

- **GESTIÓN DE PROCESOS**

# AGENDA

- Concepto de Proceso
- Planificación de Procesos
- Operaciones entre Procesos
- Comunicación interprocesos

# Concepto de Proceso

# ❑ Proceso

❑ Puede pensarse como un programa en ejecución ¿?

- ❑ Necesita recursos para llevar a cabo su tarea
- ❑ Los sistemas constan de una colección de procesos
- ❑ Todos estos procesos pueden ejecutarse en forma concurrente

# ❑ Proceso

- ❑ El SO es el responsable de la creación y eliminación de procesos del sistema y de usuario
- ❑ Planifica los procesos y provee de mecanismos para la sincronización, comunicación e interbloqueos en los procesos

# ❑ Proceso

- ❑ Los primeros sistemas informáticos permitían ejecutar un programa a la vez
- ❑ Este programa tenía control total sobre el Sistema y los recursos
- ❑ Actualmente, se pueden cargar varios programas en memoria y se ejecutan concurrentemente
- ❑ Esto, requiere mayor control y aislamiento

# ❑ Estructura del Proceso

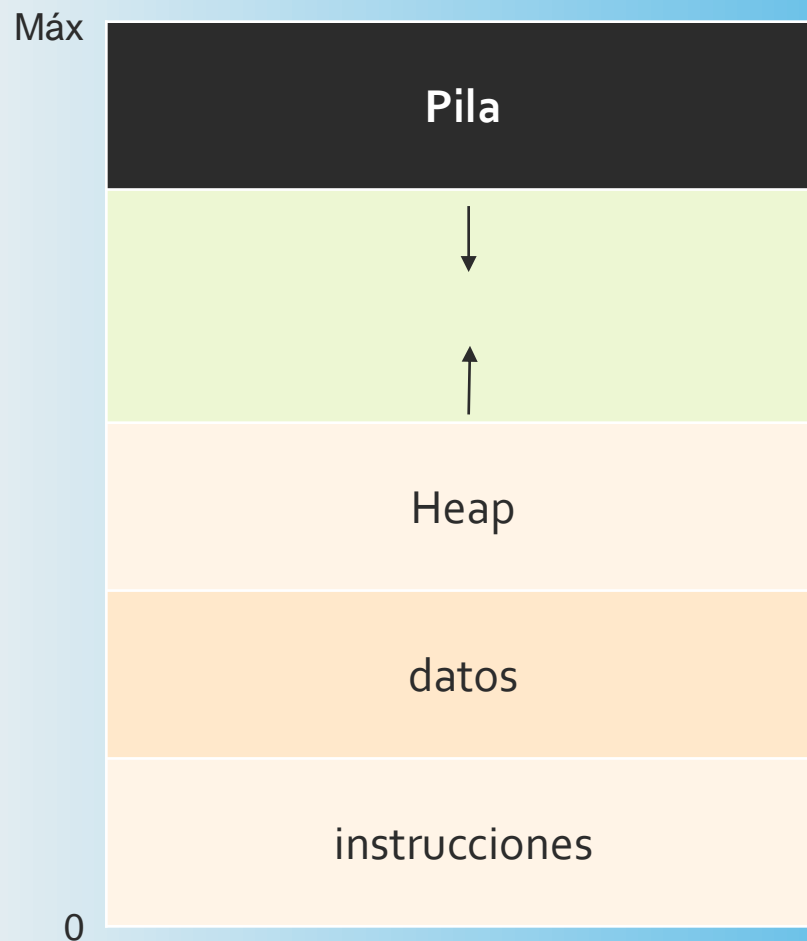
- ❑ El proceso son todas las actividades de la CPU
- ❑ Desde los trabajos de los sistemas de procesamiento por lotes, hasta los programas ejecutados en un Sistema de tiempo compartido
- ❑ Trabajo y proceso se usan indistintamente



# ❑ Estructura del Proceso

- ❑ Un proceso no es solo código de un programa
- ❑ Incluye la actividad actual representada por un contador de programa y los contenidos de los registros del procesador
- ❑ Incluye también la pila del proceso (con datos temporales)
- ❑ Incluye sección de datos (variables globales)
- ❑ Memoria (memoria asignada dinámicamente)

# ❑ Estructura del Proceso



Proceso en memoria

# ❑ Estructura del Proceso

- ❑ Un programa, por sí mismo, no es un proceso
- ❑ Un proceso es una entidad activa, con un PC y conjunto de recursos asociados
- ❑ Un programa se convierte en proceso cuando se carga en memoria
- ❑ Puede haber varios procesos asociados a un mismo programa
- ❑ Un proceso puede crear varios procesos a medida que se ejecuta

# Estado del Proceso

A medida que se ejecuta, cambia de estado

**FIGURE 2.2**

Simplified diagram of process states and transitions.

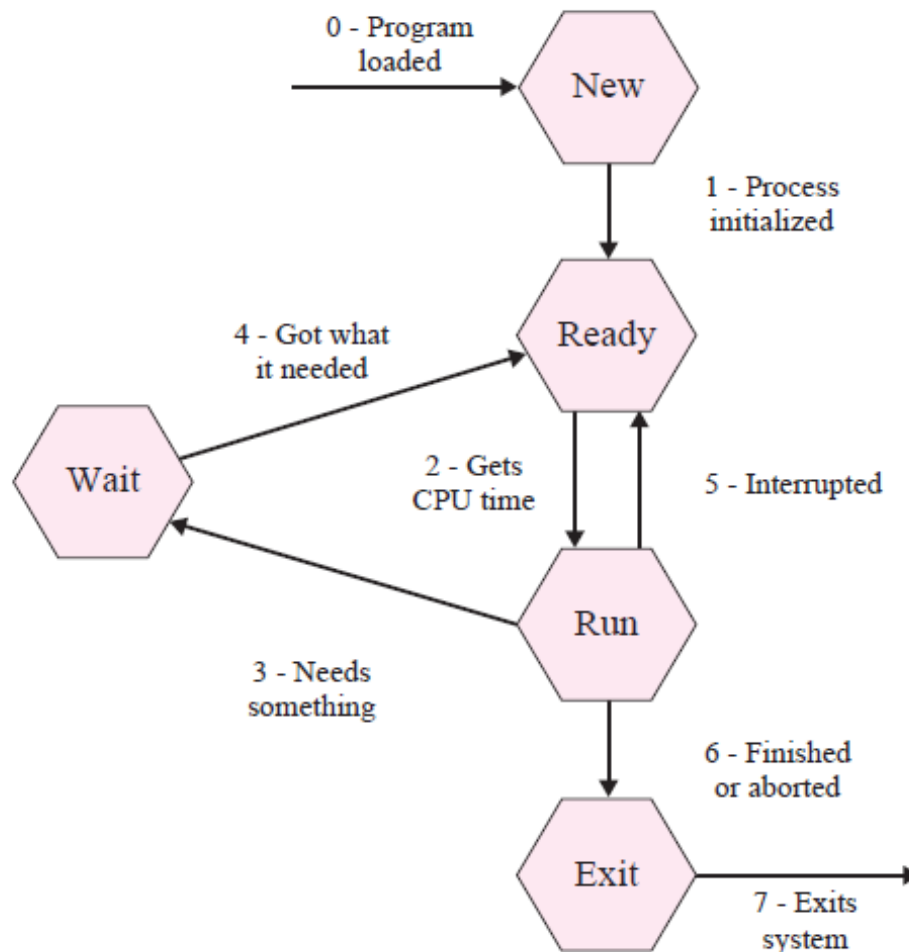
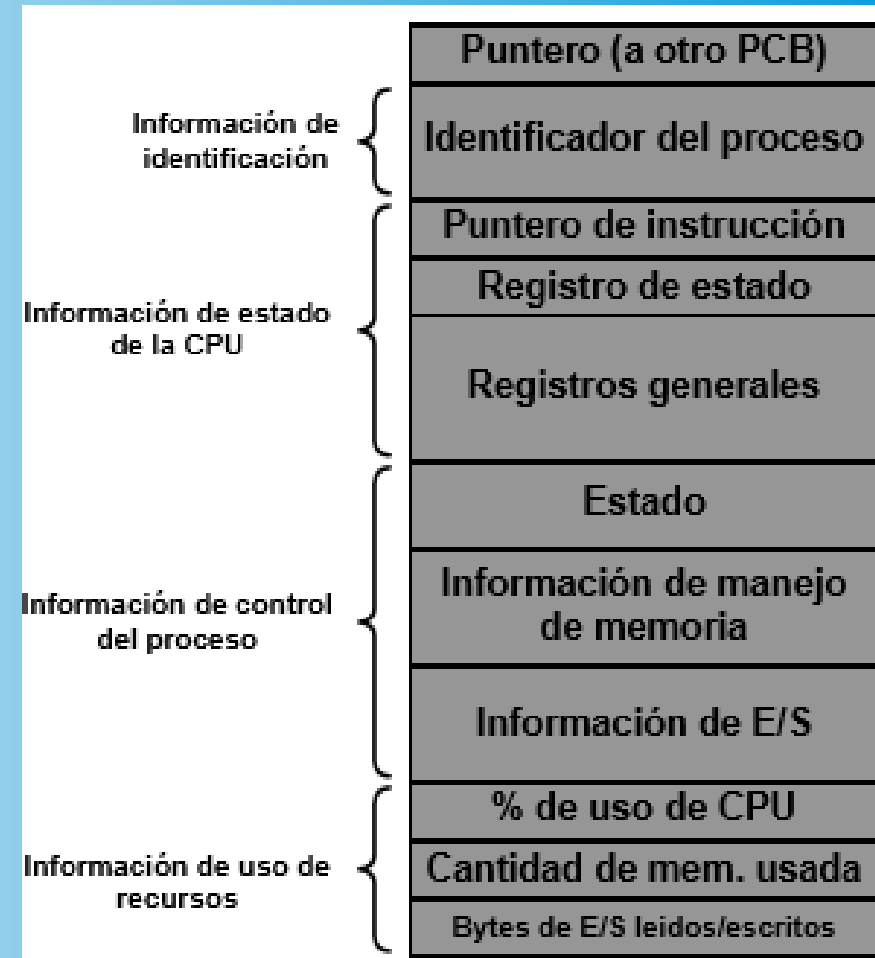


Diagrama de estados de un proceso

# ❑ Bloque de control de proceso

- ❑ PCB (Process Control Block) es una estructura de datos que permite al sistema operativo controlar diferentes aspectos de la ejecución de un proceso.



# ☐ Bloque de control de proceso

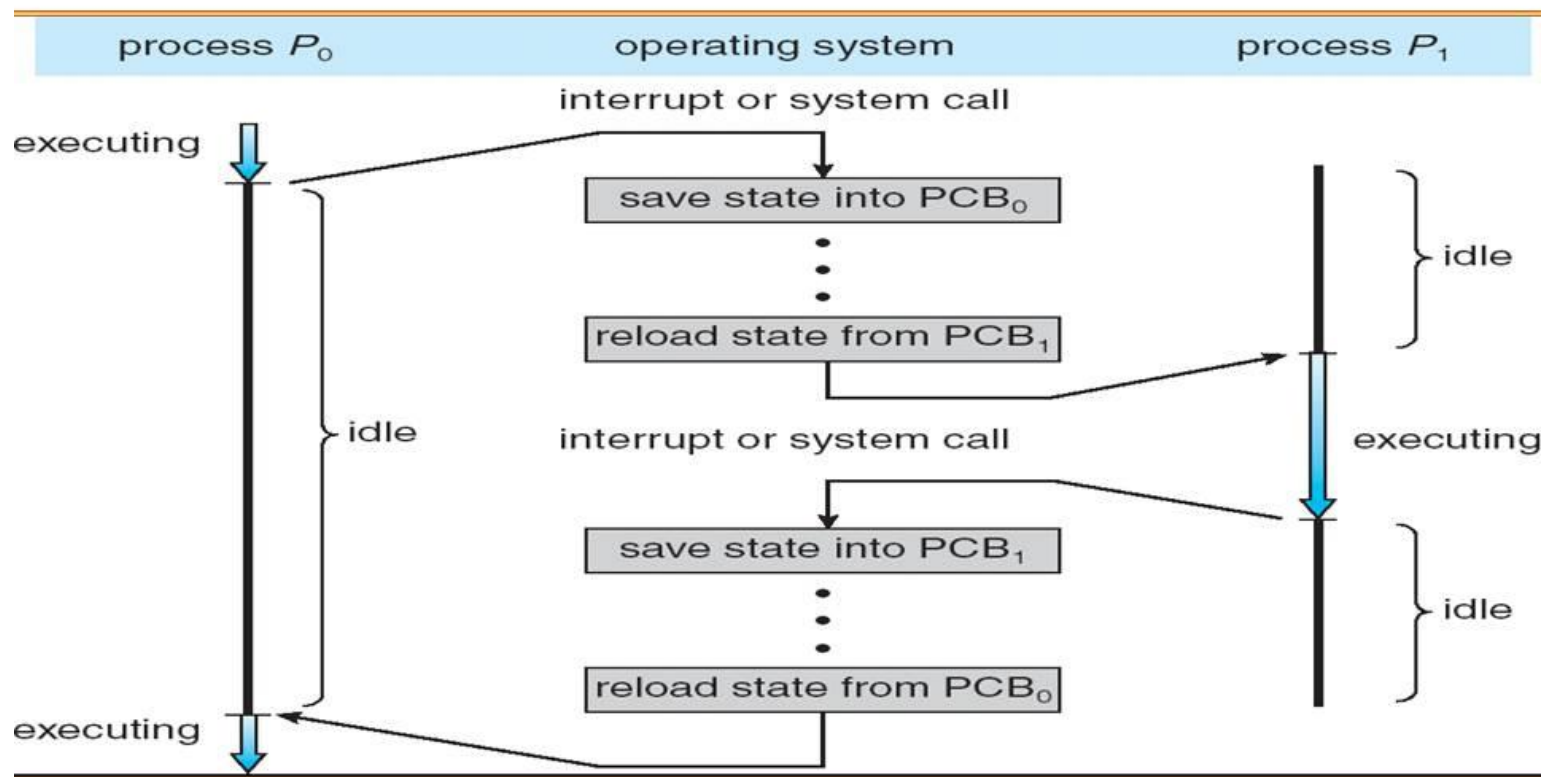
- ☐ La información típica mantenida en el PCB puede clasificarse en cuatro categorías:
  - ☐ Información de identificación
  - ☐ Información de estado de la CPU (estado suspendido)
  - ☐ Información de control del proceso
  - ☐ Información de uso de recursos:

# ❑ Bloque de control de proceso

## ❑ Un PCB contiene muchos elementos de información:

- Estado del proceso
- Contador de Programa (PC)
- Registros de la CPU
- Información de planificación de la CPU
- Información de gestión de memoria
- Información contable
- Información del estado de E/S

# ❑ Bloque de control de proceso





# ☐ Tipos de Proceso

- ☐ Procesos del Usuario
  - ☐ Aplicación del usuario
- ☐ Procesos del programa del sistema
  - ☐ Interactúan estrechamente con el SO
- ☐ Pocesos del SO
  - ☐ (Daemons)

➤ Concepto de Proceso

➤ Planificación de Procesos

➤ Operaciones entre Procesos

➤ Comunicación interprocesos

# Planificación de Procesos

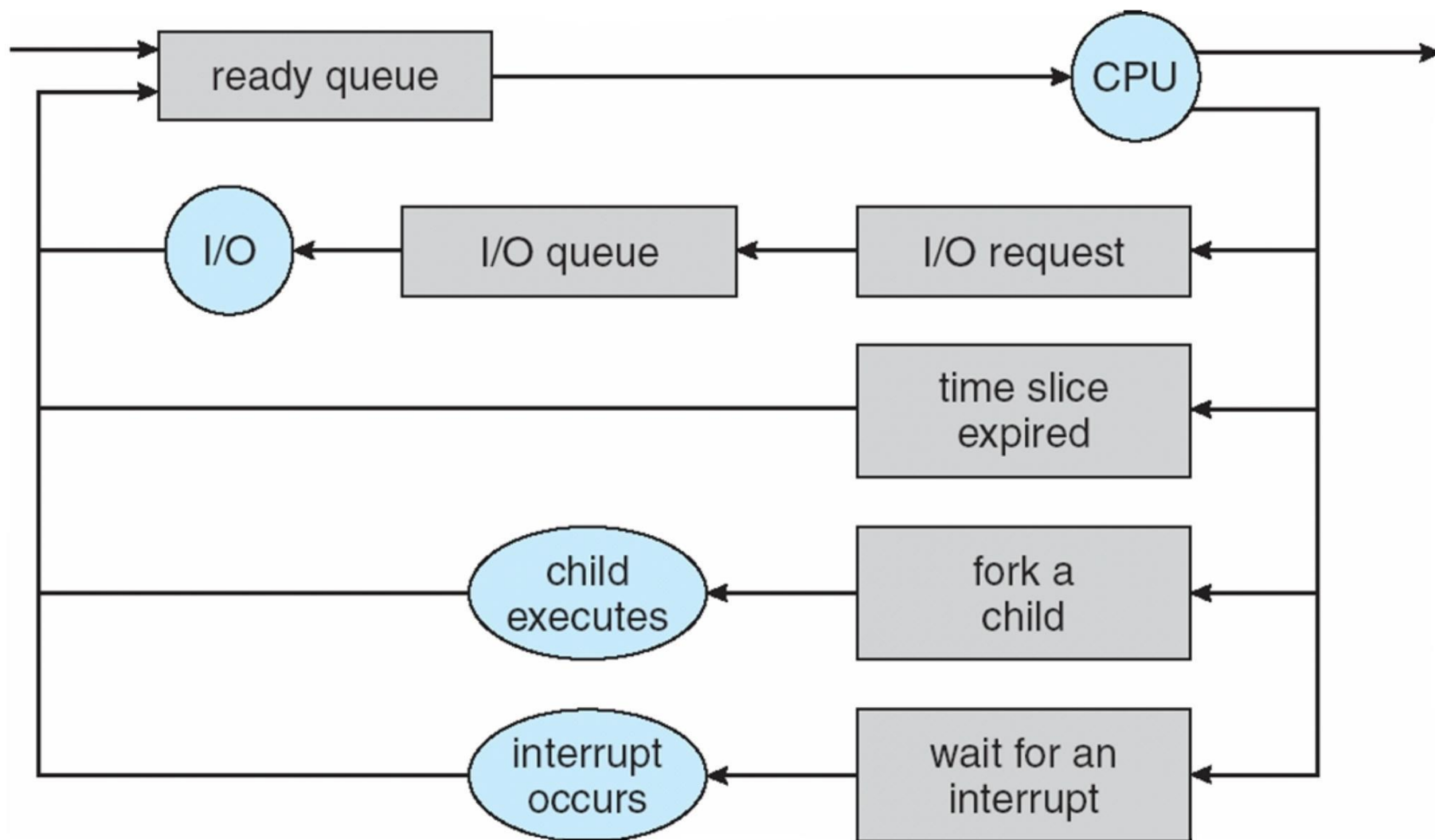
# ❑ Planificación de Procesos

- ❑ El objetivo de la multiprogramación es tener en ejecución varios procesos al mismo tiempo
- ❑ El planificador de procesos selecciona un proceso disponible para ejecutar el programa en la CPU
- ❑ En sistemas de un solo procesador, si hay muchos procesos que quieren ejecutarse deberán ser asignados

# ❑ Colas de planificación

- ❑ A medida que los procesos entran en el Sistema, se colocan en una cola de trabajos
  
- ❑ Existen varias colas.
  - ❑ Cola de listos
  - ❑ Cola de dispositivo (cada dispositivo tiene una)

# Diagrama de colas



# ❑ Planificadores

- ❑ Los procesos se mueven entre las distintas colas de planificación
- ❑ Los procesos se guardan en cola en un dispositivo de almacenamiento masivo
- ❑ El planificador de largo plazo selecciona procesos de esta cola y los carga en memoria
- ❑ El planificador a corto plazo o de la CPU selecciona procesos de entre los procesos que ya están preparados para su ejecución

# ❑ Planificadores

- ❑ La diferencia radica en la frecuencia de ejecución
- ❑ El planificador a corto plazo se ejecuta frecuentemente
- ❑ El de mediano plazo gestiona las situaciones en las que el Sistema queda escaso de recursos (swapping)
- ❑ El planificador de largo plazo controla la multiprogramación (número de procesos en memoria)



# ❑ Planificadores

- ❑ Los procesos se describen como limitados por la E/S o limitados por la CPU (CPU Intensivos)
- ❑ Es importante que el planificador a largo plazo seleccione una adecuada mezcla de procesos
- ❑ Los SO de tiempo compartido no utilizan planificación a largo plazo.

- ❑ Criterios para comparar algoritmos de planificación:
  - ❑ Equidad
  - ❑ Eficiencia de utilización de la CPU
  - ❑ Tiempo de retorno
  - ❑ Tiempo de respuesta
  - ❑ Tiempo de espera
  - ❑ rendimiento

# ❑ Planificadores

- ❑ FCFS: First Come, First Served)
- ❑ Shortest Job Next (SJN).
- ❑ Highest Response Ratio Next (HRRN)
- ❑ Planificación por Prioridad
  - ❑ Planificación por participación equitativa
- ❑ ¿Y que pasa si agrego el concepto de “expropiación”?

# ❑ Planificadores Apropiativos

- ❑ Round Robin
- ❑ Primero el menor tiempo restante
- ❑ Prioridad con apropiación
- ❑ Colas Multinivel

# □ Planificadores (Ejercicio)

Un sistema dispone de una CPU y dos dispositivos de entrada/salida: un disco y una impresora. Existen tres procesos que para ejecutarse necesitan emplear los distintos recursos del sistema tal y como aparece en esta tabla:

Proceso	Duración	Llegada	CPU	disco	CPU	impresora	CPU
A	8	0	2	1	1	2	2
B	8	1	3	1	2	1	1
C	8	2	1	2	2	1	2

Es decir, el proceso A necesita 2 unidades de tiempo la CPU, luego 1 unidad de tiempo el disco, luego 1 u.t. en CPU, luego 2 u.t. en la impresora y termina con 2 u.t. en CPU. Represente en un diagrama la planificación de estos tres procesos, considerando estas dos políticas de planificación de CPU y calcule el tiempo de espera medio y el tiempo de retorno medio en el que se ha incurrido:

a). FCFS.

b). Round Robin con  $q=2$  unidades de tiempo.

NOTA: El disco y la impresora son recursos no compartidos y se planifican siempre mediante un FCFS. Si varios procesos llegan a una cola al mismo tiempo, se considera que ha llegado antes el proceso con mayor número

➤ Concepto de Proceso

➤ Planificación de Procesos

➤ Operaciones entre Procesos

➤ Comunicación interprocesos

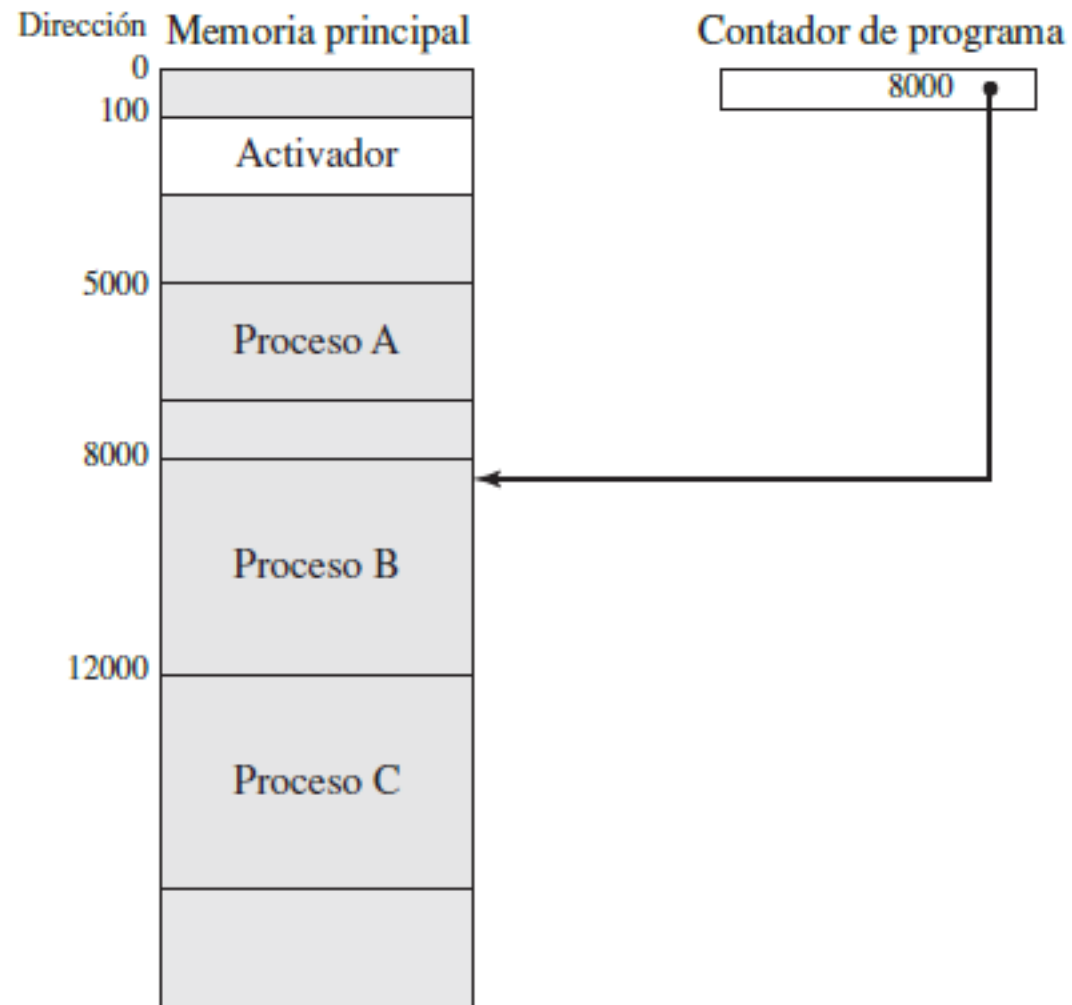
# Operaciones entre procesos

# ❑ Trazas

- ❑ Secuencia de instrucciones que se ejecutan para un proceso.
- ❑ El dispatcher (o activador) cambia el procesador de un proceso a otro.
- ❑ Se puede caracterizar el comportamiento de un determinado proceso, listando la secuencia de instrucciones que se ejecutan para dicho proceso.
- ❑ Se puede caracterizar el comportamiento de un
- ❑ procesador mostrando cómo las trazas de varios
- ❑ procesos se entrelazan.



# ❑ Traza



# □ Traza

5000	8000	12000
5001	8001	12001
5002	8002	12002
5003	8003	12003
5004		12004
5005		12005
5006		12006
5007		12007
5008		12008
5009		12009
5010		12010
5011		12011
(a) Traza del Proceso A	(b) Traza del Proceso B	(c) Traza del Proceso C

5000 = Dirección de comienzo del programa del Proceso A.

8000 = Dirección de comienzo del programa del Proceso B.

12000 = Dirección de comienzo del programa del Proceso C.

# Traza

1	5000		27	12004	
2	5001		28	12005	
3	5002				Temporización
4	5003		29	100	
5	5004		30	101	
6	5005		31	102	
		Temporización	32	103	
7	100		33	104	
8	101		34	105	
9	102		35	5006	
10	103		36	5007	
11	104		37	5008	
12	105		38	5009	
13	8000		39	5010	
14	8001		40	5011	
15	8002				Temporización
16	8003		41	100	
		Petición de E/S	42	101	
17	100		43	102	
18	101		44	103	
19	102		45	104	
20	103		46	105	
21	104		47	12006	
22	105		48	12007	
23	12000		49	12008	
24	12001		50	12009	
25	12002		51	12010	
26	12003		52	12011	
					Temporización

100 = Dirección de comienzo del programa activador.

Las zonas sombreadas indican la ejecución del proceso de activación;

la primera y la tercera columna cuentan ciclos de instrucciones;

la segunda y la cuarta columna las direcciones de las instrucciones que se ejecutan

# ❑ Operaciones entre procesos

- ❑ Los sistemas deben proporcionar un mecanismo para la creación y terminación de procesos
- ❑ Un proceso puede crear varios procesos nuevos, dando lugar a un árbol de procesos
- ❑ Los procesos se identifican por medio de un *pid*
- ❑ Un proceso hijo puede obtener recursos de su padre

# ❑ Operaciones entre procesos

- ❑ Cuando un proceso crea otro proceso nuevo, existen dos posibilidades en términos de ejecución:
  1. El padre continua ejecutándose concurrentemente con su hijo
  2. El padre espera hasta que alguno o todos sus hijos han terminado de ejecutarse
- ❑ También existen dos posibilidades en función del espacio de direcciones del nuevo proceso:
  1. El proceso hijo es un duplicado del padre
  2. El proceso hijo carga un nuevo programa

➤ Concepto de Proceso

➤ Planificación de Procesos

➤ Operaciones entre Procesos

➤ Comunicación interprocesos

# Comunicación interprocesos

# ❑ Comunicación interprocesos

- ❑ Los procesos pueden ser:
  1. Independientes
  2. Cooperativos
- ❑ Independiente cuando no puede afectar o verse afectado por otro proceso
- ❑ Cooperativos cuando puede afectar o verse afectado por los demás procesos



# ❑ Comunicación interprocesos

- ❑ La cooperación entre procesos requiere mecanismos de comunicación interprocesos

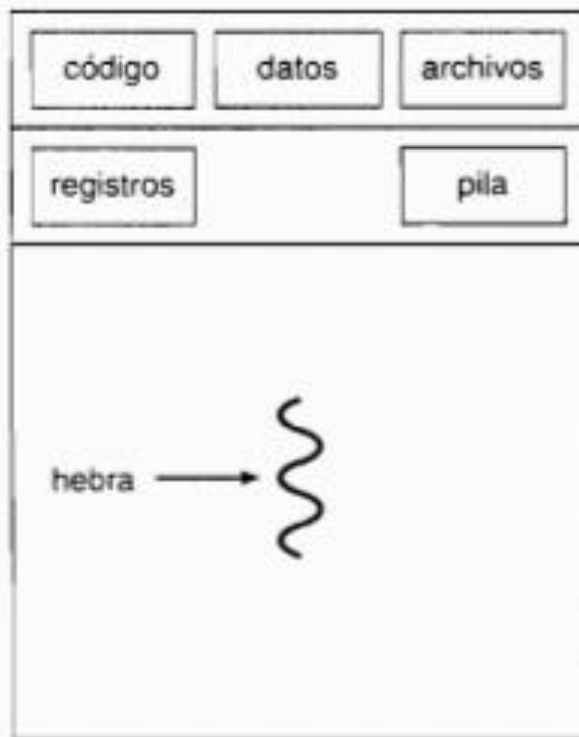
# ❑ Hilos (Hilos, Thread)

- ❑ Unidad de propiedad de los recursos: el proceso incluye un espacio de direcciones virtuales para mantener la imagen del proceso.
- ❑ Unidad de expedición: sigue un camino de ejecución que puede ser intercalada con la de otros procesos.
- ❑ Estas dos características son tratadas de manera independiente por el sistema operativo.

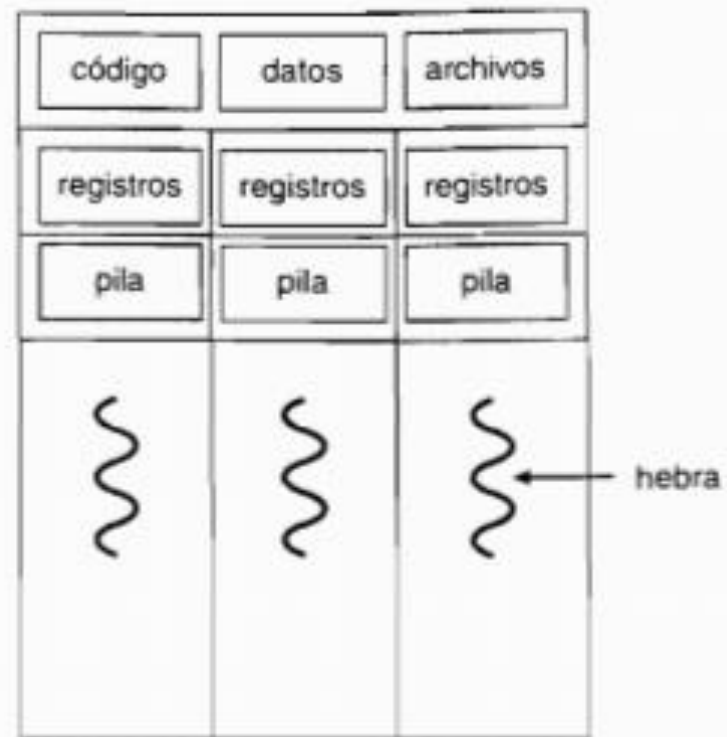
# ❑ Hilos (Hilos, Thread)

- ❑ Un hilo es una unidad básica de utilización de la CPU (Unidad de expedición)
- ❑ Esta formada por ID, un CP, conjuntos de registros y una pila
- ❑ Comparten con otros hilos la sección de código, la sección de datos y otros recursos del SO
- ❑ Varios hilos → más tareas a la vez
- ❑ Ejemplo: servidor web que recibe multiples pedidos

# □ Hilos

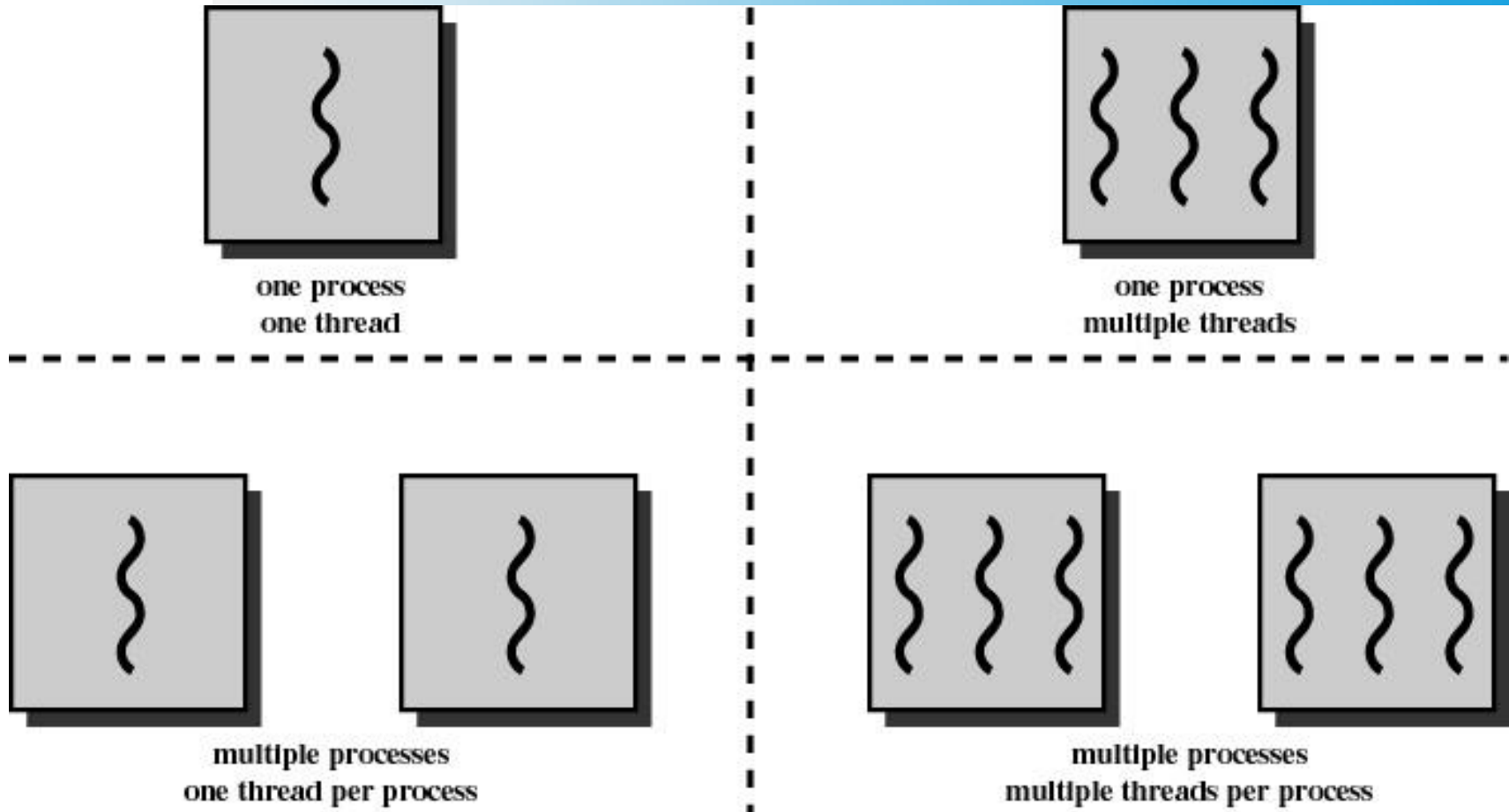


proceso de una sola hebra



proceso multihebra

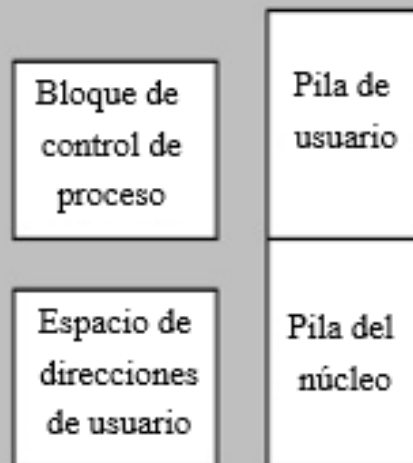
# ■ Hilos



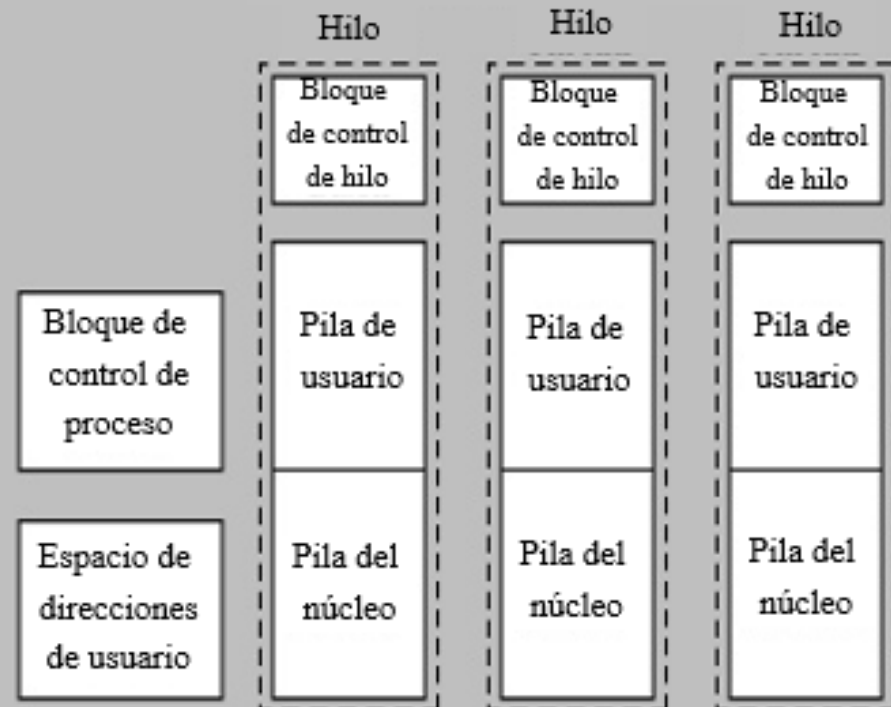
} = Instruction trace

# ■ Hilos

## Modelo de proceso monohilo



## Modelo de proceso multihilo



Modelos de proceso monohilo y multihilo.

# ❑ HILOS - VENTAJAS



## CAPACIDAD DE RESPUESTA

Continúa ejecutándose incluso si una parte del programa está bloqueado

Incrementa la capacidad de respuesta

Ej. Explorador web



## COMPARTICIÓN DE RECURSOS

Los hilos comparten memoria y recursos del proceso

La ventaja de compartir código y datos es que puede tener varios hilos de actividad diferentes dentro del mismo espacio de direcciones



## ECONOMÍA

Más económico crear y realizar cambios de contexto entre hilos

Por ej. En Solaris es 30 veces más lento crear un proceso que un hilo



## USO SOBRE ARQ. MULTIPROC.

Ventajas en configuraciones multihilos

Los hilos se ejecutan en paralelo en diferentes procesadores

Incrementan el grado de concurrencia

- ❑ La suspensión de un proceso implica la suspensión de todos los hilos de un proceso, puesto que todos comparten el mismo espacio de direcciones.
- ❑ La terminación de un proceso supone terminar con todos los hilos dentro de dicho proceso.



# ❑ Estados de los Hilos

- ❑ Hay cuatro operaciones básicas relacionadas con el cambio de estado en hilos:
  - ❑ Creación (Se crea un nuevo hilo).
  - ❑ Bloqueo.
  - ❑ Desbloqueo.
  - ❑ Terminación (Se liberan el contexto y las pilas)

# ❑ Niveles

## ❑ A nivel de usuario:

- ❑ La aplicación realiza todo el trabajo de gestión de hilos.
- ❑ El núcleo no conoce la existencia de hilos.

## ❑ A nivel de Kernel

- ❑ El núcleo mantiene la información de contexto del proceso y de los hilos.
- ❑ La planificación se realiza en función de los hilos.

# ❑ Aproximaciones combinadas

- ❑ Un ejemplo es Solaris.
- ❑ La creación de hilos se realiza en el espacio de usuario.
- ❑ La planificación y sincronización de los hilos se realiza en el espacio de usuario.

# □ Aproximaciones combinadas

Hilos : Procesos	Descripción	Sistemas de ejemplo
1 : 1	Cada hilo de ejecución es un único proceso con sus propios recursos y espacio de direcciones.	Implementaciones UNIX clásicas
M : 1	Un proceso define un espacio de direcciones y unos recursos dinámicos propios. Pueden crearse varios hilos que ejecuten en dicho proceso.	Windows NT, Solaris, OS/2, OS/390, MACH
1 : M	Un hilo puede migrar del entorno de un proceso a otro. Esto permite que un hilo se pueda mover fácilmente entre sistemas distintos.	Ra (Clouds), Esmerald
M : N	Combina los atributos de los casos M : 1 y 1 : M	TRIX

# Preguntas?

