

CAPÍTULO 2

FUNCIONES DEL SISTEMA OPERATIVO

Los Procesos



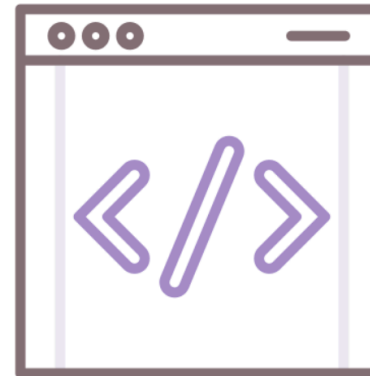
Definición y concepto básico de procesos

¿Qué es un proceso?

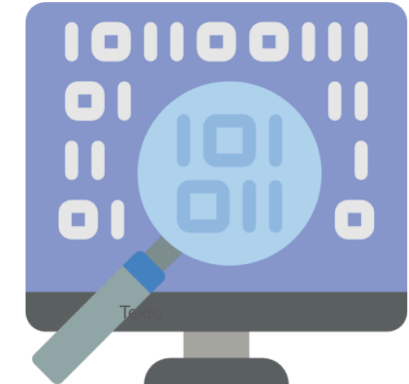
Un proceso se define como **un programa en ejecución** por parte del usuario o del sistema.

Recursos que necesitan los procesos:

- 🕒 Tiempo de CPU
- 🔧 Memoria
- 📁 Archivos
- 💻 Dispositivos de E/S



PROGRAMA



PROCESO

Características fundamentales de los procesos

Cada proceso tiene características esenciales:



Identificación única

Asignación de un **PID** (Process Identifier)



Propiedad

Identificador del usuario propietario que lo inició



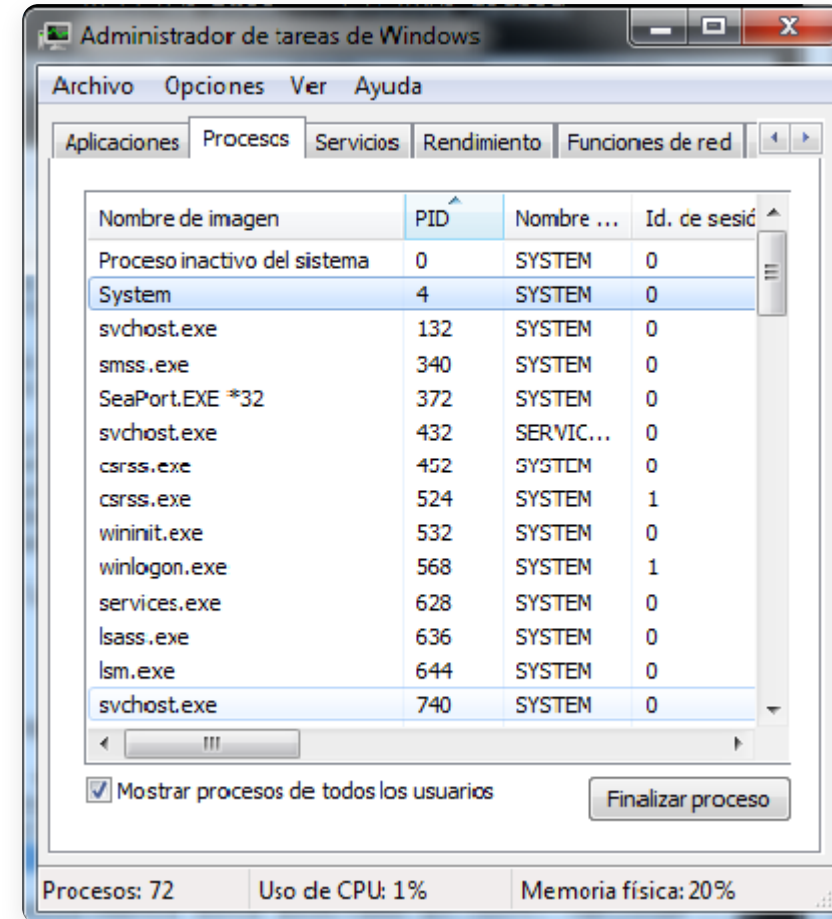
Prioridad

Nivel de prioridad para determinar qué proceso se ejecuta primero



Recursos

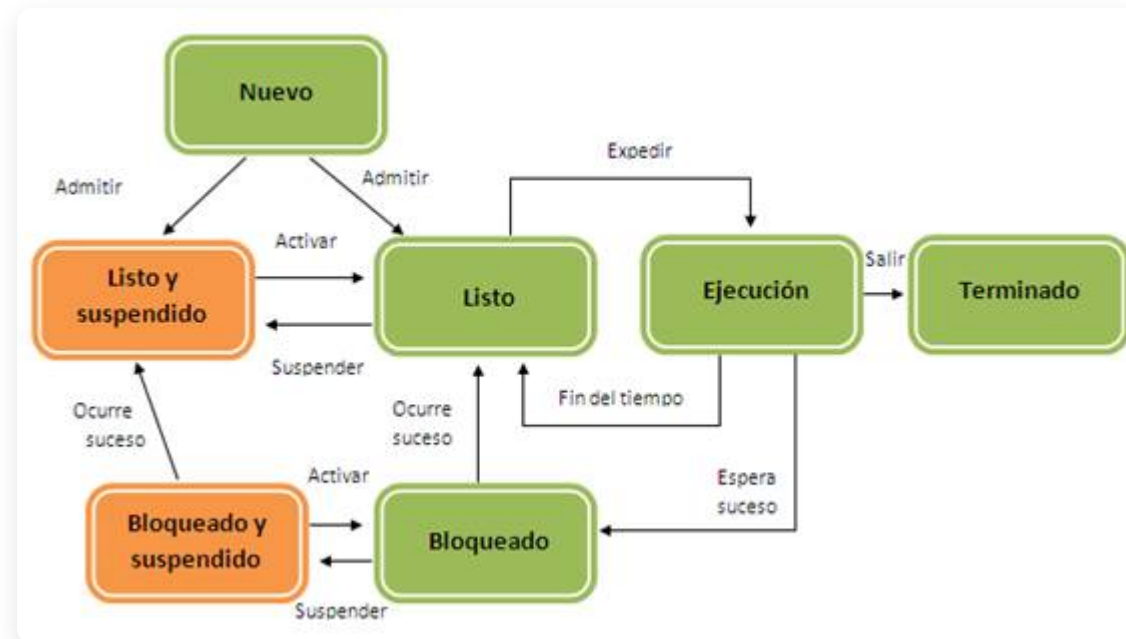
Consumen y gestionan recursos del sistema durante su ejecución



Administrador de Tareas mostrando procesos con sus PIDs y recursos asignados

Estados de un proceso

Ciclo de vida de un proceso



Nuevo/Creado

Proceso en creación, se le asignan recursos iniciales



Preparado/Listo

Esperando a ser asignado a la CPU para ejecución



Ejecución

Proceso actualmente en ejecución en la CPU



Bloqueado/Espera

Esperando un evento o recurso para continuar



Terminado

Proceso ha finalizado su ejecución

El flujo normal de ejecución sigue la secuencia: **Creado** → **Preparado** → **Ejecución** → **(Bloqueado/Espera)** → **Terminado**

Gestión de procesos

Operaciones esenciales del sistema operativo:



Creación y eliminación

Cuando un usuario inicia un programa o el sistema necesita ejecutar una tarea



Control del avance

Monitorización y gestión del progreso del proceso



Tratamiento de excepciones

Manejo de errores o situaciones inesperadas durante la ejecución



Asignación de recursos

Distribución eficiente de **procesador**, **memoria** y **dispositivos hardware**



Sistema de comunicación

Establecimiento de mecanismos para la comunicación entre procesos

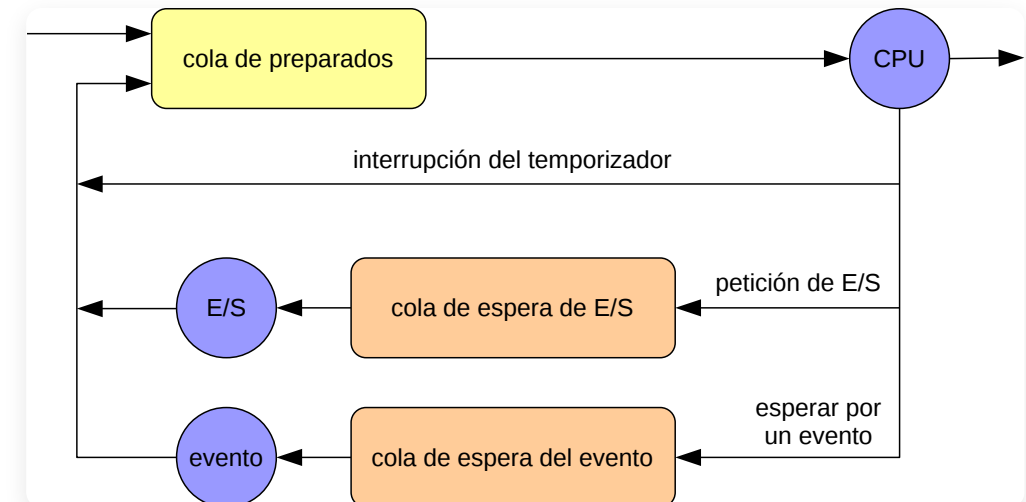


Diagrama de gestión de procesos: CPU, cola de listos, cola de E/S y cola de eventos

Procesos vs. Hebras (Threads)

Diferencias fundamentales:



Proceso

Unidad de ejecución **proprietaria de recursos**



Hebras (Threads)

Unidades de ejecución **dentro de un proceso** que comparten sus recursos

Ventajas de las hebras:

- ↔ Comunicación más fácil y eficiente
- ✓ No requiere cambio de contexto del proceso
- ⚙ Menor sobrecarga en creación y gestión

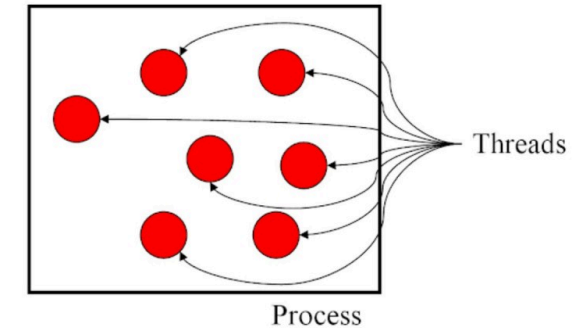
⚠ Limitación importante

La protección de recursos entre hebras del mismo proceso no es controlada por el sistema operativo

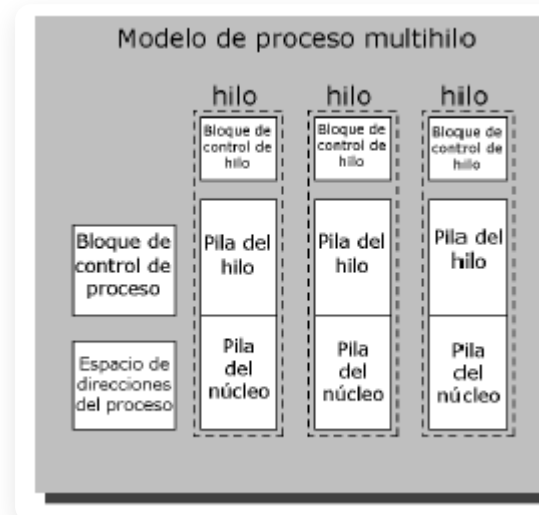


www.mentorestech.com

Process vs Threads



Un proceso puede contener múltiples hebras que comparten sus recursos



Modelo de proceso multihilo con bloque de control de proceso y múltiples hilos

Conceptos fundamentales:



Sistemas multiproceso

Permiten realizar **varios procesos simultáneamente**, ejecutando varias tareas al mismo tiempo



Sistemas multitarea

Existen **varios procesos que se ejecutan** en un mismo intervalo de tiempo

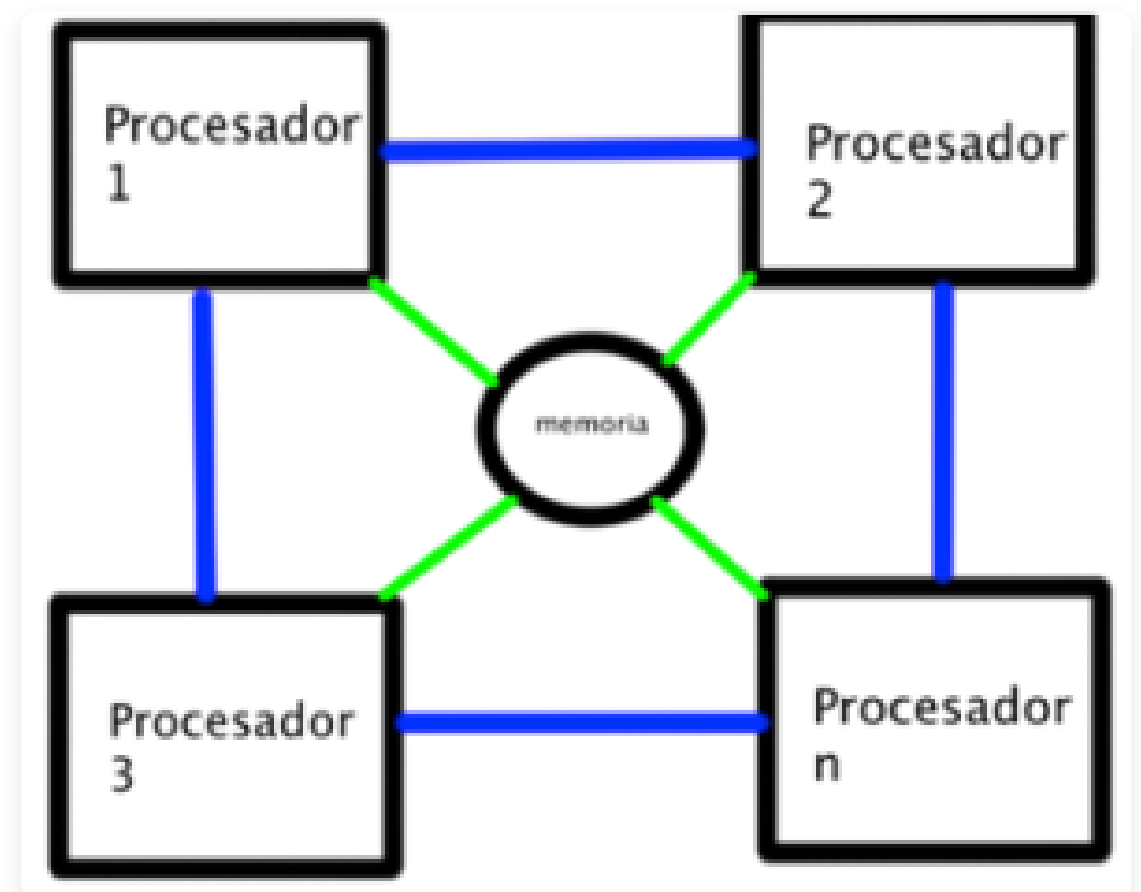
❖ Multiprogramación

Permite aprovechar de forma más eficiente los recursos del sistema, especialmente el tiempo de CPU

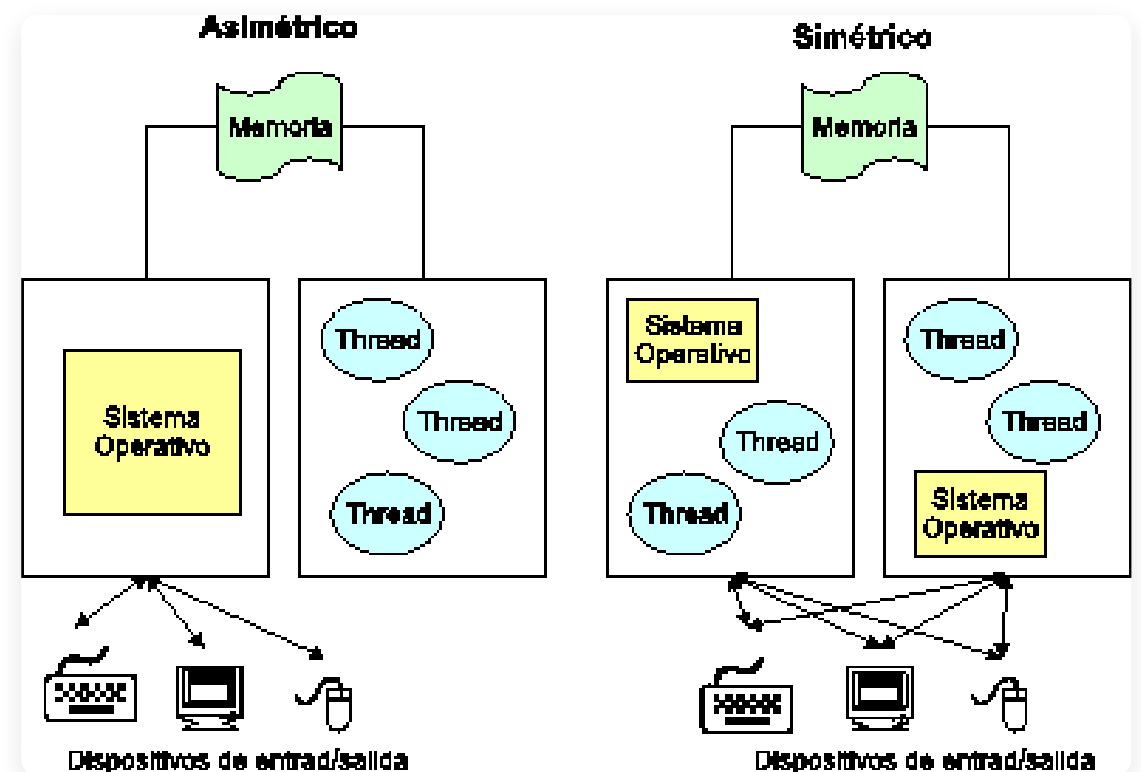
⌚ Optimización del tiempo

💻 Mejor uso de memoria

📈 Mayor rendimiento



Múltiples procesadores conectados a una memoria central



Sistemas de multiprocesamiento: asimétrico y simétrico

Planificación de procesos

Funciones del planificador (scheduler):



Asignación de CPU

Determina cómo se asigna la CPU a un solo proceso en cada momento



Cambio de proceso

Decide si se debe cambiar el proceso activo según la política de planificación



Gestión de contexto

Salva el entorno volátil del proceso actual y carga el del nuevo proceso



Ejecución

Cede el control al nuevo proceso elegido



El despachador se invoca cuando:



Un proceso finaliza



El SO decide detener el proceso activo



El proceso agota su quantum

Ejemplo práctico:

PROCESO	LLEGADA	DURACIÓN
P1	0	5
P2	4	2
P3	1	7
P4	2	1
P5	3	8



Tiempos de espera:

P1 (0-0)=0

P2 (21-4)=17

P3 (5-1)=4

P4 (12-2)=10

P5 (13-3)=10

Tiempos de retorno:

P1 = 5

P2 = 23

P3 = 12

P4 = 13

P5 = 21

Tiempo medio de espera: $(0 + 17 + 4 + 10 + 10) / 5 = 8,2$

Tiempo medio de retorno: $(5 + 23 + 12 + 13 + 21) / 5 = 14,8$

Ejemplo de planificación de procesos con tabla y diagrama de Gantt

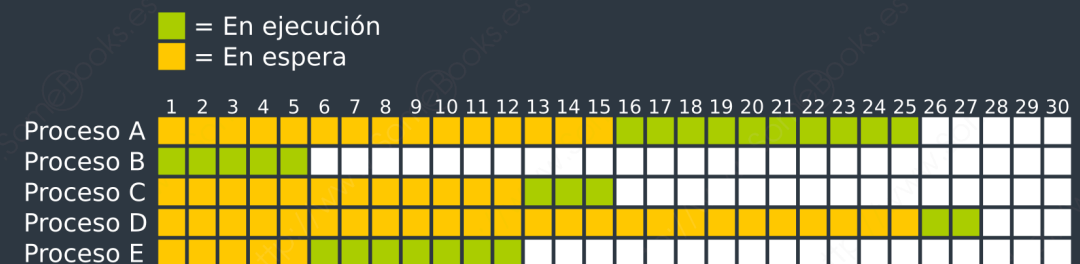


Diagrama de Gantt con la ejecución de cinco procesos a lo largo del tiempo

Ejemplo práctico de planificación

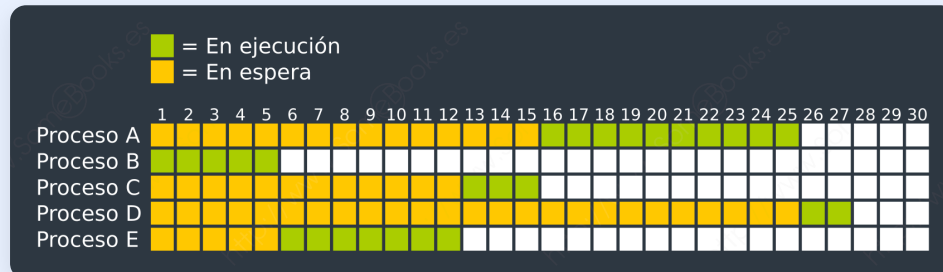
Aplicación del algoritmo FIFO (First In, First Out)

Procesos y tiempos

Proceso 1: 18 Unidades

Proceso 2: 6 Unidades

Proceso 3: 6 Unidades



Comparación de órdenes

El algoritmo **FIFO** ejecuta los procesos en el orden en que llegan. El orden de ejecución afecta significativamente al rendimiento del sistema.

Orden 1 → 2 → 3

$$(18 + 24 + 30) \div 3$$

24

Orden 2 → 3 → 1

$$(6 + 12 + 30) \div 3$$

16

💡 El segundo orden de ejecución reduce el tiempo medio de retorno en un **33%**, demostrando la importancia de una buena estrategia de planificación.

Procesos independientes vs. cooperantes

Tipos de procesos:



Independientes

Pueden ejecutarse, detenerse y rearancarse sin efectos negativos
Son **deterministas** y **reproducibles**

<> Ej: programa que calcula números primos



Cooperantes

Comparten su estado, su funcionamiento no es determinista
Pueden ser **irreproducibles** y generar situaciones inesperadas

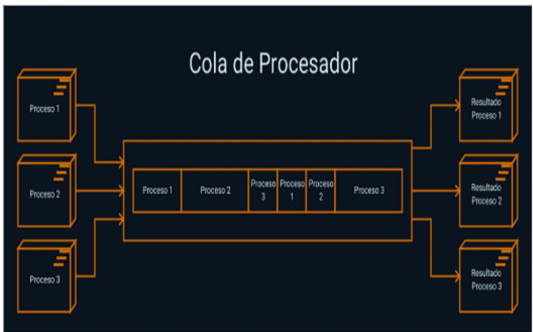
<> Ej: dos procesos que escriben en la misma pantalla



Riesgo en procesos cooperantes

Pueden producirse situaciones peligrosas donde se pierda el control del sistema

CONCURRENTE



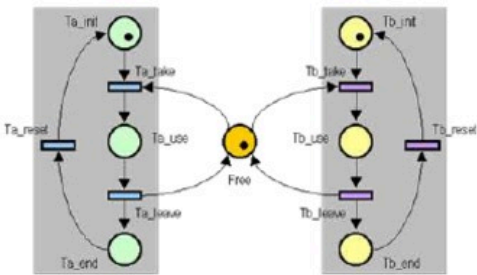
PARALELA



Comparación entre procesamiento concurrente y paralelo

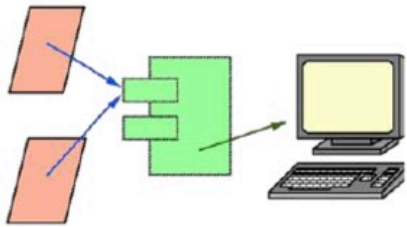
SISTEMAS OPERATIVOS

Concurrencia
Exclusión mutua y sincronización



Mg. Samuel Oporto Díaz

CONCURRENCIA



Conceptos de concurrencia, exclusión mutua y sincronización