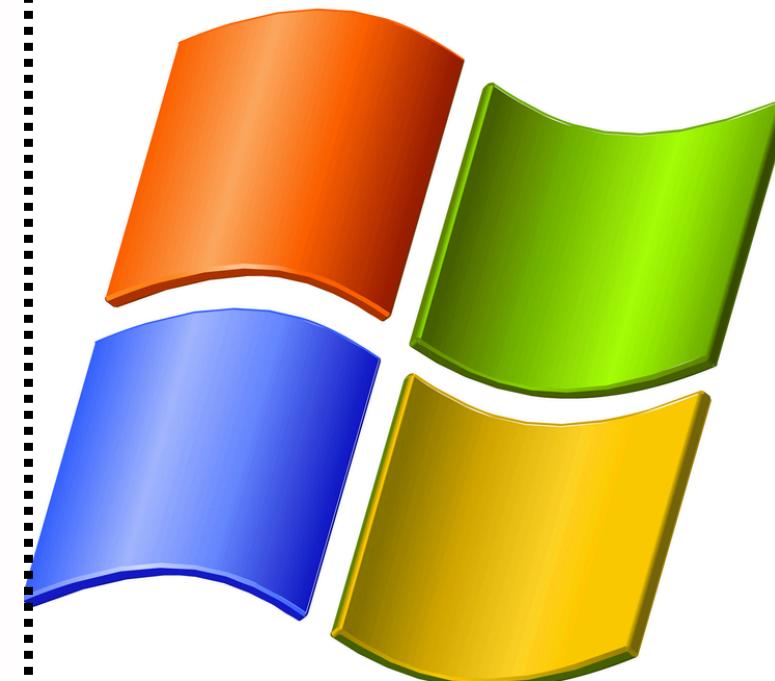
→ Cuando multiples hilos comparten el mismo recurso, puede haber condiciones de carrera, por lo cual se tiene que evitar, utilizando técnicas de la exclusión mutua.

VENTAJAS DE UNA GESTIÓN CORRECTA DE HILOS



- Al descomponer una aplicación en varios hilos secuenciales que se ejecutan en quasi-paralelo, el modelo de programación se simplifica.
- La creación de un hilo es de 10 a 100 veces más rápida que la de un proceso.
- Los hilos no producen un aumento en el rendimiento cuando todos ellos están ligados a la CPU.

SMP EN DIFERENTES SISTEMAS OPERATIVOS



symbian
OS



WINDOWS EMBEDDED COMPACT 7 (WEC7)

→ Integra soporte para que varias CPUs trabajen en conjunto bajo un mismo sistema operativo.

SMP en diferentes SO

SYMBIAN

→ Implementa SMP con el objetivo de mejorar el rendimiento sin comprometer el consumo de energía.

MAC OS

→ Rediseñó su núcleo para ofrecer un soporte robusto y natural para SMP.

Windows Embedded Compact 7 (WEC7)

Cómo lo maneja:

- Utiliza interrupciones interprocesador (IPIs) para coordinar la comunicación entre CPUs.
- Reemplaza la desactivación de interrupciones por mecanismos de exclusión mutua (spin locks), con el fin de evitar conflictos en el acceso a recursos compartidos.
- Adapta contadores y temporizadores para que puedan ser gestionados desde cualquier CPU.

MacOS

Cómo lo maneja:

- Su base es el microkernel Mach, que organiza la planificación y ejecución de hilos (threads).
- Permite que múltiples hilos de una misma tarea se ejecuten en paralelo en diferentes CPUs.
- El I/O Kit y la capa BSD también cuentan con soporte multiprocesador.
- Si un hilo se bloquea, otros pueden continuar ejecutándose sin afectar el rendimiento global.

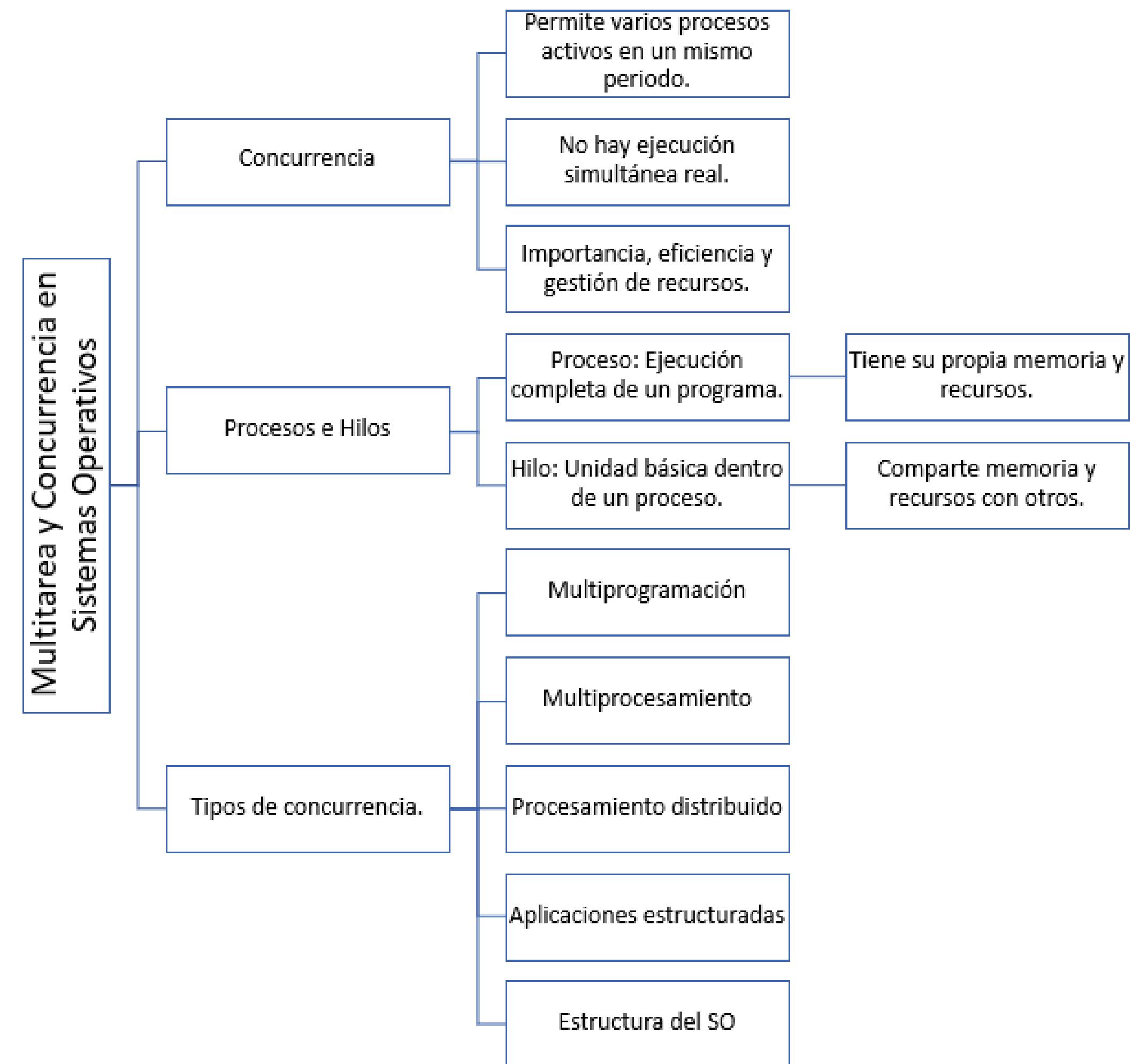
Symbian

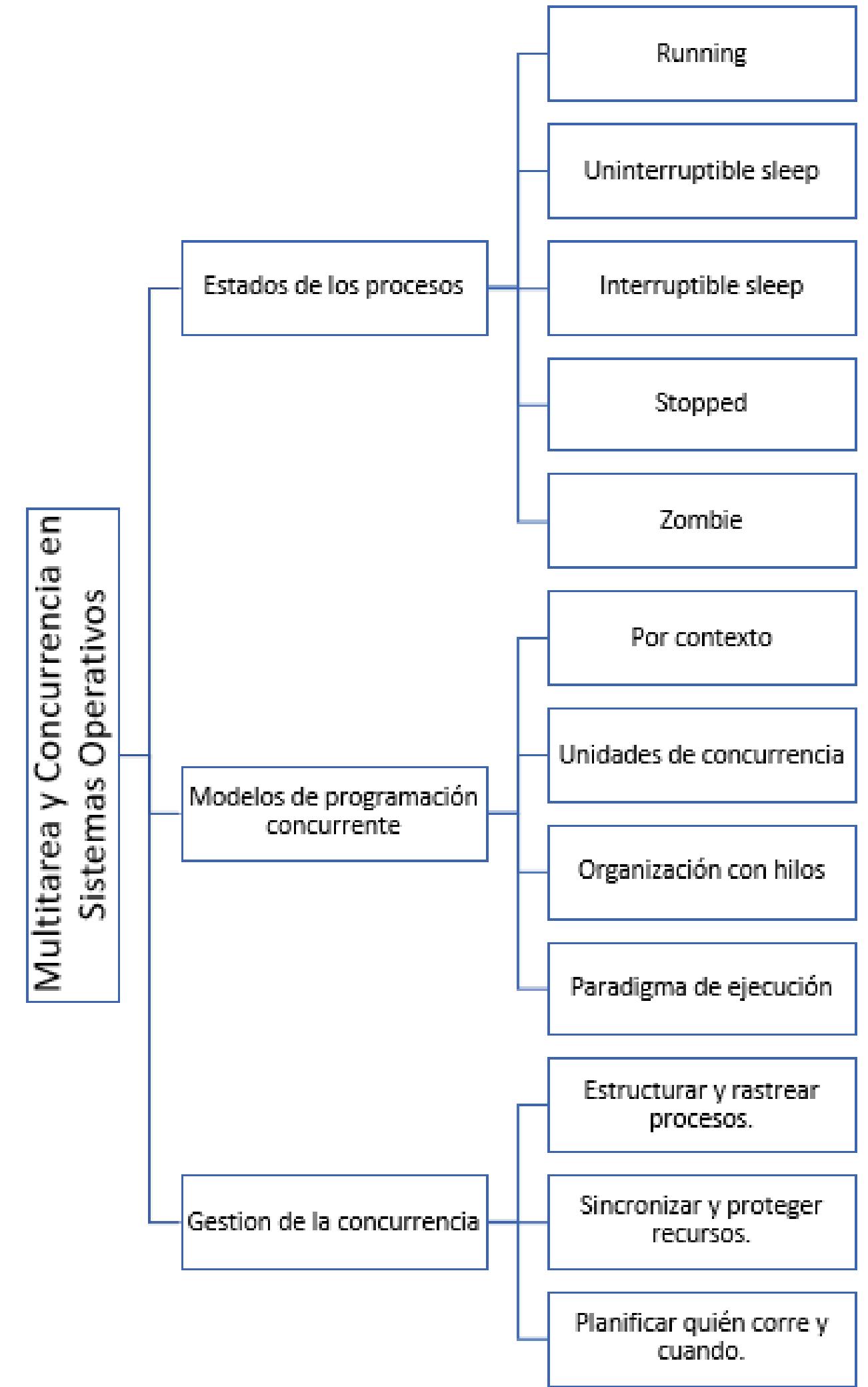
Retos principales:

- Los controladores de dispositivos y las rutinas de interrupción pueden ejecutarse en CPUs distintas, lo que puede provocar errores si no se sincronizan adecuadamente.
- Para evitar estos problemas, Symbian incorpora nuevas herramientas de sincronización y APIs de cancelación segura.

Estrategia de migración:

- Se recomienda primero probar los controladores en un solo núcleo.
- Posteriormente, se permite que los procesos se ejecuten en múltiples CPUs de manera gradual.





Multitarea y Concurrencia en Sistemas Operativos

Multiprocesamiento simétrico (SMP)

Varios procesadores comparten memoria común

Gestión de Hilos

Creación y terminación

Planificación

Sincronización

CONCLUSIÓN

Durante toda esta investigación pudimos entender que los sistemas operativos son sin duda la base de todo dispositivo moderno, ya que estos se encargan de organizar y poder administrar los recursos de una computadora. Uno de los puntos más importantes y que más se puede destacar es la multitarea, la cual en palabras sencillas, es la capacidad de poder ejecutar varias actividades al mismo tiempo, esto hace posible que un usuario pueda navegar en internet mientras escucha música o también mientras trabaja en algún documento sin que el sistema tenga alguna detención. Además, también hablamos sobre la concurrencia, que se refiere a como el sistema operativo maneja distintos procesos de una manera ordenada, alternando la ejecución y evitando así que interfieran unos con otros. También vimos la diferencia entre procesos y hilos, donde los procesos son rprogramas complejos en ejecución, mientras que los hilos son partes de un mismo proceso que permiten dividir el trabajo y poderlo hacer más eficiente.

CONCLUSIONES INDIVIDUALES

ALVARADO CANTÚ LESLY ELIZABETH

Lesly Elizabeth Alvarado Cantú 2177856 LTS Gpo: 004

Conclusión

En esta actividad fundamental me llevo mucho aprendizaje me ha ayudado a comprender más el funcionamiento de los sistemas operativos, eso es muy importante para así tener una mejor visión de lo que pasa en la computadora. Aprendí lo que significa la concurrencia, que es una prioridad del sistema operativo, esto significa que varios procesos están activos dentro de un mismo periodo y por consecuencia también se deben conocer otros conceptos como los procesos, hilos, estados, transiciones, multiprocесamiento, entre más conceptos. También conjuntamente fundamental 1 se va complementando la información para así tener el conocimiento más claro y me gusta mucho poder entender más como se ve todo el proceso detrás de la pantalla de la computadora.

ÁVILA IGNACIO JESÚS EMILIANO

Jesus Emiliano Avila Ignacio IAS 2118747
Conclusión

La concurrencia me ayudo a entender los procesos Fundamentales de los Sistemas Operativos, así como su historia y de donde surgieron me dio a entender como los procesos e hilos, SMP, Micro nucleos y los Estados de procesos e hilos ,entre otros más .pueden sincronizarse y permitir el uso del Sistema operativo y comunicación Maquina- Usuario y como se dan estos procesos basados en la concurrencia .

BALDERAS GARCÍA ANA VICTORIA

Balderas García Ana Victoria 2106047 IAS
Conclusión A.F.2

En este proyecto pudimos observar que los sistemas operativos son la base de cualquier dispositivo porque organizan y administran los recursos. Una de sus funciones principales es la multitarea: manejar varios procesos al mismo tiempo (aunque no siempre de forma simultánea).

También aprendimos que un proceso es básicamente un programa en ejecución, y un hilo es como una mini-tarea dentro de ese proceso que permite hacer varias cosas a la vez.

El sistema operativo se encarga de tres cosas clave en la concurrencia:

1. Seguir la pista a procesos e hilos.
2. Evitar choques cuando varios comparten recursos (usando semáforos, bloqueos, etc.)
3. Decidir quién corre primero con el scheduler.

También existen distintos tipos de concurrencia (como multiprogramación, multiprocesamiento o procesamiento distribuido) que ayudan a aprovechar mejor los recursos y dar mayor rendimiento.

Por último, se mencionan dos arquitecturas importantes:

- SMP (Multiprocesamiento Simétrico): varios procesadores compartiendo memoria y tareas, más balanceados y tolerantes a fallos.
- MicroKernels: más seguros y estables porque separan servicios, pero con el riesgo de perder algo de rendimiento.

GARZA CRUZ SANTIAGO

Santiago Garza Cruz 2177955

Conclusión A.F.2

Realmente pude entender la importancia de la concurrencia y multitarea y la diferencia entre los dos. Esta actividad sirvió para entender los procesos de la concurrencia, los tipos, el como funciona en los diferentes SO y las ventajas y problemas que pueden causar los procesos concurrentes.

El como se dividen las tareas y los métodos que se utilizan para poder tratar los problemas que se presentan.

GUEVARA OCHOA EDUARDO MIGUEL

Eduardo Miguel Guevara Ochoa 2053056

Al momento de investigar sobre los estudios de los procesos, entendí mejor cómo funciona un sistema operativo por dentro. Me gustó porque al principio parecía algo técnico pero me di cuenta de que aplica a la vida cotidiana y se vuelve más fácil de recordar y explicar.

Me quedé con la idea de que, así como un proceso necesita estudios para ordenarse, nosotros también necesitamos organizar nuestro tiempo y tareas para ser más eficientes.

MÉNDEZ SÁNCHEZ MARCO ANTONIO

Marco Antonio Méndez Sánchez 2177912

Esta actividad fundamental 2 me permitió adquirir nuevos conceptos y un mejor entendimiento de como funcionan los sistemas operativos para llevar a cabo las tareas y procesos que realizamos en la computadora de una manera eficiente y rápida, a través de la concurrencia o la multitarea, en los permiten que los sistemas operativos ejerzan diferentes tareas o procesos simultáneamente.

Me quedó también claro la diferencia entre un proceso y un programa, así como también entiendo como funcionan los procesos y los hilos y el porque son tan importantes en los sistemas operativos. También me quedó claro algunos otros conceptos importantes de la concurrencia como la exclusión mutua, semáforos o los bloqueos, así también vi os como es que se llevan a cabo los procesos en diferentes sistemas operativos.

PÉREZ RODRÍGUEZ JAZMÍN

Conclusión AF2,

Jazmín Pérez Rodríguez 2103948

La actividad fundamental 2 abordó principalmente la concurrencia, definiendo y pasando por conceptos importantes como los de los procesos (que ya se habían visto en la actividad fundamental 1), hilos, Transacciones y condados etc. Anteriormente llevé la materia arquitectura de computadoras y debido a esto relacioné la concurrencia con el paralelismo de máquina, al principio resultó algo confuso, sin embargo a lo largo de este actividad logré comprender la diferencia entre estos dos conceptos. La concurrencia es la capacidad del sistema operativo de intercambiar entre procesos, considero que esta capacidad es crucial en el sistema operativo ya que permite operar mejor los recursos en su arquitectura.

RAMÓN LOPEZ ANTHONY JOEL

Anthony Joel Ramón López 2058255 | TS
Conclusion A.F.2

Considero que la concurrencia y la multitarea son conceptos que parecen sencillos de comprender, hasta que tienes que entender bien a que se refiere cada uno, ya que estos dos conceptos son más que fundamentales al momento de hablar de los Sistemas Operativos. Parece sencillo diferenciar la multitarea de la concurrencia pero realmente no lo es, es necesario analizar cada concepto y que implica cada uno de ellos, y sobre todo comprender que son pareados pero no iguales.

VALDEZ SILVA ÁNGEL KARIM

Conclusion Individual Practica #2 Angel Karim
Valdez Silva
204182A ITS.

Este tema no solo es importante, es para valorar la magnitud de los avances tecnologicos que facilitan nuestra vida diaria. Entender como los sistemas operativos organizan, monitorean y controlan los procesos me da una visión más clara de la importancia de la ingeniería detrás de lo que muchos veas hacer por lucro.

Gracias a estos mecanismos, los dispositivos actuales son capaces de ejecutar multiples tareas de manera eficiente, segura y flexible; optimizando los recursos de hardware y ofreciendo al usuario una experiencia estable y fluida.



REFERENCIAS BIBLIOGRÁFICAS

- ARPACI-DUSSEAU, R. H., ARPACI-DUSSEAU, A. C., & UNIVERSITY OF WISCONSIN-MADISON. (2014). OPERATING SYSTEMS: THREE EASY PIECES (1.A ED.). ARPACI-DUSSEAU BOOKS.
- IBM. (S. F.). ¿QUÉ ES UN SISTEMA OPERATIVO? COMPONENTES DE UN SISTEMA OPERATIVO. IBM MÉXICO.
- LEZCANO, M. (2018). FUNDAMENTOS DE SISTEMAS OPERATIVOS: ENTORNOS DE TRABAJO.
- MALDONADO, R. (2024, 25 NOVIEMBRE). PARTES DE UN SISTEMA OPERATIVO: GUÍA COMPLETA 2025. KEEPCODING.
- SANTANA PÉREZ, F. J., QUESADA ARENCIBIA, A., SANTOS ESPINO, J. M., & CANDELA SOLA, S., CON CONTRIBUCIÓN DE GARCÍA RUBÉN, C. (2007). FUNDAMENTOS DE SISTEMAS OPERATIVOS: TEORÍA Y EJERCICIOS RESUELTOS. EDICIONES PARANINFO, S.A.
- SILBERSCHATZ, A. (2006). FUNDAMENTOS DE SISTEMAS OPERATIVOS (7.^a ED., EDICIÓN EN ESPAÑOL). MCGRAW HILL EDUCACIÓN.
- SISTEMAS OPERATIVOS: UNA GUÍA DE ESTUDIOS. (2014). DR. LUIS CASTELLANOS.
- TANENBAUM, A. S. (2003). SISTEMAS OPERATIVOS MODERNOS (EDICIÓN EN ESPAÑOL). PRENTICE HALL.
- WOLF, G., RUIZ, E., BERGERO, F., & MEZA, E. (S. F.). FUNDAMENTOS DE SISTEMAS OPERATIVOS (EDICIÓN LIBRE EN ESPAÑOL). DISPONIBLE BAJO LICENCIA CREATIVE COMMONS.
- ZETRA SERVICES. (S. F.). COMPONENTES DE UN SISTEMA OPERATIVO. ZETRA SERVICES.

Muchas

GRACIAS